Quantum State Transformations and Branching Distributed Temporal Logic[⋆]

Luca Viganò[1], Marco Volpe[2], and Margherita Zorzi[2]

[1] Department of Informatics, King's College London, UK
[2] Dipartimento di Informatica, Università di Verona, Italy

Abstract. The Distributed Temporal Logic DTL allows one to reason about temporal properties of a distributed system from the local point of view of the system's agents, which are assumed to execute independently and to interact by means of event sharing. In this paper, we introduce the Quantum Branching Distributed Temporal Logic QBDTL, a variant of DTL able to represent quantum state transformations in an abstract, qualitative way. In QBDTL, each agent represents a distinct quantum bit (the unit of quantum information theory), which evolves by means of quantum transformations and possibly interacts with other agents, and n-ary quantum operators act as communication/synchronization points between agents. We endow QBDTL with a DTL-style semantics, which fits the intrinsically distributed nature of quantum computing, we formalize a labeled deduction system for QBDTL, and we prove the soundness of this deduction system with respect to the given semantics. Finally, we discuss possible extensions of our system in order to reason about entanglement phenomena.

1 Introduction

Background and motivation The *Distributed Temporal Logic DTL* [12, 5, 6] allows one to reason about temporal properties of a distributed system from the local point of view of the system's agents: each asynchronous agent executes independently, evolves linearly along a time-line built upon some local events, and can interact with the other agents by means of event sharing. Distribution is implicit and properties of an entire system are formulated in terms of the local properties of the system's agents and their interaction. DTL's semantics was inspired by a conflict-free version of Winskel's *event structures* (see, e.g., [26]), enriched with information about sequential agents.

DTL has been initially proposed as a logic for specifying and reasoning about distributed information [12], but it has also been used in the context of security protocol analysis to reason about the interplay between protocol models and security properties [6]. In this paper, we show that, after a proper extension of the logic's syntax and

[⋆] The work presented in this paper was partially supported by the EU FP7 Marie Curie PIRSES-GA-2012-318986 project "GeTFun: Generalizing Truth-Functionality". Part of this work was carried out while Luca Viganò was at the Dipartimento di Informatica, Università di Verona, Italy.

semantics, DTL is also able to formally model quantum state transformations in an abstract, qualitative way.

Quantum computing is one of the most promising research fields of computer science as well as a concrete future technology (see [22] for a useful introduction to the basic notions of quantum computing as we here only very briefly summarize the notions that are relevant to our work in this paper). However, at least from the point of view of theoretical computer science, a number of foundational aspects are still underdeveloped: quantum complexity, quantum computability, quantum programming theory (and its logical account), quantum cryptography and security are all active but open research areas, which still require the development of ad hoc formal methods. These issues are complex to face since the physical model quantum computing is based on is sophisticated and all basic definitions and formal tools have to be reformulated in a non-standard way.

To illustrate this, and our contributions in this paper, in more detail, let us focus our attention on quantum data, in particular on the unit of quantum information, the *quantum bit* or *qubit*, for short. The qubit is the quantum counterpart of the classical bit and, mathematically, it is simply a normalized vector of the Hilbert Space $\mathbb{C}^2$. Qubits can assume both classical values 0 and 1 (as the classical bit) and all their *superpositional values*, i.e., linear combinations such as $\alpha|0\rangle + \beta|1\rangle$, where $\alpha, \beta \in \mathbb{C}$ are called *amplitudes*, $|\alpha|^2 + |\beta|^2 = 1$ and $|c\rangle$, for $c \in \{0, 1\}$, is the so called *Dirac Notation*, which is simply a denotation of basis states (which corresponds to the classical values a bit can assume).

Intuitively, whereas a classical bit can only be 0 or 1, a quantum bit can assume both the value 0 and the value 1 (with a certain associated probability) at the same time. It is possible to modify a quantum bit in two ways:

- by means of a suitable class of algebraic operators called *unitary transformations* (that are also called *quantum gates* and are a class of algebraic operators enjoying some good properties, which represent the pure quantum computational steps) or
- by *measuring* it, i.e., probabilistically reducing it to 0 or 1.

In this paper, we deal only with unitary transformations, leaving measurement for future work.

The definition of a qubit can, of course, be generalized: a *quantum register* or *quantum state* is the representation of a system of *n* qubits (mathematically, it is a normalized vector of the Hilbert space $\mathbb{C}^{2^n}$). As for the single qubit, a quantum state can be modified by means of unitary algebraic operators.

Abstracting from any notion of control and considering only pure quantum transformations (i.e., unitary evolution of quantum states as computational steps), it seems to be interesting to provide a logical account of such a computation. The question then is: what is a logical approach suitable to represent quantum state evolution?

Contributions The main contribution of this paper is the formalization of a logic and of an associated deduction system that allows one to formally represent and reason about unitary transformations of quantum states from a temporal multi-agent system perspective. More specifically, we view our contributions as two-fold.

First, we define the *Quantum Branching Distributed Temporal Logic* QBDTL, a significant variant of DTL that we introduce here to represent quantum state transformations in an abstract, *qualitative* way. In QBDTL, we abstract from the value of the qubits: we are not interested in encoding into our system syntactical and semantical information about amplitudes or basis values 0 and 1 (in this way, we avoid any *quantitative* information) and we focus instead on the way qubits evolve by means of unitary transformations. Following DTL's central notion, in QBDTL we do not only consider globally quantum states but also, and in particular, the single unit of information, i.e., we maintain the local perspective of the qubit in the quantum computation.

In other words, in QBDTL each agent represents a distinct qubit, which is the object/subject of computation and which evolves in time by means of quantum transformations and possibly interacts with other agents/qubits.

There is a crucial difference between our QBDTL and the original DTL formulation. DTL is based on linear time life-cycles for agents. In QBDTL, we go beyond linearity and consider branching time since we want to be as general as possible: at each step of the temporal evolution of an agent/qubit, the accessibility relation between worlds in the subtended Kripke-style model aims to capture each possible unitary transformation that can be applied to the qubit. A world (a state in the temporal life-cycle of an agent) represents (an abstraction of) a 1-qubit quantum state. n-ary quantum operators, which act simultaneously on more than one qubit (such as control operators, which play a crucial role in quantum computing), act as communication/synchronization points between agents/qubits.

Second, we give a deduction system $\mathcal{N}$(QBDTL) for QBDTL. In order to deal with all the semantical notions—temporal, quantum and synchronization information—, we follow the style of *labeled deduction* [15, 24, 25], a framework for giving uniform presentations of different non-classical logics, where labels allow one to explicitly encode in the syntax additional information, of a semantic or proof-theoretical nature, that is otherwise implicit in the logic one wants to capture.

In addition to the works on DTL, and in particular the labeled tableaux system given in [5], our starting points for $\mathcal{N}$(QBDTL) are the labeled natural deduction system for the logic *UB* (i.e., the until-free fragment of *CTL*) given in [10] and the approach developed in [19, 20], where a labeled modal deduction system with specific modalities able to describe quantum state transformations is given. Fittingly, in $\mathcal{N}$(QBDTL), we consider composed labels (i, x, q) that represent an agent/qubit i, a time instant x, and the quantum information q in the underlying semantics. A further class of labels is used to represent paths in the life-cycles of the agents.

The rules of $\mathcal{N}$(QBDTL) can then be divided into rules that formalize the *local* temporal evolution of an agent/qubit, and synchronization rules that are, in a sense, *global* as they lift the reasoning from the local perspective of the agent to the *distributed* perspective induced by agent's synchronizations.

It is important to observe that our QBDTL is not a quantum logic. Since the work of Birkhoff and von Neumann [9], various logics have been investigated as a means to formalize reasoning about propositions taking into account the principles of quantum theory, e.g., [11]. In general, it is possible to view quantum logic as a logical axiomatization of quantum theory, which provides an adequate foundation for a theory of reversible quantum processes, e.g., [21, 1–4, 13, 14]. Research has focused also on automated reasoning (e.g., model checking for quantum systems as considered in [16]) and on formal analysis of quantum protocols (e.g., [18]). Our work moves from quite a different point of view, which, to reiterate, is the wish to provide a deduction system able to represent and reason about unitary transformations of quantum states from a temporal multi-agent system perspective and, as will become clear below, thereby provide a basis to reason about other, more complex properties of quantum states such as entanglement.

Organization After a preliminary discussion about aims and motivations of our approach (Section 2), in Section 3 we introduce the logic QBDTL and a DTL-style semantics. In Section 4 we define the natural deduction system $\mathcal{N}$(QBDTL), providing some example derivations, and in Section 5 we state and prove the Soundness Theorem (of $\mathcal{N}$(QBDTL) with respect to the semantics). Section 6 is devoted to discussions about our ongoing and future works.

2 Why Branching Temporal Logic and Synchronization?

In this section, we describe how it is possible to use temporal logic and synchronization rules (the core of the DTL approach) to reason in a simple way about quantum state transformations, whenever one is not interested in the encoding of the mathematical object that represents a quantum state (i.e., a vector in a suitable Hilbert Space) but in the evolution itself as a sequence of transformations and in a notion of synchronization between different quantum bits.

Modal logics are a flexible instrument to describe qualitatively state transformations as they allow one to put the emphasis on the underlying "transition system"—the set of possible worlds of the Kripke semantics and the properties of the accessibility relations between them, which model the dynamical behavior of the system—rather than on the concrete meaning of the internal structures of possible worlds. This intuition was followed in [19, 20], where two pure modal systems were introduced and studied. In such systems, a world represents the abstraction of a quantum state and modal operators reflect general properties of quantum state transformations, since the subtended models are *S5*-models. The accessibility relation between worlds is therefore an equivalence relation, i.e., it enjoys reflexivity, symmetry and transitivity. This captures, in an abstract way, key properties of unary quantum operators: roughly speaking, reflexivity says that the class of the unitary operators includes the identity transformation; symmetry captures reversibility (it is always possible to reverse a quantum transformation, since the inverse operator is easily definable and is unitary); finally, transitivity models algebraic compositionality, i.e., the composition of two or more unitary operators is always a unitary operator [22].

The main difference between the modal systems proposed in [19, 20] and QBDTL is that whereas in the former case a world represents the abstraction of an *arbitrary* quantum state (i.e., a state that describes an arbitrary number n of qubits), in the case of QBDTL we focus on the single qubit and on its transformation by means of *unary* quantum operators and on a notion of local formula built upon a local language. Moreover, we move from a modal to a temporal system: in some sense we "unfold" the accessibility relation between worlds obtaining, for each agent, a tree-like structure that represents the agent's local life-cycle. In this way, we "link" the subtended branching temporal model to the abstract transition system induced by all the unary quantum transformations possibly occurring in each world, which are uniformly modeled in the semantics and in the deduction system by an equivalence relation. Reflexivity, symmetry and transitivity can be plainly expressed in QBDTL: for example, symmetry can be abstractly captured by the labeled formula $(i, x, q) : p \supset \exists\bigcirc\exists\bigcirc p$, where p is a propositional symbol, $\supset$ is implication and $\exists\bigcirc A$ expresses that the formula A is true at the next time instant in some possible future.

A licit question at this stage is what is the meaning of the set of propositional symbols QBDTL formulas are built upon. We maintain an abstract definition of the set (we simply say that is a set of syntactic objects), following the style of DTL and also in the spirit of modal/temporal logic as we discussed above. Then, working with labeled expressions like $(i, x, q) : A$, where the formula A is built by temporal operators, synchronization and propositional symbols, it is not actually crucial to say what propositional symbols stand for.[3] Still, it is important to consider what modal/temporal formulas, possible worlds and the accessibility relation stand for.

One could even choose to instantiate the set of propositional symbols to capture quantitative information about quantum states or general properties that permit one to reason about them. We provide here a simple example (partially related to the examples that we will provide later in Fig. 5). A possible choice is to fix a set of atomic propositions representing mathematical descriptions of the qubit, i.e., a normalized vector in $\mathbb{C}^2$. In other words, given a qubit $a = \alpha|0\rangle + \beta|1\rangle$, the encoding $\ulcorner a \urcorner$ of this mathematical description is an atomic proposition. Let s_i stand for a label (i, x, q), take p as $\ulcorner a \urcorner$ and consider the labeled formula $s_i : p \supset \exists\Box p$ (whose derivation will be given in Fig. 5 and where $\exists\Box p$ expresses that p is true at every time instant in some possible future). This labeled formula can be intuitively interpreted as follows: a (potentially infinite) sequence of identity unitary transformations does not change the mathematical description of the qubit.

Let p still be the encoding $\ulcorner a \urcorner$ of a state $a = \alpha|0\rangle + \beta|1\rangle$ and let us consider again the labeled formula $(i, x, q) : p \supset \exists\bigcirc\exists\bigcirc p$, which fits a peculiar feature of quantum computation, i.e., reversibility. This labeled formula says that: if p holds for i in some state x, then there exists a temporal path such that, in two steps, i reaches a new state in which p still holds (i.e., the mathematical description of such a state is again $\alpha|0\rangle + \beta|1\rangle$). This models the fact that if one transforms a qubit state by means of a unitary operator U, then one can obtain again the same state by applying the adjoint U^* of U, where, in the class of unitary operators, the adjoint corresponds to the inverse U^{-1}, and algebraically, one has $U^*(U(\alpha|0\rangle + \beta|1\rangle)) = U(U^*(\alpha|0\rangle + \beta|1\rangle)) = \alpha|0\rangle + \beta|1\rangle$, i.e., $U^*U = UU^* = I$, where I is the identity operator. Looking for a concrete example, we can take $\alpha = \frac{1}{\sqrt{3}}$ and $\beta = \frac{\sqrt{2}}{\sqrt{3}}$ and instantiate U to X, the complementation gate, which corresponds to an exchange between amplitudes of basis states. Among the temporal

[3] In analogy, note, e.g., that temporal logics developed to deal with concurrent systems do not possess any concurrent feature.

states reachable from x there exists, in particular, the successor state in which $\overline{p}$ and $\exists\bigcirc p$ hold, where $\overline{p} = \ulcorner \frac{\sqrt{2}}{\sqrt{3}}|0\rangle + \frac{1}{\sqrt{3}}|1\rangle \urcorner$.

In quantum computing it is useful to compose small states in order to obtain bigger quantum states (this operation has a precise algebraic meaning, see [22]). Collecting agents, one can model quantum systems of n qubits. In some sense, we can see a quantum state of n qubits as a global state built upon the local states of the single qubits. Each qubit evolves independently but, in a realistic perspective, different qubits do not always evolve asynchronously, and so sometimes they interact, by means of n-ary quantum gates. This is modeled, in our system, by means of ad hoc "tools", properly adapted from DTL: by a special construct in the local language (an operator $\copyright$ named *calling*), it is possible to express the fact that an agent/qubit i synchronizes with another agent/qubit j. This choice has a precise quantum meaning. In quantum computing, one can of course globally modify a set of n qubits by means of n-ary algebraic operators. We view n-ary quantum gates as *synchronization* points between states of different life-cycles, i.e., between states of different qubits. The inputs of an n-ary quantum gate may each have previously been subject to a sequence of other transformations, i.e., in DTL terms, a sequence of events, and the gate itself then can be seen as a transformation event that is shared by the inputs. In this paper, we model this synchronization mechanism abstractly (since, as we said, we model unitary transformations by an equivalence relation), but it is possible to plan a concrete research direction based on the further development of this interpretation of n-ary gates as synchronization mechanisms. See Section 6 for a more detailed discussion of our ongoing and future works.

3 The logic QBDTL

We introduce the Quantum Branching Distributed Temporal Logic QBDTL by presenting its syntax and semantics.

3.1 Syntax

Given a finite set $\mathsf{Id} = \{i, j, \ldots\}$ of *agent identifiers* and a set $\mathsf{Prop} = \{p, p_1, p_2, \ldots\}$ of *atomic propositions* (which characterize the current local states of the agents), we define the *local language* of an agent $i \in \mathsf{Id}$ by the following grammar:

$$\mathcal{L}_i ::= p \mid \bot \mid \mathcal{L}_i \supset \mathcal{L}_i \mid \exists\bigcirc\mathcal{L}_i \mid \exists\Box\mathcal{L}_i \mid \forall\Box\mathcal{L}_i \mid \copyright_j \mathcal{L}_j \,,$$

where $p \in \mathsf{Prop}$ and $j \in \mathsf{Id}$ with $i \neq j$. *Local formulas*, as their names suggest, hold locally for the different agents. $\bot$ is *falsum* and $\supset$ is implication. As in DTL, the communication formula $\copyright_j A$ means that agent i has just communicated (i.e., synchronized) with agent j, for whom A holds. We follow here the Peircean branching temporal logic *UB* [7] and only consider the temporal operators that are obtained as a combination of one single linear-time operator immediately preceded by one single path quantifier. More specifically, we consider here the Peircean operators

- $\exists\bigcirc$ (as we noted previously, $\exists\bigcirc A$ expresses that the formula A in the scope of this operator is true at the next time instant in some possible future),

- $\exists\Box$ ("it is true at every time instant in some possible future") and
- $\forall\Box$ ("it is true at every time instant in every possible future").

For simplicity, in this work we do not consider the temporal operator until, although such an extension would not be problematic. Moreover, as usual, other connectives and temporal operators can be defined as abbreviations.

The *global language* of QBDTL is defined by the grammar:

$$\mathcal{L} ::= @_{i_1}\mathcal{L}_{i_1} \mid \dots \mid @_{i_n}\mathcal{L}_{i_n} ,$$

where $i_1, \dots, i_n \in \mathsf{Id}$. The global formula $@_{i_k}A$ means that A holds for agent i_k.

3.2 Semantics

The models of QBDTL are inspired by those of DTL and built upon a form of Winskel's event structures (cf. [26], where also the relationship to other concurrency models is discussed). There is, however, a fundamental difference with respect to the semantics that has (actually, with respect to the slightly different semantics that in the literature have) been given for DTL, which is based on distributed families of linear life-cycles local to each agent, i.e., countable, discrete and totally ordered local events. Since our logic QBDTL is inherently branching, we need to define its semantics accordingly, and we thus modify DTL's semantics as follows.

Given an agent $i \in \mathsf{Id}$, a *branching local life-cycle* of i is an ω-tree, i.e., a pair $\lambda_i = \langle \mathsf{Ev}_i, <_i \rangle$, where Ev_i is the set of *local events of i* and $<_i \subseteq \mathsf{Ev}_i \times \mathsf{Ev}_i$ is a binary relation such that:

(i) $<_i$ is transitive and irreflexive;
(ii) for each $e \in \mathsf{Ev}_i$, the set $\{e' \in \mathsf{Ev}_i \mid e' <_i e\}$ is linearly ordered by $<_i$;
(iii) there is a $<_i$-smallest element 0_i called the *root* of λ_i;
(iv) each maximal linearly $<_i$-ordered subset of Ev_i is order-isomorphic to the natural numbers.

We write $e \to_i e'$ to denote the fact that e' is an immediate local successor of e, i.e., $e <_i e'$ and there is no e'' such that $e <_i e'' <_i e'$. A $\to_i$*-path* is a sequence of local events $(e_0, \dots, e_n)$ such that $e_k \to_i e_{k+1}$ for $0 \leq k \leq n-1$. An *e-branch b of i* is an infinite $\to_i$-path $b = (e_0, e_1, \dots)$ such that $e = e_0$ and we write $\to_i^b$ to denote the restriction of $\to_i$ to b, i.e., $e' \to_i^b e''$ iff $e' = e_k$ and $e'' = e_{k+1}$ for some k, and denote with $\mathcal{B}_i$ the set of all such $\to_i^b$. Further, we denote with $\to_i^{b*}$ the reflexive and transitive closure of $\to_i^b$.

A *local state* is a finite set $\xi \in \mathsf{Ev}_i$ down-closed for local causality, i.e., if $e <_i e'$ and $e' \in \xi$ then also $e \in \xi$. In general, each non-empty local state ξ is reached by the occurrence of an event that we call $\mathit{last}(\xi)$, from the local state $\xi \setminus \{\mathit{last}(\xi)\}$. Given $e \in \mathsf{Ev}_i$, the set $e{\downarrow}i = \{e' \in \mathsf{Ev}_i | e' \leq_i e\}$, where $\leq_i$ denotes the reflexive closure of $<_i$, is always a local state. Moreover, if ξ is non-empty, then $\mathit{last}(\xi){\downarrow}i = \xi$.

A *branching distributed life-cycle* is a family of local life-cycles

$$\lambda = \{\lambda_i = \langle \mathsf{Ev}_i, <_i \rangle\}_{i \in \mathsf{Id}}$$

such that:

(i) $\leq \, = (\bigcup_{i \in \mathsf{Id}} \leq_i)^*$ defines a partial order of *global causality* on the set of events $\mathsf{Ev} = \bigcup_{i \in \mathsf{Id}} \mathsf{Ev}_i$;

(ii) if $e, e' \in \mathsf{Ev}_i$ and $e \leq e'$ then $e \leq_i e'$.

Condition *(i)* ensures that a distributed life-cycle respects global compatibility, i.e., there is no $e \in \mathsf{Ev}_i \cap \mathsf{Ev}_j$ such that $e <_i e'$ but $e' <_j e$, while condition *(ii)* ensures that synchronization $\leq$-relates two events of an agent i only if there exists a 0_i-branch in which both the events occur.

An *S5 Kripke frame* is a pair $\langle Q, \mathcal{U} \rangle$, where Q is a non-empty set of *qubit states* and $\mathcal{U}$ is a binary equivalence relation on Q, i.e., $\mathcal{U} : Q \to Q$ is reflexive, symmetric and transitive. An *S5 Kripke model* is a triple $\mathcal{M} = \langle Q, \mathcal{U}, \mathcal{V} \rangle$, where $\langle Q, \mathcal{U} \rangle$ is an *S5* Kripke frame and $\mathcal{V} : Q \to \mathcal{P}(\mathsf{Prop})$ is a valuation function assigning to each qubit state in Q a set of atomic propositions.

A QBDTL *model* is a triple $\mu = \langle \lambda, \mathcal{M}, \pi \rangle$, where $\lambda = \{\lambda_i\}_{i \in \mathsf{Id}}$ is a distributed life-cycle, $\mathcal{M} = \langle Q, \mathcal{U}, \mathcal{V} \rangle$ is an *S5* Kripke model and $\pi = \{\pi_i\}_{i \in \mathsf{Id}}$ is a family of local functions associating to each local state a qubit state in Q; for each $i \in \mathsf{Id}$ and set Ξ_i of local states of i, the function $\pi_i : \Xi_i \to Q$ is such that:

(i) if $\xi, \xi' \in \Xi_i$, $last(\xi) \to_i last(\xi')$, $\pi(\xi) = q$ and $\pi(\xi') = q'$, then $q \mathcal{U} q'$;

(ii) if $q, q' \in Q$, $q \mathcal{U} q'$ and $\pi(\xi) = q$, then there exists $\xi' \in \Xi_i$ such that $last(\xi) \to_i last(\xi')$ and $\pi(\xi') = q'$.

In what follows, we denote $\langle \lambda_i, \mathcal{M}, \pi_i \rangle$ by μ_i.

The *global satisfaction relation* is defined by:

$$\models^{\mu} @_i A \text{ iff } \models_i^{\mu_i} A \text{ iff } \models_i^{\mu_i, \xi} A \text{ for every } \xi \in \Xi_i \, ,$$

where the *local satisfaction relation at a local state* ξ *of* i is defined by:

$$\begin{array}{lll}
\not\models_i^{\mu_i,\xi} \bot & & \\
\models_i^{\mu_i,\xi} p & \text{iff} & p \in \mathcal{V}(\pi_i(\xi)), \text{ for } p \in \mathsf{Prop} \\
\models_i^{\mu_i,\xi} A \supset B & \text{iff} & \models_i^{\mu_i,\xi} A \text{ implies } \models_i^{\mu_i,\xi} B \\
\models_i^{\mu_i,\xi} \forall\Box A & \text{iff} & \text{for all } \xi', \, last(\xi) \leq_i last(\xi') \text{ implies } \models_i^{\mu_i,\xi'} A \\
\models_i^{\mu_i,\xi} \exists\Box A & \text{iff} & \text{there exists a } last(\xi)\text{-branch } b \text{ such that for all } \xi', \\
 & & last(\xi) \to_i^{b*} last(\xi') \text{ implies } \models_i^{\mu_i,\xi'} A \\
\models_i^{\mu_i,\xi} \exists\bigcirc A & \text{iff} & \text{there exists } \xi' \text{ such that } last(\xi) \to_i last(\xi') \text{ and } \models_i^{\mu_i,\xi'} A \\
\models_i^{\mu_i,\xi} \copyright_j A & \text{iff} & last(\xi) \in \mathsf{Ev}_j \text{ and } \models_j^{\mu_j, last(\xi)\downarrow j} A
\end{array}$$

By extension, we define:

$$\begin{array}{lll}
\models^{\mu} \Gamma & \text{iff} & \models^{\mu} A \text{ for all } A \in \Gamma \\
\Gamma \models^{\mu} A & \text{iff} & \models^{\mu} \Gamma \text{ implies } \models^{\mu} A \\
\Gamma \models A & \text{iff} & \Gamma \models^{\mu} A \text{ for each QBDTL model } \mu
\end{array}$$

4 A deduction system for QBDTL

4.1 Syntax of the labeled logic

In order to formalize our labeled natural deduction system $\mathcal{N}$(QBDTL), we extend the syntax and semantics of QBDTL by introducing four kinds of *labels* (that represent agents, states, quantum information and paths in the underlying semantics) and by defining labeled and relational formulas.

First of all, we use the agent identifiers in Id as labels. Further, we assume given two fixed denumerable sets of labels Lab_S and Lab_Q. Intuitively, the labels $x, y, z, \ldots$ in Lab_S refer to local states of an agent, whereas the labels $q, q', q_1, \ldots$ in Lab_Q refer to the quantum information concerning an agent.

A *labeled formula* is then a formula of the form

$$(i, x, q) : A\,,$$

where (i, x, q) is a *composed label* with $i \in \mathsf{Id}$, $x \in \mathsf{Lab}_S$ and $q \in \mathsf{Lab}_Q$, and A is a formula in the local language $\mathcal{L}_i$ of the agent i. Note that we do not use the operator @ inside labeled formulas as it is implicitly expressed by the first element of the composed label. For instance, in order to show that a global formula $@_i A$ is valid, we will prove that the labeled formula $(i, x, q) : A$, for arbitrary x and q, is derivable in our system.

In $\mathcal{N}(\mathsf{QBDTL})$, we also need formulas modeling the relation between the states referred by the labels. We thus assume given a further set of labels Lab_B, whose elements will be denoted by $\lhd, \lhd_1, \lhd_2, \ldots$, which intuitively refer to the successor relation between local states in the local life-cycle of an agent i along a given branch.

We define

$$\mathsf{Lab}_B^+ = \mathsf{Lab}_B \cup \{r(i, x, \bigstar A) \mid i \in \mathsf{Id}, x \in \mathsf{Lab}_S, \bigstar \in \{\Box, \bigcirc\}, A \in \mathcal{L}_i\}\,.$$

The labels in $\mathsf{Lab}_B^+ \setminus \mathsf{Lab}_B$ will be used to refer to successor relations between local states along distinct branches. We will write $R, R_1, R_2, \ldots$ to denote generic elements of Lab_B^+ and we will use R^* to refer to the reflexive and transitive closure of R. Finally, we will use the symbol U to refer to the relation modeling unary quantum transformations and the symbol $\bowtie$ to denote that the local states of two agents are synchronized on a given event.

A *relational formula* is then a formula of the form

- $(i, x, q)\ R\ (i, y, q')$, or
- $(i, x, q)\ R^*\ (i, y, q')$, or
- $(i, x, q) \bowtie (j, y, q')$, or
- $q\ U\ q'$,

where $i, j \in \mathsf{Id}$, $x, y \in \mathsf{Lab}_S$, $R \in \mathsf{Lab}_B^+$, $q, q' \in \mathsf{Lab}_Q$. In the following, for simplicity, we will sometimes use metavariables of the form s_i, possibly superscripted, to refer to composed labels of the form (i, x, q).

4.2 Semantics of the labeled logic

In order to give a semantics for our labeled system, we need to define explicitly an interpretation of the labels. Given a QBDTL model μ, an *interpretation function* is a triple $\mathcal{I} = \langle \mathcal{I}_S, \mathcal{I}_Q, \mathcal{I}_B \rangle$, where:

- $\mathcal{I}_S = \{\mathcal{I}_S^i\}_{i \in \mathsf{Id}}$ is a set of functions such that $\mathcal{I}_S^i : \mathsf{Lab}_S \to \Xi_i$ for each $i \in \mathsf{Id}$;
- $\mathcal{I}_Q : \mathsf{Lab}_Q \to Q$;

- $\mathcal{I}_B = \{\mathcal{I}^i_B\}_{i \in \mathsf{Id}}$ is a set of functions such that $\mathcal{I}^i_B : \mathsf{Lab}^+_B \to \mathcal{B}_i$ for each $i \in \mathsf{Id}$, and if $r(i, x, \bigstar A) \in \mathsf{Lab}^+_B \setminus \mathsf{Lab}_B$, then:
  - $\mathcal{I}^i_B(r(i, x, \bigstar A)) = \to^b_i$ for some $\mathcal{I}^i_S(x)$-branch b;
  - if $\models^{\mu, \mathcal{I}^i_S(x)} \exists \bigstar A$, then for all $\xi \in \Xi_i$:
    - if $\bigstar = \bigcirc$, then $\mathit{last}(\mathcal{I}^i_S(x))\ \mathcal{I}^i_B(r(i, x, \bigstar A))\ \mathit{last}(\xi)$ implies $\models^{\mu,\xi} A$;
    - if $\bigstar = \square$, then $\mathit{last}(\mathcal{I}^i_S(x))\ \mathcal{I}^i_B(r(i, x, \bigstar A))^*\ \mathit{last}(\xi)$ implies $\models^{\mu,\xi} A$.

The notion of interpretation allows us to extend the truth relation to labeled formulas, as well as define truth of relational formulas. Given a QBDTL model μ and an interpretation function $\mathcal{I} = \langle \mathcal{I}_S, \mathcal{I}_Q, \mathcal{I}_B \rangle$ on it, *truth for a labeled or relational formula* γ is defined as follows:

$$\begin{array}{lll}
\models^{\mu,\mathcal{I}} (i, x, q) : A & \text{iff} & \mu_i, \mathcal{I}^i_S(x) \models_i A \text{ and } \pi_i(\mathcal{I}^i_S(x)) = \mathcal{I}_Q(q) \\
\models^{\mu,\mathcal{I}} (i, x, q)\ R\ (i, y, q') & \text{iff} & \mathit{last}(\mathcal{I}^i_S(x))\ \mathcal{I}^i_B(R)\ \mathit{last}(\mathcal{I}^i_S(y)), \pi_i(\mathcal{I}^i_S(x)) = \mathcal{I}_Q(q) \text{ and} \\
 & & \pi_i(\mathcal{I}^i_S(y)) = \mathcal{I}_Q(q') \\
\models^{\mu,\mathcal{I}} (i, x, q)\ R^*\ (i, y, q') & \text{iff} & \mathit{last}(\mathcal{I}^i_S(x))\ \mathcal{I}^i_B(R)^*\ \mathit{last}(\mathcal{I}^i_S(y)), \pi_i(\mathcal{I}^i_S(x)) = \mathcal{I}_Q(q) \text{ and} \\
 & & \pi_i(\mathcal{I}^i_S(y)) = \mathcal{I}_Q(q') \\
\models^{\mu,\mathcal{I}} (i, x, q) \bowtie (j, y, q') & \text{iff} & \mathit{last}(\mathcal{I}^i_S(x)) = \mathit{last}(\mathcal{I}^j_S(y)), \pi_i(\mathcal{I}^i_S(x)) = \mathcal{I}_Q(q) \text{ and} \\
 & & \pi_j(\mathcal{I}^j_S(y)) = \mathcal{I}_Q(q') \\
\models^{\mu,\mathcal{I}} q\ U\ q' & \text{iff} & \mathcal{I}_Q(q)\, \mathcal{U}\, \mathcal{I}_Q(q')
\end{array}$$

When $\models^{\mu,\mathcal{I}} \gamma$, for γ a labeled or relational formula, we say that γ is *true* in μ according to $\mathcal{I}$. By extension:

$$\begin{array}{lll}
\models^{\mu,\mathcal{I}} \Gamma & \text{iff} & \models^{\mu,\mathcal{I}} \gamma \text{ for all } \gamma \in \Gamma \\
\Gamma \models^{\mu,\mathcal{I}} \gamma & \text{iff} & \models^{\mu,\mathcal{I}} \Gamma \text{ implies } \models^{\mu,\mathcal{I}} \gamma \\
\models^{\mu} \gamma & \text{iff} & \text{for every interpretation function } \mathcal{I}, \models^{\mu,\mathcal{I}} \gamma \\
\models^{\mu} \Gamma & \text{iff} & \text{for every interpretation function } \mathcal{I}, \models^{\mu,\mathcal{I}} \Gamma \\
\Gamma \models \gamma & \text{iff} & \text{for every } \mathsf{QBDTL} \text{ model } \mathcal{M} \text{ and interpretation function } \mathcal{I}, \Gamma \models^{\mu,\mathcal{I}} \gamma
\end{array}$$

4.3 The rules of $\mathcal{N}$(QBDTL)

The rules of $\mathcal{N}$(QBDTL) are given in Fig. 1–4. We can classify them into four categories: (i) *local life-cycle rules* (inspired to the deduction system for the logic UB given in [10]), (ii) *distributed life-cycle rules* (reminiscent of the global labeled tableaux developed for DTL in [5]), (iii) *quantum transformations rules* (actually a fragment of the deduction systems studied in [20]) and (iv) *interaction rules*.

Local life-cycle rules (Fig. 1) These rules all infer formulas "local" to an agent i, i.e., labeled with s_i. We can divide them further into rules for classical connectives ($\bot E$, $\supset I$ and $\supset E$), rules for temporal operators ($\forall\square I$, $\forall\square E$, $\exists\square I$, $\exists\square E$, $\exists\bigcirc I$ and $\exists\bigcirc E$), relational rules ($\mathit{ser}_\lhd$, ser_{sk}, base_R, $\mathit{lin}_\lhd$, refl_R, trans_R and comp_R) and induction rules ($\mathit{ind}\forall$ and $\mathit{ind}\exists$).

$$\dfrac{\begin{array}{c}[s_i : A \supset \bot]\\ \vdots \\ s_j : \bot\end{array}}{s_i : A}\ \bot E \qquad \dfrac{\begin{array}{c}[s_i : A]\\ \vdots \\ s_i : B\end{array}}{s_i : A \supset B}\ \supset I \qquad \dfrac{s_i : A \supset B \quad s_i : A}{s_i : B}\ \supset E \qquad \dfrac{\begin{array}{c}[s_i \lhd^* s_i']\\ \vdots \\ s_i' : A\end{array}}{s_i : \forall\Box A}\ \forall\Box I \qquad \dfrac{s_i' : \forall\Box A \quad s_i' R^* s_i}{s_i : A}\ \forall\Box E$$

$$\dfrac{\begin{array}{c}[s_i R^* s_i']\\ \vdots \\ s_i' : A\end{array} \quad s_i R s_i''}{s_i : \exists\Box A}\ \exists\Box I \qquad \dfrac{(i,x,q) : \exists\Box A \quad (i,x,q)\, r(i,x,\Box A)^*\, s_i}{s_i : A}\ \exists\Box E$$

$$\dfrac{\begin{array}{c}[s_i R s_i']\\ \vdots \\ s_i' : A\end{array} \quad s_i R s_i''}{s_i : \exists\bigcirc A}\ \exists\bigcirc I \qquad \dfrac{(i,x,q) : \exists\bigcirc A \quad (i,x,q)\, r(i,x,\bigcirc A)\, s_i}{s_i : A}\ \exists\bigcirc E$$

$$\dfrac{\begin{array}{c}[s_j \lhd s_j']\\ \vdots \\ s_i : A\end{array}}{s_i : A}\ ser_{\lhd} \qquad \dfrac{\begin{array}{c}[(j,x,q)\, r(j,x,\bigstar B)\, s_j]\\ \vdots \\ s_i : A\end{array}}{s_i : A}\ ser_{sk} \qquad \dfrac{s_j R s_j' \quad \begin{array}{c}[s_j R^* s_j']\\ \vdots \\ s_i : A\end{array}}{s_i : A}\ base_R \qquad \dfrac{s_i \lhd s_i' \quad s_i \lhd s_i'' \quad s_i' : \alpha}{s_i'' : \alpha}\ lin_{\lhd}$$

$$\dfrac{s_j R s_j' \quad \begin{array}{c}[s_j R^* s_j]\\ \vdots \\ s_i : A\end{array}}{s_i : A}\ refl_R \qquad \dfrac{s_j R^* s_j' \quad s_j' R^* s_j'' \quad \begin{array}{c}[s_j R^* s_j'']\\ \vdots \\ s_i : A\end{array}}{s_i : A}\ trans_R \qquad \dfrac{s_j R_1^* s_j' \quad s_j' R_2^* s_j'' \quad \begin{array}{c}[s_j \lhd^* s_j'']\\ \vdots \\ s_i : A\end{array}}{s_i : A}\ comp_R$$

$$\dfrac{s_i' : A \quad s_i' R^* s_i \quad \begin{array}{c}[s_i' \lhd_1^* s_i'''] \quad [s_i''' \lhd_2 s_i''] \quad [s_i''' : A]\\ \vdots \\ s_i'' : A\end{array}}{s_i : A}\ ind\forall$$

$$\dfrac{(i,x,q) : A \quad (i,x,q)\, r(i,x,\Box A)^*\, s_i \quad \begin{array}{c}[(i,x,q) \lhd^* (i,y,q')] \quad [(i,y,q')\, r(i,y,\bigcirc A)\, s_i'] \quad [(i,y,q') : A]\\ \vdots \\ s_i' : A\end{array}}{s_i : A}\ ind\exists$$

In $\forall\Box I$, $\exists\Box I$ and $\exists\bigcirc I$, where $s_i' \equiv (i,x,q)$, the labels x and q are fresh. Moreover, in $\forall\Box I$, $\lhd$ is fresh.
In $ser_{\lhd}$, where $s_j' \equiv (j,x,q)$, the labels x, q and $\lhd$ are fresh.
In ser_{sk}, where $s_j \equiv (j,y,q')$, the labels y and q' are fresh.
In $comp_R$, $\lhd$ is fresh.
In $ind\forall$, where $s_i'' \equiv (i,x,q)$ and $s_i''' \equiv (i,y,q')$, the labels x, y, q, q', $\lhd_1$ and $\lhd_2$ are fresh.
In $ind\exists$, where $s_i' \equiv (i,z,q'')$, the labels y, z, q', q'' and $\lhd$ are fresh.

Fig. 1. The rules of $\mathcal{N}$(QBDTL): local life-cycle rules

$$\dfrac{s_j : A \quad s_i \bowtie s_j}{s_i : \copyright_j A}\;\copyright I \qquad \dfrac{s_i : \copyright_j A \qquad \begin{matrix}[s_i \bowtie s_j][s_j : A]\\ \vdots \\ s_k : A\end{matrix}}{s_k : A}\;\copyright E$$

$$\dfrac{s_j \bowtie s_k \qquad \begin{matrix}[s_k \bowtie s_j]\\ \vdots \\ s_i : A\end{matrix}}{s_i : A}\;symm_{\bowtie} \qquad \dfrac{s_j \bowtie s_k \quad s_k \bowtie s_l \qquad \begin{matrix}[s_j \bowtie s_l]\\ \vdots \\ s_i : A\end{matrix}}{s_i : A}\;trans_{\bowtie}$$

$$\dfrac{s_i \bowtie s_j \quad s_j R^* s'_j \quad s'_j \bowtie s'_i \qquad \begin{matrix}[s_i \lhd^* s'_i]\\ \vdots \\ s_k : A\end{matrix}}{s_k : A}\;comp_{\bowtie}$$

In $\copyright I$ and $\copyright E$, $i \neq j$. In $\copyright E$, where $s_j \equiv (j, x, q)$, the labels x and q are fresh.
In $comp_{\bowtie}$, $\lhd$ is fresh.

Fig. 2. The rules of $\mathcal{N}$(QBDTL): distributed life-cycle rules

$$\dfrac{\begin{matrix}[q\ U\ q]\\ \vdots \\ s_i : A\end{matrix}}{s_i : A}\;refl_U \qquad \dfrac{q\ U\ q' \qquad \begin{matrix}[q'\ U\ q]\\ \vdots \\ s_i : A\end{matrix}}{s_i : A}\;symm_U$$

$$\dfrac{q\ U\ q' \quad q'\ U\ q'' \qquad \begin{matrix}[q\ U\ q'']\\ \vdots \\ s_i : A\end{matrix}}{s_i : A}\;trans_U \qquad \dfrac{(i, x, q) : p \quad \gamma(j, y, q)}{(j, y, q) : p}\;prop$$

In *prop*, $\gamma(j, y, q)$ is a (labeled or relational) formula where (j, y, q) occurs and $p \in \mathsf{Prop}$ is an atomic proposition.

Fig. 3. The rules of $\mathcal{N}$(QBDTL): quantum transformation rules

Rules for classical connectives. The rule $\perp E$ is a labeled version of *reductio ad absurdum*, where we do not enforce Prawitz's side condition that $A \neq \perp$ and we do not constrain the "world" in which we derive a contradiction to be the same as in the assumption. The rules $\supset I$ and $\supset E$ are the labeled version of the standard [23] natural deduction rules for implication introduction and elimination.

Rules for temporal operators. The rules for the introduction and the elimination of $\forall\Box$, $\exists\Box$ and $\exists\bigcirc$ follow the same structure as the rules for introduction and elimination of $\Box$ in labeled systems for modal logics. Let us consider $\forall\Box I$; the idea is that the meaning of $s_i : \forall\Box A$ is given by the metalevel implication $s_i \lhd^* s'_i \Longrightarrow s'_i : A$ for an arbitrary path

$$\dfrac{q\;U\;q' \qquad \gamma(i,x,q) \qquad \begin{matrix}[(i,x,q) \triangleleft (i,y,q')]\\ \vdots \\ s_j : A\end{matrix}}{s_j : A}\;U{\Rightarrow}R \qquad \dfrac{(i,x,q)\triangleleft(i,y,q') \qquad \begin{matrix}[q\;U\;q']\\ \vdots\\ s_j : A\end{matrix}}{s_j : A}\;R{\Rightarrow}U$$

In $U{\Rightarrow}R$, $\gamma(i,x,q)$ is a (labeled or relational) formula where (i,x,q) occurs. Moreover, y is fresh.

Fig. 4. The rules of $\mathcal{N}$(QBDTL): interaction rules

denoted by the relation $\lhd$ and an arbitrary s'_i $\lhd^*$-accessible from s_i. The arbitrariness, i.e., the*freshness*, of both the path denoted by $\lhd$ and s'_i is ensured by the side-conditions of the rule, e.g., s_i must be different from s_i and not occur in any assumption on which $s'_i : A$ depends other than the discharged assumption $s_i \lhd^* s'_i$.

Introductions of $\exists\Box$ and $\exists\bigcirc$ follow the same principle, but relax the freshness condition on the label denoting the relation, thus allowing us to reason on a single specific path. Note that in this case a further premise $(s_iRs''_i)$ is required: such a premise works as a “witness”, in the sense that it ensures that the relation R considered is indeed a relation passing through the state s_i.

For what concerns the elimination rules, the intuition behind $\forall\Box E$ is that if $\forall\Box A$ holds in a state s'_i and s_i is accessible from s'_i (along some path), then it is possible to conclude that A holds in s_i. The case of $\exists\Box E$ and $\exists\bigcirc E$ is similar but complicated by the fact that the universal linear-time operator ($\Box$ or $\bigcirc$) is preceded by an existential path quantifier ($\exists$), which prevents us from inferring the conclusion for a successor along an arbitrary relation. Our solution is based on the idea (originally proposed in [10]) of using Skolem functions as names for particular relations, e.g., $r(i,x,\Box A)$ denotes a relation passing at x and such that if $\exists\Box A$ holds in x, then A holds at each successor of x along $r(x,\Box A)$.

Relational rules. Relational rules allow for modeling properties of the accessibility relations.[4] The rule $base_R$ expresses the fact that for each relation R, R^* contains R; i.e., $base_R$ says that if (i) s_j is such that there is some R-accessible s'_j and (ii) from the assumption that s'_j is also R^*-accessible from s_j we can infer some labeled formula $s_i : A$ (where s_i might be different from s_j and s'_j), then we can discharge the assumption $s_jR^*s'_j$ and conclude that indeed $s_i : A$ holds. $refl_R$ and $trans_R$ model reflexivity and transitivity of each relation, respectively, whereas $comp_R$ states that it is possible to compose two relations, i.e., if $s_jR^*_1s'_j$ and $s'_jR^*_2s''_j$, then there exists a third relation $\lhd^*$ such that $s_j \lhd^* s''_j$. We also have two rules capturing two different aspects of the seriality of the relations. $ser_\lhd$ captures the fact that, given a state s_j, there is at least a relation passing through s_j and a successor along that relation. ser_{sk} says that, given a state s_j and a Skolem function $r(j,x,\bigstar B)$, there exists a successor of s_j along that relation.

[4] Note that in these rules we use relational formulas as auxiliary formulas in order to derive labeled formulas. Rules treating relational formulas as full-fledged first class formulas, which can be assumed and derived in the rules, could also be defined in the style of [25].

Induction rules. Finally, we have two rules that model the induction principle underlying the relation between R and R^*. This modeling of the induction principle is inspired to the one proposed in [10] and it is reminiscent of deduction systems for Peano Arthimetic. An example of use of the rule $ind\exists$ can be found in Fig. 5, as we discuss below.

Distributed life-cycle rules (Fig. 2) The rules for communication (©I and ©E) follow quite closely the semantics. Consider, e.g., ©I: if agent i in state s_i synchronizes with agent j in state s_j, and A holds for j in that state, then i just communicated with agent j. The rules for synchronization are also quite intuitive, except maybe $comp_{\bowtie}$. Intuitively, $comp_{\bowtie}$ models a notion of compatibility between different synchronizations that involves the same agents and reflects condition (*ii*) in the definition of branching distributed life-cycle.

Quantum transformations rules (Fig. 3) The rules $refl_U$, $symm_U$, $trans_U$ formalize, quite straightforwardly, the reflexivity, symmetry and transitivity of the U relation, in order to uniformly model the class of algebraic unitary operators as an equivalence relation. This captures, in an abstract way, key properties of quantum operators. Roughly speaking: reflexivity says that the class of the unitary operators includes the identity transformation; symmetry captures reversibility (it is always possible to reverse a quantum transformation, since the inverse operator is easily definable and is unitary [22]); finally, transitivity models algebraic compositionality, i.e., the composition of two or more unitary operators is always a unitary operator.

The rule *prop* says that the third element in a composed label fully captures the quantum information contained in a state: thus if two composed labels (i, x, q) and (j, y, q) share the same q, each atomic proposition holding in (i, x, q) must also hold in (j, y, q).

Interaction rules (Fig. 4) The rules $U{\Rightarrow}R$ and $R{\Rightarrow}U$ model the interaction between U and R and express respectively the conditions (i) and (ii) in the definition of function π_i of QBDTL models. More specifically, $U{\Rightarrow}R$ says that if qUq' and the label (i, x, q) occurs in the labeled or relational formula $\gamma(j, x, q)$, then (i, x, q) has a $\lessdot$-successor (i, y, q'); this means that the local state labeled by y is an immediate successor of the state labeled by x in local life-cycle of the agent i, along a given branch. The rule $R{\Rightarrow}U$ captures the fact that if (i, y, q') is a $\lessdot$-successor of (i, x, q) then also the quantum labels q and q' have to be related by U.

4.4 Derivations

Given the rules in Fig. 1–4, the notions of *derivation*, *conclusion*, *open* and *discharged assumption* are the standard ones for natural deduction systems (see, e.g., [17], pp. 127-129). We write

$$\Gamma \vdash_{\mathcal{N}(\mathsf{QBDTL})} (i, x, q) : A$$

$$\dfrac{\dfrac{\dfrac{[s_i : p]^1 \quad [s_i\, r(i,x,\square p)^* s_i^a]^3 \quad \dfrac{\dfrac{\dfrac{[q' U q']^5 \quad [(i,y,q') : p]^4 \quad \dfrac{\dfrac{[(i,y,q') \lhd (i,z,q')]^6 \quad [(i,y,q') \lhd s_i^d]^7 \quad \dfrac{[(i,y,q') : p]^4 \quad [(i,y,q') \lhd (i,z,q')]^6}{(i,z,q') : p}\,\mathit{prop}}{s_i^d : p}\,\mathit{lin}_{\lhd} \quad [(i,y,q') \lhd (i,z,q')]^6}{(i,y,q') : \exists\bigcirc p}\,\exists\bigcirc I^7}{(i,y,q') : \exists\bigcirc p}\,U{\Rightarrow}R^6}{(i,y,q') : \exists\bigcirc p}\,\mathit{refl}_U^5 \quad [(i,y,q')\, r(i,y,\bigcirc p) s_i^c]^4}{s_i^c : p}\,\exists\bigcirc E}{s_i^a : p}\,\mathit{ind}\exists^4 \quad [s_i\, r(i,x,\square p) s_i^b\,]^2}{s_i : \exists\square p}\,\exists\square I^3}{s_i : \exists\square p}\,\mathit{ser}_{sk}{}^2}{s_i : p \supset \exists\square p}\,\supset I^1$$

$$\dfrac{\dfrac{\dfrac{[s_i : \copyright_j \top]^2 \quad \dfrac{\dfrac{\dfrac{\dfrac{[s'_j : \copyright_i \top]^5 \quad \dfrac{\dfrac{[s_i \bowtie s_j]^3 \quad [s_j \lhd^* s'_j]^4 \quad [s'_j \bowtie s'_i]^6 \quad \dfrac{[s_i : \forall\square p]^1 \quad [s_i \lhd_1^* s'_i]^7}{s'_i : p}\,\forall\square E}{s'_i : p}\,\mathit{comp}_{\bowtie}^7 \quad [s'_j \bowtie s'_i]^6}{s'_j : \copyright_i\, p}\,\copyright I}{s'_j : \copyright_i\, p}\,\copyright E^6}{s'_j : \copyright_i \top \supset \copyright_i\, p}\,\supset I^5}{s_j : \forall\square(\copyright_i \top \supset \copyright_i\, p)}\,\forall\square I^4 \quad [s_i \bowtie s_j]^3}{s_i : \copyright_j \forall\square(\copyright_i \top \supset \copyright_i\, p)}\,\copyright I}{s_i : \copyright_j \forall\square(\copyright_i \top \supset \copyright_i\, p)}\,\copyright E^3}{s_i : \copyright_j \top \supset \copyright_j \forall\square(\copyright_i \top \supset \copyright_i\, p)}\,\supset I^2}{s_i : \forall\square p \supset (\copyright_j \top \supset \copyright_j \forall\square(\copyright_i \top \supset \copyright_i\, p))}\,\supset I^1$$

Fig. 5. Example derivations

to say that there exists a derivation of $(i,x,q) : A$ in the system $\mathcal{N}(\mathsf{QBDTL})$ whose open assumptions are all contained in the set of (labeled and relational) formulas Γ. A derivation of $(i,x,q) : A$ in $\mathcal{N}(\mathsf{QBDTL})$ where all the assumptions are discharged is a *proof* of $(i,x,q) : A$ in $\mathcal{N}(\mathsf{QBDTL})$ and we then say that $(i,x,q) : A$ is a *theorem* of $\mathcal{N}(\mathsf{QBDTL})$ and write $\vdash_{\mathcal{N}(\mathsf{QBDTL})} (i,x,q) : A$.

Fig. 5 contains two examples of derivations (actually, proofs). The first is based on the fact that it is always possible to apply the identity transformation to a qubit. It follows that if a qubit is in a state where an atomic proposition p holds, then there exists a path along which p always holds.

The formula derived in the second example describes a property of the synchronization between qubits and can be read as a consequence of condition (*ii*) in the definition of a distributed life-cycle. If the qubit i is in a state from which a proposition p always holds in the future, then if i synchronizes with j, i.e., the two qubits are combined by means of some n-ary quantum operator, and after that, j synchronizes with i again, we end up in a state of i where p still holds. Note that in this derivation we use the *verum* $\top$ as an abbreviation for $\bot \supset \bot$.

5 Soundness

$\mathcal{N}(\mathsf{QBDTL})$ is sound with respect to the given semantics.

Theorem 1 (Soundness). *For every set Γ of labeled and relational formulas and every labeled formula $(i, x, q) : A$, it holds that $\Gamma \vdash_{\mathcal{N}(\mathsf{QBDTL})} (i, x, q) : A \;\Rightarrow\; \Gamma \models (i, x, q) : A$.*

This theorem can be shown by adapting standard proof techniques for labeled natural deduction systems [25]. The proof proceeds by induction on the structure of the derivation of $(i, x, q) : A$. The base case is when $(i, x, q) : A \in \Gamma$ and is trivial. There is one step case for every rule (where, for what concerns local life-cycle rules, we refer to [10], whose treatment can be quite easily adapted to work here). We show a few representative step cases.

Consider the case when the last rule applied is *prop*:

$$\dfrac{\begin{array}{c}\Pi_1\\ (i, x, q) : p\end{array}\quad \begin{array}{c}\Pi_2\\ \gamma(j, y, q)\end{array}}{(j, y, q) : p}\ \mathit{prop}$$

where Π_1 is a proof of $(i, x, q) : p$ from hypotheses in Γ_1 and Π_2 is a proof of $\gamma(j, y, q)$ from hypotheses in Γ_2, for some sets Γ_1, Γ_2 of formulas. By the induction hypothesis, for each model $\mu = \langle \lambda, \mathcal{M}, \pi \rangle$ and interpretation function $\mathcal{I}$, if $\models^{\mu,\mathcal{I}} \Gamma_1$ then $\models^{\mu,\mathcal{I}} (i, x, q) : p$ and if $\models^{\mu,\mathcal{I}} \Gamma_2$ then $\models^{\mu,\mathcal{I}} \gamma(j, y, q)$. We consider an $\mathcal{I}$ and a μ such that $\models^{\mu,\mathcal{I}} \Gamma = \Gamma_1 \cup \Gamma_2$, and show that $\models^{\mu,\mathcal{I}} (j, y, q) : p$. As a consequence of the induction hypothesis, we get: *(i)* $\models_i^{\mu_i,\mathcal{I}_S^i(x)} p$; *(ii)* $\pi_i(\mathcal{I}_S^i(x)) = \mathcal{I}_Q(q)$; and *(iii)* $\pi_j(\mathcal{I}_S^j(y)) = \mathcal{I}_Q(q)$. It follows from (i) that $p \in \mathcal{V}(\pi_i(\mathcal{I}_S^i(x)))$, i.e., by (ii), $p \in \mathcal{V}(\mathcal{I}_Q(q))$ and, by (iii), $p \in \mathcal{V}(\pi_j(\mathcal{I}_S^j(y)))$. By definition, we have $\models^{\mu,\mathcal{I}_S^j(y)} p$, from which we infer $\models^{\mu,\mathcal{I}} (j, y, q) : p$.

Now consider the case of an application of $\copyright I$:

$$\dfrac{\begin{array}{c}\Pi\\ (j, y, q') : A\end{array}\quad (i, x, q) \bowtie (j, y, q')}{(i, x, q) : \copyright_j A}\ \copyright I$$

where Π is a proof of $(j, y, q') : A$ from hypotheses in Γ_1. By the induction hypothesis, we have $\Gamma_1 \models (j, y, q') : A$. We want to show that $\Gamma = \Gamma_1 \cup \{(i, x, q) \bowtie (j, y, q')\} \models (i, x, q) : \copyright_j A$. Let us consider an arbitrary QBDTL model $\mu = \langle \lambda, \mathcal{M}, \pi \rangle$ and an interpretation function $\mathcal{I}$, and assume that $\models^{\mu,\mathcal{I}} \Gamma$ holds. This implies $\mathit{last}(\mathcal{I}_S^i(x)) = \mathit{last}(\mathcal{I}_S^j(y))$ and $\pi_i(\mathcal{I}_S^i(x)) = \mathcal{I}_Q(q)$. By the induction hypothesis, we also obtain $\models^{\mu,\mathcal{I}} (j, y, q') : A$, which yields $\models_j^{\mu_j,\mathcal{I}_S^j(y)} A$. By the definition of local satisfaction relation, we infer $\models_i^{\mu_i,\mathcal{I}_S^i(x)} \copyright_j A$ and then $\models^{\mu,\mathcal{I}} (i, x, q) : \copyright_j A$. Since μ and $\mathcal{I}$ are arbitrary, we can conclude $\Gamma \models (i, x, q) : \copyright_j A$.

Finally, consider the case of an application of $R{\Rightarrow}U$:

$$\dfrac{(i, y, q') \lhd (i, z, q'') \quad \begin{array}{c}[q'Uq'']\\ \Pi\\ (j, x, q) : A\end{array}}{(j, x, q) : A}\ R{\Rightarrow}U$$

where Π is a proof of $(j, x, q) : A$ from hypotheses in $\Gamma_1 \cup \{q' U q''\}$ for some set Γ_1 of formulas. By the induction hypothesis, we have $\Gamma_1 \cup \{q' U q''\} \models (j, x, q) : A$. We want to show that $\Gamma = \Gamma_1 \cup \{(i, y, q') \lhd (i, z, q'')\} \models (j, x, q) : A$. Let us consider arbitrary $\mu = \langle \lambda, \mathcal{M}, \pi \rangle$ and $\mathcal{I}$, and assume that $\models^{\mu,\mathcal{I}} \Gamma$ holds. This implies $\models^{\mu,\mathcal{I}} (i, y, q') \lhd (i, z, q'')$, from which we infer: $last(\mathcal{I}^i_S(y))\ \mathcal{I}^i_B(\lhd)\ last(\mathcal{I}^i_S(z))$; $\pi_i(\mathcal{I}^i_S(y)) = \mathcal{I}_Q(q')$; and $\pi_i(\mathcal{I}^i_S(z)) = \mathcal{I}_Q(q'')$. By condition ($i$) in the definition of a QBDTL model, this yields $\mathcal{I}_Q(q')\mathcal{U}\mathcal{I}_Q(q'')$ and thus $\models^{\mu,\mathcal{I}} q' U q''$. By the induction hypothesis, we obtain $\models^{\mu,\mathcal{I}} (j, x, q) : A$. Since μ and $\mathcal{I}$ are arbitrary, we can conclude $\Gamma \models (j, x, q) : A$.

6 Concluding Remarks

We have proved that the system $\mathcal{N}$(QBDTL) is sound with respect to the given semantics. We expect $\mathcal{N}$(QBDTL) to be also complete, since it is "built" by composing subsystems that are complete with respect to the semantics of the sublogics that they capture [10, 5, 20], with the addition of rules tightly related to the interactions between those subsystems. A thorough proof, however, requires a non-trivial use of refining techniques to get appropriate models from those obtained by a standard canonical-model construction, similarly to what happens with related temporal logics. We have thus left it for future work.

We are also working at extending QBDTL in order to deal with peculiar properties of quantum states such as *entanglement*. Roughly speaking, in physics, an *entangled* state is a quantum state where two or more qubits behave as connected, independently of their real physical distance. As a consequence, operations on an entangled qubit can (possibly) have side-effects on other entangled qubits. This phenomenon (that does not have a classical counterpart) plays a major role in quantum computing (see, e.g., the teleportation protocol [22]).

In this paper, we have modeled quantum state transformations from an abstract perspective: in QBDTL, no reference to a specific quantum computation or to a notion of input/output of values is required. This allowed us to design a manageable high-level formalization oriented to modeling the behavior of quantum systems, but it is probably not enough if one wants to capture more complex properties such as entanglement. This does not mean that one has to completely convert the qualitative approach into a quantitative one (following the "philosophy" of quantum logic, cf. the discussion in the introduction). We believe that a distributed logic is a promising tool not only for the simple description of quantum states, but also to model the correct amount of quantitative information needed to capture properties like entanglement. In some sense, we aim at integrating into the QBDTL high-level perspective, able to model the "control" of quantum computation (which treats qubits and quantum gates as black-boxes), more detailed information about quantum data, so that it is possible to "look inside" the qubits and specifically model the quantitative behavior of some interesting unitary operators.

In QBDTL, we are as general as possible with respect to the application of transformations: in a local-life cycle the subtended temporal transition tree represents at each step all the possible unary gates that can be applied to the current state, while the synchronization mechanism between agents models all possible n-ary operators. It is well known that one can fix a complete computational basis (finite or infinite) of unitary operators and represent other operators in terms of the elements of such a basis. An infinite complete basis can be defined by taking all unary operators and a particular binary quantum gate called *controlled-not* (or *cnot*). Intuitively, the cnot acts as follows: it takes in input two distinct qubits and complements the *target* qubit (the second one) if the *control* qubit (the first one) is different from 0; otherwise, it does not perform any action. Noticeably, when the control qubit assumes some specific superpositional value (e.g., $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$), the output of the cnot is an entangled state. This suggests that restricting our perspective about arbitrary n-ary gates as synchronization operators to a single binary gate, the cnot, and lifting syntax and semantics to capture its behavior would provide us with all the ingredients needed to model entanglement. Following this standpoint, one can now view synchronizations exactly as control operators: a target qubit has to synchronize (by sharing an event) with the control qubit in order to perform its own, controlled evolution. Moreover, we observe that the notion of synchronization, in presence of entanglement, assumes a non-local (with respect to time) meaning: a synchronization that creates entanglement does not only represent the sharing of local events, but it also influences the subsequent events in the local life cycle of the involved agents. We thus aim to make the connection between agent synchronization and (possible) entanglement of qubits explicit.

Finally, we are considering the explicit modeling inside QBDTL of measurement steps, which can be seen as a further class of transformations. We believe that these extensions will also enable the use of our approach for the analysis of quantum security protocols, which are based on entanglement phenomena [8], along the lines of what has been done with respect to classical security protocols by using DTL [6].

References

1. S. Abramsky and R. Duncan. A categorical quantum logic. *Math. Structures Comput. Sci.*, 16(3):469–489, 2006.
2. A. Baltag and S. Smets. The logic of quantum programs. In *Proceedings of the 2nd QPL*, 2004.
3. A. Baltag and S. Smets. LQP: the dynamic logic of quantum information. *Math. Structures Comput. Sci.*, 16(3):491–525, 2006.
4. A. Baltag and S. Smets. Quantum logic as a dynamic logic. *Synthese*, 179(2):285–306, 2011.
5. D. Basin, C. Caleiro, J. Ramos, and L. Viganò. Labelled tableaux for distributed temporal logic. *J. Log. and Comput.*, 19(6):1245–1279, Dec. 2009.
6. D. Basin, C. Caleiro, J. Ramos, and L. Viganò. Distributed Temporal Logic for the Analysis of Security Protocol Models. *Theor. Comput. Sci.*, 412(31):4007–4043, July 2011.
7. M. Ben-Ari, Z. Manna, and A. Pnueli. The temporal logic of branching time. In *Proceedings of POPL*. ACM Press, 1981.
8. C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing*, pages 175–179, 1984.
9. G. Birkhoff and J. von Neumann. The logic of quantum mechanics. *Ann. of Math. (2)*, 37(4):823–843, 1936.
10. C. Caleiro, L. Viganò, and M. Volpe. A Labeled Deduction System for the Logic UB. In *Proceedings of TIME*. IEEE CS Press, 2013.

11. M. L. Dalla Chiara. Quantum logic. In *Handbook of Philosophical Logic III*: 427–469. Reidel, 1986.
12. H.-D. Ehrich and C. Caleiro. Specifying communication in distributed information systems. *Acta Informatica*, 36:591–616, 2000.
13. K. Engesser, D. M. Gabbay, and D. Lehmann. *A New Approach to Quantum Logic*. College Publications, 2007.
14. K. Engesser, D. M. Gabbay, and D. Lehmann, editors. *Handbook of Quantum Logic and Quantum Structures*. Elsevier, 2009.
15. D. M. Gabbay. *Labelled Deductive Systems*, volume 1. Clarendon Press, 1996.
16. S. J. Gay, R. Nagarajan, and N. Papanikolaou. QMC: A Model Checker for Quantum Systems. In *Proceedings of CAV*, LNCS 5213. Springer, 2008.
17. J.-Y. Girard. *Proof theory and logical complexity. Vol. 1*. Bibliopolis, 1987.
18. P. Kouvaros and A. Lomuscio. Automatic verification of parameterised multi-agent systems. In *Proceedings of AAMAS*, 2013.
19. A. Masini, L. Viganò, and M. Zorzi. A qualitative modal representation of quantum state transformations. In *Proceedings of the 38th ISMVL*. IEEE CS Press, 2008.
20. A. Masini, L. Viganò, and M. Zorzi. Modal Deduction Systems for Quantum State Transformations. *Multiple-Valued Logic and Soft Computing*, 17(5-6):475–519, 2011.
21. P. Mittelstaedt. The modal logic of quantum logic. *J. Philos. Logic*, 8(4):479–504, 1979.
22. M. Nielsen and I. Chuang. *Quantum computation and quantum information*. Cambridge University Press, 2000.
23. D. Prawitz. *Natural Deduction: a Proof-Theoretical Study*. Almquist and Wiskell, 1965.
24. A. K. Simpson. *The Proof Theory and Semantics of Intuitionistic Modal Logic*. PhD thesis, School of Informatics, University of Edinburgh, 1994.
25. L. Viganò. *Labelled Non-Classical Logics*. Kluwer Academic Publishers, 2000.
26. G. Winskel and M. Nielsen. Event structures. In *Handbook of Logic in Computer Science IV*. Oxford University Press, 1995.