Università Ca' Foscari di Venezia

Dipartimento di Informatica
Dottorato di Ricerca in Informatica

Ph.D. Thesis: TD-2006-2

Logics for Distributed Resources

Damiano Macedonio

Supervisor
Prof. Annalisa Bossi

Supervisor
Prof. Vladimiro Sassone

PhD Coordinator
Prof. Simonetta Balsamo

January, 2006

Author's Web Page: www.dsi.unive.it/~mace

Author's e-mail: mace@dsi.unive.it

Author's address:

Dipartimento di Informatica
Università Ca' Foscari di Venezia
Via Torino, 155
30172 Venezia Mestre – Italia
tel. +39 041 2348411
fax. +39 041 2348419
web: http://www.dsi.unive.it

To Renato, my father,
who left too soon.

Abstract

This Thesis studies logical characterisations of distributed systems for the purpose of describing resources in heterogeneous environments. The focus is more on the structure and the distribution of resources than their behaviour. The research follows two complementary strategies: a *proof theoretical approach,* not related to a particular formal model, and a *model theoretical approach,* deeply related to the choice of a formal model. The former consists in specialising a pure logical formalism to express properties in a distributed system; the latter consists in defining a new logic by considering a particular formalisation for distributed systems as a model, and by interpreting the logical constructs in such a model. To develop these two differing approaches, the Thesis is organised in two parts.

Part I introduces the 'Logic' from the basis by considering *Basic Logic:* a substructural logic whose aim is to find a structure in the space of the logics. Classical, Intuitionistic, and non-modal Linear logics are all obtained as extensions of Basic Logic in a uniform way. Basic Logic is taken as the fundament of a resource semantics, that is modularly extended to Intuitionistic Linear Logic, Linear Logic and Bunched Implications Logic. This semantics, along with its extensions, is sound and complete, and provides a theorem of semantical cut-elimination.

By adding *places*, or locations, to a Modal Intuititionistic Logic we define a model that well describes distributed systems. The semantics provided for this modal logic is sound and complete, and can be further specialised to satisfy the finite model property, thus proving the decidability of the logic.

Part II introduces *bigraphs*, which are a graphical model of computation in which both *locality* and *connectivity* are prominent. Bigraphs are establishing themselves a truly general (meta)model of global systems, and appear to encompass several existing calculi and models. This part of the Thesis is devoted to the introduction of *BiLog*, a new contextual and spatial logic based on bigraphs, that aims at achieving the same generality as a description language: as bigraphs specialise to particular models, we expect BiLog to specialise to powerful logics on these. In this sense we propose BiLog as a unifying language for the description of global resources, fortified by the positive preliminary results obtained by instantiating BiLog to well known spatial logics: Spatial Tree Logic, Spatial Graph Logic, Context Tree Logic, and a dynamic spatial logic for CCS. Another positive result in this direction has been obtained for semistructured data, by focusing on XML.

Sommario

Questa Tesi studia varie caratterizzazioni logiche per sistemi distribuiti col proposito di descrivere la struttura e la distribuzione di risorse in un ambiente eterogeneo.

La ricerca segue due approcci tra loro complementari: il primo si basa sulla teoria della dimostrazione e non è correlato ad un particolare modello; il secondo, invece, sposa la teoria dei modelli ed è, pertanto, profondamente legato alla scelta del modello formale.

Con il primo approccio viene specializzato un formalismo puramente logico in modo da esprimere le proprietà di un sistema distribuito. Con il secondo, invece, viene definita una nuova logica che si basa ed utilizza come modello un particolare formalismo in grado di descrivere sistemi distribuiti.

Strutturalmente, quindi, la Tesi risulta divisa in due parti, ciascuna corrispondente ad uno dei due diversi approcci descritti.

La prima parte introduce la 'Logica' partendo dalla definizione della *Logica di Base.* Si tratta di una logica sottostrutturata proposta con l'intento di trovare un fondamento comune nello spazio delle logiche. Ed infatti da essa discendono la Logica Classica, la Logica Intuizionistica e quella Lineare (non modale) che ne costituiscono un'estensione uniforme.

Nel primo capitolo la Logica di Base è assunta come fondamento di una semantica di risorse che viene poi modularmente estesa alla Logica Lineare Intuizionistica, alla Logica Intuizionistica e alla Bunched Implications Logic. Per tale semantica e tutte le sue estensioni sono provati dei teoremi di validità, completezza ed eliminazione semantica dei tagli.

Il secondo capitolo considera una Logica Modale Intuizionistica, vi introduce il concetto di *locazione* e la interpreta in un modello formale che descrive la distribuzione di risorse. Tale modello soddisfa un teorema di validità e completezza e può essere raffinato in modo da garantire la proprietà del modello finito, che porta alla decidibilità della logica stessa.

La seconda parte della tesi considera i *bigrafi*, un modello grafico di computazione che esprime sia la *località* che l'*interconnettività* tra risorse. I bigrafi si stanno dimostrando un (meta)modello generale per sistemi distribuiti ed estendono vari calcoli e modelli già esistenti. Questa parte della Tesi è dedicata alla definizione di *BiLog*, una nuova logica contestuale e spaziale che mira, come linguaggio descrittivo, a raggiungere la stessa astrazione dei bigrafi. Infatti, così come questi generalizzano un particolare modello, ci aspettiamo che BiLog possa generalizzarne la corrispondente logica. In questo senso,

proponiamo BiLog come un (meta)linguaggio generale e unificante per la descrizione di risorse globali, forti dei risultati preliminari ottenuti con la codifica in BiLog di alcune delle logiche spaziali già conosciute: la Logica Spaziale per alberi, per grafi, per contesti e una logica spaziale dinamica che descrive il CCS. Considerando in particolare XML, abbiamo ottenuto un ulteriore risultato positivo in tale direzione per ciò che concerne i dati semistrutturati.

Acknowledgments

Writing is commonly regarded as a solitary occupation, but for me writing about research is a team effort, and I owe a great debt to several people who have very generously given me their time and input, on so many occasions.

My very first thanks go to my two supervisors. I am grateful to *Annalisa Bossi*, who has been supporting and motivating me, especially when I was first approaching computer science, and when I was abroad, despite my lack of communications. I am deeply indebted to *Vladimiro Sassone*, without whom this Thesis would not exist. He gave me so many great ideas, that only a few of them could be pursued in this dissertation.

My deep gratitude is devoted to *Giovanni Sambin*, my M.Sc. supervisor and, first of all, friend. He introduced me to the fascinating world of research, and taught me how to develop my work in a "dynamic and constructive way, like everything in life should."

I thank my official referees, Didier Galmiche, Guy McCusker and David J. Pym, for their valuable feedback and their constructive criticisms in reviewing this Thesis.

Many thanks to my co-workers, from whom I received invaluable guidance and knowledge, and with whom I learnt how to conduct research: Rohit Chadha, Giovanni Conforti, Michele Donolato, Claudia Faggian, Riccardo Focardi, Carla Piazza, and Sabina Rossi. In particular, I single out two of them, who have been not only co-authors but also genuine friends. *Rohit* has always been optimistic. He was an encouragement to me, especially when birelational models turned out to be very tough and challenging. *Giovanni* was essential for the birth of BiLog. Since we met, he has been an inexhaustible source of ideas, practical perspectives and enthusiasm, every day… and every night!

I feel very privileged to have been a member of the Foundations of Computation Group at the University of Sussex. In the last two years I have appreciated good humour and discussions with all of the members. In particular, I thank Matthew Hennessy and, again, Guy McCusker for the interest they have shown for my work and their precious suggestions as members of my Ph.D. committee; Bernhard Reus for the discussions on logic and semantics; Philippe Bidinger for the discussions on bigraphs and all his funny stories; Alberto Ciaffaglione for his hospitality; Federico Cozzi for his advice on my laptop; Adrian Francalanza for his helpfulness, especially when I was a 'shy newcomer;' and Jan Schwinghammer for his friendship and his efforts in reading the early drafts of my Thesis deciphering my 'Italian English.'

Also the first years of my Ph.D. were very stimulating and fruitful, thanks to the

people I met at the Informatics Department in Venice. I mention Michele Bugliesi, who has always given me good advice, and all the (now mostly ex) Ph.D. students who made ‘Room 13’ a special place to stay: Chiara, Claudio, Fabio, Fabrizio, Giulio, Massimiliano, Matteo, Moreno, Ombretta, Silvia, and Valentoino.

My ‘extended’ family and friends have been a great source of strength and motivation. I thank Alberto and Lisa, Benvenuto and Rita, Claudio, Cristina, Dela and Rosa, Giulio and Vittoria. I send big kisses to my youngest supporters: Francesco and my godchild Giovanni.

I am deeply grateful to Graziella, my mother. She supported me in her ‘special’ way and always trusted every decision I made.

Finally, I express my deepest gratitude and all my love to Luisa, who has been bringing so much more into my life than I could ever dream. I thank her for the patience, trust and support she has given during these hard times, while I have been working over a thousand miles away.

I dedicate this Thesis to the memory of Renato, my father, who set the example for me, but who left too soon to share my achievements.

Damiano Macedonio
Brighton, January 2006

Contents

List of Tables

List of Figures

Preface

A significant part of this Thesis is the fruit of the two years I spent with the *Foundations of Computation* group of the Informatics Department at the University of Sussex. There, I was supported first by a Marie Curie fellowship (*'**DisCo**: Semantic Foundations of Distributed Computation,'* EU IHP project HPMT-CT-2001-00290), and then by a research fellowship from the European project *'**MyThS**: Models and Types for Security in Mobile Distributed Systems,'* EU FET-GC project IST-2001-32617.

At the University of Sussex, my supervisor was Prof. Vladimiro Sassone; in addition, Prof. Matthew Hennessy and Dr. Guy McCusker served on my Ph.D. committee.

The Relational Semantics and its extension to Intuitionistic Linear Logics in Chapter 1 were obtained as the result of joint research with Prof. Giovanni Sambin, and will appear in *The Journal of Symbolic Logic* [95]. The extension to Intuitionistic Logic was obtained as the result of joint research with Prof. Giovanni Sambin and Michele Donolato.

The contents of Chapter 2 were obtained as the result of joint research with Dr. Rohit Chadha and my supervisor Prof. Vladimiro Sassone. A preliminary version of this chapter appeared as a Computer Science Report at the University of Sussex [47]. The results will appear in *The Journal of Logic and Computation* [48].

The contents of Chapter 3 were obtained as the result of joint research with Dr. Giovanni Conforti and my supervisor Prof. Vladimiro Sassone. A preliminary version of this chapter appeared as a Computer Science Report at the University of Sussex [55]. BiLog was presented at *The International Colloquium on Automata, Languages and Programming (ICALP'05)* [56]. The results on XML were presented at *The Italian Symposium on Advanced Database Systems (SEBD'05)* [54].

Before embarking on the line of research that led to this Thesis, I studied information flow security in multilevel systems. The results, obtained jointly with my supervisor Prof. Annalisa Bossi and my co-authors Prof. Riccardo Focardi, Prof. Carla Piazza and Dr. Sabina Rossi, were published in [24, 25, 26, 27, 28, 29, 30, 31]. They are an integral part of my education and represent a significant portion of the work I carried out during my Ph.D. course. However, since they are not directly related to the main topic of the dissertation, they are not included herein to preserve the consistency of the exposition.

In Venice I was supported by a three years grant from the University Ca' Foscari of Venice, by the European project *'**MyThS**: Models and Types for Security in Mobile Distributed Systems,'* EU FET-GC project IST-2001-32617, and by the MIUR project *'**MEFISTO**: MEtodi FormalI per la Sicurezza e il TempO.'*

Introduction

In our daily life it is common to deal with distributed computing resources. Prime examples are smart cards [78] used in Subscriber Identity Module (SIM) cards or next generation credit cards, moving from card issuers to card holders and in and out of mobile phones or automatic teller machines (ATMs). In a distributed environment, in general, a user often employs programs which are sent or fetched from different sites to achieve his/her goals. Such programs may be run as a code to do simple calculation tasks or as interactive parallel programs that use resources located almost anywhere in the world. Accordingly, the ability to reason about the behavioural correctness of concurrent systems holding or using such resources, as well as the need of design and implementation tools, is playing an increasing prominent role.

This prefigures exciting future perspectives, but it poses enormous challenges to computer science. Innovative paradigms for information processing and task coordination are required. In fact, traditional correctness properties and methodologies for sequential systems are no longer applicable in the context of distributed and mobile systems. The lack of any kind of central control, the continuously mutating topology of the network, the lack of reliable information, and the absence of any intrinsically trustable object imply the necessity of designing new formal models to describe and reason about properties of distributed resources. This necessity has been recognised by several authors (for instance, we cite [33, 44, 92, 99, 104, 125]).

In a global computing model, resources are shared and distributed over the network, and agents are not tied to any specific system resource or to any geographical or logical network location. They need permission to cross administrative domains and to execute on remote locations using local resources, outside their control, as well as resources belonging to the domain of origin. Resource access control aims at providing guarantees of safety and authorisation. Safety corresponds to building safeguards against misuse of data leading to run-time failures. Authorisation provides an insurance that access to resources is granted only to principals that have obtained appropriate permissions. A reliable software, based on solid theoretical foundations, is a prerequisite for the success of the global computing infrastructure.

I.1 Objectives

Following the traditional approaches, the properties of concurrent systems and distributed resources can be expressed in terms of *semantics* (e.g. behavioural equivalences [97]), *logics* [17, 42, 92, 110, 125], or *types* [98]. Here we consider logic, and we study logical characterisations of distributed systems which are suitable to describe resources in heterogeneous environments. Our principal aim is to specify logics to characterise concurrent

systems. Our focus is more on the structure and the distribution of resources than their behaviour.

A logical formalism should simplify the definition and the verification of properties for a distributed system. A formula identifies a property which assumes meaning in a specific model. On the one hand, a formula may characterise a process class: the processes that enjoy the property expressed by the formula itself [33]. On the other hand, a formula may directly model the observed properties of resources in a distributed system [92, 110, 125]. Moreover a logic helps in deriving new properties as well as establish connections between different characterisations of process properties or resource distributions. In fact, a single logic may be evaluated in different models, hence the mutual relations among models can be investigated through the logic itself.

In order to develop a logic exploiting both the spatial characteristics and the interconnections of resources in a distributed system, we identify two complementary strategies as follows.

1. A *proof theoretical approach,* that consists of specialising a pure logical calculus in order to express properties in a distributed system, and in introducing a pure logical framework suitable to characterise heterogeneous environments. This approach is not related to a particular formal model: the logic should be based on the direct observation of heterogeneous systems by extrapolating their characteristics.

2. A *model theoretical approach*, that consists of defining a logical calculus by considering a formalisation for distributed systems as a model, and by interpreting the logical constructs in such a model. This approach is deeply related to the choice of the formal model; hence, as a major requirement, the model should be the most general possible to embrace the wide range of actual distributed systems.

On the one hand, in the group of candidate languages suitable for developing the proof theoretical approach there are:

- *Modal Intuitionistic Logic* [92, 106, 107], whose modalities are not interpreted *temporarily*, but *spatially*, hence describing 'located' properties;

- *Bunched Implications* [110] or *Separation Logic* [17, 125], which provide a powerful language to describe resources in distributed systems;

- *Spatial Logics* [33, 34], which provide a powerful language to formally describe the structure of concurrent processes.

On the other hand, for the model theoretical approach, there is a wide range of process calculi to choose as a formalism for distributed system. We focus on *Bigraphs* [90, 99], which are establishing themselves a truly general (meta)model of global computing, and appear to encompass several existing calculi and models, including Petri-nets [100], CCS [103], π-calculus [90], and ambients [88]. A logic founded on bigraphs aims at achieving the same generality as a description language: as bigraphs specialise to particular models, we expect that the logic in turn specialises to powerful logics on these, e.g. Spatial Logic [34] for π-calculus, and Ambient Logic [42] for ambients.

I.2 Proof Theoretical Approach

The relationships between computation and logic are regarded as fundamental, as perceived through paradigms of programming such as *proofs-as-programs* (Curry-Howard isomorphism, in functional programming), *proofs-as-computations* (logic programming), and *proofs-as-processes* (concurrent programming). Accordingly, the modelling of concepts, mechanisms and computations is approached by researchers through logic by using methods based on automatised construction of proofs and structural analysis in substructural and constructive logics.

Semi-structured data recently arose as a central concept in the exchange of information in computer science but adequate models and logics are necessary in order to represent, manipulate and reason about such data. One difficulty is to provide models that well reflect the structures and logics that are sufficiently expressive to represent data properties, and sufficiently restricted to decide if a given model satisfies a formula and if some properties entail other properties. In this context, recent works focus on separation logics [17, 36, 111, 125].

Separation Logic [111, 125] was initially introduced to support compositional reasoning about sequential programs which manipulate pointers. Separation Logic introduced the novel logical operation $\varphi * \psi$ (the *separating conjunction*) that asserts that φ and ψ are formulae holding for *disjoint* portions of the addressable storage. The prohibition of sharing is built into the operation.

The *Logic of Bunched Implications* [110, 122] generalises the idea of separation by dealing not only with pointers, but in general with resources. It models directly the observed properties of resources. The very first model of the logic is very natural: a *set* of resources, which can be *combined* and *compared*. Mathematically, this set-up is modelled through a *partial monoid* $(M, \cdot, e, \leq)$ that is *commutative* and *partially ordered*. Such a model is useful to obtain a Kripke-style semantics which freely combines multiplicative (intuitionistic linear) and additive (intuitionistic) conjunctions. The key of the semantics is the *sharing interpretation*. For example, the elementary semantics of the multiplicative conjunction

$$m \models \varphi_1 * \varphi_2 \quad \text{iff} \quad \text{there are } n_1 \text{ and } n_2 \text{ such that } m \leq n_1 \cdot n_2,\ n_1 \models \varphi_1,\ n_2 \models \varphi_2$$

is interpreted as follows: 'the resource m is sufficient to support $\varphi_1 * \varphi_2$ just in case it can be divided into the resources n_1 and n_2 such that n_1 is sufficient to support φ_1 and n_2 is sufficient to support φ_2.' The assertions φ_1 and φ_2 – think of them as expressing properties of programs – *do not share* resources. In contrast, in the semantics of the additive conjunction

$$m \models \varphi_1 \wedge \varphi_2 \quad \text{iff} \quad m \models \varphi_1 \text{ and } m \models \varphi_2$$

the assertions φ_1 and φ_2 *share* the resource m.

Bunches are the main feature of this logic. They appear in sequents instead of contexts (i.e., lists of formulae). Intuitively, bunches are trees of formulae. They are built

by using two ways of combining formulae: multiplicative (only commutative) and additive (with weakening and contraction). Thanks to the particular structure of bunches the calculus presents two native and independent operators adjoint to conjunctions: the multiplicative $-\!\!*$, and the additive $\rightarrow$.

The Logic of Bunched Implications has been extended in [17] with a modality for locations, and it can be viewed as a separation and a spatial logic: the multiplicative connectives naturally introduce the notion of resource separation and the location modality allows to gather resources in some locations and thus introduce a notion of spatial representation. Another modal extension has been recently proposed in [124], in a Hennessy-Milner style [81], to express properties of concurrent systems specified in a calculus of resources and processes.

In an *Intuitionistic Modal Logic*, modalities $\Box\varphi$ (always) and $\Diamond\varphi$ (in the future) can be interpreted not only *temporally*, but also *spatially*: as *everywhere* and *somewhere* in a distributed system. It follows that, in addition to considering *whether* a formula is true, the logic is dependent sensitive to *where* a formula is true. Recently, many authors have moved in this direction [92, 106, 107]: intuitionistic modal logics are used as foundations of type systems by exploiting the *propositions-as-types, proofs-as-programs* paradigm [75]. An instance of this was introduced in [92].

The language of the logic in [92] includes names, called *places*, and three 'spatial' modalities. Assertions are associated with places, and are validated in places. The modalities are suitable to infer whether a property is validated in a specific place p of the system ($@p$), or in an unspecified place of the system ($\Diamond$), or in every part ($\Box$). The modality $@p$ internalises the model in the logic and hence it can be classified as a hybrid logic. Although hybrid logics are usually studied in a classical setting, an intuitionistic natural deduction for such a logic was presented in [92], whose judgements mention the places under consideration.

As noted in [92], the logic can also be used to reason about distribution of resources in addition to serving as the foundation of a type system. Atomic formulae may be regarded as resources of a distributed system, and their placement in a particular place corresponds to the distribution of resources. That paper, however, does not present a model to match the usage of the logic as a tool to reason about distributed resources.

Spatial Logics display an active parallel line of development on reasoning about concurrent processes and semi-structured data [33, 34, 36, 42]. Their aim is to describe the behaviour and the spatial structure of concurrent systems, and they have been proposed as modal logics inspecting the *spatial* nature of models, as opposed to *temporal logics* inspecting exclusively the behaviour of models.

Spatial Logics tackle the problem of describing resources in a new way. On the one hand, Bunched Implications Logic was originally founded on a simple resource model and now it is approaching to more complex models to gather all the features of a distributed system. On the other hand, Spatial Logics originate from models which are already complex and whose purpose is to deeply describe the behaviour of an heterogeneous system. Essentially, these logics lift the constructors of the underlying model to the logical level,

hence obtaining new 'spatial' connectives that describe more in detail structural properties. The semantics of spatial logics is model dependant: different requirements on the model turn into different spatial connectives.

In [42] and in [33], ambients and π-calculus have been presented as models for particular spatial logics, and other process calculi will originate other kinds of spatial logics. Formulae describe properties of the concurrent system at a precise time, therefore they are modal both in space and in time. In particular, the spatial properties that can be expressed are essentially of two kinds: whether a system is composed of two or more subsystems (i.e. the 'Composition' of π-calculus), and whether a system restricts the use of certain resources to certain subsystems (i.e. the 'Restriction' of π-calculus). When ambient calculus is the underlying model, it is possible to express locality as well. It is then clear that Spatial Logics can describe in fine details the structure of processes, and this is what is required to meaningfully describe the distribution of processes and the use of resources over a network.

Basic Logic has been introduced in [129] with the aim of finding a structure in the space of the logics, hence it represents a foundational point to introduce logics in general. Although it was not originally introduced with the specific aim of describing resources, it can also be used to explain resource logics.

Until the beginning of the last century, there was only one logic, Aristotle's Classical Logic, which was conceived as a metaphysical absolute. Starting with Brouwer's revolution, which introduced Intuitionistic Logic, several different new logics have been developed. Each of them aimed to capture some of the distinctions which can be observed in a specific field of interpretation, but which are ignored by Classical Logic. Excluding intensional logics (which consider modalities), all such logics can be grouped under three main headings: intuitionistic logic (absence of principle of double negation), quantum logic (absence of distributivity between conjunction and disjunction), and relevance and linear logic (finer control of structural rules).

Although all of these logics are derived from Classical Logic, they have been considered as mutually incompatible. Basic Logic provides a common foundation and shows that they share a common structure. Classical, intuitionistic, quantum and non-modal linear logics, are all obtained as extensions in a uniform way. The sequent calculus is defined by introducing the *principle of reflection*. A logical constant obeys this principle if it is semantically characterised by an equation binding it with a meta-linguistic link between assertions, and if its syntactic inference rules are obtained by solving that equation. All the connectives of Basic Logic satisfy reflection. As an example, consider the additive conjunction $\&$. The common explanation of the truth of a compound proposition like $\varphi \,\&\, \psi$ is that $\varphi \,\&\, \psi$ is true if and only if φ is true *and* ψ is true. In this case the connective $\&$ reflects at the level of object language the link *and* at the meta-language. The semantical equivalence that we obtain in term of sequents is "$\Gamma \vdash \varphi \,\&\, \psi$ *if and only if* $\Gamma \vdash \varphi$ *and* $\Gamma \vdash \psi$" which is called *definitional equation* for $\&$. The inference rules for $\&$ are obtained by *solving* such an equation, and we say that $\&$ is introduced according to the principle of reflection.

I.3 Model Theoretical Approach

Among the approaches and theories for the modelling, analysis and verification of concurrent distributed systems, process algebras have received a lot of attention for their mathematical rigour and modelling flexibility. The development of their theory took off over twenty years ago from the seminal CCS [97] and other calculi [13, 86] and led to the emergence of important notions of behavioural equivalences that are now part of the common way of reasoning about concurrent systems. CCS was surpassed by the introduction of π-calculus [104], which introduces name mobility and, therefore, puts network topologies under the control of the processes themselves, thus achieving extra expressiveness.

As the focus of research on concurrency moved towards system distributed over wide-area networks, the communications offered by π-calculus became less than perfect a choice for foundational calculi. This led to the definitions of several versions of the π-calculus featuring different ways of process communication. A further step towards a faithful modelling of distributed computation was the focus on migration and location failures, as in Dπ [82] for example, which introduced process migration and access control. An original viewpoint was brought forward by the ambient calculus [44]. Ambients are administrative or physical boundaries that confine their contents (including executing threads) and carry them along when autonomously moving. Ambients introduced new concepts, such as boundaries that can be crossed or even removed.

Bigraphs [90, 99] are a recent emerging model for structures in global computing, which can be instantiated to model several well-known examples, including CCS [103], the π-calculus [89, 90], and the ambient calculus [88]. Bigraphs consist essentially of two graphs sharing the same nodes, which have a *control* for specifying their nature or behaviour. The first graph, the *place graph*, is tree structured and expresses a hierarchical relationship on nodes (viz. locality in space and nesting of locations). The second graph, the *link graph*, is an hyper-graph and expresses a generic many-to-many relationships among nodes (e.g. data link, sharing of a channel). The two structures are orthogonal, so links between nodes can cross locality boundaries. Thus, bigraphs express two kinds of separation: *structural* separation (i.e. separation in the place graph) and *name* separation (i.e. separation on the link graph). By combining these two notion we obtain a 'strong' version of separation for general bigraphs.

At the top level of the tree structure sit the *regions*. Inside nodes there may be *context holes* which are uniquely identified by ordinals. Place graphs can be seen as arrows over a symmetric monoidal category whose objects are finite ordinals, and $P : m \to n$ indicates a place graph P with m holes and n regions. Given two place graphs P_1, P_2, their composition $P_1 \circ P_2$ is defined only if the holes of P_1 are as many as the regions of P_2, and amounts to *filling* holes with regions, according to the number each carries. The tensor product $P_1 \otimes P_2$ corresponds to placing the two structures side by side.

Link graphs are arrows of a partial monoidal category whose objects are (finite) sets of names, X, Y. Given a link graph $W : X \to Y$, the set X represents the *inner* names and Y represents the set of *outer* names. The composition of link graphs $W_1 \circ W_2$ corresponds to *linking* the inner names of W_1 with the corresponding outer names of W_2 and forgetting

about their identities. The tensor product $\otimes$ of link graphs is defined in the obvious way only if their inner/outer names are disjoint.

The combination of ordinals m with names X gives the bigraphical *interfaces*, that are pairs $\langle m, X\rangle$. Combining the notion of place graph and link graphs on the same nodes we obtain the notion of bigraphs, i.e., arrows $G : \langle m, X\rangle \to \langle n, Y\rangle$. Given two bigraphs G_1 and G_2, intuitively the composition $G_1 \circ G_2$ *first* places every region of G_2 in the proper hole of G_1 (place composition) and *then* joins equal inner names of G_1 and outer names of G_2 (link composition). The operation is partially defined, since it requires the inner names and the number of holes of G_1 to match the number of regions and the outer names of G_2, respectively. Shared names create the new links between the two structures. On the other hand, the tensor product $G_1 \otimes G_2$, consists of placing close the two bigraphs, only in the case that the tensor product between their link graphs is defined.

I.4 Contribution of this Thesis

The Thesis is organised in two parts, they each develop one of the two different approaches. Part I considers Basic Logic as the fundament of a resource semantics, which is modularly extended to well known logics. Then places are introduced to the logic, and in the model as well, in order to describe distributed systems. Part II introduces bigraphs as a general model for distributed systems. Bigraphs form the basis for a new contextual logic: *BiLog*. This logic is then instantiated to well known spatial logics: Spatial Tree Logic [36], Spatial Graph Logic [39], Context Tree Logic[37], and a dynamic spatial logic for CCS [35].

What follows is a detailed description of the structure of the Thesis and the results we obtained.

I.4.1 "From Logic to Models. . .

Logic from the Basis. Chapter 1 introduces the 'Logic' via Basic Logic and its principle of reflection. Connectives and logical constants are defined by a distinctive definitional equation. Definitional equations are the deep fundament for the *relational semantics* provided for the basic calculus: the equations are projected onto the model and then solved to obtain the right evaluation for all the logical entities.

The models for Basic Logic are close to those for the Logic of Bunched Implications. They are just monoids $(M, \cdot, 1)$ equipped with a binary relation R, hence dubbed *relational monoids*. The idea we follow to define the semantics is thinking of M as the set of resources in a system. We admit a representative or *null* resource (the neutral element "1") and a way of *combining* resources (the monoidal operation "$\cdot$"). In §1.4 we relate R to a production cycle, that well reflects the idea of provability in case of sequents. Nevertheless R can be easily seen as an *accessibility* relation, by saying xRy if *the resource x can*

access the resource y in the system. Such a relation induces two operators on resources:

$$x^{\rightarrow} \stackrel{\text{def}}{=} \{y \in M \,:\, xRy\} \quad \text{the resources that } x \text{ have access;}$$
$$y^{\leftarrow} \stackrel{\text{def}}{=} \{x \in M \,:\, xRy\} \quad \text{the resources that access to } y.$$

The operators are extended to subsets of resources and are used to define the evaluation of formulae. We prove a theorem of soundness and a theorem of *refined* completeness that enables a semantical proof of cut-elimination as corollary. The relational semantics is then extended in a modular way to Intuitionistic and Classical Linear Logic, Intuitionistic Logic, and Classical Logic. All the extensions allow for a refined completeness theorem, leading to a semantical cut elimination theorem. As all the semantics is carefully handled in constructive settings, proofs do not have to be redone, but just modularly extended according to the logic under consideration.

The sequent calculus **LBI**, provided for the Logic of Bunched Implications in [70, 71, 122], is introduced according to the principle of reflection as well. In particular the connectives $*$ and $\wedge$ directly reflect the two ways of combining formulae with bunches. Thanks to the definitional equations provided for **LBI**, the relational semantics is extended to the Bunched Implications Logic. In that case, models are sets with a binary relation and two monoidal operations. Such models are the combination of the monoids that gives a semantic to **ILL** and those that give a semantics to **IL**. The extended semantics gives a refined completeness theorem, thus providing a constructive semantical proof of cut elimination for **LBI**. Intuitively the two properties we add to relational monoids correspond to ask for two well defined implications: the one is (intuitionistic) linear and the other intuitionistic. Hence we obtain a *semantical diamond*: by starting from **B**, we obtain **ILL** by requiring a multiplicative implication (corresponding to $-\!\!*$), **IL** by requiring an additive implication (corresponding to $\rightarrow$), and finally **BI** by requiring both the implications, and two monoidal operations as well.

By relaxing the requirement of a refined completeness theorem, the models for **LBI** are simplified to partially ordered monoids $(M, \cdot, \leq)$, where the order $\leq$ is partial and the monoidal operation $\cdot$ is total. In fact, the extension of the monoidal semantics for **IL** can be simplified to partially ordered sets of resources $(M, \leq)$. The semantics for **LBI** is then obtained by combining the relational semantics for **ILL** and the semantics for **IL** on partially ordered sets. We prove a soundness and completeness theorem for **LBI** on partially ordered monoids, whose proof is entirely constructive. Again, this semantics shows how the logic of Bunched Implications should be intended as an extension of Intuitionistic Logic and Intuitionistic Linear Logic: starting from **B**, we obtain **ILL** by requiring an operator on subsets that is adjoint to the product between subsets, and we obtain **IL** by requiring an operator that is adjoint to the intersection between subsets, then we obtain **LBI** by requiring both the adjoint operators. Hence Bunched Implication Logic can be modularly obtained, at least syntactically, either from Intuitionistic Logic or from Intuitionistic Linear Logic.

Adding Places to Logic. The Intuitionistic Modal Logic proposed in [92] is suitable to reason about distribution of resources. This has been already noticed in [92], but that paper does not provide a model to match the usage of the logic as a tool to reason about distributed resources. Chapter 2 fills the gap by presenting a Kripke-style semantics for such a logic.

We extend Kripke semantics of the intuitionistic logic [94], by enriching possible worlds with fixed sets of places. In each possible world, different places satisfy different formulae. For the intuitionistic connectives, the satisfaction of formulae at a place in a possible world follows the standard definition [94]. The enrichment of the model with places reveals the true meaning of the modalities in the logic. The modality $@p$ expresses a property in a named place, $\Box$ corresponds to a weak form of universal quantification and expresses a common property, and $\Diamond$ corresponds to a weak form of existential quantification and expresses a property valid somewhere in the system. In the model, we interpret atomic formulae as the resources of a distributed system, and placement of atoms in a possible world corresponds to the distribution of resources.The semantics is proved to be sound and complete for the logic.

By means of a counter example, adapted from [114], we prove that the Kripke semantics does not enjoy the finite model property. Then we refine the semantics by introducing *bi-relational models* [132], the semantics given on such models is sound and complete for the logic. The reason for introducing bi-relational models is that they satisfy the finite model property, and so they allow us to prove the decidability of the logic. As for Kripke models, birelational models have a partially ordered set. In addition, birelational models also possess an equivalence relation amongst elements. Unlike the Kripke semantics, we do not enrich each world with a set of places. Instead, we define a partial function, *the evaluation function*, which attaches a name to a world in its domain. The partiality of the function is crucial to the proof of decidability.

The partial evaluation function must satisfy two important properties. One, *coherence*, states that if the function associates a name to a world then it also associates the same name to all larger states. The other, *uniqueness*, states that two different worlds accessible from one another do not evaluate to the same name. Coherence is essential for ensuring monotonicity of the logical connective $@p$, while uniqueness is essential for the ensuring soundness of introduction of conjunction and implication.

The canonical model used to prove completeness is carefully defined in order to deduce the finite model property for the birelational semantics: if a judgement is not provable in the logic, then we can construct a finite birelational model which invalidates the judgement. The proof is adapted from the case of Intuitionistic Modal Logic [132]. Then we conclude the decidability of the logic. Hence the modal logic can be used to solve *queries* in a distributed system, e.g., a P2P distributed database to address query such as "Is there such information?," "Where are these data stored?"

I.4.2 . . . and Back”

BiLog: a Contextual Spatial Logic Founded on Bigraphs. Chapter 3 exploits the bi-structural nature of the bigraphical model to introduce a ‘*contextual spatial logic*’ for bigraphs built on two orthogonal sub-logics:

- a *Place Graph Logic* (for tree contexts), to express properties of resource locations;
- a *Link Graph Logic* (for name linkings), to express connections between resources (or, more precisely, resource names).

For this reason, we name the formalism *BiLog*.

We consider the axiomatisation given in [101], that introduces bigraphical terms. Every bigraph is formalised as the composition of fixed constructor terms by using the bigraphical operations $\circ$ and $\otimes$. BiLog internalises the bigraphical term constructors in the style of the Ambient Logic [42]. Constructors are represented in the logic as constant formulae, while tensor product and composition are expressed by connectives, thus providing two binary spatial operators. The logic is parameterised with a *transparency* predicate, that establishes when a term can be directly observed in the logic: some terms are opaque and do not allow inspection of their contents. In particular, when all terms are observable, logical equivalence corresponds to congruence. Otherwise, it can be less discriminating.

The logic features a logical constant for each *transparent* construct. The satisfaction of logical constants is simply defined as the congruence to the corresponding constructor. The *horizontal decomposition* formula $A \otimes B$ is satisfied by a term that can be decomposed as the tensor product of terms satisfying A and B respectively. The *vertical decomposition* formula $A \circ B$ is satisfied by terms that can be seen as the composition of terms satisfying A and B. Moreover we define the *left* and *right adjuncts* for composition and tensor to express extensional properties.

The main point is that a resource has a spatial structure as well as a link structure associated to it. Suppose for instance to be describing a tree-shaped distribution of resources in locations. We may use atomic formulae like $\mathsf{PC}(A)$ and $\mathsf{PC}_x(A)$ to describe a resource in an unnamed location, respectively location x, of ‘type’ PC (e.g. a computer) whose contents satisfy A. We can then write $\mathsf{PC}(\top) \otimes \mathsf{PC}(\top)$ to characterise models with two unnamed PC resources whose contents satisfy the tautological formula (i.e., with anything inside). By named locations, as e.g. in $\mathsf{PC}_a(\top) \otimes \mathsf{PC}_b(\top)$, we are able to express name separation, i.e., that names a and b are different. The logic is also sufficiently expressive to force name-sharing between resources with formulae like:

$$\mathsf{PC}_a(\mathsf{In}_c(\mathbf{1}) \otimes \top) \overset{c}{\otimes} \mathsf{PC}_b(\mathsf{Out}_c(\mathbf{1}) \otimes \top).$$

This describes two PC with different names, a and b, sharing a link on a distinct name c, which models, e.g., a communication channel. Name c is used as input for the first PC and as an output for the second PC.

A bigraphical structure is, in general, a context with several holes and open links that can be filled by composition. This means that the logic can describe contexts for resources at no addition cost. We can then express formulae like $\mathsf{PC}_a(\top \otimes \mathsf{HD}(id_1 \wedge A))$ that describes a modular PC, where id_1 represents a 'pluggable' hole (e.g. some disk space in PC's hard disk). Contextual resources have many important applications. In particular, the contextual nature of bigraphs is useful to specify reaction rules to deal with dynamics, but it can also be used as a general mechanism to describe contexts of bigraphical (bigraph-shaped) data structures (cf. [54] for an example with XML).

The main technical results we present are the encoding of several static spatial logics fragments of BiLog: Spatial Tree Logic [36], Spatial Graph Logic [39], and Context Tree Logic [37]. In this sense, the contribution of Chapter 3 is to propose BiLog as a unifying language for the description of global resources.

Another positive result in this direction has been obtained for semistructured data, by focusing on XML in particular. XML data are essentially tree-shaped resources, and have been modelled with unordered labelled tree (cf. [38]). We enriched over such model of tree-shaped data by adding links on resource names, to obtain a more general model for semistructured data and XML. In addition, bigraphs naturally model XML contexts: we thus obtained, with no additional effort, a logic to describe XML contexts which can be interpreted as web services or XML transformations. In particular, §3.6 first shows how XML data (and, more generally, contexts or positive web services) can be interpreted as a bigraph. Equipped with such 'bigraphical' representation of XML data and contexts, we then give a gentle introduction to different fragments of BiLog and show how they can be applied to describe and reason about XML. The contribution of the section is therefore to identify (fragments of) BiLog as a suitable formalism for semistructured data, and illustrate its expressiveness by means of selected examples.

In some cases, BiLog is also able to deal with the dynamic behaviour of the model. Essentially, this happens because the contextual nature of the logic can be used to characterise the structure of the processes qualified to evolve. Section 3.7 shows this fact on a fragment of CCS: by using the bigraphical encoding for CCS provided in [103], we encode into BiLog a simple dynamical Spatial Logic [35] suitable to analyse CCS evolving processes. Essentially, this happens thanks to the contextual nature of the logic that can be used to characterise structural parametric reaction rules.

I

“From Logic to Models. . .

1

Logic from the Basis

In this chapter we introduce *'Logic'* from a foundational point of view. We start from the meta-level with Basic Logic, along with its foundational principles and its definitional equations, which are essential to provide a notion of model for the logic itself: the relational monoids. We prove soundness and refined completeness results for this class of models. In particular the completeness result allows a semantical proof of cut-elimination.

Basic Logic is then extended in two directions: one direction is the addition of structural rules, the other is the move to intuitionistic logic, thus obtaining Intuitionistic Linear Logic and Intuitionistic Logic. The notion of model, in turn, can be extended to these logics, and in each case the soundness and refined completeness results are retained. These newly found models are related to more traditional models of the logics so discovered: Kripke models for Intuitionistic Logic and Phase Spaces for Linear Logic. Finally, combining the two extensions leads us to the Logic of Bunched Implications.

1.1 Introduction

Basic Logic has been introduced in [129] with the aim of finding a structure in the space of logics. Classical, Intuitionistic, Quantum and Non-modal Linear Logics are all obtained as extensions in a uniform way. The logical constants and connectives are introduced by following three fundamental principles: *reflection*, *symmetry* and *visibility*. The *principle of reflection* says, in particular, that each connective reflects at object-level a link between assertions at the meta-level. This provides each connective and logical constant with a clear meaning, that is, with a semantics. Every logical entity is associated to an equation, the *definitional equation*, binding it with the corresponding meta-linguistic link. All the syntactic inference rules, expressed by Gentzen's sequents [72, 73], are obtained by solving such equations.

Nevertheless, the calculus is still lacking a mathematical interpretation (commonly called 'semantics'), and a semantical comparison with the models proposed for other logics, such as all those obtained as its extensions. Here we present a mathematical interpretation for the core calculus of Basic Logic, with additive and multiplicative connectives and constants. The models we introduce are just monoids equipped with a binary relation, that may be intended as sets of resources which can be composed, by the monoidal

operation, or compared, by the relation. The relation, in particular, induces two closure operators on subsets which are obtained by combining polarities, as in [19]. The idea, inspired by [77], is to interpret formulae as subsets which are closed in this sense. We think in terms of a production cycle, where the elements of the monoid are resources and the relation determines whether a resource can produce another one.

The evaluation of each connective is deeply founded on the definitional equations characterising the connective itself. The proof of validity is then immediate. Contrary to what happens in other logics, here the evaluation of a sequent $\Gamma \vdash \Delta$ cannot be reduced in general to the evaluation of a sequent of the form $\varphi \vdash \psi$ or to the evaluation of a single formula. In fact, here the 'comma' in the lists Γ and Δ can be replaced by a connective only when Γ (or Δ) consists of only two formulae. This is due to the property of visibility, that requires all active formulae in an inference rule to be isolated, or visible, without any passive contexts on their side in the sequent.

The completeness theorem will be proved in a 'refined' way that enables a semantic cut-elimination theorem. The proof relies on a particular model, built up from syntax, where resources are lists of formulae and the relation represents provability without using cuts; this reflects the idea of 'production' in a sequent calculus.

Then the semantics is extended directly by considering the properties required by the calculus on the syntactical side. These properties will be carefully cast on the semantical level in a way that allows to find the 'exact' conditions which enable the extension of the semantics to many logics obtained from Basic Logic, notably Paraconsistent Quantum Logic, Intuitionistic Linear Logic, and Intuitionistic Logic. In particular, the models for Intuitionistic Logic can be simplified to preordered sets. The result will provide a deep correlation between Basic Logic and its extensions. Moreover this result helps in understanding of context control in a sequent calculus, and in handling structural rules. Our path starts from the basic calculus and leads to the intuitionistic one, going through sub-structural and linear calculi, by looking at the needed requirements at every step, and exactly projecting them on the model.

The principle of reflection can be applied also to Bunched Implications Logic, and its sequent calculus [70, 71, 122] in particular. The additive and multiplicative conjunctions directly reflect the two ways of combining formulae with bunches. Thanks to the definitional equations we will provide for Bunched Implications, the relational semantics is extended to the Bunched Implications Logic. In that case, models are sets with a binary relation and two monoidal operations. Such models are the combination of the monoids that give a semantics to Intuitionistic Linear Logic and those that give a semantics to Intuitionistic Logic. The extended semantics gives a refined completeness theorem, thus providing a constructive semantical proof of cut elimination for Bunched Implications. By relaxing the requirement of a refined completeness theorem, Bunched Implications models are then simplified to partially ordered monoids, that are obtained from the relational semantics for Intuitionistic Linear Logic and the simplified semantics for Intuitionistic Logic. This semantics, sound and complete, shows how the logic of Bunched Implications can be modularly obtained, at least syntactically, either from Intuitionistic Logic or from Intuitionistic Linear Logic.

Most of the semantical extensions will be equivalent to well-known semantics. In particular, the relational monoids in which the relation is strongly symmetric (cf.§1.10) turn out to be exactly the phase spaces introduced by J.-Y. Girard as semantics of Linear Logic in [74]. This should highlight in which sense Linear Logic (without exponentials!) is a proper extension of Basic Logic.

The structural rule of exchange was introduced in Basic Logic's sequent calculus simply for reasons of convenience, to avoid duplications of implications. Since here we omit implications, it is very natural to consider the sequent calculus obtained dropping also the rule of exchange. In fact exchange is valid in a relational monoid whenever the monoid operation is commutative. Thus the relational semantics introduced here applies to *non commutative* Basic Logic, and its substructural extensions.

1.2 The Basic Calculus

This section introduces the basic sequent calculus **B**: the *Core Basic Logic*. It is the kernel of Basic Logic, the sequent calculus introduced in [129]. The calculus **B** is built on the additive and multiplicative structures of Basic Logic simply by deleting the exchange rules, that are the only structural rules of Basic Logic. In particular, **B** is non-commutative.

Basic Logic has been introduced as a logic which obeys three general principles: *reflection, symmetry* and *visibility*. Reflection is the most important, and the main novelty introduced by Basic Logic. A detailed discussion of such a principle appears in [128]. In the basic calculus **B**, the principle of reflection guides the choice of logical constants, connectives and the inference rules. The general idea is to start from the meta-level, and to understand which significance the logical entities must reflect on the language. At the meta-level some *desideratum* is pointed out, then it is expressed in the language and it drives the definition of the inference rules.

From now on, assume that $\varphi, \psi, \ldots$ denote *propositions*. Propositions are a formalisation of properties that can be asserted. At the meta-level, a proposition φ must be distinct from the assertion on it. Usually an assertion on φ is denoted as *"A is true"*, the basic calculus **B** adopts a more neutral notation like *"A is"*, that, depending on the settings, can express *A is true, A is available, A is utilised* and so on. Also, the meta-level considers more complex statements built up from assertions by using meta-linguistic links. The insight of Basic Logic is that, in order to define all the logical entities used in any sequent calculus (lists Γ, sequents $\Gamma \vdash \Delta$, rules and derivations), it is sufficient to consider only two meta-linguistic links: *and* and *yields*.

A conjunction of assertions φ_1*is and* ... *and* φ_n*is* is abbreviated by $\varphi_1, \ldots, \varphi_n$, where commas take the place both of *and* and of *is*. Following Gentzen's notations [72, 73], Γ, Δ denote any conjunction of atomic assertions $\varphi_1, \ldots, \varphi_n$, possibly empty. In general, small Greek letters will denote formulae and capital Greek letters will denote (possibly empty) lists of formulae. The meaning of a sequent $\Gamma \vdash \Delta$ is that Δ is a logical consequence of Γ, that is 'Γ *yields* Δ'. Both the *antecedent* Γ and the *consequent* Δ are called *contexts.* The

meaning of a rule of inference

$$\frac{\Gamma \vdash \Delta}{\Gamma' \vdash \Delta'}$$

is that the derivation can move from the assertion $\Gamma \vdash \Delta$ to the assertion $\Gamma' \vdash \Delta'$, or in meta-language words: (Γ *yields* Δ) *yields* (Γ' *yields* Δ'). Inference rules can have more than one premiss, above the horizontal bar and separated by a blank space. Such a space is a notation for *and*, and so

$$\frac{\Gamma \vdash \Delta \quad \Gamma' \vdash \Delta'}{\Gamma'' \vdash \Delta''}$$

is a short notation for ((Γ *yields* Δ) *and* (Γ' *yields* Δ')) *yields* (Γ'' *yields* Δ'').

Note that two meta-linguistic links are sufficient, as their meaning can change. The link *and* is a link between atomic assertions and composed assertions; the link *yields* is the $\vdash$ of the sequent and the horizontal bar of an inference rule.

All the connectives of the language are introduced to reflect a meta-linguistic link, and the definition of their rules reflect the meta-meaning of the corresponding meta-link. The reflection is given by an equation, called *definitional equation*, that expresses the main expected property of the introduced connective. For instance, if the connective $\otimes$ is intended to reflect the link between atomic assertions on the left hand side of a sequent, then the main property to require is that for all Δ, φ, ψ:

$$\psi \otimes \varphi \vdash \Delta \ \textit{ if and only if } \ \psi, \varphi \vdash \Delta,$$

where the link *if and only if* is a shorthand for *yields* in both directions. The two directions are called *implicit* $\otimes$*-reflection* and $\otimes$*-formation* respectively, and they give a first approximation of the rules for $\otimes$, that are

$$\frac{\psi \otimes \varphi \vdash \Delta}{\psi, \varphi \vdash \Delta}\ \textit{implicit}\ \otimes\textit{-reflection} \qquad \frac{\psi, \varphi \vdash \Delta}{\psi \otimes \varphi \vdash \Delta}\ \otimes\textit{-formation}$$

Formation projects the link *and* between assertions at the meta-level, reflection hints how to recover the meta-level situation. While $\otimes$-formation is a good formal rule, that can appear in a sequent calculus, implicit $\otimes$-reflection is still the statement of a desideratum, which specifies the meaning of the connective $\otimes$ only in an implicit way. In fact, such a rule assumes the meaning of $\varphi \otimes \psi$ to be already known, as the compound formula appears in the premise of the rule. The calculus must define the meaning of the connective $\otimes$ without vicious circles, hence a satisfactory rule has to be equivalent to implicit $\otimes$-reflection, without requiring assumptions on $\varphi \otimes \psi$. Doing so means *to solve* the definitional equation for the connective $\otimes$. For this process, some basic native rules are assumed. First of all, the *axioms*, a common starting point in all logical calculi: every assertion yields itself. Hence for every atomic assertion φ the sequent $\varphi \vdash \varphi$ is an axiom of the calculus. Then a way of composing proof is admitted. It is some a of logical substitution of derivations, or transitivity for the meta-linguistic link *yields*. Two ways of composition are allowed, as usual they are dubbed *cuts*:

$$\frac{\Gamma \vdash \varphi \quad \Gamma_1, \varphi, \Gamma_2 \vdash \Delta}{\Gamma_1, \Gamma, \Gamma_2 \vdash \Delta}\ \textit{cutL} \qquad \frac{\Gamma \vdash \Delta_2, \varphi, \Delta_1 \quad \varphi \vdash \Delta}{\Gamma \vdash \Delta_2, \Delta, \Delta_1}\ \textit{cutR}$$

Note that in every rule, the substituted assertion φ must appear 'isolated' in at least one side of the sequent, this is to obey the principle of visibility, described in the following.

Axioms and *cut* rules are the only 'tools' to transform $\otimes$-implicit reflection into an admissible formal rule. The first step is to make trivial the premiss of implicit $\otimes$-reflection, by considering the axiom $\varphi \otimes \psi \vdash \varphi \otimes \psi$, thus obtaining the equivalent axiom

$$\varphi, \psi \vdash \varphi \otimes \psi \quad \textit{axiom of } \otimes\textit{-reflection}.$$

The implicit $\otimes$-reflection is recovered by one application of the composition

$$\frac{\varphi, \psi \vdash \varphi \otimes \psi \quad \varphi \otimes \psi \vdash \Delta}{\varphi, \psi \vdash \Delta}$$

The final solution to the definitional equation is reached by replacing φ and ψ with arbitrary contexts Γ_1 and Γ_2, that is assuming that $\Gamma_1 \vdash \varphi$ and $\Gamma_2 \vdash \psi$ and applying two compositions

$$\frac{\Gamma_2 \vdash \psi \qquad \dfrac{\Gamma_1 \vdash \varphi \quad \varphi, \psi \vdash \varphi \otimes \psi}{\Gamma_1, \varphi \vdash \varphi \otimes \psi}}{\Gamma_1, \Gamma_2 \vdash \varphi \otimes \psi}$$

thus obtaining the rule

$$\frac{\Gamma_1 \vdash \varphi \quad \Gamma_2 \vdash \psi}{\Gamma_1, \Gamma_2 \vdash \varphi \otimes \psi} \ \textit{explicit} \otimes\textit{-reflection} \ .$$

To recover the axiom of $\otimes$-reflection it is sufficient to trivialise the premiss with the two axioms involving φ and ψ. The explicit $\otimes$-reflection is what was needed, the connective is introduced in the conclusion and there is not any vicious circle. The definitional equation for $\otimes$ is thus solved, and the $\otimes$ reflects the meta-link *and* on the left hand side of the sequent.

All the connectives and constants of the calculus **B** are introduced by solving a definitional equation. The pattern to follow is always the same: one direction of the equation gives the acceptable formation rule, the other one gives the implicit reflection, that is further refined to the axiom of reflection and then to the actual rule of the calculus: the explicit reflection.

The language $\mathcal{L}$ of the calculus **B** consists of propositional constants $\top$, $\bot$, 1 and 0, propositional variables $p, q, \ldots$, additive connectives $\oplus$ and $\&$, and multiplicative connectives $\otimes$ and $\parr$. The definitional equations for the logical entities are fully reported in Table 1.1. In particular, the connective $\parr$ reflects the meta-link *and* on the right hand side of the sequent, the connectives $\&$ and $\oplus$ reflect a meta-link *and* between sequents, the propositional constants 1 and $\bot$ reflect the empty assertion, on the left and on the right respectively, of a sequent; the propositional constants $\top$ and 0 reflect trivial assertions for a link *and* between sequents.

Table 1.1 hints also at the other two driving principles for Basic Logic: symmetry and visibility. Symmetry is a new conceptual tool, which abandons the traditional scheme that

Table 1.1 Definitional Equations

$(\otimes)$	$\psi \otimes \varphi \vdash \Delta$ if and only if $\psi, \varphi \vdash \Delta$
$(\parr)$	$\Gamma \vdash \varphi \parr \psi$ if and only if $\Gamma \vdash \varphi, \psi$
$(\oplus)$	$\psi \oplus \varphi \vdash \Delta$ if and only if $\psi \vdash \Delta$ and $\varphi \vdash \Delta$
$(\&)$	$\Gamma \vdash \varphi \,\&\, \psi$ if and only if $\Gamma \vdash \varphi$ and $\Gamma \vdash \psi$
(1)	$1 \vdash \Delta$ if and only if $\vdash \Delta$
$(\bot)$	$\Gamma \vdash \bot$ if and only if $\Gamma \vdash$
(0)	$\varphi \vdash \Delta$ and $0 \vdash \Delta$ if and only if $\varphi \vdash \Delta$
$(\top)$	$\Gamma \vdash \psi$ and $\Gamma \vdash \top$ if and only if $\Gamma \vdash \psi$

says that the rule introducing a connective is always the rule operating on the right and that the rule on the left is always the elimination rule. In Basic Logic , the logical constants and connectives are equally divided into *left* and *right* connectives. A left connective has the formation rules operating on the left, and the reflection rule operating on the right, viceversa for a right connective, with a formation rule on the right and a reflection rule on the left. Every connective has its own corresponding symmetric connective. As it can be seen from the definitional equations, the connectives $\otimes$ and $\oplus$ are symmetric to $\parr$ and $\&$ respectively, the former are right connectives and the latter are left ones. The choice of the names for the formulae and contexts in the table is ad hoc, to emphasise the symmetry among the logical entities.

The basic calculus has a strong control not only on the structural rules (exchange, weakening and contractions), but also on the contexts of the sequents. The principle of visibility, in fact, forces the definitional equations, and hence the derived rules of the calculus, to operate on formulae that are the only ones appearing either in the antecedent or the consequent of a sequent. For instance, the definitional equation for $\otimes$ does not have *passive* context on the left, namely on the left hand side of the sequent there are no other formulae but those involved on the connective. Visibility is even more clear by looking at the inference rules of **B** obtained by solving the definitional equations and that are listed in Fig. 1.1, along with axioms and composition rules. For uniformity, the rules are denoted with *L*, introduction on the left-hand side of the sequent, and with *R*, introduction on the right-hand side of the sequent, instead of reflection and formation, as previously introduced. By visibility, the left rules do not have passive context on the left, and analogously the right rules do not have passive context on the right. Such a constraint allows for an intuitive cut elimination theorem [129] that can be extended to every calculus obtained from Basic Logic.

Once the definitional equations have been solved, the formalism for **B** is a standard sequent calculus. Note in particular that, conversely, the definitional equations become formally derivable in **B**, and are properties actually verified by the calculus. Definitional equations are a crucial point in this work, since they provide the right intuitions for the definition of evaluation of formulae, as shown in § 1.4. In fact, the evaluation of formulae

Figure 1.1 Basic Sequent Calculus **B**

Axioms

$$\varphi \vdash \varphi$$

Operational Rules

Multiplicatives

$$\frac{\psi, \varphi \vdash \Delta}{\psi \otimes \varphi \vdash \Delta} \otimes L \qquad \frac{\Gamma \vdash \varphi, \psi}{\Gamma \vdash \varphi \parr \psi} \parr R$$

$$\frac{\psi \vdash \Delta_1 \quad \varphi \vdash \Delta_2}{\psi \parr \varphi \vdash \Delta_1, \Delta_2} \parr L \qquad \frac{\Gamma_2 \vdash \varphi \quad \Gamma_1 \vdash \psi}{\Gamma_2, \Gamma_1 \vdash \varphi \otimes \psi} \otimes R$$

$$\frac{\vdash \Delta}{1 \vdash \Delta} \; 1L \qquad \frac{\Gamma \vdash}{\Gamma \vdash \bot} \; \bot R$$

$$\bot \vdash \quad \bot L \qquad \vdash 1 \quad 1R$$

Additives

$$\frac{\psi \vdash \Delta \quad \varphi \vdash \Delta}{\psi \oplus \varphi \vdash \Delta} \oplus L \qquad \frac{\Gamma \vdash \varphi \quad \Gamma \vdash \psi}{\Gamma \vdash \varphi \,\&\, \psi} \,\& R$$

$$\frac{\psi \vdash \Delta}{\psi \,\&\, \varphi \vdash \Delta} \,\& L \quad \frac{\varphi \vdash \Delta}{\psi \,\&\, \varphi \vdash \Delta} \,\& L \qquad \frac{\Gamma \vdash \varphi}{\Gamma \vdash \varphi \oplus \psi} \oplus R \quad \frac{\Gamma \vdash \psi}{\Gamma \vdash \varphi \oplus \psi} \oplus R$$

$$0 \vdash \Delta \;\; 0L \qquad \Gamma \vdash \top \;\; \top R$$

Cut Rules

$$\frac{\Gamma \vdash \varphi \quad \Gamma_1, \varphi, \Gamma_2 \vdash \Delta}{\Gamma_1, \Gamma, \Gamma_2 \vdash \Delta} \; cutL \qquad \frac{\Gamma \vdash \Delta_2, \varphi, \Delta_1 \quad \varphi \vdash \Delta}{\Gamma \vdash \Delta_2, \Delta, \Delta_1} \; cutR$$

is deeply founded on the definitional equations. The equations themselves trace the right definition for the evaluation. Also the soundness lemma is not standard, since it shows that the equational definitions are semantically valid, instead of checking the soundness of the rules. The usage of the definitional equations is the main novelty of the semantics: just as definitional equations are the driving idea for the sequent calculus, they lead the choices in the mathematical semantics.

The semantics provides a *refined* completeness theorem (cf. § 1.5), that allows for a cut elimination theorem in the calculus. One may ask which is the function of the cuts, as the usage of cut rules seems to be peculiar to solve the definitional equations and to obtain the calculus **B**. It turns out that cuts are just 'accessorial' to define the calculus, as they are eliminable. In fact, an important point, outlined here for the first time, is that the definitional equations are respected also by the sequent calculus **B** deprived of cut rules.

The calculus **B** satisfies, for instance, the property that for every Δ, φ and ψ:

$$\begin{array}{c} \psi \otimes \varphi \vdash \Delta \text{ is derivable} \\ \text{without } \textit{cut} \text{ rules} \end{array} \quad \text{if and only if} \quad \begin{array}{c} \psi, \varphi \vdash \Delta \text{ is derivable} \\ \text{without } \textit{cut} \text{ rules} \end{array} \tag{1.1}$$

And so on for every other connective and constant, by following the equations in Table 1.1. This fact is formalised below.

Proposition 1 (Cut-free Equations for B). *The calculus obtained from* **B** *by removing the cut rules satisfies the definitional equations for every connective and logical constant.*

Proof. The proof follows a common pattern for every logical entity: the backward direction of the implication is guaranteed by the formation rule, the forward one is proved by induction on the length of the cut-free derivation. Consider the case of the connective $\otimes$ as a guideline. The property to prove is the cut free definitional equation outlined in (1.1). As anticipated, for the backward direction simply apply the rule $\otimes L$, and for the forward direction proceed by induction on the length of the cut free derivation of $\varphi \otimes \psi \vdash \Delta$. The base of induction is any rule without premisses, and it can only be either an axiom or $\top R$. On the one hand, if it is an axiom, then Δ is $\psi \otimes \varphi$, and $\psi, \varphi \vdash \psi \otimes \varphi$ is derived without cuts as

$$\frac{\psi \vdash \psi \quad \varphi \vdash \varphi}{\psi, \varphi \vdash \psi \otimes \varphi} \otimes R$$

On the other hand, if the applied rule is $\top R$, then Δ is the constant $\top$ and so $\psi, \varphi \vdash \top$ by $\top R$ as well. In the induction step consider the last applied rule in the derivation: (a) if it is an introduction on the left, then it can only be $\otimes L$, hence its premiss gives the claim; (b) if it is an introduction on the right, then use induction hypothesis on the premisses of the rule and obtain the claim by applying the same rule. □

As a matter of fact, the previous lemma still holds for the full Basic Logic calculus, with the two implications and exchange rules, and it can be proved by following the above argumentations. Furthermore, it can be verified for every extension of Basic Logic presented in this chapter and in [129].

1.3 Relational Monoids

The basic structures giving semantics to **B** are monoids $(M, \cdot, 1)$ equipped with a binary relation R, they are called *relational monoids* and denoted by $\mathcal{M} = (M, \cdot, 1, R)$. The monoidal operation, *associative* by definition, will reflect formulae composition. The relation, completely orthogonal to the monoidal operation, will introduce Birkhoff's polarities [19], which will be used to define the class of subsets on which to evaluate formulae. This section repeats the basic properties of this model.

Lower-case letter $x, y, z \ldots$ will range over elements of M; capital letters $A, B, C \ldots$ will range over subsets of M. As the whole framework is founded on constructive settings, this chapter embraces the definitions and the notations for subsets introduced and justified

in [130]. Accordingly, '$A \subseteq M$' means that A is a propositional function over M, and '$x \epsilon A$' that x is an element of the subset A, as it satisfies proposition A.

Through the relation, every element z determines two subsets: the subset $z^{\leftarrow}$ of the elements in *left* relation with z and the subset $z^{\rightarrow}$ of the elements in *right* relation with z:[1]

$$z^{\leftarrow} \stackrel{\text{def}}{=} \{x \in M \,:\, x\,R\,z\} \qquad \text{and} \qquad z^{\rightarrow} \stackrel{\text{def}}{=} \{y \in M \,:\, z\,R\,y\}. \tag{1.2}$$

Note that $x \epsilon y^{\leftarrow}$ if and only if $y \epsilon x^{\rightarrow}$, hence the operators are *adjoint on the right* [68]. Our first aim is to extend the operators to all subsets and respect this property, as it will be central to define the suitable subsets to evaluate formulae. So we require

$$A \subseteq B^{\leftarrow} \text{ if and only if } B \subseteq A^{\rightarrow}. \tag{1.3}$$

By considering singletons, such a property is specialised to *(i)* $x \epsilon B^{\leftarrow}$ if and only if $B \subseteq x^{\rightarrow}$, and *(ii)* $y \epsilon A^{\rightarrow}$ if and only if $A \subseteq y^{\leftarrow}$. This hints how to define the operators on subsets. In fact, by *(i)*, $x \epsilon B^{\leftarrow}$ means $y \epsilon x^{\rightarrow}$ for every $y \epsilon B$, that is $x \epsilon y^{\leftarrow}$ for every $y \epsilon B$. Thus the required definition must be

$$B^{\leftarrow} \stackrel{\text{def}}{=} \{x \in M \,:\, x\,R\,y \text{ for all } y \epsilon B\} = \bigcap_{y \epsilon B} y^{\leftarrow}. \tag{1.4}$$

Symmetrically, by *(ii)*, $y \epsilon A^{\rightarrow}$ means $x \epsilon y^{\leftarrow}$ for every $x \epsilon A$, that is $y \epsilon x^{\rightarrow}$ for every $x \epsilon A$. In this case, the definition is

$$A^{\rightarrow} \stackrel{\text{def}}{=} \{y \in M \,:\, x\,R\,y \text{ for all } x \epsilon A\} = \bigcap_{x \epsilon A} x^{\rightarrow}. \tag{1.5}$$

These definitions characterise Birkhoff's *polarities* [19]. Notations are not ambiguous for singletons, as $\{x\}^{\rightarrow}$ and $\{y\}^{\leftarrow}$ correspond to $x^{\rightarrow}$ and $y^{\leftarrow}$. Definitions (1.4) and (1.5) imply the property (1.3); hence they are the only way to extend the operators $(\,)^{\rightarrow}$ and $(\,)^{\leftarrow}$ to subsets and to respect (1.3).

Lemma 1. *Condition* (1.3) *is equivalent to the triplet of properties:*

$$A \subseteq A^{\rightarrow\leftarrow} \quad \textit{and} \quad A \subseteq A^{\leftarrow\rightarrow}. \tag{1.6}$$

$$A \subseteq B \quad \textit{implies} \quad B^{\rightarrow} \subseteq A^{\rightarrow}; \tag{1.7}$$

$$A \subseteq B \quad \textit{implies} \quad B^{\leftarrow} \subseteq A^{\leftarrow}; \tag{1.8}$$

Proof. For (1.6), apply (1.3) to $A^{\rightarrow} \subseteq A^{\rightarrow}$ and to $A^{\leftarrow} \subseteq A^{\leftarrow}$. For (1.7): $A \subseteq B$ implies $A \subseteq B^{\rightarrow\leftarrow}$ by (1.6), and $B^{\rightarrow} \subseteq A^{\rightarrow}$ by (1.3). Symmetrically for (1.8). For the forward direction of (1.3) apply (1.7) and (1.6), and for the backward one apply (1.8) and (1.6). □

Conditions (1.6)–(1.8) say that the correspondences $A \mapsto A^{\rightarrow}$ and $B \mapsto B^{\leftarrow}$ define a *Galois connection* [19] between the complete lattice $(\mathcal{P}(M), \subseteq)$ and itself, where $\mathcal{P}(M)$ and represents the class of subsets of M and $\subseteq$ is the inclusion among subsets.

[1] Here and in whole Thesis $\stackrel{\text{def}}{=}$ is the sign for definitional equality, when a definition is first given, the definiendum will always be at the left and the definiens at the right.

Corollary 1. *In a relational monoid,* $A^{\rightarrow\leftarrow\rightarrow} = A^{\rightarrow}$ *and* $A^{\leftarrow\rightarrow\leftarrow} = A^{\leftarrow}$ *for every* $A \subseteq M$.

Proof. By (1.6), $A^{\rightarrow} \subseteq A^{\rightarrow\leftarrow\rightarrow}$ and $A \subseteq A^{\rightarrow\leftarrow}$. Then $A^{\rightarrow\leftarrow\rightarrow} \subseteq A^{\rightarrow}$ by (1.7). Similarly for $A^{\leftarrow\rightarrow\leftarrow} = A^{\leftarrow}$. □

This property is useful to prove that the composition of polarities gives two closure operators. Recall that $C : \mathcal{P}(M) \longrightarrow \mathcal{P}(M)$ is called a closure operator if *(i)* $A \subseteq C\,A$, *(ii)* $C\,C\,A = C\,A$, and *(iii)* $A \subseteq B$ implies $C\,A \subseteq C\,B$ for every $A, B \subseteq M$.

Corollary 2. *The operators* $(\,)^{\rightarrow\leftarrow}$ *and* $(\,)^{\leftarrow\rightarrow}$ *are closure operators.*

Proof. Given $A, B \subseteq M$, $A \subseteq A^{\rightarrow\leftarrow}$ by (1.6); $A^{\rightarrow\leftarrow\rightarrow\leftarrow} \subseteq A^{\rightarrow\leftarrow}$ by Corollary 1; $A \subseteq B$ implies $A^{\rightarrow\leftarrow} \subseteq B^{\rightarrow\leftarrow}$ by (1.7) and (1.8). Similarly for $(\,)^{\leftarrow\rightarrow}$. □

The next lemma shows that $(\,)^{\rightarrow\leftarrow}$ resembles a Dedekind-MacNeille completion [61].

Lemma 2. *In a relational monoid,* $A^{\rightarrow\leftarrow} = \bigcap_{A\subseteq z^{\leftarrow}} z^{\leftarrow}$ *for every* $A \subseteq M$.

Proof. By definition $A^{\rightarrow\leftarrow}$ is $\bigcap_{z\,\epsilon\,A^{\rightarrow}} z^{\leftarrow}$, that is $\bigcap_{A\subseteq z^{\leftarrow}} z^{\leftarrow}$ by (1.3). □

The closure operators identify two classes of subset.

Definition 1 (Saturated Subsets). *The subset* $A \subseteq M$ *is* left saturated *if* $A = A^{\rightarrow\leftarrow}$ *and* $B \subseteq M$ *is* right saturated *if* $B = B^{\leftarrow\rightarrow}$. *Moreover* $Sat^{\leftarrow}(M)$ *and* $Sat^{\rightarrow}(M)$ *are the collections of left saturated and right saturated subsets of M respectively.*

The justification for the adjectives ‘left’ and ‘right’ derives from Corollary 1: left and right saturated subsets are just those of the form $B^{\leftarrow}$ and $A^{\rightarrow}$ respectively.

The collections $Sat^{\leftarrow}(M)$ and $Sat^{\rightarrow}(M)$ are complete lattices, where meet (*glb*) is the intersection $\cap$ and join (*lub*) is the saturation of the union $\cup$. M is the maximum among both left and right saturated subsets. The saturations of the empty subset, $\emptyset^{\rightarrow\leftarrow}$ and $\emptyset^{\leftarrow\rightarrow}$, are the minimum among left and right saturated subsets, respectively. The next theorem shows a very important correspondence between left and right saturated subsets. Such a correspondence will be useful to evaluate the formulae of the language $\mathcal{L}$.

Theorem 1. *The correspondences* $A \mapsto A^{\rightarrow}$ *and* $B \mapsto B^{\leftarrow}$ *define a dual isomorphism between the complete lattices of left and right saturated subsets. In particular, if* A_1, A_2 *are left saturated subsets and* B_1, B_2 *are right saturated subsets, then:*

$$(A_1 \cap A_2)^{\rightarrow} = (A_1{}^{\rightarrow} \cup A_2{}^{\rightarrow})^{\leftarrow\rightarrow} \qquad (B_1 \cap B_2)^{\leftarrow} = (B_1{}^{\leftarrow} \cup B_2{}^{\leftarrow})^{\rightarrow\leftarrow} \tag{1.9}$$

$$(A_1 \cup A_2)^{\rightarrow} = A_1{}^{\rightarrow} \cap A_2{}^{\rightarrow} \qquad (B_1 \cup B_2)^{\leftarrow} = B_1{}^{\leftarrow} \cap B_2{}^{\leftarrow} \tag{1.10}$$

$$\emptyset^{\rightarrow} = M \qquad \emptyset^{\leftarrow} = M \tag{1.11}$$

$$M^{\rightarrow} = \emptyset^{\leftarrow\rightarrow} \qquad M^{\leftarrow} = \emptyset^{\rightarrow\leftarrow} \tag{1.12}$$

Proof. By Corollary 1, the correspondences $A \mapsto A^{\rightarrow}$ and $B \mapsto B^{\leftarrow}$ are inverse of each other; hence they are one-one and onto. Finally, by (1.7) and (1.8), they invert inclusion and so they interchange join with meet. □

The final lemma relates the operators $(\,)^{\rightarrow}$ and $(\,)^{\leftarrow}$ with the operation in the monoid. The algebraic product between subsets is denoted by $A \cdot B \stackrel{\text{def}}{=} \{x \cdot y : x \in A, y \in B\}$.

Lemma 3. *Given $A_1, A_2, B_1, B_2 \subseteq M$, if $A_1 \subseteq B_1$ and $A_2 \subseteq B_2$ then $(A_1{}^{\rightarrow} \cdot A_2{}^{\rightarrow})^{\leftarrow} \subseteq (B_1{}^{\rightarrow} \cdot B_2{}^{\rightarrow})^{\leftarrow}$ and $(A_1{}^{\leftarrow} \cdot A_2{}^{\leftarrow})^{\rightarrow} \subseteq (B_1{}^{\leftarrow} \cdot B_2{}^{\leftarrow})^{\rightarrow}$.*

Proof. First use (1.7) compose by $\cdot$ and use (1.8). The second point is analogous. □

1.3.1 Preorder Relations

Preorders are reflexive and transitive relations. They will specialise the semantics to intuitionistic logic (cf. §1.7.2). For the sake of uniformity, here we study the basic properties of the operators $(\,)^{\leftarrow}$ and $(\,)^{\rightarrow}$ generated by preorders. Since the results are not fundamental for the semantics of **B**, reading of this section can be postponed until §1.7.2 is reached. A preorder is commonly written as $\leq$, thus we use such a notation in this section, and we will be consistent with it in the whole chapter whenever dealing with preorders.

The next proposition says that the closure operator $(\,)^{\rightarrow\leftarrow}$ and $(\,)^{\leftarrow}$ collapse if and only if the underlying relation is a preorder.

Proposition 2. *In any relational monoid, the relation is reflexive if and only if $x^{\rightarrow\leftarrow} \subseteq x^{\leftarrow}$ for every element x, and it is transitive if and only if $x^{\leftarrow} \subseteq x^{\rightarrow\leftarrow}$ for every element x.*

Proof. Reflexivity means $x \in x^{\leftarrow}$, hence $x^{\leftarrow\rightarrow} \subseteq x^{\leftarrow}$ since $(\,)^{\rightarrow\leftarrow}$ is a closure operator. Transitivity is just $x^{\leftarrow} \subseteq \bigcap_{x \in z^{\leftarrow}} z^{\leftarrow}$, namely $x^{\leftarrow} \subseteq x^{\rightarrow\leftarrow}$ by Lemma 2. □

In case of preorders, we introduce a new operator $\downarrow$. The operator $\downarrow$ is 'dual' to $(\,)^{\leftarrow}$, as it considers union instead of intersection (cf. Lemma 2):

$$\downarrow A \stackrel{\text{def}}{=} \{z : z \leq x \text{ for any } x \in A\} = \bigcup_{x \in A} x^{\leftarrow}. \tag{1.13}$$

Proposition 3. *The operator $\downarrow$ is a closure operator distributive over subset union.*

Proof. Given $A, B \subseteq M$, $A \subseteq \downarrow A$ by reflexivity; $\downarrow\downarrow A \subseteq \downarrow A$ by transitivity; and $A \subseteq B$ implies $\bigcup_{x \in A} x^{\leftarrow} \subseteq \bigcup_{x \in B} x^{\leftarrow}$. Finally, $\downarrow$ is distributive over $\cup$ by definition. □

The closure operators $(\,)^{\rightarrow\leftarrow}$ and $\downarrow$ do not collapse in general, but they do on singletons, since $x^{\rightarrow\leftarrow} = x^{\leftarrow} = \downarrow x$ for every $x \in M$. Any subset A such that $\downarrow A \subseteq A$ is called *down saturated*. The class of down saturated subsets includes the one of left saturated subsets.

Proposition 4. *For every subset $A \subseteq M$ it holds $\downarrow A \subseteq A^{\rightarrow\leftarrow}$. Thus every left saturated subset is down saturated.*

Proof. Let $z \in \downarrow A$, then $z \leq x$ for $x \in A$. Now, $A \subseteq y^{\leftarrow}$ implies $x \in y^{\leftarrow}$, then $z \in y^{\leftarrow}$ by transitivity, and conclude $z \in A^{\rightarrow\leftarrow}$. If A is left saturated, then $\downarrow A \subseteq A^{\rightarrow\leftarrow} = A$. □

As already noticed, Lemma 2 says essentially that when the relation is a preorder the operator $(\,)^{\rightarrow\leftarrow}$ corresponds to the Dedekind-MacNeille completion (cf. [61]). In fact, $x \epsilon A^{\rightarrow\leftarrow}$ if and only if $A \subseteq \downarrow z$ implies $x \epsilon \downarrow z$ for every $z \in M$. This is just the completion used in [127] to define the syntactical model for Intuitionistic Linear Logic and Intuitionistic Logic. This hints that the relational semantics may be extended to these two logics.

1.4 Soundness

This section defines how to interpret the formulae of the language $\mathcal{L}$ as saturated subsets in any relational monoid $\mathcal{M} = (M, \cdot, 1, R)$. It also proves a soundness theorem for such an interpretation. A completeness theorem is given in §1.5.

The main idea is to think of M as the set of resources in a production cycle with a representative, or *null*, resource (the neutral element "1") and a way of *combining* resources (the monoidal operation "$\cdot$"). Here the relation R represents the generation of resources, and the triple xRy expresses that *the resource x can produce the resource y*. Resource x is the (possible) *ingredient* and y the (possible) *product*.

Section 1.3 pointed out that any element in $Sat^{\leftarrow}(M)$ is of the form $B^{\leftarrow}$, namely it is the subset of the ingredients that can produce every resource in B. Equivalently any element in $Sat^{\rightarrow}(M)$ is of the form $A^{\rightarrow}$: it is the subset of all the products that can be obtained by using whatever resource in A. Intuitively, think of an element in the collection $Sat^{\leftarrow}(M)$ as a subset of (possible) ingredients, and of an element in $Sat^{\rightarrow}(M)$ as a subset of (possible) products.

The operation $\cdot$ in M is the composition of resources. To combine the resource x with y (in this order), produces the resource $x \cdot y$. In $x \cdot y$ the resources x and y are connected to each other, neither x nor y can be isolated. In particular 1 represents the resource that does not modify the resource which it is combined with.

The combination between two subsets A, B of resource is just the subset $A \cdot B$ formed by all the possible combinations between a resource of A a resource of B, namely the algebraic product between the two subsets.

Every formula is associated with a pair of saturated subset of M: a subset of ingredients (left saturated) and a subset of products (right saturated). Theorem 1 says that every left saturated subset (ingredients) determines one and only one right saturated subset (products), so there is no need to choose two saturated subsets to evaluate a formula: once a left saturated subset is chosen, the operator $(\,)^{\rightarrow}$ automatically specifies the corresponding right saturated one.

Let Frm be the set of formulae in the language $\mathcal{L}$. The *evaluation of formulae* is the function

$$V(\cdot) \quad : \quad Frm \longrightarrow \ Sat^{\leftarrow}(M).$$

It will associate every formula φ with a subset $V(\varphi)$ of ingredients, and, clearly, with the subset $V(\varphi)^{\rightarrow}$ of products.

For any propositional variable p, the value $V(p)$ in $Sat^{\leftarrow}(M)$ is assumed to be given. Then the definitional equations in Tab. 1.1 give the evaluation of constants and the inductive cases for the connectives. The only thing to fix is the interpretation of a sequent $\Gamma \vdash \Delta$, then the definition of V follows straightforward.

So suppose V to be already defined on all formulae, and first define the evaluation of the contexts that form a sequent. By reading the sequent $\Gamma \vdash \Delta$ as Γ *can produce* Δ *in the calculus* **B**, it becomes natural to associate Γ with ingredients and Δ with products. It is intuitive to associate $\Gamma = \varphi_1, \ldots, \varphi_m$ with the combination of ingredients $Ingr(\Gamma) \stackrel{\text{def}}{=} V(\varphi_1) \cdot \ldots \cdot V(\varphi_m)$, and $\Delta = \psi_1, \ldots, \psi_n$ with the combination of products $Prod(\Delta) \stackrel{\text{def}}{=} V(\psi_1)^{\rightarrow} \cdot \ldots \cdot V(\psi_n)^{\rightarrow}$.

A particular case is that of the empty context. The behaviour of the empty context in the set of formulae and the one of the neutral element in the monoid are very much alike. In fact the empty list [ ] is neutral respect to the composition with formulae, as §1.5 will show in the syntactic model. So it is natural to define $Ingr([\,]) \stackrel{\text{def}}{=} \{1\}$ and $Prod([\,]) \stackrel{\text{def}}{=} \{1\}$.

Formally, for any context $\Sigma = \sigma_1, \ldots, \sigma_m$, where $m \geq 0$, set:

$$Ingr(\Sigma) \stackrel{\text{def}}{=} \{1\} \cdot V(\sigma_1) \cdot \ldots \cdot V(\sigma_m), \tag{1.14}$$

$$Prod(\Sigma) \stackrel{\text{def}}{=} \{1\} \cdot V(\sigma_1)^{\rightarrow} \cdot \ldots \cdot V(\sigma_m)^{\rightarrow}. \tag{1.15}$$

Note that there is no ambiguity, as the monoidal operation is associative. Moreover both products contain the subset $\{1\}$, to enable the evaluation of the empty context, as just said. If the context is formed by one or more formula, then the subset $\{1\}$ does not influence the product, as it is neutral for the product between subsets. If a contexts is formed by exactly one formula φ, then $Ingr(\varphi)$ reduces to $V(\varphi)$, the ingredients associated with φ, and $Prod(\varphi)$ reduces to $V(\varphi)^{\rightarrow}$, the products associated with φ.

By rephrasing the intuition given above, a sequent $\Gamma \vdash \Delta$ is valid if *every resource associated with* Γ *can produce every resource associated with* Δ. Formally, the sequent $\Gamma \vdash \Delta$ is valid in the monoid $\mathcal{M}$ if and only if $Ingr(\Gamma) \subseteq Prod(\Delta)^{\leftarrow}$, meaning that *the resources associated with* Γ *are ingredients for the resources associated with* Δ, or equivalently, by (1.3), if and only if $Prod(\Delta) \subseteq Ingr(\Gamma)^{\rightarrow}$, meaning that *the resources associated with* Δ *are products of the resources associated with* Γ. Taking a step back, the evaluation V on formulae follows by revising the definitional equations with the idea of the production cycle. Essentially, we rewrite the definitional equations, in Fig. 1.1, by following the definition of sequent validity. Depending on the case, we will choose between the equivalent definitions. In the following we discuss every single connective.

Connective $\&$. The definitional equation says that: *'*$Ingr(\Gamma) \subseteq Prod(\varphi \,\&\, \psi)^{\leftarrow}$ *if and only if* $Ingr(\Gamma) \subseteq Prod(\varphi)^{\leftarrow}$ *and* $Ingr(\Gamma) \subseteq Prod(\psi)^{\leftarrow}$*.'* As $Prod(\sigma)^{\leftarrow} = V(\sigma)$ for every single formula σ, the previous equation is equivalent to: *'*$Ingr(\Gamma) \subseteq V(p \,\&\, q)$ *if and only if* $Ingr(\Gamma) \subseteq V(p)$ *and* $Ingr(\Gamma) \subseteq V(q)$*.'* This means that the connective $\&$ is associated with meet (intersection) between left saturated subsets, and the definition must be:

$$V(\varphi \,\&\, \psi) \stackrel{\text{def}}{=} V(\varphi) \cap V(\psi).$$

Connective $\oplus$. The definitional equation says that: *'Prod*$(\Delta) \subseteq$ *Ingr*$(\psi \oplus \varphi)^{\rightarrow}$ *if and only if Prod*$(\Delta) \subseteq$ *Ingr*$(\psi)^{\rightarrow}$ *and Prod*$(\Delta) \subseteq$ *Ingr*$(\varphi)^{\rightarrow}$*.'* As *Ingr*$(\sigma)^{\rightarrow} = V(\sigma)^{\rightarrow}$ for every single formula σ, such an equation is equivalent to: *'Prod*$(\Delta) \subseteq V(\psi \oplus \varphi)^{\rightarrow}$ *if and only if Prod*$(\Delta) \subseteq V(\psi)^{\rightarrow}$ *and Prod*$(\Delta) \subseteq V(\varphi)^{\rightarrow}$*.'* This means that $\oplus$ is associated with meet (intersection) between right saturated subsets, and so:

$$V(\psi \oplus \varphi)^{\rightarrow} \stackrel{\text{def}}{=} V(\psi)^{\rightarrow} \cap V(\varphi)^{\rightarrow}.$$

Finally, by (1.9),

$$V(\psi \oplus \varphi) = V(\psi \oplus \varphi)^{\rightarrow\leftarrow} = (\, V(\psi)^{\rightarrow} \cap V(\varphi)^{\rightarrow} \,)^{\leftarrow} = (\, V(\psi) \cup V(\varphi) \,)^{\rightarrow\leftarrow}$$

that is the join for left saturated subsets.

Connective $\parr$. According to Tab. 1.1, *'Ingr*$(\Gamma) \subseteq$ *Prod*$(\varphi \parr \psi)^{\leftarrow}$ *if and only if Ingr*$(\Gamma) \subseteq$ *Prod*$(\varphi, \psi)^{\leftarrow}$*.'* Since *Prod*$(\varphi \parr \psi)^{\leftarrow} = V(\varphi \parr \psi)$, the equation says that

$$\textit{Ingr}(\Gamma) \subseteq V(\varphi \parr \psi) \textit{ if and only if } \textit{Ingr}(\Gamma) \subseteq \textit{Prod}(\varphi, \psi)^{\leftarrow}. \tag{1.16}$$

This means that the definition must be:

$$V(\varphi \parr \psi) \stackrel{\text{def}}{=} \textit{Prod}(\varphi, \psi)^{\leftarrow} = (\, V(\varphi)^{\rightarrow} \cdot V(\psi)^{\rightarrow} \,)^{\leftarrow}.$$

In fact, the forward direction of (1.16) says that $V(\varphi \parr \psi) \subseteq \textit{Prod}(\varphi, \psi)^{\leftarrow}$, by choosing $\Gamma = \varphi \parr \psi$; while the backward direction says $\textit{Prod}(\varphi, \psi)^{\leftarrow} \subseteq V(\varphi \parr \psi)$, by choosing Γ to be an atomic formula p such that $V(p) \stackrel{\text{def}}{=} \textit{Prod}(\varphi, \psi)^{\leftarrow}$.

Connective $\otimes$. According to Tab. 1.1, *'Prod*$(\Delta) \subseteq$ *Ingr*$(\psi \otimes \varphi)^{\rightarrow}$ *if and only if Prod*$(\Delta) \subseteq$ *Ingr*$(\psi, \varphi)^{\rightarrow}$*.'* By following symmetric a reasoning with respect to the one for $\parr$, it is easy to see that the definition must be *Ingr*$(\psi \otimes \varphi)^{\rightarrow} =$ *Ingr*$(\psi, \varphi)^{\rightarrow}$ and so:

$$V(\psi \otimes \varphi) \stackrel{\text{def}}{=} \textit{Ingr}(\psi, \varphi)^{\rightarrow\leftarrow} = (\, V(\psi) \cdot V(\varphi) \,)^{\rightarrow\leftarrow}.$$

Constant 1. By Tab. 1.1: *'Prod*$(\Delta) \subseteq$ *Ingr*$(1)^{\rightarrow}$ *if and only if Prod*$(\Delta) \subseteq$ *Ingr*$([\,])^{\rightarrow}$*.'* So the only possibility is to define *Ingr*$(1)^{\rightarrow} \stackrel{\text{def}}{=}$ *Ingr*$([\,])^{\rightarrow}$, hence

$$V(1) \stackrel{\text{def}}{=} \textit{Ingr}([\,])^{\rightarrow\leftarrow} = \{1\}^{\rightarrow\leftarrow}.$$

Constant $\bot$. By Tab. 1.1: *'Ingr*$(\Gamma) \subseteq$ *Prod*$(\bot)^{\leftarrow}$ *if and only if Ingr*$(\Gamma) \subseteq$ *Prod*$([\,])^{\leftarrow}$*,'* then the evaluation has to be

$$V(\bot) \stackrel{\text{def}}{=} \textit{Prod}([\,])^{\leftarrow} = \{1\}^{\leftarrow}.$$

Constant 0. Table 1.1 says that the subset of products associated with 0 must be as big as possible. The biggest right saturated subset is M. Therefore $V(0)^{\rightarrow} \stackrel{\text{def}}{=} M$ and so:

$$V(0) \stackrel{\text{def}}{=} M^{\leftarrow} = \emptyset^{\rightarrow\leftarrow}.$$

Constant $\top$. Table 1.1 says that the subset of ingredients associated with $\top$ must be as big as possible. The biggest left saturated subset is M again, so

$$V(\top) \stackrel{\text{def}}{=} M.$$

The previous intuitive explanations justify the following formal definition.

Table 1.2 Evaluation of Formulae

$V(\top)$	$\stackrel{\text{def}}{=} M$	$V(0)$	$\stackrel{\text{def}}{=} \emptyset^{\rightarrow\leftarrow}$
$V(1)$	$\stackrel{\text{def}}{=} \{1\}^{\rightarrow\leftarrow}$	$V(\bot)$	$\stackrel{\text{def}}{=} \{1\}^{\leftarrow}$
$V(\varphi \,\&\, \psi)$	$\stackrel{\text{def}}{=} V(\varphi) \cap V(\psi)$	$V(\psi \oplus \varphi)$	$\stackrel{\text{def}}{=} (\, V(\psi) \cup V(\varphi)\,)^{\rightarrow\leftarrow}$
$V(\psi \otimes \varphi)$	$\stackrel{\text{def}}{=} (\, V(\psi) \cdot V(\varphi)\,)^{\rightarrow\leftarrow}$	$V(\varphi \parr \psi)$	$\stackrel{\text{def}}{=} (\, V(\varphi)^{\rightarrow} \cdot V(\psi)^{\rightarrow}\,)^{\leftarrow}$

Definition 2 (Inductive Definition of Validity). *Let $\mathcal{M} = (M, \cdot, 1, R)$ be a relational monoid. A given assignment V of subsets $V(p), V(q), \ldots$ of $Sat^{\leftarrow}(M)$ to propositional variables $p, q, \ldots$ is extended to an evaluation V of all formulae by the inductive clauses in Tab. 1.2. Moreover, for every list $\Sigma = \varphi_1, \ldots, \varphi_m$ (with $m \geq 0$):*

$$\begin{aligned} \mathit{Ingr}(\Sigma) &\stackrel{\text{def}}{=} \{\,1\,\} \cdot V(\varphi_1) \cdot \ldots \cdot V(\varphi_m); \\ \mathit{Prod}(\Sigma) &\stackrel{\text{def}}{=} \{\,1\,\} \cdot V(\varphi_1)^{\rightarrow} \cdot \ldots \cdot V(\varphi_m)^{\rightarrow}. \end{aligned}$$

A sequent $\Gamma \vdash \Delta$ is valid under the evaluation *V if $\mathit{Ingr}(\Gamma) \subseteq \mathit{Prod}(\Delta)^{\leftarrow}$ (or equivalently if $\mathit{Prod}(\Delta) \subseteq \mathit{Ingr}(\Gamma)^{\rightarrow}$), and* valid in *$\mathcal{M}$ if it is valid under any evaluation V in $\mathcal{M}$.*

The intuitions given above hints that a soundness theorem holds, as formally stated below.

Theorem 2 (Soundness). *A sequent deducible in* **B** *is valid in every relational monoid.*

Proof. Rather than a long and detailed proof showing that axioms are valid, and that each rule preserves validity, as it is usually done, a full proof is obtained by showing the validity of definitional equations. In fact, this is equivalent to the validity of rules. This has already been done by introducing the evaluation function! So, just the validity for *cut* rules is needed to prove. Indeed it holds as the product of subsets preserves inclusion. □

As one can see, after all the semantics is fairly standard and recalls the one for Linear Logic: the connective $\&$ is associated to intersection, $\oplus$ to union, $\otimes$ to product, $\top$ to the whole set, and so on. This is not a blame but rather a good sign, as it means that the definitional equations do their job properly, by correctly reflecting the meta-level. Indeed, the novelty of this semantics is not the semantics itself, but the approach of solving the definitional equations. And, of course, this is the first semantics given to Basic Logic.

1.5 Completeness

This section proves a slight refinement of the usual completeness theorem, that we call *refined completeness theorem*. A similar result can be found in [112] with a *strong completeness* theorem for Intuitionistic Linear Logic and its extensions. Here the adjective 'refined' is preferred to 'strong,' as it is more descriptive and does not lead to confusion

with the common idea of strong completeness, mainly used in modal logical settings. A canonical model is carefully built and the theorem is 'refined' to not only prove the semantical completeness, but also to provide *cut-eliminability*, the normal form theorem in a sequent calculus. Its proof relies on a particular relational monoid: the *syntactic model*. It will not only prove that a sequent is valid in the syntactic model if and only if it is derivable in **B**, but moreover if and only if it is provable in **B** *without* using *cut* rules. The production cycle idea is still relevant: provability is the monoidal relation, antecedents and consequents of a sequent represent ingredients and products, respectively.

Definition 3 (Syntactic Model). *The* syntactic model $\mathcal{F}$ *is the structure* $(Frm^*, \circ, [\,], \vdash_{\mathbf{B}})$, *where:*

a. The set Frm^* *includes all the (possibly empty) finite lists of formulae.*

b. The operation $\circ$ *is the concatenation between lists, namely* $\Gamma_1 \circ \Gamma_2 \stackrel{\text{def}}{=} \Gamma_1, \Gamma_2$.

c. The symbol [ ] *represents the empty list.*

d. The relation $\vdash_{\mathbf{B}}$ *says that:* $\Gamma \vdash_{\mathbf{B}} \Delta$ *if and only if* $\Gamma \vdash \Delta$ *is derivable in* **B** *without using any cut rule.*

The structure $\mathcal{F}$ is indeed a relational monoid: the concatenation between lists is associative and [ ] is its neutral element, since $\Gamma \circ [\,] = [\,] \circ \Gamma = \Gamma$.

The operators $(\,)^{\leftarrow}$ and $(\,)^{\rightarrow}$ assume a particular significance in $\mathcal{F}$. The definitions in (1.2) say that $\Sigma^{\rightarrow} \stackrel{\text{def}}{=} \{\Delta \in Frm^* : \Sigma \vdash_{\mathbf{B}} \Delta\}$ and $\Sigma^{\leftarrow} \stackrel{\text{def}}{=} \{\Gamma \in Frm^* : \Gamma \vdash_{\mathbf{B}} \Sigma\}$, hence these subsets identify the consequents and the antecedents of any context Σ. Generally, for any subset $A \subseteq Frm^*$, $A^{\rightarrow}$ identifies all the consequents that can be proved by every context in A without cuts, and $A^{\leftarrow}$ describes all antecedents proving every context in A. In fact, the definitions say:

$$A^{\rightarrow} \stackrel{\text{def}}{=} \{\Delta \in Frm^* : \Gamma \vdash_{\mathbf{B}} \Delta \text{ for all } \Gamma \,\epsilon\, A\};$$
$$A^{\leftarrow} \stackrel{\text{def}}{=} \{\Gamma \in Frm^* : \Gamma \vdash_{\mathbf{B}} \Delta \text{ for all } \Delta \,\epsilon\, A\}.$$

Derivations in the calculus **B** can be produced exclusively by the rules in Fig. 1.1. Since the relation $\vdash_{\mathbf{B}}$ requires a cut-free derivation, only the rules involving constants and connectives establish if two elements are related in the canonical model. The two cut rules cannot be used. Every rule of the cut-free calculus expresses a particular property for the operators $(\,)^{\rightarrow}$ and $(\,)^{\leftarrow}$ in the syntactic model. The correspondence between rules and model properties is stated in Tab. 1.3 that reports the rules of the sequent calculus in semantical terms, as can be checked.

The derived model properties play a prominent role in proving the Canonical Evaluation Lemma, preparatory to the main theorem. As usual, the lemma is based on a particular evaluation V of formulae in $\mathcal{F}$, called *canonical evaluation*, that evaluates every propositional variable p with the subset of all the contexts proving p without cut rules,

Table 1.3 Syntactic Properties

Axioms	$: \varphi \,\epsilon (\varphi)^{\rightarrow}$	*Axioms*	$: \varphi \,\epsilon (\varphi)^{\leftarrow}$
$\otimes L$	$: (\varphi \circ \psi)^{\rightarrow} \subseteq (\varphi \otimes \psi)^{\rightarrow}$	$\parr R$	$: (\varphi \circ \psi)^{\leftarrow} \subseteq (\varphi \parr \psi)^{\leftarrow}$
$\parr L$	$: (\psi)^{\rightarrow} \circ (\varphi)^{\rightarrow} \subseteq (\psi \parr \varphi)^{\rightarrow}$	$\otimes R$	$: (\psi)^{\leftarrow} \circ (\varphi)^{\leftarrow} \subseteq (\psi \otimes \varphi)^{\leftarrow}$
$1L$	$: ([\,])^{\rightarrow} \subseteq (1)^{\rightarrow}$	$\bot R$	$: ([\,])^{\leftarrow} \subseteq (\bot)^{\leftarrow}$
$\bot L$	$: \bot \,\epsilon ([\,])^{\leftarrow}$	$1R$	$: 1 \,\epsilon ([\,])^{\rightarrow}$
$\oplus L$	$: (\varphi)^{\rightarrow} \cap (\psi)^{\rightarrow} \subseteq (\psi \oplus \varphi)^{\rightarrow}$	$\& R$	$: (\varphi)^{\leftarrow} \cap (\psi)^{\leftarrow} \subseteq (\psi \,\&\, \varphi)^{\leftarrow}$
$\& L$	$: (\psi)^{\rightarrow} \cup (\varphi)^{\rightarrow} \subseteq (\psi \,\&\, \varphi)^{\rightarrow}$	$\oplus R$	$: (\psi)^{\leftarrow} \cup (\varphi)^{\leftarrow} \subseteq (\psi \oplus \varphi)^{\leftarrow}$
$0L$	$: Frm^{*} \subseteq (0)^{\rightarrow}$	$\top R$	$: Frm^{*} \subseteq (\top)^{\leftarrow}$

i.e. $V(p) \stackrel{\text{def}}{=} \{p\}^{\leftarrow}$. Such a subset is left saturated by Corollary 1. Thanks to the axioms of the calculus, the evaluation satisfies the property $p \,\epsilon\, V(p) \subseteq \{p\}^{\leftarrow}$. This property is inherited by every formula, as formally proved by the canonical evaluation lemma itself.

Lemma 4 (Canonical Evaluation). *Under the canonical evaluation V in $\mathcal{F}$:*

1. $\varphi \,\epsilon\, V(\varphi) \subseteq \varphi^{\leftarrow}$ *for every formula φ of $\mathcal{L}$.*
2. $\Sigma \,\epsilon\, Ingr(\Sigma)$ *and* $Prod(\Sigma)^{\leftarrow} \subseteq \Sigma^{\leftarrow}$ *for every context Σ.*

Proof. For the first point apply an induction on the structure of formulae. The thesis is verified on propositional variables by hypothesis. The properties in Tab. 1.3 provide the basic steps on constants.

Case $\top$. As $V(\top) \stackrel{\text{def}}{=} Frm^{*}$, then $\top \,\epsilon\, V(\top)$, and $V(\top) \subseteq (\top)^{\leftarrow}$ by $\top R$.

Case 0. As $V(0) \stackrel{\text{def}}{=} \emptyset^{\rightarrow\leftarrow} = (Frm^{*})^{\leftarrow}$, then $0 \,\epsilon\, V(0)$ by (1.8) applied to $0L$, and $V(0) \subseteq (0)^{\leftarrow}$, since $\emptyset^{\rightarrow\leftarrow}$ is the minimum among left saturated subsets.

Case $\bot$. As $V(\bot) \stackrel{\text{def}}{=} [\,]^{\leftarrow}$, then $\bot \,\epsilon\, V(\bot) \subseteq (\bot)^{\leftarrow}$ by $\bot L$ and $\bot R$.

Case 1. As $V(1) \stackrel{\text{def}}{=} [\,]^{\rightarrow\leftarrow}$, then $1 \,\epsilon (1)^{\rightarrow\leftarrow} \subseteq V(1)$ by (1.8) applied to $\bot L$, and $V(1) \subseteq (1)^{\leftarrow}$ by (1.8) applied to $\bot R$.

The induction step deals with connectives and assumes two induction hypothesis, by considering the sub-formulae of the current formula: hypothesis (a) says that $\varphi \,\epsilon\, V(\varphi)$ and $\psi \,\epsilon\, V(\psi)$; hypothesis (b) says that $V(\varphi) \subseteq (\varphi)^{\leftarrow}$ and $V(\psi) \subseteq (\psi)^{\leftarrow}$.

Case $\parr$. As $V(\varphi \parr \psi) \stackrel{\text{def}}{=} (\, V(\varphi)^{\rightarrow} \circ V(\psi)^{\rightarrow}\,)^{\leftarrow}$, then:

$$
\begin{aligned}
\varphi \parr \psi \;&\epsilon \{\varphi \parr \psi\}^{\rightarrow\leftarrow} \\
&\subseteq (\{\varphi\}^{\rightarrow} \circ \{\psi\}^{\rightarrow})^{\leftarrow} && \text{by (1.8) applied to } \parr L \\
&\subseteq V(\varphi \parr \psi) && \text{by Lemma 3 applied to hyp. (a);}
\end{aligned}
$$

$$\begin{aligned} V(\varphi \parr \psi) &\subseteq ((\varphi)^{\leftarrow\rightarrow} \circ (\psi)^{\leftarrow\rightarrow})^{\leftarrow} && \text{by Lemma 3 applied to hyp. (b)} \\ &\subseteq (\varphi \circ \psi)^{\leftarrow} && \text{as } \varphi \,\epsilon (\varphi)^{\rightarrow\leftarrow} \text{ and } \psi \,\epsilon (\psi)^{\rightarrow\leftarrow} \\ &\subseteq (\varphi \parr \psi)^{\leftarrow} && \text{by } \parr R. \end{aligned}$$

Case $\otimes$. As $V(\psi \otimes \varphi) \stackrel{\text{def}}{=} (\, V(\psi) \circ V(\varphi)\,)^{\rightarrow\leftarrow}$, then:

$$\begin{aligned} \varphi \otimes \psi \;&\epsilon \{\varphi \otimes \psi\}^{\rightarrow\leftarrow} \\ &\subseteq (\varphi \circ \psi)^{\rightarrow\leftarrow} && \text{by (1.8) applied to } \otimes L \\ &\subseteq V(\varphi \otimes \psi) && \text{by hyp. (a);} \end{aligned}$$

$$\begin{aligned} V(\varphi \otimes \psi) &\subseteq ((\varphi)^{\leftarrow} \circ (\psi)^{\leftarrow})^{\rightarrow\leftarrow} && \text{by hyp. (b)} \\ &\subseteq ((\varphi \otimes \psi)^{\leftarrow})^{\rightarrow\leftarrow} && \text{by } \otimes R \\ &\subseteq (\varphi \otimes \psi)^{\leftarrow} && \text{by Corollary 1.} \end{aligned}$$

Case $\&$. As $V(\varphi \,\&\, \psi) \stackrel{\text{def}}{=} V(\varphi) \cap V(\psi)$, then:

$$\begin{aligned} \varphi \,\&\, \psi \;&\epsilon \{\varphi \,\&\, \psi\}^{\rightarrow\leftarrow} \\ &\subseteq (\{\varphi\}^{\rightarrow} \cup \{\psi\}^{\rightarrow})^{\leftarrow} && \text{by (1.8) applied to } \&\, L \\ &\subseteq (\, V(\varphi)^{\rightarrow} \cup V(\psi)^{\rightarrow}\,)^{\leftarrow} && \text{by (1.7) applied to hyp. (a), and (1.8)} \\ &= (\, V(\varphi) \cap V(\psi)\,)^{\rightarrow\leftarrow} && \text{by (1.10)} \\ &= V(\varphi \,\&\, \psi); && \text{as } V(\varphi) \cap V(\psi) \text{ is left saturated;} \end{aligned}$$

$$\begin{aligned} V(\varphi \,\&\, \psi) &\subseteq (\varphi)^{\leftarrow} \cap (\psi)^{\leftarrow} && \text{by hyp. (b)} \\ &\subseteq (\varphi \,\&\, \psi)^{\leftarrow} && \text{by } \&\, R. \end{aligned}$$

Case $\oplus$. As $V(\psi \oplus \varphi) \stackrel{\text{def}}{=} (\, V(\varphi) \cup V(\psi)\,)^{\rightarrow\leftarrow}$ then:

$$\begin{aligned} \varphi \oplus \psi \;&\epsilon \{\varphi \oplus \psi\}^{\rightarrow\leftarrow} \\ &\subseteq (\{\varphi\}^{\rightarrow} \cap \{\psi\}^{\rightarrow})^{\leftarrow} && \text{by (1.8) applied to } \oplus L \\ &\subseteq (\, V(\varphi)^{\rightarrow} \cap V(\psi)^{\rightarrow}\,)^{\leftarrow} && \text{by (1.7) applied to hyp. (a), and (1.8)} \\ &= V(\varphi \oplus \psi) && \text{by (1.10);} \end{aligned}$$

$$\begin{aligned} V(\varphi \oplus \psi) &\subseteq (\varphi^{\leftarrow} \cup \psi^{\leftarrow})^{\rightarrow\leftarrow} && \text{by hyp. (b)} \\ &\subseteq ((\varphi \oplus \psi)^{\leftarrow})^{\rightarrow\leftarrow} && \text{by } \oplus R \\ &= (\varphi \oplus \psi)^{\leftarrow} && \text{by Corollary 1.} \end{aligned}$$

For the second point, consider any list of formulae $\Sigma = \sigma_1, \ldots, \sigma_m$. When $m = 0$, the point is verified by Definition 2, as [ ] is just the syntactic neutral element. When

$m \geq 1$, the property $\sigma_i \epsilon V(\sigma_i)$ for $i = 1 \dots m$ implies $\sigma_1, \dots, \sigma_m \epsilon V(\sigma_1) \circ \dots \circ V(\sigma_m)$, that is $\Sigma \epsilon Ingr(\Sigma)$. Moreover, for every $i = 1 \dots m$, the property $V(\sigma_i) \subseteq \sigma_i{}^{\leftarrow}$ implies $\sigma_i \epsilon V(\sigma_i)^{\rightarrow}$, hence $\sigma_1, \dots, \sigma_m \epsilon V(\sigma_1)^{\rightarrow} \circ \dots \circ V(\sigma_m)^{\rightarrow}$ that means $\Sigma \epsilon Prod(\Sigma)$, hence $Prod(\Sigma)^{\leftarrow} \subseteq \Sigma^{\leftarrow}$. □

The canonical evaluation lemma is all that is needed to prove the refined completeness theorem.

Theorem 3 (Refined Completeness). *If a sequent is valid in every relational monoid then it is derivable in* **B** *without using cut rules.*

Proof. Let $\Gamma \vdash \Delta$ be a sequent valid in every relational monoid. In particular, $Ingr(\Gamma) \subseteq Prod(\Delta)^{\leftarrow}$ in the syntactic model equipped with the canonical evaluation. Lemma 4 says that $\Gamma \epsilon Ingr(\Gamma)$ and $Prod(\Delta)^{\leftarrow} \subseteq \Delta^{\leftarrow}$, hence $\Gamma \epsilon \Delta^{\leftarrow}$, that is $\Gamma \vdash_{\mathbf{B}} \Delta$, namely $\Gamma \vdash \Delta$ is derivable in **B** without cut rules. □

The combination between soundness and refined completeness gives a semantical proof to the already known (cf. [129]) cut-elimination property in **B**. A 'cut' occurring in a derivation is an application of any cut rule, a 'cut-free' derivation does not exhibit cuts.

Theorem 4 (Semantical Cut Elimination). *If a sequent is derivable in* **B** *(even by using cut rules), then it admits a cut-free derivation.*

Proof. First apply Theorem 2, then Theorem 3: a sequent derivable in **B** is valid in every relational monoid, hence it is derivable without cuts. □

The cut-elimination theorem allows a better characterisation of the closure operators in the syntactical model. In fact, the redundancy of cut rules says that an equivalent calculus is obtained from **B** by removing cuts. In particular, if the sequents $\Gamma \vdash \varphi$ and $\varphi \vdash \Delta$ admit a cut-free derivation, then so does the sequent $\Gamma \vdash \Delta$. This fact means that in the syntactic model, for every formula φ:

$$\varphi^{\rightarrow\leftarrow} = \varphi^{\leftarrow}. \tag{1.17}$$

In fact, the inclusion $\varphi^{\rightarrow\leftarrow} \subseteq \varphi^{\leftarrow}$ holds as $\varphi \epsilon \varphi^{\leftarrow}$ by axioms and $\varphi^{\leftarrow}$ is left saturated. For the inclusion $\varphi^{\leftarrow} \subseteq \varphi^{\rightarrow\leftarrow}$, assume $\Gamma \epsilon \varphi^{\leftarrow}$, this means $\Gamma \vdash_{\mathbf{B}} \varphi$, then for every Δ such that $\varphi \vdash_{\mathbf{B}} \Delta$ it is the case that $\Gamma \vdash_{\mathbf{B}} \Delta$, as previously noticed, hence conclude $\Gamma \epsilon \varphi^{\rightarrow\leftarrow}$. Furthermore, $\varphi^{\leftarrow\rightarrow} = \varphi^{\rightarrow}$ by symmetry.

The previous property helps in further specifying the canonical evaluation V. Moreover Lemma 4 proves that $\varphi \epsilon V(\varphi) \subseteq \varphi^{\leftarrow}$, hence $\varphi^{\rightarrow\leftarrow} \subseteq V(\varphi) \subseteq \varphi^{\leftarrow}$ as $V(\varphi)$ is left saturated. Since the subsets on the sides coincide by (1.17), it is straightforward to conclude that for every formula φ it holds

$$V(\varphi) = \varphi^{\leftarrow} \text{ and } V(\varphi)^{\rightarrow} = \varphi^{\rightarrow}. \tag{1.18}$$

A similar characterisation exists when evaluating contexts in a sequent. To state this, it is worth to prove an intuitive extension to cut rules in the following lemma.

Lemma 5. *In the calculus* **B** *the following hold:*

1. *The sequent* $\varphi_1, \ldots, \varphi_m \vdash \Delta$ *is derivable iff*

$$\frac{\Gamma_1 \vdash \varphi_1 \ \ldots \ \Gamma_m \vdash \varphi_m}{\Gamma_1, \ldots, \Gamma_m \vdash \Delta} \tag{1.19}$$

   is an admissible rule.

2. *The sequent* $\Gamma \vdash \psi_1, \ldots, \psi_n$ *is derivable iff*

$$\frac{\psi_1 \vdash \Delta_1 \ \ldots \ \psi_n \vdash \Delta_n}{\Gamma \vdash \Delta_1, \ldots, \Delta_n} \tag{1.20}$$

   is an admissible rule.

Proof. Case 1. Let $\varphi_1, \ldots, \varphi_m \vdash \Delta$ and assume $\Gamma_1 \vdash \varphi_1 \ \ldots \ \Gamma_m \vdash \varphi_m$, then use m instances of *cutL* and conclude:

$$\dfrac{\dfrac{\Gamma_1 \vdash \varphi_1 \quad \varphi_1, \ldots, \varphi_m \vdash \Delta}{\Gamma_1, \varphi_2, \ldots, \varphi_m \vdash \Delta}\ \mathit{cutL}}{\vdots}$$
$$\dfrac{\Gamma_m \vdash \varphi_m \qquad \Gamma_1, \ldots, \Gamma_{m-1}, \varphi_m \vdash \Delta}{\Gamma_1, \ldots, \Gamma_m \vdash \Delta}\ \mathit{cutL}$$

Vice versa, if (1.19) is admissible, the sequent $\varphi_1, \ldots, \varphi_m \vdash \Delta$ is derived from axioms $\varphi_i \vdash \varphi_i$ $(i = 1, \ldots, m)$. *Case b.* Symmetrically: use n instances of *cutR*, and consider axioms $\psi_i \vdash \psi_i$ $(i = 1, \ldots, n)$. □

Thanks to Theorem 4, it is straightforward to instantiate the previous lemma in terms of the syntactic relation $\vdash_{\mathbf{B}}$:

1. To say that $\varphi_1, \ldots, \varphi_m \vdash_{\mathbf{B}} \Delta$ is equivalent to say that $\Gamma_1 \vdash_{\mathbf{B}} \varphi_1 \ \ldots \ \Gamma_m \vdash_{\mathbf{B}} \varphi_m$ implies $\Gamma_1, \ldots, \Gamma_m \vdash_{\mathbf{B}} \Delta$ for every $\Gamma_1 \ldots \Gamma_m$.

2. To say that $\Gamma \vdash_{\mathbf{B}} \psi_1, \ldots, \psi_n$ is equivalent to say that $\psi_1 \vdash_{\mathbf{B}} \Delta_1 \ \ldots \ \psi_n \vdash_{\mathbf{B}} \Delta_n$ implies $\Gamma \vdash_{\mathbf{B}} \Delta_1, \ldots, \Delta_n$ for every $\Delta_1 \ldots \Delta_n$.

And in terms of the syntactic operators in $\mathcal{F}$:

$$(\varphi_1, \ldots, \varphi_m)^{\rightarrow} = (\varphi_1{}^{\leftarrow} \circ \ldots \circ \varphi_m{}^{\leftarrow})^{\rightarrow},$$
$$(\psi_1, \ldots, \psi_n)^{\leftarrow} = (\psi_1{}^{\rightarrow} \circ \ldots \circ \psi_n{}^{\rightarrow})^{\leftarrow}$$

then, by (1.18) and by considering the canonical evaluation V conclude

$$(\varphi_1, \ldots, \varphi_m)^{\rightarrow} = (V(\varphi_1) \circ \ldots \circ V(\varphi_m))^{\rightarrow}$$
$$(\psi_1, \ldots, \psi_n)^{\leftarrow} = (V(\psi_1)^{\rightarrow} \circ \ldots \circ V(\psi_n)^{\rightarrow})^{\leftarrow}$$

that, according to the definition, is

$$\Gamma^{\rightarrow} = \mathit{Ingr}(\Gamma)^{\rightarrow} \text{ and } \Delta^{\leftarrow} = \mathit{Prod}(\Delta)^{\leftarrow}. \tag{1.21}$$

Property (1.21) is the contextual equivalent of (1.18). Intuitively, it reinforces with a syntactical point of view the intuition given when defining the semantics: the subset of ingredients associated to Γ can produce "exactly" what Γ can produce, and the set of products associated to Δ is produced "exactly" by everything that produces Δ.

Note that property (1.17) does not extend up to context. In fact, the two subsets $\Sigma^{\leftarrow}$ and $\Sigma^{\rightarrow\leftarrow}$ cannot be compared. The inclusion $\Sigma^{\leftarrow} \subseteq \Sigma^{\rightarrow\leftarrow}$ would imply the stronger form of cut rule

$$\frac{\Gamma \vdash \Sigma \quad \Sigma \vdash \Delta}{\Gamma \vdash \Delta}$$

and the backward inclusion would imply $\Sigma \vdash \Sigma$ for any context Σ. Neither property is verified in the basic calculus. Indeed, it is not sensible to require the sequent calculus to satisfy them, as they would mistake the composition of formulae on the right hand side of a sequent for the one on the left hand side.

1.6 Towards Sub-Structural Logics

So far, the sequent calculus has not included any structural rule. This section hints how to extend the relational semantics to any calculus obtained by providing **B** with any group of the structural rules reported in Fig. 1.2: exchange (*e*), weakening (*w*), and contraction (*c*). Any choice of the rules generate a distinctive calculus. Considering any repetition-free list l built by the alphabet $\{e, w, c\}$, the notation $\mathbf{B}_l$ identifies the basic calculus extended by adding the corresponding structural rules. For instance: $\mathbf{B}_{ec}$ represents the *relevance* version of the calculus, with exchange and contraction [5, 6]; $\mathbf{B}_{ew}$ is the *affine* one, with exchange and weakening [10]; and $\mathbf{B}_{ecw}$ is the core of structural Basic Logic, **BS** [129]. Moreover, as seen in [64], $\mathbf{B}_{ecw}$ represents the core for a sequential formulation of Paraconsistent Quantum Logic [58] that is a weak form of Quantum Logic [20, 59].

It is worth remarking that Proposition 1, about cut-free definitional equations, still holds for any structural extension of **B**, as will be more generally shown in §1.7 for Intuitionistic Logic, that can be seen as a contextual structural extension of the basic calculus.

For every extension, the relational semantics essentially remains the same as in §1.5. The evaluation function for formulae and contexts does not change, and neither does sequent validity. It is sufficient to reduce the class of relational monoids so that the added structural rules are validated. The required properties on models will be naturally verified in the canonical model, so there is no need to modify the completeness proof.

To prove soundness by maintaining the formulae evaluation of §1.4, it is sufficient to find the right properties validating the added substructural rules. Every structural rule fixes a property of relational monoids. Such a correspondence is essentially obtained by

Figure 1.2 Structural Rules

$$\frac{\Gamma_1, \psi, \varphi, \Gamma_2 \vdash \Delta}{\Gamma_1, \varphi, \psi, \Gamma_2 \vdash \Delta}\ eL \qquad \frac{\Gamma \vdash \Delta_1, \varphi, \psi, \Delta_2}{\Gamma \vdash \Delta_1, \psi, \varphi, \Delta_2}\ eR$$

$$\frac{\Gamma_1, \Gamma_2 \vdash \Delta}{\Gamma_1, \psi, \Gamma_2 \vdash \Delta}\ wL \qquad \frac{\Gamma \vdash \Delta_1, \Delta_2}{\Gamma \vdash \Delta_1, \psi, \Delta_2}\ wR$$

$$\frac{\Gamma_1, \psi, \psi, \Gamma_2 \vdash \Delta}{\Gamma_1, \psi, \Gamma_2 \vdash \Delta}\ cL \qquad \frac{\Gamma \vdash \Delta_1, \psi, \psi, \Delta_2}{\Gamma \vdash \Delta_1, \psi, \Delta_2}\ cR$$

expressing any rule in terms of the semantical evaluation for sequents, which is in detail:

$$\begin{aligned}
eL{:}&\ \mathit{Ingr}(\Gamma_1, \varphi, \psi, \Gamma_2) \subseteq \mathit{Prod}(\Delta)^{\leftarrow} \text{ implies } \mathit{Ingr}(\Gamma_1, \psi, \varphi, \Gamma_2) \subseteq \mathit{Prod}(\Delta)^{\leftarrow}\\
eR{:}&\ \mathit{Ingr}(\Gamma) \subseteq \mathit{Prod}(\Delta_1, \psi, \varphi, \Delta_2)^{\leftarrow} \text{ implies } \mathit{Ingr}(\Gamma) \subseteq \mathit{Prod}(\Delta_1, \varphi, \psi, \Delta_2)^{\leftarrow}\\
wL{:}&\ \mathit{Ingr}(\Gamma_1, \Gamma_2) \subseteq \mathit{Prod}(\Delta)^{\leftarrow} \text{ implies } \mathit{Ingr}(\Gamma_1, \psi, \Gamma_2) \subseteq \mathit{Prod}(\Delta)^{\leftarrow}\\
wR{:}&\ \mathit{Ingr}(\Gamma) \subseteq \mathit{Prod}(\Delta_1, \Delta_2)^{\leftarrow} \text{ implies } \mathit{Ingr}(\Gamma) \subseteq \mathit{Prod}(\Delta_1, \psi, \Delta_2)^{\leftarrow}\\
cl{:}&\ \mathit{Ingr}(\Gamma_1, \psi, \psi, \Gamma_2) \subseteq \mathit{Prod}(\Delta)^{\leftarrow} \text{ implies } \mathit{Ingr}(\Gamma_1, \psi, \Gamma_2) \subseteq \mathit{Prod}(\Delta)^{\leftarrow}\\
cR{:}&\ \mathit{Ingr}(\Gamma) \subseteq \mathit{Prod}(\Delta_1, \psi, \psi, \Delta_2)^{\leftarrow} \text{ implies } \mathit{Ingr}(\Gamma) \subseteq \mathit{Prod}(\Delta_1, \psi, \Delta_2)^{\leftarrow}
\end{aligned}$$

for every choice of the involved contexts and formulae. Such a generalisation on contexts is equivalent to a generalisation on subsets in the model. Therefore, the previous group of properties is equivalent to:

$$\begin{aligned}
eL{:}&\ (C_1 \cdot B \cdot A \cdot C_2)^{\rightarrow\leftarrow} \subseteq D^{\leftarrow} \text{ implies } (C_1 \cdot A \cdot B \cdot C_2)^{\rightarrow\leftarrow} \subseteq D^{\leftarrow}\\
eR{:}&\ C^{\rightarrow\leftarrow} \subseteq (D_1 \cdot A \cdot B \cdot D_2)^{\leftarrow} \text{ implies } C^{\rightarrow\leftarrow} \subseteq (D_1 \cdot B \cdot A \cdot D_2)^{\leftarrow}\\
wL{:}&\ (C_1 \cdot C_2)^{\rightarrow\leftarrow} \subseteq D^{\leftarrow} \text{ implies } (C_1 \cdot A \cdot C_2)^{\rightarrow\leftarrow} \subseteq D^{\leftarrow}\\
wR{:}&\ C^{\rightarrow\leftarrow} \subseteq (D_1 \cdot D_2)^{\leftarrow} \text{ implies } C^{\rightarrow\leftarrow} \subseteq (D_1 \cdot A \cdot D_2)^{\leftarrow}\\
cl{:}&\ (C_1 \cdot A \cdot A \cdot C_2)^{\rightarrow\leftarrow} \subseteq D^{\leftarrow} \text{ implies } (C_1 \cdot A \cdot C_2)^{\rightarrow\leftarrow} \subseteq D^{\leftarrow}\\
cR{:}&\ C^{\rightarrow\leftarrow} \subseteq (D_1 \cdot A \cdot A \cdot D_2)^{\leftarrow} \text{ implies } C^{\rightarrow\leftarrow} \subseteq (D_1 \cdot A \cdot D_2)^{\leftarrow}
\end{aligned}$$

for every choice of the involved subsets. Now, due to the generality of D, for the 'L' properties, and C, for the 'R' ones, it is straightforward to see that the previous points correspond respectively to:

$$(C_1 \cdot B \cdot A \cdot C_2)^{\rightarrow\leftarrow} \subseteq (C_1 \cdot A \cdot B \cdot C_2)^{\rightarrow\leftarrow} \tag{1.22}$$

$$(D_1 \cdot B \cdot A \cdot D_2)^{\leftarrow} \subseteq (D_1 \cdot A \cdot B \cdot D_2)^{\leftarrow} \tag{1.23}$$

$$(C_1 \cdot A \cdot C_2)^{\rightarrow\leftarrow} \subseteq (C_1 \cdot C_2)^{\rightarrow\leftarrow} \tag{1.24}$$

$$(D_1 \cdot D_2)^{\leftarrow} \subseteq (D_1 \cdot A \cdot D_2)^{\leftarrow} \tag{1.25}$$

$$(C_1 \cdot A \cdot C_2)^{\rightarrow\leftarrow} \subseteq (C_1 \cdot A \cdot A \cdot C_2)^{\rightarrow\leftarrow} \tag{1.26}$$

$$(D_1 \cdot A \cdot A \cdot D_2)^{\leftarrow} \subseteq (D_1 \cdot A \cdot D_2)^{\leftarrow} \tag{1.27}$$

This line of argument leads to the conclusion that a model is sound for exchange rules whenever it satisfies properties (1.22) and (1.23), for weakening rules whenever it satisfies (1.24) and (1.25), and for contraction rules whenever it satisfies (1.26) and (1.27).

Table 1.4 Semantical Structural Properties

$(x_1 \cdot w \cdot z \cdot x_2)\, R\, y$	implies	$(x_1 \cdot z \cdot w \cdot x_2)\, R\, y$	$(e2)$
$x\, R\, (y_1 \cdot z \cdot w \cdot y_2)$	implies	$x\, R\, (y_1 \cdot w \cdot z \cdot y_2)$	$(e2)$
$(x_1 \cdot x_2)\, R\, y$	implies	$(x_1 \cdot w \cdot x_2)\, R\, y$	$(w1)$
$x\, R\, (y_1 \cdot y_2)$	implies	$x\, R\, (y_1 \cdot w \cdot y_2)$	$(w2)$
$(x_1 \cdot w \cdot w \cdot x_2)\, R\, y$	implies	$(x_1 \cdot w \cdot x_2)\, R\, y$	$(c1)$
$x\, R\, (y_1 \cdot w \cdot w \cdot y_2)$	implies	$x\, R\, (y_1 \cdot w \cdot y_2)$	$(c2)$

What a clear mathematical definition needs is a bunch of properties involving the main model constituents: monoid elements, binary relation and monoidal operation. As all previous properties concern operators on subsets, they must be analysed and reduced into equivalent ones on elements, thus projecting a second order property to first order. The pattern to follow is the same for every property. Consider (1.22) as a guideline. By Lemma 1, (1.22) is equivalent to

$$(C_1 \cdot A \cdot B \cdot C_2)^{\rightarrow} \subseteq (C_1 \cdot B \cdot A \cdot C_2)^{\rightarrow}; \tag{1.28}$$

and, by assuming $C_1 = \{x_1\}$, $A = \{w\}$, $B = \{z\}$ and $C_2 = \{x_2\}$, it specialises to

$$(x_1 \cdot w \cdot z \cdot x_2)^{\rightarrow} \subseteq (x_1 \cdot z \cdot w \cdot x_2)^{\rightarrow} \tag{1.29}$$

that is actually equivalent to (1.28). In fact, consider the subsets A, B, C_1, C_2 and assume that (1.29) holds for every choice of elements in M, then in particular $(x_1 \cdot w \cdot z \cdot x_2)^{\rightarrow} \subseteq (x_1 \cdot z \cdot w \cdot x_2)^{\rightarrow}$ for every $x_1 \epsilon C_1$, $w \epsilon A$, $z \epsilon B$ and $x_2 \epsilon C_2$, hence

$$\bigcap_{\substack{x_1 \epsilon C_1 \; w \epsilon A \\ x_2 \epsilon C_2 \; z \epsilon B}} \{x_1 \cdot w \cdot z \cdot x_2\}^{\rightarrow} \subseteq \bigcap_{\substack{x_1 \epsilon C_1 \; w \epsilon A \\ x_2 \epsilon C_2 \; z \epsilon B}} \{x_1 \cdot z \cdot w \cdot x_2\}^{\rightarrow},$$

that corresponds to (1.28), according to the definition in (1.5).

Property (1.29) is what we aimed for, as it corresponds to an 'elemental' property. In fact, by unfolding the definition for $(\,)^{\rightarrow}$, it corresponds to the rule

$$(x_1 \cdot w \cdot z \cdot x_2)\, R\, y \quad \text{implies} \quad (x_1 \cdot z \cdot w \cdot x_2)\, R\, y,$$

that emerges as the essential property to require in a relational model to validate eL. A similar reasoning can be applied to (1.23)...(1.27) in order to obtain equivalent 'elemental' properties. Table 1.4 outlines all these properties. The correspondence is clearly between $(c1)$ and (1.22) for eL, $(c2)$ and (1.23) for eR, $(w1)$ and (1.24) for wL, $(w2)$ and (1.25) for wR, $(c1)$ and (1.26) for cL, $(c2)$ and (1.27) for cR. Whenever a relational monoid satisfies one of these properties, it becomes a sound model for the basic calculus **B** enriched by the corresponding structural rule.

Given a repetition-free list l built by the alphabet $\{e, c, w\}$, and in the spirit of the extensions $\mathbf{B}_l$ for the logical calculus, the notation $\mathcal{M}_l$ identifies a relational monoid that

satisfies the corresponding properties of Tab.1.4. For instance, $\mathcal{M}_{ew}$ satisfies (e1), (e2), (w1) and (w2). Such a notation helps to express concisely a theorem of soundness for every extension, whose proof has been already exhibited by the previous reasoning, here and in §1.4.

Theorem 5 (Soundness on Structural Extensions). *A sequent derivable in* $\mathbf{B}_l$ *is valid in every relational monoid* $\mathcal{M}_l$.

On the other hand, the rules of Tab.1.4 give completeness at no additional cost. In fact, the canonical model in §1.5 clearly satisfies any of the properties as soon as the underlying logical calculus is enriched by the corresponding structural rule. This is easy to check by rewriting the semantical structural properties in terms of the canonical relation $\vdash_{\mathbf{B}_l}$, as they become the corresponding structural rule. Since the evaluation of formulae does not change, Lemma 4 of canonical evaluation still holds, then Theorem 3 of refined completeness can be specialised to any structural extension of the basic calculus, thus obtaining a refined completeness theorem for every structural extension of **B**.

Theorem 6 (Refined Completeness on Structural Extensions). *A sequent valid in every relational monoid* $\mathcal{M}_l$ *is derivable in* $\mathbf{B}_l$ *without using cut rules.*

In particular, the theorem says that the relational monoids satisfying all the properties of Tab. 1.4 are sound and (refined) complete models for Paraconsistent Quantum Logic [58, 58], mentioned at the beginning of this section.

Finally, the combination of the two previous theorems extends Theorem 4 by proving that every structural extension of the basic calculus enjoys the cut elimination property.

It is worth emphasising that the properties in Tab. 1.4 are 'essential' for soundness, in the sense that they are the *weakest* ones required to have soundness for any structural extension. They actually seem the rephrasing of structural rules in terms of monoidal relation R instead of the logical yielding $\vdash$, and this is due to the way of reasoning we used to obtain them. Indeed, they may be refined into more elegant and usual rules, and it may be possible to find equivalent or stronger properties that still enables soundness and (refined) completeness, but the properties to verify in the proofs will still be those in Tab. 1.4, as they exactly match the soundness requirements. Moreover, the fee to pay for a simplified model could be to miss the modularity in the proof of completeness, as the canonical model might change to fit the new semantical requirements.

A first intuitive simplification can be provided for the commutative calculus $\mathbf{B}_e$. Usually exchange rule is semantically expressed in the models by requiring a suitable algebraic operation to be commutative. Consider then the *commutative relational monoids*, namely those with a commutative monoidal operation. Clearly they satisfy properties c1 and c2, hence they are sound for the corresponding logical calculus. On the other hand, the canonical model for $\mathbf{B}_e$ as defined in §1.5 is not commutative, as lists have an intrinsic order that does not make the model operation, i.e., their merging, commute. Refined completeness can be recovered by slightly changing the set which the canonical model is built on. It is sufficient to consider $Frm^{\circledast}$: the set of all non-ordered lists, i.e., finite multisets, of formulae in $\mathcal{L}$.

In detail, the syntactic model enforcing completeness of the class of commutative relational monoids for $\mathbf{B}_e$ is the following refinement of Definition 3:

$$\mathcal{F}' \stackrel{\text{def}}{=} (Frm^{\circledast}, \circ, [\ ], \vdash_{\mathbf{B}_e}), \tag{1.30}$$

where $\vdash_{\mathbf{B}_e}$ is the cut free derivability in $\mathbf{B}_e$, i.e. for $\Gamma, \Delta \in Frm^{\circledast}$:

$$\Gamma \vdash_{\mathbf{B}_e} \Delta \quad \text{iff} \quad \Gamma \vdash \Delta \text{ is derivable in } \mathbf{B}_e \text{ withouts cuts.} \tag{1.31}$$

Relation $\vdash_{\mathbf{B}_e}$ between non-ordered lists is well defined, as $\mathbf{B}_e$ does not consider the position of formulae in the contexts, thanks to exchange rules. Therefore, $\mathcal{F}'$ is a commutative relational monoid.

All lemmas and corollaries proved for $\mathbf{B}$ and $\mathcal{F}$ in §1.5 are still verified for $\mathbf{B}_e$ and $\mathcal{F}'$. So a completeness theorem can be proved by following the proof of Theorem 3. Soundness and completeness for commutative models are summarised in the following theorem.

Theorem 7 (Commutative Relational Monoids). *A sequent deducible in $\mathbf{B}_e$ is valid in every commutative relational monoid* (Soundness). *Conversely, a sequent valid in every commutative relational monoid is derivable in $\mathbf{B}_e$ without using any cut rule* (Refined Completeness).

To see how to extend this commutative semantics to all the structural extensions of $\mathbf{B}$, it is sufficient to choose the right properties among those of Tab 1.4. In particular, commutative monoids satisfying properties ($c1$), ($c2$), ($w1$) and ($w2$) are sound and (refined) complete models for $\mathbf{B}_{ewc}$, hence for Paraconsistent Quantum Logic [20, 58, 59]. As a matter of fact, Theorem 7 is suitable to prove a cut elimination result for $\mathbf{B}_e$ and its structural extensions, simply by following the lines of Theorem 4.

1.7 Towards Intuitionistic Logics

As first discovered in [73] and then applied to Basic Logic in [129], what makes a sequent calculus 'intuitionistic' is the *liberalisation of contexts on the left*, that allows the *passive* contexts to appear (only) on the left hand side of the yielding operator in every rule. By passive contexts, we mean sequents of formulae not involved in the formula introduced by the rule. The presence of left passive contexts breaks the visibility principle. Incidentally, to liberalise contexts both on left and right produces a classical sequent calculus.

The intuitionistic calculi obtained from the basic calculus $\mathbf{B}$ are Intuitionistic Logic (**IL**) and Intuitionistic Linear Logic (**ILL**), the latter identifying the non-modal fragment of Intuitionistic Linear Logic, dubbed IMALL in [74]. Both calculi admit exchange rules, so they actually extend $\mathbf{B}_e$.

Left liberalisation of sequent is reached in $\mathbf{B}_e$ by relaxing visibility on the left-hand side of sequents. This can be done two ways: syntactical or more foundational. The former, developed in [129], adds a passive left context to every rule of the sequent calculus,

Table 1.5 Definitional Equations without Left Visibility

$(\otimes)$	$\Gamma, \psi \otimes \varphi \vdash \chi$ if and only if $\Gamma, \psi, \varphi \vdash \chi$
$(\oplus)$	$\Gamma, \psi \oplus \varphi \vdash \chi$ if and only if $\Gamma, \psi \vdash \chi$ and $\Gamma, \varphi \vdash \chi$
$(\&)$	$\Gamma \vdash \varphi \,\&\, \psi$ if and only if $\Gamma \vdash \varphi$ and $\Gamma \vdash \psi$
(1)	$\Gamma, 1 \vdash \chi$ if and only if $\Gamma \vdash \chi$
(0)	$\Gamma \vdash \chi$ and $\Gamma, 0 \vdash \chi$ if and only if $\Gamma \vdash \chi$
$(\top)$	$\Gamma \vdash \psi$ and $\Gamma \vdash \top$ if and only if $\Gamma \vdash \psi$
$(\rightarrow)$	$\Gamma \vdash \varphi \rightarrow \psi$ if and only if $\Gamma, \varphi \vdash \psi$

thus making it intuitionistic. The latter is entirely new and plays on the reflection principle by introducing passive contexts directly in the definitional equations, that are solved as in §1.2 to obtain the corresponding sequent calculus. This is the way chosen in this section to introduce the intuitionistic sequent calculi. In such a way, the two calculi are not only a syntactical, but also a meta-linguistical extension of the basic calculus **B**, as they are directly developed on the extensions of the definitional equations, and not on the already existing sequent calculus.

The difference between **IL** and **ILL** is that the former is a proper generalisation of the latter one, as it allows weakening and contraction rules, see Fig. 1.2. Therefore, the foundational introduction presented in this section can deal with **ILL**, then **IL** will be recovered by adding structural rules to the obtained sequent calculus.

The intrinsic nature of **ILL**, as well as **IL**, is asymmetric since visibility is broken on the left hand side of the sequent and every rule presents an arbitrary context on the left. So it seems natural to consider only sequents of the form $\Gamma \vdash \varphi$, where Γ is a (possibly empty) list of formulae and φ is a single formula. The language $\mathcal{L}$ for **ILL** is the same as for **B**, but without $\parr$ and $\bot$, as it is pointless to introduce them, due to the asymmetric structure of sequents. In fact, they reflect comma and empty context on the right hand side of the sequent, and they have no meaning in case of a single formula on the right.

The definitional equations for **ILL** are then obtained from Tab. 1.1 by adding passive contexts on the left of sequents and by considering single formulae on their right. The equations are fully outlined in Tab. 1.5. In particular, the definitional equation for $\oplus$ does not change, as the introduction for such a connective is on the right. Moreover, the last definitional equation introduces a new connective: the *implication*, $\rightarrow$. Implication is the main feature of an intuitionistic calculus, and it naturally finds its 'identity' whenever contexts are liberalised on the left, as it will be fully described further in this section.

Definitional equations are solved as in §1.2 by using axioms, exchange and cut rules. The structure of the axioms does not change, as they reflect on the logical level the fact that every assertion yields itself. The exchange rule expresses that the order among assumptions is irrelevant. As there can only be a single formula on the right, the only meaningful exchange and cut rules are *eL* and *cutL*. The complete calculus is outlined in Fig. 1.3, and in the following we show how to solve a few definitional equations: those

for $\&$, whose 'left' rule will appear with a passive context, even though the definitional equation does not change; those for 1, whose equation is solved in a shorter way; and those for $\to$, the new connective.

Consider the definitional equation for $\&$, the aim is to find the corresponding rules $\& L$ and $\& R$. The pattern to follow is again the one described in §1.2. The two directions of definitional the equation produces directly the rules

$$\frac{\Gamma \vdash \varphi \,\&\, \psi}{\Gamma \vdash \varphi} \quad \frac{\Gamma \vdash \varphi \,\&\, \psi}{\Gamma \vdash \psi} \;\textit{implicit } \&\textit{-reflection} \qquad \frac{\Gamma \vdash \varphi \quad \Gamma \vdash \psi}{\Gamma \vdash \varphi \,\&\, \psi} \;\&\textit{-formation}$$

The latter one corresponds to $\& R$. The former one needs to be set up to obtain $\& L$. The first step is again to trivialise the premisses by considering the axiom $\varphi \,\&\, \psi \vdash \varphi \,\&\, \psi$, thus obtaining the equivalent

$$\varphi \,\&\, \psi \vdash \varphi \qquad \varphi \,\&\, \psi \vdash \psi \qquad \textit{axioms of } \otimes\textit{-reflection.}$$

In this case, implicit $\&$-reflection is recovered by one application of cut rule

$$\frac{\Gamma \vdash \varphi \,\&\, \psi \quad \varphi \,\&\, \psi \vdash \varphi}{\Gamma \vdash \varphi} \qquad \frac{\Gamma \vdash \varphi \,\&\, \psi \quad \varphi \,\&\, \psi \vdash \psi}{\Gamma \vdash \psi}$$

The final solution is reached by assuming $\Gamma, \varphi \vdash \chi$ and $\Gamma, \psi \vdash \chi$, and by applying two cuts:

$$\frac{\varphi \,\&\, \psi \vdash \varphi \quad \Gamma, \varphi \vdash \chi}{\Gamma, \varphi \,\&\, \psi \vdash \chi} \qquad \frac{\varphi \,\&\, \psi \vdash \psi \quad \Gamma, \psi \vdash \chi}{\Gamma, \varphi \,\&\, \psi \vdash \chi}$$

Thus obtaining the two $\& R$ rules. Again, axioms of $\&$-reflection is obtained by trivialising the premisses with two axioms involving φ and ψ.

To solve the definitional equation for constant 1 is even quicker. The backward direction of definition gives $1R$ rule directly. On the other hand, the forward one gives the *implicit* 1*-reflection*

$$\frac{\Gamma, 1 \vdash \chi}{\Gamma \vdash \chi}$$

Then by trivialising the premisses with the axiom $1 \vdash 1$ it gives the axiom of 1-reflection, $\vdash 1$, that is the correct $1L$ rule to chose.

A peculiar connective to linear and intuitionistic calculi is implication. It reflects the link *yields*, the sign $\vdash$ itself, by moving formulae from the left hand side of the sequent to the right one. Intuitively, we say that whenever $\varphi \to \psi$ is asserted, ψ can be asserted in turn simply by adding φ to the current assumptions. This idea is formalised by the corresponding equation

$$\Gamma \vdash \varphi \to \psi \quad \text{ if and only if } \quad \Gamma, \varphi \vdash \psi. \tag{1.32}$$

Such a definitional equation clearly gains meaning when left passive contexts are allowed, due to the presence of Γ on the left hand side. To introduce an form of implication in the

basic calculus **B** involves a deeper understanding of the meta-linguistic link *yields*, as hinted in [129].

The equation, solved as in §1.2, introduces directly the rules

$$\frac{\Gamma \vdash \varphi \to \psi}{\Gamma, \varphi \vdash \psi} \ \textit{implicit} \ \to \textit{-reflection} \qquad \frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \to \psi} \ \to \textit{-formation}$$

The latter one corresponds to $\to R$. The former one will provide $\to L$. By trivialising the premisses with $\varphi \to \psi \vdash \varphi \to \psi$, the implicit $\to$-formation becomes

$$\varphi \to \psi, \varphi \vdash \psi \quad \textit{axioms of} \ \to \textit{-reflection.}$$

And, again, implicit $\to$-reflection is recovered by one application of cut rule

$$\frac{\Gamma \vdash \varphi \to \psi \quad \varphi \to \psi, \varphi \vdash \psi}{\Gamma, \varphi \vdash \psi}$$

The final solution to the definitional equation, $\to L$, is reached by assuming $\Gamma_1 \vdash \varphi$ and $\Gamma_2, \psi \vdash \chi$ and by applying two cuts:

$$\frac{\dfrac{\Gamma_1 \vdash \varphi \quad \varphi \to \psi, \varphi \vdash \psi}{\Gamma_1, \varphi \to \psi \vdash \psi} \qquad \Gamma_2, \psi \vdash \chi}{\Gamma_1, \Gamma_2, \varphi \to \psi \vdash \chi}$$

Axiom of $\to$-reflection is obtained by trivialising the premisses with two axioms involving φ and ψ.

The next proposition points out another approach to extend **B** to intuitionistic calculi. It witnesses the power of the implication and its corresponding definitional equation. In fact, the proposition says that the equations in Tab. 1.1, without visibility on the left, are obtained from those of Fig. 1.1, that satisfies visibility, simply by adding the implication along with its definitional equation. In other words, this means that the full power of the calculus **ILL** can be obtained simply by adding the connective $\to$ to the original basic calculus $\mathbf{B}_e$ without requiring any liberalisation on the contexts involved in the original definitional equations. This fact is central in extending the relational semantics into intuitionistic settings, as it says that, in order to find a sound semantics for **ILL**, it is sufficient to consider the relational monoids with a native notion of implication. Hence, when extending the semantics we will focus only on the last equation of Tab. 1.5, and we will study how to validate it.

Proposition 5. *The definitional equations for the connectives* 1, 0, $\top$, $\otimes$, $\oplus$, $\&$ *of* $\mathbf{B}_e$, *restricted to single formulae on the right hand side of sequents and enriched by the connective* $\to$ *along with the corresponding definitional equation, are equivalent to those defining* **ILL**.

Proof. As definitional equations defining **ILL** extend those defining $\mathbf{B}_e$, the only thing to check is whether the system composed by definitional equations for $\mathbf{B}_e$ and the equation

Figure 1.3 Sequent Calculus **ILL**

Axioms

$$\varphi \vdash \varphi$$

Multiplicatives

$$\frac{\Gamma, \psi, \varphi \vdash \chi}{\Gamma, \psi \otimes \varphi \vdash \chi} \otimes L \qquad \frac{\Gamma_2 \vdash \varphi \quad \Gamma_1 \vdash \psi}{\Gamma_2, \Gamma_1 \vdash \varphi \otimes \psi} \otimes R$$

$$\frac{\Gamma \vdash \chi}{\Gamma, 1 \vdash \chi}\ 1L \qquad \vdash 1\ \ 1R$$

Additives

$$\frac{\Gamma, \psi \vdash \chi}{\Gamma, \psi \,\&\, \varphi \vdash \chi} \,\&\, L \quad \frac{\Gamma, \varphi \vdash \chi}{\Gamma, \psi \,\&\, \varphi \vdash \chi} \,\&\, L \qquad \frac{\Gamma \vdash \varphi \quad \Gamma \vdash \psi}{\Gamma \vdash \varphi \,\&\, \psi} \,\&\, R$$

$$\frac{\Gamma, \psi \vdash \Delta \quad \Gamma, \varphi \vdash \chi}{\Gamma, \psi \oplus \varphi \vdash \chi} \oplus L \qquad \frac{\Gamma \vdash \varphi}{\Gamma \vdash \varphi \oplus \psi} \oplus R \quad \frac{\Gamma \vdash \psi}{\Gamma \vdash \varphi \oplus \psi} \oplus R$$

$$\Gamma, 0 \vdash \chi\ \ 0L \qquad \Gamma \vdash \top\ \ \top R$$

Implication

$$\frac{\Gamma_1 \vdash \varphi \quad \Gamma_2, \psi \vdash \chi}{\Gamma_1, \Gamma_2, \varphi \to \psi \vdash \chi} \to L \qquad \frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \to \psi} \to R$$

Exchange

$$\frac{\Gamma_1, \varphi, \psi, \Gamma_2 \vdash \chi}{\Gamma_1, \psi, \varphi, \Gamma_2 \vdash \chi}\ exch$$

Cut

$$\frac{\Gamma_1 \vdash \varphi \quad \Gamma_2, \varphi \vdash \chi}{\Gamma_1, \Gamma_2 \vdash \chi}\ cut$$

for $\to$ satisfies equations in Tab. 1.5. The proof checks connective by connective. Clearly the equations for $\&$, $\top$ and $\to$ are satisfied, as they are common to the two systems. All the reasonings for the other connectives, $\otimes$ and $\oplus$, and constants, 1 and 0, follow a common path: whenever assuming one of the equations in Tab.1.1, then the connective $\to$, mainly its property to move formulae from one side of the sequent to the other, is fundamental in showing that the corresponding equation in Tab. 1.5 is verified as well. As a guideline, consider the connective $\otimes$. By assuming its basic definitional equation

$$\psi \otimes \varphi \vdash \chi \text{ if and only if } \psi, \varphi \vdash \chi \tag{1.33}$$

prove that

$$\Gamma, \psi \otimes \varphi \vdash \chi \text{ if and only if } \Gamma, \psi, \varphi \vdash \chi. \tag{1.34}$$

for every context $\Gamma = \gamma_1, \ldots, \gamma_n$. Thanks to the definitional equation for $\to$ and *exch* rule, the sequent $\gamma_1, \ldots, \gamma_n, \psi \otimes \varphi \vdash \chi$ is equivalent to $\gamma_2, \ldots, \gamma_n, \psi \otimes \varphi \vdash \gamma_1 \to \chi$, that is in turn equivalent to $\psi \otimes \varphi \vdash \gamma_n \to (\ldots \gamma_2 \to (\gamma_1 \to \chi))$, by iterating the process. The last one is a context suitable for definition (1.33), hence it is equivalent to $\psi, \varphi \vdash \gamma_n \to (\ldots \gamma_2 \to$

$(\gamma_1 \to \chi)$), and, again by applying the definitional equation of $\to$ for n times, it turns to be equivalent to $\Gamma, \psi, \varphi \vdash \chi$. As a matter of fact, the last one is a well known result in categorical logic, see *closed* categories for instance. Finally conclude that definitional equation (1.34) holds. □

As a matter of fact, a similar result is easily proved for **IL**, simply by considering structural rules. A final important property is that the definitional equations are still verified in the calculus **IL** deprived of cuts, as will be proved by the following proposition. The proof is an extension of the one given for the basic calculus in Proposition 1, and it follows a common pattern for every logical entity: the backward implications of definitional equations are verified thanks to the introduction rules, the forward ones are verified by an induction on the length of the cut-free derivations. The induction will take care of structural rules, contraction in particular, by proving an equivalent refinement of the considered definitional equation. The connectives $\otimes$, $\oplus$ and $\to$ are discussed in detail, as $\otimes$ and $\to$ will be a key point for the completeness of the extended semantics, and $\oplus$ needs particular care, due to contraction rule. Here we prove the result for **IL**, and it can be clearly extended to its subcalculus **ILL**.

Proposition 6 (Cut-free Equations for IL). *The definitional equations without visibility on the left, and enriched by the equation for implication are satisfied by the cut-free calculus derived from* **IL**.

Proof. As anticipated, the proof has a general pattern for every logical entity, so consider the connectives $\otimes$, $\oplus$ and $\to$ as guideline.

Case $\otimes$. The corresponding equation to prove in the cut-free calculus is

$$\begin{array}{c}\Gamma, \psi \otimes \varphi \vdash \chi \text{ is derivable} \\ \text{without } \textit{cut} \text{ rules}\end{array} \quad \text{if and only if} \quad \begin{array}{c}\Gamma, \psi, \varphi \vdash \chi \text{ is derivable} \\ \text{without } \textit{cut} \text{ rules.}\end{array} \tag{1.35}$$

The backward direction is a simple application of $\otimes L$ rule. For the forward direction, consider $n \geq 1$ and prove the following equivalent statement:

$$\begin{array}{l}\text{if } \Gamma, \psi_1 \otimes \varphi_1, \ldots, \psi_n \otimes \varphi_n \vdash \chi \text{ is derivable without } \textit{cut} \text{ rules} \\ \qquad\qquad \text{then } \Gamma, \psi_1, \varphi_1, \ldots, \psi_n, \varphi_n \vdash \chi \text{ is derivable without } \textit{cut} \text{ rules.}\end{array}$$

Assume $\Gamma, \psi_1 \otimes \varphi_1, \ldots, \psi_n \otimes \varphi_n \vdash \chi$ derivable without cuts, then proceed by induction on the length of derivation. The base of induction is any rule without premisses, which can only be either an axiom, or $\top R$, or $0L$. On the one hand, if it is an axiom, then χ is $\psi \otimes \varphi$, Γ is empty, n is 1, and $\psi, \varphi \vdash \psi \otimes \varphi$ is derived without cuts as for Proposition 1. On the other hand, if the applied rule is either $\top R$ or $0L$, then the same rule can be applied to a passive context with ψ_i, φ_i instead of $\psi_i \otimes \varphi_i$ (for $i = 1 \ldots n$).

In the induction step consider the last applied rule in the derivation. If every $\psi_i \otimes \varphi_i$ (for $i = 1 \ldots n$) is in the passive context of the rule (either left or right rule), then use induction hypothesis on its premisses, and obtain the claim by applying the rule itself. The rules that can involve any $\psi_i \otimes \varphi_i$ (for $i = 1 \ldots n$) are $\otimes L$ and structural rules. If the

rule is $\otimes L$ then it introduces exactly one $\psi_i \otimes \varphi_i$, and it is sufficient to apply induction hypothesis on its premisses. If the last applied rule is exchange, then use induction on the premisses and apply it again. On the other hand, if the last applied rule is weakening, then, without loss of generality, assume that the involved formula is $\psi_1 \otimes \varphi_1$, hence the premiss is

$$\Gamma, \psi_1 \otimes \varphi_1, \psi_1 \otimes \varphi_1, \ldots, \psi_n \otimes \varphi_n \vdash \chi,$$

then conclude the thesis by induction hypothesis and by applying two contraction rules. Finally, if the last applied rule is weakening, then assume, without loss of generality, that the involved formula is $\psi_1 \otimes \varphi_1$, hence the premiss is

$$\Gamma, \psi_2 \otimes \varphi_2, \ldots, \psi_n \otimes \varphi_n \vdash \chi,$$

then apply induction hypothesis and two weakening rules, with ψ and φ respectively.

Case $\oplus$. The equation to prove is

$$\begin{array}{ccc} \Gamma, \psi \oplus \varphi \vdash \chi \text{ is derivable} \\ \text{without } \textit{cut} \text{ rules} \end{array} \quad \text{if and only if} \quad \begin{array}{c} \Gamma, \psi \vdash \chi \text{ and } \Gamma, \varphi \vdash \chi \text{ are} \\ \text{derivable without } \textit{cut} \text{ rules.} \end{array}$$

Also in this case, the backward direction is a simple application of $\oplus L$ rule. For the forward direction, prove the following equivalent statement:

$$\begin{array}{l} \text{if } \Gamma, \psi \oplus \varphi, \ldots, \psi \oplus \varphi \vdash \chi \text{ is derivable without } \textit{cut} \text{ rules} \\ \qquad\qquad \text{then } \Gamma, \psi \vdash \chi \text{ and } \Gamma, \varphi \vdash \chi \text{ are derivable without } \textit{cut} \text{ rules.} \end{array}$$

Again, assume that $\Gamma, \psi \oplus \varphi, \ldots, \psi \oplus \varphi \vdash \chi$ is derivable without cuts, then proceed by induction on the length of derivation. The proof mimics the one of $\otimes$. For the basic step, the only interesting case is the axiom $\psi \oplus \varphi \vdash \psi \oplus \varphi$, which is solved by $\oplus R$:

$$\frac{\psi \vdash \psi}{\psi \vdash \psi \oplus \varphi} \oplus R \qquad \frac{\varphi \vdash \varphi}{\varphi \vdash \psi \oplus \varphi} \oplus R$$

The induction step considers the last applied rule in the derivation. If every $\psi \oplus \varphi$ is in the passive context of the rule (either left or right rule), then the induction hypothesis are applied to its premisses and the claim is obtained by applying the rule itself. The rules involving $\psi \oplus \varphi$ (for $i = 1 \ldots n$) can be $\oplus L$ and the structural rules. If the rule is $\oplus L$ then it introduces exactly one $\psi_i \otimes \varphi_i$, and its premisses are $\Gamma, \psi \oplus \varphi, \ldots, \psi \vdash \chi$ and $\Gamma, \psi \oplus \varphi, \ldots, \varphi \vdash \chi$. Now, by induction hypothesis $\Gamma, \psi, \psi \vdash \chi$ and $\Gamma, \varphi, \psi \vdash \chi$ for the former one, and $\Gamma, \psi, \varphi \vdash \chi$ and $\Gamma, \varphi, \varphi \vdash \chi$ for the latter one. The claim is obtained by applying two contraction rules. If the last applied rule is exchange, then use induction on the premisses and apply it again. On the other hand, if the last applied rule is a contraction on $\psi \oplus \varphi$, then use induction hypothesis on its premiss and apply two contraction rules, on ψ and φ respectively. Finally, if the last applied rule is weakening, then use induction hypothesis on its premiss and apply weakening with ψ and φ if necessary.

Case $\rightarrow$. In this case, the equation to prove in the cut-free calculus is

$$\begin{array}{c}\Gamma, \varphi \vdash \psi \text{ is derivable} \\ \text{without } \textit{cut} \text{ rules}\end{array} \quad \text{if and only if} \quad \begin{array}{c}\Gamma \vdash \varphi \rightarrow \psi \text{ is derivable} \\ \text{without } \textit{cut} \text{ rules.}\end{array}$$

Again, the backward direction is a simple application of $\otimes L$ rule. The forward direction is proved by induction on the length of the derivation. The proof follows the same lines as for (1.35). The only interesting case for the base step is the axiom $\varphi \rightarrow \psi \vdash \varphi \rightarrow \psi$, which is solved by using $\rightarrow L$:

$$\frac{\varphi \vdash \varphi \quad \psi \vdash \psi}{\varphi \rightarrow \psi, \varphi \vdash \psi} \rightarrow L$$

On the other hand, in the induction step consider the last applied rule. Any rule on the left cannot involve the formula $\varphi \rightarrow \psi$, hence use induction on its premisses and finish by applying the rule itself. The only left rule that can be applied is $\rightarrow L$, due to the visibility of the right. In this case its premiss is what is needed. □

As anticipated, this result still holds even in a calculus without structural rules, hence it naturally extends up to **ILL**, as can be easily checked. In particular, to prove the above proposition directly on **ILL** it is not necessary to consider some equivalent property as we did for **IL**.

A property that will be useful to prove completeness for **ILL** is a direct consequence of Proposition 6. For every context Σ, define

$$\Sigma^{\otimes} \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } \Sigma = [\,]; \\ \sigma_1 \otimes \ldots \otimes \sigma_n & \text{if } \Sigma = \sigma_1, \ldots, \sigma_n \text{ with } n \geq 1. \end{cases} \tag{1.36}$$

Then it is easy to check that in **IL**, as well as in **ILL**:

$$\begin{array}{c}\Gamma, \Sigma \vdash \chi \text{ is derivable} \\ \text{without } \textit{cut} \text{ rules}\end{array} \quad \text{if and only if} \quad \begin{array}{c}\Gamma \vdash \Sigma^{\otimes} \rightarrow \chi \text{ is derivable} \\ \text{without } \textit{cut} \text{ rules.}\end{array} \tag{1.37}$$

Note that the order for the composition in $\Sigma^{\otimes}$ irrelevant, as the passive left context makes the connective $\otimes$ associative.

1.7.1 Relational Semantics for Intuititionistic Linear Logic

As already observed, Proposition 5 hints how to extend the relational semantics to **ILL**. As the basic calculus $\mathbf{B}_e$ becomes equivalent to **ILL** simply by adding the connective $\rightarrow$ along with the corresponding definitional equation, the semantics can in turn be extended to **ILL** by defining the right evaluation just for this new connective, thus extending the function V, given in Tab. 1.2. Hence our goal is to fix a commutative relational monoid, and to find a binary operation on subsets that reflects the logical connective $\rightarrow$ into the

semantics. The operation will follow the spirit of [127], and it will turn out as the *implication* among subsets, that we denote as $A \to B$. To define such an operator, the contribution of definitional equations is, once again, essential.

By following the lines of §1.3, the evaluation $V(\varphi \to \psi)$ of the formula $\varphi \to \psi$ must meet two requirements: (1) to be left saturated, and (2) to satisfy the semantics counter part of the definitional equation for the implication. As for the second point, the property to satisfy is

$$Ingr(\Gamma) \subseteq Prod(\varphi \to \psi)^{\leftarrow} \quad \text{if and only if} \quad Ingr(\Gamma, \varphi) \subseteq Prod(\psi)^{\leftarrow}$$

for every choice of $\Gamma = \gamma_1, \ldots, \gamma_n$. Defining C to be the product $V(\gamma_1) \cdot \ldots \cdot V(\gamma_n)$, the definition says that the previous requirement means that for every $C \subseteq M$:

$$C \subseteq V(\varphi \to \psi) \quad \text{if and only if} \quad (C \cdot V(\varphi)) \subseteq V(\psi).$$

Then, assuming $V(\varphi) = A$, $V(\psi) = B$ and $V(\varphi \to \psi) = A \to B$, the property that must be satisfied by new binary operation $A \to B$ between subsets is

$$C \subseteq A \to B \quad \text{if and only if} \quad (C \cdot A) \subseteq B, \tag{1.38}$$

for every choice of $A, B, C \subseteq M$ with A and B left saturated. In particular, this property says that $\to$ between left saturated subsets is a 'good implication,' in the sense that it is adjoint to the monoidal operation, similarly to the logical implication, adjoint to $\otimes$.

To extend the property in (1.38) to general subsets, we need to consider the saturation operators. As we will see, the fact that A is left saturated is irrelevant, hence (1.38) can be generalised to

$$C \subseteq A \to B \quad \text{if and only if} \quad (C \cdot A) \subseteq B^{\to\leftarrow} \tag{1.39}$$

for every $A, B, C \subseteq M$. Now, by assuming C to be the singleton $\{x\}$, the definition of $A \to B$ becomes an unescapable choice, since the equation (1.39) becomes

$$x \,\epsilon\, A \to B \quad \text{if and only if} \quad x \cdot A \subseteq B^{\to\leftarrow},$$

that forces to define

$$A \to B \stackrel{\text{def}}{=} \{x \,:\, x \cdot A \subseteq B^{\rightarrow\,\leftarrow}\}. \tag{1.40}$$

It is easy to see that this definition grants the property

$$(A \to B) \cdot A \subseteq B^{\to\leftarrow}. \tag{1.41}$$

In particular, thanks to this property, the equation in (1.39), and (1.38) as well, is directly verified without requiring any additional property to the monoidal model. Therefore, if we define $V(\varphi \to \psi)$ to be $V(\varphi) \to V(\psi)$, then a *sufficient* condition that makes the relational monoids sound for **ILL** is that the implication is left saturated for every pair of subsets, as required by the definition of the evaluation function. One thing that is worth to investigate is whether such a requirement is also the *minimum* that enables soundness.

As pointed out in §1.7.2, the introduction of implication in the logic breaks visibility on the left hand side of sequents. This means that the logic can isolate a single formula from the rest of the context in order to work only on it. In fact, in the sequent $\gamma_1, \ldots, \gamma_n, \varphi \vdash \chi$ all the formulae in the antecedent are connected, and to work on one of them, say φ for instance, the formal system must detect φ among all the others, to 'detach' φ from the whole list, and then to work on φ by introducing the logical constructs. From the semantic point of view, a context evaluation is the saturation of the product among all the interpretations of the formulae that constitute the context itself. In particular, if $V(\gamma_i) = A_i$, $V(\varphi) = A$ and $V(\chi) = C$, then the evaluation of the previous context is $A_1 \cdot \ldots \cdot A_n \cdot B \subseteq C^{\rightarrow\leftarrow}$, that means

$$(A_1 \cdot \ldots \cdot A_n \cdot B)^{\rightarrow\leftarrow} \subseteq C^{\rightarrow\leftarrow}.$$

To distinguish φ, alias B, the rest of the product must be distinguished as a saturated subset in order to be independent from B. Hence, the evaluation must at least satisfy the requirement

$$(A_1 \cdot \ldots \cdot A_n \cdot B)^{\rightarrow\leftarrow} \subseteq C^{\rightarrow\leftarrow} \text{ if and only if } (A_1 \cdot \ldots \cdot A_n)^{\rightarrow\leftarrow} \cdot B^{\rightarrow\leftarrow} \subseteq C^{\rightarrow\leftarrow}$$

that is equivalent to ask for

$$(A \cdot B)^{\rightarrow\leftarrow} \subseteq C^{\rightarrow\leftarrow} \text{ if and only if } A^{\rightarrow\leftarrow} \cdot B^{\rightarrow\leftarrow} \subseteq C^{\rightarrow\leftarrow}$$

for every $A, B, C \subseteq M$. The properties of the saturation operators in Lemma 1 say that $(A \cdot B)^{\rightarrow\leftarrow} \subseteq C^{\rightarrow\leftarrow}$ if and only if $A \cdot B \subseteq C^{\rightarrow\leftarrow}$, then the previous requirement is equivalent to

$$A \cdot B \subseteq C^{\rightarrow\leftarrow} \text{ if and only if } A^{\rightarrow\leftarrow} \cdot B^{\rightarrow\leftarrow} \subseteq C^{\rightarrow\leftarrow}$$

for every $A, B, C \subseteq M$. And, again because $(\,)^{\rightarrow\leftarrow}$ is a closure operator and the operation is commutative, it is easy to see that the previous is equivalent to

$$A \cdot B \subseteq C^{\rightarrow\leftarrow} \text{ if and only if } A \cdot B^{\rightarrow\leftarrow} \subseteq C^{\rightarrow\leftarrow} \tag{1.42}$$

for every $A, B, C \subseteq M$. In particular property (1.42) says that B must be intended as a saturated subset and then it is possible to work on it by following the semantical definitions. The requirement in (1.42) is an essential requirement for a relational monoid to be a sound model for **ILL**.

Another approach to extend the relational semantics to **ILL** is to consider the definitional equations without left visibility, as they are written in Tab. 1.5. The connective that is mostly influenced by the presence of a passive 'left' context is $\otimes$, as it reflects the comma between formulae. In this case, the definitional equation for $\otimes$ is semantically rephrased as

$$Ingr(\Gamma, \psi \otimes \varphi) \subseteq Prod(\chi)^{\leftarrow} \text{ if and only if } Ingr(\Gamma, \psi, \varphi) \subseteq Prod(\chi)^{\leftarrow}$$

for every context Γ and formulae ψ, φ, χ. As $Prod(\Delta)^{\leftarrow}$, $Ingr(\Gamma, \psi \otimes \varphi)$ and $Ingr(\Gamma, \psi, \varphi)$ are generic left saturated subsets, this requirement is equivalent to ask that

$$Ingr(\Gamma, \psi \otimes \varphi) = Ingr(\Gamma, \psi, \varphi)$$

for every Γ, ψ and φ. Assuming that $\Gamma = \varphi_1, \ldots, \varphi_n$, this means that

$$(V(\varphi_1) \cdot \ldots \cdot V(\varphi_n) \cdot V(\psi \otimes \varphi))^{\rightarrow\leftarrow} = (V(\varphi_1) \cdot \ldots \cdot V(\varphi_n) \cdot V(\psi) \cdot V(\varphi))^{\rightarrow\leftarrow}$$

for any choice of the involved formulae. If $A = V(\varphi_1) \cdot \ldots \cdot V(\varphi_n)$, $V(\psi) = A_1$ and $V(\varphi) = A_2$, the definition of $\otimes$ evaluation says that the right requirement is

$$(A \cdot (A_1 \cdot A_2)^{\rightarrow\leftarrow})^{\rightarrow\leftarrow} = (A \cdot A_1 \cdot A_2)^{\rightarrow\leftarrow}.$$

Hence, to semantically verify the definitional equation for $\otimes$, the right property to ask for is:

$$(A \cdot B^{\rightarrow\leftarrow})^{\rightarrow\leftarrow} = (A \cdot B)^{\rightarrow\leftarrow}$$

for every $A, B \subseteq M$. As the inclusion from right to left always holds in every relational monoid, this property is equivalent to $(A \cdot B^{\rightarrow\leftarrow})^{\rightarrow\leftarrow} \subseteq (A \cdot B)^{\rightarrow\leftarrow}$. Moreover, thanks to the properties of closure operators, we conclude that the semantical property generated from the definitional equation of $\otimes$ without visibility on the left, is

$$A \cdot B^{\rightarrow\leftarrow} \subseteq (A \cdot B)^{\rightarrow\leftarrow} \tag{1.43}$$

for every $A, B \subseteq M$. It corresponds to the topological requirement in [127], dubbed *stability*. We conclude that stability is an *essential* requirement for a relational monoid to be a sound model for **ILL**, as it validates the definitional equation for $\otimes$.

And now the wheel has come: in (1.43) we found a necessary requirement for soundness that turns to be equivalent to ask for a saturated implication between subsets, as stated by next proposition. The proposition, in fact, shows that the properties in (1.42) and (1.43) are equivalent to say that the implication between subsets is left saturated. Hence the fact that $A \to B$ is saturated for every couple of subsets A, B is a *necessary and sufficient condition* to extend the relational semantics to **ILL**.

Proposition 7. *In every commutative relational monoid $\mathcal{M} = (M, \cdot, 1, R)$, the following properties are equivalent:*

1. *For every $A, B \subseteq M$, the implication $A \to B$ is left saturated.*
2. *$A \cdot B \subseteq C^{\rightarrow\leftarrow}$ if and only if $A \cdot B^{\rightarrow\leftarrow} \subseteq C^{\rightarrow\leftarrow}$, for every $A, B, C \subseteq M$.*
3. *$A \cdot B^{\rightarrow\leftarrow} \subseteq (A \cdot B)^{\rightarrow\leftarrow}$, for every $A, B, C \subseteq M$.*

Proof. Assume that point 1 is verified, then prove point 2 The backward implication of point 2 is verified in every relational monoid, as $B \subseteq B^{\rightarrow\leftarrow}$. To prove the forward implication, assume that $A \cdot B \subseteq C^{\rightarrow\leftarrow}$. As the implication satisfies property (1.38), this

assumption implies that $B \subseteq A \to C$, then $B^{\to\leftarrow} \subseteq A \to C$ since the implication between subsets is left saturated. Again by property (1.38), conclude $A \cdot B^{\to\leftarrow} \subseteq C^{\to\leftarrow}$.

Point 3 is verified, as soon as point 2 is assumed. In fact, the closure properties say that $A \cdot B \subseteq (A \cdot B)^{\to\leftarrow}$, hence $A \cdot B^{\to\leftarrow} \subseteq (A \cdot B)^{\to\leftarrow}$ by point 2.

Finally, assume that point 3 is verified, and prove point 1. Point 3 applied to A and $A \to B$ says that $A \cdot (A \to B)^{\to\leftarrow} \subseteq (A \cdot (A \to B))^{\to\leftarrow}$. Thanks to (1.41) this means $A \cdot (A \to B)^{\to\leftarrow} \subseteq B^{\to\leftarrow}$, hence $(A \to B)^{\to\leftarrow} \subseteq A \to B$ and conclude that $A \to B$ is left saturated. □

Proposition 7, provided in terms of the operator $(\,)^{\to\leftarrow}$, is verified more generally for any closure operator. A similar result appears also in [112], but it is not emphasised there. Indeed, such a result is fundamental here, as it actually represents the keystone to control the contexts in the semantics. The proposition, in fact, states the semantic counter part on liberalising the contexts on the left hand side of sequents.

Moreover Proposition 7 shows the properties that are equivalently needed to have soundness for **ILL**. Now, as for §1.6, what a clear mathematical definition needs is an 'elemental' condition on the monoidal constituents: we need to find a first order property that is equivalent to those listed in Proposition 7, as they are expressed on subsets, and hence they are second order properties.

The equivalent first order condition can be found by defining a particular class of subsets, parameterised by pairs of elements in M. For every $x, y \in M$, define:

$$Gap(x,y) \stackrel{\text{def}}{=} \{z \,:\, z \cdot xRy\}. \tag{1.44}$$

The name is due to its particular interpretation in the production cycle. Consider $z \,\epsilon\, Gap(x,y)$, then the definition says that y is produced by composing z with x. In other words z is an ingredient sufficient for x in order to produce y. We can say that z is the 'gap' that x needs to fill in order to produce y.

The sought first order condition for soundness is to require that every subset $Gap(x,y)$ is left saturated. In fact this is a condition that can be written at first order and that corresponds to the properties in Proposition 7, as stated by next proposition. First note two properties that derive straight from the definition in (1.44). For every $x, y \in M$ it holds:

$$z \,\epsilon\, G(x,y) \quad \text{if and only if} \quad z \cdot x \,\epsilon\, y^{\leftarrow},$$

hence, for every $C \subseteq M$

$$C \subseteq G(x,y) \quad \text{if and only if} \quad C \cdot x \subseteq y^{\leftarrow}. \tag{1.45}$$

Proposition 8. *In every commutative relational monoid $\mathcal{M} = (M, \cdot, 1, R)$, the following properties are equivalent:*

1. *$Gap(x,y)$ is left saturated for every $x, y \in M$.*
2. *$A \cdot B \subseteq C^{\to\leftarrow}$ if and only if $A \cdot B^{\to\leftarrow} \subseteq C^{\to\leftarrow}$, for every $A, B, C \subseteq M$.*

Proof. First assume point 1 and prove point 2. Since the backward direction of the implication in point 2. is always verified, it is sufficient to prove the forward one. Since $(\,)^{\rightarrow\leftarrow}$ is a closure operator it holds $A\cdot B \subseteq (A\cdot B)^{\rightarrow\leftarrow}$. As $(A\cdot B)^{\rightarrow\leftarrow}$ is the intersection among $y^{\leftarrow}$ for every $y \in (A\cdot B)^{\rightarrow}$, this means that for every $x\,\epsilon\, A$ and $y\,\epsilon (A\cdot B)^{\rightarrow}$ it holds $B\cdot x \subseteq y^{\leftarrow}$, that is $B \subseteq \mathit{Gap}(x,y)$ by (1.45). This implies that $B^{\rightarrow\leftarrow} \subseteq \mathit{Gap}(x,y)$, since the latter subset is left saturated by hypothesis. Hence, again by (1.45), this implies $B^{\rightarrow\leftarrow}\cdot x \subseteq y^{\leftarrow}$ for every $x\,\epsilon\, A$ and $y\,\epsilon (A\cdot B)^{\rightarrow}$. Conclude $A\cdot B^{\rightarrow\leftarrow} \subseteq (A\cdot B)^{\rightarrow\leftarrow}$.

Conversely, assume point 2 and prove point 1. By the definition in (1.44), $G(x,y)\cdot x \subseteq y^{\leftarrow}$, hence $G(x,y)^{\rightarrow\leftarrow}\cdot x \subseteq y^{\leftarrow}$ by point 2 since $y^{\leftarrow}$ is left saturated. Again by (1.44), this means that $G(x,y)^{\rightarrow\leftarrow} \subseteq G(x,y)$, hence $\mathit{Gap}(x,y)$ is left saturated. □

A relational monoid is said to be *Gap saturated* when it is commutative and the subset $\mathit{Gap}(x,y)$ is left saturated for every $x,y \in M$. As anticipated above, this condition can be written at first order. In fact consider $x,y \in M$, then to say that $\mathit{Gap}(x,y)$ is left saturated means that $\mathit{Gap}(x,y)^{\rightarrow\leftarrow} \subseteq \mathit{Gap}(x,y)$. By Lemma 2 this is

$$\bigcap_{\mathit{Gap}(x,y)\subseteq s^{\leftarrow}} s^{\leftarrow} \subseteq \mathit{Gap}(x,y)$$

and, by unfolding all the involved definitions, this is equivalent to:

$$\forall s(\,\forall t(\,t\cdot x\,R\,y \rightarrow t\,R\,s\,)\rightarrow z\,R\,s\,)\rightarrow z\cdot x\,R\,y. \tag{1.46}$$

Here we call this property *elemental gap saturation*, but it appears in [137] and corresponds to *continuity* in *linear frames*, that are algebraic models for the Intuitionistic Linear Logic that includes modalities.

Now it is easy to formulate a soundness theorem for **ILL** and Gap saturated birelational models. As should be clear from the previous reasoning, the evaluation is extended to implication by

$$V(\varphi\rightarrow\psi) \stackrel{\text{def}}{=} V(\varphi)\rightarrow V(\psi) \tag{1.47}$$

Theorem 8 (Soundness). *A sequent deducible in* **ILL** *is valid in every Gap saturated relational monoid.*

Proof. Soundness can be proved by showing the validity of all the meta-linguistic equations defining the calculus **ILL**, see Fig. 1.3, and the cut rule, used to solve the equations in order to define the calculus. Thanks to Proposition 5 it is sufficient to verify the definitional equations for **B**, the cut rule, and the definitional equation for $\rightarrow$. All the work has been already done: §1.4 has proved that relational monoids are sound for **B**'s definitional equations and this section proved that Gap saturated relational monoids are sound for the definitional equation of $\rightarrow$. □

Once soundness is proved, we consider completeness. Thanks to the modular approach of the logic and the constructivism of the proofs, the results proved in §1.5 for **B** can be extended to **ILL**. The canonical model has already been defined in §1.5, and refined in §1.6 to the commutative case. Here it will be further extended to consider the calculus **ILL**. Then there will be only two things to check:

1. The canonical model belongs to the class of the models that are sound for **ILL**, namely it is Gap saturated.

2. The Canonical Evaluation Lemma 4 extends to the new connective $\rightarrow$.

In particular, to have refined completeness, everything must be proven by carefully dealing with cut rule: derivations cannot use cuts. The canonical model for **ILL** is obtained by refining the one given in (1.30). The difference is in the binary relation, once again it reflects the provability of the sequents, but it must consider the asymmetric nature of the calculus, as sequents in **ILL** can only have a single formula on the right hand side. The usual solution adopted for models of Linear Logic, see [74, 127], is to consider the set of formulae instead of lists of formulae with $\otimes$ as monoidal operation. This might be done also in this case, but it would be suitable to prove only standard completeness, not the refined one. In fact, the monoidal operation is required to be associative and commutative, and so is $\otimes$ only if the set of formulae is the quotient by the provability relation $\vdash$, that becomes an equivalence relation only by allowing cuts.

Then the underlying set must be the set of non-ordered lists of formulae $Frm^{\circledast}$. There is now another inconvenience: with $Frm^{\circledast}$ as underlying set, the evaluation cannot simply be the cut-free provability in **ILL**, once again because of the asymmetry of the calculus **ILL**. If the relation is simply the provability in the calculus the canonical evaluation lemma is not valid for 0. In fact, assume to define $\Gamma R \Delta$ if and only if $\Gamma \vdash \Delta$ is derivable in **ILL**, then this means that Δ can only be a single formula. The evaluation of 0 is $V(0) \stackrel{\text{def}}{=} \emptyset^{\rightarrow\leftarrow}$ that in the canonical model is $(Frm^{\circledast})^{\leftarrow}$. In this case $(Frm^{\circledast})^{\leftarrow}$ is $\emptyset$ as $(\varphi_1, \ldots \varphi_n)^{\leftarrow} = \emptyset$ whenever $n \neq 1$. Then Lemma 4 cannot hold for 0, as it requires $0 \in V(0)$ in the canonical model.

In order to prove Lemma 4, the solution is to change the binary relation in the model. It will still be the provability between contexts and single formulae, and it will become the trivial one between contexts, as described by the following definition.

Definition 4 (Syntactic Model for ILL). *The* syntactic model $\mathcal{F}_{\mathbf{ILL}}$ *corresponds to the structure* $(Frm^{\circledast}, \circ, [\,], \vdash_{\mathbf{ILL}})$, *where:*

1. *$Frm^{\circledast}$ is the set of non-ordered lists of formulae in $\mathcal{L}$.*

2. *The operation $\circ$ is the concatenation between lists.*

3. *The symbol $[\,]$ represents the empty list.*

4. *The relation $\vdash_{\mathbf{ILL}}$ is defined as*

$$\Gamma \vdash_{\mathbf{ILL}} \Delta \text{ if and only if } \begin{cases} \Gamma \vdash \Delta \text{ in } \mathbf{ILL} \text{ without cut rules} & \text{if } \Delta \in Frm; \\ \text{always} & \text{otherwise.} \end{cases}$$

Note that in case of the constant 0: (*i*) the sequent $0 \vdash \delta$ is cut-free derivable thanks to the rule $0L$, and (*ii*) the definition says that $0 R \delta_1, \ldots, \delta_n$ whenever $n \neq 1$. Hence

$0\,\epsilon(Frm^{\circledast})^{\leftarrow} = \emptyset^{\rightarrow\leftarrow}$. In general for every single formula φ a context Γ is in $\varphi^{\leftarrow}$ if and only if $\Gamma \vdash \varphi$ in **ILL** without cuts. In particular property (1.37) becomes:

$$\Gamma, \Sigma\,\epsilon\,\chi^{\leftarrow} \quad \text{if and only if} \quad \Gamma\,\epsilon(\Sigma^{\otimes} \to \chi)^{\leftarrow}. \tag{1.48}$$

Moreover, defining *Contx* as the set of contexts that are not a single formula, including the empty context, for every $A \subseteq Frm^{\circledast}$ it is the case that $Contx \subseteq A^{\rightarrow}$, and conversely for every $\Gamma\,\epsilon\,Contx$, the subset $\Gamma^{\leftarrow}$ is the whole $Frm^{\circledast}$. Hence conclude that for every $B \subseteq Frm^{\circledast}$

$$B^{\leftarrow} = \bigcap_{\delta\,\epsilon\,B} \delta^{\leftarrow} \tag{1.49}$$

where δ ranges exclusively over single formulae, as the contribution of the contexts in *Contx* is irrelevant for the intersection.

The canonical model $\mathcal{F}_{\mathbf{ILL}}$ is in the class of sound models for **ILL**: it is a relational monoid, as saw in §1.5, and it is gap saturated. To show the gap saturation, we should prove the property (1.46) on elements or one of the equivalent properties on subsets outlined by Proposition 7. It is easier to show that the implication between subsets is left saturated.

Proposition 9. *In the canonical model $\mathcal{F}_{\mathbf{ILL}}$, the implication $A \to B$ is left saturated for every $A, B \subseteq Frm^{\circledast}$.*

Proof. Consider $A, B \subseteq Frm^{\circledast}$, then the definition in 1.40 says that $\Gamma\,\epsilon\,A \to B$ if and only if $\Gamma \cdot A \subseteq B^{\rightarrow\leftarrow}$. Now, the saturation $B^{\rightarrow\leftarrow}$ is $\cap_{\Delta\epsilon B^{\rightarrow}}\Delta^{\leftarrow}$, that is $\cap_{\delta\epsilon B^{\rightarrow}}\delta^{\leftarrow}$ thanks to (1.49). Hence $\Gamma\,\epsilon\,A \to B$ means $\Gamma \cdot A \subseteq \cap_{\delta\epsilon B^{\rightarrow}}\delta^{\leftarrow}$. Then for every $\Sigma\,\epsilon\,A$ and $\delta\,\epsilon\,B^{\rightarrow}$ it holds $\Gamma, \Sigma\,\epsilon\,\delta^{\leftarrow}$, and furthermore $\Gamma\,\epsilon(\Sigma^{\otimes} \to \delta)^{\leftarrow}$ by (1.48). Conclude that

$$A \to B = \bigcap_{\substack{\Sigma\,\epsilon\,A \\ \delta\,\epsilon\,B^{\rightarrow}}} (\Sigma^{\otimes} \to \delta)^{\leftarrow}.$$

Then $A \to B$ is left saturated, as intersection of left saturated subsets. □

The other point to prove is that the canonical evaluation lemma still holds for the new model $Frm^{\circledast}$. On single formulae, the relation R of Definition 4 corresponds to the one in Definition 3, and moreover it is defined 'ad hoc' to deal with $V(0)$ as explained above. Hence the only thing to check is that the induction done in the proof of Lemma 4 extends to the new connective $\to$. The principal attention is needed to check that all the reasoning is cut-free. To this proposal, the rules of the sequent calculus must first be translated into properties of the syntactic model, as done in §1.5 with Tab.1.3. There is a nice property involving $\to L$.

Proposition 10. *In the syntactical model $\mathcal{F}_{\mathbf{ILL}}$, the rule $\to L$ corresponds to:*

$$(\varphi \to \psi) \circ (\varphi)^{\leftarrow} \subseteq (\psi)^{\rightarrow\leftarrow}. \tag{1.50}$$

Proof. With reference to Fig. 1.3, taking Γ_2 empty, the rule $\to L$ says that

$$\frac{\Gamma_1 \vdash \varphi \quad \psi \vdash \delta}{\varphi \to \psi, \Gamma_1 \vdash \delta} ,$$

that in the syntactical model is

$$\frac{\Gamma_1 \,\epsilon\, \varphi^{\leftarrow} \quad \psi \,\epsilon\, \delta^{\leftarrow}}{\varphi \to \psi, \Gamma_1 \,\epsilon\, \delta^{\leftarrow}} .$$

Hence $(\varphi \to \psi) \circ (\varphi)^{\leftarrow} \subseteq \delta^{\leftarrow}$ for every δ such that $\psi \,\epsilon\, \delta^{\leftarrow}$. The property trivially extends op to contexts Δ, then conclude that $(\varphi \to \psi) \circ (\varphi)^{\leftarrow} \subseteq \Delta^{\leftarrow}$ for every Δ such that $\psi \,\epsilon\, \Delta^{\leftarrow}$. Then $(\varphi \to \psi) \circ (\varphi)^{\leftarrow} \subseteq \bigcap_{\psi \,\epsilon\, \Delta^{\leftarrow}} \Delta^{\leftarrow}$. Thanks to Lemma 2 this means that $(\varphi \to \psi) \circ (\varphi)^{\leftarrow} \subseteq \psi^{\to\leftarrow}$.

On the other hand assume property (1.50) and prove $\to L$, as it is written in Fig. 1.3. Consider the premisses $\Gamma_1 \vdash \varphi$ and $\Gamma_2, \psi \vdash \delta$. They means that $\Gamma_1 \,\epsilon\, \varphi^{\leftarrow}$ and, by (1.48), that $\psi \,\epsilon (\Gamma_2^{\otimes} \to \delta)^{\leftarrow}$. Now, (1.50) says that $\varphi \to \psi, \Gamma_1 \,\epsilon (\Gamma_2^{\otimes} \to \delta)^{\leftarrow}$, that is $\varphi \to \psi, \Gamma_1, \Gamma_2 \,\epsilon\, \delta^{\leftarrow}$ again by (1.48). Conclude that $\varphi \to \psi, \Gamma_1, \Gamma_2 \vdash \delta$. □

With regard to the rule $\to R$, it is easy to verify that in the syntactic model it corresponds to the property

$$\Gamma, \varphi \,\epsilon (\psi)^{\leftarrow} \text{ implies } \Gamma \,\epsilon (\varphi \to \psi)^{\leftarrow} . \tag{1.51}$$

These two properties are sufficient to extend the inductive step of Lemma 4 to the connective $\to$.

Proposition 11. *Provided that $\varphi \,\epsilon\, V(\varphi) \subseteq \varphi^{\leftarrow}$ and $\psi \,\epsilon\, V(\psi) \subseteq \psi^{\leftarrow}$, then $\varphi \to \psi \,\epsilon\, V(\varphi \to \psi) \subseteq (\varphi \to \psi)^{\leftarrow}$.*

Proof. Assume that $\varphi \,\epsilon\, V(\varphi) \subseteq \varphi^{\leftarrow}$ and $\psi \,\epsilon\, V(\psi) \subseteq \psi^{\leftarrow}$. To prove that $\varphi \to \psi \,\epsilon\, V(\varphi \to \psi)$ use the hypothesis (*i*) $V(\varphi) \subseteq (\varphi)^{\leftarrow}$ and (*ii*) $\psi \,\epsilon\, V(\psi)$. Then

$$\begin{array}{lll} (\varphi \to \psi) \circ V(\varphi) & \subseteq (\varphi \to \psi) \circ (\varphi)^{\leftarrow} & \text{by } (i) \\ & \subseteq (\psi)^{\to\leftarrow} & \text{by (1.50)} \\ & \subseteq V(\psi)^{\to\leftarrow} & \text{by } (ii) \end{array}$$

hence conclude that $(\varphi \to \psi) \,\epsilon\, V(\varphi) \to V(\psi) = V(\varphi \to \psi)$ by definition.

On the other hand, to prove that $V(\varphi \to \psi) \subseteq (\varphi \to \psi)^{\leftarrow}$ use the hypothesis (*iii*) $\varphi \,\epsilon\, V(\varphi)$ and (*iv*) $V(\psi) \,\epsilon (\psi)^{\leftarrow}$. Then assume $\Gamma \,\epsilon\, V(\varphi \to \psi)$, which is $\Gamma \circ V(\varphi) \subseteq V(\psi)$ by definition, hence deduce $\Gamma, \varphi \,\epsilon\, V(\psi)$ by (*iii*), and $\Gamma, \varphi \,\epsilon (\psi)^{\leftarrow}$ by (*iv*). Conclude that $\Gamma \,\epsilon\, \psi^{\leftarrow}$ by (1.51), hence the thesis. □

We conclude that the following holds

Theorem 9 (Refined Completeness for ILL). *If a sequent is valid in every Gap saturated relational monoid, then it is derivable in* **ILL** *without cuts.*

As for **B**, this theorem provides a semantical cut-elimination result for **ILL**. It is easy to check that this fact makes the syntactical relation a preorder between single formulae. This fact suggests that the semantics may be specified by considering monoids with relations that are preorders, or even partial orders between the classes induced by the logical equivalence that derives from a preorder. As matter of fact this can be done in case the demand for a refined completeness is released; but an order relation does not further simplify the semantics as it does for Intuitionistic Logic (see §1.7.2). Nevertheless, this provides a link with *pretopologies* [127], as it can be seen that Gap saturated preordered monoids are actually pretopologies: topological models that have been proved sound and complete for **ILL** in [127].

1.7.2 Relational Semantics for Intuitionistic Logic

The results in § 1.6, for structural rules, and § 1.7.1, for **ILL**, hint how to extend soundness and (refined) completeness results to the intuititionistic calculus **IL**. As described in §1.7, **IL** is obtained by extending **ILL** (see Fig. 1.3) with structural rules: weakening and contraction. Then the sound and (refined) complete models for **IL** are those sound and (refined) complete for **ILL** that enable the structural rules: the gap saturated relational monoids satisfying the rules ($w1$) and ($c1$), cf. Tab. 1.4.

Theorem 10 (Soundness and Refined Completeness for IL). *If a sequent is derivable in* **IL***, then it is valid in every gap saturated relational monoid that satisfies* ($w1$) *and* ($c1$)*. Moreover, if it is valid in every gap saturated relational monoid satisfying* ($w1$) *and* ($c1$)*, then it is derivable in* **IL** *without using cut rules.*

For completeness, in particular, the syntactical model $\mathcal{F}_{\mathbf{IL}}$ is the same as Definition 4, but with the provability in **ILL** replaced by the provability in **IL**. Once again, cut elimination is a direct consequence of the this semantical result.

Corollary 3 (Semantical Cut Elimination for IL). *If a sequent is derivable in* **IL** *(even by using cut rules), then it admits a cut-free derivation in* **IL***.*

Gap saturated relational monoids satisfying ($w1$) and ($c1$) may sound a bit baroque and too elaborated, when compared with other models given in literature for Intuitionistic Logic, but it is worth stressing that they provide a semantical cut elimination result to the calculus in a complete constructive way, and, in our knowledge, this is not provided by any other 'simpler' model. Indeed, the work in [112] shows a semantical cut elimination result for Intuitionistic Logic, but the models used there are again monoids with a closure operator that turns out to need the same properties required to the operator $(\,)^{\rightarrow\leftarrow}$ in the relational monoids.

By comparing this semantics with other models for Intuitionist Logic, there may be a twofold source of concern: usually there is no need to have a binary operation to evaluate intuitionistic formulae, and the binary relation R is commonly taken to be a partial order. Our conjecture is that the monoidal operation and the 'generic' relation are an essential

requirement to a refined completeness, or, in other words, to a constructive semantical cut-elimination. In fact, when the syntactic model is built by considering sequences of formulae, an operation is implicitly assumed: sequence composition. Furthermore, the relation cannot be transitive, since it subsumes the 'provability' between contexts. As a matter of fact, the work in [83] provides a semantical cut elimination theorem for Intuitionistic Logic in terms of a Kripke semantics founded on partially ordered sets, that looks more intuitive, indeed. However, such a result is not constructive, as the completeness proof is given by a 'classical' reasoning on counter-models.

As a refined completeness result for Intuitionistic Logic has been already achieved, we may focus on improving the semantics to obtain a more elegant model for soundness and (not refined) completeness. Moreover, with the cut-elimination theorem in hand, we can use cut to show that a more usual notion of syntactical model, similar to Lindenbaum-style term models [136], provides a completeness theorem.

First of all, it is worth noticing that in the syntactical model $\mathcal{F}_{\mathbf{IL}}$ the relation $\vdash_{\mathbf{IL}}$ restricted to formulae is actually transitive, as Corollary 3 proves that the cut-free calculus admits the rule

$$\frac{\varphi \vdash \chi \quad \chi \vdash \psi}{\varphi \vdash \psi}$$

Moreover, axioms make the relation $\vdash_{\mathbf{IL}}$ reflexive on formulae. Hence, by restricting the underlying set to be the set of formulae, the canonical model becomes a preordered set. So, thinking to the elements of the canonical model as single formulae, we focus our reasoning on preordered relations.

In the rest part of the section we consider the models for **IL**, namely gap saturated relational monoids satisfying $(c1)$ and $(w1)$, and we consider the relation to be a preorder. The aim is to study how to simplify this models, by preserving soundness and completeness results. Then assume $\mathcal{M} = (M, \cdot, 1, \leq)$ to be a gap saturated preordered monoid satisfying $(c1)$ and $(w1)$.

When the syntactical model is restricted on formulae, dropping contexts, it is not a monoid anymore. In fact, the syntactical operation composes the formulae by ',' and it produces contexts as a result, hence it is not an internal binary operation on the set of formulae.

The question to address in the following is whether a binary operation is actually essential to prove a soundness theorem for Intuitionistic Logic with respect to the semantics defined in Tab. 1.2 and (1.47), or the requirement for a monoidal structure can be dismissed.

First of all, it is common knowledge that in **IL** the operators $\otimes$ and $\&$ collapse, due to the structural rules. This hints that a sound model may get rid of the monoidal operation. In fact, as $\varphi \otimes \psi$ is logically equivalent to $\varphi \,\&\, \psi$, it must be the case that also the semantical evaluations collapse in every sound and complete model for **ILL**, namely

$$V(\varphi \otimes \psi) = V(\varphi \,\&\, \psi). \tag{1.52}$$

This is easy to check thanks to the properties $(w1)$ and $(c1)$. First recall that $V(\varphi \otimes \psi) =$

$(V(\varphi)\cdot V(\psi))^{\rightarrow\leftarrow}$, and $V(\varphi \,\&\, \psi) = V(\varphi)\cap V(\psi)$ by definition. Then the following lemma is what is needed to verify property (1.52).

Lemma 6. *If A, B are left saturated subsets in a gap saturated preordered monoid that satisfies* $(w1)$ *and* $(c1)$*, then* $(A\cdot B)^{\rightarrow\leftarrow} = A\cap B$*.*

Proof. Property $(w1)$ provides the inclusion $(A\cdot B)^{\rightarrow\leftarrow}\subseteq A\cap B$, thanks to the equivalent property (1.24) and the fact that A and B are left saturated. On the other hand, property $(c1)$ provides the inclusion $A\cap B\subseteq (A\cdot B)^{\rightarrow\leftarrow}$. In fact, the equivalent property (1.26) says that $A\cap B\subseteq ((A\cap B)\cdot(A\cap B))^{\rightarrow\leftarrow}$, hence $A\cap B\subseteq (A\cdot B)^{\rightarrow\leftarrow}$, as $A\cap B\subseteq A$ and $A\cap B\subseteq B$ implies $(A\cap B)\cdot(A\cap B)\subseteq A\cdot B$. □

Note that this lemma holds in general for any relation, as the properties of a preorder have not been used in the proof. The lemma suggests that the monoidal operation may be removed, since the operation is not essential to evaluate the connective $\otimes$. At this point, as the constant 1 collapses to $\top$, the implication seems to be the only connective that needs a binary operation on the semantical counter part is the implication. Then consider its semantical evaluation: the implication between subsets.

Given $A, B\subseteq M$, and according to (1.40), $x\,\epsilon\, A\rightarrow B$ means that $x\cdot A\subseteq B^{\rightarrow\leftarrow}$. For gap saturated monoids, this is equivalent to $(\{x\}^{\rightarrow\leftarrow}\cdot A^{\rightarrow\leftarrow})^{\rightarrow\leftarrow}\subseteq B^{\rightarrow\leftarrow}$ thanks to Proposition 7 and closure properties. Finally, $(\{x\}^{\rightarrow\leftarrow}\cdot A^{\rightarrow\leftarrow})^{\rightarrow\leftarrow} = \{x\}^{\rightarrow\leftarrow}\cap A^{\rightarrow\leftarrow}$ by Lemma 6. Hence $x\in A\rightarrow B$ is equivalent to $\{x\}^{\rightarrow\leftarrow}\cap A^{\rightarrow\leftarrow}\subseteq B^{\rightarrow\leftarrow}$. Now, Proposition 4 says that $\downarrow\{x\}\subseteq\{x\}^{\rightarrow\leftarrow}$ and $\downarrow A\subseteq A^{\rightarrow\leftarrow}$, hence

$$x\,\epsilon\, A\rightarrow B \quad \text{implies} \quad \downarrow x\cap\downarrow A\subseteq B^{\rightarrow\leftarrow}. \tag{1.53}$$

As the second member of (1.53) does not involve the monoidal operation, a good candidate for an 'intuitionistic' semantical implication seems to be the operator

$$A\Rightarrow B \stackrel{\text{def}}{=} \{x \,:\, \downarrow x\cap\downarrow A\subseteq B^{\rightarrow\leftarrow}\}. \tag{1.54}$$

Likewise to relational monoids, it is easy to check that this definition provides the property

$$C\subseteq (A\Rightarrow B) \quad \text{if and only if} \quad \downarrow C\cap\downarrow A\subseteq B^{\rightarrow\leftarrow}. \tag{1.55}$$

Moreover, as $A\rightarrow B$ is left saturated for gap saturated relational monoids, the subset $A\Rightarrow B$ must be left saturated, as well, in order to give an evaluation to the intuitionistic implication, . We need a first order condition, as discussed for gap saturation in case of **ILL**. Again, we define a particular class of subsets parameterised by pairs of elements in M. For every $x, y\in M$, define

$$Low(x,y) \stackrel{\text{def}}{=} \{z \,:\, \downarrow z\cap\downarrow x\leq y\}. \tag{1.56}$$

A comparison with (1.44) hits the similarities between *Gap* and *Low*. A direct consequence of the definition is the following proposition.

Proposition 12. *In every preordered set $\mathcal{M} = (M, \leq)$, the following properties are equivalent:*

1. *$A \Rightarrow B$ is left saturated for every $A, B \subseteq M$.*

2. *$Low(x, y)$ is left saturated for every $x, y \in M$.*

Proof. The first point implies the second one. In fat, the subset $Low(x, y)$ is $\{z : \downarrow z \cap \downarrow x \subseteq \{y\}^{\rightarrow\leftarrow}\} = \{x\} \Rightarrow \{y\}$ as the operators $\downarrow$ and $(\,)^{\rightarrow\leftarrow}$ collapse on singletons.

On the other hand, to prove that the second point implies the first one, assume that $Low(x, y)$ is left saturated for every $x, y \in M$. Property (1.55) implies that $(A \Rightarrow B) \cap \downarrow A \subseteq B^{\rightarrow\leftarrow}$, hence $(A \Rightarrow B) \cap \downarrow x \subseteq \downarrow y$ for every $x \epsilon A$ and $y \epsilon B$, by Lemma 2 and (1.13). Then $(A \Rightarrow B) \subseteq Low(x, y)$ for every $x \epsilon A$ and $y \epsilon B$, by definition. As $Low(x, y)$ is left saturated by hypothesis, $(A \Rightarrow B)^{\rightarrow\leftarrow} \subseteq Low(x, y)$ for every $x \epsilon A$ and $y \epsilon B$. Then $\downarrow((A \Rightarrow B)^{\rightarrow\leftarrow}) \subseteq Low(x, y)$ for every $x \epsilon A$ and $y \epsilon B$, by Proposition 4. Now conclude that $\downarrow((A \Rightarrow B)^{\rightarrow\leftarrow}) \cap \downarrow A \subseteq B^{\rightarrow\leftarrow}$, by reversing the reasoning. Hence $(A \Rightarrow B)^{\rightarrow\leftarrow} \subseteq (A \Rightarrow B)$, that makes $(A \Rightarrow B)$ left saturated. □

A preordered set M is said to be *Low saturated* when the subsets $Low(x, y)$ are left saturated for every $x, y \in M$. Likewise §1.7.1, this is a condition that is easy to write at first order, hence it is the right property to require for a soundness result with respect to Intuitionistic Logic. The low saturated preordered sets are what we aimed for: they are not monoids anymore, and they are sound and complete models for Intuitionistic Logic, as we formally show in the remainder of the section.

The language $\mathcal{L}_{int}$ of Intuitionistic Logic consists of propositional variables, the propositional constants $\top$ and 0, and the connectives $\oplus$, $\&$ and $\rightarrow$. Their definitional equations are the corresponding ones in Fig. 1.3. The derived intuitionistic calculus, obtained by solving those definitional equation, is the fragment of **ILL** without $\otimes$ and 1. The evaluation of formulae and contexts in a low saturated preordered set $(M, \leq)$ is directly derived form the evaluation in gap saturated relational monoids, and for sake of clarity it is outlined in Tab. 1.6. Note the intersection with M in the definition of $Ingr(\,)$. This is necessary to evaluate the empty context, that reflected in the logic by $\top$, as 1 it collapse with $\top$ in **ILL**, due to structural rules. Moreover , thanks to Proposition 4, the evaluation $V(\varphi)$ is both left and down saturated for every formula φ. The general properties and definition of validity of a sequent remain the same as in §1.4. All these definitions are suitable for a soundness theorem for Intuitionistic Logic and low saturated preordered sets.

Theorem 11 (Soundness for Intuitionistic Logic). *A sequent deducible in Intuitionistic Logic is valid in every low saturated preordered set.*

Proof. As for Theorem 2 it is sufficient to prove the validity of definitional equations, cut and structural rules. The proof proceeds with the same argumentation as in § 1.7.1, in particular the semantical counter parts of the definitional equations for $\rightarrow$ and $\oplus$ derives from property (1.55), as the evaluation V is down saturated for every formula. □

Table 1.6 Evaluation of Formulae in Low Saturated Preordered Sets

$V(\top)$	$\stackrel{\text{def}}{=} M$	$V(0)$	$\stackrel{\text{def}}{=} \emptyset^{\rightarrow\leftarrow}$
$V(\varphi \,\&\, \psi)$	$\stackrel{\text{def}}{=} V(\varphi) \cap V(\psi)$	$V(\psi \oplus \varphi)$	$\stackrel{\text{def}}{=} (\, V(\psi) \cup V(\varphi)\,)^{\rightarrow\leftarrow}$
$V(\varphi \rightarrow \psi)$	$\stackrel{\text{def}}{=} V(\varphi) \Rightarrow V(\psi)$	$Ingr(\gamma_1, \ldots, \gamma_n)$	$\stackrel{\text{def}}{=} V(\gamma_1) \cap \ldots \cap V(\gamma_n) \cap M$

As anticipated, the syntactical model suitable for completeness is $\mathcal{F}_{int} \stackrel{\text{def}}{=} (Frm_{int}, \vdash)$, that is the set *Frm* of formulae of the language $\mathcal{L}_{int}$ with the intuitionistic provability as preorder. A interesting property in such a model is that

$$\downarrow\varphi \cap \downarrow\psi = \downarrow(\varphi \,\&\, \psi); \tag{1.57}$$

the forward inclusion derives from $\&L$, the other one from $\&R$.

To prove completeness, first of all $\mathcal{F}_{int}$ must be proved to be in the class of the models sound for Intuitionistic Logic, or rather that it is low saturated. Thanks to Proposition 12 it is sufficient to prove that the subset $A \Rightarrow B$ is left saturated for every couple of subsets $A, B \subseteq Frm_{int}$, and this is the result provided by next proposition. Its proof is similar to the one of Proposition 9 for $\mathcal{F}_{\mathbf{ILL}}$; in particular, the role of the conjunction ',' is taken by the connective $\&$. In fact it is easy to see that the three following sequents are equivalent in Intuitionistic Logic:

$$\chi, \varphi \vdash \psi \qquad \chi \,\&\, \varphi \vdash \psi \qquad \chi \vdash \varphi \rightarrow \psi. \tag{1.58}$$

In particular by defining

$$\Gamma^{\&} \stackrel{\text{def}}{=} \begin{cases} \top & \text{if } \Gamma = [\,]; \\ \gamma_1 \,\&\, \ldots \,\&\, \gamma_n & \text{if } \Gamma = \gamma_1, \ldots, \gamma_n \text{ with } n \geq 1. \end{cases} \tag{1.59}$$

it easy to verify

$$\Gamma \vdash \varphi \quad \text{ if and only if } \quad \Gamma^{\&} \vdash \varphi; \tag{1.60}$$

$$\downarrow(\Gamma^{\&}) = \downarrow\gamma_1 \cap \ldots \cap \downarrow\gamma_n \cap M \tag{1.61}$$

for every context Γ.

Moreover $\downarrow\chi \cap \downarrow\varphi \subseteq \downarrow\psi$ is equivalent to $\downarrow(\chi \,\&\, \varphi) \subseteq \downarrow\psi$, by (1.57). Due to transitivity, i.e., the cut rule, this says that $\chi \,\&\, \varphi \vdash \psi$, and this is equivalent to $\chi \vdash \varphi \rightarrow \psi$, by (1.58), namely $\chi \,\epsilon\, \downarrow(\varphi \rightarrow \psi)$. Hence

$$\downarrow\chi \cap \downarrow\varphi \subseteq \downarrow\psi \quad \text{ if and only if } \quad \chi \,\epsilon\, \downarrow(\varphi \rightarrow \psi) \tag{1.62}$$

Proposition 13. *The subset $A \Rightarrow B$ is left saturated for every couple of subsets $A, B \subseteq Frm_{int}$.*

Proof. Assume $\chi \in A \Rightarrow B$. Then $\downarrow\chi \cap \downarrow A \subseteq B^{\rightarrow\leftarrow}$ by definition. Recalling Lemma 2 and (1.13), and fixing $\varphi \epsilon A$ and ψ such that $B \subseteq \downarrow\psi$, this means that $\downarrow\chi \cap \downarrow\varphi \subseteq \downarrow\psi$, and this is equivalent to $\chi \epsilon \downarrow(\varphi \to \psi)$ by (1.62). Conclude that

$$A \Rightarrow B = \bigcap_{\substack{\varphi \epsilon A \\ B \subseteq \downarrow\psi}} \downarrow(\varphi \to \psi).$$

Then $A \Rightarrow B$ is left saturated, as intersection of left saturated subsets. □

Now the syntactical model $\mathcal{F}_{int}$ has been proved to be a low saturated preordered set. The last step before completeness is to define the canonical evaluation on atoms as $V(p) \stackrel{\text{def}}{=} \downarrow p$ and to verify the canonical evaluation lemma. The results already obtained for **B** and **ILL** extends up to the current model, the only think to check is the evaluation for the implication, as it has been defined 'ad hoc' for the preordered set. By Recalling the observation in (1.18) at the end of §1.5, the canonical evaluation lemma is specialised to the following.

Lemma 7 (Canonical Evaluation for $\mathcal{F}_{int}$). *Under the canonical evaluation V in $\mathcal{F}_{int}$, for every formula φ of $\mathcal{L}_{int}$ it holds $V(\varphi) = \downarrow\varphi$. Moreover for every context Γ it holds $Ingr(\Gamma) = \downarrow(\Gamma^{\&})$.*

Proof. Proceed by induction on the structure of formulae. The basic step and the inductive cases for connectives $\&$ and $\oplus$ are derived form Lemma 4. Only the inductive step for the connective $\to$ must be checked. Assume that $V(\varphi) = \downarrow\varphi$ and $V(\psi) = \downarrow\psi$, the goal is to prove that $V(\varphi \to \psi) = \downarrow(\varphi \to \psi)$. Let $\chi \epsilon V(\varphi \to \psi)$, hence $\downarrow\chi \cap \downarrow V(\varphi) \subseteq V(\psi)^{\rightarrow\leftarrow}$ by definition, that is $\downarrow\chi \cap \downarrow\varphi \subseteq \downarrow\psi$ by hypothesis, and $\chi \epsilon \downarrow(\varphi \to \psi)$ by (1.62). Then the goal is proved.

For the second part of the thesis, let $\Gamma = \gamma_1, \ldots, \gamma_n$, then $Ingr(\Gamma) = \downarrow\gamma_1 \cap \ldots \cap \downarrow\gamma_n \cap M$, thanks to the first part of the lemma, hence $Ingr(\Gamma) = \downarrow(\Gamma^{\&})$ by (1.61). □

From this lemma it is easy to conclude the completeness theorem.

Theorem 12 (Completeness for Intuitionistic Logic). *If a sequent is valid in every Low saturated preordered set, then it is provable in Intuitionistic Logic.*

Proof. Assume $\Gamma \vdash \varphi$ be valid in every Low saturated preordered set. In particular, $Ingr(\Gamma) \subseteq V(\varphi)$ in the syntactic model $\mathcal{F}_{int}$ equipped with the canonical evaluation. Note that Lemma 7 says $\downarrow(\Gamma^{\&}) \subseteq \downarrow\varphi$, and this means $\Gamma^{\&} \vdash \varphi$, hence conclude that $\Gamma \vdash \varphi$ is provable in Intuitionistic Logic by (1.60). □

Here the models have been considered preordered, to be as consistent as possible with the previous sections of the chapter, and to keep uniformity among the presented canonical models. The completeness result can be nevertheless presented for *partial orders* $(\leq)$ as well. In fact, the canonical model becomes partial ordered by considering the usual set of equivalence classes induced by provability. Hence we can state a more general result.

Theorem 13 (Soundness and Completeness for Partial Orders). *A sequent is valid in every Low saturated partially ordered set if and only if it is provable in Intuitionistic Logic.*

1.7.3 Kripke Semantics

The last result of the previous section provides a link with Kripke's possible worlds semantics [94], as its models are just partially ordered sets. Usually Kripke semantics is given in term of a *forcing* relation between worlds, i.e., the elements of the partially ordered set, and formulae of the language. Such a relation is written as $x \Vdash \varphi$ and pronounced as '*x forces* φ.' In the case of the current relational semantics, $V(\varphi)$ can be seen as the subset of worlds that force the formula φ, hence it is straightforward to obtain an equivalent forcing relation by defining:

$$x \Vdash \varphi \quad \stackrel{\text{def}}{=} \quad x \in V(\varphi).$$

The fact that the evaluation $V(\varphi)$ is left saturated, hence down saturated, states that the forcing relation satisfies Kripke monotonicity, it is sufficient to consider an inverse order. Moreover, the definitions in Tab. 1.6 provide the inductive definition to the forcing relation. In the case of $\top$ and $\&$ it produces the usual Kripke clauses

$$\begin{array}{ll} x \Vdash \top & \text{for every } x \in M, \\ x \Vdash \varphi \,\&\, \psi & \text{if and only if } x \Vdash \varphi \text{ and } x \Vdash \psi. \end{array}$$

For the connective $\rightarrow$, the definition says that $V(\varphi \rightarrow \psi)$ is $\{x : \downarrow x \cap V(\varphi) \subseteq (\psi)\}$, hence $x \epsilon V(\varphi \rightarrow \psi)$ means that for every $y \leq x$, $y \epsilon V(\varphi)$ implies $y \epsilon V(\psi)$. In terms of forcing relation:

$$x \Vdash \varphi \rightarrow \psi \quad \text{if and only if} \quad \text{for every } y \leq x,\ y \Vdash \varphi \text{ implies } y \Vdash \psi,$$

that is the usual Kripke semantics for implication, again by inverting the order. The cases for the constant 0 and the connective $\oplus$ are not standard, as the definition becomes:

$$\begin{array}{ll} x \Vdash 0 & \text{if and only if } x \leq y \text{ for every } y \in M, \\ x \Vdash \varphi \oplus \psi & \text{if and only if } x \leq y \text{ for every } y \\ & \quad \text{such that } z \leq y \text{ for every } z \Vdash \varphi \text{ or } z \Vdash \psi. \end{array}$$

They look like more elaborated than in the original Kripke semantics. Our conjecture is that this is due to the fact that the soundness and completeness result we provide is entirely constructive. In our knowledge there are no constructive proof of the completeness of Kripke model, as the semantics require to know exactly the meaning of the connective *'or'* at the meta-level, hence admitting only a classical proof.

The definition resembles the one for Beth models [15], that provide a constructive proof for completeness. The relation between Low saturated partial ordered sets and Beth models will be the subject for further investigations.

The presence of a preorder and a closure operator reminds the definition of formal topologies [126]. It turns out that Low saturated preordered sets are actually formal topologies.

1.8 Towards Bunched Implications Logic

The Logic of Bunched Implications, introduced in [110], is a substructural system in which a multiplicative (linear) and an additive (intuitionistic) implication are freely combined. The calculus, in its propositional version, arises from a deep analysis of the proof-theoretic relationship between conjunction and implication. The meta-level considers structural rules carefully, and it assumes two meta-linguistic '*and*' links among formulae: one of them is linear, the other intuitionistic. Contexts are not lists of assertions anymore, but rather *bunches* of assertions (cf. [5]). A Bunch is a contexts with two combining operations: a single comma ',' that takes the place of the *linear* meta-linguistic *and*, and a semicolon ';' that takes the place of the *intuitionistic* meta-linguistic *and*. The former admits no structural rule except *Exchange*, whereas the latter admits also *Weakening* and *Contraction* (cf. Fig. 1.2).

Formally, bunches are structured as trees with the internal nodes labelled with either "," or ";" and leaves labelled with assertions. Bunches may be also represented by lists of lists, cf. [122]. They are generated by the grammar

$$\Gamma ::= \varphi \mid \emptyset_m \mid \Gamma, \Gamma \mid \emptyset_a \mid \Gamma; \Gamma$$

where φ is an assertion, $\emptyset_m$ and $\emptyset_a$ are empty bunches, multiplicative and additive respectively, their meaning will be clear in relation with the congruence relation between bunches. We write $\Gamma(\Delta)$ to refer to Δ as a *sub-bunch* of Γ, for a bunch Γ in which Δ appears as a sub-tree. We write $\Gamma(-)$ to denote a bunch which is incomplete and which may be completed by placing a bunch in its hole, and we will use this notation to refer to that part of $\Gamma(\Delta)$ which is not part of Δ. We require that "," and ";" be commutative monoid operations, giving rise to the *coherent equivalence*, $\Gamma \equiv \Delta$, defined as the least equivalence relation on bunches that satisfies:

1. Commutative monoid equations for $\emptyset_a$ and ";"
2. Commutative monoid equations for $\emptyset_m$ and ","
3. Congruence: if $\Delta \equiv \Delta'$ then $\Gamma(\Delta) \equiv \Gamma(\Delta')$.

The Logic of Bunched Implications can be introduced by the principle of reflection, as it has been done for all the logics presented till now. The language of Bunched Implications Logic consists of propositional constants $\top$, $\bot$ and 1, propositional variables $p, q, \ldots$, multiplicative connectives $*$ and $-\!\!*$, and additive connectives $\wedge$, $\rightarrow$ and $\vee$. Sequents are of the form

$$\Gamma \vdash \varphi$$

where Γ is a bunch and φ a formula, both produced by the language. Every connective and constant is introduced by a definitional equation, as outlined in Tab. 1.7. In particular, the connectives $*$ and $\wedge$ reflect the two meta-links ',' and ';' respectively; the propositional constants I and $\top$ reflect the empty assertions, multiplicative and additive respectively; the

Table 1.7 Definitional Equations for **LBI**

$(*)$	$\Gamma(\varphi * \psi) \vdash \chi$	if and only if	$\Gamma(\varphi, \psi) \vdash \chi$
$(-\!\!*)$	$\Gamma \vdash \varphi -\!\!* \psi$	if and only if	$\Gamma, \varphi \vdash \psi$
(I)	$\Gamma(I) \vdash \chi$	if and only if	$\Gamma(\emptyset_m) \vdash \chi$
$(\wedge)$	$\Gamma(\varphi \wedge \psi) \vdash \chi$	if and only if	$\Gamma(\varphi; \psi) \vdash \chi$
$(\rightarrow)$	$\Gamma \vdash \varphi \rightarrow \psi$	if and only if	$\Gamma; \varphi \vdash \psi$
$(\top)$	$\Gamma(\top) \vdash \chi$	if and only if	$\Gamma(\emptyset_a) \vdash \chi$
$(\vee)$	$\Gamma(\varphi \vee \psi) \vdash \chi$	if and only if	$\Gamma(\varphi) \vdash \chi$ and $\Gamma(\psi) \vdash \chi$
$(\bot)$	$\Gamma \vdash \chi$ and $\bot \vdash \chi$	if and only if	$\Gamma \vdash \chi$

connective $\vee$ reflects the meta-link *and* between sequents; and the propositional constant $\bot$ reflects trivial assertions for a link *and* between contexts. A special remark is needed for implications, of which there are actually two in the calculus. As we saw in §1.7, implication is inextricably bound up with conjunction, or at least with the antecedent-forming operations used to formulate sequents. In fact, as outlined in (1.32), the character of the implication in a logic is married to, and in a sense determined by, that of the meta-linguistic *and* among assertions. Since in case of bunches these links are two, in turn the possible implications are two. The connective $-\!\!*$ is bound up with ',' hence with $*$, and $\rightarrow$ with ';' hence with $\wedge$.

Definitional equations are solved as in §1.3: the backward direction of an equation gives the formation rule, the forward direction gives the implicit reflection rule, that is further specialised to the corresponding axiom of reflection, and finally to the explicit reflection rule. The basic rules assumed in this case are axioms, a structural rule involving the coherent equivalence, and a more complex form of *cut* involving the structure of bunches:

$$\frac{\Gamma \vdash \chi}{\Delta \vdash \chi}\,(\Delta \equiv \Gamma)\,E \qquad\qquad \frac{\Gamma \vdash \varphi \quad \Delta(\varphi) \vdash \chi}{\Delta(\Gamma) \vdash \chi}\ \textit{cut}$$

It is surprising to see how the pattern to follow to solve the definitional equations is persistent for every connective both of Bunched Implications Logic and of the previous logics we studied. To make this clear, we solve the definitional equations for Bunched Implications' multiplicatives.

Consider the definitional equation for $*$. The backward direction gives directly the $*L$ rule. The forward direction gives

$$\frac{\Gamma(\varphi * \psi) \vdash \chi}{\Gamma(\varphi, \psi) \vdash \chi}\ \textit{implicit } *\textit{-reflection.}$$

Then trivialise the premiss, by considering the axiom $\varphi * \psi \vdash \varphi * \psi$, thus obtaining the equivalent axiom

$$\varphi, \psi \vdash \varphi * \psi \quad \textit{axiom of } * -\textit{reflection}.$$

The implicit $*$-reflection is recovered by one application of the composition

$$\frac{\varphi, \psi \vdash \varphi * \psi \quad \Gamma(\varphi * \psi) \vdash \chi}{\Gamma(\varphi, \psi) \vdash \chi}$$

Then final solution is reached by replacing φ and ψ with arbitrary contexts Γ_1 and Γ_2, that is assuming that $\Gamma_1 \vdash \varphi$ and $\Gamma_2 \vdash \psi$ and applying two compositions

$$\frac{\Gamma_2 \vdash \psi \qquad \dfrac{\Gamma_1 \vdash \varphi \quad \varphi, \psi \vdash \varphi * \psi}{\Gamma_1, \psi \vdash \varphi * \psi}}{\Gamma_1, \Gamma_2 \vdash \varphi * \psi}$$

thus obtaining the rule

$$\frac{\Gamma_1 \vdash \varphi \quad \Gamma_2 \vdash \psi}{\Gamma_1, \Gamma_2 \vdash \varphi * \psi} \; *R \; .$$

To recover the axiom of $*$-reflection it is sufficient to trivialise the premiss with the two axioms involving φ and ψ.

The case for $-\!\!*$ is analogous: the involved contexts are different, but the structure is the same. Consider the definitional equation for $-\!\!*$. The backward direction gives directly the $-\!\!* \; R$ rule. The forward direction gives

$$\frac{\Gamma \vdash \varphi -\!\!* \psi}{\Gamma, \varphi \vdash \psi} \; \textit{implicit } -\!\!* \textit{-reflection.}$$

Then trivialise the premiss, by considering the axiom $\varphi -\!\!* \psi \vdash \varphi -\!\!* \psi$, thus obtaining the equivalent axiom

$$\varphi -\!\!* \psi, \varphi \vdash \psi \quad \textit{axiom of } -\!\!* \textit{ −reflection.}$$

The implicit $-\!\!*$-reflection is recovered by one application of the composition

$$\frac{\Gamma \vdash \varphi -\!\!* \psi \quad \varphi -\!\!* \psi, \varphi \vdash \psi}{\Gamma, \varphi \vdash \psi}$$

Then final solution is reached by assuming $\Gamma \vdash \varphi$ and $\Delta(\psi) \vdash \chi$, and by applying two compositions:

$$\frac{\dfrac{\Gamma \vdash \varphi \quad \varphi -\!\!* \psi, \varphi \vdash \psi}{\Gamma, \varphi -\!\!* \psi \vdash \psi} \qquad \Delta(\psi) \vdash \chi}{\Delta(\Gamma, \varphi -\!\!* \psi) \vdash \chi}$$

Thus obtaining the $-\!\!* \; L$ rule. The axiom of $-\!\!*$-reflection is obtained by trivialising the premisses with two axioms involving φ and ψ. It is worth noticing that the rule $-\!\!* \; L$ (see Fig. 1.4) is slightly different than the corresponding one presented in [70, 71, 122], defined as:

$$\frac{\Gamma \vdash \varphi \quad \Delta(\Delta', \psi) \vdash \chi}{\Delta(\Delta', \Gamma, \varphi -\!\!* \psi) \vdash \chi} \; [-\!\!* \; L]$$

It is easy to see that the two rules are equivalent thanks to the structure of bunches. In fact, $[\mathrel{-\!\!*}\, L]$ is a particular case of $\mathrel{-\!\!*}\, L$ (think of bunches as trees); vice versa $\mathrel{-\!\!*}\, L$ is a consequence of $[\mathrel{-\!\!*}\, L]$ thanks to the coherent equivalence on bunches:

$$\dfrac{\Gamma \vdash \varphi \quad \dfrac{\dfrac{\Delta(\psi) \vdash \chi}{\Delta(\emptyset_m, \psi) \vdash \chi}\, E}{\Delta(\emptyset_m, \Gamma, \varphi \mathrel{-\!\!*} \psi) \vdash \chi}\, [\mathrel{-\!\!*}\, L]}{\Delta(\Gamma, \varphi \mathrel{-\!\!*} \psi) \vdash \chi}\, E$$

We prefer the rule $\mathrel{-\!\!*}\, L$ for uniformity. As a matter of fact, also the rule $[\mathrel{-\!\!*}\, L]$ is a solution of the definitional equations.

To solve the definitional equation for constant I is very quick. The backward direction of the definition gives IR rule directly. On the other hand, the forward one gives the *implicit I-reflection*

$$\dfrac{\Gamma(I) \vdash \chi}{\Gamma(\emptyset_m) \vdash \chi}$$

Then by trivialising the premisses with the axiom $I \vdash I$ it gives the axiom of I-reflection, $\emptyset_m \vdash I$, that is the correct IL rule to chose.

The full sequent calculus is outline in Fig. 1.4, and it corresponds to the one in [70, 122].[2] Again, once solved, the definitional equations become properties actually verified by the calculus. Moreover, as in §1.3, the equational definitions are satisfied by the cut-free system, as formalised below.

Proposition 14 (Cut-Free Equations for LBI). *The calculus obtained from* **LBI** *by removing the cut rule satisfies the definitional equations for every connective and logical constant.*

Proof. As for Proposition 1 the proof follows a common pattern for every connective and constant: the backward direction of every equation is guaranteed by the formation rule, the forward direction is proved by induction on the length of the derivation. □

For every bunch Γ we define the *characteristic formula* $\widetilde{\Gamma}$ inductively as follows:

$$\begin{array}{rcl@{\qquad}rcl} \widetilde{\varphi} & \stackrel{\text{def}}{=} & \varphi & & & \\ \widetilde{\emptyset_m} & \stackrel{\text{def}}{=} & I & \widetilde{\emptyset_a} & \stackrel{\text{def}}{=} & \top \\ \widetilde{\Gamma, \Delta} & \stackrel{\text{def}}{=} & \widetilde{\Gamma} * \widetilde{\Delta} & \widetilde{\Gamma; \Delta} & \stackrel{\text{def}}{=} & \widetilde{\Gamma} \wedge \widetilde{\Delta} \end{array}$$

Thanks to the previous proposition it is easy to see that

$$\Gamma \vdash \varphi \text{ without using } \mathit{cut} \quad \text{if and only if} \quad \widetilde{\Gamma} \vdash \varphi \text{ without using } \mathit{cut}. \tag{1.63}$$

This fact will be useful in next section to evaluate bunches in the semantics.

[2] We refer to the errata provided for [122].

Figure 1.4 Sequent Calculus **LBI**

Axioms

$$\varphi \vdash \varphi$$

Operational Rules

Multiplicatives

$$\frac{\Gamma(\varphi,\psi)\vdash\chi}{\Gamma(\varphi * \psi)\vdash\chi}\ *L \qquad \frac{\Gamma\vdash\varphi \quad \Delta\vdash\psi}{\Gamma,\Delta\vdash\varphi * \psi}\ *R$$

$$\frac{\Gamma\vdash\varphi \quad \Delta(\psi)\vdash\chi}{\Delta(\Gamma,\varphi \mathrel{-\!\!*} \psi)\vdash\chi}\ \mathrel{-\!\!*} L \qquad \frac{\Gamma,\varphi\vdash\psi}{\Gamma\vdash\varphi \mathrel{-\!\!*} \psi}\ \mathrel{-\!\!*} R$$

$$\frac{\Gamma(\emptyset_m)\vdash\chi}{\Gamma(I)\vdash\chi}\ IL \qquad \frac{}{\emptyset_m\vdash I}\ IR$$

Additives

$$\frac{\Gamma(\varphi;\psi)\vdash\chi}{\Gamma(\varphi\wedge\psi)\vdash\chi}\ \wedge L \qquad \frac{\Gamma\vdash\varphi\Delta\vdash\psi}{\Gamma;\Delta\vdash\varphi\wedge\psi}\ \wedge R$$

$$\frac{\Gamma\vdash\varphi \quad \Delta(\psi)\vdash\chi}{\Delta(\Gamma;\varphi\to\psi)\vdash\chi}\ \to L \qquad \frac{\Gamma;\varphi\vdash\psi}{\Gamma\vdash\varphi\to\psi}\ \to R$$

$$\frac{\Gamma(\emptyset_a)\vdash\chi}{\Gamma(\top)\vdash\chi}\ \top L \qquad \frac{}{\emptyset_a\vdash\top}\ \top R$$

$$\frac{\Gamma(\varphi)\vdash\chi \quad \Gamma(\psi)\vdash\chi}{\Gamma(\varphi\vee\psi)\vdash\chi}\ \vee L \qquad \frac{\Gamma\vdash\varphi}{\Gamma\vdash\varphi\vee\psi}\ \vee R \quad \frac{\Gamma\vdash\psi}{\Gamma\vdash\varphi\vee\psi}\ \vee R$$

$$\frac{}{\bot\vdash\varphi}\ \bot L$$

Cut Rule

$$\frac{\Gamma\vdash\varphi \quad \Delta(\varphi)\vdash\chi}{\Delta(\Gamma)\vdash\chi}\ cut$$

Structural Rules

$$\frac{\Gamma(\Delta)\vdash\chi}{\Gamma(\Delta;\Delta')\vdash\chi}\ W \qquad \frac{\Gamma\vdash\chi}{\Delta\vdash\chi}\ (\Delta\equiv\Gamma)\,E \qquad \frac{\Gamma(\Delta;\Delta)\vdash\chi}{\Gamma(\Delta)\vdash\chi}\ C$$

1.8.1 Relational Semantics for Bunched Implications

As noticed in [110], the additive connectives of **LBI** correspond to those of Intuitionistic Logic **IL**, whereas the multiplicative connectives correspond to those of Multiplicative, Intuitionistic Linear Logic, or **IMLL** [74], that is the multiplicative fragment of the calculus **ILL** defined in §1.7. Thanks to this observation it is easy to adapt the relational semantics to **LBI**. It is natural to merge the two classes of relational monoids that are sound and (refined) complete for **IL** and **ILL** in order to obtain sound and (refined) complete models for **LBI**. The idea yields two orthogonal monoidal structures on a single set equipped with a relation: the *relational bi-monoids*.

Table 1.8 Evaluation of Formulae in Relational Bi-Monoids

$V(I)$	$\stackrel{\text{def}}{=}$	$\{1\}^{\rightarrow\leftarrow}$	$V(\top)$	$\stackrel{\text{def}}{=}$	$\{0\}$
$V(\varphi * \psi)$	$\stackrel{\text{def}}{=}$	$(\, V(\varphi) \times V(\psi)\,)^{\rightarrow\leftarrow}$	$V(\varphi \wedge \psi)$	$\stackrel{\text{def}}{=}$	$(\, V(\varphi) + V(\psi)\,)^{\rightarrow\leftarrow}$
$V(\varphi \mathrel{-\!\!*} \psi)$	$\stackrel{\text{def}}{=}$	$V(\varphi) \rightarrow^{\times} V(\psi)$	$V(\varphi \rightarrow \psi)$	$\stackrel{\text{def}}{=}$	$V(\varphi) \rightarrow^{+} V(\psi)$
$V(\varphi \vee \psi)$	$\stackrel{\text{def}}{=}$	$(\, V(\varphi) \cup V(\psi)\,)^{\rightarrow\leftarrow}$	$V(\bot)$	$\stackrel{\text{def}}{=}$	$\emptyset^{\rightarrow\leftarrow}$

Definition 5 (Relational Bi-Monoid). *A structure* $\mathcal{B} = (M, \times, +, 1, 0, R)$ *is a* relational bi-monoid *if*

- $\mathcal{B}^{\times} \stackrel{\text{def}}{=} (M, \times, 1, R)$ *is a gap saturated relational monoid.*
- $\mathcal{B}^{+} \stackrel{\text{def}}{=} (M, +, 0, R)$ *is a gap saturated relational monoid that satisfies the properties* $(c1)$ *and* $(w1)$ *(cf. Tab. 1.4).*

So, given a relational bi-monoid $\mathcal{B}$, the evaluation of connectives and constants is inherited form **ILL** and projected on $\mathcal{B}^{\times}$, for the multiplicative fragment, and it is inherited from **IL** and projected on $\mathcal{B}^{+}$, for additive one. Moreover there is no need to evaluate contexts, since with respect to cut-free provability they are equivalent to the corresponding characteristic formula. Hence, a given evaluation $V(p)$ on propositional variables p is extended on **LBI** formulae by the inductive clauses in Tab. 1.8. The two implications $\rightarrow^{\times}$ and $\rightarrow^{+}$ denote the corresponding operators in $\mathcal{B}^{\times}$ and $\mathcal{B}^{+}$, as defined in (1.40). The sequent $\Gamma \vdash \varphi$ is said to be *valid* in $\mathcal{B}$ if

$$V(\widetilde{\Gamma}) \subseteq V(\varphi) \qquad \text{for every evaluation } V \text{ on propositional variables.}$$

From the definitions, it is straightforward to prove the soundness theorem.

Theorem 14 (Soundness for Bunched Implications). *A sequent deducible in* **LBI** *is valid in every relational bi-monoid.*

Proof. As for Theorem 2, it is sufficient to check the validity of definitional equations. The equations reduces to the definition of the evaluations in the cases of $*$ and $\wedge$. The other cases are showed by using the semantical properties of the operators $\rightarrow^{\times}$ and $\rightarrow^{+}$ between subsets. Finally, the *cut* rule is valid as the combinations of subsets preserve inclusion. □

And now completeness. The syntactical model needed for the Canonical Evaluation Lemma follows the spirit of Definition 4. The elements of the syntactical bi-monoid are bunches and the two operations are the two way of combining them.

Definition 6 (Syntactical Model for LBI). *The* syntactic model $\mathcal{F}_{\mathbf{LBI}}$ *corresponds to the structure* $(Bunch, \times^{\star}, +^{\star}, \emptyset_m, \emptyset_a, \vdash_{\mathbf{LBI}})$, *where:*

1. *Bunch is the set of bunches generated by the language of* **LBI**.
2. *The operation* $\times^\star$ *is the composition by ','*
3. *The operation* $+^\star$ *is the composition by ';'*
4. *The symbol* $\emptyset_m$ *represents the empty multiplicative bunch.*
5. *The symbol* $\emptyset_a$ *represents the empty additive bunch.*
6. *The relation* $\vdash_{\mathbf{LBI}}$ *is defined as*

$$\Gamma \vdash_{\mathbf{LBI}} \Delta \quad \textit{if and only if} \quad \begin{cases} \Gamma \vdash \Delta \textit{ in } \mathbf{LBI} \textit{ without cut rules} & \textit{if } \Delta \in \textit{Frm}; \\ \textit{always} & \textit{otherwise.} \end{cases}$$

It is easy to check that $\mathcal{F}_{\mathbf{LBI}}$ is actually a relational bi-monoid, and, thanks to the proofs in §1.7.1 and §1.7.2, that the Canonical Evaluation Lemma is verified once again (cf. Lemma 4 and Proposition 11). Hence conclude the refined completeness theorem. Its proof follows the ones provided for Theorem 3, by recalling the property (1.63).

Theorem 15 (Refined Completeness for LBI). *If a sequent is valid in every relational bi-monoid, then it is derivable in* **LBI** *without using cut rules.*

Then, relational bi-monoids provide a semantical cut elimination result for the sequent calculus **LBI**.

Theorem 16. *If a sequent is derivable in* **LBI** *(even by using cut rules), then it admits a cut-free derivation in* **LBI**.

Relational bi-models resemble the *bicartesian doubly closed categories* introduced in [110]. The paper notes that the semantics of proofs for **IL** is given by using cartesian closed categories, and the one for **IMALL** by using symmetric monoidal closed categories. In each case, the paper observes that introduction rules for implications (the left rules in a sequent calculus) correspond to adjunctions where the internal hom is a right adjoint: to a cartesian product, for **IL**, and a tensor product, for **IMLL**. These two adjunctions can be seen also in relational bi-monoids: they are provided by the two monoidal operations and the relative implications.

Similarly to what we did here for relational bi-monoids, the paper [110] asks for a category that has all the structures necessary to model *both* **IL** and **MILL**, thus defining the doubly closed categories, that are categories equipped with two monoidal closed structures, with finite coproducts, and such that one of the closed structures is cartesian and the other is symmetric monoidal.

As observed in §1.7.2 for the models of Intuitionistic Logic, relational bi-monoids may seem a bit baroque and too elaborated, and they recall very closely the structure and the properties of the sequent calculus. Nevertheless, their definition is justified first by the

Table 1.9 Evaluation of Formulae in **LBI** Partially Ordered Monoids

$V(I)$	$\stackrel{\text{def}}{=}$	$\{1\}^{\rightarrow\leftarrow}$	$V(\top)$	$\stackrel{\text{def}}{=}$	M
$V(\varphi * \psi)$	$\stackrel{\text{def}}{=}$	$(V(\varphi) \times V(\psi))^{\rightarrow\leftarrow}$	$V(\varphi \wedge \psi)$	$\stackrel{\text{def}}{=}$	$V(\varphi) \cap V(\psi)$
$V(\varphi \mathrel{-\!\!*} \psi)$	$\stackrel{\text{def}}{=}$	$V(\varphi) \rightarrow V(\psi)$	$V(\varphi \rightarrow \psi)$	$\stackrel{\text{def}}{=}$	$V(\varphi) \Rightarrow V(\psi)$
$V(\varphi \vee \psi)$	$\stackrel{\text{def}}{=}$	$(V(\varphi) \cup V(\psi))^{\rightarrow\leftarrow}$	$V(\bot)$	$\stackrel{\text{def}}{=}$	$\emptyset^{\rightarrow\leftarrow}$

semantical cut elimination they provide, and then by the complete constructivism of the reasoning in all the proofs.

Again, as in §1.7.2, by releasing the requirement for a refined completeness theorem, we can obtain a more elegant model that is proved to be sound and complete for **LBI** by following a constructive path in the proof. The model is essentially obtained by merging the models of Intuitionistic Linear Logic, again the Gap saturated relational monoids of §1.7.1, and the simplified models for Intuitionistic Logic, the Low saturated preordered sets of §1.7.2. All the proofs have already been shown in the previous sections, our job here is just to re-read the results from **LBI**'s point of view. Hence the models suitable for the evaluation are those described below.

Definition 7 (LBI Partially Ordered Monoids). *The structure* $(M, \cdot, 1, \leq)$ *is a* **LBI** partially ordered monoid *if*

- $(M, \cdot, 1, \leq)$ *is a Gap saturated relational monoid.*
- $(M, \leq)$ *is a Low saturated partially ordered set.*

The two orthogonal structures provide two kinds of implication operators between subsets: $\rightarrow$, defined in (1.40) and associated to $\cdot$, and $\Rightarrow$, defined in (1.54) and associated to intersection between subsets. Then it is straightforward to adapt the evaluation function as outlined in Tab. 1.9. Soundness Theorem is a direct consequence of the reasonings in §1.7.1 and §1.7.2.

The syntactical model suitable for completeness is obtained by specialising Definition 6 according to what observed in §1.7.2. First we define the equivalence relation $\approx_{\mathbf{LBI}}$ between formulae, saying that $\varphi \approx_{\mathbf{LBI}} \psi$ means '$\varphi \vdash \psi$ *in* **LBI** *if and only if* $\psi \vdash \varphi$ *in* **LBI**. We denote by $[\varphi]$ the subset of the formulae that are equivalent to φ. Then the we get the structure $(Frm_{/\vdash}, *, [I], \leq_{\mathbf{LBI}})$, where

- $Frm_{/\mathbf{LBI}}$ is the set of the classes generated on Frm by the equivalence relation $\approx_{\mathbf{LBI}}$.
- $*$ is the generalisation to classes of the corresponding connective, defined as

$$[\varphi] * [\psi] \stackrel{\text{def}}{=} [\varphi * \psi] \quad \text{for } [\varphi], [\psi] \in Frm_{/\mathbf{LBI}}.$$

- $[I]$ is the class corresponding to the logical constant I.

- $\leq_{\mathbf{LBI}}$ is defined as $[\varphi] \leq_{\mathbf{LBI}} [\psi]$ if and only if $[\varphi \vdash \psi]$ is derivable in **LBI**.

It is easy to check that this model is well defined and that it satisfies Definition 7. Moreover the canonical evaluation lemma is still verified by using the same argumentations given in §1.7.2. Hence we can conclude that **LBI** partially ordered monoids are sound and complete for the Logic of Bunched Implications.

Theorem 17 (Soundness and Completeness for LBI Partially Ordered Monoids). *A sequent is valid in every* **LBI** *Partially Ordered Monoid if and only if it is provable in Bunched Implications Logic.*

LBI partially ordered monoids recall the *Kripke resource semantics* first presented in [110], and then refined in [122, 123], where the elements of the monoid are intended as resources and the monoidal operation as composition of resources. That semantics does not require for specific properties on monoids, but cannot be proved complete with respect to the constant $\bot$, the details are in [122, 123]. A similar semantics is presented in [70], and refined in [71], by Grothendieck topological monoids, that are preordered monoids equipped with a kind of closure operator to deal with $\bot$ and $\vee$, and that provide completeness for the whole calculus **LBI**. The topological model is used to further specify the Kripke resource semantics in order to obtain completeness for **LBI**, including $\bot$. The solution is to introduce a element π, absorbent for the product, i.e., $\pi \cdot m = \pi$ for every element of the monoids. This is related to the semantics we presented in this section as $\emptyset^{\rightarrow\leftarrow}$ satisfies the property required of π.

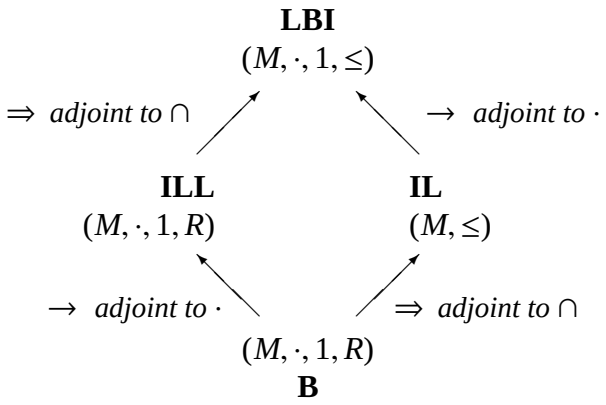
1.9 Semantical Diamond

The two semantics defined for **LBI** allow to compare Bunched Implications Logic with Intuitionistic Linear Logic and Intuitionistic Logic. Relational bi-monoids are obtained by *merging* the two semantics provided for **ILL** and **IL**. Then **LBI** is the *combination* of the two intuitionistic calculi, as already outlined by the bicartesian doubly closed categories.

More interesting is the interpretation offered by **LBI** partially ordered monoids. These monoids can be seen from two different points of view. On the one hand, a **LBI** partially ordered monoid is Gap saturated – as a model for **ILL** – that is required to be Low saturated. Hence the semantics for **LBI** is obtained by requiring the Low saturation property on the models for **ILL**, and this is equivalent to add an adjunction to the intersections between subsets. Thus **LBI** partially ordered monoids are a modular extension of the models for **ILL**. On the other hand, a **LBI** partially ordered monoid is Low saturated – as a model for **IL** – that is required to be Gap saturated. Hence the semantics for **LBI** can be obtained also by enriching the models for **IL** with a monoidal operation satisfying the Gap saturation property, and this is equivalent to define an operation on subset with a proper adjoint operation. Thus, in this case, **LBI** partially monoids are a modular extension of the models for **IL**.

We have just obtained the semantic diamond of Fig. 1.5 that outlines the semantical interrelations among **B**, **ILL**, **IL** and **LBI**. The basic calculus **B** is at the bottom, and

Figure 1.5 Semantical Diamond



its semantics is provided by relational monoids. Starting from **B**, and specialising the relation to be a preorder, we can obtain either **ILL**, by requiring an adjoint operator to the product, or **IL**, by requiring an adjoint operator to the intersection. Then **LBI** can be obtained either from **ILL** by requiring an operator adjoint to the product, or from **IL** by requiring an operator adjoint to the intersection. Hence Bunched Implication Logic can be modularly obtained, at least syntactically, either from Intuitionistic Logic – by following the left hand side of the diamond – or from Intuitionistic Linear Logic – by following the right hand side od the diamond. We conclude that, from a semantical point of view, **LBI** is a *modular extension* both of **ILL** and of **IL**. In particular, the Logic of Bunched Implications results in a proper extension of Intuitionistic Linear Logic, to which is required a new logical operator adjoint to the linear $\&$.

1.10 Towards Symmetric Logics

As [65] will fully show the relational semantics can be extended to a complete semantics for every symmetric logic obtained from Basic Logic, such as Linear Logic and Classical Logic (see [66, 129]). The relational monoids suitable to give a sound and complete semantics to the symmetric extensions are those with a symmetric relation. In this case the operators $(\,)^{\leftarrow}$ and $(\,)^{\rightarrow}$ coincide, and are dubbed $(\,)^{-}$.

Section 1.6 has already shown how to deal with structural rules. In particular, commutative monoids satisfying properties $(c1)$, $(c2)$, $(w1)$ and $(w2)$ are sound and (refined) complete models of Paraconsistent Quantum Logic [20, 58, 59].

A sound and complete semantics for (commutative) Orthologic[59, 76] is obtained by requiring the additional properties:

$$\text{For all } x_1, x_2, y \in M: \text{ if } x_1 \cdot x_2 \, R \, y \text{ then } x_1 \cdot y \cdot x_2 \, R \, 1.$$
$$\text{For all } x, y_1, y_2 \in M: \text{ if } x \, R \, y_1 \cdot y_2 \text{ then } 1 R \, y_1 \cdot y \cdot y_2.$$

And commutative monoids satisfying the same properties with the equivalence requirement

$$\text{For all } x_1, x_2, y \in M : x_1 \cdot x_2 \, R \, y \text{ if and only if } x_1 \cdot y \cdot x_2 \, R \, 1. \tag{1.64}$$

$$\text{For all } x, y_1, y_2 \in M : x \, R \, y_1 \cdot y_2 \text{ if and only if } 1 R \, y_1 \cdot y \cdot y_2 \tag{1.65}$$

provide a complete semantics for Linear Logic [74] without exponentials.

The properties (1.64) and (1.65) are equivalent to say that R is a *strongly symmetric relation,* which we define to be a relation satisfying the property

$$\text{for all } x, y, z \in M : \text{ if } x \cdot y \, R \, z \text{ then } x \cdot z \, R \, y \, . \tag{1.66}$$

Note that a strongly symmetric relation is symmetric as well; to see it choose $x = 1$ in (1.66). The adjective 'strongly' comes just from the fact that the relation turns out to be symmetric with respect to any element of the monoid, and not only the neutral element as for symmetric relations.

A set-theoretic semantics for linear Logic is provided by Girard's *phase spaces* [3, 74]. A phase space is a pair $(M, \bot)$, where M is a commutative monoid and $\bot$ is a subset of M on which no special requirement is assumed. For every $A \subseteq M$ the corresponding *orthogonal* subset is $A^\bot \stackrel{\text{def}}{=} \{x \in M \, : \, x \cdot y \, \epsilon \, \bot \text{ for every } y \, \epsilon \, A\}$.

Phase spaces can be seen as a particular case of relational monoids: they are just commutative relational monoids where the relation is strongly symmetric. In fact, the operator $(\,)^-$ for strongly symmetric monoids corresponds to the operator $(\,)^\bot$ of phase spaces, and it is easy to check the evaluation of formulae in phase spaces coincides with the evaluation defined in §1.4 when instantiated on strongly symmetric monoids. Such a result is formally proved in the proposition below.

Proposition 15. *Any phase space is a strongly symmetric monoid, and, conversely, any strongly symmetric monoid is a phase space.*

Proof. Let $(M, \bot)$ be a phase space. Then $(M, \cdot)$ is a commutative monoid and it becomes a strongly symmetric monoid $(M, \cdot, 1, R)$ if we define:

$$\text{for all } x, y \in M : \quad x \, R \, y \stackrel{\text{def}}{=} x \cdot y \, \epsilon \, \bot \, .$$

Obviously the relation R is strongly symmetric. Moreover the operators $(\,)^-$ and $(\,)^\bot$ coincide on subsets of M. In fact for any $A \subseteq M$:

$$\begin{aligned} A^- &\stackrel{\text{def}}{=} \{y \in M : x \, R \, y \text{ for all } x \, \epsilon \, A\} \\ &= \{y \in M : x \cdot y \, \epsilon \, \bot \text{ for all } x \, \epsilon \, A\} \stackrel{\text{def}}{=} A^\bot. \end{aligned}$$

In particular

$$\{1\}^- = \{y \in M : 1 \cdot y \, \epsilon \, \bot\} = \{y \in M : y \, \epsilon \, \bot\} = \bot \tag{1.67}$$

Vice versa, let $(M, \cdot, 1, R)$ be a strongly symmetric monoid; then it reduces to a phase space by defining $\bot \stackrel{\text{def}}{=} \{1\}^{-}$. In such way, for any $A \subseteq M$:

$$\begin{array}{rcll} A^{\bot} & \stackrel{\text{def}}{=} & \{y \in M : x \cdot y \,\epsilon\, \bot \text{ for all } x \,\epsilon\, A\} & \\ & = & \{y \in M : x \cdot y \,R\, 1 \text{ for all } x \,\epsilon\, A\} & \\ & = & \{y \in M : x \cdot 1 \,R\, y \text{ for all } x \,\epsilon\, A\} & \text{by (1.66)} \\ & = & \{y \in M : x \,R\, y \text{ for all } x \,\epsilon\, A\} \stackrel{\text{def}}{=} A^{-}. & \end{array}$$

□

The syntactical model suitable for completeness will have a symmetric relation, hence it is not the one presented here. For the sake of uniformity, the symmetric semantics will be fully detailed in [65].

1.11 Conclusions and Related Work

The original purpose of this work has been to fill the gap between the meta-theoretical interpretation of Basic Logic and a complete mathematical interpretation, which is commonly called semantics. Basic Logic was introduced in [129] by means of definitional equations, which are a perfect interpretation at the meta-level indeed, but to date there has been a notable lack of work on the corresponding mathematical semantics. The draft [77] presents a first attempt, that considers just the additive fragment of Basic Logic with structural rules, and provides a complete semantics based on a relational structure without any kind of binary operation. The approach in [77] uses Birkhoff's polarities to evaluate formulae, and has been the main inspiration for this work. Here we introduce a monoidal structure to evaluate the multiplicative fragment of the calculus, thus providing a semantics to the core of Basic Logic, by allowing the control of structural rules and contexts.

The relational monoid semantics has been defined by following the guidelines used for the basic calculus itself: the definitional equations [129, 128]. These equations are directly cast on the monoidal structures used to evaluate formulae. The idea of considering monoidal elements as resources is not new, and can be found in many other semantics, see for instance [74, 110]. In this chapter, the relation between resources has been intended as 'production,' but it can be seen also as an accessibility, sharing or dependency relation. Resources in [74] are related to their consumption, the work in [70, 71, 110, 122, 123] is focused on resource sharing and considers the monoidal operation as a separation operator.

As [129] did for sequent calculi, this work extends the semantics to well known logics, such as Paraconsistent Quantum Logic [20, 58, 59], Linear Logic [74], Intuitionistic Linear Logic [112, 126], and Intuitionistic Logic [94, 136]. Doing so, it provides a semantical link between Basic Logic and the other logics, and among the other logics in turn, since it extends the relational semantics by using its foundational principles. Surprisingly, the extensions are equivalent to the semantical counter part of the corresponding logics: phase

spaces [74], linear frames [137], pretopologies [127], Kripke semantics [14, 94, 136], and formal topologies [126].

The relationship between phase semantics and Galois connections has been first noted in [11], and, in connection with Linear Logic, the use of Birkhoff's polarities appears also in [69, 137]. In particular, the strongly symmetric monoids in §1.7.1 correspond to commutative reduced phasoids, see [137]. Moreover, a gap saturation property appears also in [137], where it is called 'continuity' for linear frames.

The completeness proof in §1.5 is done in such a 'refined' and sensible way that, besides proving completeness itself, it provides a semantical cut elimination theorem, that is inherited by all the extensions. The result enforces what has already been proved in [129] by a syntactical cut-elimination theorem for Basic Logic, modularly propagated to all the calculi obtained from Basic Logic.

Cut-elimination has been a prominent issue in Proof Theory since Gentzen's seminal work [72]. It represents the keystone for 'good' properties of deduction systems, such as disjunction, consistency and witness property in intuitionistic frameworks. In some case, notably in propositional settings, it allows also to prove decidability. In computer science, logic programming depends in a vital way on cut-elimination, since proof search is restricted to rules which are, at the very least, cut-free, cf. [96].

Most commonly, cut-elimination is proved syntactically: by verifying the termination of a cut-elimination algorithm, as in Gentzen's original proof [69, 73]. Another approach, known since Beth, Hintikka and others [135], is to prove the redundancy of cuts by proving the completeness of the cut-free calculus with respect to some notion of model [7, 118, 134]. This is the approach followed by the relational monoid semantics. It has been recently used in [112] to prove cut elimination of Intuitionistic Linear Logic, first and higher order; in [62] for Intuitionistic Higher-Order Logic; in [12] for a wide variety of sequent systems of nonclassical logics, both in propositional and predicate cases, including LK and LJ (cf. [72]); and in [83] for the intuitionistic sequent calculus LJ. The approach provided in [112] is the closest to relational monoids. In fact, in intuitionistic settings, the closure operator of [112] and the saturation operator $(\,)^{\rightarrow\leftarrow}$ exibit a very strong similarity. In particular, gap saturation provides a first order property for the corresponding closure properties, expressed at second order in [112]. Furthermore, the canonical model of §1.7.2 substantially corresponds to the canonical model proposed in [112], as can be seen by comparing the respective canonical evaluation lemmas.

2

Adding Places to Logic

In this chapter we study an intuitionistic, hybrid modal logic suitable for reasoning about distribution of resources. The modalities of the logic allow validation of properties in a *particular place*, in *some* place and in *all* places. We give a sound and complete Kripke semantics for the logic extended with disjunctive connectives. The extended logic can be seen as an instance of *Hybrid IS5*. We also give a sound and complete birelational semantics, and show that it enjoys the finite model property: if a judgement is not valid in the logic, then there is a finite birelational counter-model. Hence, we prove that the logic is decidable.

2.1 Introduction

In the current computing paradigm, distributed resources spread over and shared amongst different nodes of a computer system are very common. For example, printers may be shared in local area networks, or distributed data may store documents in parts at different locations. The traditional reasoning methodologies are not easily scalable to these systems as they may lack implicitly trust-able objects such as a central control.

This has resulted in the innovation of several reasoning techniques. A popular approach in the literature has been the use of algebraic systems such as process algebra [44, 104, 82]. These algebras have rich theories in terms of semantics [104], logics [43, 33, 81, 110], and types [82]. Another approach is logic-oriented [91, 92, 107, 106, 108, 124]: intuitionistic modal logics are used as foundations of type systems by exploiting the *propositions-as-types, proofs-as-programs* paradigm [75]. An instance of this was introduced in [91, 92]. The logic introduced there is the focus of our study. It uses the conjunctive connectives $\wedge$ and $\top$, and implication $\rightarrow$.

The formulae in this logic also include names, called *places*. Assertions in the logic are associated with places, and are validated in places. In addition to considering *whether* a formula is true, we are also interested in *where* a formula is true. In order to achieve this, the logic has three modalities. The modalities allow us to infer whether a property is validated in a specific place of the system ($@p$), or in an unspecified place of the system ($\Diamond$), or in any part of the system ($\Box$). The modality $@p$ internalises the model in the logic, and hence the logic can be classified as a hybrid logic [8, 9, 21, 22, 23, 32, 120, 121].

A natural deduction for the logic is given in [91, 92], and the judgements in the logic mention the places under consideration. The rules for $\Diamond$ and $\Box$ resemble those for existential and universal quantification of first-order intuitionistic logic. We extend the logic with disjunctive connectives, and extend the natural deduction system to account for these. The deduction system is essentially a conservative extension of propositional intuitionistic logic; and it is in this sense that we will use the adjective "intuitionistic" for the extended logic throughout the chapter.

As noted in [91, 92], the logic can also be used to reason about distribution of resources in addition to serving as the foundation of a type system. The papers [91, 92], however, lack a model to match the usage of the logic as a tool to reason about distributed resources. Here, we bridge the gap by presenting a Kripke-style semantics [94] for the logic extended with disjunctive connectives. In Kripke-style semantics, formulae are considered valid if they remain valid when the atoms mentioned in the formulae change their value from false to true. This is achieved by using a partially ordered set of *possible states*. Informally, more atoms are true in larger states.

We extend the Kripke semantics of the intuitionistic logic [94], enriching each possible state with a set of places. The set of places in Kripke states is not fixed, and different possible Kripke states may have *different* sets of places. However, the set of places vary in a conservative way: larger Kripke states contain larger set of places. In each possible state, different places satisfy different formulae. In the model, we interpret atomic formulae as resources of a distributed system, and placement of atoms in a possible state corresponds to the distribution of resources.

The enrichment of the model with places reveals the true meaning of the modalities in the logic. The modality $@p$ expresses a property in a named place. The modality $\Box$ corresponds to a weak form of spatial universal quantification and expresses a property common to all places, and the modality $\Diamond$ corresponds to a weak form of spatial existential quantification and expresses a property valid somewhere in the system. For the intuitionistic connectives, the satisfaction of formulae at a place in a possible state follows the standard definition [94].

To give semantics to a logical judgement, we allow models with more places than those mentioned in the judgement. This admits the possibility that a user may be aware of only a certain subset of names in a distributed system. This is crucial in the proof of soundness and completeness as it allows us to create witnesses for the existential ($\Diamond$) and the universal ($\Box$) modalities. The Kripke semantics reveals that the extended logic can be seen as the hybridisation of the well-known intuitionistic modal system *IS5* [63, 113, 117, 119, 67, 132].

Following [63, 117, 67, 132], we also introduce a sound and complete birelational semantics for the logic. The reason for introducing birelational semantics is that it allows us to prove decidability. Birelational semantics typically enjoy the *finite model property* [114, 132]: if a judgement is not provable, then there is a finite counter-model. On the other hand, Kripke semantics do not satisfy the finite model property [114, 132]. As in Kripke models, birelational models have a partially ordered set. The elements of this set are called *worlds*. In addition to the partial order, birelational models also have an

equivalence relation amongst worlds, called the *accessibility*, or *reachability*, relation. Unlike the Kripke semantics, we do not enrich each world with a set of places. Instead, we have a partial function, the *evaluation function*, which attaches a name to a world in its domain. As we shall see, the partiality of the function is crucial to the proof of decidability.

The partial evaluation function must satisfy two important properties. One, *coherence*, states that if the function associates a name to a world then it also associates the same name to all larger states. The other, *uniqueness*, states that two different worlds accessible from one another do not evaluate to the same name. Coherence is essential for ensuring monotonicity of the logical connective $@p$, and uniqueness is essential for the ensuring soundness of introduction of conjunction and implication.

Following [132], we also introduce an encoding of the Kripke models into birelational models. The encoding maps a place in a Kripke state into a world of the corresponding birelational model. The encoding ensures that if a formula is validated at a place in a state of the Kripke model, then it is also validated at the corresponding world. The encoding allows us to conclude soundness of Kripke semantics from soundness of birelational semantics. It also allows us to conclude completeness of the birelational models from completeness of Kripke semantics. We emphasise here that any birelational model resulting from the encoding is restricted in the sense that any two worlds reachable from each other are not related in the partial order. Therefore, the finite model property may fail for Kripke semantics even if it holds for birelational models. Birelational semantics gives us more models, and the fact that reachable worlds can be ordered is essential to achieve finite model property for birelational semantics, see §2.5.2 and [114, 132].

Surprisingly, the soundness of the birelational models was not straightforward. The problematic cases are the inference rules for introduction of $\Box$ and the elimination of $\Diamond$. In Kripke semantics, soundness is usually proved by duplicating places in a conservative way [32, 132]. The partiality of the evaluation function, along with the coherence and uniqueness conditions however impeded in obtaining such a result. It has been noted in [132] that the soundness is also non-trivial in the case of birelational models for Intuitionistic Modal Logic. However, the problems with soundness here arise purely because of the hybrid nature of the logic. Soundness is obtained by using a mathematical construction that creates a new birelational model from a given one. In the new model, the set of worlds consists of the reachability relation of the old model, and we add new worlds to witness the existential and universal properties.

The proof of completeness follows standard techniques from intuitionistic logics, and given a judgement that is not provable in the logic we construct a *canonical Kripke model* that invalidates the judgement. However, following [132], the construction of this model is done in a careful way so that it assists in the proof of decidability. The encoding of Kripke models into birelational models gives us a *canonical birelational model*. The worlds of canonical birelational models consists of triples: a finite set of places Q, a finite set of sentences Δ, and a special place q which is the evaluation of the world.

The set of worlds in the canonical birelational models may be infinite. We show that by identifying the worlds in the birelational model up-to renaming of places, we can

construct an equivalent finite model, called the *quotient model*. This allows us to deduce the finite model property for the birelational semantics, and hence decidabilty of the logic. The proof is adapted from the case of Intuitionistic Modal Logic [132]. The partiality of the evaluation function is crucial in the proof.

The rest of the Chapter is organised as follows. In §2.2, we introduce the logic and the Kripke semantics. In §2.5, we introduce the birelational semantics, and prove the soundness of the logic with respect to birelational models. The encoding of Kripke models into birelational models is also given and it allows us to conclude soundness of Kripke semantics. The construction of canonical models and completeness is discussed in §2.6. In §2.7, we construct the quotient model and prove the finite model property for birelational models. Related work is discussed in §2.8, and our results are summarised in §2.9.

2.2 The Logic

We now introduce, through examples, the logic presented in [91, 92] extended with disjunctive connectives, thus giving us the full set of intuitionistic connectives. The logic can be used to reason about heterogeneous distributed systems. To gain some intuition, consider a *distributed peer to peer database* where the information is partitioned over multiple communicating nodes (peers).

Informally, the database has a set of nodes, or *places*, and a set of resources (data) distributed amongst these places. The nodes are chosen from the elements of a fixed set, denoted by $p, q, r, s, \ldots$ Resources are represented by atomic formulae $A, B, \ldots \in \mathit{Atoms}$. Intuitively, an atom A is valid in a place p if that place can access the resource identified by A.

Were we reasoning about a particular place, the logical connectives of the intuitionistic framework would be sufficient. For example, assume that a particular document, doc, is partitioned in two parts, doc_1 and doc_2, and in order to gain access to the document a place has to access both of its parts. This can be formally expressed as the logical formula: $(\mathsf{doc}_1 \wedge \mathsf{doc}_2) \rightarrow \mathsf{doc}$, where $\wedge$ and $\rightarrow$ are the logical conjunction and implication. If doc_1 and doc_2 are stored in a particular place, then the usual intuitionistic rules allow to infer that the place can access the entire document.

The intuitionistic framework is extended in [92] to reason about different places. An assertion in such a logic takes the form "φ ***at*** p", meaning that formula φ is valid at place p. The construct "***at***" is a meta-linguistic symbol and points to the place where the reasoning is located. For example, doc_1 ***at*** p and doc_2 ***at*** p formalise the notion that the parts doc_1 and doc_2 are located at the node p. If, in addition, the assertion $((\mathsf{doc}_1 \wedge \mathsf{doc}_2) \rightarrow \mathsf{doc})$ ***at*** p is valid, we can conclude that the document doc is available at p.

The logic is a conservative extension of Intuitionistic Logic in the sense that if we restrict our attention to formulae without modalities then the 'local' proof system in a single place p mimics the standard intuitionistic one. For instance, the deduction described

above is formally

$$\dfrac{\dfrac{;\Delta \vdash^{\{p\}} \mathsf{doc}_1 \ \boldsymbol{at}\ p \quad ;\Delta \vdash^{\{p\}} \mathsf{doc}_2 \ \boldsymbol{at}\ p}{;\Delta \vdash^{\{p\}} \mathsf{doc}_1 \wedge \mathsf{doc}_2 \ \boldsymbol{at}\ p}\wedge I \qquad ;\Delta \vdash^{\{p\}} (\mathsf{doc}_1 \wedge \mathsf{doc}_2) \rightarrow \mathsf{doc} \ \boldsymbol{at}\ p}{;\Delta \vdash^{\{p\}} \mathsf{doc} \ \boldsymbol{at}\ p}\rightarrow E \tag{2.1}$$

where $\Delta \stackrel{\text{def}}{=} (\mathsf{doc}_1 \wedge \mathsf{doc}_2) \rightarrow \mathsf{doc} \ \boldsymbol{at}\ p, \mathsf{doc}_1 \ \boldsymbol{at}\ p, \mathsf{doc}_2 \ \boldsymbol{at}\ p$. It is easy to see that this derivation becomes a standard intuitionistic one if rewritten without the 'place' $\boldsymbol{at}\ p$.

In the assertion $\varphi \ \boldsymbol{at}\ p$, φ will not contain any occurrences of the construct $\boldsymbol{at}$. Instead, φ will use modalities $@p$, one for each place in the system, to cast the meta-linguistic $\boldsymbol{at}$ at the language level. A modality $@p$ internalises resources at the location p, and the modal formula $\varphi@p$ means that the property φ is valid at p, and not necessarily anywhere else. Indeed both $\varphi \ \boldsymbol{at}\ p$ and $\varphi@p$ will have the same semantics, and it is possible to define an equivalent logic in which the construct $\boldsymbol{at}$ is not needed. However, we will prefer to keep the distinction in the logic as was the case in [91, 92]. Also, the introduction and elimination rules for the modality @ are more elegant if we maintain this distinction. We need to keep track of where the reasoning is happening, and if we confuse $\boldsymbol{at}$ with @ then we will always need sentences of the form $\varphi@p$. In that case @-elimination could be applied only when the formula has two or more occurrences of @, namely only when it is of the form $\varphi@\ldots@p@q$.

An assertion of the form $\varphi@p \ \boldsymbol{at}\ p'$ means that we are located at the place p', and we are reasoning about the property φ that is validated at place p. For example, suppose that the place p has the first half of the document, i.e., $\mathsf{doc}_1 \ \boldsymbol{at}\ p$, and p' has the second one, i.e., $\mathsf{doc}_2 \ \boldsymbol{at}\ p'$. In the logic we can formalise the fact that p' can send the part doc_2 to p by using the assertion $(\mathsf{doc}_2 \rightarrow (\mathsf{doc}_2@p)) \ \boldsymbol{at}\ p'$. The rules of the logic will conclude $\mathsf{doc}_2 \ \boldsymbol{at}\ p$ and so $\mathsf{doc} \ \boldsymbol{at}\ p$. The formal derivation, (if we look ahead at the rules in Fig. 2.1), is

$$\dfrac{\dfrac{;\Delta \vdash^{\{p,p'\}} \mathsf{doc}_2 \ \boldsymbol{at}\ p' \quad ;\Delta \vdash^{\{p,p'\}} (\mathsf{doc}_2 \rightarrow (\mathsf{doc}_2@p)) \ \boldsymbol{at}\ p'}{;\Delta \vdash^{\{p,p'\}} (\mathsf{doc}_2@p) \ \boldsymbol{at}\ p'}\rightarrow E}{;\Delta \vdash^{\{p,p'\}} \mathsf{doc}_2 \ \boldsymbol{at}\ p}@E$$

Where $\Delta \stackrel{\text{def}}{=} \mathsf{doc}_2 \ \boldsymbol{at}\ p', (\mathsf{doc}_2 \rightarrow (\mathsf{doc}_2@p)) \ \boldsymbol{at}\ p'$. Moreover, $\mathsf{doc} \ \boldsymbol{at}\ p$ is derived by enriching Δ with the assumptions $\mathsf{doc}_1 \ \boldsymbol{at}\ p, (\mathsf{doc}_1 \wedge \mathsf{doc}_2) \rightarrow \mathsf{doc} \ \boldsymbol{at}\ p$, and by mimicking the derivation in (2.1).

The logic also has two other modalities to accommodate reasoning about properties valid at different locations, which we discuss briefly. Knowing exactly where a property holds is a strong ability, and we may only know that the property holds somewhere without knowing the specific location where it holds. To deal with this, the logic has the modality $\Diamond$: the formula $\Diamond\varphi$ means that φ holds in some place of the system. In the example above, the location of doc_2 is not important as long as we know that this document is located in some place from where it can be sent to p. Formally, this can be expressed by the logical formula $\Diamond(\mathsf{doc}_2 \wedge (\mathsf{doc}_2 \rightarrow (\mathsf{doc}_2@p))) \ \boldsymbol{at}\ p'$. By assuming this formula, we can infer

doc_2 ***at*** p, and hence the document doc is available at p. We will illustrate this inference at the end of the section (see Ex. 1).

Even if we deal with resources distributed in heterogeneous places, certain properties are valid everywhere. For this purpose, the logic has the modality $\Box$: the formula $\Box\varphi$ means that φ is valid everywhere. In the example above, p can access the document doc, if there is a place that has the part doc_2 and can send it everywhere. This can be expressed by the formula $\Diamond(\mathsf{doc}_2 \wedge (\mathsf{doc}_2 \rightarrow \Box\mathsf{doc}_2))$ ***at*** p'. The rules of the logic would allow us to conclude that doc_2 is available at p. Therefore the document doc is also available at p. We will illustrate this inference at the end of the section (see Ex. 2).

We now define formally the logic. As mentioned above, it is essentially the logic introduced in [92] enriched with the disjunctive connectives $\vee$ and $\bot$, thus achieving the full set of intuitionistic connectives. This allows us to express properties such as: the document doc_2 is located either at p itself or at q (in which case p has to fetch it). This can be expressed by the formula $(\mathsf{doc}_2 \vee (\mathsf{doc}_2@q \wedge (\mathsf{doc}_2@q \rightarrow \mathsf{doc}_2)))$ ***at*** p.

For the rest of the chapter, we shall assume a fixed countable set of atomic formulae *Atoms*, and we vary the set of places. Given a countable set of places *Pl*, let *Frm*(*Pl*) be the set of formulae built from the following grammar:

$$\varphi ::= A \mid \top \mid \bot \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi \mid \varphi@p \mid \Box\varphi \mid \Diamond\varphi.$$

Here the syntactic category p stands for elements from *Pl*, and the syntactic category A stands for elements from *Atoms*. The elements in *Frm*(*Pl*) are said to be *pure formulae*, and are denoted by small Greek letters $\varphi, \psi, \mu \ldots$ An assertion of the form φ ***at*** p is called a *sentence*. We denote by capital Greek letters $\Gamma, \Gamma_1, \ldots$ (possibly empty) finite sets of pure formulae, and by capital Greek letters $\Delta, \Delta_1, \ldots$ (possibly empty) finite sets of sentences.

Each judgement in this logic is of the form

$$\Gamma; \Delta \vdash^P \varphi \ \textbf{\textit{at}}\ p$$

where

- The *global context* Γ is a (possibly empty) finite set of pure formulae, and represents the properties assumed to hold at every place of the system.
- The *local context* Δ is a (possibly empty) finite set of sentences; since a sentence is a pure formula associated to a place, Δ represents what we assume to be valid in specific places.
- The sentence φ ***at*** p says that φ is derived to be valid in the place p by assuming $\Gamma; \Delta$.
- The set of places P represents the part of the system we are focusing on.

In the judgement, it is assumed that the places mentioned in Γ and Δ are drawn from the set P. More formally, if $\mathsf{PL}(X)$ denotes the set of places that appear in a syntactic object

Figure 2.1 Natural Deduction

$$\frac{}{\Gamma;\Delta,\varphi\ \mathbf{at}\ p \vdash^P \varphi\ \mathbf{at}\ p}\ L \qquad \frac{}{\Gamma,\varphi;\Delta \vdash^P \varphi\ \mathbf{at}\ p}\ G$$

$$\frac{}{\Gamma;\Delta \vdash^P \top\ \mathbf{at}\ p}\ \top I \qquad \frac{\Gamma;\Delta \vdash^P \bot\ \mathbf{at}\ p}{\Gamma;\Delta \vdash^P \psi\ \mathbf{at}\ p}\ \bot E$$

$$\frac{\Gamma;\Delta \vdash^P \varphi_i\ \mathbf{at}\ p}{\Gamma;\Delta \vdash^P \varphi_1 \vee \varphi_2\ \mathbf{at}\ p}\ \vee I_i\ \ (i=1,2)$$

$$\frac{\Gamma;\Delta \vdash^P \varphi_1 \vee \varphi_2\ \mathbf{at}\ p \quad \Gamma;\Delta,\varphi_1\ \mathbf{at}\ p \vdash^P \psi\ \mathbf{at}\ p \quad \Gamma;\Delta,\varphi_2\ \mathbf{at}\ p \vdash^P \psi\ \mathbf{at}\ p}{\Gamma;\Delta \vdash^P \psi\ \mathbf{at}\ p}\ \vee E$$

$$\frac{\Gamma;\Delta \vdash^P \varphi_i\ \mathbf{at}\ p \quad i=1,2}{\Gamma;\Delta \vdash^P \varphi_1 \wedge \varphi_2\ \mathbf{at}\ p}\ \wedge I \qquad \frac{\Gamma;\Delta \vdash^P \varphi_1 \wedge \varphi_2\ \mathbf{at}\ p}{\Gamma;\Delta \vdash^P \varphi_i\ \mathbf{at}\ p}\ \wedge E_i\ \ (i=1,2)$$

$$\frac{\Gamma;\Delta,\varphi\ \mathbf{at}\ p \vdash^P \psi\ \mathbf{at}\ p}{\Gamma;\Delta \vdash^P \varphi \to \psi\ \mathbf{at}\ p} \to I \qquad \frac{\Gamma;\Delta \vdash^P \varphi \to \psi\ \mathbf{at}\ p \quad \Gamma;\Delta \vdash^P \varphi\ \mathbf{at}\ p}{\Gamma;\Delta \vdash^P \psi\ \mathbf{at}\ p} \to E$$

$$\frac{\Gamma;\Delta \vdash^P \varphi\ \mathbf{at}\ p}{\Gamma;\Delta \vdash^P \varphi@p\ \mathbf{at}\ p'}\ @I \qquad \frac{\Gamma;\Delta \vdash^P \varphi@p\ \mathbf{at}\ p'}{\Gamma;\Delta \vdash^P \varphi\ \mathbf{at}\ p}\ @E$$

$$\frac{\Gamma;\Delta \vdash^{P+q} \varphi\ \mathbf{at}\ q}{\Gamma;\Delta \vdash^P \Box\varphi\ \mathbf{at}\ p}\ \Box I \qquad \frac{\Gamma;\Delta \vdash^P \Box\varphi\ \mathbf{at}\ p \quad \Gamma,\varphi;\Delta \vdash^P \psi\ \mathbf{at}\ p'}{\Gamma;\Delta \vdash^P \psi\ \mathbf{at}\ p'}\ \Box E$$

$$\frac{\Gamma;\Delta \vdash^P \varphi\ \mathbf{at}\ p}{\Gamma;\Delta \vdash^P \Diamond\varphi\ \mathbf{at}\ p'}\ \Diamond I \qquad \frac{\Gamma;\Delta \vdash^P \Diamond\varphi\ \mathbf{at}\ p' \quad \Gamma;\Delta,\varphi\ \mathbf{at}\ q \vdash^{P+q} \psi\ \mathbf{at}\ p''}{\Gamma;\Delta \vdash^P \psi\ \mathbf{at}\ p''}\ \Diamond E$$

X, then it must be the case that $\mathsf{PL}(\Gamma) \cup \mathsf{PL}(\Delta) \cup \mathsf{PL}(\varphi\ \mathbf{at}\ p) \subseteq P$. Any judgement not satisfying this condition is assumed to be undefined.

A natural deduction system without disjunctive connectives is given in [91, 92]. The natural deduction system with disjunctive connectives is given in Fig. 2.1. The most interesting rules are $\Diamond E$, the elimination of $\Diamond$, and $\Box I$, the introduction of $\Box$. In these rules, $P + p$ denotes the disjoint union $P \cup \{p\}$, and witnesses the fact that the place p occurs in neither Γ, nor Δ, nor φ, nor ψ. If $p \in P$, then $P + p$ is undefined, and any judgement containing such notation is assumed to be undefined in order to avoid a side condition stating this requirement.

The rule $\Diamond E$ explains how we can use formulae valid at some unspecified location: we

introduce a new place and extend the local context by assuming that the formula is valid there. If any assertion that does not mention the new place is validated thus, then it is also validated using the old local context. The rule $\Box I$ says that if a formula is validated in some new place, without any local assumption on that new place, then that formula must be valid everywhere.

The rules $\Diamond I$ and $\Box E$ are reminiscent of the introduction of the existential quantification, and the elimination of universal quantification in first-order intuitionistic logic. This analogy, however, has to be taken carefully. For example, if $\Gamma; \Delta \vdash^P \Diamond\psi \ \mathbf{at}\ p$, then we can show using the rules of the logic that $\Gamma; \Delta \vdash^P \Box\Diamond\psi \ \mathbf{at}\ p$. In other words, if a formula ψ is true at some unspecified place, then every place can deduce that there is some (unspecified) place where ψ is true.

Also note that, as stated, the rule $\bot E$ has a 'local' flavour: from $\bot \ \mathbf{at}\ p$, we can infer any other property in the same place, p. However, the rule has a 'global' consequence. If we have $\bot \ \mathbf{at}\ p$, then we can infer $\bot @q \ \mathbf{at}\ p$. Using $@E$, we can then infer $\bot \ \mathbf{at}\ q$. Hence, if a set of assumptions makes a place inconsistent, then it will make all places inconsistent.

As we shall see in §2.4, the Kripke semantics of this logic would be similar to the one given for intuitionistic system *IS5* [113, 119, 132]. Hence this logic can be seen as an instance of *Hybrid IS5* [32]. Before we proceed to define the Kripke semantics, we illustrate our derivation system by a couple of examples. The first example will demonstrate the use of rule $\Diamond E$ and $@E$, while the second example will demonstrate the use of $\Box E$.

Example 1. Let $p, p' \in P$ and ψ be the formula $\mathsf{doc}_2 \wedge (\mathsf{doc}_2 \to \mathsf{doc}_2@p)$.We can derive

$$; \Diamond\psi \ \mathbf{at}\ p' \vdash^P \mathsf{doc}_2 \ \mathbf{at}\ p$$

as follows:

$$\dfrac{\dfrac{}{; \Diamond\psi \ \mathbf{at}\ p' \vdash^P \Diamond\psi \ \mathbf{at}\ p'}\ L \qquad \begin{array}{c}\vdots\ \pi\\ ; \Diamond\psi \ \mathbf{at}\ p', \psi \ \mathbf{at}\ q \vdash^{P+q} \mathsf{doc}_2 \ \mathbf{at}\ p\end{array}}{; \Diamond\psi \ \mathbf{at}\ p' \vdash^P \mathsf{doc}_2 \ \mathbf{at}\ p}\ \Diamond E$$

where, given $q \notin P$ and $\Delta' \stackrel{\text{def}}{=} \Diamond\psi \ \mathbf{at}\ p', \psi \ \mathbf{at}\ q$, the derivation π is:

$$\dfrac{\dfrac{\dfrac{}{; \Delta' \vdash^{P+q} \psi \ \mathbf{at}\ q}\ L}{; \Delta' \vdash^{P+q} \mathsf{doc}_2 \ \mathbf{at}\ q}\ \wedge E \qquad \dfrac{\dfrac{}{; \Delta' \vdash^{P+q} \psi \ \mathbf{at}\ q}\ L}{; \Delta' \vdash^{P+q} \mathsf{doc}_2 \to \mathsf{doc}_2@p \ \mathbf{at}\ q}\ \wedge E}{; \Delta' \vdash^{P+q} \mathsf{doc}_2@p \ \mathbf{at}\ q}\ \to E$$
$$; \Delta' \vdash^{P+q} \mathsf{doc}_2 \ \mathbf{at}\ p\ @E$$

Example 2. Let $p, p' \in P$ and ψ be the formula $\mathsf{doc}_2 \wedge (\mathsf{doc}_2 \to \Box\mathsf{doc}_2)$. Pick $q \notin P$ and let $\Delta' \stackrel{\text{def}}{=} \Diamond\psi \ \mathbf{at}\ p', \psi \ \mathbf{at}\ q$. Just as in Example 1, we can derive

$$; \Diamond\psi \ \mathbf{at}\ p' \vdash^P \mathsf{doc}_2 \ \mathbf{at}\ p$$

as follows:

$$\dfrac{\dfrac{}{;\Diamond\psi \textbf{ at } p' \vdash^{P} \Diamond(\mathsf{doc}_2 \wedge (\mathsf{doc}_2 \to \Box\mathsf{doc}_2)) \textbf{ at } p'}\, L \qquad \begin{array}{c}\vdots\,\pi_1\\ ;\Delta' \vdash^{P+q} \mathsf{doc}_2 \textbf{ at } p\end{array}}{;\Diamond\psi \textbf{ at } p' \vdash^{P} \mathsf{doc}_2 \textbf{ at } p}\, \Diamond E$$

where π_1 is the derivation

$$\dfrac{\begin{array}{c}\vdots\,\pi_2\\ ;\Delta' \vdash^{P+q} \Box\mathsf{doc}_2 \textbf{ at } q\end{array} \qquad \dfrac{}{\mathsf{doc}_2;\Delta' \vdash^{P+q} \mathsf{doc}_2 \textbf{ at } p}\, G}{;\Delta' \vdash^{P+q} \mathsf{doc}_2 \textbf{ at } p}\, \Box E$$

where π_2 is similar to the proof π in 1:

$$\dfrac{\dfrac{\dfrac{}{;\Delta' \vdash^{P+q} \mathsf{doc}_2 \wedge (\mathsf{doc}_2 \to \Box\mathsf{doc}_2) \textbf{ at } q}\, L}{;\Delta' \vdash^{P+q} \mathsf{doc}_2 \textbf{ at } q}\, \wedge E \qquad \dfrac{\dfrac{}{;\Delta' \vdash^{P+q} \mathsf{doc}_2 \wedge (\mathsf{doc}_2 \to \Box\mathsf{doc}_2) \textbf{ at } q}\, L}{;\Delta' \vdash^{P+q} \mathsf{doc}_2 \to \Box\mathsf{doc}_2 \textbf{ at } q}\, \wedge E}{;\Delta' \vdash^{P+q} \Box\mathsf{doc}_2 \textbf{ at } q}\, \to E$$

2.3 Modal Proofs as Distributed Programs

The previous section showed how concisely the logic can express facts about the placement of resources in a system. Indeed, there is a more deep computational interpretation via the propositions-as-types, proofs-as-programs paradigm [75]. In fact, the logic introduced in §2.2 has been developed in [91, 92] as a new foundation for distributed programming languages, without considering the disjunctive connectives. There, the modal proofs are interpreted as distributed programs. More specifically, the proof terms for the various modalities have computational interpretations as *remote procedure calls*, commands to *broadcast* computations to all nodes in the network, commands to use *portable* code, and commands to invoke computational *agents* that can find their own way to safe places in the network where they can execute.

The work [92] introduces the proof terms of the logic, without $\vee$ or $\bot$, and shows how they may be given an operational interpretation as a distributed programming language, called λ_{rpc}. The logical formulae serve as types that prevent distributed programs from 'going wrong' by attempting to access resources that are unavailable at the place where the program is currently operating. Table 2.1 presents the syntax of programs and their types, and Fig. 2.2 presents the typing rules for the language, which are the natural deduction-style proof rules for the logic.

The *types* correspond to the formulae of the logic. The usage of the meta variable τ, rather than φ, indicates a shift in the interpretation. Moreover the syntax included a set of base types (**b**). Since [92] discovered two different operational interpretations of $\Box\varphi$, and it is worth explaining both of them in this section, Tab. 2.1 extends the language of formulae (types) to include an extra modality $\boxdot\tau$ to handle the second interpretation. To

Table 2.1 Syntax of λ_{rpc}

Types

$$\tau \quad ::= \quad \mathbf{b} \mid \top \mid \tau_1 \to \tau_2 \mid \tau_1 \wedge \tau_2 \mid \tau@p \mid \Box\tau \mid \ \boxdot\,\tau \mid \Diamond\tau$$

Proof Terms/Programs

e	::=	$c \mid x \mid \mathbf{sync}(x) \mid \mathbf{run}(x[p]) \mid ()$	Const / Var / $\top$
		$\lambda x{:}\tau.e \mid e_1 e_2$	Functions ($\to$)
		$\langle e_1, e_2\rangle \mid \pi_i e$	Pairs ($\wedge$)
		$\mathbf{ret}(e, p) \mid \mathbf{rpc}(e, p)$	Remote Procedure Calls (@)
		$\mathbf{close}(\lambda p.e) \mid \mathbf{bc}\ e_1\ \textbf{\textit{at}}\ p\ \mathbf{as}\ x\ \mathbf{in}\ e_2$	Broadcast ($\Box$)
		$\mathbf{port}(\lambda p.e) \mid \mathbf{pull}\ e_1\ \textbf{\textit{at}}\ p\ \mathbf{as}\ x\ \mathbf{in}\ e_2$	Portable ($\boxdot$)
		$\mathbf{agent}[e, p] \mid \mathbf{go}\ e_1\ \textbf{\textit{at}}\ p\ \mathbf{return}\ x, q\ \mathbf{in}\ e_2$	Agent ($\Diamond$)

support the two universal modalities, the logical global context Γ is separated into two parts, $\Gamma_\Box$ and $\Gamma_\boxdot$, during type checking. Hence the overall type checking judgment has the form:

$$\Gamma_\Box; \Gamma_\boxdot; \Delta \vdash^P e : \tau\ \textbf{\textit{at}}\ p.$$

By deleting either $\Box$ or $\boxdot$, and the associated context, we can recover exactly the same logic as in §2.2.

The *programs* include an unspecified set of constants (c), and the standard introduction and elimination forms for unit, functions and pairs. Variables from each different context are used in different ways. Some syntactic sugar has been added to the standard proof terms as a mnemonic for the different sorts of uses. Uses of local variables from Δ are just like ordinary uses of variables in a standard (call-by-value) functional language so they are left undecorated. Variables in $\Gamma_\Box$ refer to computations that have been broadcast at some earlier point. In order to use such a variable, the program must *synchronise* with the concurrently executing computation. Hence, we write $\mathbf{sync}(x)$ for such uses. Variables in $\Gamma_\boxdot$ refer to portable closures. Using a variable in this context means to *running* the closure with the current place p as an argument. Hence, we write $\mathbf{run}(x[p])$ for such uses.

The modality $\tau@p$ has an operational interpretation as a *remote procedure call*. The introduction form $\mathbf{ret}(e, p)$ constructs a 'return value' for a remote procedure call. This 'return value' can actually be an arbitrary expression e, which will be returned to and run at the place p. The elimination form $\mathbf{rpc}(e, p')$ is the remote procedure call itself. It sends the expression e to the remote site p' where e will be evaluated. If the expression is well typed, it will eventually evaluate to $\mathbf{ret}(e', p)$: a return value that can be run safely at the caller's place, which, in this case, is place p.

The introduction form for $\Box\tau$ is $\mathbf{close}(\lambda p.e)$. It creates a closure that may be *broadcast* by the elimination form $\mathbf{bc}\ e_1\ \textbf{\textit{at}}\ p_1\ \mathbf{as}\ x\ \mathbf{in}\ e_2$ to every node in the network. More specifically, the elimination form executes e_1 at p_1, expecting e_1 to evaluate to $\mathbf{close}(\lambda p.e)$. When it does, the broadcast expression chooses a new universal reference for the closure,

Figure 2.2 Typing Rules for λ_{rpc}

$$\frac{}{\Gamma_\Box; \Gamma_\boxdot; \Delta, x{:}\tau\ \textbf{\textit{at}}\ p \vdash^P x : \tau\ \textbf{\textit{at}}\ p}\ L$$

$$\frac{}{\Gamma_\Box, x{:}\tau; \Gamma_\boxdot; \Delta \vdash^P \textbf{sync}(x) : \tau\ \textbf{\textit{at}}\ p}\ G\Box \qquad \frac{}{\Gamma_\Box; \Gamma_\boxdot, x{:}\tau; \Delta \vdash^P \textbf{run}(x[p]) : \tau\ \textbf{\textit{at}}\ p}\ G\boxdot$$

$$\frac{}{\Gamma_\Box; \Gamma_\boxdot; \Delta \vdash^P () : \top\ \textbf{\textit{at}}\ p}\ \textit{Unit} \qquad \frac{}{\Gamma_\Box; \Gamma_\boxdot; \Delta \vdash^P c : \textbf{b}\ \textbf{\textit{at}}\ p}\ \textit{Const}$$

$$\frac{\Gamma_\Box; \Gamma_\boxdot; \Delta \vdash^P e_i : \tau_i\ \textbf{\textit{at}}\ p \quad i = 1,2}{\Gamma_\Box; \Gamma_\boxdot; \Delta \vdash^P \langle e_1, e_2\rangle : \tau_1 \wedge \tau_2\ \textbf{\textit{at}}\ p}\ \wedge I \qquad \frac{\Gamma_\Box; \Gamma_\boxdot; \Delta \vdash^P e : \tau_1 \wedge \tau_2\ \textbf{\textit{at}}\ p}{\Gamma_\Box; \Gamma_\boxdot; \Delta \vdash^P \pi_i e : \tau_i\ \textbf{\textit{at}}\ p}\ \wedge E_i\ {\scriptstyle (i=1,2)}$$

$$\frac{\Gamma_\Box; \Gamma_\boxdot; \Delta, x{:}\tau_1\ \textbf{\textit{at}}\ p \vdash^P e : \tau_2\ \textbf{\textit{at}}\ p}{\Gamma_\Box; \Gamma_\boxdot; \Delta \vdash^P \lambda x{:}\tau_1.e : \tau_1 \to \tau_2\ \textbf{\textit{at}}\ p}\ \to I$$

$$\frac{\Gamma_\Box; \Gamma_\boxdot; \Delta \vdash^P e_1 : \tau_1 \to \tau_2\ \textbf{\textit{at}}\ p \quad \Gamma; \Delta \vdash^P e_2 : \tau_1\ \textbf{\textit{at}}\ p}{\Gamma_\Box; \Gamma_\boxdot; \Delta \vdash e_1 e_2 : \tau_2\ \textbf{\textit{at}}\ p}\ \to E$$

$$\frac{\Gamma_\Box; \Gamma_\boxdot; \Delta \vdash^P e : \tau\ \textbf{\textit{at}}\ p}{\Gamma_\Box; \Gamma_\boxdot; \Delta \vdash^P \textbf{ret}(e, p) : \tau@p\ \textbf{\textit{at}}\ p'}\ @I \qquad \frac{\Gamma_\Box; \Gamma_\boxdot; \Delta \vdash^P e : \tau@p\ \textbf{\textit{at}}\ p'}{\Gamma_\Box; \Gamma_\boxdot; \Delta \vdash^P \textbf{rpc}(e, p') : \tau\ \textbf{\textit{at}}\ p}\ @E$$

$$\frac{\Gamma_\Box; \Gamma_\boxdot; \Delta \vdash^{P+q} e : \tau\ \textbf{\textit{at}}\ q}{\Gamma_\Box; \Gamma_\boxdot; \Delta \vdash^P \textbf{close}(\lambda p.e) : \Box\tau\ \textbf{\textit{at}}\ p}\ \Box I \qquad \frac{\Gamma_\Box; \Gamma_\boxdot; \Delta \vdash^{P+q} e : \tau\ \textbf{\textit{at}}\ q}{\Gamma_\Box; \Gamma_\boxdot; \Delta \vdash^P \textbf{agent}[e, p] : \boxdot\tau\ \textbf{\textit{at}}\ p}\ \boxdot I$$

$$\frac{\Gamma_\Box; \Gamma_\boxdot; \Delta \vdash^P e_1 : \Box\tau\ \textbf{\textit{at}}\ p \quad \Gamma_\Box, x{:}\tau; \Gamma_\boxdot; \Delta \vdash^P e_2 : \tau'\ \textbf{\textit{at}}\ p'}{\Gamma_\Box; \Gamma_\boxdot; \Delta \vdash^P \textbf{bc}\ e_1\ \textbf{\textit{at}}\ p\ \textbf{as}\ x\ \textbf{in}\ e_2 : \tau'\ \textbf{\textit{at}}\ p'}\ \Box E$$

$$\frac{\Gamma_\Box; \Gamma_\boxdot; \Delta \vdash^P e_1 : \boxdot\tau\ \textbf{\textit{at}}\ p \quad \Gamma_\Box; \Gamma_\boxdot, x{:}\tau; \Delta \vdash^P e_2 : \tau'\ \textbf{\textit{at}}\ p'}{\Gamma_\Box; \Gamma_\boxdot; \Delta \vdash^P \textbf{pull}\ e_1\ \textbf{\textit{at}}\ p\ \textbf{as}\ x\ \textbf{in}\ e_2 : \tau'\ \textbf{\textit{at}}\ p'}\ \boxdot E$$

$$\frac{\Gamma_\Box; \Gamma_\boxdot; \Delta \vdash^P e : \tau\ \textbf{\textit{at}}\ p}{\Gamma_\Box; \Gamma_\boxdot; \Delta \vdash^P \textbf{agent}[e, p] : \Diamond\tau\ \textbf{\textit{at}}\ p'}\ \Diamond I$$

$$\frac{\Gamma_\Box; \Gamma_\boxdot; \Delta \vdash^P e_1 : \Diamond\tau\ \textbf{\textit{at}}\ p' \quad \Gamma_\Box; \Gamma_\boxdot; \Delta, x{:}\tau\ \textbf{\textit{at}}\ q \vdash^{P+q} e_2 : \tau'\ \textbf{\textit{at}}\ p''}{\Gamma_\Box; \Gamma_\boxdot; \Delta \vdash^P \textbf{go}\ e_1\ \textbf{\textit{at}}\ p'\ \textbf{return}\ x, p\ \textbf{in}\ e_2 : \tau'\ \textbf{\textit{at}}\ p''}\ \Diamond E$$

which is bound to x, and sends $\lambda p.e$ to every place in the network where it is applied to the current place and the resulting expression is associated with its universal reference. Finally, expression e_2 is executed with the universal reference bound to x. Remote procedure calls or broadcasts generated during evaluation of e_2 may refer to the universal reference bound to x, which is safe, since x has been broadcast everywhere.

Objects of type $\boxdot\tau$ are portable closures; they may be *run anywhere*. The elimination form **pull** e_1 **at** p_1 **as** x **in** e_2 takes advantage of this portability by first computing e_1 at p_1, which should result in a value with the form **port**$(\lambda p.e)$. Next, it pulls the closure $\lambda p.e$ from p_1 and substitutes it for x in e_2. The typing rules will allow x to appear anywhere,

including in closures in e_2 that will eventually be broadcast or remotely executed. Once again, this is safe since e is portable and runs equally well everywhere.

The connective $\Diamond\tau$ represents the type of a *computational agent* that knows where it can go to produce a value with type τ. Such an agent is introduced by packaging an expression with a place where the expression may successfully be run to completion. The elimination form **go** e_1 **at** p_1 **return** x, p **in** e_2 first evaluates e_1 at p_1, producing an agent **agent**$[e, p_2]$. Next, it commands the agent to go to the hidden place p_2 and execute its encapsulated computation there. When the agent has completed its task, it synchronises with the current computation and e_2 continues with p bound to p_2 and x bound to the value that is safe to use at p_2.

Example 3. To gain a little more intuition about how to write programs in this language, we consider the computational interpretations of some of the proofs from §2.2. Consider the set of places P containing p, q, p' and q, and a context Δ containing the following assumptions:

dI_p	:	doc_1 ***at*** p	doc_1 is located at p
dII_p	:	doc_2 ***at*** p	doc_2 is located at p
$\mathsf{dII}_{p'}$	:	doc_2 ***at*** p'	doc_2 is located at p'
paste	:	$(\mathsf{doc}_1 \wedge \mathsf{doc}_2) \to \mathsf{doc}$ ***at*** p	pasting together the two parts at p
P′toP	:	$\mathsf{doc}_2 \to (\mathsf{doc}_2@p)$ ***at*** p'	sending doc_2 from p' to p
toP	:	$\Box(\mathsf{doc}_2 \to (\mathsf{doc}_2@p))$ ***at*** p	broadcasting the request to send doc_2 to p

Creating the whole document doc, involving local computation only:

$$;\Delta \vdash^P \mathsf{paste}(\langle \mathsf{dI}_p, \mathsf{dII}_p \rangle) : \mathsf{doc}\ \textbf{\textit{at}}\ p.$$

Fetching doc_2, involving a remote procedure call in which the computation $\mathsf{P'toP}(\mathsf{dII}_{p'})$ is executed at p':

$$;\Delta \vdash^P \textbf{rpc}(\mathsf{P'toP}(\mathsf{dII}_{p'}), p') : \mathsf{doc}_2\ \textbf{\textit{at}}\ p.$$

Fetching then pasting together:

$$;\Delta \vdash^P (\lambda x{:}\mathsf{doc}_2.\mathsf{paste}(\langle \mathsf{dI}_p, x \rangle))(\textbf{rpc}(\mathsf{P'toP}(\mathsf{dII}_{p'}), p')) : \mathsf{doc}\ \textbf{\textit{at}}\ p.$$

Broadcasting p's request to all nodes, then fetching the second half of the document from node q (recall that in general, uses of these global variables involves synchronising with the broadcast expression; below the broadcast expression is a value, but we synchronise anyway):

$$;\Delta, \mathsf{dII}_q : \mathsf{doc}_2\ \textbf{\textit{at}}\ q \vdash^P \textbf{bc}\ \mathsf{toP}\ \textbf{\textit{at}}\ p\ \textbf{as}\ \mathsf{toP'}\ \textbf{in rpc}(\textbf{sync}(\mathsf{toP'})\mathsf{dII}_q, q) : \mathsf{doc}_2\ \textbf{\textit{at}}\ p.$$

Another way to manage a part of the document is to make it portable. For instance, if q contains the second part of the required document, then p can pull it from its resident location. Remember that portable values are polymorphic closures that are 'run' when used. In this case, the closure simply returns the appropriate part of the document.

$$;\Delta, \mathsf{d}_q : \boxdot\mathsf{doc}_2\ \textbf{\textit{at}}\ q \vdash^P \textbf{pull}\ \mathsf{d}_q\ \textbf{\textit{at}}\ q\ \textbf{as}\ x\ \textbf{in run}(x[p]) : \mathsf{doc}_2\ \textbf{\textit{at}}\ p.$$

Table 2.2 Run-Time Syntax of λ_{rpc}

Networks

$\mathfrak{N} \quad ::= \quad (P, \mathfrak{L})$

Process Environments

$\mathfrak{L} \quad ::= \quad \cdot \mid \mathfrak{L}, l \to e \ \textbf{\textit{at}}\ p$

Values

$v \quad ::= \quad c \mid \lambda x{:}\tau.e \mid \langle v_1, v_2 \rangle \mid \textbf{ret}(e, p) \mid \textbf{close}(\lambda p.e) \mid \textbf{port}(\lambda p.e) \mid \textbf{agent}[e, p]$

Run-Time Terms

$e \quad ::= \quad \ldots \mid \textbf{sync}(l) \mid \textbf{run}(\lambda p.e[p_1]) \mid \textbf{sync}(\textbf{rpc}(l, p))$
$\textbf{sync}(\textbf{bc}\ l\ \textbf{\textit{at}}\ p\ \textbf{as}\ x\ \textbf{in}\ e_2) \mid \textbf{sync}(\textbf{pull}\ l\ \textbf{\textit{at}}\ p\ \textbf{as}\ x\ \textbf{in}\ e_2)$
$\textbf{sync}_1(\textbf{go}\ l\ \textbf{\textit{at}}\ p\ \textbf{return}\ x, q\ \textbf{in}\ e) \mid \textbf{sync}_2(\textbf{go}\ l\ \textbf{\textit{at}}\ p\ \textbf{return}\ x, q\ \textbf{in}\ e)$

Evaluation Contexts

$C \quad ::= \quad [\,] \mid C\,e \mid v\,C \mid \langle C, e \rangle \mid \langle v, C \rangle \mid \pi_i\,C$

2.3.1 Operational Semantics and Safety

To distinguish between the two very different interpretations of $\Box$, papers [91, 92] give an operational semantics at a lower level of abstraction than proof reduction by including an explicit, concrete network in the semantics as shown in Tab. 2.3. Nevertheless, the basis for the semantics is the interaction of introduction and elimination rules as the proof theory suggests. The various new syntactic objects used to specify the operational model are listed in Tab. 2.2.

Networks $\mathfrak{N}$ are pairs consisting of a set of places P, and a distributed process environment $\mathfrak{L}$. Places have been discussed before. The process environment $\mathfrak{L}$ is a *finite partial map* from places p in P to process IDs to expressions. These partial maps are written as lists of elements with the form $l \to e\ \textbf{\textit{at}}\ p$. Papers [91, 92] assume that no pair of place and location (p and l) appears in two different components of the map. They do not distinguish between maps that differ only in the ordering of their elements. The application $\mathfrak{L}(p)(l)$ denotes e when $\mathfrak{L} = \mathfrak{L}_0, l \to e\ \textbf{\textit{at}}\ p$.

Run-time terms, newly introduced in Tab. 2.2, only occur at run time to give an operation semantics to the program. These terms are used to represent expressions, which are suspended part-way through evaluation and are waiting to synchronise with remotely executing expressions. Finally the evaluation contexts C specify the order of evaluation.

In order to show that the network is well-typed at every step in evaluation, [91] adds typing rules to give types to the run-time terms and it also give well-formedness conditions for the network as a whole. The typing judgment for a network has the form

$$\vdash \mathfrak{L} : \Gamma_{\Box}; \cdot; \Delta.$$

As this section is only meant to grant an intuition of the computational interpretation of

Table 2.3 Operational Semantics of λ_{rpc} ($\mathfrak{L} \mapsto \mathfrak{L}'$)

sync *OS*	$\mathfrak{L}, l' \to C[\mathbf{sync}(l)]$ ***at*** $p, l \to v$ ***at*** p $\mapsto \mathfrak{L}, l' \to C[v]$ ***at*** $p, l \to v$ ***at*** p
run *OS*	$\mathfrak{L}, l \to C[\mathbf{run}(\lambda p.e[p_1])]$ ***at*** p_2 $\mapsto \mathfrak{L}, l \to C[e[p_1/p]]$ ***at*** p_2
$\to$ *OS*	$\mathfrak{L}, l \to C[(\lambda x{:}\tau.e)v]$ ***at*** p $\mapsto \mathfrak{L}, l \to C[e[v/x]]$ ***at*** p
$\wedge$ *OS*	$\mathfrak{L}, l \to C[\pi_i\langle v_1, v_2\rangle]$ ***at*** p $\mapsto \mathfrak{L}, l \to C[v_i]$ ***at*** p
@ *OS1*	$\mathfrak{L}, l \to C[\mathbf{rpc}(e, p_1)]$ ***at*** p_0 $\mapsto \mathfrak{L}, l \to C[\mathbf{sync}(\mathbf{rpc}(l_1, p_1))]$ ***at*** $p_0, l_1 \to e$ ***at*** p_1
@ *OS2*	$\mathfrak{L}, l \to C[\mathbf{sync}(\mathbf{rpc}(l_1, p_1))]$ ***at*** $p_0, l_1 \to \mathbf{ret}(e, p_0)$ ***at*** p_1 $\mapsto \mathfrak{L}, l \to C[e]$ ***at*** $p_0, l_1 \to \mathbf{ret}(e, p_0)$ ***at*** p_1
$\Box$ *OS1*	$\mathfrak{L}, l \to C[\mathbf{bc}\ e_1$ ***at*** p_1 **as** x **in** $e_2]$ ***at*** p_0 $\mapsto \mathfrak{L}, l \to C[\mathbf{sync}(\mathbf{bc}\ l_1$ ***at*** p_1 **as** x **in** $e_2)], l_1 \to e_1$ ***at*** p_1
$\Box$ *OS2*	$\mathfrak{L}, l \to C[\mathbf{sync}(\mathbf{bc}\ l_1$ ***at*** p_1 **as** x **in** $e_2)]$ ***at*** $p_0, l_1 \to \mathbf{close}(\lambda p.e)$ ***at*** p_1 $\mapsto \mathfrak{L}, l \to C[e_2[l_2/x]]$ ***at*** $p_0, l_1 \to \mathbf{close}(\lambda p.e)$ ***at*** $p_1, \{l_2 \to e[q/p]$ ***at*** $q\}_{q\in P}$
$\boxdot$ *OS1*	$\mathfrak{L}, l \to C[\mathbf{pull}\ e_1$ ***at*** p_1 **as** x **in** $e_2]$ ***at*** p_0 $\mapsto \mathfrak{L}, l \to C[\mathbf{sync}(\mathbf{pull}\ l_1$ ***at*** p_1 **as** x **in** $e_2)]$ ***at*** $p_0, l_1 \to e_1$ ***at*** p_1
$\boxdot$ *OS2*	$\mathfrak{L}, l \to C[\mathbf{sync}(\mathbf{pull}\ l_1$ ***at*** p_1 **as** x **in** $e_2)]$ ***at*** $p_0, l_1 \to \mathbf{port}(\lambda p.e)$ ***at*** p_1 $\mapsto \mathfrak{L}, l \to C[e_2[\lambda p.e/x]]$ ***at*** $p_0, l_1 \to \mathbf{port}(\lambda p.e)$ ***at*** p_1
$\Diamond$ *OS1*	$\mathfrak{L}, l \to C[\mathbf{g}\ e_1$ ***at*** p_1 **return** x, q **in** $e_2]$ ***at*** p_0 $\mapsto \mathfrak{L}, l \to C[\mathbf{sync}_1(\mathbf{go}\ l_1$ ***at*** p_1 **return** x, q **in** $e_2)]$ ***at*** $p_0, l_1 \to e_1$ ***at*** p_1
$\Diamond$ *OS2*	$\mathfrak{L}, l \to C[\mathbf{sync}_1(\mathbf{go}\ l_1$ ***at*** p_1 **return** x, q **in** $e_2)]$ ***at*** $p_0, l_1 \to \mathbf{agent}[e, p_2]$ ***at*** p_1 $\mapsto \mathfrak{L}, l \to C[\mathbf{sync}_2(\mathbf{go}\ l_2$ ***at*** p_2 **return** x, q **in** $e_2)]$ ***at*** p_0, $l_1 \to \mathbf{agent}[e, p_2]$ ***at*** $p_1, l_2 \to e$ ***at*** p_2
$\Diamond$ *OS3*	$\mathfrak{L}, l \to C[\mathbf{sync}_2(\mathbf{g}\ l_1$ ***at*** p_1 **return** x, q **in** $e_2)]$ ***at*** $p_0, l_1 \to v$ ***at*** p_1 $\mapsto \mathfrak{L}, l \to C[e_2[p_1/q][v/x]]$ ***at*** $p_0, l_1 \to v$ ***at*** p_1

the logic, and it does not represent the original contribution of this Thesis, we refer to [91] for further details.

The state of a network $\mathfrak{N} = (P, \mathfrak{L})$ evolves according to the operational rules listed in Tab. 2.3. These rules specify a relation with the form $\mathfrak{L} \mapsto \mathfrak{L}'$. The type system is sound with respect to such an operational semantics. The proofs of Preservation and Progress theorems, stated below, follow the usual strategy (see [91]).

Theorem 18 (Preservation). *If* $\vdash \mathfrak{L} : \Gamma_{\Box}; \cdot; \Delta$ *and* $\mathfrak{L} \mapsto \mathfrak{L}'$*, then there exists* $\Gamma'_{\Box}$ *and* Δ' *such that* $\vdash \mathfrak{L} : \Gamma'_{\Box}; \cdot; \Delta'$.

Theorem 19 (Progress). *If* $\vdash \mathfrak{L} : \Gamma_{\Box}; \cdot; \Delta$ *then either* $\mathfrak{L} \mapsto \mathfrak{L}'$*, or* $\mathfrak{L}(p)(l)$ *is a value, for all places* p *in* P*, and for all* l *in the domain of* $\mathfrak{L}(p)$.

The lambda calculi presented in this section gives an operational view of the logic, but to date there has been a notable lack of work on the corresponding semantics. Next sections address precisely this issue and provide the theoretical foundations for applications of such a logic.

2.4 Kripke Semantics

There are a number of semantics for intuitionistic logic and intuitionistic modal logics that allow for a completeness theorem [32, 93, 132, 63, 67, 113, 117]. In this section, we concentrate on the semantics introduced by Kripke [94, 138], as it is convenient for applications and fairly simple. This would provide a formalisation of the intuitive concepts introduced above.

In Kripke semantics for Intuitionistic propositional Logic, logical assertions are interpreted over Kripke models. The validity of an assertion depends on its behaviour as the truth values of its atoms change from false to true according to a Kripke model. A Kripke model consists of a *partially ordered* set of *Kripke states*, and an *interpretation*, I, that maps atoms into states. The interpretation tells which atoms are true in a state. It is required that if an atom is true in a state, then it must remain true in all larger states. Hence, in a larger state more atoms may become true. Consider a logical assertion built from the atoms $A_1, \ldots, A_n$. The assertion is said to be valid in a state if it continues to remain valid in all larger states.

In order to express the full power of the logic introduced above, we need to enrich the model by introducing places. We achieve this by associating a set of places P_k to each Kripke state k. The formulae of the logic are validated in these places. The interpretation is indexed by the Kripke states, and the interpretation I_k maps atoms into the set P_k. Since we consider atoms to be resources, the map I_k tells how resources are distributed in the Kripke state k.

In the case of intuitionistic propositional logic, an atom validated in a Kripke state is validated in all larger states. In order to achieve the corresponding thing, we shall require that all places appearing in a Kripke state appear in every larger state. Furthermore, we require that if I_k maps an atom into a place, then I_l should map the atom in the same place for all states l larger than k. In terms of resources, it means that places in larger states have possibly more resources.

The Kripke models that we shall define now are similar to those defined for the intuitionistic modal system *IS5* [63, 67, 113, 117, 32, 132]. In the definition, K is the set of Kripke states, and its elements are denoted by $k, l, \ldots$ The relation $\leq$ is the partial order on the set of states.

Definition 8 (Kripke Model). A quadruple $\mathcal{K} = (K, \leq, \{P_k\}_{k\in K}, \{I_k\}_{k\in K})$ is a *Kripke model* if

- K is a (non empty) set;

- $\leq$ is a partial order on K;
- P_k is a *non-empty* set of places for all $k \in K$;
- $P_k \subseteq P_l$ if $k \leq l$;
- $I_k : Atoms \to Pow(P_k)$ is such that $I_k(A) \subseteq I_l(A)$ for all $k \leq l$.

Let $Pls = \bigcup_{k \in K} P_k$. We shall say that Pls is the set of places of $\mathcal{K}$.

The definition tells only how resources, i.e. atoms, are distributed in the system. To give semantics to the whole set of formulae $Frm(Pls)$, we need to extend I_k. The interpretation of a formula depends on its composite parts, and if it is valid in a place in a given state, then it remains valid at the same place in all larger states. For example, the formula $\varphi \wedge \psi$ is valid in a state k at place $p \in P_k$, if both φ and ψ are true at place p in all states $l \geq k$.

The introduction of places in the model allows the interpretation of the spatial modalities of the logic. Formula $\varphi@p$ is satisfied at a place in a state k, if it is true at p in all states $l \geq k$; $\Diamond\varphi$ and $\Box\varphi$ are satisfied at a place in state k, if φ is true respectively at some or at every place in all states $l \geq k$.

We extend now the interpretation of atoms to interpretation of formulae by using induction on the structure of the formulae. The interpretation of formulae is similar to that used for Modal Intuitionistic Logic [63, 67, 113, 117, 32, 132].

Definition 9 (Semantics). Let $\mathcal{K} = (K, \leq, \{P_k\}_{k \in K}, \{I_k\}_{k \in K})$ be a Kripke model with set of places Pls. Given $k \in K$, $p \in P_k$, and a pure formula φ with $\mathsf{PL}(\varphi) \subseteq Pls$, we define $(k, p) \models \varphi$ inductively as:

$$
\begin{array}{lcll}
(k,p) & \models & A & \text{iff } p \in I_k(A); \\
(k,p) & \models & \top & \text{iff } p \in P_k; \\
(k,p) & \models & \bot & \text{never;} \\
(k,p) & \models & \varphi \wedge \psi & \text{iff } (k,p) \models \varphi \text{ and } (k,p) \models \psi; \\
(k,p) & \models & \varphi \vee \psi & \text{iff } (k,p) \models \varphi \text{ or } (k,p) \models \psi; \\
(k,p) & \models & \varphi \to \psi & \text{iff } (l \geq k \text{ and } (l,p) \models \varphi) \text{ implies } (l,p) \models \psi; \\
(k,p) & \models & \varphi@q & \text{iff } q \in P_k \text{ and } (k,q) \models \varphi; \\
(k,p) & \models & \Box\varphi & \text{iff } (l \geq k \text{ and } q \in P_l) \text{ implies } (l,q) \models \varphi; \\
(k,p) & \models & \Diamond\varphi & \text{iff there exists } q \in P_k \text{ such that } (k,q) \models \varphi.
\end{array}
$$

We pronounce $(k, p) \models \varphi$ as *'(k, p) forces φ'*, or *'(k, p) satisfies φ'*. We write $k \models \varphi$ ***at*** p if $(k, p) \models \varphi$.

It is clear from the definition that if $k \models \varphi$ ***at*** p, then $\mathsf{PL}(\varphi$ ***at*** $p) \subseteq P_k$. Please note that in this extension, except for logical implication and the modality $\Box$, we have not considered larger states in order to interpret a modality or a connective. It turns out that the satisfaction of a formula in a state implies the satisfaction in all larger states, as stated in the following proposition.

Proposition 16 (Kripke Monotonicity). *Let $\mathcal{K} = (K, \leq, \{P_k\}_{k\in K}, \{I_k\}_{k\in K})$ be a Kripke model with set of places Pls. The relation $\models$ preserves the partial order on K,* i.e.*, for each $k, l \in K$, $p \in P_k$, and $\varphi \in Frm(P_k)$, if $l \geq k$ then $(k, p) \models \varphi$ implies $(l, p) \models \varphi$.*

Proof. Standard, by induction on the structure of formulae. □

Consider now the distributed database described before. We can express the same properties inferred in §2.2 by using a Kripke model. Fix a Kripke state k. The assumption that the two parts, $\mathsf{doc}_1, \mathsf{doc}_2$, can be combined in p in a state k to give the document doc can be expressed as $(k, p) \models (\mathsf{doc}_1 \wedge \mathsf{doc}_2) \rightarrow \mathsf{doc}$. If the resources doc_1 and doc_2 are assigned to the place p, i.e., $(k, p) \models \mathsf{doc}_1$ and $(k, p) \models \mathsf{doc}_2$, then, since $(k, p) \models \mathsf{doc}_1 \wedge \mathsf{doc}_2$, it follows that $(k, p) \models \mathsf{doc}$.

Let us consider a slightly more complex situation. Suppose that $k \models \Diamond(\, \mathsf{doc}_2 \wedge (\mathsf{doc}_2 \rightarrow \Box\mathsf{doc}_2)\,)$ ***at*** p'. According to the semantics of $\Diamond$, there is some place r such that $(k, r) \models \mathsf{doc}_2 \wedge (\mathsf{doc}_2 \rightarrow \Box\mathsf{doc}_2)$. The semantics of $\wedge$ tells us that $(k, r) \models \mathsf{doc}_2$ and $(k, r) \models (\mathsf{doc}_2 \rightarrow \Box\mathsf{doc}_2)$. Since $(k, r) \models \mathsf{doc}_2$, we know from the semantics of $\rightarrow$ that $(k, r) \models \Box\mathsf{doc}_2$, and from the semantics of $\Box$ that $(k, p) \models \mathsf{doc}_2$. Therefore, if doc_1 is placed at p in the state k, then the whole document doc would become available at place p in state k.

To give semantics to the judgements of the logic, we need to extend the definition of forcing relation to judgements. We begin by extending the definition to contexts.

Definition 10 (Forcing on Contexts). Let $\mathcal{K} = (K, \leq, \{P_k\}_{k\in K}, \{I_k\}_{k\in K})$ be a distributed Kripke model. Given a state k in K, a finite set of pure formulae Γ, and a finite set of sentences Δ such that $\mathsf{PL}(\Gamma; \Delta) \subseteq P_k$; we say that k forces the context $\Gamma; \Delta$ (and we write $k \models \Gamma; \Delta$) if

1. for every $\varphi \in \Gamma$ and every $p \in P_k$: $(k, p) \models \Box\varphi$;
2. for every ψ ***at*** $q \in \Delta$: $(k, q) \models \psi$.

Finally, we extend the definition of forcing to judgements.

Definition 11 (Satisfaction for a Judgment). Let $\mathcal{K} = (K, \leq, \{P_k\}_{k\in K}, \{I_k\}_{k\in K})$ be a Kripke model. The judgement $\Gamma; \Delta \vdash^P \mu$ **at** *p is said to be valid in $\mathcal{K}$* if

- $\mathsf{PL}(\Gamma) \cup \mathsf{PL}(\Delta) \cup \mathsf{PL}(\mu) \cup \{p\} \subseteq P$;
- for every $k \in K$ such that $P \subseteq P_k$, if $k \models \Gamma; \Delta$ then $(k, p) \models \mu$.

Moreover, we say that $\Gamma; \Delta \vdash^P \mu$ **at** *p is valid* (and we write $\Gamma; \Delta \models \mu$ ***at*** p) if it is valid in every Kripke model.

Although, it is possible to obtain soundness and completeness of Kripke semantics directly, we shall not do so in this chapter. Instead, they will be derived as corollaries. Soundness will follow from the soundness of birelational semantics and encoding of Kripke models into birelational models. Completeness will emerge as a corollary in the proof of construction of finite counter-model.

2.5 Birelational Models

One other semantics given for modal intuitionistic logics in literature is birelational semantics [63, 67, 117, 132]. As in the case of Intuitionistic Modal Logics [114, 132], birelational semantics for our logic enjoys the finite model property, while Kripke semantics does not.

Birelational models, like Kripke models, have a set of partially ordered states. The partially ordered states will be called *worlds*, and we use $u, v, w, \ldots$ to range over them. Formulae will be validated in worlds, and if a formula is validated in a world, then it will be validated in all larger worlds. To validate atoms we have the interpretation I, which maps atoms into a subset of worlds. If I maps an atom into a world, then it will map the atom in all larger worlds.

In addition to the partial order, however, there is also a second binary relation on the set of states which is called *reachability* or *accessibility* relation. Intuitively, uRw means that w will be reachable from u. As our logic is a hybridisation for *IS5*, the relation R will be an equivalence relation. The relation R will also satisfy a technical requirement, the *reachability condition*, that is necessary to ensure monotonicity and soundness of logic evaluation.

Unlike the Kripke semantics, the states will not have a set of places associated to them. Instead, there is a *partial* function, *Eval*, which maps a world to a *single* place. In a sense which we will make precise in §2.5.2, a world in a birelational model corresponds to a place in a specific Kripke state. As we shall see later, the partiality of the function *Eval* is crucial in the proof of the finite model property. In the case $Eval(w)$ is defined and is p, we shall say that *w evaluates* to p.

In addition to partiality, *Eval* will also satisfy two other properties: *coherence* and *uniqueness*. Coherence says that if a world evaluates to p, then all larger worlds evaluate to p. Together with the reachability condition, coherence will ensure the monotonicity of the modality @. Uniqueness will say that no two worlds reachable from each other can evaluate to the same place. Uniqueness will be essential for the soundness of introduction of conjunction ($\wedge I$), and of implication ($\rightarrow I$). The formal definition of the models is below.

Definition 12 (Birelational Model). Given a set of places *Pls*, a *birelational model on Pls* is a quintuple $\mathcal{W}_{Pls} = (W, \leq, R, I, Eval)$, where

1. W is a (non empty) set, ranged over by $v, v', w, w', \ldots$.
2. $\leq$ is a *partial* order on W.
3. $R \subseteq W \times W$ is an *equivalence relation* and satisfies the *reachability condition*:

$$\textit{if } w' \geq w\, R\, v \textit{ then there exists } v' \textit{ such that } w'\, R\, v' \geq v;$$

4. $I : Atoms \rightarrow Pow(W)$ is such that if $w \in I(A)$ then $w' \in I(A)$ for all $w' \geq w$.

5. $Eval : W \to Pls$ is a *partial* function. We write $v\uparrow$ if $Eval(v)$ is not defined, $v\downarrow$ if $Eval(v)$ is defined, and $v\downarrow p$ if $Eval(v)$ is defined and equal to p.

   Moreover, the following properties hold:

   (a) *coherence:* for any $v \in W$, if $v\downarrow p$ then $w\downarrow p$ for every $w \geq v$;

   (b) *uniqueness:* for every $v \in W$ such that $v\downarrow p$, if $v\,R\,v'$ and $v'\downarrow p$, then $v = v'$.

In addition to the reachability condition, usually there is another similar condition in birelational models for intuitionistic modal logics [63, 67, 117, 132]:

$$\textit{if } w\,R\,v \leq v' \textit{ then there exists } w' \textit{such that } w \leq w'\,R\,v'.$$

In this case, as R is an equivalence relation, the property is an immediate consequence of the reachability condition.

As for Kripke models, the interpretation of atoms extends to formulae. A formula $\varphi@p$ is true in a world w, if there is a reachable world which evaluates to p and where φ is valid. A formula $\Diamond\varphi$ is valid in a world w, if there is a reachable world (not necessarily in the domain of $Eval$) where φ is valid. A formula $\Box\varphi$ is valid in a world w if φ is valid in all worlds reachable from worlds w' larger than w.

Definition 13 (Bi-Forcing Semantics). Let $\mathcal{W}_{Pls} = (W, \leq, R, I, Eval)$ be a birelational model on Pls. Given $w \in W$, and a pure formula $\varphi \in Frm(Pls)$, we define the forcing relation $w \Vdash \varphi$ inductively as follows:

$w \Vdash A$	iff	$w \in I(A)$;
$w \Vdash \top$		for all $w \in W$;
$w \Vdash \bot$		never;
$w \Vdash \varphi \wedge \psi$	iff	$w \Vdash \varphi$ and $w \Vdash \psi$;
$w \Vdash \varphi \vee \psi$	iff	$w \Vdash \varphi$ or $w \Vdash \psi$;
$w \Vdash \varphi \to \psi$	iff	($v \geq w$ and $v \Vdash \varphi$) implies $v \Vdash \psi$;
$w \Vdash \varphi@q$	iff	there exists v such that $w\,R\,v$, $v\downarrow q$ and $v \Vdash \varphi$;
$w \Vdash \Box\varphi$	iff	($v \geq w$ and $v\,R\,v'$) implies $v' \Vdash \varphi$;
$w \Vdash \Diamond\varphi$	iff	there exists $v \in W$ such that wRv and $v \Vdash \varphi$.

We pronounce $w \Vdash \varphi$ as 'w forces φ,' or '*w satisfies φ*.'

As for Kripke models, this relation is monotone.

Proposition 17 (Monotonicity). *Let $\mathcal{W}_{Pls}$ be a birelational model on Pls. The relation $\Vdash$ preserves the partial order in W, namely, for every world w in W and $\varphi \in Frm(Pls)$, if $v \geq w$ then $w \Vdash \varphi$ implies $v \Vdash \varphi$.*

Proof. The proof is straightforward, and proceeds by induction on the structure of formulae. Here, we just consider the induction step in which φ is of the form $\varphi_1@p$. Suppose that $w \Vdash \varphi_1@p$. Then there is a w' such that $w\,R\,w'$, $w'\downarrow p$ and $w' \Vdash \varphi_1$.

Consider now $v \geq w$. Since $w\,R\,w'$, by the reachability condition we obtain that there is a world v' such that $v\,R\,v'$ and $v' \geq w'$. As $w' \models \varphi_1$, by induction hypothesis we obtain $v' \models \varphi_1$. Now, as $v' \geq w'$ and $w'{\downarrow}\,p$, we get $v'{\downarrow}\,p$ by coherence property. Finally, as $v\,R\,v'$, we get $v \models \varphi_1@p$ by definition. □

Example 4. Consider the birelational model $\mathcal{W}_{exam}$ with two worlds, say w_1 and w_2. We take $w_1 \leq w_2$, and both worlds are reachable from each other. The world w_2 evaluates to p, while the evaluation of w_1 is undefined. Let A be an atom. We define $I(A)$ to be the singleton $\{w_2\}$. For any formula φ, we abbreviate $\varphi \to \bot$ as $\neg\varphi$.

Consider the pure formula $\neg A$. Now, by definition, $w_2 \models A$ and therefore $w_2 \not\models \neg A$. Also, as $w_1 \leq w_2$, we get $w_1 \not\models \neg A$. This means that $w_2 \models \neg\neg A$, and $w_1 \models \neg\neg A$. Hence, we get $w_1, w_2 \models \Box\neg\neg A$.

On the other hand, consider the formula $\neg\neg\Box A$. We have by definition that $w_1 \not\models A$. As w_1 is reachable from both w_1 and w_2, we deduce that $w_1, w_2 \not\models \Box A$. Using the semantics of $\to$, we get that $w_1, w_2 \not\models \neg\neg\Box A$.

We now extend the semantics to the judgements of the logic. We begin by extending the semantics to contexts.

Definition 14 (Bi-Forcing on Contexts). Let $\mathcal{W}_{Pls} = (W, \leq, R, I, Eval)$ be a birelational model on *Pls*. Given a finite set of pure formulae Γ, and a finite set of sentences Δ, such that $\mathsf{PL}(\Gamma; \Delta) \subseteq Pls$; we say that $w \in W$ forces the context $\Gamma; \Delta$ (and we write $w \models \Gamma; \Delta$) if

1. for every $\varphi \in \Gamma$: $w \models \Box\varphi$, and
2. for every ψ ***at*** $q \in \Delta$: $w \models \psi@q$.

In order to extend the semantics to judgements, we need one more definition. We say that a place p is reachable from a world v, if there is a world which evaluates to p and is reachable from v. The set of all places reachable from a world v will be denoted by $Reach(v)$. More formally,

$$Reach(v) \stackrel{\text{def}}{=} \{p \ : w{\downarrow}\,p \text{ for some } w \in W, v\,R\,w\}$$

It can be easily shown by using the reachability condition and coherence that if $v \leq w$, then every place reachable from v is also reachable from w.

Proposition 18 (Reachability). *Given any birelational model, then:*

1. *If $v \leq w$, then $Reach(v) \subseteq Reach(w)$.*
2. *If $v\,R\,w$, then $Reach(v) = Reach(w)$.*

We are now ready to extend the satisfaction to judgements.

Definition 15 (Bi-Satisfaction for Judgments). The sequent $\Gamma; \Delta \vdash^P \varphi$ ***at*** p is said to be valid in the birelational model $\mathcal{W}_{Pls} = (W, \leq, R, I, Eval)$ if:

- $PL(\Gamma) \cup PL(\Delta) \cup \{p\} \subseteq P$;
- for any $w \in W$ such that $P \subseteq Reach(w)$: $w \models \Gamma; \Delta$ implies $w \models \varphi@p$.

Moreover, we say that $\Gamma; \Delta \vdash^P \mu$ **at** p *is bi-valid* (and we write $\Gamma; \Delta \models^P \mu$ ***at*** p) if it is valid in every birelational model.

Example 5. Consider the birelational model $\mathcal{W}_{exam}$ on two worlds w_1 and w_2 discussed in Ex. 4. We had $w_1, w_2 \models \Box\neg\neg A$ and $w_1, w_2 \not\models \neg\neg\Box A$. Therefore, the judgement $; \vdash^{\{p\}} \Box\neg\neg A$ ***at*** p is bi-valid in the model $\mathcal{W}_{exam}$, while the judgement $; \Box\neg\neg A$ ***at*** $p \vdash^{\{p\}} \neg\neg\Box A$ ***at*** p is not bi-valid in $\mathcal{W}_{exam}$. In fact, we will later on show that the judgement $; \Box\neg\neg A$ ***at*** $p \vdash^{\{p\}} \neg\neg\Box A$ ***at*** p is valid in every finite Kripke model. Therefore, this example, adapted from [114, 132], will demonstrate that the finite model property does not hold in the case of Kripke semantics.

2.5.1 Soundness

The proof of soundness of birelational models has several subtleties, that arise as a consequence of the inference rules for the introduction of $\Box$ ($\Box$ I), and elimination of $\Diamond$ ($\Diamond$ E). Let us illustrate this for the case of $\Box I$. Recall the inference rule of $\Box I$ from Fig. 2.1:

$$\frac{\Gamma; \Delta \vdash^{P+q} \varphi \textbf{ at } q}{\Gamma; \Delta \vdash^{P} \Box\varphi \textbf{ at } p}\ \Box I$$

To show the soundness of this rule, we must show that the judgement $\Gamma; \Delta \vdash^P \Box\varphi$ ***at*** p is bi-valid whenever the judgement $\Gamma; \Delta \vdash^{P+q} \varphi$ ***at*** q is bi-valid. Now, to show that the judgement $\Gamma; \Delta \vdash^P \Box\varphi$ ***at*** p is bi-valid, we must consider an arbitrary world, say w, in an arbitrary birelational model, say $\mathcal{W}_{Pls}$, such that $P \subseteq Reach(w)$ and $w \models \Gamma; \Delta$. We need to prove that $w \models \Box\varphi@p$ also. For this, we need to show that for any world v in $\mathcal{W}_{Pls}$ such that $w \leq w' \, R \, v$ for some w', it is the case that $v \models \varphi$. Pick one such v and fix it.

Please note that without loss of generality, we can assume that *Pls* does not contain q (otherwise, we can always rename q in the model). To use the hypothesis that $\Gamma; \Delta \vdash^{P+q} \varphi$ ***at*** q is bi-valid, we must consider a modification of $\mathcal{W}_{Pls}$. One strategy, that is adopted in the case of Kripke semantics [32], is to add new worlds v'_q, one for each world $v' \geq v$. The new worlds v'_q duplicate v' in all respects except that they evaluate to q. If the resulting construction yields a birelational model, then $Reach(v'_q)$ would contain P as well as q.

The next step would be to show that any formula ψ, that does not refer to the place q, is satisfied by v'_q if and only if it is satisfied by v'. Using this, that v'_q forces the context $\Gamma; \Delta$ in the new model also. Then, we can use the hypothesis to obtain that v'_q satisfies $\varphi@q$. Since v'_q evaluates to q, we will get that v'_q forces φ. As φ does not refer to q, we will get that v' forces φ. We can then conclude the proof by observing that $v \geq v$, and choosing v' to be v.

In fact, if the world v was in the domain of *Eval*, then the above outline would have worked. However, this breaks down in case $v\uparrow$. To illustrate this, suppose that there is a

world v' such that $v \leq v'$, $v'\uparrow$ and $v\,R\,v'$. In the construction of the extension, we would thus have two worlds v_q and v'_q reachable from each other, that evaluate to the same place q, which would violate the uniqueness condition.

This breakdown is fatal for the proof and cannot be fixed. Coherence demands that $v'_q\downarrow q$ if $v_q\downarrow q$. So, we cannot fiddle with the evaluation. We cannot even relax uniqueness as this will be needed for soundness of introduction of conjunction ($\wedge$ I) and of implication ($\rightarrow$ I). Furthermore, we cannot require that the evaluation is a total function: it is the partiality of this function that gives us the finite model property. Indeed, if the function was total, the class of birelational models would be equivalent to the class of Kripke models, and we would have not gained anything by using birelational models.

Our strategy to prove soundness is to construct a birelational model from $\mathcal{W}_{Pls}$, called q-extension, whose worlds are the union of two sets. The first one of these sets is the reachability relation R of $\mathcal{W}_{Pls}$. The second one will be the Cartesian product $\{q\} \times W$, where W is the set of worlds of $\mathcal{W}_{Pls}$. Hence, the worlds of the q-extension are ordered pairs. A world (w', w) will evaluate to the same place as w', and (q, w) will evaluate to q. Two worlds will be reachable from each other only if they agree in the second entry.

The construction would guarantee (see Lemma 9) that given $\psi \in Frm(Pls)$, the world (w', w) satisfies ψ if and only if w' does, and the world (q, w) satisfies ψ if and only if w does. The proof of soundness of $\Box I$ would work as follows. Let v be a fixed world. Consider the world (q, v) in the q-extension. We will show that v satisfies $\Gamma; \Delta$, and hence (q, v) satisfies $\Gamma; \Delta$. The set of reachable places from (q, v) contains P as well as q , and we can thus conclude that (q, v) satisfies $\varphi@q$. Since (q, v) evaluates to q, we conclude that (q, v) satisfies φ. As mentioned above, this is equivalent to saying that v satisfies φ.

We are ready to carry out this proof formally. We begin by constructing the q-extension, and showing that this is a birelational model.

Lemma 8 (q-Extension). *Let $\mathcal{W}_{Pls} = (W, \leq, R, I, Eval)$ be a birelational model on Pls. Given a new place $q \notin Pls$, we define the q-extension $\mathcal{W}\langle q\rangle_{Pls'}$ to be the quintuple $(W', \leq', R', I', Eval')$, where*

1. $Pls' \stackrel{\text{def}}{=} Pls \cup \{q\}$.

2. $W' \stackrel{\text{def}}{=} R \cup (\{q\} \times W)$.

3. $\leq' \subseteq W' \times W'$ *is defined as:*

   - $(w', w) \leq' (v', v)$ *if and only if* $w' \leq v'$ *and* $w \leq v$,
   - $(q, w) \leq' (q, v)$ *if and only if* $w \leq v$;

4. $R' \subseteq W' \times W'$ *is defined as:*

   - $(a, b)\,R'\,(c, d)$ *if and only if* $b = d$, *for* $(a, b), (c, d) \in W'$.

5. $I' : Atoms \rightarrow Pow(W')$ *is defined as:*

- $I'(A) \stackrel{\text{def}}{=} \{ (w', w) \mid w' \in I(A),\ w' R w \} \cup \{ (q, w) \mid w \in I(A) \}$;

6. *$Eval' : W' \to Pls'$ is defined as*

   - $Eval'((w', w)) \stackrel{\text{def}}{=} Eval(w')$ *for every* $(w', w) \in R$,[1]
   - $Eval'((q, w)) \stackrel{\text{def}}{=} q$ *for every* $w \in W$.

The q-extension is a birelational model.

Proof. We need to show the five properties of Definition 12.

1. Clearly W' is a non empty set if W is.

2. Since $\leq$ is a partial order, then $\leq'$ is a partial order too.

3. The relation R' is an equivalence by definition. We show that R' satisfies the reachability condition by cases. There are four possible cases.

Case a. Assume that $(v', v) \geq' (w', w)\, R'\, (w'', w)$.
The hypothesis says that $v \geq w$, $v' \geq w'$, $v' R v$, $w' R w$ and $w'' R w$. Since R is an equivalence, we get $v' \geq w' R w''$. Using the reachability condition for R, there exists $v'' \in W$ such that $v' R v'' \geq w''$. Hence, we conclude $(v', v)\, R'\, (v'', v) \geq (w'', w)$.

Case b. Assume that $(q, v) \geq' (q, w)\, R'\, (w', w)$.
This means that $v \geq w$ and $w R w'$. By the reachability condition for R, there is a v' such that $v R v' \geq w'$, and we conclude $(q, v)\, R'\, (v', v) \geq' (w', w)$.

Case c. Assume that $(v', v) \geq' (w', w)\, R'\, (q, w)$.
This means $v \geq w$, and we conclude $(v', v)\, R'\, (q, v) \geq' (q, w)$.

Case d. Assume that $(q, v) \geq' (q, w)\, R'\, (q, w)$.
We have $v \geq w$, and we conclude $(q, v)\, R'\, (q, v) \geq' (q, w)$.

4. To check monotonicity for I', we consider two cases:

Case a. Assume that $(w', w) \in I'(A)$.
This means that $w' \in I(A)$. If $(v', v) \geq' (w', w)$, then $v' \geq w'$. By the monotonicity of I, we get $v' \in I(A)$. Hence $(v', v) \in I'(A)$.

Case b. Assume that $(q, w) \in I(A)$.
This means that $w \in I(A)$. If $(q, v) \geq' (q, w)$, then $v \geq w$. By the monotonicity of I, we get $v \in I(A)$. Hence $(q, v) \in I'(A)$.

5. According to the definition, $Eval'$ is a partial function. We need to verify the two properties required for a birelational model.

[1] In the equality, the left hand side is defined only if the right hand side is.

Coherence. We have to show that if a world in the new model evaluates to some place, then all the higher worlds evaluate to the same place. There are two possible cases.

Case a. Assume that $(v', v) \geq' (w', w)$, and $(w', w)\downarrow p$
We get by definition, $v' \geq w'$ and $w'\downarrow p$. By coherence on the model $\mathcal{W}_{Pls}$, we get $v'\downarrow p$. Hence $(v', v)\downarrow p$.

Case b. Assume that $(q, v) \geq' (q, w)$.
We have by definition, $(q, v)\downarrow q$ and $(q, w)\downarrow q$.

Uniqueness. We have to show that two different worlds reachable from each other cannot evaluate to the same place. As (q, v) always evaluates to q, two worlds (w, v) and (q, w) cannot evaluate to the same place. There are two other possible cases.

Case a. Suppose $(v', v)\, R'(w', w)$, $(w', w)\downarrow p$ and $(v', v)\downarrow p$.
We have by definition $v'\, R\, v$, $w'\, R\, w$, $v = w$, $w'\downarrow p$ and $v'\downarrow p$. Since R is an equivalence and $v = w$, we get $v'\, R\, w'$. By uniqueness on $\mathcal{W}_{Pls}$, we get $v' = w'$. Therefore $(v', v) =' (w', w)$

Case b. Suppose that $(q, v)\, R'(q, w)$, $(q, w)\downarrow q$ and $(q, v)\downarrow q$.
We have by definition $v = w$, and hence $(q, v) = (q, w)$. □

We will now show that if a pure formula, say ψ, does not mention q, then (w', w) satisfies ψ only if w' does. Furthermore, (q, w) satisfies ψ only if w does.

Lemma 9 ($\mathcal{W}\langle u, q\rangle_{Pls'}$ Is Conservative). *Let $\mathcal{W}_{Pls} = (W, \leq, R, I, Eval)$ be a birelational model, and let $\mathcal{W}\langle q\rangle_{Pls'} = (W', \leq', R', I', Eval')$ be its q-extension. Let $\models$ and $\models'$ extend the interpretation of atoms in $\mathcal{W}_{Pls}$ and $\mathcal{W}\langle q\rangle_{Pls'}$ respectively. For every $\varphi \in Frm(Pls)$ and $w \in W$, it holds*

1. *for every $w'\, R\, w$, $(w', w) \models'\varphi$ if and only if $w' \models \varphi$; and*
2. *$(q, w) \models'\varphi$ if and only if $w \models \varphi$.*

Proof. We prove both the points simultaneously by induction on the structure of formulae in $Frm(Pls)$.

Base of induction. The two points are verified on atoms, on $\top$, and on $\bot$ by definition.

Induction hypothesis. We consider a formula $\varphi \in Frm(Pls)$, and assume that the two points hold for all sub-formulae φ_i of φ. In particular, we assume that *for every $w \in W$:*

1. *for every $w'\, R\, w$, $(w', w) \models'\varphi_i$ if and only if $w' \models \varphi_i$; and*
2. *$(q, w) \models'\varphi_i$ if and only if $w \models \varphi_i$.*

We shall prove the lemma only for the modal connectives and for the logical connective $\rightarrow$. The other cases can be treated similarly. We shall also only consider point *1*, as the treatment of point *2* is analogous. We pick $w \in W$ and $w'\, R\, w$, and fix them.

- *Case* $\varphi = \varphi_1 \to \varphi_2$. Suppose $(w', w) \models' \varphi_1 \to \varphi_2$. Then

$$\text{for every } (v', v) \geq' (w', w), \text{ we have } (v', v) \models' \varphi_1 \text{ implies } (v', v) \models' \varphi_2. \tag{2.2}$$

  We need to show that $w' \models \varphi$. Pick $v' \geq w'$ such that $v' \models \varphi_1$, and fix it. It suffices to show that $v' \models \varphi_2$.

  We have $v' \geq w' R w$. By the reachability condition, there exists $v \in W$ such that $v' R v \geq w$. Hence, $(v', v) \geq' (w', w)$.

  The induction hypothesis says that $(v', v) \models' \varphi_1$. We have $(v', v) \models' \varphi_2$ by (2.2) above. Hence $v' \models \varphi_2$, by applying induction hypothesis one more time.

  For the other direction, assume that $w' \models \varphi_1 \to \varphi_2$. Then

$$\text{for every } v' \geq w', \text{ we have } v' \models \varphi_1 \text{ implies } v' \models \varphi_2. \tag{2.3}$$

  Now consider $(v', v) \geq' (w', w)$, and assume $(v', v) \models' \varphi_1$. From $(v', v) \geq' (w', w)$, we have $v' \geq w'$. From $(v', v) \models' \varphi_1$ and induction hypothesis, we have $v' \models \varphi_1$. Since $v' \geq w'$, we get from (2.3) above, $v' \models \varphi_2$. Therefore $(v', v) \models' \varphi_2$, by induction hypothesis once again. We conclude by definition that $(v', v) \models' \varphi_1 \to \varphi_2$.

- *Case* $\varphi = \varphi_1@p$. Since $\varphi_1@p \in \mathit{Frm}(\mathit{Pls})$, we have $p \neq q$.

  $(w', w) \models' \varphi_1@p$ is equivalent to saying that there is a world $(v', w) \in W'$ such that: $(v', w)\ R'\ (w', w)$, $(v', w)\downarrow p$, and $(v', w) \models' \varphi_1$.

  By induction hypothesis and definition of q-extension, this is equivalent to say that there exists $v' \in W$ such that: $v' R w$, $v'\downarrow p$, and $v' \models \varphi_1$. This is equivalent to say that $w \models \varphi_1@p$ by definition.

- *Case* $\varphi = \Diamond\varphi_1$.

  Suppose $(w', w) \models' \Diamond\varphi_1$. Then there is a world in W' such that this world is reachable from (w', w), and which satisfies φ_1. There are two possibilities for this world: it can be of the form (v, w), or of the form (q, w).

  If it is of the form (v, w), then by definition we have $v R w$. Since R is an equivalence and $w R w'$, we have $v R w'$. Furthermore, since $(v, w) \models' \varphi$, we get by induction hypothesis $v \models \varphi_1$. Therefore, $w' \models \Diamond\varphi_1$ by definition.

  If the world is of the form (q, w), then by induction hypothesis, $w \models \varphi_1$. Since $w' R w$, we get $w' \models \Diamond\varphi_1$.

  For the other direction, if $w' \models \Diamond\varphi_1$ then there exists $v R w'$ such that $v \models \varphi_1$. Since R is an equivalence, we have $v R w$. Hence (v, w) is a world of the q-extension, and $(v, w) \models' \varphi_1$ by induction hypothesis. Since $(v, w)\, R'\, (w', w)$, we conclude $(w', w) \models' \Diamond\varphi_1$.

- *Case* $\varphi = \Box\varphi_1$. Suppose that $(w', w) \models' \Box\varphi_1$. This means that φ_1 is forced by every world reachable from some world larger that (w', w). In particular, we have that

$$\text{for every } (v', v) \geq (w', w), \text{ if } (v'', v)\, R'\, (v', v) \text{ then } (v'', v) \models' \varphi_1. \tag{2.4}$$

  We need to show that $w' \models \Box\varphi_1$. Pick v', v'' such that $v' \geq w'$, and $v'' \,R\, v'$, and fix them. It suffices to show that $v'' \models \varphi_1$.

  Since $v' \geq w'$ and $w' \,R\, w$, the reachability condition for R says that there exists $v \in W$ such that $v' \,R\, v \geq w$. By transitivity, we have $v'' \,R\, v$ too. Hence $(v', v) \geq' (w', w)$ and $(v'', v)\, R'\, (v', v)$. Property (2.4) says that $(v'', v) \models' \varphi_1$, and so $v'' \models \varphi_1$ by induction hypothesis.

  For the other direction, assume $w' \models \Box\varphi_1$. Then

$$\text{for every } v' \geq w', \text{ if } v'' \,R\, v' \text{ then } v'' \models \varphi. \tag{2.5}$$

  We need to show that $(w', w) \models' \Box\varphi_1$.

  Consider a world $(v', v) \geq' (w', w)$, and fix it. We have $v' \,R\, v$, $v' \geq w'$ and $v \geq w$. Now, consider any world reachable from (v', v). We need to show that this world satisfies φ_1. There are two possible cases.

  This world is of the form (v'', v). In this case, we have that $v'' \,R\, v$. Since $v' \,R\, v$, we get $v'' \,R\, v'$. Since $v' \geq w'$, we get $v'' \models \varphi_1$ by (2.5). Hence, $(v'', v) \models' \varphi_1$, by induction hypothesis.

  In the other case, the world is of the form (q, v). Since $v \,R\, v'$ and $v' \geq w'$, we have $v \models \varphi_1$ by (2.5). Therefore, $(q, v) \models' \varphi_1$ by induction hypothesis. □

We need one more proposition which says that if a world satisfies a context then any world reachable from and/or greater than it also satisfies the context.

Proposition 19 (Forcing in Reachable Places). *Let* $\mathcal{W}_{Pls} = (W, \leq, R, V, Eval)$ *be a birelational model on Pls. Let* Γ *be a finite set of pure formulae,* Δ *be a finite set of sentences* Δ*, and w be a world in W such that* $w \models \Gamma; \Delta$*. Then*

1. $v \models \Gamma; \Delta$ *for every* $v \,R\, w$*, and*
2. $v \models \Gamma; \Delta$ *for every* $v \geq w$*.*

Proof. The second part of the proposition is an easy consequence of monotonicity of the logic. For the first part, pick $v \,R\, w$ and fix it. We need to show that if ψ is a formula in Γ then $v \models \Box\psi$, and that if φ ***at*** p is a sentence in Δ then $v \models \varphi@p$.

Now, if $\psi \in \Gamma$, then we have that $w \models \Box\psi$. Let v', v'' be two worlds such that $v'' \,R\, v' \geq v$. We will show that $v'' \models \psi$. As v'' is arbitrary, we will get that $v \models \Box\psi$.

We have $v' \geq v$ and $v \,R\, w$. By the reachability condition, we get that there is a w' such that $v' \,R\, w' \geq w$. Since, $v'' \,R\, v'$, and R is an equivalence, we get $v'' \,R\, w' \geq w$. Finally, since $w \models \Box\psi$, we get $v'' \models \psi$ as required.

If φ ***at*** $p \in \Delta$, then we have that $w \models \varphi@p$. Therefore, there is a world w' such that $w'{\downarrow}\, p$, $w\, R\, w'$ and $w' \models \varphi$. Since R is an equivalence, we get $v\, R\, w'$. Therefore $v \models \varphi@p$, and we are done. □

We are ready to prove soundness, which depends on Lemmas 8 and 9.

Theorem 20 (Bi-Soundness). *If the judgement* $\Gamma; \Delta \vdash^P \mu$ **at** p *is derivable in the logic, then it is bi-valid.*

Proof. The proof proceeds by induction on n, the number of inference rules applied in the derivation of the judgement $\Gamma; \Delta \vdash^P \mu$ ***at*** p. The inference rules are given in Fig. 2.1. The base case, where only one inference rule is used to derive the judgement, follows easily from the definition. We discuss the induction step.

Induction hypothesis ($n > 1$). We assume that the theorem holds for any judgement that is deducible by applying less than n instances of inference rules, and consider a judgement $\Gamma; \Delta \vdash^P \mu$ ***at*** p derivable in the logic by using exactly n instances.

We fix a model $\mathcal{W}_{Pls} = (W, \leq, R, V, Eval)$ on *Pls*, and let $\models$ be the forcing relation in this model. Let $w \in W$ be such that $P \subseteq Reach(w)$ and $w \models \Gamma; \Delta$. Fix w for the rest of the proof. We have to show $w \models \mu@p$. We proceed by cases by considering the last rule applied to obtain $\Gamma; \Delta \vdash^P \mu$ ***at*** p. For the sake of clarity, we consider only the cases in which the last rule is introduction of implication($\rightarrow I$), introduction of $\Box$ ($\Box$ I), and elimination of $\Diamond$ ($\Diamond$ E). The treatment of the other rules is similar.

- Case $\rightarrow I$. If the last inference rule used was $\rightarrow I$ then μ is of the form $\varphi \rightarrow \psi$, and $\mathsf{PL}(\Gamma; \Delta) \cup \mathsf{PL}(\varphi) \cup \mathsf{PL}(\psi) \cup \{p\} \subseteq P$. Furthermore, $\Gamma; \Delta, \varphi$ ***at*** $p \vdash^P \psi$ ***at*** p by using less than n instances of the inference rules. By induction hypothesis, $\Gamma; \Delta, \varphi$ ***at*** $p \vdash^P \psi$ ***at*** p is bi-valid. We have to prove that there exists $v\, R\, w$ such that $v{\downarrow}\, p$, and $v \models \varphi \rightarrow \psi$.

  Since $P \subseteq Reach(w)$, there exists $v\, R\, w$ such that $v{\downarrow}\, p$. We will prove that $v \models \varphi \rightarrow \psi$. Pick $v' \geq v$ and fix it. We need show that if $v' \models \varphi$, then $v' \models \psi$ also.

  We have $v'{\downarrow}\, p$ by coherence property, and $v' \models \Gamma; \Delta$ by Proposition 19. Also as R is reflexive, we have $v'\, R\, v'$. If we assume that $v' \models \varphi$, then we get by definition that $v' \models \varphi@p$. Hence, we get $v' \models \Gamma; \Delta, \varphi$ ***at*** p. By induction hypothesis $\Gamma; \Delta, \varphi$ ***at*** $p \vdash^P \psi$ ***at*** p is bi-valid, and therefore $v' \models \psi@p$.

  Therefore, there is a world reachable from v' which evaluates to p and which forces ψ. Since $v'{\downarrow}\, p$ and $v'\, R\, v'$, uniqueness says that this world must be v' itself. Therefore $v' \models \psi$, as required.

- *Case* $\Box I$. Then μ is of the form $\Box\varphi$. Moreover, $\mathsf{PL}(\Gamma; \Delta) \cup \mathsf{PL}(\varphi) \cup \{p\} \subseteq P$, and $\Gamma; \Delta \vdash^{P+q} \varphi$ ***at*** q for some $q \notin P$ by using less that n instances of the rules. By induction hypothesis, $\Gamma; \Delta \vdash^{P+q} \varphi$ ***at*** q is bi-valid. Without loss of generality, we can assume that $q \notin Pls$ (otherwise, we can rename q in *Pls*).

  We have that $w \models \Gamma; \Delta$, and we need to show that $w \models \Box\varphi@p$. Note that $p \in P$, and $P \subseteq Reach(w)$. Therefore there is a $w' \in Reach(w)$ such that $w'{\downarrow}\, p$. Pick such a w',

and fix it. By Proposition 19, $w' \models \Gamma; \Delta$. We shall show that $w' \models \Box\varphi$, and we will be done.

In order to show that $w' \models \Box\varphi$, we have to show that $v' \models \varphi$ for every v, v' such that $v' \, R \, v \geq w$. Pick such v, v' and fix them. We have $v' \models \Gamma; \Delta$ by Proposition 19. Since $P \subseteq Reach(w)$ and $v' \, R \, v \geq w$, we get $P \subseteq Reach(v')$ by Proposition 18.

Let $Pls' = Pls \cup \{q\}$, and let $\mathcal{W}\langle q \rangle_{Pls'}$ be the q-extension of the birelational model. Let $\models'$ be the forcing relation on $\mathcal{W}\langle u, q \rangle$. From the hypothesis $v' \models \Gamma; \Delta$ and Lemma 9, we get $(v', v') \models' \Gamma; \Delta$.

From definition of q-extension, it is clear that $Reach((v', v')) = Reach(v') \cup \{q\}$. Hence $P + q \subseteq Reach((v', v'))$. We can now apply the induction hypothesis on the world (v', v'), and obtain $(v', v') \models' \varphi@q$. By the definition of the q-extension, this is equivalent to $(q, v') \models' \varphi$. Lemma 9 then implies that $v' \models \varphi$, as required.

- *Case* $\Diamond E$. Then for some $p' \in P$ and $\varphi \in Frm(P)$ we can derive $\Gamma; \Delta \vdash^{P} \Diamond\varphi \; \mathbf{at} \; p'$ and $\Gamma; \Delta, \varphi \; \mathbf{at} \; q \vdash^{P+q} \mu \; \mathbf{at} \; p$ by using less than n instances of the rules. By induction hypothesis, $\Gamma; \Delta \vdash^{P} \Diamond\varphi \; \mathbf{at} \; p'$ and $\Gamma; \Delta, \varphi \; \mathbf{at} \; q \vdash^{P+q} \mu \; \mathbf{at} \; p$ are bi-valid.

  As is the case of $\Box I$, we can assume that $q \notin Pls$. We need to show that $w \models \mu@p$. Since $w \models \Gamma; \Delta$, the induction hypothesis says that $w \models \Diamond\varphi@p'$. Therefore using the definition of forcing and equivalence of the relation R, there is a world w' such that $w \, R \, w'$ and $w' \models \varphi$. Since $w \, R \, w'$, Proposition 19 implies that $w' \models \Gamma; \Delta$.

  Consider now the q-extension $\mathcal{W}\langle q \rangle$ of $\mathcal{W}$, with $\models'$ as forcing relation on the q-extension. Since $w' \models \varphi$ and $w' \models \Gamma; \Delta$, Lemma 9 says that $(q, w') \models' \varphi$ and $(q, w') \models' \Gamma; \Delta$. As $(q, w')\!\downarrow q$, we get $(q, w') \models' \Gamma; \Delta, \varphi \; \mathbf{at} \; q$. Finally, as $P + q \subseteq Reach(w') \cup \{q\} = Reach((q, w'))$, induction hypothesis gives us $(q, w') \models' \mu@p$. By Lemma 9, we get that $w' \models \mu@p$.

  Hence, there is a w'' such that $w' \, R \, w''$ such that $w'' \models \mu$ and $w''\!\downarrow p$. Since $w \, R \, w'$ and R is an equivalence, we get $w \, R \, w''$. Therefore $w \models \mu@p$, as required. □

This theorem provides not only soundness for birelational models, but also for Kripke models, thanks to the encoding presented in next section.

2.5.2 Relating Kripke and Birelational Models

In this section, we shall present an encoding of Kripke models in birelational models that preserves the forcing relation. This will allow us to prove the soundness of the logic for Kripke models.

In particular, given a Kripke model with a set of states K, we construct a birelational model whose worlds are pairs (k, p) where $k \in K$ and p is a place in the Kripke state k. Two worlds will be related if they come from the same Kripke state. The world (l, p) will be greater that (k, q) only if $l \geq k$ and $p = q$. The world (k, p) will evaluate to p, and an atom will be interpreted in the world (k, p) only if it is placed in p in the Kripke state

k. The construction will guarantee that the Kripke state k forces an assertion $\psi@p$ if and only if the corresponding world (k, p) forces the formula ψ .

Proposition 20 (Encoding). *Given a Kripke model, $\mathcal{K} = (K, \leq, \{P_k\}_{k\in K}, \{I_k\}_{k\in K})$ with set of places Pls, we define its $\mathcal{K}$-*birelational model $\mathcal{W}^{\mathcal{K}}_{Pls}$ *to be the quintuple $(W', \leq', R', I', Eval')$, where*

1. $W' \stackrel{\text{def}}{=} \bigcup_{k\in K}\{(k, p) : p \in P_k\}$;
2. $\leq' \subseteq W' \times W'$ *is defined as:* $(k, p) \leq' (l, q)$ *if and only if* $k \leq l$ *and* $p = q$;
3. $R' :\subseteq W' \times W'$ *is defined as:* $(k, p)\, R'(l, q)$ *if and only if* $k = l$;
4. $I' : Atoms \to Pow(W')$ *is defined as:* $I(A) \stackrel{\text{def}}{=} \{ (k, p) \mid p \in I_k(A) \}$;
5. $Eval' : W' \to Pls'$ *is defined as:* $Eval(k, p) \stackrel{\text{def}}{=} p$.

$\mathcal{W}^{\mathcal{K}}_{Pls}$ *is a birelational model.*

Proof. We need to check that the construction satisfies the properties of a birelational model. The proof is straightforward, and here we just illustrate the proof of the reachability condition.

Assume that $(k', p') \geq' (k, p)\, R'(l, q)$. Then it must be the case that $k' \geq k$, $k = l$ and $q \in P_l$. Since $k = l$, we get $q \in P_k$. Furthermore, as $k' \geq k$, we have $P_k \subseteq P_{k'}$. Therefore $q \in P_{k'}$.

Consider the world (k', q). We get $(k', p')\, R'(k', q) \geq' (k, q)$ by definition. □

The encoding preserves the forcing relation:

Proposition 21 (Forcing Preservation). *Let $\mathcal{K} = (K, \leq, \{P_k\}_{k\in K}, \{I_k\}_{k\in K})$ be a Kripke model with set of places Pls. Let $\mathcal{W}^{\mathcal{K}}_{Pls} = (W', \leq', R', I', Eval')$ be the $\mathcal{K}$-birelational model. Let $\models_{\mathcal{K}}$ and $\models_{\mathcal{W}}$ extend the interpretation of atoms in $\mathcal{K}$ and $\mathcal{W}^{\mathcal{K}}_{Pls}$ respectively. For every $\varphi \in Frm(Pls)$, $k \in K$, and $p \in P_k$, we have:*

$$(k, p) \models_{\mathcal{K}} \varphi \text{ if and only if } (k, p) \models_{\mathcal{W}} \varphi.$$

Proof. We proceed by induction on the formula $\varphi \in Frm(Pls)$. The statement of the proposition is easily verified on $\top$, $\bot$ and on atoms.

Induction hypothesis. We consider a formula $\varphi \in Frm(Pls)$, and assume that the proposition holds for each of its sub-formulae. For sake of clarity, we just illustrate the cases of logical implication, and modalities $@p$ and $\Box$.

- *Case $\varphi = \varphi_1 \to \varphi_2$.*

  Suppose $(k, p) \models_{\mathcal{K}} \varphi_1 \to \varphi_2$. We need to show that $(k, p) \models_{\mathcal{W}} \varphi_1 \to \varphi_2$. Pick $(l, q) \geq' (k, p)$ such that $(l, q) \models_{\mathcal{W}} \varphi_1$, and fix it. It suffices to show that $(l, q) \models_{\mathcal{W}} \varphi_2$ also.

Since $(l, q) \geq' (k, p)$, we have $q = p$ and $l \geq k$. Also, as $(l, q) \models_{\mathcal{W}} \varphi_1$ and $q = p$, we get $(l, p) \models_{\mathcal{K}} \varphi_1$ by induction hypothesis. Since $(k, p) \models_{\mathcal{K}} \varphi_1 \to \varphi_2$ and $l \geq k$, we get $(l, p) \models_{\mathcal{K}} \varphi_2$. By induction hypothesis once again, we get $(l, q) = (l, p) \models_{\mathcal{W}} \varphi_2$, and we are done.

For the other direction, suppose that $(k, p) \models_{\mathcal{W}} \varphi_1 \to \varphi_2$. We need to show that $(k, p) \models_{\mathcal{K}} \varphi_1 \to \varphi_2$. Pick $l \geq k$ such that $(l, p) \models_{\mathcal{K}} \varphi_1$, and fix it. It suffices to show that $(l, p) \models_{\mathcal{K}} \varphi_2$.

As $(l, p) \models_{\mathcal{K}} \varphi_1$, we have by induction hypothesis that $(l, p) \models_{\mathcal{W}} \varphi_1$. Since $l \geq k$, we get $p \in P_l$ and $(l, p) \geq' (k, p)$. Therefore, as $(k, p) \models_{\mathcal{W}} \varphi_1 \to \varphi_2$, we get that $(l, p) \models_{\mathcal{W}} \varphi_2$. By induction hypothesis, we get $(l, p) \models_{\mathcal{K}} \varphi_2$.

- *Case* $\varphi = \varphi_1@q$.

  Then $(k, p) \models_{\mathcal{K}} \varphi$ means that $q \in P_k$ and $(k, q) \models_{\mathcal{K}} \varphi_1$. By induction hypothesis and definition, this is equivalent to saying that there exists $(k, q)\, R'(k, p)$ such that $(k, q)\!\downarrow q$, and $(k, q) \models_{\mathcal{W}} \varphi_1$. This is equivalent to saying that $(k, p) \models_{\mathcal{W}} \varphi_1@q$.

- *Case* $\varphi = \Box\varphi_1$.

  Then $(k, p) \models_{\mathcal{K}} \varphi$ means that for every $l \geq k$ and every $q \in P_l$, we have $(l, q) \models_{\mathcal{K}} \varphi_1$. By induction hypothesis and definition, this is equivalent to: for every $(l, p) \geq' (k, p)$ and $(l, q)\, R'(l, p)$, it is the case that $(l, q) \models_{\mathcal{W}} \varphi_1$. This is equivalent to saying that $(k, p) \models_{\mathcal{W}} \Box\varphi_1$. □

One thing that is worth pointing out is that in the resulting birelational model, the evaluation is *total*. It is easy to see the converse: every birelational model with a total evaluation can be encoded as a Kripke model such that the forcing relation is preserved. In the reverse encoding, the set of Kripke states is the set of equivalence classes under reachability, and the set of places associated to a class is the set of all the evaluations of its elements. Therefore, the class of Kripke models corresponds semantically to the class of birelational models in which the evaluation is total.

The encoding cannot be preserved if we consider birelational worlds with partial evaluation. Please note that this is not just a consequence of having undefined worlds in birelational models. If this was the case, we could have added "undefined" places in each Kripke state. The real issue is that when the evaluation is partial, two "undefined" worlds reachable by each other can be ordered: a situation that will be ruled out if the evaluation was total as a consequence of coherence and uniqueness. In Kripke models, however, "reachability" and order are essentially orthogonal. Hence, the reverse encoding will fail to preserve the forcing relation.

This is no accident, and as we have pointed out before, partiality of the evaluation in birelational models is essential for the proof of the finite model property. This was illustrated by the "finite model" $\mathcal{W}_{exam}$ in Ex. 4. In $\mathcal{W}_{exam}$, it is the case that $w_1 \leq w_2$, $w_1\, R\, w_2$, $w_1\!\uparrow$ and $w_2\!\downarrow p$. As discussed there, this model allows us to refute the judgement $; \Box\neg\neg A \textbf{ at } p \vdash^{\{p\}} \neg\neg\Box A \textbf{ at } p$. As we will see later, the judgement will be valid in every finite Kripke model.

We shall now use the encoding and soundness of logic with respect to birelational models to show soundness of Kripke semantics.

Corollary 4 (Soundness). *If* $\Gamma;\Delta \vdash^P \mu$ **at** p *is derivable in the logic, then it is valid in every Kripke model.*

Proof. Suppose that the judgement $\Gamma;\Delta \vdash^P \mu$ ***at*** p is derivable. Then it must be the case that $\mathsf{PL}(\Gamma) \cup \mathsf{PL}(\Delta) \cup \mathsf{PL}(\mu) \cup \{p\} \subseteq P$. Let $\mathcal{K} = (K, \leq, \{P_k\}_{k\in K}, \{I_k\}_{k\in K})$ be a Kripke model with set of places *Pls*. Let $\models_{\mathcal{K}}$ extend the interpretation of atoms to formulae on this Kripke model. Let k be a Kripke state of this model such that $P \subseteq P_k$ and $k \models_{\mathcal{K}} \Gamma;\Delta$. We need to show that $(k, p) \models_{\mathcal{K}} \mu$.

Consider the encoding of the Kripke model $\mathcal{K}$ into a birelational model. Let $\mathcal{W}^{\mathcal{K}}_{Pls} = (W', \leq', R', I', Eval')$ be the $\mathcal{K}$-birelational model, and consider the world $(k, p) \in W'$. If $\models_{\mathcal{W}}$ is the extension of interpretation of atoms in this model, we claim that $(k, p) \models_{\mathcal{W}} \Gamma;\Delta$.

If $\psi \in \Delta$ then as $k \models_{\mathcal{K}} \Gamma;\Delta$, we get by definition $(k, p) \models_{\mathcal{K}} \Box\psi$. By Proposition 21, we get that $(k, p) \models_{\mathcal{W}} \Box\psi$.

If ψ ***at*** $q \in \Gamma$, then we have by definition $(k, q) \models_{\mathcal{K}} \psi$. By Proposition 21, we get that $(k, q) \models_{\mathcal{W}} \psi$. Now, by construction $(k, p)\, R'(k, q)$, and hence we get $(k, p) \models_{\mathcal{W}} \psi@q$.

Therefore, we get that $(k, p) \models_{\mathcal{W}} \Gamma;\Delta$. As the logic is sound over birelational models, we get $(k, p) \models_{\mathcal{W}} \mu@p$. This implies that $(k, p) \models_{\mathcal{K}} \mu@p$, by Proposition 21 once again. Finally, this is the same as $(k, p) \models_{\mathcal{K}} \mu$, by definition, and we have done. □

2.6 Bounded Contexts and Completeness

In this section, we shall prove completeness of the logic with respect to both Kripke and birelational semantics. The proof will follow a modification of standard proofs of completeness of intuitionistic logics[94, 132, 32, 138], and we will construct a particular Kripke model: the *canonical bounded Kripke model*. The reason for the term "bounded" shall become clear later on. We will prove that a judgement $\Gamma;\Delta \vdash^P \mu$ ***at*** p is valid in the canonical bounded model if and only if it is derivable in the logic. Then we will use the encoding of the Kripke models into birelational models (see §2.5.2), which will allow us to prove completeness of birelational models. The resulting model will be used to prove the finite model property in §2.7.3. The construction of the model is adapted from [132].

We also point out that we shall prove the completeness results in the case where P is finite. This is not a serious restriction for completeness, and the result can be extended to judgements where P is infinite. The real advantage of using a finite set of places is that it will assist in the proof of finite model property as we will see in §2.7.

We begin by defining sub-formulae of a pure formula. A *sub-formula* of a pure formula φ is inductively generated as:

- φ is a sub-formula of itself;
- if any of $\varphi_1 \wedge \varphi_2$, $\varphi_1 \vee \varphi_2$, and $\varphi_1 \to \varphi_2$ is a sub-formula of φ, then so are φ_1 and φ_2; and

- if any of $\Box\varphi_1$, $\Diamond\varphi_1$, and $\varphi_1@p$ is a sub-formula of φ, then so is φ_1.

Given any set of pure formulae Θ, the *sub-formula closure* Θ^*, is the set of sub-formulae of each of its members. Formally: $\Theta^* \stackrel{\text{def}}{=} \{\psi : \psi \text{ is a subformula of } \varphi \in \Theta\}$. *Bounded contexts* are defined by using sub-formulae closure.

Definition 16 (Bounded Contexts). Given a finite set of places P and a finite set of pure formulae $\Theta \in Frm(P)$, a pair (Q, Δ) is a $(P, \Theta)-$*bounded context* if

- Q is a finite set of places that contains P, *i.e.*, $P \subseteq Q$; and
- Δ is a finite set of sentences of the form φ ***at*** q, where $\varphi \in \Theta^*$ and $q \in Q$.

The bounded contexts will be used as Kripke states in the canonical model. However, we will need particular kinds of bounded contexts.

Definition 17 (Prime Bounded Contexts). Let P be a finite set of places, and $\Theta, \Gamma \subseteq Frm(P)$ be two finite sets of pure formulae. A $(P, \Theta)-$bounded context (Q, Δ) is said to be $\Gamma-$*prime* if

- $\Gamma; \Delta \vdash^Q \varphi$ ***at*** q for $\varphi \in \Theta^*$ and $q \in Q$, implies that φ ***at*** $q \in \Delta$ (Θ-deductive closure);
- $\Gamma; \Delta \nvdash^Q \bot$ ***at*** q for every $q \in Q$ (Consistency);
- $\Gamma; \Delta \vdash^Q \varphi \vee \psi$ ***at*** q for $\varphi \vee \psi \in \Theta^*$ and $q \in Q$, implies that either φ ***at*** $q \in \Delta$ or ψ ***at*** $q \in \Delta$ (Θ-disjunction property); and
- $\Gamma; \Delta \vdash^Q \Diamond\varphi$ ***at*** q for $\Diamond\varphi \in \Theta^*$ and $q \in Q$, implies that there exists $q' \in Q$ such that φ ***at*** $q' \in \Delta$ (Θ-diamond property).

As an example, let A be an atom. Let $P = \{p\}$, $\Theta = \{A@p\}$ and $Q = \{p, q\}$. Consider the following sets of sentences:

- $\Delta_1 = \{A$ ***at*** $p,\ A$ ***at*** $q,\ A@p$ ***at*** $p\}$;
- $\Delta_2 = \{A$ ***at*** $p,\ A$ ***at*** $q,\ A@p$ ***at*** $p,\ A@p$ ***at*** $q\}$; and
- $\Delta_3 = \{A$ ***at*** $p,\ A$ ***at*** $q,\ A@p$ ***at*** $p,\ A@p$ ***at*** $q,\ \Diamond A$ ***at*** $q\}$.

Clearly, we have that $P \subseteq Q$. If ψ ***at*** r is a sentence in Δ_1 or Δ_2, then ψ is a sub-formula of Θ and $r \in Q$. Therefore, (Q, Δ_1) and (Q, Δ_2) are $(P, \Theta)-$ bounded contexts. On the other hand, (Q, Δ_3) is not a $(P, \Theta)-$bounded context as $\Diamond A$ is not a sub-formula of $A@p$.

If we let Γ to be the list $\{A\}$, then it follows easily that $\Gamma; \Delta_1 \vdash^Q A$ ***at*** p. Using the inference rule of introduction of @, we get $\Gamma; \Delta_1 \vdash^Q A@p$ ***at*** q. However, we have that $A@p$ ***at*** $q \notin \Delta_1$. Therefore, (Q, Δ_1) is not $\Gamma-$prime. On the other hand, (Q, Δ_2) is $\Gamma-$prime.

The canonical model will be built by choosing the Kripke states to be prime bounded contexts. We will first show that bounded contexts can be extended to prime bounded contexts. Before we proceed, we state a proposition that says that the cut-rule is admissible in the logic. In [91], this has been proved for the logic without the disjunctive connectives. The proof can be extended for the logic with disjunctive connectives:

Proposition 22. *If* $\Gamma;\Delta \vdash^P \mu$ **at** p_1 *and* $\Gamma;\Delta,\mu$ **at** $p_1 \vdash^P \psi$ **at** p*, then* $\Gamma;\Delta \vdash^P \psi$ **at** p*.*

Proof. The proof is by induction on the number of inference rules used in derivation of $\Gamma;\Delta,\mu$ ***at*** $p_1 \vdash^P \psi$ ***at*** p. □

We now show the existence of prime extensions:

Lemma 10 (Prime Bounded Extension). *Let* (Q,Δ) *be a* (P,Θ)*−bounded context, and* ψ *be a pure formula in Frm*(P)*. Given a finite subset* $\Gamma \subseteq$ *Frm*(P) *and* $q \in Q$ *such that* $\Gamma;\Delta \nvdash^Q \psi$ **at** q*, there exists a* (P,Θ)*−bounded context* (Q',Δ') *such that*

1. (Q',Δ') *is* Γ*−prime,*
2. (Q',Δ') *extends* (Q,Δ), i.e., $Q \subseteq Q'$*, and* $\Delta \subseteq \Delta'$*, and*
3. $\Gamma;\Delta' \nvdash^{Q'} \psi$ **at** q*.*

Proof. Please note that by definition P,Θ and Θ^* are finite sets. Pick new places $q_{\Diamond\varphi}$, one for each formula $\Diamond\varphi \in \Theta^*$. Let $Q_\Diamond$ be the set of all such places. As the set Θ^* is finite, $Q_\Diamond$ is also a finite set. Finally, let Σ be the set of sentences φ ***at*** q such that $\varphi \in \Theta^*$ and $q \in Q \cup Q_\Diamond$. As Θ^*, Q and $Q_\Diamond$ are finite sets, Σ is also finite.

The set Δ' required in the lemma would be a subset of Σ, and the set Q' would be a subset of $Q \cup Q_\Diamond$. These sets would be obtained by a series of extensions Δ_n, Q_n which will satisfy certain properties:

Property 1. *For every* $n \geq 0$

1. $Q_n \subseteq Q \cup Q_\Diamond$*, and* $\Delta_n \subseteq \Sigma$*;*
2. $Q_n \subseteq Q_{n+1}$*,* $\Delta_n \subseteq \Delta_{n+1}$*;*
3. (Q_n,Δ_n) *is* (P,Θ)*-bounded context; and*
4. $\Gamma;\Sigma_n \nvdash^{Q_n} \psi$ **at** q*.*

The series is constructed inductively. In the induction, at an odd step we will create a witness for a formula of the type $\Diamond\varphi$. At an even step we deal with disjunction property. We shall also construct two sets:

- $\mathit{treated}_n^\Diamond$, that will be the set of the formulae $\Diamond\varphi \in \Theta^*$ for which we have already created a witness.
- $\mathit{treated}_n^\vee$, that will be the set of the formulae $\psi_1 \vee \psi_2$ ***at*** $q \in \Sigma$ which satisfy the disjunction property.

We pick an enumeration of Θ^*, and fix it. We start off by defining $\mathit{treated}_0^\Diamond = \emptyset$, $\mathit{treated}_0^\vee = \emptyset$, $Q_0 = Q$, and $\Delta_0 = \Delta$. It is clear from the hypothesis of the lemma that Q_0 and P_0 satisfy the four points of Property1.

Then we proceed inductively, and assume that Q_n, Δ_n $(n \geq 0)$ have been constructed satisfying Property 1. In step $n+1$, we consider two cases:

1. If $n+1$ is odd, then pick the first formula $\psi_1 \vee \psi_2 \in \Theta^*$ in the enumeration of Θ^* , such that

   - $\Gamma; \Delta_n \vdash^{Q_n} \psi_1 \vee \psi_2$ ***at*** r, for some $r \in Q_n$;
   - $\psi_1 \vee \psi_2$ ***at*** $r \notin \mathit{treated}_n^{\vee}$.

   If no such formula exists, then let $Q_{n+1} = Q_n$ and $\Delta_{n+1} = \Delta_n$. In this case Q_{n+1} and Δ_{n+1} satisfy the four points of Property 1 by induction.

   Otherwise, if both $\Gamma; \Delta_n, \psi_1$ ***at*** $r \vdash^{Q_n} \psi$ ***at*** q and $\Gamma; \Delta_n, \psi_2$ ***at*** $r \vdash^{Q_n} \psi$ ***at*** q, then we can deduce $\Gamma; \Delta_n \vdash^{Q_n} \psi$ ***at*** q. However, we have that Δ_n, Q_n satisfy Property 1. Hence, it must be the case that either $\Gamma; \Delta_n, \psi_1$ ***at*** $r \nvdash^{Q_n} \psi$ ***at*** q or $\Gamma; \Delta_n, \psi_2$ ***at*** $r \nvdash^{Q_n} \psi$ ***at*** q.

   We define $\Delta_{n+1} = \Delta_n \cup \{\psi_1$ ***at*** $r\}$ if $\Gamma; \Delta_n, \psi_1$ ***at*** $r \nvdash^{Q_n} \psi$ ***at*** p, and $\Delta_{n+1} = \Delta_n \cup \{\psi_2$ ***at*** $r\}$ otherwise. We define $Q_{n+1} = Q_n$. We have by construction $Q_n \subseteq Q_{n+1}$, $Q_{n+1} \subseteq Q \cup Q_\Diamond$ and $\Delta_n \subseteq \Delta_{n+1}$.

   We have $r \in Q_n$. By definition, the set Θ^* is closed under sub-formulae. Therefore as $\psi_1 \vee \psi_2 \in \Theta^*$, we have both ψ_1 and ψ_2 are in Θ^*. This implies that ψ_1 ***at*** r and ψ_1 ***at*** r are in Σ, and (Q_{n+1}, Δ_n) is $(P, \Theta)-$bounded context.

   Also by construction $\Gamma; \Delta_{n+1} \nvdash^{Q}_{n+1} \psi$ ***at*** q. Therefore, Q_{n+1}, Δ_{n+1} satisfies Property 1. Finally, we let $\mathit{treated}_{n+1}^{\vee} = \mathit{treated}_n^{\vee} \cup \{\psi_1 \vee \psi_2$ ***at*** $r\}$ and $\mathit{treated}_{n+1}^{\Diamond} = \mathit{treated}_n^{\Diamond}$.

2. If $n+1$ is even, pick the first formula $\Diamond\varphi$ in the enumeration of Θ^* such that

   - $\Gamma; \Delta_n \vdash^{Q_n} \Diamond\varphi$ ***at*** r, for some $r \in Q_n$;
   - $\Diamond\varphi \notin \mathit{treated}_n^{\Diamond}$.

   Let $Q_{n+1} = Q_n + q_{\Diamond\varphi}$, $\Delta_{n+1} = \Delta_n \cup \{\varphi$ ***at*** $q_{\Diamond\varphi}\}$, $\mathit{treated}_{n+1} = \mathit{treated}_n \cup \{\Diamond\varphi\}$ and $\mathit{treated}_{n+1}^{\vee} = \mathit{treated}_n^{\vee}$. We have by construction that Q_{n+1} and Δ_{n+1} satisfy the first three points of Property1. We claim that $\Gamma; \Delta_{n+1} \nvdash^{Q_{n+1}} \psi$ ***at*** q also.

   Suppose that $\Gamma; \Delta_{n+1} \vdash^{Q_{n+1}} \psi$ ***at*** q, *i.e.*, $\Gamma; \Delta_n, \varphi$ ***at*** $q_{\Diamond\varphi} \vdash^{Q+q_{\Diamond\varphi}} \psi$ ***at*** q. We also have that $\Gamma; \Delta_n \vdash^{Q_n} \Diamond\varphi$ ***at*** r. In fact, by the inference rule $\Diamond E$:

$$\frac{\Gamma; \Delta_n \vdash^{Q_n} \Diamond\varphi \textbf{ at } r \quad \Gamma; \Delta_n, \varphi \textbf{ at } q_{\Diamond\varphi} \vdash^{Q+q_{\Diamond\varphi}} \psi \textbf{ at } q}{\Gamma; \Delta_n \vdash^{Q_n} \psi \textbf{ at } q}\ \Diamond E$$

   This contradicts the hypothesis on Q_n, Δ_n. Hence $\Gamma; \Delta_{n+1} \nvdash^{Q_{n+1}} \psi$ ***at*** q. Therefore, Q_{n+1} and Δ_{n+1} satisfy Property1.

Therefore, we get by construction that Q_n, Δ_n satisfy Property 1. We define $Q' = \bigcup_{n\geq 0} Q_n$, and $\Delta'' = \bigcup_{n\geq 0} \Delta_n$. Now, using Property 1, $Q' \subseteq Q \cup Q_\Diamond$ and $\Delta'' \subseteq \Sigma$. This implies that Q' and Δ'' are finite sets. (Note that this means that the series (Q_n, Δ_n) is eventually constant). Using Property 1, we can easily show that (Q', Δ'') is a $(P, \Theta)-$bounded context, and $\Gamma; \Delta'' \nvdash^{Q'} \psi$ ***at*** q.

Finally, we define Δ' to be the set of all sentences φ ***at*** $s \in \Sigma$ such that $\Gamma; \Delta'' \vdash^{Q'} \varphi$ ***at*** s. As a consequence of Proposition 22, we get that

$$\Gamma; \Delta' \vdash^{Q'} \mu \textbf{ at } r \text{ if and only if } \Gamma; \Delta'' \vdash^{Q'} \mu \textbf{ at } r \tag{2.6}$$

Clearly, Δ' extends Δ'' and hence Δ. Furthermore, (Q', Δ') is $(P, \Theta)-$bounded by construction. Also we get $\Gamma; \Delta' \nvdash^{Q'} \psi$ ***at*** q, thanks to the equivalence (2.6). We only need to show that (Q', Δ') is Γ-prime.

1. (Deductive Closure) The set Δ' is deductively closed, by construction.

2. (Disjunction Property) Assume that $\Gamma; \Delta' \vdash^{Q'} \psi_1 \vee \psi_2$ ***at*** r, for $\psi_1 \vee \psi_2 \in \Theta^*$ and $q \in Q'$. Then let n be the least number such that $\Gamma; \Delta_n \vdash^{Q_n} \psi_1 \vee \psi_2$ ***at*** r. Clearly, $\psi_1 \vee \psi_2$ ***at*** $q \notin treated_n^{\vee}$, and $\Gamma; \Delta_m \vdash^{Q_m} \psi_1 \vee \psi_2$ ***at*** q for every $m \geq n$. Eventually $\psi_1 \vee \psi_2$ ***at*** q has to be treated at some odd stage $h \geq n$. Hence, either ψ_1 ***at*** $r \in \Delta_{h+1}$ or ψ_2 ***at*** $r \in \Delta_{h+1}$. Therefore, ψ_1 ***at*** $q \in \Delta'$ or ψ_2 ***at*** $q \in \Delta'$.

3. (Diamond Property) Assume that $\Gamma; \Delta' \vdash^{Q'} \Diamond\varphi$ ***at*** r, for $\Diamond\varphi \in \Theta^*$ and $r \in Q'$. Then let n be the least number such that $\Gamma; \Delta_n \vdash^{Q_n} \Diamond\varphi$ ***at*** r. As in the previous case, we assert that $\Diamond\varphi$ ***at*** q is treated for some even number $h \geq n$. We get φ ***at*** $q_{\Diamond\varphi} \in \Delta'$ by construction.

4. (Consistency) If $\Gamma; \Delta' \vdash^{Q'} \bot$ ***at*** r, then $\Gamma; \Delta' \vdash^{Q'} \psi@q$ ***at*** r by the inference rule $\bot E$. Therefore, $\Gamma; \Delta' \vdash^{Q'} \psi$ ***at*** q by $@E$, which contradicts our construction. Hence, $\Gamma; \Delta' \nvdash^{Q'} \bot$ ***at*** q.

We conclude that (Q', Δ') is a Γ-prime and $(P, \Theta)-$ bounded context extending (Q, Δ) such that $\Gamma; \Delta \nvdash^{Q'} \varphi$ ***at*** p. □

We finally construct the bounded canonical model. In the model, the set of Kripke states is the set of prime bounded contexts (Q, Δ) ordered by inclusion. A place belongs to the state (Q, Δ) only if it is in Q, and an atom A is placed in a place r in the state (Q, Δ) only if A ***at*** $r \in \Delta$. More formally, we have

Definition 18 (Bounded Canonical Model). Given a finite set of places P and two finite sets of pure formulae $\Theta, \Gamma \subseteq Frm(P)$, the Γ-*prime and* $(P, \Theta)-$*bounded canonical model* is the quadruple $\mathcal{K}_{can} \stackrel{\text{def}}{=} (K, \leq, \{P_k\}_{k \in K}, \{I_k\}_{k \in K})$, where

- the set K is the set of all $(P, \Theta)-$bounded contexts that are Γ-prime;
- $(Q_1, \Delta_1) \leq (Q_2, \Delta_2)$ if and only if $Q_1 \subseteq Q_2$ and $\Delta_1 \subseteq \Delta_2$; and
- $P_{(Q,\Delta)} \stackrel{\text{def}}{=} Q$;
- for $k = (Q, \Delta)$, the function $I_k : Atoms \to Pow(P_k)$ is defined as

$$I_{(Q,\Delta)}(A) \stackrel{\text{def}}{=} \{q \in Q : A \textbf{ at } q \in \Delta\}.$$

Given a finite set of places P and a finite set of formulae $\Gamma \in Frm(P)$, we say that Γ is consistent if $\Gamma; \nvdash^P \bot$ ***at*** p for any $p \in P$. If Γ is consistent, then Lemma 10 guarantees that the set of states in the canonical model is non-empty. This ensures that the bounded canonical model is a Kripke model.

Lemma 11 (Canonical Evaluation). *Given a finite set places P, and two finite sets of pure formulae $\Theta, \Gamma \in Frm(P)$ such that Γ is consistent, let $\mathcal{K}_{can}$ be the $\Gamma-$prime and $(P,\Theta)-$bounded canonical model. Then*

1. *$\mathcal{K}_{can}$ is a Kripke model; and*
2. *if $\Vdash_{\mathcal{K}}$ is the forcing relation on $\mathcal{K}_{can}$, then for every $\varphi \in \Theta^*$, every $(Q,\Delta) \in K$, and every $q \in Q$ it holds: $(Q,\Delta) \Vdash_{\mathcal{K}} \varphi$ **at** q if and only if φ **at** $q \in \Delta$.*

Proof. Clearly, all the properties required for a Kripke model are verified. All we have to prove is the part *2* of the lemma. The proof is standard, and we proceed by induction on the structure of the formula $\varphi \in \Theta^*$. In the induction hypothesis, we assume that part *2* of the lemma is valid on all sub-formulae of φ that are in Θ^*. Please note that if $\varphi \in \Theta^*$, then all of the sub-formulae of φ are in Θ^*. Hence, we can apply the induction hypothesis on all the sub-formulae of φ. Here, we just illustrate the inductive case in which φ is $\Box\varphi_1$.
Case $\Box\varphi_1$. Assume that $(Q,\Delta) \Vdash_{\mathcal{K}} \Box\varphi_1$ ***at*** q, where $\Box\varphi_1 \in \Theta^*$. By definition, this means that for every $(Q',\Delta') \geq (Q,\Delta)$ and every $r \in Q'$, it is the case that $(Q',\Delta') \Vdash_{\mathcal{K}} \varphi_1$ ***at*** r (and therefore φ_1 ***at*** $r \in \Delta'$ by induction hypothesis).

Chose a new place $s \notin Q$ and fix it. We claim that $\Gamma; \Delta \vdash^{Q+s} \varphi_1$ ***at*** s. Suppose $\Gamma; \Delta \nvdash^{Q+s} \varphi_1$ ***at*** s. Then by Lemma 10, there is a set of places Q' extending $Q + s$ and, a Γ-prime and $(P,\Theta)-$bounded context (Q',Δ') extending (Q,Δ) such that $\Gamma; \Delta' \nvdash^{Q'} \varphi_1$ ***at*** s. This means φ_1 ***at*** $s \notin \Delta'$. Since (Q',Δ') is greater than (Q,Δ), we obtain a contradiction.

Therefore, we conclude that $\Gamma; \Delta \vdash^{Q+s} \varphi_1$ ***at*** s. By using the inference rule of introduction of $\Box$ ($\Box I$), we get that $\Gamma; \Delta \vdash^{Q} \Box\varphi_1$ ***at*** q. Since (Q,Δ) is Γ-prime and (P,Θ)-bounded, $\Box\varphi_1$ ***at*** $q \in \Delta$.

For the other direction, let $\Box\varphi_1$ ***at*** $q \in \Delta$. Pick a Kripke state $(Q',\Delta') \geq (Q,\Delta)$, and fix it. We need to show that $(Q',\Delta') \Vdash_{\mathcal{K}} \varphi_1$ ***at*** q. Now $\Delta \subseteq \Delta'$, and therefore $\Box\varphi_1$ ***at*** $q \in \Delta'$. We can apply the inference rule of elimination of $\Box$ ($\Box E$) to prove that $\Gamma, \Delta' \vdash^{Q'} \varphi_1$ ***at*** s for every $s \in Q'$.

By definition of the canonical model, (Q',Δ') is Γ-prime. Therefore, φ_1 ***at*** $s \in \Delta'$ for every $s \in Q'$. Hence by induction hypothesis, $(Q',\Delta') \Vdash_{\mathcal{K}} \varphi_1$ ***at*** s for every $s \in Q'$. As (Q',Δ') is an arbitrary Kripke state larger than (Q,Δ), we get that $(Q,\Delta) \Vdash_{\mathcal{K}} \Box\varphi_1$ ***at*** q. $\Box$

We are now ready to prove completeness. It will imply the completeness theorem for birelational models as a corollary. We will later on recall the proof of this theorem when we deal with the finite model property.

Theorem 21 (Completeness). *If P is finite and the judgement $\Gamma; \Delta \vdash^P \varphi$ **at** p is valid in every Kripke model, then it is provable in the logic.*

Proof. Assume that $\Gamma;\Delta \models^P \varphi$ ***at*** p is valid. We have:

1. $\mathsf{PL}(\Gamma) \cup \mathsf{PL}(\Delta) \cup \mathsf{PL}(\varphi) \cup \{p\} \subseteq P$.
2. If $\mathcal{K} = (K, \leq, \{P_k\}_{k\in K}, \{I_k\}_{k\in K})$ is a Kripke model, then for every $k \in K$ such that $P \subseteq P_k$, $k \models \varphi$ ***at*** p whenever $k \models \Gamma;\Delta$.

We need to show that $\Gamma;\Delta \vdash^P \varphi$ ***at*** p.

Assume that $\Gamma;\Delta \nvdash^P \varphi$ ***at*** p. We fix $\Theta \stackrel{\text{def}}{=} \{\Box\psi : \psi \in \Gamma\} \cup \{\mu : \mu$ ***at*** $q \in \Delta\} \cup \{\varphi\}$. Please note that $\Theta \in \mathit{Frm}(P)$ and (P,Δ) is a (P,Θ)-bounded context. By Lemma 10, there is a Γ-prime and $(P,\Theta)-$ bounded context (Q,Σ) extending (P,Δ) such that $\Gamma;\Sigma \nvdash^Q \varphi$ ***at*** p. We get φ ***at*** $p \notin \Sigma$. Fix (Q,Σ).

Now consider the Γ-prime and (P,Θ)-bounded canonical model $\mathcal{K}_{can}$ as constructed in Definition 18, and let $\models_{\mathcal{K}}$ be the forcing relation in $\mathcal{K}_{can}$. Consider the Kripke state (Q,Σ). We claim that $(Q,\Sigma) \models_{\mathcal{K}} \Gamma;\Delta$.

Pick $\psi \in \Gamma$, $r \in Q$ and fix them. We first show that $\Gamma;\Sigma \vdash^Q \Box\psi$ ***at*** r. In the proof, we first choose a new place $m \notin Q$, and then use the inference rule G to conclude that ψ ***at*** r is derivable from Γ,Σ. We then use the inference rule $\Box I$ to obtain $\Gamma;\Sigma \vdash^Q \Box\psi$ ***at*** r. More formally,

$$\dfrac{\dfrac{}{\Gamma;\Sigma \vdash^{Q+m} \psi \textbf{ at } m}\, G}{\Gamma;\Sigma \vdash^{Q} \Box\psi \textbf{ at } r}\, \Box I$$

As $\psi \in \Gamma$, we have that $\Box\psi \in \Theta$. As $r \in Q$, we have by definition of prime contexts, $\Box\psi$ ***at*** $r \in \Sigma$. Using Lemma 11, we get that $(Q,\Sigma) \models_{\mathcal{K}} \Box\psi$ ***at*** r.

Furthermore, Δ is contained in Σ. Therefore, by Lemma 11, $(Q,\Sigma) \models_{\mathcal{K}} \mu$ ***at*** q whenever μ ***at*** $q \in \Delta$.

Hence, we get that the Kripke state $(Q,\Sigma) \models \Gamma;\Delta$. By our assumption, we get $(Q,\Sigma) \models_{\mathcal{K}} \varphi$ ***at*** p also. By Lemma 11, we get φ ***at*** $p \in \Sigma$. However our choice of Q,Σ was such that φ ***at*** $p \notin \Sigma$. We have just reached a contradiction, and hence we can conclude that $\Gamma;\Delta \vdash^P \varphi$ ***at*** p. □

Now, by the encoding of Kripke models into birelational models (see Proposition 21), if a judgement is valid in all birelational models then it is valid in all Kripke models. As the class of Kripke models is complete, we get that the class of birelational models is also complete for the logic.

Corollary 5. *If P is finite and the judgement $\Gamma;\Delta \vdash^P \varphi$* **at** *$p$ is bi-valid in every birelational model, then it is provable in the logic.*

Proof. Suppose that the judgement $\Gamma;\Delta \vdash^P \varphi$ ***at*** p is not provable in the logic. Then by Theorem 21, there is a Kripke model $\mathcal{K}$ with a state k such that k forces $\Gamma;\Delta$ but does not force φ ***at*** p. Let $\mathcal{W}^{\mathcal{K}}_{Pls}$ be the $\mathcal{K}$-birelational model obtained by the encoding of $\mathcal{K}$ as defined in Proposition 20, and consider the world (k,p). It can be shown using Proposition 21 that the world (k,p) forces $\Gamma;\Delta$ but not φ ***at*** p. Hence, the judgement $\Gamma;\Delta \vdash^P \varphi$ ***at*** p is not bi-valid. □

Now, the proofs in this section can be suitably modified to allow P to be infinite, as they do not actually require context sets to be finite. Finiteness is actually required for the proof of the finite model property, and not for completeness.

There is another way in which we can deduce the completeness results when P is infinite. For this, we take recourse to the following proposition which states that, to derive a judgment, it is sufficient just to consider the set of places appearing in the formulae of the judgement itself. This was proved for the logic without disjunctive connectives in [91], and the proof can be extended for the whole logic.

Proposition 23. *Let* $P_0 = \mathsf{PL}(\Gamma) \cup \mathsf{PL}(\Delta) \cup \mathsf{PL}(\varphi) \cup \{p\}$, *and* $P_0 \subseteq P$. *Then* $\Gamma; \Delta \vdash^{P} \varphi$ **at** p *if and only if* $\Gamma; \Delta \vdash^{P_0} \varphi$ **at** p.

Proof. The proof is by induction on the length of derivations. □

In order to use completeness result for judgements in which P is infinite, we proceed as follows. Suppose that

$$\Gamma; \Delta \nvdash^{P} \varphi \textbf{ at } p.$$

Let $P_0 = \mathsf{PL}(\Gamma) \cup \mathsf{PL}(\Delta) \cup \mathsf{PL}(\varphi) \cup \{p\}$. Please observe that by the above proposition, we get

$$\Gamma; \Delta \nvdash^{P_0} \varphi \textbf{ at } p.$$

Using Theorem 21, we get a Kripke world $\mathcal{K}$ with a Kripke state k such that k forces $\Gamma; \Delta$ but not φ ***at*** p. Furthermore, k has at least P_0 places. Without loss of generality, we can assume that $\mathcal{K}$ does not contain any place in the set $P \setminus P_0$ (otherwise we can rename them). Now pick $p_0 \in P$, and fix it. In each Kripke state of $\mathcal{K}$ add new places $P \setminus P_0$, each duplicating p_0. It can be shown that in the resulting model the Kripke state k still forces $\Gamma; \Delta$ but not φ ***at*** p. Therefore, we obtain completeness for Kripke semantics when P is infinite. For the birelational models, we can once again use the encoding of Kripke models into birelational models.

2.7 Finite Model Property

In this section, we will show that if a judgement $\Gamma; \Delta \vdash^{P} \varphi$ ***at*** p is not provable in the logic, then there is a finite birelational model that invalidates it. The proof will use the counter-model from the proof of completeness in §2.6. The birelational model constructed in the proof of completeness consists of worlds of the form (Q, Δ, q), where (Q, Δ) are prime bounded contexts and $q \in Q$. The model constructed may be infinite as it may contain infinite many worlds. However, by using techniques similar to those used in [132], we will be able to construct a finite model that is equivalent to the counter-model. The key technique in the construction is the identification of triples (Q, Δ, q) that differ only in renaming of places other than those in P. We start the proof by discussing *renaming functions*.

2.7.1 Renaming Functions

First, we discuss renaming of places in formulae and judgements. Given any two sets of places Q_1, Q_2, a *renaming function* is a function $f : Q_1 \to Q_2$. Intuitively, f renames a place q in Q_1 as $f(q)$.

Given a renaming function $f : Q_1 \to Q_2$, we can extend f to a function from the set $Frm(Q_1)$ into the set $Frm(Q_2)$ by replacing all occurrences of places q by $f(q)$. More formally,

- $f(A) \stackrel{\text{def}}{=} A$ for all atoms A;
- $f(\varphi_1 \circ \varphi_2) \stackrel{\text{def}}{=} f(\varphi_1) \circ f(\varphi_2)$ for $\circ \in \{\vee, \wedge, \to\}$;
- $f(\varphi@q) \stackrel{\text{def}}{=} f(\varphi)@f(q)$;
- $f(\Diamond\varphi) \stackrel{\text{def}}{=} \Diamond f(\varphi)$ and $f(\Box\varphi) \stackrel{\text{def}}{=} \Box f(\varphi)$.

This can be further extended to contexts $\Gamma; \Delta$ by applying f to all formulae in Γ and all sentences in Δ, with f extended to sentences as $f(\varphi \textbf{ at } q) \stackrel{\text{def}}{=} f(\varphi) \textbf{ at } f(q)$.

If f is a renaming function, then we can transform a proof of a judgement $\Gamma; \Delta \vdash^{Q_1} \varphi \textbf{ at } q$ to a proof of the judgement $f(\Gamma; \Delta) \vdash^{Q_2} f(\varphi) \textbf{ at } f(q)$:

Lemma 12 (Provability Preservation under Renaming). *Let $f : Q_1 \to Q_2$ be a renaming function. Then for any set of pure formulae Γ, any set of sentences Δ, any formula φ and any place q such that $\mathsf{PL}(\Gamma) \cup \mathsf{PL}(\Delta) \cup \mathsf{PL}(\varphi) \cup \{q\} \subseteq Q_1$, we have:*

$$\Gamma; \Delta \vdash^{Q_1} \varphi \textbf{ at } q \textit{ implies } f(\Gamma; \Delta) \vdash^{Q_2} f(\varphi) \textbf{ at } f(q).$$

Proof. Intuitively, in order to obtain a proof of $f(\Gamma; \Delta) \vdash^{Q_2} f(\varphi) \textbf{ at } f(q)$, replace all occurrences of places r in the proof of $\Gamma; \Delta \vdash^{Q_1} \varphi \textbf{ at } q$ by $f(r)$.

More formally, we prove the lemma by induction on n, the number of inference rules applied to derive the judgement $\Gamma; \Delta \vdash^{Q_1} \varphi \textbf{ at } q$. Please note that the induction is on the number of inference rules applied, and we will vary the sets Q_i, Δ, and the formula φ in the proof. Please recall that the inference rules are given in Fig. 2.1.

Base Case ($n = 1$). Then the rule applied is one amongst L, G, and $\top I$. If the applied rule is L, then $\varphi \textbf{ at } q \in \Delta$. Hence $f(\varphi) \textbf{ at } f(q) \in f(\Delta)$. An application of the rule L gives us $f(\Gamma; \Delta) \vdash^{Q_2} f(\varphi) \textbf{ at } f(q)$. The cases of G and $\top I$ follow immediately.

Induction hypothesis ($n > 1$). We proceed by cases, and consider the last rule applied to obtain $\Gamma; \Delta \vdash^{Q_1} \varphi \textbf{ at } q$. The treatment of the rules involving the logical connectives is fairly straightforward, and we show the three most interesting cases: $@I$, $\Box I$, and $\Diamond E$.

$@I$: Assume that the last rule applied is $@I$. Then $\varphi = \psi@r$, for some pure formula $\psi \in Frm(Q_1)$ and some place $r \in Q_1$. Furthermore, $\Gamma; \Delta \vdash^{Q_1} \psi \textbf{ at } r$ is derivable by using less than n instances of the rules.

The induction hypothesis says that $f(\Gamma; \Delta) \vdash^{Q_2} f(\psi) \textbf{ at } f(r)$. Using the rule $@I$, we get $\Gamma; \Delta \vdash^{Q_2} f(\psi)@f(r) \textbf{ at } f(q)$. We conclude by observing that $f(\psi)@f(r)$ is $f(\varphi)$ by definition.

$\Box I$: Assume that the last rule applied is $\Box I$. Then $\varphi = \Box\psi$ for some pure formula $\psi \in Frm(Q_1)$. Moreover, there is a $q_1' \notin Q_1$ such that $\Gamma; \Delta \vdash^{Q_1+q_1'} \psi$ ***at*** q_1' is derivable by using less than n instances of the inference rules. Let $Q_1{}' = Q_1 \cup \{q_1'\}$. Choose $q_2' \notin Q_2$, and let $Q_2{}' = Q_2 \cup \{q_2'\}$. We define $f' : Q_1{}' \to Q_2{}'$ as $f'(r) = f(r)$ for $r \in Q_1$, and $f'(q_1') = q_2'$.

The induction hypothesis says that $f'(\Gamma; \Delta) \vdash^{Q_2+q_2'} f'(\psi)$ ***at*** q_2'. As Γ, Δ and ψ do not contain q_1', we have $f'(\Gamma; \Delta) = f(\Gamma; \Delta)$ and $f'(\psi) = f(\psi)$. Therefore, by using the inference rule $\Box I$, we get $f(\Gamma; \Delta) \vdash^{Q_2} \Box f(\psi)$ ***at*** $f(q)$. We conclude by observing that $f(\Box\psi) = \Box f(\psi)$.

$\Diamond E$: Assume that the last rule applied is $\Diamond E$. Then there exists a pure formula $\mu \in Frm(P)$, and the places $q_1', q_1'' \notin Q_1$ such that:

- $\Gamma; \Delta \vdash^{Q_1} \Diamond\mu$ ***at*** q_1'' is derivable by using less than n instances of inference rules; and
- $\Gamma; \Delta, \mu$ ***at*** $q_1' \vdash^{Q_1+q_1'} \varphi$ ***at*** q is derivable by using less than n instances of inference rules.

We get $f(\Gamma; \Delta) \vdash^{Q_2} \Diamond f(\mu)$ ***at*** $f(q_1'')$, by applying the induction hypothesis on the first judgement, .

Now, let $Q_1{}' = Q_1 \cup \{q_1'\}$ and $\Delta' = \Delta \cup \{\mu$ ***at*** $q_1'\}$. We choose $q_2' \notin Q_2$. We define $f' : Q_1' \to Q_2'$ as $f'(r) = f(r)$ for $r \in Q_1$, and $f'(q_1') = q_2'$.

We obtain that $f'(\Gamma; \Delta, \mu$ ***at*** $q_1') \vdash^{Q_2+q_2'} f'(\varphi)$ ***at*** $f'(q)$, by applying the induction hypothesis on the second judgement. Now, f' is the same as f on Q_1, and therefore by definition $f'(\Gamma; \Delta, \mu$ ***at*** $q_1') = f(\Gamma; \Delta), f(\mu)$ ***at*** q_2'. Hence, we can declare that $f(\Gamma; \Delta), f(\mu)$ ***at*** $q_2' \vdash^{Q_2+q_2'} f(\varphi)$ ***at*** q.

We conclude $f(\Gamma; \Delta) \vdash^{Q_2} f(\varphi)$ ***at*** $f(q)$, by using the inference rule $\Diamond E$. □

For example, let us consider $Q_1 = \{p, q\}$ and let $Q_2 = \{r\}$. Let $f : Q_1 \to Q_2$ be the function $f(p) = r, f(q) = r$. Let A be an atom, and let Γ to be the empty list. We have $\Gamma; A$ ***at*** $p \vdash^{Q_1} A@p$ ***at*** q. Then by the Lemma 12, $\Gamma; A$ ***at*** $r \vdash^{Q_2} A@r$ ***at*** r.

2.7.2 Pointed Contexts and Morphisms

Let P, Q be a finite sets of places such that $P \subseteq Q$. Let $\Theta \subseteq Frm(P)$ be a finite set of pure formulae with sub-formula closure Θ^*. Please recall that given a finite set of sentences Δ, we say that (Q, Δ) is a $(P, \Theta)-$bounded context if for every sentence φ ***at*** r it is the case that $\varphi \in \Theta^*$ and $r \in Q$. Given a $(P, \Theta)-$bounded context (Q, Δ), we will say that (Q, Δ, q) is a *pointed* $(P, \Theta)-$*bounded context* if $q \in Q$. Henceforth, we refer to such triples as $(P, \Theta)-$*pcontexts*. The element q is said to be *the point* of the pcontext (Q, Δ, q). Following [132], we lift the notion of renaming functions to morphisms between pcontexts:

Definition 19 (Morphism). Let w_1 and w_2 be two $(P, \Theta)-$pcontexts, and for $i = 1, 2$ let $w_i = (Q_i, \Delta_i, q_i)$. *A morphism* from w_1 to w_2 is a renaming function $f : Q_1 \to Q_2$ such that

1. $f(p) = p$ for every $p \in P$;
2. if φ ***at*** $q \in \Delta_1$ then φ ***at*** $f(q) \in \Delta_2$; and
3. $f(q_1) = q_2$.

We write $w_1 \precsim w_2$ whenever there is a morphism from w_1 to w_2. Furthermore, we write $w_1 \backsimeq w_2$ if $w_1 \precsim w_2$ and $w_2 \precsim w_1$.

The first part of the definition says that the renaming function does not change the places in P. Now for every sentence φ ***at*** $q \in \Delta_1$, it is the case that $\varphi \in Frm(P)$. Therefore, the second condition is equivalent to saying that $f(\Delta_1) \subseteq \Delta_2$. Hence, $(Q_1, \Delta_1, q_1) \precsim (Q_2, \Delta_2, q_2)$ intuitively means that Δ_2 has "more" sentences than Δ_1 up-to renaming. Finally, the third part says that a morphism preserves the point of a pcontext.

For example, let $P = \{p\}$, $\Theta = \{A\}$, and $Q_1 = Q_2 = \{p, q, r\}$. Let $f : Q_1 \to Q_2$ be the renaming function defined as $f(p) = p$, $f(q) = r$ and $f(r) = q$. Consider the three sets of sentences:

- $\Delta_1 = \Delta_2 = \{A$ ***at*** q, A ***at*** $p\}$, and
- $\Delta' = \{A$ ***at*** p, A ***at*** $r\}$.

We have $f(A$ ***at*** $q) = A$ ***at*** r. Now, we have that A ***at*** $r \notin \Delta_2$ and A ***at*** $r \in \Delta'$. Therefore, f is not a morphism from (Q_1, Δ_1) to (Q_2, Δ_2). On the other hand, f is a morphism from (Q_1, Δ_1) to (Q_2, Δ').

Clearly, $\precsim$ is a preorder. The identity function gives reflexivity, and function composition gives transitivity. This makes the relation $\backsimeq$ an equivalence relation. If w is a pcontext, then we shall use $[w]$ to denote the class of the pcontexts equivalent to w with respect to the relation $\backsimeq$. We shall use these equivalence classes as the worlds of the finite counter-model, and the order amongst the worlds will be given by the preorder $\precsim$. We will now show that the relation $\backsimeq$ partitions the set of pcontexts into finite number of classes. Please note that it is in this proof, we use the fact that the set P is finite:

Lemma 13 (Finite Partition). *The set of* $(P, \Theta)-$*pcontexts is partitioned into a finite number of equivalence classes by the equivalence* $\backsimeq$.

Proof. We will show that every $(P, \Theta)-$pcontext is equivalent to a *canonical pcontext*. The set of canonical pcontexts will be finite. Before we proceed, please note that P and Θ are finite sets by definition. Hence, the sub-formula closure Θ^* and the powerset $Pow(\Theta^*)$ must be finite sets.

We will now define the set of canonical pcontexts. For each $\Lambda \subseteq \Theta^*$ we choose a new place $\boldsymbol{r}_\Lambda \notin P$ such that $\boldsymbol{r}_{\Lambda_1} \neq \boldsymbol{r}_{\Lambda_2}$ if $\Lambda_1 \neq \Lambda_2$. Let $R \stackrel{\text{def}}{=} \{\boldsymbol{r}_\Lambda : \Lambda \subseteq \Theta^*\}$. The cardinality of R is the same as the cardinality of $Pow(\Theta^*)$, and hence R is finite. A canonical pcontext

will have places amongst $P \cup R$. Furthermore, the canonical pcontext will contain the sentence φ ***at*** $\boldsymbol{r}_{\Lambda}$ if and only if $\boldsymbol{r}_{\Lambda}$ is a place in the pcontext and $\varphi \in \Lambda$. More formally, we say that the triple (Q, Σ, q) is a *canonical* (P, Θ)-*pcontext* if

- Q is a set of places such that $P \subseteq Q \subseteq P \cup R$.
- Δ is the union of two sets Δ_P and Δ_R, where
  1. Δ_P is a set of sentences such that φ ***at*** $s \in \Delta_P$ means that $\varphi \in \Theta^*$ and $s \in P$; and
  2. Δ_R is the set of *all* sentences φ ***at*** $\boldsymbol{r}_{\Lambda}$, where $\varphi \in \Lambda$ and $\boldsymbol{r}_{\Lambda} \in Q \cap R$. In other words, $\Delta_R \stackrel{\text{def}}{=} \{\varphi \textbf{ at } \boldsymbol{r}_{\Lambda} : \varphi \in \Lambda,\ \boldsymbol{r}_{\Lambda} \in Q \cap R\}$.
- $q \in Q$.

Clearly, a triple that satisfies the above points is a $(P, \Theta)-$pcontext. Furthermore, as the sets P, R, Θ^* are finite, the set of canonical pcontexts must be finite also.

We will now show that for every pcontext $w = (Q, \Delta, q)$ there is a canonical pcontext equivalent to it. This would immediately give us that the number of equivalence classes induced by $\backsimeq$ is finite.

Let $w = (Q, \Delta, q)$ be a $(P, \Theta)-$pcontext, and fix it. For $s \in Q$, let $H(s) \subseteq \Theta^*$ be the set of formulae φ such that φ ***at*** $s \in \Delta$.

We now define $w' = (Q', \Delta', q')$, the canonical pcontext equivalent to w as follows. P will be contained in Q'. For each $s \in Q \setminus P$, we add the place $\boldsymbol{r}_{H(s)}$ to Q'. For $p \in P$, a sentence φ ***at*** p will be in Δ' only if it is in Δ. A sentence φ ***at*** $\boldsymbol{r}_{H(s)}$ will be in Q' only if $\varphi \in H(s)$. Finally, the point q' will be q if $q \in P$. Otherwise the point q' will be $\boldsymbol{r}_{H(q)}$. More formally, we define:

- $Q' \stackrel{\text{def}}{=} P \cup \{\boldsymbol{r}_{H(s)} : s \in Q \setminus P\}$
- $\Delta' \stackrel{\text{def}}{=} \Delta_P \cup \Delta_R$, where
  - $\Delta_P \stackrel{\text{def}}{=} \{\varphi \textbf{ at } p : \varphi \textbf{ at } p \in \Delta \text{ and } p \in P\}$
  - $\Delta_R \stackrel{\text{def}}{=} \{\varphi \textbf{ at } \boldsymbol{r}_{H(s)} : s \in Q \setminus P \text{ and } \varphi \in H(s)\}$
- $q' \stackrel{\text{def}}{=} \begin{cases} q & \text{if } q \in P; \\ \boldsymbol{r}_{H(q)} & \text{if } q \in Q \setminus P. \end{cases}$

Clearly, (Q', Δ', q') is a canonical $(P, \Theta)-$pcontext. Moreover, the renaming functions

$$f : Q \longrightarrow Q' \qquad f(s) \stackrel{\text{def}}{=} \begin{cases} s & \text{if } s \in P; \\ \boldsymbol{r}_{H(s)} & \text{otherwise.} \end{cases}$$

$$g : Q' \longrightarrow Q \qquad g(t) \stackrel{\text{def}}{=} \begin{cases} t & \text{if } t \in P; \\ q & \text{if } t = q'; \\ l & \text{otherwise, where } l \in Q \setminus P \text{ is chosen s.t.} \\ & t = \boldsymbol{r}_{H(l)}. \end{cases}$$

are morphisms from w to w' and from w' to w, respectively. We conclude that $w \backsimeq w'$. □

2.7.3 The Finite Counter-Model

Given a finite set of places P, two finite sets of pure formulae $\Gamma, \Theta \subseteq Frm(P)$, let $\mathcal{K}_{can}$ be the $\Gamma-$prime and $(P,\Theta)-$bounded canonical Kripke model as defined in §2.6 (see Definition 18). Now, let $\mathcal{W}_{can} = (W, \leq, R, I, Eval)$ be the $\mathcal{K}_{can}-$ birelational model obtained by using the encoding of $\mathcal{K}_{can}$ into a birelational model (see §2.5.2). We call $\mathcal{W}_{can}$ the $\Gamma-$prime and $(P,\Theta)-$bounded canonical birelational model. Please recall from the proof of completeness (see §2.6) that if a judgement $\Gamma; \Sigma \vdash^P \varphi \ \textbf{\textit{at}}\ p$ is not provable, then $\mathcal{W}_{can}$ provides the birelational counter-model for the judgement for an appropriate choice of Θ.

The worlds of $\mathcal{W}_{can}$ are pcontexts (Q, Δ, q) where (Q, Δ) are contexts $\Gamma-$prime and $(P,\Theta)-$bounded. Two worlds $w_1 = (Q_1, \Delta_1, q_1)$ and $w_2 = (Q_2, \Delta_2, q_2)$ are reachable from each other if $Q_1 = Q_2$ and $\Delta_1 = \Delta_2$. Furthermore, $(Q_1, \Delta_1, q_1) \leq (Q_2, \Delta_2, q_2)$ if $Q_1 \subseteq Q_2$, $\Delta_1 \subseteq \Delta_2$ and $q_1 = q_2$. A world $w = (Q, \Delta, q) \in I(A)$ for some atom A if $A \ \textbf{\textit{at}}\ q \in \Delta$. The evaluation is a total function, and $E((Q, \Delta, q)) = q$. Furthermore, as a consequence of definition of canonical models, a world $w = (Q, \Delta, q)$ forces a formula $\varphi \in \Theta^*$ if and only if $\varphi \ \textbf{\textit{at}}\ q \in \Delta$.

Even though the worlds in canonical birelational are composed of bounded pcontexts, the set of the worlds may itself be infinite. Following [132], we shall construct a model, called the *quotient model*, equivalent to the canonical model. For this model, we will use morphisms between pcontexts. Please recall that given pcontexts w_1 and w_2, $w_1 \precsim w_2$ if there is a morphism from w_1 into w_2, and $w_1 \backsimeq w_2$ if $w_1 \precsim w_2$ and $w_2 \precsim w_1$. The relation $\precsim$ is a preorder and $\backsimeq$ is an equivalence. The set of equivalence classes generated by $\backsimeq$ is finite by Lemma 13. We write $[w]$ for the equivalence class of w.

In the quotient canonical model, the set of worlds will be $W_{/\backsimeq}$, the set of equivalence classes generated by $\backsimeq$ on W. We have that $W_{/\backsimeq}$ is finite. Our construction will ensure that w in the canonical birelational model forces a formula $\varphi \in \Theta^*$ only if $[w]$ forces φ.

In the quotient model, $[w_1]$ will be less than $[w_2]$ only if $w_1 \precsim w_2$. As $\precsim$ is a preorder, it follows easily that this ordering is well-defined. If R is the reachability relation on the canonical model, then $[w_1]$ is reachable from $[w_2]$ in the quotient model only if there is some $w_1' \in [w_1]$ and $w_2' \in [w_2]$ such that $w_1' \, R \, w_2'$. The equivalence of $\backsimeq$ ensures that reachability relation is well-defined. If I is the interpretation of atoms in the canonical model and $w = (Q, \Delta, q)$, then an atom A will be placed in a world $[w]$ only if $A \ \textbf{\textit{at}}\ q \in \Delta$. Since a morphism between pcontexts always preserves points, the interpretation function is also well-defined.

Finally, the evaluation of a world $[w]$ in the canonical model will be *partial*. It is defined only if the point of w is in P, and in that case the evaluation of $[w]$ is the point of w. Please note that morphisms between pcontexts always fixes elements in P, and therefore the evaluation is also well-defined. Moreover, *partiality* is essential for the well-definedness of the evaluation as a morphism of pcontexts may not preserve places other than those in P.

We start by defining the quotient model formally, and show that this is indeed a birelational model.

Definition 20 (Quotient Canonical Model). Given a finite set of places P, two finite sets of pure formulae $\Gamma, \Theta \subseteq \mathit{Frm}(P)$, let $\mathcal{W}_{can} = (W, \leq, R, I, \mathit{Eval})$ be the $\Gamma-$prime and $(P, \Theta)-$ bounded canonical birelational model with set of places Pls. The *quotient model of* $\mathcal{W}_{can}$ has set of places P, and is defined to be the quintuple $(W_{/\simeq}, \leq', R', I', \mathit{Eval}')$, where

1. The set $W_{/\simeq}$ is the set of the equivalence classes generated by the relation $\simeq$ on W.
2. The binary relation $\leq'$ is defined as: $[w_1] \leq' [w_2]$ if and only if $w_1 \lesssim w_2$.
3. The binary relation R' is defined as: $[w_1] R' [w_2]$ if and only if there exists $w_1' \in [w_1]$ and $w_2' \in [w_2]$ such that $w_1' R w_2'$.
4. The function $I' : \mathit{Atoms} \to \mathit{Pow}(W_{/\simeq})$ is defined as:

$$I'(A) \stackrel{\text{def}}{=} \{[w] : w \in I(A)\}$$

5. The partial function $\mathit{Eval}' : W_{/\simeq} \to P$ is defined as:

$$\mathit{Eval}'([w]) \stackrel{\text{def}}{=} \begin{cases} p & \text{if } w = (Q, \Delta, p) \text{ and } p \in P; \\ \text{not defined} & \text{otherwise.} \end{cases}$$

As we discussed before, $\leq'$, R', I' and Eval' in the quotient model are well-defined. We show that the relation R' is an equivalence:

Lemma 14 (Reachability is an Equivalence). *Given a finite set of places P, two finite sets of pure formulae $\Gamma, \Theta \subseteq \mathit{Frm}(P)$, let $\mathcal{W}_{can} = (W, \leq, R, I, \mathit{Eval})$ be the $\Gamma-$prime and $(P, \Theta)-$bounded canonical birelational model. Let $\mathcal{W}_{/\simeq} = (W_{/\simeq}, \leq', R', I', \mathit{Eval}')$ be the quotient model of $\mathcal{W}_{can}$. Then R' is an equivalence.*

Proof. The reflexivity and symmetry of R' follow from the reflexivity and symmetry of R in the model $\mathcal{W}_{can}$. We need to show that R' is transitive.

Pick $[w_1], [w_2], [w_3] \in W_{/\simeq}$ such that $[w_1] R' [w_2] R' [w_3]$, and fix them. By definition, the assumption $[w_1] R' [w_2] R' [w_3]$ is equivalent to saying that there are $w_1', w_2', w_2'', w_3' \in W$ such that $w_1 \simeq w_1' R w_2' \simeq w_2$ and $w_2 \simeq w_2'' R w_3' \simeq w_3$. As $\simeq$ is an equivalence, we get

$$w_1' R w_2' \simeq w_2'' R w_3'. \tag{2.7}$$

In order to prove transitivity, we will first show that there are two worlds v_1 and v_3 in W such that $w_1' \simeq v_1 R v_3 \simeq w_3'$. This will give us by definition $[w_1'] R' [w_3']$, and hence $[w_1] R' [w_3]$.

Now, the assumptions in (2.7) and the definition of R say that

1. $w'_1 = (Q_1, \Delta_1, q_1)$ and $w'_2 = (Q_1, \Delta_1, q_2)$, where (Q_1, Δ_1) is a context Γ-prime and (P, Θ)−bounded, and $q_1, q_2 \in Q_1$.

2. $w''_2 = (Q_2, \Delta_2, q'_2)$ and $w''_3 = (Q_2, \Delta_2, q_3)$, where (Q_2, Δ_2) is a context Γ-prime and (P, Θ)−bounded, and $q'_2, q_3 \in Q_2$.

3. $(Q_1, \Delta_1, q_2) \backsimeq (Q_2, \Delta_2, q'_2)$, *i.e.,* there exist two morphisms $f : Q_1 \to Q_2$ and $g : Q_2 \to Q_1$ such that $f(q_2) = q'_2$ and $g(q'_2) = q_2$.

Without loss of generality, we can assume that $Q_1 = P \cup R_1$ and $Q_2 = P \cup R_2$ with $R_1 \cap R_2 = \emptyset$ (otherwise, we can rename the places in Δ_2 and R_2).

$(Q_1 \cup Q_2, \Delta_1 \cup \Delta_2)$ is (P, Θ)−bounded as (Q_1, Δ_1) and (Q_2, Δ_2) are bounded contexts. We let $v_1 \stackrel{\text{def}}{=} (Q_1 \cup Q_2, \Delta_1 \cup \Delta_2, q_1)$ and $v_3 \stackrel{\text{def}}{=} (Q_1 \cup Q_2, \Delta_1 \cup \Delta_2, q_3)$.

Now, consider the triple $v_1 = (Q_1 \cup Q_2, \Delta_1 \cup \Delta_2, q_1)$. We have $(Q_1 \cup Q_2, \Delta_1 \cup \Delta_2, q_1) \backsimeq (Q_1, \Delta_1, q_1)$, by considering the two renaming functions

$$G_1 : \; Q_1 \cup Q_2 \longrightarrow Q_1 \qquad\qquad G_2 : \; Q_1 \longrightarrow Q_1 \cup Q_2$$
$$G_1(q) \stackrel{\text{def}}{=} \begin{cases} q & \text{if } q \in Q_1; \\ g(q) & \text{if } q \in Q_2 \end{cases} \qquad\qquad G_2(q) \stackrel{\text{def}}{=} q$$

Please note that as g is a morphism, $g(q) = q$ if $q \in Q_1 \cap Q_2 = P$. Therefore, G_1 is well-defined and $G_1(q_1) = q_1$. Now, suppose that $\varphi \; \mathbf{at} \; q \in \Delta_1 \cup \Delta_2$. If $\varphi \; \mathbf{at} \; q \in \Delta_1$, then $\varphi \; \mathbf{at} \; G_1(q) \in \Delta_1$ as $G_1(q) = q$ in that case. If $\varphi \; \mathbf{at} \; q \in \Delta_2$, then $\varphi \; \mathbf{at} \; G_1(q) \in \Delta_1$ because in this case $G_1(q) = g(q)$ and g is a morphism. Therefore, G_1 is a morphism of pcontexts. G_2 is a morphism between pcontexts trivially, and hence we get $w'_1 \backsimeq v_1$.

Similarly, $(Q_1 \cup Q_2, \Delta_1 \cup \Delta_2, q_3) \backsimeq (Q_2, \Delta_2, q_3)$ by considering the morphisms

$$F_1 : \; Q_1 \cup Q_2 \longrightarrow Q_2 \qquad\qquad F_2 : \; Q_2 \longrightarrow Q_1 \cup Q_2$$
$$F_1(q) \stackrel{\text{def}}{=} \begin{cases} f(q) & \text{if } q \in Q_1; \\ q & \text{if } q \in Q_2 \end{cases} \qquad\qquad F_2(q) \stackrel{\text{def}}{=} q$$

We get that $v_3 \backsimeq w'_3$.

If v_1 and v_3 are worlds in $\mathcal{W}_{can}$, then $v_1 \, R \, v_3$ by definition. In that case v_1 and v_3 are the worlds we are looking for. In order to show that v_1 and v_3 are indeed worlds in $\mathcal{W}_{can}$ we need to show that the (P, Θ)−bounded context $(Q_1 \cup Q_2, \Delta_1 \cup \Delta_2)$ is Γ-prime.

In order to show that $(Q_1 \cup Q_2, \Delta_1 \cup \Delta_2)$ is Γ−prime we need to show the four properties required by Definition 17. We will prove here only the Θ-deductive closure property. The treatment of other properties is similar.

Assume that $\Gamma; \Delta_1 \cup \Delta_2 \vdash^{Q_1 \cup Q_2} \varphi \; \mathbf{at} \; q$ for some $\varphi \in \Theta$. We consider two cases. If $q \in Q_1$, then consider the renaming function G_1 defined above. Now G_1 fixes Q_1 and applies g to Q_2. Therefore, $G_1(\Gamma) = \Gamma$, $G_1(\Delta_1 \cup \Delta_2) = \Delta_1 \cup g(\Delta_2)$, $G_1(\varphi) = \varphi$ and $G_1(q) = q$. Now, as g is a morphism we get that $g(\Delta_2) \subseteq \Delta_1$. Therefore, using Lemma 12 and applying the renaming function G_1 to the judgement $\Gamma; \Delta_1 \cup \Delta_2 \vdash^{Q_1 \cup Q_2} \varphi \; \mathbf{at} \; q$, we get that $\Gamma; \Delta_1 \vdash^{Q_1} \varphi \; \mathbf{at} \; q$. As Δ_1 is Γ-prime, $\varphi \; \mathbf{at} \; q \in \Delta_1 \subseteq \Delta_1 \cup \Delta_2$. Likewise, if $q \in Q_2$, we conclude that $\varphi \; \mathbf{at} \; q \in \Delta_2 \subseteq \Delta_1 \cup \Delta_2$. □

We now show that the quotient model is a birelational model.

Proposition 24 (Birelational Preservation). *Consider $\mathcal{W}_{can} = (W, \leq, R, I, Eval)$, the $\Gamma-$prime and $(P,\Theta)-$bounded canonical birelational model with set of places Pls. Let $\mathcal{W}_{/\backsimeq} = (W_{/\backsimeq}, \leq', R', I', Eval')$ be the quotient model of $\mathcal{W}_{can}$. Then $\mathcal{W}_{/\backsimeq}$ is a finite birelational model with set of places P.*

Proof. The finiteness of $\mathcal{W}_{/\backsimeq}$ follows from Lemma 13. We need to verify all the properties listed in Definition 12.

1. Clearly $W_{/\backsimeq}$ is a non empty set.

2. The relation $\leq'$ is a partial order since $\precsim$ is a preorder, and $\backsimeq$ is the equivalence induced by $\precsim$.

3. R' is an equivalence by Lemma 14. We prove the reachability condition. Consider $[w_1], [w_1'], [w_2] \in W_{/\backsimeq}$ such that $[w_2] \geq' [w_1] R' [w_1']$. We need to prove that there exists $[w_2'] \in W_{/\backsimeq}$ such that $[w_2] R' [w_2'] \geq' [w_1']$.

   Now, the hypothesis $[w_2] \geq' [w_1] R' [w_1']$ means:

   - $w_1 = (Q_1, \Delta_1, q_1)$ and $w_1' = (Q_1, \Delta_1, q_1')$ where (Q_1, Δ_1) is a $\Gamma-$prime and $(P,\Theta)-$bounded context, and $q_1, q_1' \in Q_1$;
   - $w_2 = (Q_2, \Delta_2, q_2)$ where (Q_2, Δ_2) is a $\Gamma-$prime and $(P,\Theta)-$bounded context, and $q_2 \in Q_2$; and
   - there is a morphism $f : Q_1 \to Q_2$ from w_1 to w_2.

   We define $w_2' \stackrel{\text{def}}{=} (Q_2, \Delta_2, f(q_1'))$. Clearly $w_2 \in W$, $w_2 R w_2'$, and f is also a morphism from w_1' to w_2'. Therefore $[w_2] R' [w_2'] \geq' [w_1']$, as required.

4. In order to check the monotonicity of I', consider $[w_1], [w_2] \in W_{/\backsimeq}$ such that $[w_1] \leq' [w_2]$. Then $w_1 = (Q_1, \Delta_1, q_1)$, $w_2 = (Q_2, \Delta_2, q_2)$, and there exists a morphism f from w_1 to w_2 such that $f(q_1) = q_2$.

   We need to prove that if $[w_1] \in I'(A)$, then $[w_2] \in I'(A)$ also. Now assume that $[w_1] \in I'(A)$. By definition, this means that A ***at*** $q_1 \in \Delta_1$. As f is a morphism, we get A ***at*** $f(q_1) \in \Delta_2$, and hence A ***at*** $q_2 \in \Delta_2$. Therefore $[w_2] \in I'(A)$ as required.

5. According to the definition, $Eval'$ is a partial function. We need to verify coherence and uniqueness.

   Coherence. Consider $[w_1], [w_2] \in W_{/\backsimeq}$ such that $[w_1] \leq' [w_2]$, and assume that $[w_1]{\downarrow} q$. Then $q \in P$, and $w_1 = (Q_1, \Delta_1, q)$ for some Q_1, Δ_1. $[w_1] \leq' [w_2]$ means that is a morphism from w_1 to w_2 that fixes q. Therefore, $w_2 = (Q_2, \Delta_2, q)$ for some Q_2 and Δ_2. By definition, we conclude that $[w_2]{\downarrow} q$.

Uniqueness Consider $[w_1], [w_2] \in W_{/\preceq}$ such that $[w_1] R' [w_2]$. This means that there exist $w_1', w_2' \in W$ such that $w_1 \preceq w_1' R w_2' \preceq w_2$. Assume that $[w_1]{\downarrow} q$ and $[w_2]{\downarrow} q$. Then $w_1'{\downarrow} q$ and $w_2'{\downarrow} q$ in $\mathcal{W}_{can}$. The uniqueness property in $\mathcal{W}_{can}$ says that $w_1' = w_2'$. Hence $w_1 \preceq w_1' \preceq w_2$. We conclude $[w_1] = [w_2]$ as required. □

We will show that a world w forces a formula in Θ^* in the canonical birelational model if and only if $[w]$ forces the formula in the quotient model. For this, we will need the following proposition which states that given worlds $w_1 \precsim w_2$ in the canonical model, if w_1 forces a formula in Θ^* then so does w_2:

Proposition 25 (Forcing Preservation under Morphisms). *Given a finite set of places P, two finite sets of pure formulae $\Gamma, \Theta \subseteq Frm(P)$, let $\mathcal{W}_{can} = (W, \leq, R, I, Eval)$ be the $\Gamma-$prime and $(P,\Theta)-$ bounded canonical birelational model. Let $\models_{\mathcal{W}}$ be the extension of interpretation I to formulae. Then for every $w_1, w_2 \in W$, and $\varphi \in \Theta^*$:*

1. *If $w_1 \precsim w_2$, then $w_1 \models_{\mathcal{W}} \varphi$ implies $w_2 \models_{\mathcal{W}} \varphi$.*
2. *If $w_1 \preceq w_2$, then $w_1 \models_{\mathcal{W}} \varphi$ if and only if $w_2 \models_{\mathcal{W}} \varphi$.*

Proof. We prove the first point as the second one is straightforward consequence of the first one. Consider $w_1, w_2 \in W$, such that $w_1 \precsim w_2$. This means that $w_1 = (Q_1, \Delta_1, q_1)$ and $w_2 = (Q_2, \Delta_2, q_2)$ where (Q_i, Δ_i) are Γ-prime and (P,Θ)-bounded contexts for $i = 1, 2$. Moreover, there is a morphism $f : Q_1 \to Q_2$ such that $f(q_1) = q_2$.

Assume that $w_1 \models_{\mathcal{W}} \varphi$ for some $\varphi \in \Theta^*$. This means from the definition of canonical birelational model that φ ***at*** $q_1 \in \Delta_1$. Since f is a morphism from w_1 to w_2, we get that φ ***at*** $q_2 \in \Delta_2$. Once again, we get from the definition of canonical birelational model that $w_2 \models_{\mathcal{W}} \varphi$. □

We are now ready to prove that if the world w in the canonical birelational model forces $\varphi \in \Theta^*$, then the world $[w]$ in the quotient model also forces φ, and vice-versa.

Lemma 15 (Quotient Forcing Preservation). *Given a finite set of places P, two finite sets of pure formulae $\Gamma, \Theta \subseteq Frm(P)$, let $\mathcal{W}_{can} = (W, \leq, R, I, Eval)$ be the $\Gamma-$prime and $(P,\Theta)-$ bounded canonical birelational model. Let $\mathcal{W}_{/\preceq} = (W_{/\preceq}, \leq', R', I', Eval')$ be the quotient model of $\mathcal{W}_{can}$. Let $\models_{\mathcal{W}}$ and $\models_{/\preceq}$ extend the interpretations I and I' to formulae respectively. Then, for every $\varphi \in \Theta^*$ and $w \in W$:*

$$w \models_{\mathcal{W}} \varphi \text{ if and only if } [w] \models_{/\preceq} \varphi.$$

Proof. The proof proceeds by induction on the structure of the formula $\varphi \in \Theta^*$.

Base case. The lemma is verified on $\top$, and on $\bot$ by definition. Consider now the case when $\varphi = A \in Atoms$. Then $w \models_{\mathcal{W}} A$ means $w = (Q, \Delta, q)$ for some Q, Δ, q and A ***at*** $q \in \Delta$. Hence, $[w] \in I'(A)$, and therefore $[w] \models_{/\preceq} A$.

Induction hypothesis. We consider a formula $\varphi \in \Theta^*$, and we assume that the lemma holds for each sub-formula of φ that is in Θ^*. We will proceed by cases on the structure of

φ. For the sake of clarity, we will just consider the case of implication and the modalities. The other cases can be dealt with similarly. Please note that as Θ^* is closed under sub-formulae, the induction hypothesis can be applied to all sub-formulae of φ.

Before we proceed with the cases, we observe that if $w_1 = (Q_1, \Delta_1, q_1)$ and $w_2 = (Q_2, \Delta_2, q_2)$ are two worlds in W such $w_1 \leq w_2$, then $w_1 \preceq w_2$. This is because by definition $w_1 \leq w_2$ means that $Q_1 \subseteq Q_2$, $\Delta_1 \subseteq \Delta_2$ and $q_1 = q_2$. The morphism between w_1 and w_2 is given by the injection of Q_1 into Q_2.

Case $\varphi = \varphi_1 \to \varphi_2$**.** Let $w \models_{\mathcal{W}} \varphi$. We need to show that $[w] \models_{/\preceq} \varphi$. Consider $[w'] \geq' [w]$. Then $w' \succsim w$. By Proposition 25, we have $w' \models_{\mathcal{W}} \varphi$. As $\varphi = \varphi_1 \to \varphi_2$, we get that $w' \models_{\mathcal{W}} \varphi_2$ whenever $w' \models_{\mathcal{W}} \varphi_1$.

If we assume $[w'] \models_{/\preceq} \varphi_1$ then $w' \models_{\mathcal{W}} \varphi_1$ by induction hypothesis. Hence $w' \models_{\mathcal{W}} \varphi_2$. The induction hypothesis says that $[w'] \models_{/\preceq} \varphi_2$. As $[w']$ is an arbitrary world larger that $[w]$, we can conclude that $[w] \models_{/\preceq} \varphi_1 \to \varphi_2$.

For the other direction, let $[w] \models_{/\preceq} \varphi$. This means that for every $[w'] \geq' [w]$: if $[w'] \models_{/\preceq} \varphi_1$, then $[w'] \models_{/\preceq} \varphi_2$.

Consider now $w' \geq' w$. We have $[w'] \succsim [w]$ also. If we assume $w' \models_{\mathcal{W}} \varphi_1$, then the induction hypothesis says that $[w'] \models_{/\preceq} \varphi_1$. Then $[w'] \models_{/\preceq} \varphi_2$, and so $w' \models_{\mathcal{W}} \varphi_2$ by induction hypothesis. We conclude that $w \models_{\mathcal{W}} \varphi_1 \to \varphi_2$.

Case $\varphi = \Box\varphi_1$**.** Let $w \models_{\mathcal{W}} \varphi$. We need to show that $[w] \models_{/\preceq} \Box\varphi_1$. Consider $[w_1] \geq' [w]$ and $[w_2] R' [w_1]$. It suffices to show that $[w_2] \models_{/\preceq} \varphi_1$. The hypothesis $[w_2] R' [w_1] \geq' [w]$ means that $w_1 \succsim w$ and $w_2 \preceq w_3 R w_4 \preceq w_1$ for some worlds $w_3, w_4 \in W$. We get that $w_4 \succsim w$ as $\preceq$ is a preorder .

We have $w_4 \succsim w$, and hence $w_4 \models_{\mathcal{W}} \Box\varphi_1$ by Proposition 25. By definition of forcing, $w_3 \models_{\mathcal{W}} \varphi_1$. Therefore $w_2 \models_{\mathcal{W}} \varphi_1$ by Proposition 25. The induction hypothesis says that $[w_2] \models_{/\preceq} \varphi_1$, and so we conclude $[w] \models_{/\preceq} \Box\varphi_1$.

For the other direction, let $[w] \models_{/\preceq} \Box\varphi_1$. Consider $w_1 \geq w$ and $w_2 R w_1$. We have to show that $w_2 \models \varphi_1$.

We have $w_1 \succsim w$, and hence $[w_1] \geq [w]$. We also have by the definition of the quotient model that $[w_2] R' [w_1]$. Therefore, as $[w] \models_{/\preceq} \Box\varphi_1$, we get that $[w_2] \models_{/\preceq} \varphi_1$. Hence $w_2 \models_{\mathcal{W}} \varphi_1$ by induction hypothesis. We conclude that $w \models_{\mathcal{W}} \Box\varphi_1$.

Case $\varphi = \Diamond\varphi_1$**.** Let $w \models_{\mathcal{W}} \varphi$. Then there exists $w_1 R w$ such that $w_1 \models_{\mathcal{W}} \varphi_1$. So we have $[w_1] R' [w]$ by the definition of quotient model. Also $[w_1] \models_{/\preceq} \varphi_1$ by induction hypothesis. Hence $[w] \models_{/\preceq} \Diamond\varphi_1$.

For the other direction, let $[w] \models_{/\preceq} \varphi$. Then there exists $[w_1] R' [w]$ such that $[w_1] \models_{/\preceq} \varphi_1$. This means that there are w_1' and w' such that $w_1 \preceq w_1' R w' \preceq w$, and $w_1 \models_{\mathcal{W}} \varphi_1$ by induction hypothesis. By Proposition 21, we get that $w_1' \models_{\mathcal{W}} \varphi_1$. Therefore, by definition of forcing, $w' \models_{\mathcal{W}} \Diamond\varphi_1$. By Proposition 21 once again, $w \models_{\mathcal{W}} \Diamond\varphi_1$.

Case $\varphi = \varphi_1@q$**.** As $\varphi \in \Theta^*$ and $\Theta^* \subseteq \mathit{Frm}(P)$, we get that $q \in P$.

Now, if $w \models_{\mathcal{W}} \varphi$ then there exists $w_1 R w$ such that $w_1 \models_{\mathcal{W}} \varphi_1$ and $w_1{\downarrow}\, q$. We have $[w_1] R' [w]$ by definition of quotient model. As $q \in P$, we also have $[w_1]{\downarrow}\, q$. Therefore, $[w] \models_{/\preceq} \varphi_1@q$.

For the other direction, let $[w] \models_{/\preceq} \varphi$. Then there exists $[w_1] R' [w]$ such that $[w_1] \models_{/\preceq} \varphi_1$, and $[w_1]{\downarrow}\, q$. This means that there are w_1' and w' such that $w_1 \preceq w_1' R w' \preceq w$, and $w_1 \models_{\mathcal{W}} \varphi_1$ by induction hypothesis. Furthermore, $w_1{\downarrow}\, q$ and $w_1'{\downarrow}\, q$. By Proposition 21, we get that $w_1' \models_{\mathcal{W}} \varphi_1$. Hence, by definition of forcing, $w' \models_{\mathcal{W}} \varphi_1@q$. By Proposition 21 once again, $w \models_{\mathcal{W}} \varphi_1@q$. □

As a result of Lemma 15, we have a way to going from a canonical model to an equivalent finite model. As shown above, the canonical model forces a formula if and only if its finite quotient does, and we get finite model property:

Theorem 22 (Finite Model Property). *Assume that P is a finite set of places. If the judgement $\Gamma; \Delta \vdash^P \varphi$* **at** *$p$ is not provable, then there exists a* finite birelational model *$\mathcal{W}$ with set of places P, such that $\Gamma; \Delta \vdash^P \varphi$* **at** *$p$ is not valid in $\mathcal{W}$.*

Proof. We fix $\Theta \stackrel{\text{def}}{=} \{\Box\psi; \psi \in \Gamma\} \cup \Gamma \cup \{\psi : \psi \,\textbf{\textit{at}}\, q \in \Delta\} \cup \mathsf{PL}(\varphi) \cup \{p\}$. Consider the Γ-prime and $(P, \Theta)-$bounded canonical birelational model $\mathcal{W}_{can}$. From the proof of completeness in §2.6 there is a world of $\mathcal{W}_{can}$, say w, such that w evaluates to P and w forces $\Gamma; \Delta$ but not φ.

Consider the quotient $\mathcal{W}_{/\preceq}$ of $\mathcal{W}_{can}$. $\mathcal{W}_{/\preceq}$ is a finite birelational model and has set of places P. The world $[w]$ evaluates to p. Furthermore, as a consequence of Lemma 15, we can easily show that $[w]$ forces $\Gamma; \Delta$ but not φ. Therefore, $\mathcal{W}_{/\preceq}$ is the required finite counter-model. □

Decidability is based on Harrop criterion, cf. [80], saying that every finitely axiomatisable modal logic with the finite model property is decidable.

Corollary 6 (Decidability). *The provability of the judgement $\Gamma; \Delta \vdash^P \varphi$* **at** *$p$ is decidable in the logic.*

Proof. Let P' be $\mathsf{PL}(\Gamma) \cup \mathsf{PL}(\Delta) \cup \mathsf{PL}(\varphi) \cup \{p\}$. By Proposition 23, $\Gamma; \Delta \vdash^P \varphi \,\textbf{\textit{at}}\, p$ if and only if $\Gamma; \Delta \vdash^{P'} \varphi \,\textbf{\textit{at}}\, p$. As the function PL can be effectively computed, we just need to consider the judgement $\Gamma; \Delta \vdash^{P'} \varphi \,\textbf{\textit{at}}\, p$ for the decidability result.

We can enumerate all proofs in the logic in which the set of places considered is finite. Hence, we obtain an effective enumeration of all provable judgements. We can also effectively enumerate all finite birelational models, and effectively check whether the judgement $\Gamma; \Delta \vdash^{P'} \varphi \,\textbf{\textit{at}}\, p$ is refutable in a given finite birelational model. As a consequence of the finite model property proved above, $\Gamma; \Delta \vdash^{P'} \varphi \,\textbf{\textit{at}}\, p$ is refutable only if it is refutable in some finite birelational model. By performing these enumerations and checks simultaneously, we obtain an effective test for provability of $\Gamma; \Delta \vdash^{P'} \varphi \,\textbf{\textit{at}}\, p$. □

The procedure detailed in the corollary above would not have worked if we had used Kripke models instead of birelational models. This is because the finite model property fails for Kripke models. For example, consider the judgement $; \Box\neg\neg A \ \mathbf{at}\ p \vdash^{\{p\}} \neg\neg\Box A \ \mathbf{at}\ p$. We claim that this judgement is valid for every *finite* Kripke model.

Indeed, let k be a Kripke state in some finite Kripke model $\mathcal{K}$ such that $(k, p) \models \Box\neg\neg A$. Pick $l \geq k$ in $\mathcal{K}$ such that l is maximal with respect to the ordering of Kripke states. As $(k, p) \models \Box\neg\neg A$, we get by definition that $(l, r) \models \neg\neg A$ for every place r in the state l. From the semantics of implication and the fact that l is a maximal state, it must be the case that $(l, r) \models A$ for every place r in the state l. Again, as l is maximal, we get $(l, p) \models \Box A$, and therefore $(l, p) \models \neg\neg\Box A$. As the model is finite, there is always a maximal l above any $k' \geq k$, and then $(l, p) \models \Box A$. We conclude $(k, p) \models \neg\neg\Box A$.

On the other hand, we showed that the judgement is not valid in the finite model $\mathcal{W}_{exam}$ in Ex. 4. The model $\mathcal{W}_{exam}$ has two worlds w_1 and w_2 such that $w_1 \leq w_2,\ w_1 R w_2,\ I(A) = \{w_2\},\ w_1\uparrow$ and $w_2\downarrow p$. As we discussed there, $w_2 \models \Box\neg\neg A$ and $w_2 \not\models \neg\neg\Box A$. As we mentioned before, this example is adapted from [114, 132].

2.8 Related Work

The logic we studied is an extension of the logic introduced in [91, 92]. In [91, 92], it was used as the foundation of a type system for a distributed λ-calculus by exploiting the *proofs-as-terms and propositions-as-types* paradigm. The proof terms corresponding to modalities have computational interpretation in terms of remote procedure calls ($@p$), commands to broadcast computations ($\Box$), and commands to use portable code ($\Diamond$). The authors also introduce a sequent calculus for the logic without disjunctive connectives, and prove that it enjoys cut elimination. Although the authors demonstrate the usefulness of logic in reasoning about the distribution of resources, they do not have a corresponding model.

The *proofs-as-terms and propositions-as-types* paradigm has also been used in [107, 108, 106]. In [107], the logic studied is an intuitionistic modal logic derived from *IS5*, and the modalities have a spatial flavour. Specifically, Kripke states are taken to be nodes on a network. The connective $\Box$ refelects the mobility of portable code, and $\Diamond$ reflects the address of a fixed resources. The work in [108] extends [107, 91, 92] to a lambda calculus for classical hybrid *S5* with network-wide continuations, which arise naturally from the underlying classical logic. These continuations create a new relationship between the two modalities and give a computational interpretation of theorems of classical hybrid *S5*. In [106], the relationship modal logics and type systems for Grid computing is investigated. The objects with type $\Box$ are interpreted as jobs that may be injected into the Grid and run anywhere. The main difference from [108, 107, 91, 92] is that the underlying logic is based on *S4* rather than *S5*. Whereas [108, 107, 91, 92] assume all nodes are connected to all other nodes, networks may have a more refined accessibility relation.

From a logical point of view, the logic we presented can be viewed as a hybrid modal logic [8, 9, 21, 22, 23, 120, 121]. A hybrid logic internalises the model in the logic

by using modalities built from pure names. The original idea of internalising the model into formulae was proposed in [120, 121], and has been further investigated in [8, 9, 21, 22, 23]. This work has been mostly carried out in the classical setting. More recently, classical hybrid logic is combined with Linear Temporal Logic in [115], and the logic accounts for both temporal and spatial aspects. Intuitionistic versions of hybrid logics were investigated in [32, 91, 92].

There are several intuitionistic modal logics in the literature, and [132] is a good source on them. The modalities in [132] have a temporal flavour, and the spatial interpretation was not recognised then. In [132], for example, the accessibility relation expresses the next step of a computation. The work in [32] extends the modal systems in [132], and creates hybrid versions of the modal systems by introducing *nominals*, a new kind of propositional symbols projecting semantics into the logic. A natural deduction system for these hybrid systems along with a normalisation result is also given in [32]. A Kripke semantics along with a proof of soundness and completeness is also introduced.

The extension we gave to the logic in [91, 92] is a hybrid version of the intuitionistic modal system *IS5* [113, 119, 132]. The modality $@p$ internalises the model in the logic. In the modal system *IS5*, first introduced in [119], the accessibility relation among places is total. The main difference in the logic presented in [32] and the logic in [91, 92] is that names in [91, 92] only occur in the modality $@p$.

From the point of view of semantics, Kripke semantics were first introduced in [94] for intuitionistic first-order logic. Kripke semantics for intuitionistic modal systems were developed in [63, 113, 117, 67, 132]. Birelational models for intuitionistic modal logic were introduced independently in [63, 67, 117]. They are in general useful to prove the finite model property as demonstrated in [114, 132]. The finite model property fails for Kripke semantics [132, 114], and an example for this was adapted here.

Some other examples of work on logics for resources are separation logics [125] and **BI**, the Logic of Bunched Implications [110, 122, 123]. Separation logic is an extension of Hoare logic that permits reasoning about low-level imperative programs with shared mutable data structure. Formulae are extended by introducing a 'separating conjunction' whose subformulae are meant to hold for disjoint parts of the system, thus enabling a concise and flexible description of structures with controlled sharing. **BI** is the theoretical base to separation logics. While Separation Logic is based on particular storage models, **BI** describe resources more generally and its model theory is inspired by a primitive of resource composition.

As explained in 1.8, the Logic of Bunched Implications is a substructural system which freely combines propositional Intuitionistic Logic and the Multiplicative fragment of propositional Linear Logic. In [110, 122, 123], the authors give a Kripke model based on monoids. The formulae of the logic are the resources, and are interpreted as elements of the monoid. The monoidal operation is reflected in the logic by the multiplicative connective. The focus of this work is the sharing of resources, and not their distribution.

BI-Loc, presented in [17], extends the Logic of Bunched Implication by introducing a modality for locations. Its models are *resource trees*: node-labelled trees in which nodes contain resources belonging to a monoid. Every label gives rise to a corresponding logical

modality which precisely indicates the location where a formula holds. Although **BI-Loc** offers a separation operator to express properties holding in different parts of the system, its propositional fragment cannot state properties verified in an unspecified node or in every node of the system. To fill this gap, authors introduce quantifications on locations and paths. Validity is undecidable for the full **BI-Loc** with quantifications, but it becomes decidable by avoiding the multiplicative (linear) implication.

The Logic of Bunched Implications has been recently extended in [124] with modalities, in a Hennessy-Milner style [81]. The new logic, **MBI**, is suitable to express properties of concurrent systems specified in a calculus of resources and processes. This gives a modal logic and a semantics that combines Kripke relational semantics with **BI** Kripke monoid semantics. A similar approach is presented in [33], where a Spatial Logic models the asynchronous π-calculus [104]. The logic is developed in classical settings and lacks a notion of resources. The main aim of Spatial Logic is to describe the behaviour and the spatial structure of concurrent systems. The logic is modal in space and in time, and a formula describes a property of a particular part of a concurrent system at a particular time.

Locations can be added to Spatial Logic along the lines of [43] which gives a modal logic based on Ambient Calculus [44]. Ambients are intended as locations, and there is a modality $m\ [\_]$ for every ambient name m which specifies the location where a property holds. These spatial modalities have an intensional flavour and 'hybridise' spatial logics as the modality $@p$ 'hybridises' *IS5* in the current thesis. However, the locations in Ambient Logic unlike this chapter have an intensional hierarchy which is reflected in the logic by having nested formulae like $m\ [n\ [\top]]$.

2.9 Conclusions

We studied the hybrid modal logic presented in [91, 92], and extended the logic with disjunctive connectives. Formulae in the logic contain names, also called places. The logic is useful to reason about placement of resources in a distributed system. We gave two sound and complete semantics for the logic.

In one semantics, we interpreted the judgements of the logic over Kripke-style models [94]. Typically, Kripke models [94] consist of partially ordered Kripke states. In our case, each Kripke state has a set of places, and different places satisfy different formulae. Larger Kripke states have larger sets of places, and the satisfaction of atoms corresponds to the placement of resources. The modalities of the logic allow formulae to be satisfied in a named place ($@p$), some place ($\Diamond$) and every place ($\Box$). The Kripke semantics can be seen as an instance of hybrid *IS5* [113, 119, 32, 132].

In the second semantics, we interpreted the judgements over birelational models [63, 67, 117, 132]. Typically, birelational models have a set of partially ordered worlds. In addition to the partial order, there is also a reachability relation amongst worlds. In order to interpret the modality $@p$ in the system, we also introduced a partial evaluation function on the set of worlds. The hybrid nature of the logic presented difficulties in the proof of

soundness. The difficulties are addressed using a mathematical construction that creates a new model from a given one. The set of worlds in the constructed model is the union of two sets. One of these sets is the reachability relation, and the worlds in the second set witness the existential and universal properties.

As in the case of intuitionistic modal systems [63, 67, 113, 117, 132], we demonstrated that the birelational models introduced here enjoy the finite model property: a judgement is not provable in the logic if and only if it is refutable in some finite model. The finite model property allowed us to conclude decidability. The partiality of the evaluation function was essential in the proof of the finite model property.

As future work, we are considering other extensions of the logic. A major limitation of the logic presented in [91, 92] is that if a formula φ is validated at some named place, say p, then the formula $\varphi@p$ can be inferred at every other place. Similarly, if $\Diamond\varphi$ or $\Box\varphi$ can be inferred at one place, then they can be inferred at any other place. In a large distributed system, we may want to restrict the rights of accessing information in a place. This can be done by adding an accessibility relation as is done in the case of other intuitionistic modal systems [132, 32]. We are currently investigating if the proof of the finite model property can be adapted to the hybrid versions of other intuitionistic modal systems. We are also investigating the computational interpretation of these extensions. This would result in extensions of λ-calculus presented in [91, 92]. We also plan to investigate adding temporal modalities to the logic. This will help us to reason about both space and time.

From a purely logical point of view, the meta-logic here to reason about soundness and completeness is classical. In order to obtain a full intuitionistic account for the logic, another line of investigation would be to consider categorical and/or topological semantics for the logic. This would allow us to obtain soundness and completeness results when the meta-logic is intuitionistic.

II

. . . and Back”

3

BiLog: a Contextual Spatial Logic Founded on Bigraphs

In this chapter we found the logic on a specific model: bigraphs. Bigraphs are emerging as an interesting (meta)model for concurrent calculi, like CCS, ambients, π-calculus, and Petri nets. They are built orthogonally on two structures: a hierarchical place graph for locations and a link (hyper)graph for connections. Aiming at describing bigraphical structures, we introduce a general framework, BiLog, whose semantics is given by arrows in monoidal categories. We then instantiate the framework to bigraphical structures and we obtain a logic that is a natural composition of a place graph logic and a link graph logic. We explore the concepts of separation and sharing in these logics and we prove that they generalise the well known spatial logics for trees, graphs and tree contexts. As an application, we show how XML data with links and web services can be modelled by bigraphs and described by BiLog. The framework can be extended by introducing dynamics in the model and a temporal modality in the logic in the usual way. However, in some interesting cases, temporal modalities can be already expressed in the static framework. To testify this, we show how to encode a minimal spatial logic for CCS in the instance of BiLog describing bigraphs.

3.1 Introduction

To describe and reason about structured, distributed, and dynamic resources is one of the main goals of global computing research. Recently, many *spatial logics* have been studied to fulfill this aim. The term 'spatial,' as opposed to 'temporal,' refers to the use of modal operators inspecting the structure of the terms in the model, rather than a temporal behaviour. Spatial logics are usually equipped with a separation/composition operator that *splits* a term into two parts, to 'talk' about them separately. The notion of *separation* is interpreted differently in different logics.

- In 'separation' logics [111], it is used to reason about dynamic update of heap-like structures, and it is *strong* as it forces names of resources in separated components to be disjoint. Consequently, term composition is usually partially defined.

- In static spatial logics, for instance for trees [36], graphs [39] and trees with hidden names [40], the separation/composition does not require any constraint on terms, and names are usually shared between separated parts.

- In dynamic spatial logics, too, the separation is intended only for locations in space (e.g. for ambients [42] or π-calculus [33]).

Context tree logic, introduced in [37], integrates the first approach above with a spatial logic for trees. The result is a logic able to express properties of tree-shaped structures (and contexts) with pointers, and it is used as an assertion language for Hoare-style program specifications in a tree memory model. Essentially, Spatial Logic founds its semantics on model structure.

Bigraphs [90, 99] are an emerging model for structures in global computing, that can be instantiated to model several well-known examples, including λ-calculus [102], CCS [103], π-calculus [90], ambients [88] and Petri nets [100]. Bigraphs consist essentially of two graphs sharing the same nodes. The first graph, the *place graph*, is tree structured and expresses a hierarchical relationship on nodes (viz. locality in space and nesting of locations). The second graph, the *link graph*, is an hyper-graph and expresses a generic *"many-to-many"* relationship among nodes (e.g. data link, sharing of a channel). The two structures are orthogonal, so links between nodes can cross locality boundaries. Thus, clarify the difference between structural separation (i.e., separation in the place graph) and name separation (i.e., separation on the link graph).

In this chapter we introduce a spatial logic for bigraphs as a natural composition of a place graph logic, for tree contexts, and a link graph logic, for name linkings. The main point is that a resource has a spatial structure as well as a link structure associated to it. Suppose for instance to be describing a tree-shaped distribution of resources in locations. We may use an atomic formula like $\mathsf{PC}(A)$ to describe a resource of 'type' PC (e.g. a personal computer) whose contents satisfy A, and a formula like $\mathsf{PC}_x(A)$ to describe the same resource at the location x. Note that the location type is orthogonal to the name. We can then write $\mathsf{PC}(\mathbf{T}) \otimes \mathsf{PC}(\mathbf{T})$ to characterise terms with two unnamed PC resources whose contents satisfy the tautological formula (i.e., with anything inside). Named locations, as e.g. in $\mathsf{PC}_a(\mathbf{T}) \otimes \mathsf{PC}_b(\mathbf{T})$, can express name separation, i.e., that names a and b are different (because separated by $\otimes$). Furthermore, link expressions can force name-sharing between resources with formulae like $\mathsf{PC}_a(\mathsf{in}_c \otimes \mathbf{T}) \overset{c}{\otimes} \mathsf{PC}_b(\mathsf{out}_c \otimes \mathbf{T})$. The formula describes two PC with different names, a and b, 'uniquely' sharing a link on a distinct name c, which models, e.g. a communication channel. Name c is used as input (in) for the first PC and as an output (out) for the second PC. No other name is shared and c cannot be used elsewhere inside PCs.

A bigraphical structure is, in general, a context with several holes and open links that can be filled by composition. The logic therefore describes contexts for resources at no additional cost. We can then express formulae like $\mathsf{PC}_a(\mathbf{T} \otimes \mathsf{HD}(id_1))$, that describes a modular computer PC, where id_1 represents a 'plug-able' hole in the hard disc HD. Contextual resources have many important applications. In particular, the contextual nature

of bigraphs is useful to characterise their dynamics, but it can also be used as a general mechanism to describe contexts of bigraphical data structures (cf. [54, 84]).

As bigraphs are establishing themselves as a truly general (meta)model of global systems, and appear to encompass several existing calculi and models (see for instance [90, 88, 100, 103]), our bigraph logic, *BiLog*, aims at achieving the same generality as a description language: as bigraphs specialise to particular models, we expect BiLog to specialise to powerful logics on these. In this sense, the contribution of this chapter is to propose BiLog as a unifying language for the description of global resources. We will explore this path in future work, fortified by the embedding results for the static spatial logics presented in §3.5, and the positive preliminary results obtained for semistructured data (cf.§3.6) and CCS (cf.§3.7).

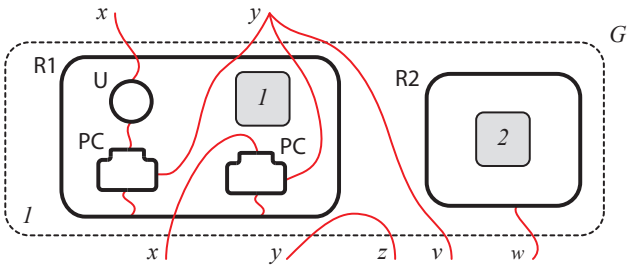
The chapter is organised as follows: 3.2 provides a crash course on bigraphs; §3.3 introduces the general framework and model theory of BiLog; §3.4 shows how to derive some useful connectives, such as a temporal modality and assertions constraining the "type" of terms; §3.5 instantiates the framework to obtain logics for place, link and bi-graphs; §3.6 focus on the applications of BiLog to XML data; §3.7 studies how to deal with dynamic models. An abridged version of this work appears in the conference paper [56] and the application to XML was presented in [54]. Here a new embedding result for a dynamic logic based on CCS [35] is added to our main technical result, that is the embedding of the static spatial logics of [36], [39] and [37] by BiLog. In particular, CCS embedding is based on an structural way of expressing the 'next-step' modality by composition adjuncts and bigraphical contexts. Moreover we show proofs, examples and properties with more details.

3.2 An Informal Introduction to Bigraphs

Bigraphs formalise distributed systems by focusing on two of their main characteristics: locality and interconnections. A bigraph consists of a set of *nodes*, which may be nested in a hierarchical tree structure, the so-called *place graph*, and have ports that may be connected to each other by *links*, the so-called *link graph*. Place graphs express locality, that is the physical arrangement of the nodes. Link graphs are hyper-graphs and formalise connections among nodes. The orthogonality of the two structures dictates that nestings impose no constrain upon interconnections.

The bigraph G of Fig. 3.1 represents a system where people and things interact. We imagine two offices with employees logged on PCs. Every entity is represented by a node, shown with bold outlines, and every node is associated with a *control* (either PC, U, R1, R2). Controls represent the kinds of nodes, and have fixed *arities* that determine their number of ports. Control PC marks nodes representing personal computers, and its arity is 3: in clockwise order, the ports represent a keyboard interacting with an employee U, a LAN connection interacting with another PC and open to the outside network, and the mains plug of the office R. The employee U may communicate with another one via the

Figure 3.1 A Bigraph $G : \langle 2, \{x, y, z, v, w\}\rangle \to \langle 1, \{x, y\}\rangle$.



upper port in the picture. The nesting of nodes (place graph) is shown by the inclusion of nodes into each other; the connections (link graph) are drawn as lines.

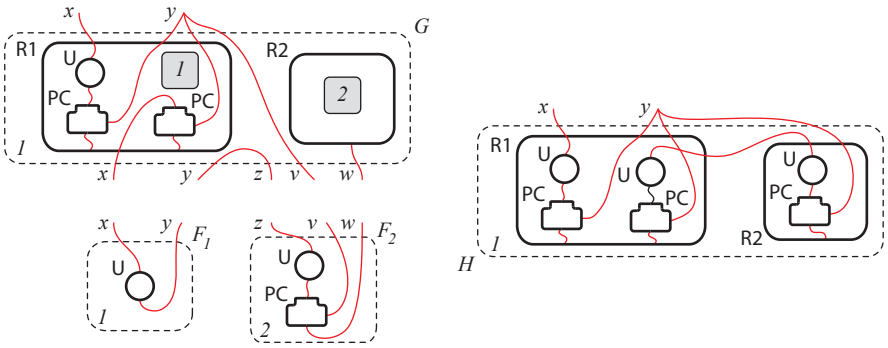
At the top level of the nesting structure sit the *regions*. In Fig. 3.1 there is one sole region (the dotted box). Inside nodes there may be 'context' *holes*, drawn as shaded boxes, which are uniquely identified by ordinals. The hole marked by 1 represents the possibility for another user U to get into office R1 and sit in front of a PC. The hole marked by 2 represents the possibility to plug a subsystem inside office R2.

Place graphs can be seen as *arrows* over a symmetric monoidal category whose objects are finite ordinals. We write $P : m \to n$ to indicate a place graph P with m holes and n regions. In Fig. 3.1, the place graph of G has type $2 \to 1$. Given the place graphs P_1, P_2, their composition $P_1 \circ P_2$ is defined only if the holes of P_1 are as many as the regions of P_2, and amounts to *filling* holes with regions, according to the number each carries. The tensor product $P_1 \otimes P_2$ is not commutative, as it lays the two place graphs one next to the other (in order), thus obtaining a graph with more regions and holes, and it 'renumbers' regions and holes 'from left to right'.

Link graphs are arrows of a partial monoidal category whose objects are (finite) sets of names. In particular, we assume a denumerable set Λ of names. A link graph is an arrow $X \to Y$, with X, Y finite subsets of Λ. The set X represents the *inner* names (drawn at the bottom of the bigraph) and Y represents the set of *outer* names (drawn on the top). The link graph connects ports to names or to *edges* (represented in Fig. 3.1 by a line between nodes), in any finite number. A link to a name is *open*, i.e., it may be connected to other nodes as an effect of composition. A link to an edge is *closed*, as it cannot be further connected to ports. Thus, edges are *private*, or hidden, connections. The composition of link graphs $\mathcal{W} \circ \mathcal{W}'$ corresponds to *linking* the inner names of $\mathcal{W}$ with the corresponding outer names of $\mathcal{W}'$ and forgetting about their identities. As a consequence, the outer names of $\mathcal{W}'$ (resp. inner names of $\mathcal{W}$) are not necessarily inner (resp. outer) names of $\mathcal{W} \circ \mathcal{W}'$. Thus link graphs can perform substitution and renaming, so the outer names in $\mathcal{W}'$ can disappear in the outer names of this means that either names may be renamed or edges may be added to the structure. As in [90], the tensor product of link graphs is defined in the obvious way only if their inner (resp. outer) names are disjoint.

By combining ordinals with names we obtain *interfaces*, i.e., couples $\langle m, X\rangle$ where

Figure 3.2 Bigraphical Composition, $H \equiv G \circ (F_1 \otimes F_2)$.



m is an ordinal and X is a finite set of names. By combining the notion of place graph and link graphs on the same set of nodes we obtain the notion of bigraphs. In particular a bigraph is an arrow $G : \langle m, X\rangle \to \langle n, Y\rangle$, and we say that $\langle m, X\rangle$ and $\langle n, Y\rangle$ are its *innerface* and *outerface*, respectively.

Figure 3.2 represents a more complex situation. Its top left-hand side reports the system of Fig. 3.1, in its bottom left-hand side F_1 represents a user U ready to interact with a PC or with some other users, F_2 represents a user logged on its laptop, ready to communicate with other users. The system with F_1 and F_2 represents the tensor product $F = F_1 \otimes F_2$. The right-hand side of Fig. 3.2 represents the composition $G \circ F$. The idea is to insert F into the context G. The operation is partially defined, since it requires the inner names and the number of holes of G to match the outer names and the number of regions of F, respectively. Shared names create the new links between the two structures. Intuitively, composition *first* places every region of F in the proper hole of G (place composition) and *then* joins equal inner names of G and outer names of F (link composition). In the example, as a consequence of the composition the user U in the first region of F is logged on PC, the user U in the second region of F is in room R2. Moreover note the edge connecting the inner names y and z in G, its presence produces a link between the two users of F after the composition, imagine a phone call between the two users.

3.3 BiLog: Syntax and Semantics

The final aim of the chapter is to define a logic able to describe bigraphs and their substructures. Since bigraphs, place graphs, and link graphs are arrows of a (partial) monoidal category, we first introduce a meta-logical framework having monoidal categories as models; then we adapt it to model the orthogonal structures of place and link graphs. Finally, we specialise the logic to model the whole structure of (abstract) bigraphs.

Following the approach of spatial logics, we introduce connectives that reflect the structure of the model. In this case, models are monoidal categories and the logic describes spatially the structure of their *arrows*.

Table 3.1 Typing Rules

$$\frac{type(\Omega) = I \to J}{\Omega : I \to J} \qquad \frac{G : I' \to J \quad F : I \to I'}{G \circ F : I \to J}$$

$$\frac{G : I_1 \to J_1 \quad F : I_2 \to J_2 \quad I = I_1 \otimes I_2 \quad J = J_1 \otimes J_2}{G \otimes F : I \to J}$$

The meta-logical framework we propose is inspired by the bigraph axiomatisation presented in [101]. The model of the logic is composed by *terms* of a general language with *horizontal* and *vertical* compositions and a set of unary constructors. Terms are related by a *structural congruence* that satisfies the axioms of monoidal categories, and possibly more. The corresponding model theory is parameterised on basic constructors and structural congruence. To be as free as possible in choosing the level of intensionality, the logic is defined on a *transparency* predicate. Its role is to identify the terms allowing inspection of their content, *transparent* terms, and the ones that do not, *opaque* terms. We inspect the logical equivalence induced by the logic and we observe that it corresponds to the structural congruence when every term is transparent, and it becomes less discriminating with the introduction of *opaque terms*, cf. §3.3.2.

3.3.1 Terms

To evaluate formulae, we consider the terms freely generated from a set of constructors Θ, ranged over by Ω, by using the (partial) operators: composition ($\circ$) and tensor ($\otimes$). The order of binding precedence is $\circ$, $\otimes$. BiLog terms are defined by the following grammar:

G, G'	::=	Ω	constructor (for $\Omega \in \Theta$)
		$G \circ G'$	vertical composition
		$G \otimes G'$	horizontal composition

When defined, the two operations must satisfy the *bifunctoriality property* of monoidal categories, thus we refer to these terms also as *bifunctorial terms*.

Terms represent structures built on a (partial) monoid $(\mathcal{M}, \otimes, \epsilon)$ whose elements are dubbed *interfaces* and denoted by I, J. To model nominal resources, such as heaps or link graphs, we allow the monoid to be partial.

Intuitively, terms represent typed structures with a source and a target interface ($G : I \to J$). Structures can be placed one near to the other (horizontal composition) or one inside the other (vertical composition). Each Ω in Θ has a fixed type $type(\Omega) = I \to J$. For each interface I, we assume a distinguished construct $id_I : I \to I$. The types of constructors, together with the rules in Tab. 3.1, determine the type of each term. Terms of type $\epsilon \to J$ are called *ground*.

The term obtained by tensor is well typed when both corresponding tensors on source and target interface are defined, namely they are separated structures. On the other hand,

Table 3.2 Axioms

Congruence Axioms:	
$G \equiv G$	Reflexivity
$G \equiv G'$ implies $G' \equiv G$	Symmetry
$G \equiv G'$ and $G' \equiv G''$ implies $G \equiv G''$	Transitivity
$G \equiv G'$ and $F \equiv F'$ implies $G \circ F \equiv G' \circ F'$	Congruence $\circ$
$G \equiv G'$ and $F \equiv F'$ implies $G \otimes F \equiv G' \otimes F'$	Congruence $\otimes$
Monoidal Category Axioms:	
$G \circ id_I \equiv G \equiv id_J \circ G$	Identity
$(G_1 \circ G_2) \circ G_3 \equiv G_1 \circ (G_2 \circ G_3)$	Associativity
$G \otimes id_\epsilon \equiv G \equiv id_\epsilon \otimes G$	Monoid Identity
$(G_1 \otimes G_2) \otimes G_3 \equiv G_1 \otimes (G_2 \otimes G_3)$	Monoid Associativity
$id_I \otimes id_J \equiv id_{I \otimes J}$	Interface Identity
$(G_1 \otimes F_1) \circ (G_2 \otimes F_2) \equiv (G_1 \circ G_2) \otimes (F_1 \circ F_2)$	Bifunctoriality

composition is defined only when the two involved terms *share* a common interface. In the following, we consider only well typed terms.

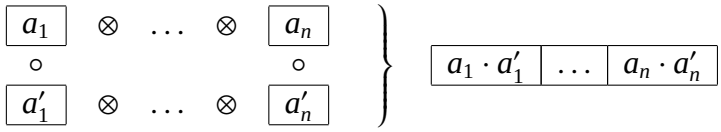
Terms are defined up to the structural congruence $\equiv$ described in Tab. 3.2. It subsumes the axioms of the monoidal categories. All axioms are required to hold whenever both sides are well typed. Throughout the chapter, when using $=$ or $\equiv$ we imply that both sides are defined; and when we need to remark that a bigraphical expression E is well given, we write $(E)\downarrow$. Later on, the congruence will be refined to model specialised structures, such as place graphs, link graphs or bigraphs.

The axioms correspond to those for (partial) monoidal categories. In particular we constrain the structural congruence to satisfy the bifunctoriality property between product and composition. Thus, we can interpret our terms as arrows of the free monoidal category on $(\mathcal{M}, \otimes, \epsilon)$ generated by Θ. In this case the term congruence corresponds to the equality of the corresponding arrows.

Example 6. An intuitive example of bifunctorial terms is provided by located resources. Every location is represented by a *cell*; every cell can contain a resource. Horizontal composition represents the merging of cells, and vertical composition combines the resources included in the cells. This model will provide a semantics to the logical operators we are defining, and will show that BiLog, although inspired by bigraphs, is not only connected to the bigraphical framework (cf. Ex. 7).

The set of resources is a monoidal structure $(M, \lambda, \cdot)$ freely generated by a set Λ of resource generators. The resource monoid may possibly be partial. In this case, the monoid of interfaces is the commutative monoid of ordinals $(\mathbf{N}, 0, +)$, freely generated by $\{1\}$. We define the constructor $\boxed{\lambda} : 1 \to 1$ for the neutral element λ and a constructor $\boxed{a} : 1 \to 1$ for each element $a \in \Lambda$. Every element $\boxed{}$ represents a cell, the constructor $\boxed{a}$ represents a cell containing the resource generator a. Table 3.3 outlines the two composition operators. The vertical composition $\circ$ between two cells $\boxed{a_1}$ and $\boxed{a_2}$ corresponds to

Figure 3.3 Cell Compositions



combine – when possible – the two generators contained in the cells, thus producing the cell $\boxed{a_1 \cdot a_2}$ containing the resource $a_1 \cdot a_2$. This operation produces a cell $\boxed{m}$ for every resource $m \in M$. The horizontal composition $\otimes$ consists of aligning two cells, thus producing lists of cells.

The terms generated by these settings are *resources vectors* Their inner and outer faces correspond to their size. The horizontal composition $\otimes$ is in general the juxtaposition of vectors. Given the vectors $\boxed{m_1 \mid ... \mid m_n} : n \to n$, of size n, and $\boxed{m'_1 \mid ... \mid m'_{n'}} : n' \to n'$, of size n', the composition $\otimes$ is formally defined as

$$\boxed{m_1 \mid ... \mid m_n} \otimes \boxed{m'_1 \mid ... \mid m'_{n'}} \stackrel{\text{def}}{=} \boxed{m_1 \mid ... \mid m_n \mid m'_1 \mid ... \mid m'_{n'}}.$$

The resulting vector is typed by $(n + n') \to (n + n')$, and has size $n + n'$.

The vertical composition $\circ$ is defined only between vectors with equal size, and corresponds to combine the resources cell by cell, as follows:

$$\boxed{m_1 \mid ... \mid m_n} \circ \boxed{m'_1 \mid ... \mid m'_n} \stackrel{\text{def}}{=} \boxed{m_1 \cdot m'_1 \mid ... \mid m_n \cdot m'_n}.$$

The two operations satisfy the bifunctorial property, which represents here the possibility to chose either to concatenate the vectors first and then to combine the resources, or vice versa. For cells, the bifunctorial property says

$$\left(\boxed{m_1} \otimes \boxed{m_2}\right) \circ \left(\boxed{m_3} \otimes \boxed{m_4}\right) = \left(\boxed{m_1} \circ \boxed{m_3}\right) \otimes \left(\boxed{m_2} \circ \boxed{m_4}\right).$$

The two terms above correspond to $\boxed{m_1 \cdot m_3 \mid m_2 \cdot m_4}$. The bifunctorial provides two possible normal forms for terms: *(i)* the *horizontal outermost* $\left(\boxed{a_1} \circ \ldots \circ \boxed{a_n}\right) \otimes \ldots \otimes \left(\boxed{a_1^m} \circ \ldots \circ \boxed{a_{n^m}^m}\right)$, with $a_i^j \in \Lambda$, that first combines by $\circ$ and then by $\otimes$; and *(ii)* the *vertical outermost* $\left(\boxed{a_1} \otimes \ldots \otimes \boxed{a_n}\right) \circ \ldots \circ \left(\boxed{a_1^m} \otimes \ldots \otimes \boxed{a_{n^m}^m}\right)$, where $a_i^j \in \Lambda \cup \{\lambda\}$ and $a_i^j = \lambda$ implies $a_i^{j+1} = \lambda$, that first combines by $\otimes$ and then by $\circ$. The congruence on resource vectors is represented by the equality on normal forms, and it satisfies all the axioms of Tab. 3.2. In Particular, id_0 represents the empty resource vector, id_1 corresponds to $\boxed{\lambda}$, and in general id_n is $\boxed{\lambda \mid \ldots \mid \lambda} : n \to n$.

The properties of these particular terms depend strictly on the choice of the underlying resource monoid, which can be either non-commutative (whenever considering sequences of resources, or ordered trees), or commutative (whenever considering multisets of resources, or unordered trees), or partial (whenever dealing with heaps). This example is rather limited, in the sense that inner and outer faces are forced to be equals, an there

are only two kinds of constructors. The full generality will be reached with bigraphs. The aim of this model is to hint that BiLog can characterise models not directly based on bigraphs, as Ex. 7 will show.

3.3.2 Transparency

In general not every structure of the model corresponds to an observable structure in a spatial logic. A classical example is ambient logic. Some mobile ambient constructors have their logical equivalent, e.g. ambients $\mathbf{a}[-]$, and other ones are not directly mapped in the logic, e.g. the **in** and **out** capabilities. In this case the observability of the structure is distinguished from the observability of the computational terms: some terms are used to express behaviour and other to express structure. Moreover there are terms representing both notions since ambients can be opened.

The structure may be used not only to represent the distribution or the shape of resources but also to encode their behaviour. We may want to avoid a direct representation of some structures at logical level of BiLog. A natural solution is to define a notion of *transparency* over the structure. In such a way, entities representing the structure are *transparent*, while entities encoding behaviour are *opaque* and cannot be distinguished by the logical spatial connectives. Transparent terms allow the logic to see their entire structure while opaque terms block the inspection at some opacity point. A notion of transparency can also appear in models without temporal behaviour. In fact, consider a model with a variable access control policy determined by some structural characteristics. Thus, some terms may be either transparent or opaque, depending on the current policy, and the visibility in the logic, or in the query language, will be influenced by this.

When the model is dynamic, the reacting contexts, namely those with a possible temporal evolution, are specified with an activeness predicate. We may be tempted to identify transparency and activeness. Although these concepts collapse in some case, they are orthogonal in general. There may be transparent terms that are active, such as a public 'browse-able' directory; opaque terms that are active, such as an agent that hides its contents; passive transparent terms, such as a portable code; and passive opaque terms, such as controls encoding synchronisation.

More generally the transparency predicate prevents logical identification of terms. As an example, consider an XML document. We may want to restrict our attention to a particular set of nodes; we could, e.g., ignore data values when interested in the structure. In other situations, we may want a different logic focused on values, but not on node attributes.

Transparency is essentially a way to restrict the observational power of the structural logic. Notice that in general such a restriction of the observational power in the static logic does not imply a restriction of observational power in the dynamic counterpart. In fact, a next step modality may induce a 'new' intensionalisation of the controls by observing how the model evolves, as shown in [35] and [131].

Table 3.3 BiLog$(M, \otimes, \epsilon, \Theta, \equiv, \tau)$

$\mathbf{\Omega}$ ::=	$\mathbf{id}_I \mid \ldots$	a constant formula for every Ω s.t. $\tau(\Omega)$		
A, B ::=	$\mathbf{F}$	false	$A \Rightarrow B$	implication
	$\mathbf{id}$	identity	$\mathbf{\Omega}$	constant constructor
	$A \otimes B$	tensor product	$A \circ B$	composition
	$A \multimap B$	left comp. adjunct	$A \mathrel{-\!\!\triangleright} B$	right comp. adjunct
	$A \circ\!\!- B$	left prod. adjunct	$A \multimap B$	right prod. adjunct

$G \models \mathbf{F}$	iff	never
$G \models A \Rightarrow B$	iff	$G \models A$ implies $G \models B$
$G \models \mathbf{\Omega}$	iff	$G \equiv \Omega$
$G \models \mathbf{id}$	iff	exists I s.t. $G \equiv id_I$
$G \models A \otimes B$	iff	exists G_1, G_2 s.t. $G \equiv G_1 \otimes G_2$, with $G_1 \models A$ and $G_2 \models B$
$G \models A \circ B$	iff	exists G_1, G_2. s.t. $G \equiv G_1 \circ G_2$, with $\tau(G_1)$ and $G_1 \models A$ and $G_2 \models B$
$G \models A \multimap B$	iff	for all G', the fact that $G' \models A$ and $\tau(G')$ and $(G' \circ G)\downarrow$ implies $G' \circ G \models B$
$G \models A \mathrel{-\!\!\triangleright} B$	iff	$\tau(G)$ implies that for all G', if $G' \models A$ and $(G \circ G')\downarrow$ then $G \circ G' \models B$
$G \models A \circ\!\!- B$	iff	for all G', the fact that $G' \models A$ and $(G' \otimes G)\downarrow$ implies $G' \otimes G \models B$
$G \models A \multimap B$	iff	for all G', the fact that $G' \models A$ and $(G \otimes G')\downarrow$ implies $G \otimes G' \models B$

3.3.3 Formulae

BiLog internalises the constructors of bifunctorial terms in the style of the ambient logic [42]. Constructors appear in the logic as constant formulae, while tensor product and composition are expressed by connectives. Thus the logic presents two binary spatial operators. This contrasts with other spatial logics, with a single one: Spatial and Ambient Logics [33, 42], with parallel composition $A \mid B$, Separation Logic [111], with separating conjunction $A * B$, and Context Tree Logic [37], with application $K(P)$. Both the operators inherit the monoidal structure and non-commutativity properties from the model.

The logic is parameterised by the transparency predicate $\tau(\,)$: as explained in the previous section, opaque terms do not allow inspection of their contents. We say that a term G is transparent, or observable, if $\tau(G)$ is verified. We will see that when all terms are observable the logical equivalence corresponds to $\equiv$. We assume that id_I and ground terms are always transparent, and τ preserves the congruence $\equiv$, and the compositions $\otimes$ and $\circ$.

Given the monoid $(\mathcal{M}, \otimes, \epsilon)$, the set of simple terms Θ, the transparency predicate τ and the structural congruence relation $\equiv$, the logic BiLog$(\mathcal{M}, \otimes, \epsilon, \Theta, \equiv, \tau)$ is formally

defined in Tab. 3.3. The satisfaction relation $\models$ gives the semantics. The logic features a constant $\mathbf{\Omega}$ for each transparent construct Ω. In particular it has the identity $\mathbf{id}_I$ for each interface I. The satisfaction of logical constants is simply the congruence to the corresponding constructor. The *horizontal decomposition* formula $A \otimes B$ is satisfied by a term that can be decomposed as the tensor product of two terms satisfying A and B respectively. The degree of separation enforced by $\otimes$ between terms plays a fundamental role in the various instances of the logic, notably link graph and place graph. The *vertical decomposition* formula $A \circ B$ is satisfied by terms that can be the composition of terms satisfying A and B. We shall see that in some cases both connectives correspond to well known spatial ones. We define the *left* and *right adjuncts* for composition and tensor to express extensional properties. The left adjunct $A \leftarrow B$ expresses the property of a term to satisfy B whenever inserted in a context satisfying A. Similarly, the right adjunct $A \multimap B$ expresses the property of a context to satisfy B whenever filled with a term satisfying A. A similar description holds for $\circ\!\!-$ and $-\!\!\circ$, the adjoints of $\otimes$. Clearly these adjoints collapse whenever the tensor is commutative in the model.

Example 7. Consider the resource vectors defined in Ex. 6. When a BiLog formula is interpreted in that context, it represents a class of resource vectors. For sake of simplicity, we assume that all this terms are transparent. Thus, when instantiated on these terms, BiLog provides a formula $\boxed{\boldsymbol{a}}$ for each constructor $\boxed{a}$. The semantics of $\boxed{\boldsymbol{a}}$ represents the class of all the terms whose normal form is the constructor $\boxed{a}$. For instance, $\boxed{a \cdot \lambda} \otimes id_0 \models \boxed{\boldsymbol{a}}$. The formula $A \otimes B$ means that a resource vector can be horizontally divided into two resource vectors satisfying A and B respectively. For instance the formula $\boxed{\boldsymbol{a}} \otimes \mathbf{T}$ is satisfied by all the resource vectors having $\boxed{a}$ as first cell. On the other hand, the formula $\boxed{\boldsymbol{a}} \circ \mathbf{T}$ implicitly says that a resource vector is composed by a single cell containing a resource whose generators include a. In addition, if the resource monoid is not commutative, the previous formula says that the first element in the composition is actually a. The formula $\mathbf{T} \otimes A \otimes \mathbf{T}$ characterises resources vectors with a subvector satisfying A. In particular $\mathbf{T} \otimes (A \circ \mathbf{id}_1) \otimes \mathbf{T}$ means that one of the cells in the vector satisfied A. Finally, if we use $\mathbf{T} \otimes (\mathbf{T} \circ \boxed{\boldsymbol{a}} \circ \mathbf{T}) \otimes \mathbf{T}$ says that the resource a appears somewhere in the resource vector. More generally the formula $\mathbf{id}_1 \circ \mathbf{T}$ means that the resource vector has size 1, then it is a simple sequence.

The formula $\boldsymbol{Cell} \stackrel{\text{def}}{=} \mathbf{id}_1 \circ (\neg\mathbf{id}_1 \wedge (\neg(\neg\mathbf{id}_1 \circ \neg\mathbf{id}_1))$ states that a resource vector is not empty and it is not composed by two not empty vectors, then it is a single cell. The *Cell* formula is useful to define two operators that correspond to the Kleene stars for the bigraphical combinators. Let $\boxed{\boldsymbol{a}}^{\otimes *} \stackrel{\text{def}}{=} \neg\left(\mathbf{T} \otimes \left(\boldsymbol{Cell} \wedge \neg\boxed{\boldsymbol{a}}\right) \otimes \mathbf{T}\right)$. This formula is satisfied by resource vectors that are not composed by cells different from $\boxed{a}$. Thus $\boxed{\boldsymbol{a}}^{\otimes *}$ characterises resource vectors of the kind $\boxed{a} \otimes \ldots \otimes \boxed{a}$, namely elements of the Kleene star generated by $\boxed{a}$ and the composition $\otimes$. This idea can be extended to a formula A:

$$A^{\otimes *} \stackrel{\text{def}}{=} \neg\left(\mathbf{T} \otimes (\boldsymbol{Cell} \wedge \neg A) \otimes \mathbf{T}\right);$$
$$A^{\circ *} \stackrel{\text{def}}{=} \neg\left(\mathbf{T} \otimes (\boldsymbol{Cell} \wedge \neg A) \otimes \mathbf{T}\right).$$

A vector of resources satisfies $A^{\otimes *}$ if it is composed only by cells satisfying A.

3.3.4 Properties

Here we show some basic results about BiLog. In particular, we observe that, in presence of trivial transparency, the induced logical equivalence coincides with the structural congruence of the terms. Such a property is fundamental to describe, query and reason about bigraphical data structures, as e.g. XML (cf. §3.6). In other terms, BiLog is *intensional* in the sense of [131], namely it can observe internal structures, as opposed to the extensional logics used to observe the behaviour of dynamic system. Inspired by [85], it would be possible to study a fragment of BiLog without the intensional operators $\otimes$, $\circ$, and constants.

The lemma below states that the relation $\models$ respects the congruence.

Lemma 16 (Congruence Preservation). *For every couple of terms G and G', if $G \models A$ and $G \equiv G'$ then $G' \models A$.*

Proof. Induction on the structure of the formula, by recalling that the congruence is required to preserve the typing and the transparency. In detail

Case F. Nothing to prove.

Case $\mathbf{\Omega}$. By hypothesis $G \models \mathbf{\Omega}$ and $G \equiv G'$. By definition $G \equiv \Omega$ and by transitivity $G' \equiv \Omega$, thus $G' \models \mathbf{\Omega}$.

Case id. By hypothesis $G \models \mathbf{id}$ and $G \equiv G'$. Hence there exists an I such that $G' \equiv G \equiv id_I$ and so $G' \models \mathbf{id}$.

Case $A \Rightarrow B$. By hypothesis $G \models A \Rightarrow B$ and $G \equiv G'$. This means that if $G \models A$ then $G \models B$. By induction if $G' \models A$ then $G \models A$. Thus if $G' \models A$ then $G \models B$ and again by induction $G' \models B$.

Case $A \otimes B$. By hypothesis $G \models A \otimes B$ and $G \equiv G'$. Thus there exist G_1, G_2 such that $G' \equiv G \equiv G_1 \otimes G_2$ and $G_1 \models A$ and $G_2 \models B$. Hence $G' \models A \otimes B$.

Case $A \circ B$. By hypothesis $G \models A \circ B$ and $G \equiv G'$. Thus there exist G_1, G_2 such that $G' \equiv G \equiv G_1 \circ G_2$ and $\tau(G_1)$ and $G_1 \models A$ and $G_2 \models B$. Hence $G' \models A \circ B$.

Case $A \mathrel{-\!\!\circ} B$. By hypothesis $G \models A \mathrel{-\!\!\circ} B$ and $G \equiv G'$. Thus for every G'' such that $G'' \models A$ and $\tau(G'')$ and $(G'' \circ G)\downarrow$ it holds $G'' \circ G \models B$. Now $G \equiv G'$ implies $G'' \circ G \equiv G'' \circ G'$; moreover the congruence preserves typing, so $(G'' \circ G')\downarrow$. By induction $G'' \circ G' \models B$, then conclude $G' \models A \mathrel{-\!\!\circ} B$.

Case $A \mathrel{-\!\!\circ\!\!\!\rightarrow} B$. If $\tau(G')$ is not verified, then $G' \models A \mathrel{-\!\!\circ\!\!\!\rightarrow} B$ trivially holds. Suppose $\tau(G')$ to be verified. As $G \equiv G'$ and transparency preserves congruence, $\tau(G)$ is verified as well. By hypothesis for each G'' satisfying A such that $(G \circ G'')\downarrow$ it holds

$G \circ G'' \models B$, and by induction $G' \circ G'' \models B$, as $G \equiv G'$ and $(G \circ G'')\downarrow$ implies $(G' \circ G'')\downarrow$ and $G \circ G'' \equiv G' \circ G''$. This proves $G' \models A \multimap B$.

Case $A \circ\!\!- B$ (and symmetrically $A \multimap B$)**.** By hypothesis $G \models A \circ\!\!- B$ and $G \equiv G'$. Thus for each G'' such that $G'' \models A$ and $(G'' \otimes G)\downarrow$ then $G'' \otimes G \models B$. Now $G \equiv G'$ implies $G'' \otimes G \equiv G'' \otimes G'$, again the congruence must preserve typing so $(G'' \otimes G')\downarrow$. Thus by induction $G'' \otimes G' \models B$. The generality of G'' implies $G' \models A \circ\!\!- B$. □

BiLog induces a logical equivalence $=_L$ on terms in the usual sense. We say that $G_1 =_L G_2$ if for every formula A, $G_1 \models A$ implies $G_2 \models A$ and vice versa. It is easy to prove that the logical equivalence corresponds to the congruence in the model if the transparency predicate is true for every term.

Theorem 23 (Logical Equivalence and Congruence). *When the transparency predicate is always true, then $G =_L G'$ if and only if $G \equiv G'$ for every term G, G'.*

Proof. The forward direction is proved by defining the characteristic formula for terms, as every term can be expressed as a formula. In fact, the transparency predicate is total, hence every constant term corresponds to a constant formula. The converse is a direct consequence of Lemma 16. □

The logical equivalence is less discriminating in presence of opaque constructors. For instance, the logic cannot distinguish two opaque constructors of equal type.

The particular characterisation of the logical equivalence as the congruence in the case of trivial transparency can be generalised to a congruence 'up-to-transparency.' That means we can find an equivalence relation between trees that is 'tuned' by τ: the more τ covers, the less the equivalence distinguishes. This relation will be better understood when we instantiate the logic to particular terms. A possible definition of transparency will be provided in 3.5.6.

3.4 BiLog: Derived Operators

Table 3.4 outlines several operators that can be derived in BiLog. The classical operators and those constraining the interfaces are self-explanatory. The 'dual' operators are worth explaining. The formula $A \ominus B$ is satisfied by terms G such that for every possible decomposition $G \equiv G_1 \otimes G_2$ either $G_1 \models A$ or $G_2 \models B$. For instance, $A \ominus A$ describes terms where A is true in, at least, one part of each $\otimes$-decomposition. The formula $\mathbf{F} \ominus (\mathbf{T}_{\rightarrow I} \Rightarrow A) \ominus \mathbf{F}$ describes those terms where every component with outerface I satisfies A. Similarly, the composition $A \bullet B$ expresses structural properties universally quantified on every $\circ$-decomposition. Both these connectives are useful to specify security properties or types.

The adjunct dual $A \leftarrow B$ describes terms that can be inserted into a particular context satisfying A to obtain a term satisfying B, it is a sort of existential quantification on

Table 3.4 Derived Operators

$\mathbf{T}$, $\wedge$, $\vee$, $\Leftrightarrow$, $\Leftarrow$, $\neg$			Classical operators
A_I	$\stackrel{\text{def}}{=}$	$A \circ \mathbf{id}_I$	Constraining the source to be I
$A_{\to J}$	$\stackrel{\text{def}}{=}$	$\mathbf{id}_J \circ A$	Constraining the target to be J
$A_{I \to J}$	$\stackrel{\text{def}}{=}$	$(A_I)_{\to J}$	Constraining the type to be $I \to J$
$A \circ_I B$	$\stackrel{\text{def}}{=}$	$A \circ \mathbf{id}_I \circ B$	Composition with interface I
$A \leftarrow_J B$	$\stackrel{\text{def}}{=}$	$A_{\to J} \leftarrow B$	Contexts with J as target guarantee
$A \rightarrowtriangle_I B$	$\stackrel{\text{def}}{=}$	$A_I \rightarrowtriangle B$	Composing with terms having I as source
$A \ominus B$	$\stackrel{\text{def}}{=}$	$\neg(\neg A \otimes \neg B)$	Dual of tensor product
$A \bullet B$	$\stackrel{\text{def}}{=}$	$\neg(\neg A \circ \neg B)$	Dual of composition
$A \blacktriangleleft B$	$\stackrel{\text{def}}{=}$	$\neg(A \leftarrow \neg B)$	Dual of composition left adjunct
$A \blacktriangleright B$	$\stackrel{\text{def}}{=}$	$\neg(A \rightarrowtriangle \neg B)$	Dual of composition right adjunct
$A^{\exists\otimes}$	$\stackrel{\text{def}}{=}$	$\mathbf{T} \otimes A \otimes \mathbf{T}$	Some horizontal term satisfies A
$A^{\forall\otimes}$	$\stackrel{\text{def}}{=}$	$\mathbf{F} \ominus A \ominus \mathbf{F}$	Every horizontal term satisfies A
$A^{\exists\circ}$	$\stackrel{\text{def}}{=}$	$\mathbf{T} \circ A \circ \mathbf{T}$	Some vertical term satisfies A
$A^{\forall\circ}$	$\stackrel{\text{def}}{=}$	$\mathbf{F} \bullet A \bullet \mathbf{F}$	Every vertical term satisfies A
$\diamondsuit A$	$\stackrel{\text{def}}{=}$	$(\mathbf{T} \circ A)_\epsilon$	Somewhere modality (on ground terms)
$\boxdot A$	$\stackrel{\text{def}}{=}$	$\neg\diamondsuit\neg A$	Anywhere modality (on ground terms)

contexts. For instance $(\mathbf{\Omega}_1 \vee \mathbf{\Omega}_2) \blacktriangleleft A$ describes the union between the class of two-region bigraphs (with no names in the outerface) whose merging satisfies A, and terms that can be inserted either in Ω_1 or Ω_2 resulting in a term satisfying A. Similarly the dual adjunct $A \blacktriangleright B$ describes contextual terms G such that there exists a term satisfying A that inserted in G gives a term satisfying B.

The formulae $A^{\exists\otimes}$, $A^{\forall\otimes}$, $A^{\exists\circ}$, and $A^{\forall\circ}$ correspond to quantifications on the horizontal/vertical structure of terms. For instance $\mathbf{\Omega}^{\forall\circ}$ describes terms that are a finite (possibly empty) composition of simple terms Ω. Next section discusses spatial modalities $\diamondsuit$ and $\boxdot$.

Following lemma states a first property involving the derived connectives, by proving that the interfaces for transparent terms can be observed.

Lemma 17 (Type Observation). *For every term G, it holds: $G \models A_{I \to J}$ if and only if $G : I \to J$ and $G \models A$ and $\tau(G)$.*

Proof. For the forward direction, assume that $G \models A_{I \to J}$, then $G \equiv id_J \circ G' \circ id_I$ with $G' \models A$ and $\tau(G')$. Now, $id_J \circ G' \circ id_I : I \to J$. By Lemma 16: $G : I \to J$ and $G \models A$ and $\tau(G)$. The converse is a direct consequence of the semantics definition. □

Thanks to the derived operators involving interfaces, the equality between interfaces,

$I = J$, is derivable by $\otimes$ and $\circ\!\!-$, as

$$\mathbf{T} \otimes (id_\epsilon \wedge (id_I \circ\!\!- id_J)). \tag{3.1}$$

Whenever a bigraph satisfies such a formula, the interfaces I and J are equal. To gather the basic idea, assume the bigraph G satisfies (3.1). This means that $G \equiv G_1 \otimes G_2$ with $G_1 \models \mathbf{T}$ and $G_2 \models id_\epsilon \wedge (id_I \circ\!\!- id_J)$. By definition, the latter is equivalent to $G_2 \equiv \epsilon$ and $G_2 \models id_I \circ\!\!- id_J$. Then $G \equiv G_1$ and $\epsilon \models id_I \circ\!\!- id_J$, by Lemma 16. Hence $\epsilon \otimes id_I \models id_J$, that entails $id_I \equiv id_J$. Clearly, the last equality holds only if $I = J$. By reversing the reasoning, it is easy to see that whenever $I = J$, every bigraph satisfies (3.1).

3.4.1 Somewhere Modality

The idea of *sublocation*, $\sqsubseteq$ defined in [43], can be extended to the bigraphical terms. A sublocation corresponds to a subterm and it is formally defined on ground terms as follows. The definition of sublocation makes sense only for ground terms, as the structure of 'open' terms (i.e., with holes) is not known a priori. Formally it is defined as follows.

Definition 21 (Sublocation). *Given two terms $G : \epsilon \to J$ and $G' : \epsilon \to J'$, term G' is defined to be a sublocation for G, and write $G' \sqsubseteq G$, inductively by:*

- *$G' \sqsubseteq G$, if $G' \equiv G$;*
- *$G' \sqsubseteq G$, if $G \equiv G_1 \otimes G_2$, with $G' \sqsubseteq G_1$ or $G' \sqsubseteq G_2$;*
- *$G' \sqsubseteq G$, if $G \equiv G_1 \circ G_2$, with $\tau(G_1)$ and $G' \sqsubseteq G_2$.*

This relation, introduce a *"somewhere"* modality in the logic. Intuitively, a term satisfies *"somewhere"A* whenever one of its sublocations satisfies *A*. Rephrasing the semantics given in [43], a term ground term G satisfies the formula "*somewhere*"*A* if and only if there exists $G' \sqsubseteq G$ such that $G' \models A$. Quite surprisingly, such a modality is expressible in the logic. In fact, in case of ground terms, the previous requirement is the semantics of the derived connective $\Diamond$, defined in Tab. 3.4.

Proposition 26. *For every ground term G:*

$$G \models \Diamond A \text{ if and only if there exists } G' \sqsubseteq G \text{ such that } G' \models A.$$

Proof. First prove a supporting property characterising the relation between a term and its sublocations.

Property 2. *For every ground term G and G', it holds: $G' \sqsubseteq G$ if and only if there exists a term C such that $\tau(C)$ and $G \equiv C \circ G'$.*

The direction from right to left is a simple application of Definition 21. The direction from left to right is proved by induction on Definition 21. For the *basic step*, the implication clearly holds if $G' \sqsubseteq G$ in case $G' \equiv G$. The *inductive step* distinguishes two cases.

If $G' \sqsubseteq G$ is due to the fact that $G \equiv G_1 \otimes G_2$, with $G' \sqsubseteq G_1$ or $G' \sqsubseteq G_2$. Without loss of generality, assume $G' \sqsubseteq G_1$. The induction says that there exists C such that $\tau(C)$ and $G_1 \equiv C \circ G'$. Hence, $G \equiv (C \circ G') \otimes G_2$. Now the typing is: $C : I_C \to J_C$; $G' : \epsilon \to I_C$; $G_2 : \epsilon \to J_2$; and $G : \epsilon \otimes \epsilon \to J_C \otimes J_2$. So $G \equiv (C \circ G') \otimes (G_2 \circ id_\epsilon)$. As the interface ϵ is the neutral element for the tensor product between interfaces, compose $C \otimes G_2 : I_C \otimes \epsilon \to J_C \otimes J_2$, and $G' \otimes id_\epsilon : \epsilon \otimes \epsilon \to I_C \otimes \epsilon$. Hence the term $(C \otimes G_2) \circ (G' \otimes id_\epsilon)$ is defined. Note that $\tau(C \otimes G_2)$ is true,as $\tau(G_2)$ is verified since $G_2 : \epsilon \to J_2$ and $\tau(C)$ is true by induction. Hence, by bifunctoriality property, conclude $G \equiv (C \otimes G_2) \circ G'$, with $\tau(C \otimes G_2)$, as aimed.

On the other hand, if $G' \sqsubseteq G$ is due to the fact that $G \equiv G_1 \circ G_2$, with $\tau(G_1)$ and $G' \sqsubseteq G_2$. The induction says that there exists C such that $\tau(C)$ and $G_2 \equiv C \circ G'$. Hence, $G \equiv G_1 \circ (C \circ G')$. Conclude $G \equiv (G_1 \circ C) \circ G'$, with $\tau(G_1 \circ C)$.

Suppose now that $G \models \Diamond A$, this means that $G \models (\mathbf{T} \circ A)_\epsilon$. According to Tab. 3.3, this means that there exist C and G' such that $G' \models A$ and $\tau(C)$, and $G \equiv C \circ G'$. Finally, by Property 2, this means $G' \sqsubseteq G$ and $G' \models A$. □

The *everywhere* modality ($\boxminus$) is dual to $\Diamond$. A term satisfies the formula $\boxminus A$ if each of its sublocations satisfies A.

3.4.2 Logical Properties Deriving from Categorical Axioms

For every axiom of the model, the logic proves a corresponding property. In particular, the bifunctoriality property is expressed by formulae

$$(A_I \circ B_{\to I}) \otimes (A'_J \circ B'_{\to J}) \Leftrightarrow (A_I \otimes A'_J) \circ (B_{\to I} \otimes B'_{\to J})$$

valid when $(I \otimes J)\downarrow$.

In general, given two formulae A, B we say that A *yields* B, and we write $A \vdash B$, if for every term G it is the case that $G \models A$ implies $G \models B$. Moreover, we write $A \dashv\vdash B$ to say both $A \vdash B$ and $B \vdash A$.

Assume that I and J are two interfaces such that their tensor product $I \otimes J$ is defined. Then, the bifunctoriality property in the logic is expressed by

$$(A_I \circ B_{\to I}) \otimes (A'_J \circ B'_{\to J}) \dashv\vdash (A_I \otimes A'_J) \circ (B_{\to I} \otimes B'_{\to J}). \tag{3.2}$$

Proposition 27. *Whenever* $(I \otimes J)\downarrow$*, the equation (3.2) holds in the logic.*

Proof. Prove separately the two way of the satisfaction. First prove $(A_I \circ B_{\to I}) \otimes (A'_J \circ B'_{\to J}) \vdash (A_I \otimes A'_J) \circ (B_{\to I} \otimes B'_{\to J})$. Assume that $G \models (A_I \circ B_{\to I}) \otimes (A'_J \circ B'_{\to J})$. This means that there exist $G' : I' \to I''$, $G'' : J' \to J''$ such that $I' \otimes J'$ and $I'' \otimes J''$ are

defined, and $G \equiv G' \otimes G''$, with $G' \models A_I \circ B_{\to I}$ and $G'' \models A'_J \circ B'_{\to J}$. Now, $G' \models A_I \circ B_{\to I}$ means that there exist G_1 and G_2 such that *(i)* $G' \equiv G_1 \circ G_2$, *(ii)* $G_1 : I \to J'$, with $\tau(G_1)$ and $G_1 \models A$, and *(iii)* $G_2 : I' \to I$, with $G_2 \models B$. Similarly, $G'' \models A'_J \circ B'_{\to J}$ means *(i)* $G'' \equiv G'_1 \circ G'_2$ and *(ii)* $G'_1 : J \to J''$, with $\tau(G'_1)$ and $G'_1 \models A'$, and *(iii)* $G'_2 : I'' \to J$, with $G_2 \models B'$. In particular, conclude $G \equiv (G_1 \circ G_2) \otimes (G'_1 \circ G'_2)$. As $I \otimes J$ is defined, $(G_1 \otimes G'_1) \circ (G_2 \otimes G'_2)$ is an admissible composition. The bifunctoriality property implies $G \equiv (G_1 \otimes G'_1) \circ (G_2 \otimes G'_2)$. Moreover $\tau(G_1 \otimes G'_1)$, as $\tau(G_1)$ and $\tau(G'_1)$. Hence conclude that $G \models (A_I \otimes A'_J) \circ (B_{\to I} \otimes B'_{\to J})$, as required.

For the converse, prove $(A_I \otimes A'_J) \circ (B_{\to I} \otimes B'_{\to J}) \vdash (A_I \circ B_{\to I}) \otimes (A'_J \circ B'_{\to J})$. Assume that $G \models (A_I \otimes A'_J) \circ (B_{\to I} \otimes B'_{\to J})$. By following the same lines as before, deduce that $G \equiv (G_1 \otimes G'_1) \circ (G_2 \otimes G'_2)$, where *(i)* $\tau(G_1 \otimes G'_1)$, *(ii)* $G_1 : I \to J'$ such that $G_1 \models A$, *(iii)* $G'_1 : J \to J''$ such that $G'_1 \models A'$, *(iv)* $G_2 : I' \to I$ such that $G_2 \models B$, and *(v)* $G'_2 : I'' \to J$ such that $G_2 \models B'$. Also in this case, the tensor product of the required interfaces can be performed. Hence compose $(G_1 \circ G_2) \otimes (G'_1 \circ G'_2)$. Again, the bifunctoriality property implies $G \equiv (G_1 \circ G_2) \otimes (G'_1 \circ G'_2)$. Finally, by observing that $\tau(G_1 \otimes G'_1)$ implies $\tau(G_1)$ and $\tau(G'_1)$, deduce $G_1 \circ G_2 \models (A_I \circ B_{\to I})$ and $(G'_1 \circ G'_2) \models (A'_J \circ B'_{\to J})$. Then conclude $G \models (A_I \circ B_{\to I}) \otimes (A'_J \circ B'_{\to J})$. □

3.5 BiLog: Instances and Encodings

In this section BiLog is instantiated to describe place graphs, link graphs and bigraphs. A spatial logic for bigraphs is a natural composition of a place graph logic, for tree contexts, and a link graph logic, for name linkings. Each instance admits an embedding of a well known spatial logic.

3.5.1 Place Graph Logic

Place graphs are essentially ordered lists of regions hosting unordered labelled trees with holes, namely contexts for trees. Tree labels correspond to controls $\mathsf{K} : 1 \to 1$ belonging to a fixed signature $\mathcal{K}$. The monoid of interfaces is the monoid $(\omega, +, 0)$ of finite ordinals m, n. Ordinals represent the number of holes and regions of place graphs. Place graph terms are generated from the set

$$\Theta = \{1 : 0 \to 1, id_n : n \to n, join : 2 \to 1, \gamma_{m,n} : m + n \to n + m\} \cup \mathcal{K}$$

The only structured terms are the controls K, representing regions containing a single node with a hole inside. All the other constructors are *placings* and represent trees $m \to n$ with no nodes: the place identity id_n is neutral for composition; the constructor 1 represents a barren region; *join* is a mapping of two regions into one; $\gamma_{m,n}$ is a permutation that interchanges the first m regions with the following n. The structural congruence $\equiv$ for place graph terms is refined, in Tab. 3.5, by the usual axioms for symmetry of $\gamma_{m,n}$ and by the place axioms that essentially turn the operation $join \circ (\_ \otimes \_)$ in a commutative monoid with 1 as neutral element. In particular, the places generated by composition and

Table 3.5 Additional Axioms for Place Graphs Structural Congruence

Symmetric Category Axioms:			
$\gamma_{m,0}$	$\equiv$	id_m	Symmetry Id
$\gamma_{m,n} \circ \gamma_{n,m}$	$\equiv$	$id_{m \otimes n}$	Symmetry Composition
$\gamma_{m',n'} \circ (G \otimes F)$	$\equiv$	$(F \otimes G) \circ \gamma_{m,n}$	Symmetry Monoid
Place Axioms:			
$join \circ (1 \otimes id_1)$	$\equiv$	id_1	Unit
$join \circ (join \otimes id_1)$	$\equiv$	$join \circ (id_1 \otimes join)$	Associativity
$join \circ \gamma_{1,1}$	$\equiv$	$join$	Commutativity

tensor product from $\gamma_{m,n}$ are *permutations*. A place graph is *prime* if it has type $I \to 1$, namely it has a single region.

Example 8. The term

$$G \stackrel{\text{def}}{=} (service \circ (join \circ (name \otimes description))) \otimes (push \circ 1)$$

is a place graph of type $2 \to 2$, on a signature containing *service*, *name*, *description*, and *push*. It represents an ordered pair of trees. The first tree is labelled *service* and has *name* and *description* as (unordered) children, both children are actually contexts with a single hole. The second tree is ground as it has a single node without children. The term G is congruent to $(service \otimes push) \circ (join \otimes 1) \circ (description \otimes name)$. Such a contextual pair of trees can be interpreted as semi-structured partial data (e.g. an XML message, a web service descriptor) that can be filled by composition. The order among holes is a major issue in the composition, for instance, $(\mathsf{K}_1 \otimes \mathsf{K}_2) \circ (\mathsf{K}_3 \otimes 1)$ is different from $(\mathsf{K}_1 \otimes \mathsf{K}_2) \circ (1 \otimes \mathsf{K}_3)$, as node K_3 plugs into K_1 in the first case, and inside K_2 in the second one.

Fixed the transparency predicate τ on each control in $\mathcal{K}$, the Place Graph Logic PGL$(\mathcal{K}, \tau)$ is $BiLog(\omega, +, 0, \equiv, \mathcal{K} \cup \{1, join, \gamma_{m,n}\}, \tau)$. We assume the transparency predicate τ to hold for *join* and $\gamma_{m,n}$. The statement of Theorem 23 can be extended to PGL by using a similar proof, thus such a logic can describe place graphs precisely. The logic resembles a propositional spatial tree logic, in the style of [36]. The main differences are that PGL models contexts of trees and that the tensor product is not commutative, unlike the parallel composition in [36], and it enables the modelling of the order among regions. The logic can express a commutative separation by using **join** and the tensor product, namely the *parallel composition* operator $A \mid B \stackrel{\text{def}}{=} \mathbf{join} \circ (A_{\to 1} \otimes B_{\to 1})$. At the term level, this separation, which is purely structural, corresponds to $join \circ (P_1 \otimes P_2)$, that is a total operation on all prime place graphs. More precisely, the semantics says that $P \models A \mid B$ means that there exist $P_1 : I_1 \to 1$ and $P_2 : I_2 \to 1$ such that: $P \equiv join \circ (P_1 \otimes P_2)$ and $P_1 \models A$ and $P_2 \models B$.

Table 3.6 Information Tree Terms (over Λ) and Congruence

T, T'	::=	0	empty tree consisting of a single root node
		$a[T]$	single edge tree labelled $a \in \Lambda$ leading to the subtree T
		$T \mid T'$	tree obtained by merging the roots of the trees T and T'

$T \mid 0$	$\equiv$	T	neutral element
$T \mid T'$	$\equiv$	$T' \mid T$	commutativity
$(T \mid T') \mid T''$	$\equiv$	$T \mid (T' \mid T'')$	associativity

Table 3.7 Propositional Spatial Tree Logic

A, B	::=	$\mathbf{F}$	anything	$a[A]$ location
		$\mathbf{0}$	empty tree	$A@a$ location adjunct
		$A \Rightarrow B$	implication	$A \mid B$ composition
		$A \triangleright B$	composition adjunct	

T	$\models_{\text{STL}}$	$\mathbf{F}$	iff	never
T	$\models_{\text{STL}}$	$\mathbf{0}$	iff	$F \equiv 0$
T	$\models_{\text{STL}}$	$A \Rightarrow B$	iff	$T \models_{\text{STL}} A$ implies $T \models_{\text{STL}} B$
T	$\models_{\text{STL}}$	$a[A]$	iff	there exists T' s.t. $T \equiv a[T']$ and $T' \models_{\text{STL}} A$
T	$\models_{\text{STL}}$	$A@a$	iff	$a[T] \models_{\text{STL}} A$
T	$\models_{\text{STL}}$	$A \mid B$	iff	there exists T_1, T_2 s.t.
				$T \equiv T_1 \mid T_2$ and $T_1 \models_{\text{STL}} A$ and $T_2 \models_{\text{STL}} B$
T	$\models_{\text{STL}}$	$A \triangleright B$	iff	for every T': if $T' \models_{\text{STL}} A$ implies $T \mid T' \models_{\text{STL}} B$

3.5.2 Encoding STL

Not surprisingly, prime ground place graphs are isomorphic to the unordered trees modelling the static fragment of ambient logic. Here we show that, when the transparency predicate is always verified, BiLog restricted to prime ground place graphs is equivalent to the propositional Spatial Tree Logic of [36] (STL in the following). The logic STL expresses properties of unordered labelled trees T constructed from the empty tree 0, the labelled node containing a tree $a[T]$, and the parallel composition of trees $T_1 \mid T_2$, as detailed in Tab. 3.6. Labels a are elements of a denumerable set Λ. STL is a static fragment of the ambient logic [42] and it is characterised by the usual classical propositional connectives, the spatial connectives 0, $a[A]$, $A \mid B$, and their adjuncts $A@a$, $A \triangleright B$. The language of the logic and its semantics is outlined in Tab. 3.7.

Table 3.8 encodes the tree model of STL into prime ground place graphs, and STL operators into PGL operators. We assume a bijective encoding between labels and controls, and we associate every label a with a distinct control $\mathsf{K}(a)$ of arity 0. As already said, we

Table 3.8 Encoding STL in PGL over Prime Ground Place Graphs

Trees into Prime Ground Place Graphs

$$[\![0]\!] \stackrel{\text{def}}{=} \mathbf{1} \qquad [\![a[T]]\!] \stackrel{\text{def}}{=} \mathsf{K}(a) \circ [\![T]\!]$$
$$[\![T_1 \mid T_2]\!] \stackrel{\text{def}}{=} join \circ ([\![T_1]\!] \otimes [\![T_2]\!])$$

STL formulae into PGL formulae

$$[\![\mathbf{0}]\!] \stackrel{\text{def}}{=} 1 \qquad [\![a[A]]\!] \stackrel{\text{def}}{=} \mathsf{K}(a) \circ_1 [\![A]\!]$$
$$[\![\mathbf{F}]\!] \stackrel{\text{def}}{=} \mathbf{F} \qquad [\![A@a]\!] \stackrel{\text{def}}{=} \mathsf{K}(a) \leftarrow_1 [\![A]\!]$$
$$[\![A \Rightarrow B]\!] \stackrel{\text{def}}{=} [\![A]\!] \Rightarrow [\![B]\!] \qquad [\![A \mid B]\!] \stackrel{\text{def}}{=} [\![A]\!] \mid [\![B]\!]$$
$$[\![A \triangleright B]\!] \stackrel{\text{def}}{=} ([\![A]\!] \mid \mathbf{id}_1) \leftarrow_1 [\![B]\!]$$

assume the transparency predicate to be verified on every control. The monoidal properties of parallel composition are guaranteed by the symmetry and unit axioms of *join*. The equations are self-explanatory once we remark that: *(i)* the parallel composition of STL is the structural commutative separation of PGL; *(ii)* tree labels can be represented by the corresponding controls of the place graph; *(iii)* location and composition adjuncts of STL are encoded by the left composition adjunct, as they add logically expressible contexts to the tree. This encoding is actually a bijection tree to prime ground place graphs. In fact, there is an *inverse encoding* $(\!|\ |\!)$ for prime ground place graphs in trees defined on the normal forms of [101].

The theorem of discrete normal form in [101] implies that every ground place graph $g : 0 \to 1$ can be expressed as

$$g = join_n \circ (M_0 \otimes \ldots \otimes M_{n-1}) \tag{3.3}$$

where every M_j is a molecular prime ground place graph of the form $M = \mathsf{K}(a) \circ g$, with $ar(\mathsf{K}(a)) = 0$. As an auxiliary notation, $join_n$ is inductively defined as $join_0 \stackrel{\text{def}}{=} 1$, and $join_{n+1} \stackrel{\text{def}}{=} join \circ (id_1 \otimes join_n)$. The bifunctoriality property implies

$$join_n \circ (M_0 \otimes \ldots \otimes M_{n-1}) \equiv$$
$$\equiv join \circ (M_0 \otimes (join \circ (M_1 \otimes (join \circ (\ldots \otimes (join \circ (M_{n-2} \otimes M_{n-1}))))))).$$

The work in [101] says that the normal form in (3.3) is unique, up to permutations.

For every prime ground place graph, the inverse encoding $(\!|\ |\!)$ considers its discrete normal form and it is inductively defined as follows

$$(\!| join_0 |\!) \stackrel{\text{def}}{=} 0$$
$$(\!| \mathsf{K}(a) \circ q |\!) \stackrel{\text{def}}{=} a[(\!| q |\!)]$$
$$(\!| join_s \circ (M_0 \otimes \ldots \otimes M_{s-1}) |\!) \stackrel{\text{def}}{=} (\!| M_0 |\!) \mid \ldots \mid (\!| M_{s-1} |\!)$$

The encodings $[\![\ ]\!]$ and $(\!(\ )\!)$ are one the inverse of the other, hence they give a bijection from trees to prime ground place graphs, which is fundamental in the proof of the following theorem.

Theorem 24 (Encoding STL). *For each tree T and formula A of STL:*

$$T \models_{\text{STL}} A \quad \textit{if and only if} \quad [\![T]\!] \models [\![A]\!].$$

Proof. The theorem is proved by structural induction on STL formulae. The transparency predicate is not considered here, as it holds on every control. The basic step deals with the constants **F** and **0**. Case **F** follows by definition. For the case **0**, $[\![T]\!] \models [\![\mathbf{0}]\!]$ means $[\![T]\!] \models 1$, that by definition is $[\![T]\!] \equiv 1$ and so $T \equiv (\!(\, [\![T]\!] \,)\!) \equiv (\!(\, 1 \,)\!) \stackrel{\text{def}}{=} 0$, namely $T \models_{\text{STL}} \mathbf{0}$.

The inductive steps deal with connectives and modalities.

Case $A \Rightarrow B$**.** Assuming $[\![T]\!] \models [\![A \Rightarrow B]\!]$ means $[\![T]\!] \models [\![A]\!] \Rightarrow [\![B]\!]$; by definition this says that $[\![T]\!] \models [\![A]\!]$ implies $[\![T]\!] \models [\![B]\!]$. By induction hypothesis, this is equivalent to say that $T \models_{\text{STL}} A$ implies $T \models_{\text{STL}} B$, namely $T \models_{\text{STL}} A \Rightarrow B$.

Case $a[A]$**.** Assuming $[\![T]\!] \models [\![a[A]]\!]$ means $[\![T]\!] \models \mathsf{K}(a) \circ_1 ([\![A]\!])$. This amount to say that there exist $G : 1 \to 1$ and $g : 0 \to 1$ such that $[\![T]\!] \equiv G \circ g$ and $G \models \mathsf{K}(a)$ and $g \models [\![A]\!]$, that is $[\![T]\!] \equiv \mathsf{K}(a) \circ g$ with $g \models [\![A]\!]$. Since the encoding is bijective, this is equivalent to $T \equiv (\!(\, \mathsf{K}(a) \circ g \,)\!) \stackrel{\text{def}}{=} a[(\!(\, g \,)\!)]$ with $g \models [\![A]\!]$. Since $g : 0 \to 1$, the induction hypothesis says that $(\!(\, g \,)\!) \models A$. Hence it is the case that $T \models_{\text{STL}} a[A]$.

Case $A@a$**.** Assuming $[\![T]\!] \models [\![A@a]\!]$ means $[\![T]\!] \models \mathsf{K}(a) \leftarrow_1 A$. This is equivalent to say that for every G such that $G \models \mathsf{K}(a)$, if $(G \circ [\![T]\!])\downarrow$ then $G \circ [\![T]\!] \models [\![A]\!]$. According to the definitions, this is $\mathsf{K}(a) \circ [\![T]\!] \models [\![A]\!]$, and so $[\![a[T]]\!] \models [\![A]\!]$. By induction hypothesis, this is $a[T] \models_{\text{STL}} A$. Hence $T \models_{\text{STL}} A@a$ by definition.

Case $A \mid B$**.** Assuming that $[\![T]\!] \models [\![A \mid B]\!]$ means $[\![T]\!] \models [\![A]\!] \mid [\![B]\!]$. This is equivalent to say that $[\![T]\!] \models \mathbf{join} \circ ([\![A]\!]_{\to 1} \otimes [\![B]\!]_{\to 1})$, namely there exist $g_1, g_2 : 0 \to 1$ such that $[\![T]\!] \equiv \mathit{join} \circ (g_1 \otimes g_2)$ and $g_1 \models [\![A]\!]$ and $g_2 \models [\![B]\!]$. As the encoding is bijective this means that $T \equiv (\!(\, g_1 \,)\!) \mid (\!(\, g_2 \,)\!)$, and the induction hypothesis says that $(\!(\, g_1 \,)\!) \models A$ and $(\!(\, g_2 \,)\!) \models B$. By definition this is $T \models_{\text{STL}} A \mid B$.

Case $A \triangleright B$**.** Assuming that $[\![T]\!] \models [\![A \triangleright B]\!]$ means $[\![T]\!] \models \mathbf{join}([\![A]\!] \otimes \mathbf{id}_1)) \leftarrow_1 [\![B]\!]$, namely for every $G : 1 \to 1$ such that $G \models \mathbf{join}([\![A]\!] \otimes \mathbf{id}_1)$ it holds $G \circ [\![T]\!] \models [\![B]\!]$. Now, $G : 1 \to 1$ and $G \models \mathbf{join}([\![A]\!] \otimes \mathbf{id}_1)$ means that there exists $g : 0 \to 1$ such that $g \models [\![A]\!]$ and $G \equiv \mathit{join}(g \otimes \mathit{id}_1)$. Hence it is the case that for every $g : 0 \to 1$ such that $g \models [\![A]\!]$ it holds $\mathit{join}(g \otimes \mathit{id}_1) \circ [\![T]\!] \models [\![B]\!]$, that is $\mathit{join}(g \otimes [\![T]\!]) \models [\![B]\!]$ by bifunctoriality property. Since the encoding is a bijection, this is equivalent to say that for every tree T' such that $[\![T']\!] \models [\![A]\!]$ it holds $\mathit{join}([\![T']\!] \otimes [\![T]\!]) \models [\![B]\!]$, that is $[\![T' \mid T]\!] \models [\![B]\!]$. By induction hypothesis, for every T' such that $T' \models_{\text{STL}} A$ it holds $T' \mid T \models_{\text{STL}} B$, that is the semantics of $T \models_{\text{STL}} A \triangleright B$. □

Differently from STL, PGL can also describe structures with several holes and regions. In §3.6 we show how PGL describes contexts of tree-shaped semistructured data. Consider, for instance, a function taking two trees and returning the tree obtained by merging their roots. Such a function is represented by the term *join*, which solely satisfies the formula **join**. Similarly, the function that takes a tree and encapsulates it inside a node *labelled* by K, is represented by the term K and captured by the formula K. Moreover, the formula $\mathbf{join} \circ (\mathsf{K} \otimes (\mathbf{T} \circ \mathbf{id}_1))$ expresses all contexts of form $2 \to 1$ that place their first argument inside a K node and their second one as a sibling of such node.

3.5.3 Link Graph Logic (LGL).

Fixed a denumerable set of names Λ, we consider the monoid $(\mathcal{P}_{fin}(\Lambda), \uplus, \emptyset)$, where $\mathcal{P}_{fin}(\_)$ is the finite powerset operator and $\uplus$ is the subset disjoint union. Link graphs are the structures arising from such a monoid. They can describe nominal resources, common in many areas: object identifiers, location names in memory structures, channel names, and ID attributes in XML documents. The fact that names cannot be implicitly shared does not mean that we can refer to them or link them explicitly (e.g. object references, location pointers, fusion in fusion calculi, and IDREF in XML files). Link graphs describe connections between resources performed by means of names, that are *references*.

Wiring terms are a structured way to map a set of inner names X into a set of outer names Y. They are generated by the constructors: $/a : \{a\} \to \emptyset$ and ${}^{a}/_{X} : X \to a$. The closure $/a$ hides the inner name a in the outer face. The substitution ${}^{a}/_{X}$ associates all the names in the set X to the name a. We denote wirings by ω, substitutions by σ, τ, and bijective substitutions, dubbed *renamings*, by α, β. Substitution can be specialised in: $a \stackrel{\text{def}}{=} {}^{a}/_{\emptyset}$ and $a \leftarrow b \stackrel{\text{def}}{=} {}^{a}/_{\{b\}}$ and $a \leftrightarrows b \stackrel{\text{def}}{=} {}^{a}/_{\{a,b\}}$. The constructor a represents the introduction of name a, the term $a \leftarrow b$ corresponds to rename b to a, and $a \leftrightarrows b$ links, or fuses, a and b to name a.

Given a signature $\mathcal{K}$ of controls K with arity function $ar(\mathsf{K})$ we generate link graphs from wirings and the constructor $\mathsf{K}_{\vec{a}} : \emptyset \to \vec{a}$ with $\vec{a} = a_1, \ldots, a_k$, $\mathsf{K} \in \mathcal{K}$, and $k = ar(\mathsf{K})$. The control $\mathsf{K}_{\vec{a}}$ represents a resource of kind K with named ports $\vec{a}$. Any ports may be connected to other node ports via wiring compositions.

In this case, the structural congruence $\equiv$ is refined as outlined in Tab. 3.9 with obvious axioms for links, modelling α-conversion and extrusion of closed names. We assume the transparency predicate τ true on wiring constructors.

Fixed the transparency predicate τ for each control in $\mathcal{K}$, the Link Graph Logic $\mathrm{LGL}(\mathcal{K}, \tau)$ is $\mathit{BiLog}(\mathcal{P}_{fin}(\Lambda), \uplus, \emptyset, \equiv, \mathcal{K} \cup \{/a, {}^{a}/_{X}\}, \tau)$. Theorem 23 can be extended to LGL by using a similar proof, thus such a logic describes the link graphs precisely. The logic expresses structural spatiality for resources and strong spatiality (separation) for names, and it can therefore be viewed as a generalisation of Separation Logic for contexts and multi-ports locations. On the other side, the logic can describe resources with local (hidden or private) names between resources, and in this sense the logic is a generalisation of Spatial Graph Logic [39]: it is sufficient to consider the edges as resources.

Table 3.9 Additional Axioms for Link Graph Structural Congruence

Link Axioms:	${}^{a}/_{a}$	$\equiv$	id_a	Link Identity
	$/a \circ {}^{a}/_{b}$	$\equiv$	$/b$	Closing renaming
	$/a \circ a$	$\equiv$	id_ϵ	Idle edge
	${}^{b}/_{(Y \uplus a)} \circ (id_Y \otimes {}^{a}/_{X})$	$\equiv$	${}^{b}/_{Y \uplus X}$	Composing substitutions
Link Node Axiom:	$\alpha \circ \mathsf{K}_{\vec{a}}$	$\equiv$	$\mathsf{K}_{\alpha(\vec{a})}$	Renaming

Moreover, if we consider identity as a constructor, it is possible to define

$$a \leftarrow b \stackrel{\text{def}}{=} (a \leftrightarrows b) \circ (a \otimes id_b).$$

In LGL the formula $A \otimes B$ describes a decomposition into two *separate* link graphs, sharing neither resources, nor names, nor connections, that satisfy A and B respectively. Since it is defined only on link graphs with disjoint inner/outer sets of names, the tensor product is a kind a *spatial/separation* operator, in the sense that it separates the model into two distinct parts that cannot share names.

In this case, horizontal decomposition inherits the commutativity property from the monoidal tensor product. If we want a name a to be shared between separated resources, we need to make the sharing explicit, and the sole way to do that is through the link operation. We therefore need a way to first separate the names occurring in two wirings as to apply the tensor, and then link them back together.

As a shorthand, if $G : X \to Y$ and $G' : X' \to Y'$ with $Y \subset X'$, we write $[G']G$ for $(G' \otimes id_{X' \setminus Y}) \circ G$ and if $\vec{a} = a_1, \ldots, a_n$ and $\vec{b} = b_1, \ldots, b_n$, we write $\vec{a} \leftarrow \vec{b}$ for $a_1 \leftarrow b_1 \otimes \ldots \otimes a_n \leftarrow b_n$, similarly for $\vec{a} \leftrightarrows \vec{b}$. From the tensor product it is possible to derive a product with sharing on $\vec{a}$. Moreover, given $G : X \to Y$ and $G' : X' \to Y'$ with $X \cap X' = \emptyset$, we choose a list $\vec{b}$ (with the same length as $\vec{a}$) of fresh names. The composition with sharing $\vec{a}$ is

$$G \overset{\vec{a}}{\otimes} G' \stackrel{\text{def}}{=} [\vec{a} \leftrightarrows \vec{b}]([\vec{b} \leftarrow \vec{a}]G \otimes G').$$

In this case, the tensor product is well defined since all the common names $\vec{a}$ in G are renamed to fresh names, while the sharing is re-established afterwards by linking the $\vec{a}$ names with the $\vec{b}$ names.

By extending this sharing to all names we define the parallel composition $G \mid G'$ as a total operation. However, such an operator does not behave 'well' with respect to the composition, as shown in [101]. In addition a direct inclusion of a corresponding connective in the logic would impact the satisfaction relation by expanding the finite horizontal decompositions to the boundless possible name-sharing decompositions. (This may be the main reason why logics describing models with name closure and parallel composition are undecidable [53].) This is due to the fact that the set of names shared by a parallel

composition is not known in advance, and therefore parallel composition can only be defined by using an existential quantification over the entire set of shared names.

Names can be internalised and effectively made private to a bigraph by the closure operator $/a$. The effect of composition with $/a$ is to add a new edge with no public name, and therefore to make a disappear from the outerface, and be completely hidden to the outside. Separation is still expressed by the tensor connective, which not only separates places, but also makes sure that no edge – whether visible or hidden – crosses the separating line.

As a matter of fact, without name quantification it is not possible to build formulae that explore a link, since the latter has the effect of hiding names. For this task, we employ the name variables $x_1, ..., x_n$ and the fresh name quantification $\mathcal{N}$. in the style of Nominal Logic [116]. The semantics is defined as

$$G \models \mathcal{N}x_1 \dots x_n.A \quad \textit{iff} \quad \begin{array}{l}\textit{there exist } a_1 \dots a_n \notin fn(G) \cup fn(A) \\ \textit{such that } G \models A\{a_1/x_1 \dots a_n/x_n\},\end{array}$$

where $A\{a_1/x_1 \dots a_n/x_n\}$ is the usual variable substitution.

By fresh name quantification we define a notion of $\vec{a}$-linked name quantification for fresh names, whose purpose is to identify names linked to $\vec{a}$, as

$$\vec{a}\,\mathbf{L}\,\vec{x}.A \stackrel{\text{def}}{=} \mathcal{N}\vec{x}.((\vec{a} \leftleftarrows \vec{x}) \otimes \mathbf{id}) \circ A.$$

The formula above expresses that the variables in $\vec{x}$ denote in A names that are linked in the term to $\vec{a}$, and the role of $(\vec{a} \leftleftarrows \vec{x})$ is to link the fresh names $\vec{x}$ with $\vec{a}$, while **id** deals with names not in $\vec{a}$. We also define a *separation-up-to* as the decomposition in two terms that are separated apart from the link on the specific names in $\vec{a}$, which crosses the separation line:

$$A \overset{\vec{a}}{\otimes} B \stackrel{\text{def}}{=} \vec{a}\,\mathbf{L}\,\vec{x}.(((\vec{x} \leftarrow \vec{a}) \otimes \mathbf{id}) \circ A) \otimes B. \tag{3.4}$$

The idea of the formula above is that the shared names $\vec{a}$ are renamed in fresh names $\vec{x}$, so that the product can be performed and finally $\vec{x}$ is linked to $\vec{a}$ to actually have the sharing.

The following lemma states that the two definition are consistent.

Lemma 18 (Separation-up-to). *If $g \models A \overset{\vec{x}}{\otimes} B$ with $g : \epsilon \to X$, and $\vec{x}$ is the vector of the elements in X, then there exist $g_1 : \epsilon \to X$ and $g_2 : \epsilon \to X$ such that $g \equiv g_1 \overset{\vec{x}}{\otimes} g_2$ and $g_1 \models A$ and $g_2 \models B$.*

Proof. Simply apply the definitions and observe that the identities must be necessarily id_ϵ, as the outer face of g is restricted to be X. □

The corresponding parallel composition operator is not directly definable by using the separation-up-to. In fact, in arbitrary decompositions the name shared are not all known a priori, hence we would not know the vector $\vec{x}$ in the operator sharing/separation operator $\overset{\vec{x}}{\otimes}$. However, next section shows that a careful encoding is possible for the parallel composition of spatial logics with nominal resources.

Table 3.10 Spatial Graph Terms (with Local Names) and Congruence

G, G'	::=	nil		empty graph
		$a(x,y)$		single edge graph labelled $a \in \Lambda$ connecting the nodes x, y
		$G \mid G'$		=composing the graphs G, G', with sharing of nodes
		$(\nu x)G$		the node x is local in G
$G \mid nil$	$\equiv$	G		neutral element
$G \mid G'$	$\equiv$	$G' \mid G$		commutativity
$(G \mid G') \mid G''$	$\equiv$	$G \mid (G' \mid G'')$		associativity
$(\nu x)G$	$\equiv$	$(\nu y)G\{x \leftarrow y\}$		renaming, when $y \notin fn(G)$
$(\nu x)nil$	$\equiv$	nil		extrusion Zero
$G \mid (\nu x)G'$	$\equiv$	$(\nu x)(G \mid G')$		extrusion composition, when $x \notin fn(G)$
$(\nu x)a(y,z)$	$\equiv$	$a(y,z)$		extrusion edge, when $x \neq y, z$
$(\nu x)(\nu y)G$	$\equiv$	$(\nu y)(\nu x)G$		extrusion restriction

3.5.4 Encoding SGL

We show that LGL can be seen as a contextual (multi-edge) version of Spatial Graph Logic (SGL) [39]. The logic SGL expresses properties of directed graphs G with labelled edges. The notation $a(x,y)$ represents an edge from the node x to y and labelled by a. The graphs G are built from the empty graph nil and the edge $a(x,y)$ by using the parallel composition $G_1 \mid G_2$ and the binding for local names of nodes $(\nu x)G$. The syntax and the structural congruence for spatial graphs are outlined in Tab. 3.10.

The graph logic combines standard propositional logic with the structural connectives: composition and basic edge. Although we focus on its propositional fragment, the logics of [39] also includes edge label quantifier and recursion. In [39] SGL is used as a pattern matching mechanism of a query language for graphs. In addition, the logic is integrated with *transducers* to allow graph transformations. The applications of SGL include description and manipulation of semistructured data. Table 3.11 depicts the syntax and the semantics of the fragment we consider.

We consider a signature $\mathcal{K}$ with controls of arity 2, we assume a bijective function associating every label a to a distinct control $\mathsf{K}(a)$. The ports of the controls represent the starting and arrival node of the associated edge. The transparency predicate is defined to be verified on every control. The resulting link graphs are interpreted as contextual graphs with labelled edges, whereas the resulting class of ground link graphs is isomorphic to the graph model of SGL.

Table 3.12 encodes the graphs modelling SGL into ground link graphs and SGL formulae into LGL formulae. The encoding is parametric on a finite set X of names containing the free names of the graph under consideration. Observe that when we force the outer face of the graphs to be a fixed finite set X, the encoding of parallel composition is simply the separation-up-to $\vec{x}$, where $\vec{x}$ is a list of all the elements in X. Notice also

Table 3.11 Propositional Spatial Graph Logic (SGL)

φ, ψ	::=	$\mathbf{F}$	false	$a(x,y)$	an edge from x to y
		$\mathbf{nil}$	empty graph	$\varphi \mid \psi$	composition
		$\varphi \Rightarrow \psi$	implication		

G	$\models_{\text{STL}}$	$\mathbf{F}$	iff	never
G	$\models_{\text{STL}}$	$\mathbf{nil}$	iff	$G \equiv nil$
G	$\models_{\text{STL}}$	$\varphi \Rightarrow \psi$	iff	$G \models_{\text{STL}} \varphi$ implies $G \models_{\text{STL}} \psi$
G	$\models_{\text{STL}}$	$a(x,y)$	iff	$G \equiv a(x,y)$
G	$\models_{\text{STL}}$	$\varphi \mid \psi$	iff	there exists G_1, G_2 s.t. $G \equiv G_1 \mid G_2$ and $G_1 \models_{\text{STL}} \varphi$ and $G_2 \models_{\text{STL}} \psi$

Table 3.12 Encoding Propositional SGL in LGL over Two Ported Ground Link Graphs

Spatial Graphs into Two-ported Ground Link Graphs

$$
\begin{aligned}
[\![nil]\!]_X &\stackrel{\text{def}}{=} X \\
[\![a(x,y)]\!]_X &\stackrel{\text{def}}{=} \mathsf{K}(a)_{x,y} \otimes X \setminus \{x,y\} \\
[\![(\nu x)G]\!]_X &\stackrel{\text{def}}{=} ((/x \otimes id_{X\setminus\{x\}}) \circ [\![G]\!]_{\{x\}\cup X})) \otimes (\{x\} \cap X) \\
[\![G \mid G']\!]_X &\stackrel{\text{def}}{=} [\![G]\!]_X \stackrel{\vec{x}}{\otimes} [\![G']\!]_X
\end{aligned}
$$

SGL formulae into LGL formulae

$$
\begin{aligned}
[\![\mathbf{nil}]\!]_X &\stackrel{\text{def}}{=} X & \qquad [\![a(x,y)]\!]_X &\stackrel{\text{def}}{=} \mathsf{K}(a)_{x,y} \otimes (X \setminus \{x,y\}) \\
[\![\mathbf{F}]\!]_X &\stackrel{\text{def}}{=} \mathbf{F} & \qquad [\![\varphi \Rightarrow \psi]\!]_X &\stackrel{\text{def}}{=} [\![\varphi]\!]_X \Rightarrow [\![\psi]\!]_X \\
[\![\varphi \mid \psi]\!]_X &\stackrel{\text{def}}{=} [\![\varphi]\!]_X \stackrel{\vec{x}}{\otimes} [\![\psi]\!]_X
\end{aligned}
$$

how local names are encoded into name closures. Thanks to the Connected Normal Form of [101], it is easy to prove that ground link graphs featuring controls with exactly two ports are isomorphic to spatial graph models. As we impose a bijection between arrows labels and controls, the signature and the label set must have the same cardinality.

Lemma 19 (Isomorphism for Spatial Graphs). *There is a mapping* $(\!|\ |\!)$ *from two-ported ground bigraphs to spatial graphs, such that for every set X of names:*

1. *The mapping* $(\!|\ |\!)$ *is inverse to* $[\![\ ]\!]_X$.
2. *For every ground link graph g with outer face X in the signature featuring a countable set of controls* K*, all with arity* 2*, it holds* $fn((\!|\, g \,|\!)) = X$ *and* $[\![(\!|\, g \,|\!)]\!]_X \equiv g$.
3. *For every spatial graph G with* $fn(G) = X$ *it holds* $[\![G]\!]_X : \epsilon \to X$ *and* $(\!|\, [\![G]\!]_X \,|\!) \equiv G$.

Proof. The idea is to interpret link graphs as bigraphs of type $\epsilon \to \langle 1, X\rangle$ without nested nodes. As proved in [101], bigraphs without nested nodes and $\langle 1, X\rangle$ as outerface have the following normal form (where $Z \subseteq X$):

$$G ::= (/Z \mid id_{\langle 1,X\rangle}) \circ (X \mid M_0 \mid \ldots \mid M_{k-1})$$
$$M ::= \mathsf{K}_{x,y}(a) \circ 1$$

The inverse encoding is based on such a normal form:

$$(\!|\, (/Z \mid id_{\langle 1,X\rangle}) \circ (X \mid M_0 \mid \ldots \mid M_{k-1}) \,|\!) \stackrel{\text{def}}{=} (\nu Z)\,(\mathit{nil} \mid (\!|\, M_0 \,|\!) \mid \ldots \mid (\!|\, M_{k-1} \,|\!))$$
$$(\!|\, \mathsf{K}_{x,y}(a) \circ 1 \,|\!) \stackrel{\text{def}}{=} a(x,y)$$

Notice that the extrusion properties of local names correspond to node and link axioms. The encodings $[\![\ ]\!]$ and $(\!|\ |\!)$ provide a bijection, up to congruence, between graphs of SGL with free names X and ground link graphs with outer face X and built by controls of arity two. □

The previous lemma is fundamental in proving the soundness of the encoding for SGL in BiLog, stated in the following theorem.

Theorem 25 (Encoding SGL). *For every graph G, every finite set X that contains $fn(G)$, and every formula φ of the propositional fragment of SGL:*

$$G \models_{\text{SGL}} \varphi \quad \textit{if and only if} \quad [\![G]\!]_X \models [\![\varphi]\!]_X.$$

Proof. By induction on formulae of SGL. The transparency predicate is not considered here, as it is verified on every control. The basic step deals with the constants **F**, **nil** and $a(x,y)$. Case **F** follows by definition. For the case **nil**, $[\![G]\!]_X \models [\![\mathbf{nil}]\!]_X$ means $[\![G]\!]_X \models X$, that by definition is $[\![G]\!]_X \equiv X$ and so $G \equiv (\!|\, [\![G]\!]_X \,|\!) \equiv (\!|\, X \,|\!) \stackrel{\text{def}}{=} \mathit{nil}$, namely $G \models_{\text{SGL}} \mathbf{nil}$. For the case $a(x,y)$, to assume $[\![G]\!]_X \models [\![a(x,y)]\!]_X$ means $[\![G]\!]_X \models \mathsf{K}(a)_{x,y} \otimes X \setminus \{x,y\}$. So $G \equiv (\!|\, [\![G]\!]_X \,|\!) \equiv (\!|\, \mathsf{K}(a)_{x,y} \otimes X \setminus \{x,y\} \,|\!) \equiv a(x,y)$, that is $G \models_{\text{SGL}} a(x,y)$. The inductive steps deal with connectives.

Case $\varphi \Rightarrow \psi$. To assume $[\![G]\!]_X \models [\![\varphi \Rightarrow \psi]\!]_X$ means $[\![G]\!]_X \models [\![\varphi]\!]_X \Rightarrow [\![\psi]\!]_X$; by definition this says that $[\![G]\!]_X \models [\![\varphi]\!]_X$ implies $[\![G]\!]_X \models [\![\psi]\!]_X$. By induction hypothesis, this is equivalent to say that $G \models_{\text{SGL}} \varphi$ implies $G \models_{\text{SGL}} \psi$, namely $G \models_{\text{SGL}} \varphi \Rightarrow \psi$.

Case $\varphi \mid \psi$. To assume $[\![G]\!]_X \models [\![\varphi \mid \psi]\!]_X$ means $[\![G]\!]_X \models [\![\varphi]\!]_X \stackrel{\vec{x}}{\otimes} [\![\psi]\!]_X$. By Lemma 18 there exists g_1, g_2 such that $[\![G]\!]_X \equiv g_1 \stackrel{\vec{x}}{\otimes} g_2$ and $g_1 \models [\![\varphi]\!]_X$ and $g_2 \models [\![\psi]\!]_X$. Let $G_1 = (\!|\, g_1 \,|\!)$ and $G_2 = (\!|\, g_2 \,|\!)$, Lemma 19 says that $[\![G_1]\!]_X \equiv g_1$ and $[\![G_2]\!]_X \equiv g_2$, and by conservation of congruence, $[\![G_1]\!]_X \models [\![\varphi]\!]_X$ and $[\![G_2]\!]_X \models [\![\psi]\!]_X$. Hence the induction hypothesis says that $G_1 \models_{\text{SGL}} \varphi$ and $G_2 \models_{\text{SGL}} \psi$. In addition $[\![G_1 \mid G_2]\!]_X \equiv [\![G_1]\!]_X \stackrel{\vec{x}}{\otimes} [\![G_2]\!]_X \equiv g_1 \stackrel{\vec{x}}{\otimes} g_2 \equiv [\![G]\!]_X$. Conclude that G admits a parallel decomposition with parts satisfying A and B, thus $G \models_{\text{SGL}} \varphi \mid \psi$. □

Also, LGL enables the encoding of Separation Logics on heaps: names used as identifiers of location are forcibly separated by tensor product, while names used for pointers are shared/linked. However we do not encode it explicitly since in §3.5.7 we will encode a more general logic: the Context Tree Logic [37].

3.5.5 Pure Bigraph Logic

By combining link graphs and place graphs we generate all the *(abstract pure) bigraphs* of [90]. In this case the underlying monoid is the product of link and place interfaces, namely $(\omega \times \mathcal{P}_{fin}(\Lambda), \otimes, \epsilon)$ where $\langle m, X\rangle \otimes \langle n, X\rangle \stackrel{\text{def}}{=} \langle m+n, X \uplus Y\rangle$ and $\epsilon \stackrel{\text{def}}{=} \langle 0, \emptyset\rangle$. As a short notation, we use X for $\langle 0, X\rangle$ and n for $\langle n, \emptyset\rangle$.

A set of constructors for bigraphical terms is obtained as the union of place and link graph constructors, except the controls which are subsumed by the new *discrete ion* constructors, denoted by $\mathsf{K}_{\vec{d}} : 1 \to \langle 1, \vec{d}\rangle$. It represents a prime bigraph containing a single node with ports named $\vec{d}$ and an hole inside. Bigraphical terms are thus defined in relation to a control signature $\mathcal{K}$ and a set of names Λ, as detailed in [101].

The structural congruence for bigraphs corresponds to the sound and complete bigraph axiomatisation of [101]. The additional axioms are reported in Tab. 3.13: they are essentially a combination of the axioms for link and place graphs, with slight differences due to the interfaces monoid. In detail, we define the symmetry as $\gamma_{I,J} \stackrel{\text{def}}{=} \gamma_{m,n} \otimes id_{X \uplus Y}$ where $I = \langle m, X\rangle$ and $J = \langle n, Y\rangle$, and we restate the node axiom by taking care of the places.

PGL excels at expressing properties of *unnamed* resources, that are resources accessible only by following the structure of the term. On the other hand, LGL characterises names and their links to resources, but it has no notion of locality. A combination of them ought to be useful to model nominal spatial structures, either private or public.

BiLog promises to be a good (contextual) spatial logic for (semi-structured) resources with nominal links, thanks to bigraphs' orthogonal treatment of locality and connectivity. To testify this, 3.5.7 shows how recently proposed Context Logic for Trees (CTL) [37] can be encoded into bigraphs. The idea of the encoding is to extend the encoding of STL with (single-hole) contexts and identified nodes. First, 3.5.6 gives some details on the transparency predicate.

3.5.6 Transparency on Bigraphs

In the logical framework we gave the minimal restrictions on the transparency predicate to prove our results. Here we show a way to define a transparency predicate. The most natural way is to make the transparent terms a sub-category of the more general category of terms. This essentially means to impose the product and the composition of two transparent terms to be transparent. Thus transparency on all terms can be derived from a transparency policy, i.e., a predicate $\tau_\Theta(\,)$ defined only on the constructors as follows.

Table 3.13 Additional Axioms for Bigraph Structural Congruence

Symmetric Category Axioms:			
$\gamma_{I,\epsilon}$	$\equiv$	id_I	Symmetry Id
$\gamma_{I,J} \circ \gamma_{J,I}$	$\equiv$	$id_{I \otimes J}$	Symmetry Composition
$\gamma_{I',J'} \circ (G \otimes F)$	$\equiv$	$(F \otimes G) \circ \gamma_{I,J}$	Symmetry Monoid
Place Axioms:			
$join \circ (1 \otimes id_1)$	$\equiv$	id_1	Unit
$join \circ (join \otimes id_1)$	$\equiv$	$join \circ (id_1 \otimes join)$	Associativity
$join \circ \gamma_{1,1}$	$\equiv$	$join$	Commutativity
Link Axioms:			
${}^{a}/_{a}$	$\equiv$	id_a	Link Identity
$/a \circ {}^{a}/_{b}$	$\equiv$	$/b$	Closing renaming
$/a \circ a$	$\equiv$	id_ϵ	Idle edge
${}^{b}/_{(Y \uplus a)} \circ (id_Y \otimes {}^{a}/_{X})$	$\equiv$	${}^{b}/_{Y \uplus X}$	Composing substitutions
Node Axiom:			
$(id_1 \otimes \alpha) \circ \mathsf{K}_{\vec{a}}$	$\equiv$	$\mathsf{K}_{\alpha(\vec{a})}$	Renaming

Definition 22 (Transparency). *Given the monoid of interfaces $(\mathcal{M}, \otimes, \epsilon)$, the set of constructors Θ, the congruence $\equiv$ and a transparency policy predicate τ_Θ defined on the constructors in Θ we define the transparency on terms as follows:*

$$\frac{G \equiv id_I}{\tau(G)} \qquad \frac{\exists I.G : \epsilon \to I}{\tau(G)} \qquad \frac{G \equiv \Omega \quad \tau_\Theta(\Omega)}{\tau(G)}$$

$$\frac{G \equiv G_1 \otimes G_2 \quad \tau(G_1) \quad \tau(G_2)}{\tau(G)} \qquad \frac{G \equiv G_1 \circ G_2 \quad \tau(G_1) \quad \tau(G_2)}{\tau(G)}$$

Next lemma proves that the conditions we required on the transparency predicate holds for this particular definition.

Lemma 20 (Transparency Properties). *If G is ground or G is an identity then $\tau(G)$ is verified. Moreover, if $G \equiv G'$ then $\tau(G)$ is equivalent to $\tau(G')$.*

Proof. The former statement is verified by definition. The latter is proved by induction on the derivations. □

We assume every bigraphical constructor, which is not a control, to be transparent and the transparency policy to be defined only on the controls. The transparency the policy can be defined, for instance, by security requirements.

Table 3.14 Trees with Pointers and Tree Contexts

T, T'	::=	0	empty tree
		$a_x[T]$	a tree labelled a with identifier x and subtree T
		$T \mid T'$	partial parallel composition
C	::=	$-$	an hole (the identity context)
		$a_x[C]$	a tree context labelled a with identifier x and subtree C
		$T \mid C$	context right parallel composition
		$C \mid T$	context left parallel composition

3.5.7 Encoding CTL

Paper [37] presents a spatial context logic to describe programs manipulating a tree structured memory. The model of the logic is the set of unordered labelled trees T and *linear contexts* C, which are trees with a unique hole. Every node has a name, so to identify memory locations. From the model, the logic is dubbed Context Tree Logic, CTL in the following. Given a denumerable set of labels and a denumerable set of identifiers, trees and contexts are defined in Tab. 3.14: a represents a label and x an identifier. The insertion of a tree T in a context C, denoted by $C(T)$, is defined in the standard way, and corresponds to fill the unique hole of C with the tree T. A *well formed tree* or *context* is one where the node identifiers are unique. The model of the logic is composed by trees and contexts that are well formed. In particular, composition, node formation and tree insertion are *partial* as they are restricted to well-formed trees. The structural congruence between trees is the smallest congruence that makes the parallel operator to be commutative, associative and with the empty tree as neutral element. Such a congruence is naturally extended to contexts.

The logic exhibits two kinds of formulae: P, describing trees, and K, describing tree contexts. It has two spatial constants, the empty tree for P and the hole for K, and four spatial operators: the node formation $a_x[K]$, the application $K(P)$, and its two adjuncts $K \triangleright P$ and $P_1 \triangleleft P_2$. The formula $a_x[K]$ describes a context with a single root labelled by a and identified by x, whose content satisfies K. The formula $K \triangleright P$ represents a tree that satisfies P whenever inserted in a context satisfying K. Dually, $P_1 \triangleleft P_2$ represents contexts that composed with a tree satisfying P_1 produce a tree satisfying P_2. The complete syntax of the logic is outlined in Tab. 3.15, the semantics in 3.16.

CTL can be naturally embedded in an instance of BiLog. The complete structure of the Context Tree Logic has also link values. For sake of simplicity, we restrict our attention to the fragment without links. As already said, the terms giving a semantics to CTL do not to share identifiers: two nodes cannot have the same identifier, as it represents a precise location in the memory. This is easily obtained with bigraph terms by encoding the identifiers as names and the composition as tensor product, that separates them. We encode such a structure in BiLog by lifting the application to a particular kind

Table 3.15 Context Tree Logic (CTL)

P, P'	::=	*false*	
		$\mathbf{0}$	empty tree formula
		$K(P)$	context application
		$K \triangleleft P$	context application adjunct
		$P \Rightarrow P'$	implication
K, K'	::=	*false*	
		$-$	identity context formula
		$a_x[K]$	node context formula
		$P \triangleright P'$	context application adjunct
		$P \mid K$	parallel context formula
		$K \Rightarrow K'$	implication

of composition, and similarly for the two adjuncts.

The tensor product on bigraphs is both a spatial separation, like in the models for STL, and a partially-defined separation on names, like pointer composition for separation logic. Since we deal with both names and places, we define a formula $\mathbf{id}_{\langle m,-\rangle}$ to represent identities on places by constraining the place part of the interface to be fixed and leaving the name part to be free: $\mathbf{id}_{\langle m,-\rangle} \stackrel{\text{def}}{=} \mathbf{id}_m \otimes (\mathbf{id} \wedge \neg(\mathbf{id}_1^{\exists\otimes}))$. The semantics says that $G \models id_{\langle m,-\rangle}$ means that there exits a set of names X such that $G \equiv id_m \otimes id_X$. By using such an identity formula we define the corresponding typed composition $\circ_{\langle m,-\rangle}$ and the typed adjuncts $\leftarrow_{\langle m,-\rangle}$, $\rightarrow_{\langle m,-\rangle}$:

$$\begin{aligned} A \circ_{\langle m,-\rangle} B &\stackrel{\text{def}}{=} A \circ \mathbf{id}_{\langle m,-\rangle} \circ B \\ A \leftarrow_{\langle m,-\rangle} B &\stackrel{\text{def}}{=} (\mathbf{id}_{\langle m,-\rangle} \circ A) \leftarrow B \\ A \rightarrow_{\langle m,-\rangle} B &\stackrel{\text{def}}{=} (A \circ \mathbf{id}_{\langle m,-\rangle}) \leftarrow B \end{aligned}$$

We then define the operator $*$ for the parallel composition with separation operator $*$ as both a term constructor and a logical connective:

$$D * E \stackrel{\text{def}}{=} [join](D \otimes E) \qquad \text{for } D \text{ and } E \text{ prime bigraphs}$$

$$A * B \stackrel{\text{def}}{=} (\mathbf{join} \otimes \mathbf{id}_{\langle 0,-\rangle}) \circ (A_{\rightarrow\langle 1,-\rangle} \otimes B_{\rightarrow\langle 1,-\rangle}) \qquad \text{for } A \text{ and } B \text{ formulae}$$

The operator $*$ enables the encoding of trees and contexts to bigraphs. In particular, we consider a signature with controls of arity 1 and we define the transparency predicate to be verified on every control. Moreover we assume a bijective function from tags to controls: $a_x \longmapsto \mathsf{K}(a)_x$. The details are outlined in Tab. 3.17. The encodings of trees turn out to be *ground prime discrete bigraphs*: bigraphs with open links and type $0 \to \langle 1, X\rangle$. The result in [101] says that the normal form, up to permutations, for ground prime discrete bigraphs is:

$$g = (join_k \otimes id_X) \circ (M_1 \otimes \ldots \otimes M_k),$$

Table 3.16 Semantics for CTL

T	$\models_{\mathcal{T}}$	*false*	iff	never
T	$\models_{\mathcal{T}}$	$\mathbf{0}$	iff	$T \equiv 0$
T	$\models_{\mathcal{T}}$	$K(P)$	iff	there exist C, T' s.t. $C(T')$ well-formed, and $T \equiv C(T')$ and $C \models_{\mathcal{K}} K$ and $T' \models_{\mathcal{T}} P$
T	$\models_{\mathcal{T}}$	$K \triangleleft P$	iff	for every C: $C \models_{\mathcal{K}} K$ and $C(T)$ well-formed implies $C(T) \models_{\mathcal{T}} P$
T	$\models_{\mathcal{T}}$	$P \Rightarrow P'$	iff	$T \models_{\mathcal{T}} P$ implies $T \models_{\mathcal{T}} P'$
C	$\models_{\mathcal{K}}$	*false*	iff	never
C	$\models_{\mathcal{K}}$	$-$	iff	$C \equiv -$
C	$\models_{\mathcal{K}}$	$a_x[K]$	iff	there exists C' s.t. $a_x[C']$ well-formed, and $C \equiv a_x[C']$ and $C' \models_{\mathcal{K}} K$
C	$\models_{\mathcal{K}}$	$P \triangleright P'$	iff	for every T: $T \models_{\mathcal{T}} P$ and $C(T)$ well-formed implies $C(T) \models_{\mathcal{T}} P'$
C	$\models_{\mathcal{K}}$	$P \mid K$	iff	there exist C', T s.t. $T \mid C'$ well-formed, and $C \equiv T \mid C'$ and $T \models_{\mathcal{T}} P$ and $C' \models_K K$
C	$\models_{\mathcal{K}}$	$K \Rightarrow K'$	iff	$C \models_{\mathcal{K}} K$ implies $T \models_{\mathcal{T}} K'$

where M_i are called *discrete ground molecules* and are of the form $M = (\mathsf{K}(a)_x \otimes id_Y)g$. We can now define the reverse encoding $(\![\ ]\!)$ of $[\![\ ]\!]$, from ground prime discrete bigraphs to trees, involving such a normal form:

$$\begin{aligned} (\![join_0]\!) &\stackrel{\text{def}}{=} 0 \\ (\![(\mathsf{K}(a)_x \otimes id_Y) \circ g]\!) &\stackrel{\text{def}}{=} a_x[(\![g]\!)] \\ (\![(join_k \otimes id_Y) \circ (M_1 \otimes \ldots \otimes M_k)]\!) &\stackrel{\text{def}}{=} (\![M_1]\!) * \ldots * (\![M_k]\!) \end{aligned}$$

Moreover, the encodings of linear contexts turn out to be *unary discrete bigraphs* G: bigraphs with open links and type $\langle 1, X\rangle \to \langle 1, Y\rangle$. Again, the result in [101] implies that the normal form, up to permutations, for unary discrete bigraphs is:

$$G = (join_k \otimes id_Y) \circ (R \otimes M_1 \otimes \ldots \otimes M_{k-1})$$

where M_i are discrete ground molecules and R can be either id_1 or $(\mathsf{K}_{\vec{a}} \otimes id_Y) \circ Q$. Again, we can define the reverse encoding $(\![\ ]\!)$ of $[\![\ ]\!]$, from unary discrete bigraphs to linear contexts, involving such a normal form:

$$\begin{aligned} (\![id_1]\!) &\stackrel{\text{def}}{=} - \\ (\![(\mathsf{K}(a)_x \otimes id_Y) \circ Q]\!) &\stackrel{\text{def}}{=} a_x[(\![Q]\!)] \\ (\![(join_k \otimes id_Y) \circ (R \otimes M_1 \otimes \ldots \otimes M_{k-1})]\!) &\stackrel{\text{def}}{=} (\![R]\!) \mid (\![M_1]\!) \mid \ldots \mid (\![M_{k-1}]\!) \end{aligned}$$

Table 3.17 Encoding CTL in BiLog over Prime Discrete Ground Bigraphs

Trees into prime ground discrete bigraphs

$$\begin{aligned}
[\![0]\!] &\stackrel{\text{def}}{=} 1 \\
[\![a_x[T]]\!] &\stackrel{\text{def}}{=} (\mathsf{K}(a)_x \otimes id_{fn(T)}) \circ [\![T]\!] \\
[\![T_1 \mid T_2]\!] &\stackrel{\text{def}}{=} [\![T_1]\!] * [\![T_2]\!]
\end{aligned}$$

Contexts into unary discrete bigraphs

$$\begin{aligned}
[\![-]\!]_C &\stackrel{\text{def}}{=} id_1 \\
[\![a_x[C]]\!]_C &\stackrel{\text{def}}{=} (\mathsf{K}(a)_x \otimes id_{fn(C)}) \circ [\![C]\!]_C \\
[\![T \mid C]\!]_C &\stackrel{\text{def}}{=} [\![T]\!] * [\![C]\!]_C \\
[\![C \mid T]\!]_C &\stackrel{\text{def}}{=} [\![C]\!]_C * [\![T]\!]
\end{aligned}$$

TLformulae into PGL formulae

$$\begin{aligned}
[\![\mathit{false}]\!]_P &\stackrel{\text{def}}{=} \mathbf{F} \\
[\![\mathbf{0}]\!]_P &\stackrel{\text{def}}{=} \mathbf{1} \\
[\![K(P)]\!]_P &\stackrel{\text{def}}{=} [\![K]\!]_K \circ_{\langle 1,_\rangle} [\![P]\!]_P \\
[\![K \triangleleft P]\!]_P &\stackrel{\text{def}}{=} [\![K]\!]_K \leftarrow_{\langle 1,_\rangle} [\![P]\!]_P \\
[\![P \Rightarrow P']\!]_P &\stackrel{\text{def}}{=} [\![P]\!]_P \Rightarrow [\![P']\!]_P
\end{aligned}$$

CTL formulae into PGL formulae

$$\begin{aligned}
[\![\mathit{false}]\!]_K &\stackrel{\text{def}}{=} \mathbf{F} \\
[\![-]\!]_K &\stackrel{\text{def}}{=} \mathbf{id}_1 \\
[\![P \triangleright P']\!]_K &\stackrel{\text{def}}{=} [\![P]\!]_P \rightarrow_{\langle 1,_\rangle} [\![P']\!]_P \\
[\![a_x[K]]\!]_K &\stackrel{\text{def}}{=} ((\mathsf{K}(a)_x) \otimes id_{\langle 0,_\rangle}) \circ [\![K]\!]_K \\
[\![P \mid K]\!]_K &\stackrel{\text{def}}{=} [\![P]\!]_P * [\![K]\!]_K \\
[\![K \Rightarrow K']\!]_K &\stackrel{\text{def}}{=} [\![K]\!]_K \Rightarrow [\![K']\!]_K
\end{aligned}$$

As the bigraphical model is specialised to context trees, so BiLog logic is specialised to the Context Tree Logic. The encodings of the connectives and the constants are in Tab. 3.17, and their soundness is shown in the next lemma.

Theorem 26 (Encoding Context Tree Logic). *For each tree T and formula P of CTL, $T \models_{\mathcal{T}} P$ if and only if $[\![T]\!] \models [\![P]\!]_P$. Also, for each context C and formula K of CTL, $C \models_{\mathcal{K}} K$ if and only if $[\![C]\!]_C \models [\![K]\!]_K$.*

Proof. Follow the lines of Theorem 24 and 25, by structural induction on CTL formulae and by exploiting the fact that the encoding of contexts trees into unary discrete bigraphs is bijective. □

The encoding shows that the models introduced in [37] are a particular kind of discrete bigraphs with one port for each node and a number of holes and roots limited to one. Hence, this shows how BiLog for discrete bigraphs is a generalisation of Context Tree Logic to contexts with several holes and regions. On the other hand, since STL is more general than separation logic, cf. [37], and it is used to characterise programs that manipulate tree structured memory model, BiLog can express separation logic as well.

3.6 BiLog for XML Data and Contexts

XML data are essentially tree-shaped resources. Starting from [38], where XML data were modelled by unordered labelled trees, much work on spatial logic for semistructured data and XML has been proposed [39, 40, 57]. A query language on semistructured data

based on Ambient Logic was studied in [41]. Here we add links on resource names to that tree-shaped model, so as to obtain a more general framework for semistructured data and XML. A similar step was undertaken in [46]. As bigraphs naturally model XML contexts, here we improve on [46] by showing that BiLog is suitable to describe XML contexts, which can be interpreted as web services or XML transformations.

Here we focus on the applications of BiLog to XML data. In particular, we first show how XML data, contexts, and a class of web services can be interpreted as a bigraph. Then, equipped with a 'bigraphical' representation of XML data and contexts, we show how BiLog can describe and reason about XML.

3.6.1 Modelling XML Contexts as Bigraphs

The importance of the underlying hierarchical structure in XML, as well as the fact that links are used only to model relations between nodes, suggests bigraphs as good models for XML documents. Ground bigraphs represent XML documents, while those with holes represent XML contexts. The interpretation is trivial when nominal constraints (such as `ID` and `IDREF` attributes and namespaces) are not considered. Without nominal attributes there is in fact no link between nodes, and XML tree structures can be mapped to place graphs by associating tags and values to bigraphical controls with arity zero. This yields an ambient-like formalism [38].

To model nominal resources and links, controls must be enriched by identification and pointer ports, connected to each other by the link graph. The model so obtained is similar to the one in [46], where *trees with dangling pointers* are considered. In addition, link graphs model local names, and so also unnamed connections.

As seen in §3.5.5, the main constituents of a bigraph are the discrete ions $\mathsf{K}_{\vec{a}}$, whose ports are linked to the names in $\vec{a}$. In XML settings, a ion represents a *tag* with some *attributes*. Since ports are unambiguously identified, they can be associated to attributes. The first port of a ion is associated to a (unique) name, which identifies, as an `ID` attribute, the element represented by the ion. Other ports are linked either to other nodes' `ID`s, so acting effectively as `IDREF`s, or to internal edges connected to internal nodes, so representing general attributes. Example 9 will clarify the idea. Embedding a ion into the hole of another ion, represents the inclusion of the corresponding elements.

XML data are encoded as ground bigraphs as outlined in Tab. 3.18. Without attributes, XML data are completely modelled by the place graph, since the arity is zero for every bigraphical control. When dealing with attributes, names and edges represent XML attributes and XML links between elements, respectively. We consider the `ID`s used in XML data as names and we assume two functions for values:

$K_{val}(v)$ maps the value v to a single node with no outer names, no nodes and no holes inside, and it is actually used to encode the value v by bigraphs.

$K_{val}(v)_a$ maps the value v a single node with outer name a, no nodes and no holes inside, and it is auxiliary to encode values linked to attributes.

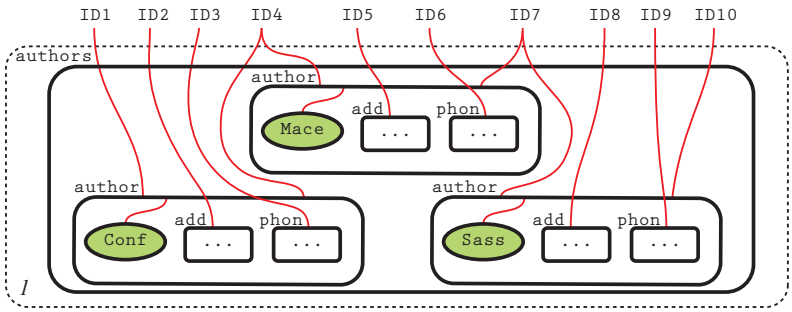
Table 3.18 XML Documents as Ground Bigraphs

$(\!\lvert v \rvert\!)$	$\stackrel{\text{def}}{=}$	$K_{val}(v)$	value
$(\!\lvert v \rvert\!)_a$	$\stackrel{\text{def}}{=}$	$K_{val}(v)_a$	value linked to an attribute name a
$(\!\lvert \vec{v} \rvert\!)_{\vec{b}}$	$\stackrel{\text{def}}{=}$	$(\!\lvert v_1 \rvert\!)_{b_1} \otimes \ldots \otimes (\!\lvert v_n \rvert\!)_{b_n}$	with $\vec{v} = v_1 \ldots v_n$ and $\vec{b} = b_1 \ldots b_n$
$(\!\lvert \emptyset \rvert\!)$	$\stackrel{\text{def}}{=}$	1	empty tree
$(\!\lvert T \rvert\!)$	$\stackrel{\text{def}}{=}$	$/\vec{a} \circ \sigma \circ K_{tag}(t)_{u,\vec{u},\vec{b}} \circ join_{n+k}((\!\lvert \vec{v} \rvert\!)_{\vec{b}} \otimes \alpha_1 \circ (\!\lvert T_1 \rvert\!) \otimes \ldots \otimes \alpha_n \circ (\!\lvert T_n \rvert\!))$	
where $T = \langle t,\, \texttt{ID} = u,\, \vec{a} = \vec{u},\, \vec{b} = \vec{v} \rangle\, T_1, ..., T_n\, \langle /t \rangle$ is an XML tree			
with	$\vec{a}$	$= a_1 \ldots a_k$	link attributes
	$\vec{u}$	$= u_1 \ldots u_k$	names
	$\vec{b}$	$= b_1 \ldots b_p$	value attributes
	$\vec{v}$	$= v_1 \ldots v_k$	values
	α_i		renaming the names of T_i into fresh names
	σ	$\stackrel{\text{def}}{=} \alpha_1^{-1} \cup \ldots \cup \alpha_n^{-1}$	inverse renaming
	$/\vec{a}$	$\stackrel{\text{def}}{=} /a_1 \otimes \ldots \otimes /a_p$	closure of the names in $\vec{a}$
	$join_{n+k}$		merging among $n+k$ bigraphs (definable from *join*)

We assume a class $\mathcal{K}_{tag}$ of controls. Let t be an XML tag, and *Att* the list of attributes for t. Being finite and ordered, the list *Att* can be associated to an ordinal $\#Att$. In particular, every attribute can be identified by the position. So the tag t is associated to $K_{tag}(t,)_{\vec{u}}$, which represents the ion with control $K_{tag}(t) \in \mathcal{K}_{tag}$ and arity $\#Att$. The vector $\vec{u}$ indicates the names connected to the control. These names correspond to the `ID`s associated to the attributes in *Att*. A value attribute is encoded as a value inside the node and connected to the port whose position marks the corresponding attribute. Identifiers (`ID`) and links (`IDREF`) attributes become *names* of the tag and can be connected with other names to model references. The connection is performed by link graph constructors: $a \leftleftarrows b$, to create a reference, and $/a$, to create a closed connection for attributes.

In Tab. 3.18 the term 1 corresponds to the empty tree. The core of the translation is the encoding of (non empty) trees. Here, the role of *join* is to group together the (encodings of the) set of children of T and the (encodings of the) values linked to attributes. The renamings α_i guarantee that the product is defined and they are obtained by choosing fresh names, not appearing in the encoded tree, and by combining operators $a \leftarrow b$. The bigraph obtained by *join* is single-rooted, thus it fits in the ion associated to the tag t. After the composition with the ion, names are renamed in order to actualise all the references, finally the links between the root and the values linked to attributes are closed. The renaming is obtained by considering the inverse of α_i (definable by using the operators $a \leftarrow b$ and $a \leftleftarrows b$), and the closure is obtained by combining the closures of the names

Figure 3.4 XML Encoding



associated to attributes.

Example 9. Consider a database that stores scientific papers and information about their authors, and focus on the fragment quoted in the document below.

```
<authors>
 <author n="ID1" name="Conf" coauth="ID4">
   <add n="ID2">"..."</add>   <phon n="ID3">"..."</phon>
 </author>
 <author n="ID4" name="Mace" coauth="ID7">
   <add n="ID5">"..."</add>   <phon n="ID6">"..."</phon>
 </author>
 <author n="ID7" name="Sass" coauth="ID10">
   <add n="ID8">"..."</add>   <phon n="ID9">"..."</phon>
 </author>
</authors>
```

Tag `author` has the following attributes: an identifier `IDn`, a link to another author `coauth`, that is an `IDREF`, and a general attribute `name`. In the corresponding bigraphical encoding (see Fig. 3.4) every tag `author` is associated to a control of arity three. Exploiting the order of the ports, we identify a port with the corresponding XML attribute unambiguously. In the picture we assume the ports ordered clockwise. The first port corresponds to the identifier, `ID`, and is connected to an outer name. The second one corresponds to the general attribute `name`, and is connected by a closed link to a value. The final attribute corresponds to the reference, `coauth`, and it is connected to a name that corresponds to another `author` tag.

This encoding does not capture the order among children of a node, so they represent lists of unordered trees connected through links. This model can be used for XML data whose document order is not relevant, as, for instance, for XML encodings of relational databases [2], or for distributed XML documents in a P2P computing, or *semantics web* where attaching meaning to denote order is undesirable. Sorting disciplines may provide an encoding that respects the order.

More generally, a bigraph represents a context for unordered XML data, just because there can be holes in it. So in Ex. 9 we can imagine holes in place of some nodes. This yields a contextual XML document, representing a function, or *web service*, that takes a list of XML files and returns their composition in the context, by fitting every file in the relative position. In this way, besides plain XML documents, we can model web services.

3.6.2 BiLog for XML Contexts

This section informally discusses how BiLog can be used for describing, querying and reasoning about XML. We analyse three possible cases: *(i)* PGL to model XML data trees and tree contexts, without nominal resources; *(ii)* logics for *discrete bigraphs* to model XML data trees with identified nodes; *(iii)* BiLog to model XML data trees with soft-link connections, that are implemented with nominal resources.

XML without `IDs` As said in §3.6.1, without nominal resources XML amounts to unordered labelled tree. In [38] the author outlines the similarities between such a model and ambient calculus. Then Ambient Logic is used in [41] to introduce a query language for semistructured. In §3.5.2 and § 3.5.7 we show that PGL extends the static fragment of ambient logic and models general contexts of tree-shaped resources. Hence it can describe XML contexts, without attributes.

The models of PGL are *positive* functions $m \to n$, which produce a list of n XML contexts from a list of m XML contexts. The adjective 'positive' means that the functions can only *add* structure to the parameters, without removing or replace any part of XML data. In this sense, XML contexts are viewed as positive XML web services that take XML documents and return XML documents. This is similar to Positive Active XML [1], but presents a remarkable difference, as the bigraphical model does not handle ordered trees. We use a *list* of parameters and a *list* of resulting contexts. For instance, consider a web service *wb* that satisfies the formula $K_1(id_1) \mid K_2(id_2)$. This web service takes two trees and puts the first inside a node labelled by K_1, then it puts the second inside a node labelled by K_2, and finally it performs a parallel composition between the two resulting trees. The ordered parameters are required to fix the exact correspondence between holes and roots. The web service *wb* is characterised by the formula above, but it satisfies also the formula $\mathsf{K}_1(id_1) \mid \mathbf{T}$. The formula characterises web services which have at least one hole and are the composition of a node with arity one labelled by K_1 in parallel with something else. In this sense a notion of *type* for web services arises: we can use PGL to formalise web service types and constraints.

Since XML active documents are contexts, PGL actually describes active XML documents and web service in an unique framework. In addition, an approach similar to TQL [41] can be used to query Active XML documents and web service. PGL may be eventually used to type web service in order to avoid useless invocations.

XML Contexts with identified nodes A simple tree structure does not allow logic and model to directly identify the resources, which are accessed only through navigation. When XML documents have nominal resources in addition to the tree structure, names can refer to locations, hence the resulting model can be seen as an extension of a heap memory model. In particular, names are intrinsically separated by the tensor product. Trees with names correspond to discrete bigraphs, namely place graphs with named resources but no name sharing between different resources. PGL extended by named controls K_x and renamings $x \leftarrow y$ is suitable to describe these models. In detail, K_x denotes a node labelled by K, with name identifier x, and an hole inside. The rename $x \leftarrow y$ is suitable to map names of different sources to different identifiers. The tensor product constraints two models to be separated both in locality and in names. In fact, a models satisfies $A \otimes B$ if it has two sub-models satisfying A and B respectively and with disjoint sets of identifiers, i.e., disjoint outer faces. Such a PGL extension characterises (contexts of) resources which can be accessed either by navigation through the tree structure or by using name controls as pointers.

XML Contexts with Connections For XML data models, nodes which are not related by a parent-child relationship can be connected either explicitly by `ID` and `IDREF` attributes or implicitly by namespaces. BiLog's notion of sharing can model connections between resources to treat structures with pointers. Sharing is obtained through links between names of resources. In Tab. 3.18, identifiers are encoded as tag names and `IDREF`s as pointers to names in the same document. The connection between `ID` and `IDREF` is expressed in BiLog by closed names. Moreover the 'separation-up-to' operator, defined in (3.4), can express properties like "The author of paper X has a relationship with the author of paper Y," which express separation on resources, since there are different authors for different papers, but sharing on linked names. BiLog can also express XML contexts with links. For instance a alteration to a namespace can be represented by a link composed to an identity, and unnamed resources can be represented by closed names.

3.7 Towards Dynamics

A main feature of a distributed system is mobility, or dynamics in general. In dealing with communicating and nomadic processes, the interest is to describe not only their internal structure, but also their behaviour. So far, it has been shown how BiLog can describe structures, this section is intended to study how to express evolving systems. BiLog is able to deal with the dynamic behaviour of models. Essentially, this is due to its the contextual nature, suitable to characterise structural parametric reaction rules that model dynamics.

The usual way to express dynamics with a logic is to introduce a *next step* modality ($\Diamond$), that hints how the system develops in the future. In general, a process satisfies the formula $\Diamond A$ if it may evolve into a process satisfying A.

In process algebras, dynamics is often presented by *reaction* (or rewriting) rules of the form $r \longrightarrow r'$, meaning that the term r (the *redex*) is replaced by r' (the *reactum*) in *suitable* contexts, named *active*. The 'activeness' is defined on the structure of contexts by a predicate δ.

In general, a *bigraphical reactive system* is a bigraphical system provided with a set of parametric reaction rules, namely a set S of pairs[1] $(R, R' : I \to J)$, where R and R' are the redex and the reactum of a parametric reaction. We consider only ground bigraphs, as they identifies processes, contrary to non-ground bigraphs that are open and identifies contexts. The active bigraphs are identified by the predicate δ, closed for compositions and *id*s. A ground bigraph g reacts to g' (written $g \longrightarrow g'$) if there is a couple $(R, R') \in S$, a set of names Y, a bigraph D (usually not ground) with $\delta(D)$ true, and a ground bigraph d, such that:

$$g \equiv D \circ (R \otimes id_Y) \circ d \quad \text{and} \quad g' \equiv D \circ (R' \otimes id_Y) \circ d.$$

When the model is enriched with a dynamical framework, the usual way to introduce the modality $\Diamond$ is to extend the relation $\models$ by defining '$g \models \Diamond A$ *iff* $g \longrightarrow g'$ *and* $g' \models A$.' According to the formulation of the reduction given above, we obtain

$$g \models \Diamond A \quad \textit{iff} \quad \textit{there exist} (R, R') \in S,\ id_Y,\ D \textit{ active, and } d \textit{ ground such that } g \equiv D \circ (R \otimes id_Y) \circ d \textit{ and } D \circ (R' \otimes id_Y) \circ d \models A. \quad (3.5)$$

One may wonder whether the modality $\Diamond$ is the only way to express a temporal evolution in BiLog. It turns out that BiLog has a built in notion of dynamics. There are several cases in which BiLog itself is sufficient to express the computation. One of them is the encoding of CCS, shown in the following.

We focus on the fairly small fragment of CCS considered in [35], consisting of prefix and parallel composition only; P, Q will range over CCS *processes*; a, b, c over *actions*, chosen in the enumerable set *Acts*; and $\overline{a}, \overline{b}, \overline{c}$ over *coactions*. Process syntax is defined by the following grammar:

$$\begin{array}{lcl} P & ::= & \mathbf{0} \mid \lambda.P \mid P \mid P \\ \lambda & ::= & a \mid \overline{a} \end{array}$$

As operator ν is not included, all the actions appearing in a process are not bound; this fact yields the encoding to produce bigraphs with open links. Moreover, as *Acts* will actually be the set of names for the bigraphs used to encode CCS processes, we will refer to its elements as names. In particular, the 'names' of a CCS process are all the elements of *Acts* appearing in its syntax, both as actions and as coactions. For instance, the names in the process $a.\overline{c}.b.\overline{a}.\mathbf{0}$ are a, b, c.

[1] This is a simplification to capture the case of CCS presented in this section. In general, bigraphical theory does not require R and R' to have the same inner face.

The *structural congruence* $\equiv$ is defined as the least congruence on processes such that $P \mid \mathbf{0} \equiv P$, $P \mid Q \equiv Q \mid P$ and $P \mid (Q \mid R) \equiv (P \mid Q) \mid R$. Finally, the usual *reduction operational semantics* gives dynamics:

$$\frac{}{a.P \mid \overline{a}.Q \to P \mid Q} \qquad \frac{P \to Q}{P \mid R \to Q \mid R} \qquad \frac{P \equiv P' \quad P' \to Q' \quad Q' \equiv Q}{P \to Q} \tag{3.6}$$

The work [103] presents a bigraphical encoding for this CCS. The bigraphs suitable to encode CCS are built by two controls with arity 1: act for actions and coact for coactions. As mentioned above, every action $a \in Acts$ is treated as a name in the bigraphical model. The corresponding constructors assume the form act_a and coact_a. Reactions are intuitively expressed as

$$\mathsf{act}_a\Box_1 \mid \mathsf{coact}_a\Box_2 \longrightarrow a \mid \Box_1 \mid \Box_2. \tag{3.7}$$

Rules are parametric, in the sense that the two holes, $\Box_1$ and $\Box_2$, can be filled up by any process, and the link a is introduced to maintain the same interface between redex and reactum. By definition, redex can be replaced by the reactum in any bigraphical active context. As the active contexts are identified by the predicate δ, in this particular case such a predicate has to project CCS's active contexts into bigraphs. It is easy to see that rules in (3.6) imply that active CCS contexts have the form '$P \mid \Box$,' whose corresponding bigraphical context is '$[\![P]\!] \mid \Box$,' where $[\![P]\!]$ is the bigraphical encoding for P. Since Lemma 22 will prove that the encoding introduced in this section is bijective on bigraphs that are ground, *prime* (i.e., with a single root, as for the definition on place graphs) and with open links, the formal definition for an active bigraphical context is

$$g \mid \Box, \tag{3.8}$$

for $g : \epsilon \to \langle 1, Z \rangle$ ground, prime and with open links. Moreover, controls act and coact are declared to be *passive*, i.e., no reaction can occur inside them. It is straightforward to conclude that the most general context ready to react has the form '$\Box_0 \mid \mathsf{act}_a\Box_1 \mid \mathsf{coact}_a\Box_2$' and the most general reaction is

$$\Box_0 \mid \mathsf{act}_a\Box_1 \mid \mathsf{coact}_a\Box_2 \longrightarrow \Box_0 \mid a \mid \Box_1 \mid \Box_2, \tag{3.9}$$

where holes $\Box_0$, $\Box_1$ and $\Box_2$ has to be filled in by prime ground bigraphs with open links. Such a reduction turns out to be compositional with parallel operator.

The encoding maps CCS processes into ground, prime open linked bigraphs, and it is denoted by $[\![\ ]\!]_X$. Such an encoding is parameterised by a *finite* subset $X \subseteq Acts$; it yields ground bigraphs with outer face $\langle 1, X \rangle$ and open links. The value $[\![P]\!]_X$ is defined only if the names in P belong to X:

$$\begin{aligned}
[\![\mathbf{0}]\!]_X &\stackrel{\text{def}}{=} 1 \otimes X \\
[\![a.P]\!]_X &\stackrel{\text{def}}{=} (\mathsf{act}_a \overset{a}{\otimes} id_X) \circ [\![P]\!]_X \\
[\![\overline{a}.P]\!]_X &\stackrel{\text{def}}{=} (\mathsf{coact}_a \overset{a}{\otimes} id_X) \circ [\![P]\!]_X \\
[\![P \mid Q]\!]_X &\stackrel{\text{def}}{=} (join \otimes id_X) \circ ([\![P]\!]_X \overset{X}{\otimes} [\![Q]\!]_X)
\end{aligned}$$

where $a \in X$, and the sharing/separation operator $\overset{X}{\otimes}$ stands for $\overset{\vec{d}}{\otimes}$ where $\vec{d}$ is any array of all the elements in X.

Note, in particular, that the sharing tensor '$_- \overset{a}{\otimes} id_X$' enables the definition to be compositional, as the outer face is $\langle 1, X\rangle$ for every encoding. Moreover, such a sharing tensor allows the process filling the hole in act_a (and coact_a) to perform other a actions. In fact, consider the simple CCS process $a.\overline{a}.\mathbf{0}$, then $[\![\, a.\overline{a}.\mathbf{0} \,]\!]_{\{a\}}$ is $(\mathsf{act}_a \overset{a}{\otimes} id_{\{a\}}) \circ (\mathsf{coact}_a \overset{a}{\otimes} id_{\{a\}}) \circ (1 \otimes a)$. Clearly, the composition is granted by the sharing operator.

In the encoding for parallel, operator *join* makes tensor commutative. There is a straight correspondence between parallel operators in the two calculi, as $[\![\, P \mid Q \,]\!]_X$ corresponds to $[\![\, P \,]\!]_X \mid [\![\, Q \,]\!]_X$, that is the parallel operator on bigraphs, defined in [101]. The result stated in Lemma 22 says that the encoding is bijective on prime ground bigraphs with open links. First, Lemma 21 provides a general result on bigraphs and parallel composition. It says that to add names that already appear in a bigraph dos not alter the bigraph itself.

Lemma 21 (Adding Names). *If x is in the outer names of G, then $G \mid x \equiv G$.*

Proof. Express the parallel in terms of renamings, linkings and tensor product as in [101], and use axioms of [101]. Assume $G : \langle m, X\rangle \to \langle n, \{x\} \cup Y\rangle$, with $y \notin \{x\} \cup Y$. Then $G \mid x$ corresponds to $(id_{\langle n,Y\rangle} \otimes (x \leftleftarrows y)) \circ (G \otimes ((y \leftarrow x) \circ x))$, that is $(id_{\langle n,Y\rangle} \otimes (x \leftleftarrows y)) \circ (G \otimes y)$ by the third link axiom. By bifunctoriality property, this is congruent to $(id_{\langle n,Y\rangle} \otimes (x \leftleftarrows y)) \circ (id_{\langle n,Y\rangle} \otimes id_x \otimes y) \circ (G \otimes id_\epsilon)$, and again to $((id_{\langle n,Y\rangle} \circ id_{\langle n,Y\rangle}) \otimes ((x \leftleftarrows y) \circ (id_x \otimes y))) \circ G$. The latter is congruent to $(id_{\langle n,Y\rangle} \otimes id_x) \circ G$, by the second link axiom. Since $(id_{\langle n,Y\rangle} \otimes id_x) \circ G \equiv G$, conclude the thesis. □

Lemma 21 is useful to prove that the encoding is bijective on ground prime bigraphs with open links.

Lemma 22 (Bijective Translation). *For every finite subset $X \subseteq Acts$:*

1. *The translation $[\![\cdot]\!]_X$ is surjective on prime ground bigraphs with outerface $\langle 1, X\rangle$ and open links.*

2. *For every couple of processes P, Q and for every finite subset $X \subseteq Acts$ containing all the names in P and Q, it holds: $P \equiv Q$ iff $[\![P]\!]_X \equiv [\![Q]\!]_X$.*

Proof. Prove point (1) by showing that every prime ground bigraph with outerface $\langle 1, X\rangle$ has at least one pre-image for the translation $[\![\cdot]\!]_X$. Proceed by induction on the number of nodes in bigraphs. The Connected Normal Form (CNF) for bigraphs presented in [101] simplifies the proof. According to [101], every prime ground bigraph G with outerface $\langle 1, X\rangle$ and open links has the following connected normal form: $G ::= X \mid F$, where $F ::= M_1 \mid \ldots \mid M_k$, with $M ::= (\mathsf{K}_a \mid id_Y) \circ F$ for $a \in Acts$ and $\mathsf{K}_a \in \{\mathsf{act}_a, \mathsf{coact}_a\}$. In particular, a term M is a *ground molecule*.

The base of induction is X, intended as a bigraph, and clearly $[\![\mathbf{0}]\!]_X = X$. For the inductive step, consider a bigraph G with at least one node. This means $G = X \mid ((\mathsf{K}_a \mid$

$id_Y) \circ F) \mid G'$. Without losing generality, assume $\mathsf{K}_a = \mathsf{act}_a$, so $G = ((\mathsf{act}_a \mid id_X) \circ (X \mid F)) \mid (X \mid G')$ by Lemma 21. Now, the induction says that there exist P and Q such that $[\![P]\!]_X = X \mid F$ and $[\![Q]\!]_X = X \mid G'$, hence conclude $[\![a.P \mid Q]\!]_X = G$.

The forward implication of point (2) is proved by showing that the translation is sound with respect to the rules of congruence in CCS. This has been already proved in [101], where the parallel operator between bigraphs is shown to be commutative and associative, and to have 1 as a unit. Moreover, by Lemma 21, the bigraph $1 \otimes X$ is the unit for the parallel operator on prime ground bigraphs with outerface $\langle 1, X \rangle$.

The following claim, stated in [103], is the crucial step in proving the reverse implication of point (2). Its proof considers the connected normal form for bigraphs.

Claim. If G_i $(i = 1 \ldots m)$ and F_j $(j = 1 \ldots n)$ are ground molecules and $G_1 \mid \ldots \mid G_m \equiv F_1 \mid \ldots \mid F_n$, then $m = n$ and $G_i \equiv F_{\pi(i)}$ for some permutation π on m.

The proof of the reverse implication of point (2) proceeds by induction on the structure of the CCS process P. The base of induction is $P = \mathbf{0}$, in this case the statement is verified since $[\![Q]\!]_X \equiv [\![\mathbf{0}]\!]_X = X$ implies $Q \equiv \mathbf{0} \mid \ldots \mid \mathbf{0}$. For the inductive step, let $P \equiv a_1.P_1 \mid \ldots \mid a_m.P_m$ for any $m \geq 1$, and assume $[\![Q]\!] \equiv [\![P]\!]$. Furthermore we have $Q \equiv b_1.Q_1 \mid \ldots \mid b_n.Q_n$, then

$$
\begin{aligned}
[\![P]\!]_X = \quad & (\mathsf{act}_{a_1} \overset{a_1}{\otimes} id_X) \circ [\![P_1]\!]_X \mid \ldots \mid (\mathsf{act}_{a_m} \overset{a_m}{\otimes} id_X) \circ [\![P_m]\!]_X \\
[\![Q]\!]_X = \quad & (\mathsf{act}_{b_1} \overset{b_1}{\otimes} id_X) \circ [\![Q_1]\!]_X \mid \ldots \mid (\mathsf{act}_{b_m} \overset{b_m}{\otimes} id_X) \circ [\![Q_m]\!]_X
\end{aligned}
$$

Since the two translations are both a parallel compositions of ground molecules, the previous claim says that $m = n$, and there exists a permutation π on m such that $a_i \equiv a_{\pi(i)}$ and $[\![Q_i]\!] \equiv [\![P_{\pi(i)}]\!]$. By induction $Q_i \equiv P_{\pi(i)}$, hence $Q \equiv P$. □

Paper [103] proves that the translation preserves and reflects the reactions, namely: $P \longrightarrow P'$ *if and only if* $[\![P]\!]_X \longrightarrow [\![P']\!]_X$. A similar result is obtained in this case.

In the current bigraphical system, reaction rules are defined as $(\mathsf{act}_a \mid id_{Y_1}) \mid (\mathsf{coact}_a \mid id_{Y_2}) \longrightarrow a \mid id_{\langle 1, Y_1 \rangle} \mid id_{\langle 1, Y_2 \rangle}$. It is easy to see that this can be mildly sugared to obtain the rule introduced in (3.7). Moreover, the active contexts introduced in (3.8) can be specialised as $g \mid (id_1 \otimes id_Y)$, for $g : \epsilon \to \langle 1, Z \rangle$ ground, prime and with open links. Moreover, Y, Y_1 and Y_2 must be finite sets of names, viz., the outer names of the term that can fill the contexts. Finally, the general reaction (3.9) is specialised as

$$(id_1 \otimes id_Y) \mid (\mathsf{act}_a \mid id_{Y_1}) \mid (\mathsf{coact}_a \mid id_{Y_2}) \longrightarrow (id_1 \otimes id_Y) \mid a \mid id_{Y_1} \mid id_{Y_2}. \tag{3.10}$$

When a reacting (ground) bigraph is a CCS encoding, such as $[\![P]\!]_X$, it can actually be decomposed into a redex, essentially the one in the left-hand side of (3.10), and a ground bigraph with a well defined structure, essentially with three regions. The composition of such a bigraph with the corresponding reactum, essentially the one in the right-hand side of (3.10), gives the result of the reaction. Lemma23 expresses such a characterisation. Redex and Reactum are formally outlined in Tab. 3.19. They complex structure is due to the fact that tensor product is defined only disjoint names, and this is guaranteed by

Table 3.19 Reacting Contexts for CCS Encodings

Bigraphs:

$$Redex_a^{y_1,y_2,Y_1,Y_2} \stackrel{\text{def}}{=} W \circ (id_Y \otimes join) \circ (id_Y \otimes join \otimes id_1) \circ \{((y_1 \leftarrow a) \otimes id_1) \circ \circ \mathsf{act}_a \otimes id_{Y_1} \otimes ((y_2 \leftarrow a) \otimes id_1) \circ \mathsf{coact}_a \otimes id_{Y_2} \otimes id_{\langle 1,X\rangle}\}$$

$$React_a^{Y_1,Y_2} \stackrel{\text{def}}{=} W' \circ (id_{Y'} \otimes join) \circ (id_{Y'} \otimes join \otimes id_1)$$

Wirings:

$$W \stackrel{\text{def}}{=} ((X \leftleftarrows Y_1) \otimes id_1) \circ (id_{Y_1} \otimes (X \leftleftarrows Y_2) \otimes id_1) \circ \circ (id_{Y_1} \otimes id_{Y_2} \otimes id_{X\setminus\{a\}} \otimes (a \leftleftarrows y_1) \otimes\otimes id_1) \circ \circ (id_{Y_1} \otimes id_{Y_2} \otimes id_{X\setminus\{a\}} \otimes id_{\{y_1\}} \otimes (a \leftleftarrows y_2) \otimes id_1)$$

$$W' \stackrel{\text{def}}{=} ((X \leftleftarrows Y_1) \otimes id_1) \circ (id_{Y_1} \otimes (X \leftleftarrows Y_2) \otimes id_1)$$

Supporting Sets:

$$Y \stackrel{\text{def}}{=} \{y_1, y_2\} \cup Y_1 \cup Y_2 \cup X$$

$$Y' \stackrel{\text{def}}{=} Y_1 \cup Y_2 \cup X$$

renamings. To better understand the table, it is worth to reintroduce some syntactic sugar, as in (3.9). According to such a notation, $Redex_a^{y_1,y_2,Y_1,Y_2}$ and $React_a^{Y_1,Y_2}$ are simply $\Box_0 \mid \mathsf{act}_a\Box_1 \mid \mathsf{coact}_a\Box_2$ and $\Box_0 \mid \Box_1 \mid \Box_2$, where the sets of names X, Y_1, Y_2 are respectively associated to the holes $\Box_0, \Box_1, \Box_2$ and they must be disjoint to allow the tensor product. Names y_1 and y_2 are useful to join the action with the corresponding coaction, they must be disjoint with X, Y_1 and Y_2. Wirings W, W' and *join* operators assure that the outerfaces are $\langle 1, X\rangle$.

Lemma 23 (Reducibility). *For every CCS process P, the following are equivalent.*

1. *The translation $[\![P]\!]_X$ can perform the reduction $[\![P]\!]_X \longrightarrow G$.*
2. *There exist bigraphs $G_1, G_2, G_3 : \epsilon \to \langle 1, X\rangle$ and name $a \in X$, such that $[\![P]\!]_X \equiv ((\mathsf{act}_a \mid id_X) \circ G_1) \mid ((\mathsf{coact}_a \mid id_X) \circ G_2) \mid G_3$ and $G \equiv G_1 \mid G_2 \mid G_3$.*
3. *There exist actions $a \in X$ and $y_1, y_2 \notin X$, and two mutually disjoint subsets $Y_1, Y_2 \subseteq$ Acts with the same cardinality as X, but disjoint with X, y_1, y_2, and there exist the bigraphs $H_1 : \epsilon \to \langle 1, Y_1\rangle$, $H_2 : \epsilon \to \langle 1, Y_2\rangle$, and $H_3 : \epsilon \to \langle 1, X\rangle$ with open links, such that $[\![P]\!]_X \equiv Redex_a^{y_1,y_2,Y_1,Y_2} \circ (H_1 \otimes H_2 \otimes H_3)$ and $G \equiv React_a^{Y_1,Y_2} \circ (H_1 \otimes H_2 \otimes H_3)$, where $Redex_a^{y_1,y_2,Y_1,Y_2}$, $React_a^{Y_1,Y_2}$ are defined in Tab. 3.19.*

Proof. First prove that points (1) and (2) are equivalent. Assume that the bigraph $[\![P]\!]_X$ can perform a reaction. This means that $[\![P]\!]_X \equiv ((\mathsf{act}_a \mid id_{Y_1}) \circ G'_1) \mid ((\mathsf{coact}_a \mid id_{Y_2}) \circ G'_2) \mid G'_3$ and that $G \equiv a \mid G'_1 \mid G'_2 \mid G'_3$ for some suitable ground bigraphs G'_1, G'_2 and G'_3 and action $a \in X$. Since the type of both $[\![P]\!]_X$ and G is $\epsilon \to \langle 1, X\rangle$, Lemma 21 says that $G \equiv (X \mid G'_1) \mid (X \mid G'_2) \mid (X \mid G'_3)$ and $[\![P]\!]_X \equiv ((\mathsf{act}_a \mid id_X) \circ (X \mid G'_1)) \mid ((\mathsf{coact}_a \mid$

$id_X) \circ (X \mid G_2')) \mid (X \mid G_3')$. Then define G_i to be $X \mid G_i'$ for $i = 1, 2, 3$, and conclude that $G \equiv G_1 \mid G_2 \mid G_3$ and $[\![P]\!]_X \equiv ((\mathsf{act}_a \mid id_X) \circ G_1) \mid ((\mathsf{coact}_a \mid id_X) \circ G_2) \mid G_3$.

Then prove that point (2) implies point (3). Assume that $[\![P]\!]_X \equiv ((\mathsf{act}_a \mid id_X) \circ G_1) \mid ((\mathsf{coact}_a \mid id_X) \circ G_2) \mid G_3$ and $G \equiv G_1 \mid G_2 \mid G_3$, with $G_1, G_2, G_3 : \epsilon \to \langle 1, X \rangle$. Chose two actions $y_1, y_2 \notin X$ and two mutually disjoint subsets $Y_1, Y_2 \subseteq Acts$ with the same cardinality as X, but disjoint with X, y_1, y_2, and follow the definition of parallel operator in [101] to obtain

$$[\![P]\!]_X \equiv W \circ (id_Y \otimes join) \circ (id_Y \otimes join \otimes id_1) \circ \{((y_1 \leftarrow a) \otimes \\ \otimes id_{\langle 1, Y_1 \rangle}) \circ (\mathsf{act}_a \otimes id_{Y_1}) \circ ((Y_1 \leftarrow X) \otimes id_{\langle 1, Y_2 \rangle}) \circ G_1 \otimes ((y_2 \leftarrow a) \otimes \\ \otimes id_1) \circ (\mathsf{coact}_a \otimes id_{Y_2}) \circ ((Y_2 \leftarrow X) \otimes id_1) \circ G_2 \otimes G_3\}$$

and

$$G \equiv W' \circ (id_{Y'} \otimes join) \circ (id_{Y'} \otimes join \otimes id_1) \circ \\ \circ \{((Y_1 \leftarrow X) \otimes id_{\langle 1, Y_2 \rangle}) \circ G_1 \otimes ((Y_2 \leftarrow X) \otimes id_1) \circ G_2 \otimes G_3\}$$

where $Y = \{y_1\} \cup Y_1 \cup \{y_2\} \cup Y_2 \cup X$ and $Y' = Y_1 \cup Y_2 \cup X$. The bigraphs W and W' are defined in Tab. 3.19, they both link the subsets Y_1 and Y_2 with X, and moreover W links y_1 and y_2 with a. By bifunctoriality property, $[\![P]\!]_X$ is rewritten as

$$W \circ (id_Y \otimes join) \circ (id_Y \otimes join \otimes id_1) \circ \{((y_1 \leftarrow a) \otimes id_1) \circ \\ \circ \mathsf{act}_a \otimes id_{Y_1} \otimes ((y_2 \leftarrow a) \otimes id_1) \circ \mathsf{coact}_a \otimes id_{Y_2} \otimes G_3 \} \circ \\ \circ \{ ((Y_1 \leftarrow X) \otimes id_1) \circ G_1 \otimes ((Y_2 \leftarrow X) \otimes id_1) \circ G_2 \},$$

and, again by bifunctoriality property, as

$$W \circ (id_Y \otimes join) \circ (id_Y \otimes join \otimes id_1) \circ \{((y_1 \leftarrow a) \otimes id_1) \circ \\ \circ \mathsf{act}_a \otimes id_{Y_1} \otimes ((y_2 \leftarrow a) \otimes id_1) \circ \mathsf{coact}_a \otimes id_{Y_2} \otimes id_{\langle 1, X \rangle} \} \circ \\ \circ \{ ((Y_1 \leftarrow X) \otimes id_1) \circ G_1 \otimes ((Y_2 \leftarrow X) \otimes id_1) \circ G_2 \otimes G_3 \}.$$

Point (3) follows by defining $H_i' = ((Y_i \leftarrow X) \otimes id_1) \circ G_i$ for $i = 1, 2$, and $H_3 = G_3$. Note that the three bigraphs G_i and H_i have open links as so does $[\![P]\!]_X$. Finally, point (3) implies point (2), by inverting previous reasoning. □

By following the ideas of [103] it is easy to demonstrate that there is an exact match between the reactions generated in CCS and in the bigraphical system. This a consequence of the fact that CCS reacting contexts are clearly identified and easily transferred in bigraphical settings.

Proposition 28 (Matching Reactions). *For every finite set X, that contains all the names appearing in P and Q, it holds: $P \to Q$ if and only if $[\![P]\!]_X \longrightarrow [\![Q]\!]_X$.*

Proof. For the forward direction, proceed by induction on the number of the rules applied in the derivation for $P \to Q$ in CCS. The base of the induction is the only rule without premisses, meaning that P is $a.P_1 \mid \overline{a}.P_2$ and Q is $P_1 \mid P_2$. The translation is sound as regards this rule, since the reactive system says

$$((\mathsf{act}_a \mid id_X) \circ [\![P_1]\!]_X) \mid ((\mathsf{coact}_a \mid id_X) \circ [\![P_2]\!]_X) \longrightarrow X \mid [\![P_1]\!]_X \mid [\![P_2]\!]_X.$$

The induction step considers two cases. First, assume that $P \to Q$ is derived from $P' \to Q'$, where P is $P' \mid R$ and Q is $Q' \mid R$. Then the induction hypothesis says that $[\![P']\!]_X \longrightarrow [\![Q']\!]_X$, hence $[\![P']\!]_X \mid [\![R]\!]_X \longrightarrow [\![Q']\!]_X \mid [\![R]\!]_X$. Conclude that $[\![P]\!]_X \longrightarrow [\![Q]\!]_X$, as $[\![P]\!]_X$ is $[\![P']\!]_X \mid [\![R]\!]_X$ and $[\![Q]\!]_X$ is $[\![Q']\!]_X \mid [\![R]\!]_X$. Second, assume that $P \to Q$ is derived from the congruences $P \equiv P'$ and $Q' \equiv Q$, and from the transition $P' \to Q'$. By Lemma 22, $[\![P]\!]_X \equiv [\![P']\!]_X$ and $[\![Q']\!]_X \equiv [\![Q]\!]_X$, and, by induction hypothesis, $[\![P']\!]_X \longrightarrow [\![Q']\!]_X$. Conclude $[\![P]\!]_X \longrightarrow [\![Q]\!]_X$, since the reduction is defined up to congruence.

For the reverse implication, assume $[\![P]\!]_X \longrightarrow [\![Q]\!]_X$. Lemma 23 says that there exist the bigraphs $G_1, G_2, G_3 : \epsilon \to \langle 1, X \rangle$ and the name $a \in X$ such that $[\![P]\!]_X \equiv ((\mathsf{act}_a \mid id_X) \circ G_1) \mid ((\mathsf{coact}_a \mid id_X) \circ G_1) \mid G_3$ and $G \equiv G_1 \otimes G_2 \otimes G_3$. Now, Lemma 22 says that for every $i = 1, 2, 3$ there exists a CCS process P_i such that $[\![P_i]\!]$ corresponds to G_i, hence $[\![P]\!] \equiv [\![a.P_1 \mid \overline{a}.P_2 \mid P_3]\!]$ and $[\![Q]\!] \equiv [\![P_1 \mid P_2 \mid P_3]\!]$. Again, Lemma 22 says that $P \equiv a.P_1 \mid \overline{a}.P_2 \mid P_3$ and $Q \equiv P_1 \mid P_2 \mid P_3$, then $P \to Q$. □

Tanks to Lemma 22, the previous result can be further specialised: whenever a bigraphical encoding reacts, so does the corresponding CCS process.

Proposition 29 (Conservative Reaction). *If $[\![P]\!]_X \longrightarrow G$ for a CCS process P, then there exists a CCS process Q such that $[\![Q]\!]_X = G$ and $P \to Q$.*

Proof. Assume $[\![P]\!]_X \longrightarrow G$, then point (2) of Lemma 23 says that G has type $\epsilon \to \langle 1, X \rangle$ and open links, as so does $[\![P]\!]_X$. Lemma 22 says that there exists a process Q such that $[\![Q]\!]_X \equiv G$. Conclude $P \to Q$ by Lemma 28. □

Paper [35] introduces $\mathcal{L}_{spat}$, a spatial logic suitable to describe structure and behaviour of CCS processes. The formulae of such a logic are generated by $A, B ::= 0 \mid A \wedge B \mid A|B \mid \neg A \mid A \triangleright B \mid \Diamond A$. It includes the void constant 0 and the basic spatial operators: composition $|$, and its adjunct $\triangleright$. It presents also a temporal operator, next step modality $\Diamond$, to capture process dynamics. Table. 3.20 outlines the semantics of $\mathcal{L}_{spat}$ in term of CCS processes, as defines in [35]. In particular, parallel connective describes processes that are the parallel composition between two processes that satisfies the corresponding formulae. A process satisfies $A \triangleleft B$ if it satisfies the formula B whenever put in parallel with any process satisfying A. Finally, next step $\Diamond A$ is satisfied by a process that can evolve into a process satisfying A.

The logic $\mathcal{L}_{spat}$ can be encoded in a suitable instantiation of BiLog, without using the modality defined in (3.5), but exploiting BiLog expressivity, suitable to characterise reacting contexts. It is sufficient to instantiate the logic BiLog$(\mathcal{M}, \otimes, \epsilon, \Theta, \equiv, \tau)$ to obtain

Table 3.20 Semantics of Formulae $\mathcal{L}_{spat}$ in CCS

P	$\models_{spat}$	0	if $P \equiv \mathbf{0}$
P	$\models_{spat}$	$\neg A$	if not $P \models_{spat} A$
P	$\models_{spat}$	$A \wedge B$	if $P \models_{spat} A$ and $P \models_{spat} B$
P	$\models_{spat}$	$A \mid B$	if there exist R, Q, s.t. $P \equiv R \mid Q$, $R \models_{spat} A$ and $Q \models B_{spat}$
P	$\models_{spat}$	$A \triangleright B$	if for every Q, $Q \models_{spat} A$ implies $P \mid Q \models_{spat} B$
P	$\models_{spat}$	$\Diamond A$	if there exist P' s.t. $P \longrightarrow P'$ and $P' \models_{spat} A$

the bigraphical encoding of CCS. We define Θ to be composed by the standard constructor for a bigraphical system with $\mathcal{K} = \{\mathsf{act}, \mathsf{coact}\}$. Moreover, transparency predicate τ must be always true. This fact is determinant for the soundness of the logical encoding, as it enables BiLog to fully describe any bigraphical term, and, therefore, to detect all reacting contexts by simply analysing their 'spatial' structure.

Lemma 23 is informally rephrased by saying that reactions for encoded CCS processes are determined by couples of the form $(Redex_a, Reactum_a)$, cf. Tab. 3.19, and every reacting process is characterised by

$$[\![P]\!]_X \longrightarrow [\![Q]\!]_X \textit{ iff there exists a bigraph } g \textit{ and } a \in X \textit{ such that } [\![P]\!]_X \equiv Redex_a \circ g \textit{ and } [\![Q]\!]_X \equiv Reactum_a \circ g.$$

Since τ is always true, it is possible to define a characteristic formula for every redex and reactum, simply by rewriting every bigraphical constructor and operator with the correspondent logical constant in their bigraphical encodings. For the new names y_1, y_2, and the new subsets Y_1, Y_2, denote with $\mathbf{Redex}_a^{y_1,y_2,Y_1,Y_2}$ and $\mathbf{React}_a^{Y_1,Y_2}$ the characteristic formulae for $Redex_a^{y_1,y_2,Y_1,Y_2}$ and $React_a^{Y_1,Y_2}$, respectively. Clearly, $G \models \mathbf{Redex}_a^{y_1,y_2,Y_1,Y_2}$ if and only if $G \equiv Redex_a^{y_1,y_2,Y_1,Y_2}$, and the same for reactum. This has a prominent role in defining the encoding of the temporal modality in BiLog.

Table 3.21 formally defines logical encoding, that is parameterised on the set X of names, as so does the process encoding. The encodings for logical connectives and spatial composition are self-explanatory. In particular, spatial composition requires the sharing of all the names in X: it corresponds to the logical parallel operator when the set of bigraph names is fixed and finite, as happens for processes encoded by $[\![\ ]\!]_X$. The encoding for $\triangleright$ introduces an auxiliary notation. Intuitively, formula $\mathbf{A}_X$ is defined to constrain a bigraph to be the encoding of a CCS process and to satisfy $[\![A]\!]_X$. In fact, $G \models \mathbf{A}_X$ means that G satisfies $[\![A]\!]_X$, it has type $\epsilon \to \langle 1, X \rangle$ and its links are open, as a bigraph satisfies **Open** only if no closure appears in any of its decompositions. Proposition 30 will show that a bigraph satisfies $[\![P]\!]_X \models [\![A \triangleright B]\!]_X$ if it satisfies $[\![B]\!]_X$ whenever connected in parallel with any encoding of a CCS process satisfying $[\![A]\!]_X$.

In the encoding for the temporal modality $\Diamond$, the supporting formula **Triple** is satisfied by processes that are the composition of three single-rooted ground bigraphs whose

Table 3.21 Encoding of $\mathcal{L}_{spat}$ into BiLog

Encodings:

$$
\begin{array}{rcl}
[\![0]\!]_X & \stackrel{\text{def}}{=} & X \otimes \mathbf{1} \\
[\![\neg A]\!]_X & \stackrel{\text{def}}{=} & \neg [\![A]\!]_X \\
[\![A \wedge B]\!]_X & \stackrel{\text{def}}{=} & [\![A]\!]_X \wedge [\![B]\!]_X \\
[\![A \mid B]\!]_X & \stackrel{\text{def}}{=} & \mathbf{join} \circ ([\![A]\!]_X \overset{X}{\otimes} [\![B]\!]_X) \\
[\![A \triangleright B]\!]_X & \stackrel{\text{def}}{=} & И Y.(((Y \leftarrow X) \otimes \mathbf{id}_1) \circ \mathbf{A}_X) \multimap (\mathbf{join} \circ ((X \leftleftarrows Y) \otimes \mathbf{id}_1) \leftarrow [\![B]\!]_X) \\
[\![\Diamond A]\!]_X & \stackrel{\text{def}}{=} & \bigvee_{a \in X} И y_1.y_2.Y_1.Y_2.\ \mathbf{Redex}_a^{y_1,y_2,Y_1,Y_2} \circ [(\mathbf{React}_a^{Y_1,Y_2} \leftarrow [\![A]\!]_X) \wedge \mathbf{Triple}]
\end{array}
$$

Supporting Formulae:

$$
\begin{array}{rcl}
\mathbf{Open} & \stackrel{\text{def}}{=} & \neg И x.\ \Diamond(/x \circ \mathbf{T}) \\
\mathbf{A}_X & \stackrel{\text{def}}{=} & [\![A]\!]_X \wedge \mathbf{T}_{\epsilon \to \langle 1, Y_2 \rangle} \wedge \mathbf{Open} \\
\mathbf{Triple} & \stackrel{\text{def}}{=} & \mathbf{T}_{\epsilon \to \langle 1, Y_1 \rangle} \otimes \mathbf{T}_{\epsilon \to \langle 1, Y_2 \rangle} \otimes \mathbf{T}_{\epsilon \to \langle 1, X \rangle}
\end{array}
$$

outerfaces have the same number of names as X. Proposition 30 will show that a process satisfies $[\![\Diamond A]\!]_X$ if and only if it is the combination between a particular redex and a bigraph that satisfies the requirement of Lemma 23, and moreover that the corresponding reactum satisfies $[\![A]\!]_X$.

Proposition 30 formalises the main result of the section. It expresses the semantical equivalence between $\mathcal{L}_{spat}$ and its encoding in BiLog, note, in particular, the requirement for a finite set of actions performable by the CCS processes. Such a limitation is not due to the presence of the next step operator. Indeed, inspecting the proof, one can see that the induction step for the temporal operator still holds in the case of a infinite set of actions. The limitation, in fact, is due to the adjoint operator $\triangleright$: the number of names shared between the processes must be bound. This happens because of the different choice for the logical product operator in BiLog. On one hand, spatial logic has parallel operator built in. This means that the logic does not care about the names that are actually shared between the processes. On the other hand, BiLog has a strong control on the names shared between two processes, and they must be known with accuracy.

Proposition 30. *If the set of names in every CCS process is bounded to be a finite set X, then $P \models_{spat} A$ if and only if $[\![P]\!]_X \models [\![A]\!]_X$.*

Proof. Proceed by induction on formula structure. Base of induction is formula 0. To assume $[\![P]\!]_X \models [\![0]\!]_X$ means $[\![P]\!]_X \equiv X \otimes 1$, that correspond to $P \equiv \mathbf{0}$, hence $P \models_{spat} 0$ by definition.

Inductive step deals with connectives. Treatments of $\neg$, $\wedge$ and $\mid$ are similar; hence focus on parallel operator.

Case $A \mid B$. To say $[\![P]\!]_X \models [\![A \mid B]\!]_X$ means that there are two bigraphs g_1, g_2,

with $g_1 \models [\![A]\!]_X$ and $g_1 \models [\![B]\!]_X$, such that $[\![P]\!]_X \equiv \mathit{join} \circ (g_1 \overset{X}{\otimes} g_2)$. The bigraphs g_1, g_2 must have type $\epsilon \to \langle 1, X \rangle$ and open links, as so does $[\![P]\!]_X$. By Lemma 22, there are two processes Q_1 and Q_2 such that $[\![Q_1]\!]_X$ and $[\![Q_2]\!]_X$ are g_1 and g_2, respectively. Then conclude $[\![P]\!]_X \equiv \mathit{join} \circ ([\![Q_1]\!]_X \overset{X}{\otimes} [\![Q_2]\!]_X)$, that means $P \equiv Q_1 \mid Q_2$, again by Lemma 22. Moreover, induction hypothesis says that $Q_1 \models A$ and $Q_2 \models B$, hence $P \models_{spat} A \mid B$.

Case $A \triangleright B$. Assume $[\![P]\!]_X \models [\![A \triangleright B]\!]_X$, then by definition there exists a fresh set Y of actions such that for every G satisfying $(((Y \leftarrow X) \otimes \mathbf{id}_1) \circ \mathbf{A}_X)$ it holds $[\![P]\!]_X \otimes G \models \mathbf{join} \circ ((X \leftleftarrows Y) \otimes \mathbf{id}_1) \leftarrowtriangle [\![B]\!]_X$, that is

$$\mathit{join} \circ ((X \leftleftarrows Y) \otimes \mathit{id}_1) \circ ([\![P]\!]_X \otimes G) \models [\![B]\!]_X \tag{3.11}$$

Now $G \models (((Y \leftarrow X) \otimes \mathbf{id}_1) \circ \mathbf{A}_X)$ means that there is $g \models \mathbf{A}_X$ such that $G \equiv ((Y \leftarrow X) \otimes \mathbf{id}_1) \circ g$. As previously discussed (cf. the introduction to the current proposition) $g \models \mathbf{A}_X$ says that $g \models [\![A]\!]_X$ and that g is a bigraph with open link and type $\epsilon \to \langle 1, X \rangle$. By Lemma 22, g is $[\![Q]\!]_X$ for some CCS process Q whose actions are in X.

Hence, as the set of actions *Acts* corresponds to X, (3.11) is rephrased by saying that for *every* CCS process Q such that $[\![Q]\!]_X \models [\![A]\!]_X$ it holds

$$\mathit{join} \circ ((X \leftleftarrows Y) \otimes \mathit{id}_1) \circ ([\![P]\!]_X \otimes ((Y \leftarrow X) \otimes \mathit{id}_1) \circ [\![Q]\!]_X) \models [\![B]\!]_X$$

that is $[\![P \mid Q]\!]_X \models [\![B]\!]_X$. Then, the induction hypothesis says that for every Q, if $Q \models_{spat} A$ then $P \mid Q \models_{spat} B$, namely $P \models_{spat} A \triangleright B$.

Case $\Diamond A$. to assume $[\![P]\!]_X \models [\![\Diamond A]\!]_X$ signifies that there exists an action $a \in X$ such that

$$[\![P]\!]_X \equiv \mathit{Redex}_a^{y_1, y_2, Y_1, Y_2} \circ H \tag{3.12}$$

where y_1, y_2 are fresh names, Y_1, Y_2 are fresh subsets with the same cardinality as X, and H is a bigraph satisfying

$$H \models (\mathbf{React}_a^{Y_1, Y_2} \leftarrowtriangle [\![A]\!]_X) \wedge \mathbf{Triple}. \tag{3.13}$$

In particular, Property (3.13) amounts to assert the two following points.

1. $H \models \mathbf{React}_a^{Y_1, Y_2} \leftarrowtriangle [\![A]\!]_X$, that means

$$\mathit{React}_a^{Y_1, Y_2} \circ H \models [\![A]\!]_X. \tag{3.14}$$

2. $H \models \mathbf{T}_{\epsilon \to \langle 1, Y_1 \rangle} \otimes \mathbf{T}_{\epsilon \to \langle 1, Y_2 \rangle} \otimes \mathbf{T}_{\epsilon \to \langle 1, X \rangle}$, that means

$$H \equiv H_1 \otimes H_2 \otimes H_3 \tag{3.15}$$

with $H_i : \epsilon \to \langle 1, Y_i \rangle$, for $i = 1, 2$, and $H_3 : \epsilon \to \langle 1, X \rangle$.

Now $[\![P]\!]_X \equiv Redex^{y_1,y_2,Y_1,Y_2} \circ (H_1 \otimes H_2 \otimes H_3)$, by (3.12) and (3.15). This means $[\![P]\!]_X \longrightarrow React_a^{Y_1,Y_2} \circ (H_1 \otimes H_2 \otimes H_3)$, by Lemma 23. Furthermore, the bigraphs H_1, H_2, H_3 have open links, as so does $[\![P]\!]_X$. Hence Lemma 22 says that there exists the CCS process Q such that $[\![Q]\!]_X$ corresponds to $React_a^{Y_1,Y_2} \circ (H_1 \otimes H_2 \otimes H_3)$, hence $P \rightarrow Q$ by Proposition 28. Finally, (3.14) says that $[\![Q]\!]_X \models [\![A]\!]_X$, and this means $Q \models_{spat} A$ by induction hypothesis. Conclude that $[\![P]\!]_X \models [\![\Diamond A]\!]_X$ is equivalent to $P \rightarrow Q$ with $Q \models_{spat} A$, namely $P \models_{spat} \Diamond A$. □

The main steps in encoding CCS spatial logic into BiLog have been to encode the underlying calculus into bigraphical settings, to find the right reaction rules and, and then to characterise the corresponding reactive contexts by BiLog formulae. This hints how it may be possible to extend such a result to other calculi, such as π and ambients by employing their encodings, already provided in [90, 88].

3.8 Conclusions and Realted Work

This chapter moves a first step towards describing global resources by focusing on bigraphs. Our final objective is to design a general dynamic logic able to cope uniformly with all the models bigraphs have been proved useful for, as of today these include λ-calculus [102], Petri-nets [100], CCS [103], π-calculus [90], ambient calculus [88], and context-aware systems [18]. We introduced BiLog, a logic founded on bigraphs, whose formulae describe arrows in monoidal categories.

We have seen how the 'separation' plays in various fragments of the logic. For instance, in the case of *Place Graph Logic*, where models are bigraphs without names, the separation is purely structural and coincides with the notion of parallel composition in Spatial Tree Logic. Dually, as the models for *Link Graph Logic* are bigraphs with no location, the separation in such a logic is disjointness of nominal resources. Finally, for *Bigraph Logic*, where nodes of the model are associated with names, the separation is not only structural, but also nominal, since the constraints on composition force port identifiers to be disjoint. In this sense, it can be seen as the separation in memory structures with pointers, like Separation Logic's heap structures [111], and trees with either pointers [37] or hidden names [40].

In §3.6 we sketched the application of BiLog to describe XML data, and we plan to extend the logic to more sophisticated semistructured data models. The similarities between XML and bigraphs have been pointed out independently also in [84] where XML is proposed as a language to codify bigraphs. In §3.6 we have focused on the other way around, by considering 'bigraphs as models for XML'.

In §3.7 we showed how BiLog can deal with dynamics. A natural solution is adding a temporal modality basically describing bigraphs that can compute according to a Bigraphical Reactive System [90]. When the transparency predicate enables the inspection of 'dynamic' controls, BiLog is '*intensional*' in the sense of [131], as it can observe internal structures. In the case of the bigraphical system describing CCS [103], BiLog can

be so intensional that its static fragment directly expresses a temporal modality. A transparency predicate specifies which structures can be directly observed by the logic, while a temporal modality, along with the spatial connectives, allows to deduce the structure by observing the behaviour. It would be interesting to isolate some fragments of the logic and investigate how the transparency predicate influences their expressivity and intensionality, as done in [85].

We have not addressed a logic for tree with hidden names. As a matter of fact, we have such a logic. More precisely we can encode abstract trees into bigraphs by controls ambs with arity one. The name assigned to this control will actually be the name of the ambient. Extrusion and renaming of abstract trees have their correspondence with closure and substitution of bigraphical terms. At the logical level we may encode operators of tree logic with hidden names as follows:

$$
\begin{array}{rcl}
\copyright\, a & \stackrel{\text{def}}{=} & ((a \leftarrow a) \otimes \mathbf{id}) \circ \mathbf{T} \\
\mathbf{C}x.\, A & \stackrel{\text{def}}{=} & И x.\, (/x \otimes \mathbf{id}) \circ A \\
a \circledR A & \stackrel{\text{def}}{=} & (\neg \copyright\, a \wedge A) \vee (/a \otimes \mathbf{id}) \circ A \\
\mathbf{H}x.\, A & \stackrel{\text{def}}{=} & И x.\, x \circledR A
\end{array}
$$

The operator $\copyright\, a$ says that the name a appears in the outer face of the bigraphs. The new quantifier $\mathbf{C}x.\, A$ expresses the fact that in a process satisfying A a name has been closed. The revelation $\circledR$ says that A can be asserted by revealing the restricted name a, which may be hidden in the model as it must either to be closed by an edge or not to appear in the model. The hiding quantification **H** is derived as in [45].

4

Concluding Remarks

Over the last years, the contribution of theoretical computer science to the design and implementation of programming languages has been widespread and highly effective. Exemplars are the operational and denotational, or logical, foundations of languages such as ML [79, 105], Prolog [51] and Scheme [133]. Even Java has been benefitted [49] and, very recently, theoretical research in resource semantics and substructural logic has led to significant advances in understanding of delicate concepts such as mutable data, as exemplified by separation logic's handling of constructs such as pointers [87, 111].

The contribution of theoretical work to design, understanding, and delivery of systems, however, has been much less widespread and much less effective. We propose this Thesis as a contribution to the programme of addressing this weakness in systems theory, and we provide a logical analysis of some of the key structural aspects of distributed systems, or network, architectures. In particular:

- We introduce *'Logic'* from a foundational point of view, with Basic Logic and its principles, which are essential to provide a resource model (relational monoids) proved to be sound and refined complete. In particular, the completeness result allows a semantical proof of cut-elimination.

- We extend Basic Logic in two directions: one direction is the addition of structural rules, the other is the move to intuitionistic logic, thus obtaining Intuitionistic Linear Logic and Intuitionistic Logic. By combining the two extensions we obtain the Logic of Bunched Implications. In turn, we extend to these logics the notion of model, along with the soundness and refined completeness results.

- To express resource placement, we introduce an intuitionistic, hybrid modal logic in which formulae and sentences are explicitly about particular 'places' of the considered system. Several modalities allow validation of properties at a named place, at some unknown place and at every place. We provide a sound and complete Kripke semantics and a more general one, termed Birelational Models, which, in addition to soundness and completeness, enjoys the finite model property, thus allowing the decidability of the logic to be demonstrate.

- We found a logic, BiLog, with a specific model in mind: the bigraphs model of distributed computation. We define BiLog as a "spatial logic for monoidal categories,"

in the style of Tree and Ambient Logics. By varying the underlying monoidal category we obtain various different logics: Place Graph Logic (PGL) has as its model the place-graph part of the bigraphs model; Link Graph Logic (LGL) has the link-graphs as its model; and Bigraph Logic combines these two to give a logic for discussing bigraphs themselves.

- We propose BiLog as a general description language and, to support this idea, we provide several results: Tree Logic is encoded in PGL; Graph Logic is encoded in LGL; and Bigraph Logic is powerful enough to encode Context Tree Logic and a Spatial Logic for CCS. In particular, in this last encoding, BiLog is sufficiently expressive to identify pairs of terms which are redex and reactum in the usual reduction semantics of CCS, thus allowing the encoding of 'temporal' modalities on CCS terms without having temporal operators in the logic.

- We discuss how BiLog can be used for describing, querying and reasoning about XML. We analyse three possible cases: PGL to model XML data trees and tree contexts, without nominal resources; PGL extended by named controls and renamings to model XML data trees with identified nodes; and Bigraph Logic to model XML data trees with soft-link connections, that are implemented with nominal resources.

Resources have been often modelled with monoids. The monoidal operation expresses the resource combination and a binary relation may be intended as resource production, accessibility, sharing or interdependency. Prime example are Bunched Implications' resource monoids [71, 110, 122], the spatial semantics for *Pointer Logic* [87] based on partial monoids, and the semantics for Linear Logic based on resource consumption [74].

Relational monoids have been proposed in §1.3 as a generalisation of resource semantics. In fact, we related our semantics to more traditional models for the various extensions of Basic Logic: phase spaces [74], linear frames [137], pretopologies [127], Kripke semantics [14, 94, 136], and formal topologies [126].

In particular, the two semantics presented in §1.8.1 provide complementary new results about Bunched Implications Logic and they leave open the possibility of further investigations on decidability and proof search results in the spirit of [71]. In particular, the semantics presented in [70], and refined in [71], is deeply related to the semantics of **LBI** partially ordered monoids, as the specific treatment of inconsistency, $\bot$, involves a topological closure operator. This similarity may enable the study of decidability and proof search results on **LBI** and then to go backward and project those results on Basic Logic and its extensions.

Chapter 2 shows how to extend an intuitionistic logic with the notion of location. As a next step in the research, we may think to introduce the idea of a structure among the located resources. Doing this, we might achieve either the descriptive power of the logic proposed in [17], which enriches the Logic of Bunched Implications with locations; or the generality of the logic defined in [4], which is a sort of separation logic with places that describes hierarchical storage; or the expressivity of the logic presented in [16], which is a multi-modal intuitionistic linear logic with locations. Indeed, the works [4, 16, 17] hint

that the results presented in Chapter 2 may be positively applied to **LBI** or **IL**. This is what we are currently investigating.

A major limitation of the logic presented in Chapter 2 is that if a formula φ is validated at some named place, say p, then the formula $\varphi@p$ can be inferred at every other place. Similarly, if $\Diamond\varphi$ or $\Box\varphi$ can be inferred at one place, then they can be inferred at any other place. In a large distributed system, we may want to restrict the rights of accessing information in a place. This can be done by adding an accessibility relation as is done in the case of other intuitionistic modal systems [132, 32].

We are currently investigating if the proof of the finite model property can be adapted to the hybrid versions of other intuitionistic modal systems. We are also investigating the computational interpretation of these extensions. This would result in extensions of λ-calculus presented in §2.3, which provides a link between the modal logic with places and the world of computation via the Curry-Howard isomorphism. We also plan to investigate adding temporal modalities to the logic. This will help us to reason about space and time.

From a purely logical point of view, in Chapter 2, the meta-logic that reasons about soundness and completeness is classical. In order to obtain a full intuitionistic and constructive account, we plan to extend the results of Chapter 1 to modal logics.

Chapter 3 presents another way of describing global resources, with BiLog. Our final objective is to design a general dynamic logic able to cope uniformly with all the models bigraphs have been proved useful for.

BiLog may at first appear complex and over-provided of connectives. On the contrary, the backbone of the logic is relatively simple, consisting of two operators regulated by elementary monoidal and interchange laws. Such a structure gives then rise to many – occasionally complex – derived connectives. This is a fundamental expressiveness property that does not put us off: BiLog is in fact meant to be a comprehensive meta-level framework in which several different logics can be isolated, understood and compared.

In particular, we have seen how the 'separation' plays in various fragments of the logic. For instance, in the case of PGL, where models are bigraphs without names, the separation is purely structural and coincides with the notion of parallel composition in Spatial Tree Logic. Dually, as the models for LGL are bigraphs with no location, the separation in such a logic is disjointness of nominal resources.

For Bigraph Logic, where nodes of the model are associated with names, the separation is not only structural, but also nominal, since the constraints on composition force port identifiers to be disjoint. In this sense, it can be seen as the separation in memory structures with pointers, like Separation Logic's heap structures [111], and trees with either pointers [37] or hidden names [40].

Section 3.3.2 introduces the transparency predicate τ to hint how to restrict BiLog's descriptive power. The idea is to limit the structures that the logic can observe and express with its formal language. Although the definition of τ is justified by several examples directly related to computer science, all the results we present are proved on the assumption that the transparency predicate is always true. This happens because our aim here is to introduce BiLog and show its generality as a descriptive language.

Indeed, τ predicate deserves further investigations. Fist we plan to consider the par-

ticular characterisation of logical equivalence provided by Theorem 23 and generalise the result to a congruence 'up-to' transparency. That means we may find an equivalence relation between terms that is tuned by τ: more τ covers, less the equivalence distinguishes.

The study of the transparency predicate is orthogonal to the dynamics in BiLog. In fact, restriction of the observational power in the static logic does not hinder in general a restriction of the observational power in the dynamic counterpart, that is because the next step modality could allow a re-intensionalisation of the controls by observing the evolution of the model (c.f. [35, 131]).

A transparency predicate specifies which structures can be directly observed by the logic, while a temporal modality, along with the spatial connectives, allows to deduce the structure by observing the behaviour. It would be interesting to isolate some fragments of the logic and investigate how the transparency predicate influences their expressivity and intensionality, as done in [85].

Section 3.6 sketches the application of BiLog to describe XML data. We intend to extend the logic to semistructured data in general. The work in [52] provides further investigations in this sense.

The existential/universal quantifiers are omitted in BiLog as they imply an undecidable satisfaction relation (cf. [50]), while we aim at a decidable logic. The decidability of BiLog is an open question. We plan to extend the result of [36] to isolate decidable fragments of BiLog.

To obtain a robust logical setting, we are developing a proof theory, and, in particular, a sequent calculus that will be useful to compare BiLog with other spatial logics, not only with respect to the model theory, but also from a proof theoretical point of view.

Several important questions remain: as bigraphs have an interesting dynamics, specified by reactions rules, we plan to extend BiLog to such a framework. Building on the encodings of ambient and π-calculi into bigraphical reactive systems, we expect a dynamic BiLog to be able to express both ambient logic [42] and spatial logics for π-calculus [33].

Finally, the work in [60] suggests to instantiate BiLog by considering the *Binding Bigraphs*, to which it provides an axiomatisation in the spirit of [101]. Moreover, the work [109] hints how to explore the role of the newly defined *Kind Bigraphs* within BiLog.

Bibliography

[1] S. Abiteboul, O. Benjelloun, and T.Milo. Positive active XML. In *Proc. of Symposium on Principles of Database Systems (PODS)*, pages 35–45. ACM Press, 2004.

[2] S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web: from relations to semistructured data*. Morgan Kaufmann, 1999.

[3] M. Abrusci. Phase semantics and sequent calculus for pure noncommutative classical linear propositional logic. *The Journal of Symbolic Logic*, 56:1403–1451, 1991.

[4] A. Ahmed, L. Jia, and D. Walker. Reasoning about hierarchical storage. In *Proc. of the Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 33–44. IEEE Computer Society Press, 2003.

[5] A. R. Anderson and N. D. Belnap. *Entailment:the logic of relevance and necessity*, volume I. Princeton University Press, 1975.

[6] A. R. Anderson, N. D. Belnap, and J. M. Dunn. *Entailment:the logic of relevance and necessity*, volume II. Princeton University Press, 1992.

[7] P. B. Andrews. Resolution in type theory. *Journal of Symbolic Logic*, 36(3):414–432, 1971.

[8] C. Areces and P. Blackburn. Bringing them all together. *Journal of Logic and Computation*, 11(5):657–669, 2001.

[9] C. Areces, P. Blackburn, and M. Marx. Hybrid logics: Characterization, interpolation and complexity. *Journal of Symbolic Logic*, 66:997–1010, 2001.

[10] A. Asperti. Light affine logic. In *Proc. of the Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 300–308. IEEE Computer Society Press, 1998.

[11] A. Avron. The semantics and proof theory of linear logic. *Theoretical Computer Sience*, 57:161–184, 1988.

[12] F. Belardinelli, P. Jipsen, and H. Ono. Algebraic aspects of cut elimination. *Studia Logica*, 77(2):209–240, 2004.

[13] J. Bergstra and W. Klop. Process algebra for synchronous communication. *Information and Computation*, 60, 1984.

[14] E.W. Beth. Semantic construction of intuitionistic logic. *Kon. Neder. Akad. van Wetensch. Afd. Let. Med. Nieuwe Reeks*, 19(11):357–388, 1956.

[15] E.W. Beth. *The Foundations of Mathematics*. North-Holland Publ. Co., Amsterdam, 2nd edition, 1965.

[16] N. Biri and D. Galmiche. A modal linear logic for distribution and mobility (abstract). In *Proc. of International Workshop on Linear Logic (WLL)*, Copenhagen, Danemark, 2002.

[17] N. Biri and D. Galmiche. A separation logic for resource distribution. In *Proc of IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 2914 of *LNCS*, pages 23–37. Springer Verlag, 2003.

[18] L. Birkedal, S. Debois, E. Elsborg, T. Hildebrandt, and H. Niss. Bigraphical models of context-aware systems. In *Proc. of Foundations of Software Science and Computation Structures (FOSSACS)*, 2006. To appear.

[19] G. Birkhoff. *Lattice Theory*, volume 25 of *AMS Colloquium publications*. American Mathematical Society, Providence, Rhode Island, 3rd edition, 1967.

[20] G. Birkhoff and J. von Neumann. The logic of quantum mechanics. *Annals of Mathematics*, 37:823–843, 1936.

[21] P. Blackburn. Internalizing labelled deduction. *Journal of Logic and Computation*, 10:137–168, 2000.

[22] P. Blackburn. Representation, reasoning, and relational structures: a hybrid logic manifesto. *Logic Journal of the IGPL*, 8:339–365, 2000.

[23] P. Blackburn and J. Seligman. What are hybrid languages? In M. Kracht, M. de Rijke, H. Wansing, and M. Zakharyaschev, editors, *Advances in modal logic*, volume 1, pages 41–62. CSLI, 1996.

[24] A. Bossi, R. Focardi, D. Macedonio, C. Piazza, and S. Rossi. Unwinding in information flow security. In *Proc. of Workshop MEFISTO*, volume 99 of *ENTCS*, pages 127–154. Elsevier Sciences, 2004.

[25] A. Bossi, D. Macedonio, C. Piazza, and S. Rossi. Secure contexts for information flow security. Technical Report CS-2002-18, Dipartimento di Informatica, Università Ca' Foscari di Venezia, Italy, 2002.

[26] A. Bossi, D. Macedonio, C. Piazza, and S. Rossi. Compositional action refinement and information flow security. Technical Report CS-2003-13, Dipartimento di Informatica, Università Ca' Foscari di Venezia, Italy, 2003.

[27] A. Bossi, D. Macedonio, C. Piazza, and S. Rossi. Information flow security and recursive systems. In *Proc. of the Italian Conference on Theoretical Computer Science (ICTCS'03)*, volume 2841 of *LNCS*, pages 369–382. Springer-Verlag, 2003.

[28] A. Bossi, D. Macedonio, C. Piazza, and S. Rossi. P_BNDC and replication. Technical Report CS-2003-6, Dipartimento di Informatica, Università Ca' Foscari di Venezia, Italy, 2003.

[29] A. Bossi, D. Macedonio, C. Piazza, and S. Rossi. Secure contexts (extended abstract). In *Electronic Proceedings of the Workshop on Issue in the Theory of Security (WITS'03)*, 2003.

[30] A. Bossi, D. Macedonio, C. Piazza, and S. Rossi. Secure contexts for confidential data. In *Proc. of the 16th IEEE Computer Security Foundations Workshop (CSFW'03)*, pages 14–28. IEEE Computer Society Press, 2003.

[31] A. Bossi, D. Macedonio, C. Piazza, and S. Rossi. Information flow in secure contexts. *Journal of Computer Security*, 13(3):391–422, 2005.

[32] T. Braüner and V. de Paiva. Towards constructive hybrid logic (extended abstract). In *Elec. Proc. of Methods for Modalities 3 (M4M3)*, 2003.

[33] L. Caires and L. Cardelli. A spatial logic for concurrency (Part I). In *Proc. of International Symposium on Theoretical Aspects of Computer Software (TACS)*, volume 2215 of *LNCS*, pages 1–37. Springer-Verlag, 2001.

[34] L. Caires and L. Cardelli. A spatial logic for concurrency (part II). In *Proc. of International Conference on Concurrency Theory (CONCUR)*, volume 2421 of *LNCS*, page 209. Springer-Verlag, 2002.

[35] L. Caires and É. Lozes. Elimination of quantifiers and undecidability in spatial logics for concurrency. In *Proc. of International Conference on Concurrency Theory (CONCUR)*, volume 3170 of *LNCS*, pages 240–257. Springer-Verlag, 2004.

[36] C. Calcagno, L. Cardelli, and A. D. Gordon. Deciding validity in a spatial logic for trees. In *Proc. of ACM SIGPLAN Workshop on Types in Language Design and Implementation (TLDI)*, pages 62 – 73. ACM Press, 2003.

[37] C. Calcagno, P. Gardner, and U. Zarfaty. A context logic for tree update. In *Proc. of ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)*, pages 271–282. ACM Press, 2005.

[38] L. Cardelli. Describing semistructured data. *SIGMOD Record, Database Principles Column*, 30(4), 2001.

[39] L. Cardelli, P. Gardner, and G. Ghelli. A spatial logic for querying graphs. In *Proc. of International Colloquium on Automata, Languages and Programming (ICALP)*, volume 2380 of *LNCS*, pages 597 – 610. Springer-Verlag, 2002.

[40] L. Cardelli, P. Gardner, and G. Ghelli. Manipulating trees with hidden labels. In *Proc. of International Conference on Foundations of Software Science and Computational Structures (FOSSACS)*, volume 2620 of *LNCS*, pages 216–232. Springer-Verlag, 2003.

[41] L. Cardelli and G. Ghelli. TQL: A query language for semistructured data based on the ambient logic. *Mathematical Structures in Computer Science*, 14:285–327, 2004.

[42] L. Cardelli and A. D. Gordon. Ambient logic. *Mathematical Structures in Computer Science*. To appear.

[43] L. Cardelli and A. D. Gordon. Anytime, anywhere. Modal logics for mobile ambients. In *Proc. of ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)*, pages 365–377. ACM Press, 2000.

[44] L. Cardelli and A. D. Gordon. Mobile ambients. *Theoretical Computer Science, Special Issue on Coordination*, 240(1):177–213, 2000.

[45] L. Cardelli and A. D. Gordon. Logical properties of name restriction. In *Proc. of International Conference on Typed Lambda Calculi and Applications (TCLA)*, volume 2044 of *LNCS*, pages 46–60. Springer-Verlag, 2001.

[46] L. Cardelli, P.Gardner, and G. Ghelli. Querying trees with pointers. Unpublished notes.

[47] R. Chadha, D. Macedonio, and V. Sassone. A hybrid intuitionistic logic: Semantics and decidability (extended version). Computer Science Report 2005:07, University of Sussex, 2005.

[48] R. Chadha, D. Macedonio, and V. Sassone. A hybrid intuitionistic logic: Semantics and decidability. *Journal of Logic and Computation*, February 2006. To appear.

[49] P. Chalin. Reassessing JML's logical foundation. In *Proc. of Workshop on Formal Techniques for Java-like Programs (FTfJP)*, 2005.

[50] W. Charatonik and J.M. Talbot. The decidability of model checking mobile ambients. In *Proc. of International Workshop on Computer Science Logic (CSL)*, volume 2142 of *LNCS*, pages 339 – 354. Springer-Verlag, 2001.

[51] A. Colmerauer and P. Roussel. *History of Programming Languages*, chapter VII The birth of Prolog. ACM Press/Addison-Wesley, 1996.

[52] G. Conforti. *Spatial Logics for Semistructured Resources*. Ph.D. Thesis, Informatics Department, University of Pisa, 2005.

[53] G. Conforti and G. Ghelli. Decidability of freshness, undecidability of revelation. In *Proc. of International Conference on Foundations of Software Science and Computational Structures (FOSSACS)*, volume 2987 of *LNCS*, pages 105–120. Springer-Verlag, 2004.

[54] G. Conforti, D. Macedonio, and V. Sassone. Bigraphical logics for XML. In *Proc. of Italian Symposium on Advanced Database Systems (SEBD'05)*, pages 392 – 399, 2005.

[55] G. Conforti, D Macedonio, and V Sassone. BiLog: spatial logics for bigraphs. Computer Science Report 2005:02, University of Sussex, 2005.

[56] G. Conforti, D. Macedonio, and V. Sassone. Spatial logics for bigraphs. In *Proc. of International Colloquium on Automata, Languages and Programming (ICALP'05)*, volume 3580 of *LNCS*, pages 766 – 778. Springer-Verlag, 2005.

[57] Silvano Dal Zilio and Denis Lugiez. A logic you can count on. In *Proc. of ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)*, pages 135–146. ACM Press, 2004.

[58] M. L. dalla Chiara and R. Giuntini. Paraconsistent quantum logic. *Foundations of Physics*, 19:891–904, 1989.

[59] M. L. dalla Chiara and R. Giuntini. *Handbook of Philosophical Logic*, volume 6, chapter Quantum Logics, pages 129–228. Kluwer Academic Publishers, 2nd edition, 2002.

[60] T. C. Damgaard and L. Birkedal. Axiomatizing binding bigraphs (revised). Technical Report TR-2005-71, IT University of Copenhagen, 2005.

[61] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge Universtity Press, Cambridge, 2nd edition, 1991.

[62] M. De Marco and J. Lipton. Completeness and cut-elimination in the intuitionistic theory of types. Draft, April 2004.

[63] W. B. Ewald. *Time, Modality and Intuitionism*. PhD thesis, University of Oxford, 1978.

[64] C. Faggian. Basic logic and linear negation: a new approach to orthologic. Draft, 1997.

[65] C. Faggian, D. Macedonio, and G. Sambin. Towards modularity in proof theory. To appear.

[66] C. Faggian and G. Sambin. From basic logic to quantum logics with cut-elimination. In *Proceedings of the International Quantum Structures Association Berlin*, volume 37 of *International Journal of Theoretical Physics (Special Issue)*, pages 31–37, 1996.

[67] G. Fisher Servi. Semantics for a class of intuitionistic modal calculi. In M. L. dalla Chiara, editor, *Italian Studies in the Philosophy of Science*, pages 59–72. Reidel Publishing Company, 1981.

[68] P. Freyd. *Abelian Categories: an Introduction to the Theory of Functors*. Harper and Row, New York, 1964.

[69] J. Gallier. Constructive logics. Part II: linear logic and proof nets. Technical report, CIS Departement University of Pennsylvania, 1991.

[70] D. Galmiche, D. Méry, and D. J. Pym. Resource tableaux (extended abstract). In *Proc. of International Workshop on Computer Science Logic (CSL)*, volume 2471 of *LNCS*, pages 183–198. Springer-Verlag, 2002.

[71] D. Galmiche, D. Méry, and D. J Pym. The semantics of BI and resource tableaux. *Mathematical Structures in Computer Science*, 15(6):1033–1088, 2005.

[72] G. Gentzen. Untersuchungen über das logische schließen (I-II). *Mathematische Zeitschrift*, 39:176–210, 405–431, 1935.

[73] G. Gentzen. *The Collected Papers of Gerhard Gentzen*. North-Holland, 1969. Edited by M.E. Szabo.

[74] J.-Y. Girard. Linear logic. *Theoretical Computer Sience*, 50:1–102, 1987.

[75] J.-Y. Girard. *Proofs and Types*. Cambridge University Press, 1989.

[76] R. Goldblatt. Semantics analysis of orthologic. *Journal of Philosophical Logic*, 3:19–35, 1974.

[77] F. Guidi. Basic pairs as semantics for the conjunctive fragment of the calculus BS. Manuscript, September 2000.

[78] U. Hansmann, M.S. Nicklous, Thomas Schäck, and F. Seliger. *Smart Card Application Development Using Java*. Springer, 2000.

[79] R. Harper, D. Macqueen, R. Milner, and M. Tofte. *The Definition of Standard ML (Revised)*. MIT Press, 1997.

[80] R. Harrop. On the existence of finite models and decision procedures for propositional calculi. In *Proc. of Cambridge Philosophical Society*, volume 54, pages 1–13, 1958.

[81] M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM*, 32(1):137–161, 1985.

[82] M. Hennessy and J. Riely. Resource access control in systems of mobile agents. *Information and Computation*, 173:82–120, 2002.

[83] O. Hermant. Semantic cut elimination in the intuitionistic sequent calculus. In *Proc. of Typed Lambda Calculi and Applications: 7th International Conference (TLCA'05)*, volume 3461 of *LNCS*, pages 221–233. Springer-Verlag, 2005.

[84] T. Hildebrandt and J.W. Winther. Bigraphs and (Reactive) XML, an XML-centric model of computation. Technical Report TR-2005-26, University of Copenhagen, February 2005.

[85] D. Hirschkoff. An extensional spatial logic for mobile processes. In *Proc. of International Conference on Concurrency Theory (CONCUR)*, volume 3170 of *LNCS*, pages 325–339. Springer-Verlag, 2004.

[86] C. A. R. Hoare. *Communicating Sequential Processes*. Pretience Hall, 1985.

[87] S. Ishtiaq and P. W. O'Hearn. BI as an assertion language for mutable data structures. In *Proc. of ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)*. ACM Press, 2001.

[88] O. H. Jensen. Forthcoming Ph.D. Thesis. Aalborg University.

[89] O. H. Jensen and R. Milner. Bigraphs and transitions. In *Proc. of the ACM SIGPLAN-SIGACT symposium on Principles of programming languages (POPL)*, pages 38–49. ACM Press, 2003.

[90] O. H. Jensen and R. Milner. Bigraphs and mobile processes (revised). Technical Report UCAM-CL-TR-580, University of Cambridge, February 2004.

[91] L. Jia and D. Walker. Modal proofs as distributed programs. Technical Report TR-671-03, Princeton University, 2003.

[92] L. Jia and D. Walker. Modal proofs as distributed programs (extended abstract). In *Proc. of European Symposium on Programming (ESOP)*, volume 2986 of *LNCS*, pages 219–233. Springer Verlag, 2004.

[93] S. A. Kripke. Semantical analysis of modal logic I: Normal modal propositional calculi. In *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, volume 9, pages 67–96, 1963.

[94] S. A. Kripke. Semantical analysis of intuitionistic logic (I). In *Proc. of Logic Colloquium, Oxford 1963*, pages 92–130. North-Holland Publishing Company, 1965.

[95] D. Macedonio and G.Sambin. Relational semantics for basic logic. *The Journal of Symbolic Logic*. To appear.

[96] D. Miller, G. Nadathur, F. Pfenning, and A. Scedrov. Uniform proofs as a foundation for logic programming. *Annals of Pure and Applied Logic*, 51(1-2):125–157, 1991.

[97] R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.

[98] R. Milner. Sorts in the π-calculus (extended abstract). In *Proc. of the 3rd Workshop on Concurrency and Compositionality*, volume 191 of *GMD-Studien*. GMD, Bonn, 1991.

[99] R. Milner. Bigraphical reactive systems. In *Proc. of International Conference on Concurrency Theory (CONCUR)*, volume 2154 of *LNCS*, pages 16–35. Springer-Verlag, 2001.

[100] R. Milner. Bigraphs for Petri nets. In *Lectures on Concurrency and Petri Nets: Advances in Petri Nets*, volume 3098 of *LNCS*, pages 686–701. Springer-Verlag, 2004.

[101] R. Milner. Axioms for bigraphical structure. *Mathematical Structures in Computer Science*, 15(6):1005–1032, 2005.

[102] R. Milner. Bigraphs whose names have multiple locality. Technical Report UCAM-CL-TR-603, University of Cambridge, January 2005.

[103] R. Milner. Pure bigraphs: Structure and dynamics. *Information and Computation*, 204(1):60–122, 2006.

[104] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, parts I and II. *Information and Computation*, 100(1):1–77, 1992.

[105] Robin Milner. A proposal for standard ML. In *Proc. of ACM Symposium on LISP and functional programming*, pages 184 – 197. ACM Press, 1984.

[106] J. Moody. Modal logic as a basis for distributed computation. Technical Report CMU-CS-03-194, Carnegie Mellon University, 2003.

[107] T. Murphy, VII, K. Crary, R. Harper, and F. Pfenning. A symmetric modal lambda calculus for distributed computing. In *Proc. of the Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 286–295. IEEE Computer Society Press, 2004.

[108] T. Murphy, VII, R. Harper, and K. Crary. Distributed control flow with classical modal logic. In *Proc. of International Workshop on Computer Science Logic (CSL)*, volume 3634 of *LNCS*, pages 51–69. Springer Verlag, 2005.

[109] S. O'Conchuir. Kind bigraphs - static theory. Thecnical Report TCD-CS-2005-36, Trinity College Dublin, Computer Science Department, 2005.

[110] P. W. O'Hearn and D. J. Pym. The logic of bunched implications. *Bulletin of Symbolic Logic*, 5(2):215–244, 1999.

[111] P. W. O'Hearn, J. C. Reynolds, and H. Yang. Local reasoning about programs that alter data structures. In *Proc. of International Workshop on Computer Science Logic (CSL)*, volume 2142 of *LNCS*, pages 1–19. Springer-Verlag, 2001.

[112] M. Okada. A uniform semantic proof for cut-elimination and completeness fo various first and higher order logics. *Theoretical Computer Science*, 281:471–498, 2002.

[113] H. Ono. *On Some Intuitionistic Modal Logics*, volume 13, pages 687–722. Publications of RIMS, Kyoto University, 1977.

[114] H. Ono and N.-Y. Suzuki. Relations between intuitionistic modal logics and intermediate predicate logics. *Reports on Mathematical Logic*, 22:65–87, 1988.

[115] D. Pattinson and B. Reus. A complete temporal and spatial logic for distributed systems. In *Frontiers of Combining Systems (FroCoS)*, volume 3717 of *LNAI*, pages 122–137. Springer-Verlag, 2005.

[116] A. M. Pitts. Nominal logic: a first order theory of names and binding. In *Proc. of International Symposium on Theoretical Aspects of Computer Software (TACS)*, volume 2215 of *LNCS*, pages 219–242. Springer-Verlag, 2001.

[117] G. D. Plotkin and C. P. Stirling. *Theoretical Aspects of Reasoning About Knowledge*, chapter A Framework for Intuititionistic Modal Logic. J. Y. Halpern, 1986.

[118] D. Prawitz. Hauptsatz for higher order logic. *Journal of Symbolic Logic*, 33(3):452–457, 1968.

[119] A. N. Prior. *Time and Modality*. Oxford University Press, 1957.

[120] A. N. Prior. *Past, Present and Future*. Oxford University Press, 1967.

[121] A. N. Prior. *Papers on Time and Tense*. Oxford University Press, 1968.

[122] D. J. Pym. *The Semantics and Proof Theory of the Logic of Bunched Implications*, volume 26 of *Applied Logic Series*. Kluwer Academic Publishers, 2002.

[123] D. J. Pym, P. W. O'Hearn, and H. Yang. Possible worlds and resources: The semantics of BI. *Theoretical Computer Science*, 315(1):257–305, 2004.

[124] D. J. Pym and C. Tofts. A calculus and logic of resources and processes. Technical Report HPL-2004-170R1, HP Laboratories Bristol, 2005.

[125] J. Reynolds. Separation logic: a logic for shared mutable data structures. In *Proc of the Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 55–74. IEEE Computer Society Press, 2002.

[126] G. Sambin. Intuitionistic formal spaces - a first communication. In *Mathematical Logic and its Applications*, pages 187–204. Plenum Press, New York, 1987.

[127] G. Sambin. Pretopologies and completeness proofs. *The Journal of Symbolic Logic*, 60:861–878, 1995.

[128] G. Sambin. Basic logic, a structure in the space of logic, 1998. To appear.

[129] G. Sambin, G. Battilotti, and C. Faggian. Basic logic: Reflection, symmetry, visibility. *The Journal of Symbolic Logic*, 65:979–1013, 2000.

[130] G. Sambin and S. Valentini. Building up a toolbox for Martin-Löf's type theory: Subset theory. In G. Sambin and J. Smith, editors, *Twenty-five years of constructive type theory. Proceedings of the Congress held in Venice, October, 1995*, pages 221–224. Oxford U.P., 1998.

[131] D. Sangiorgi. Extensionality and intensionality of the ambient logic. In *Proc. of ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)*, pages 4–13. ACM Press, 2001.

[132] A. K. Simpson. *The Proof Theory and Semantics of Intuitionistic Modal Logic*. PhD thesis, University of Edinburgh, 1994.

[133] G. J. Sussman and G. L. Steele Jr. Scheme: An interpreter for extended lambda calculus. *Higher-Order and Symbolic Computation*, 11(4):405–439, 1998.

[134] M. Takahashi. A proof of cut-elimination theorem in simple type-theory. *Journal of the Mathematical Society of Japan*, 19(4):399–410, 1967.

[135] A. S. Troelstra and H. Schwichtenberg. *Basic Proof Theory*. Number 43 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2nd edition, 2000.

[136] A. S. Troelstra and D. van Dalen. *Constructivism in Mathematics: An Introduction*, volume 2. Elsevier Science Publishers, 1988.

[137] A. Ursini. Semantical investigations of linear logic. Rapp. Matematico CS-2002-18, Università di Siena, 1995.

[138] D. van Dalen. *Logic and Structure*. Springer Verlag, 4th extended edition, 2004.