Non-Determinism, Non-Termination and the Strong Normalization of System T

Federico Aschieri[1] and Margherita Zorzi[2]

[1] École Plume, LIP (UMR 5668), École Normale Supérieure de Lyon, France
[2] Università degli Studi di Verona, Italy

Abstract. We consider a de'Liguoro-Piperno-style extension of the pure lambda calculus with a non-deterministic choice operator as well as a non-deterministic iterator construct, with the aim of studying its normalization properties. We provide a simple characterization of non-strongly normalizable terms by means of the so called "zoom-in" perpetual reduction strategy. We then show that this characterization implies the strong normalization of the simply typed version of the calculus. As straightforward corollary of these results we obtain a new proof of strong normalization of Gödel's System T by a simple translation of this latter system into the former.

1 Introduction

The idea of defining the concept of redundancy or *detour* in an arithmetical proof [17] and the result that shows the possibility of eliminating all the detours in any proof, are real cornerstones of modern logic. Such results, known under the name of *normalization* or strong normalization, are interesting for a great deal of reasons. For example:

– Via the Curry-Howard correspondence, they can be translated as proofs of the termination of programs written in typed system like Gödel's T [6], Spector's B [19] or Girard's F [6]. Indeed, this is now the standard way of presenting normalization results.
– They are tools for proving consistency of logical systems and thus give rise, in the classical case, to Tarski models (see for example [12]).
– Many intuitionistic ([10]) and classical realizabilities ([12]), as well as functional interpretations [19], are built on ideas coming from normalization techniques.

Unfortunately, proving normalization properties of strong logical systems is difficult and when one succeeds, the resulting proof is often of little combinatorial information. This is of course due to the famous Gödel incompleteness theorems, which force normalization proofs to employ powerful mathematical methods.

In the case of the strong normalization of Gödel's System T, the most flexible and elegant proof is due to Tait (see [6]), which uses the abstract concept of reducibility. In our opinion, there are at least two reasons why the proof is not very intuitive from the combinatorial point of view. First, the predicates of reducibility are defined by formulas with arbitrarily many nested quantifiers. This is a strength, because, to put it as Girard [6], "the deep reason why reducibility works where combinatorial intuition fails, is its logical complexity" . However, this complexity also hampers a concrete understanding of the normalization process and, in fact, condemns combinatorial intuition to failure. Secondly, reducibility is in reality an instance of much more general techniques that can be used for proving a variety of results (for example, weak Church-Rosser property) in an elegant way. We are of course referring to logical relations and realizability. This is evident in Krivine's work [11], where realizability is carried out as a generalization of the Tait-Girard methods. Thus reducibility appears not to be tailored specifically for normalization problems (this observation can also be addressed to Sanchis' technique [18], which allows to reason in a well-founded way about terms of System T, and can be exploited to prove strong normalization).

Among other known normalization techniques one finds the one using infinite terms of Tait [22], more interesting combinatorially, but not suitable to prove strong normalization, the one of Gandy [8], the one of Joachimiski-Matthes [14], similar in spirit to that of Sanchis, and the one using ordinal analysis of Howard [7].

In this paper, we return to the problem of the strong normalization of Gödel's T, with the aim of better understanding its combinatorial structure. That is, we want to provide a *concrete* normalization proof instead of an *abstract* one. In particular, we show how strong normalization can be derived by just examining the terms produced by a simple reduction strategy. For this purpose, we start from some combinatorial ideas, due essentially to Van Daalen [20] and Levy [13] (but also present in Nederpelt [16]) and extended later by David and Nour to various systems of simple types [4]. These ideas inspired Melliès [15] to define a perpetual reduction strategy, so called zoom-in (discovered independently by Plaisted, Sorensen and Gramlich, see [9] for more details), which will be the heart of our method. In [1] (see also [2]), the zoom-in strategy has been employed to characterize non-strongly normalizable lambda terms, and derive as corollary the strong normalization of the simply typed lambda calculus and the intersection types. These latter results were obtained also by Melliès and David. The novelty in [1] consisted in the explicit statement of a characterization theorem. If with $t_1 \ldots t_2$ we denote any term of the form $(((u)u_1)\ldots)u_n$, where $u = t_1$ and $u_n = t_2$, then it is proved that:

Theorem 1 (Characterization of non-strongly normalizable terms). *Let u be an non-strongly normalizable lambda term. Then there exists an infinite reduction $u_1, u_2, \ldots, u_n, \ldots$ and an infinite sequence of terms $t_1, t_2, \ldots, t_n, \ldots$ such that $u_1 = u$ and for every i, u_i contains a subterm of the form $t_i \ldots t_{i+1}$.*

(for a more detailed formulation, see section §2). Notice how the strong normalization easily follows from the Characterization Theorem: non-strongly normalizable Church-typed lambda terms cannot exist, otherwise the type of each t_i would strictly contain the type of t_{i+1}. Remark also how each one of these terms t_i dynamically passes from the status of argument to the status of function applied to some other arguments: this is the crucial property of the reduction.

One natural question was then whether the Characterization Theorem could be extended to a pure lambda calculus with pairs and constants, containing at least booleans, numerals, the if-then-else if and iterator It constructs. In such a way, one would also obtain as corollary the strong normalization of its typed version, i.e. System T. Unfortunately, the Characterization Theorem does not extend so easily. The reason is that one passes from a pure, functional world – the lambda calculus – to an impure world in which booleans and numerals are treated as basic objects, but also retain a sort of functional behavior.

For example, one has a rule $\mathsf{if}\,\mathtt{False}\,u\,v \mapsto v$. But what would be the difference with a hypothetical reduction $\mathtt{False}\,u\,v \mapsto v$, in which $\mathtt{False}$ would behave as the encoding of false in lambda calculus $\lambda x\lambda y.\,y$? Syntactically, if is treated just as a placeholder, being the boolean $\mathtt{False}$ the one which comes makes the real job. Similarly, one has a rule $\mathsf{It}\,u\,v\,\overline{2} \mapsto (v)(v)u$. But what would be the difference with a hypothetical reduction $\overline{2}uv \mapsto (v)(v)u$, in which $\overline{2}$ would behave as the Church-numeral two $\lambda f\lambda x.\,(f)(f)x$?

As a consequence of the use of objects as "hidden" functionals, one loses the Characterization Theorem: when one of the t_i above is, say, $\mathtt{False}$ or $\overline{2}$, we cannot expect it to pass from argument to head position in any meaningful way. The solution to this issue is radical: remove reduction rules involving booleans and numerals and simulate them with actual functionals. The idea is to use *non-determinism.* As in de' Liguoro and Piperno [5], we add to lambda-calculus a non-deterministic choice operator $\mathsf{if}^\star$, with rules $\mathsf{if}^\star\,u\,v \mapsto u$ and $\mathsf{if}^\star\,u\,v \mapsto v$, in order to simulate all possible if reductions. We also add a non-deterministic iterator operator $\mathsf{It}^\star$, with rules of the form $\mathsf{It}^\star\,u\,v \mapsto (v)\ldots(v)u$ (one for each possible number of occurrences of v), in order to simulate all possible It reductions. We obtain as a result a non-deterministic lambda calculus $\Lambda^\star$ which enjoys the Characterization Theorem; its typed version $\mathsf{T}^\star$ will thus have the strong normalization property. We shall then prove strong normalization of System T by translating it into $\mathsf{T}^\star$ – almost trivially. It will be enough to substitute the normal versions of if and It with their non-deterministic counterparts $\mathsf{if}^\star$ and $\mathsf{It}^\star$.

Plan of the Paper. In Section §2 we introduce the non-deterministic lambda calculus $\Lambda^\star$ and prove the Characterization Theorem of its non-strongly normalizable terms. In Section §3, as a corollary, we prove the strong normalization of the non-deterministic typed system $\mathsf{T}^\star$. Section §4 is finally devoted to the proof of the strong normalization of Gödel's System T, by translation into $\mathsf{T}^\star$.

2 The Non-Deterministic Lambda Calculus $\Lambda^\star$ with Pairs and Constants

In this section we define and study the non-deterministic lambda calculus $\Lambda^\star$, whose typed version will serve in section §4 to interpret Gödel's System T. In particular, we are going to give a syntactical characterization of the non-strongly normalizable terms of $\Lambda^\star$.

The non-deterministic lambda calculus $\Lambda^\star$ is formally described in Figure 1. Its deterministic part is a standard lambda calculus (for which we refer to [11]) augmented with pairs, projections, and some arbitrary set of constants $\mathsf{c}_0, \mathsf{c}_1, \ldots$ without any associated reduction rule. In this latter set, one will typically put 0, S, $\mathtt{True}$, $\mathtt{False}$, but no assumption will be made in this section. The non-deterministic part of $\Lambda^\star$ comprises as constants the non-deterministic choice operator $\mathsf{if}^\star$, as in de' Liguoro-Piperno [5], Dal Lago-Zorzi [3], and the non-deterministic iterator $\mathsf{It}^\star$. For $\mathsf{It}^\star\, u\, v$ one has denumerably many possible reductions:

$$\mathsf{It}^\star\, u\, v \mapsto u, \quad \mathsf{It}^\star\, u\, v \mapsto (v)u, \quad \mathsf{It}^\star\, u\, v \mapsto (v)(v)u, \quad \mathsf{It}^\star\, u\, v \mapsto (v)(v)(v)u \ \ldots$$

We point out that, as remarked in [5], $\mathsf{It}^\star$ can already be defined by $\mathsf{if}^\star$. But that is no longer possible in a typed setting, and so we had to leave $\mathsf{It}^\star$ in the syntax.

We now recall some very basic facts and definitions. We retain the Krivine parenthesis convention for pure lambda calculus and extend it to $\Lambda^\star$. The term $(t)u$ will be written as tu and $[u]\pi_i$ as $u\pi_i$, if there is no ambiguity. Thus every lambda term t can be uniquely written in the form $\lambda x_1 \ldots \lambda x_m.\ vt_1 \ldots t_n$, where $m, n \geq 0$, for every i, t_i is either a term or the symbol π_0 or π_1, and v is a variable or a constant or pair $\langle t, u\rangle$ or one of the following redexes: $(\lambda x.u)t$, $\mathsf{if}^\star\, t\, u$, $\mathsf{It}^\star\, t\, u$, $[\langle t, u\rangle]\pi_i$. If v is a redex, v is called the head redex of t. A term is said to be an application if it is of the form tu, an abstraction if it is of the form λxu. If t' is a subterm of t we will write $t \trianglerighteq t'$, reading t *contains* t'. Finally:

Definition 1 (Strongly Normalizable Terms). *We write $t \mapsto t'$ iff t' is obtained from t by contracting a redex in t according to the reduction rules in Figure 1. A sequence (finite or infinite) of terms $t_1, t_2, \ldots, t_n, \ldots$ is said to be a reduction of t, if $t = t_1$, and for all i, $t_i \mapsto t_{i+1}$. A term t of $\Lambda^\star$ is strongly normalizable if there is no infinite reduction of t. We denote with SN the set of strongly normalizable terms of $\Lambda^\star$.*

The reduction tree of a strongly normalizable lambda term is well-founded. It is well-known that it is possible to assign to each node of a well-founded tree an ordinal number, that it decreases passing from a node to any of its sons. We will call the *ordinal size* of a lambda term $t \in \mathsf{SN}$ the ordinal number assigned to the root of its reduction tree and we denote it by $h(t)$; thus, if $t \mapsto u$, then $h(t) > h(u)$. To fix ideas, one may define $h(t) := \mathsf{sup}\{h(u) + 1 \mid t \mapsto u\}$.

Constants $\mathsf{c} ::= \mathsf{It}^\star \mid \mathsf{if}^\star \mid \mathsf{c}_0 \mid \mathsf{c}_1 \ldots$

Terms $t, u ::= x \mid \lambda x.t \mid (t)u \mid \langle t, u\rangle \mid [t]\pi_0 \mid [t]\pi_1 \mid \mathsf{c}$

Reduction Rules

$$(\lambda x.u)t \mapsto u[t/x] \qquad [\langle u_0, u_1\rangle]\pi_i \mapsto u_i, \text{ for i=0,1}$$

$$\mathsf{if}^\star\, u\, v \mapsto u \qquad \mathsf{if}^\star\, u\, v \mapsto v$$

$$\mathsf{It}^\star uv \mapsto \overbrace{(v)\ldots(v)}^{n\ times}\, u, \qquad \text{for each natural number } n$$

Fig. 1. Non-Deterministic Lambda Calculus $\Lambda^\star$

2.1 The zoom-in reduction

In order to really understand the phenomenon of non-termination in lambda calculus it is crucial to isolate the mechanisms that are really essential to produce it. For example, in the term $(\lambda y.\, y)(\lambda x.\, xx)\lambda x.\, xx$ (beware Krivine's notation!), the part that generates an infinite reduction is $(\lambda x.\, xx)\lambda x.\, xx$; the term $\lambda y.\, y$ is only a disturbing context and should be ignored. This is because the smallest non-strongly normalizing subterm of our term is $(\lambda x.\, xx)\lambda x.\, xx$. We thus arrive at the notion of elementary term: a non-strongly normalizable term that cannot be decomposed into smaller non-strongly normalizable terms.

Definition 2 (Elementary Terms). *A term tu is said to be* elementary *if $t \in \mathsf{SN}$, $u \in \mathsf{SN}$ and $tu \notin \mathsf{SN}$.*

We observe that an elementary term cannot be of the form $xt_1 \dots\ t_n$, since $t_1, \dots, t_n \in \mathsf{SN}$, and hence $xt_1 \dots\ t_n \in \mathsf{SN}$. Similarly, it cannot be neither of the form $\mathsf{c}_i t_1 \dots t_n$ nor $(\langle t, u\rangle)t_1 \dots t_n$ nor $[\lambda xt]\pi_i t_1 \dots\ t_n$. Therefore, every elementary lambda term is either of the form $(\lambda xu)tt_1 \dots t_n$ or $\mathsf{if}^\star t\, u\, t_1 \dots t_n$ or $\mathsf{It}^\star t\, u\, t_1 \dots t_n$ or $[\langle t, u\rangle]\pi_i t_1 \dots t_n$ (and clearly $u, t, t_1, \dots t_n \in \mathsf{SN}$).

Proposition 1. *Suppose $v \notin \mathsf{SN}$. Then v has an elementary subterm.*

Proof. By induction on v.

- If $v = x$ or $v = \mathsf{c}$, it is trivially true.
- If $v = ut$, and $u \in \mathsf{SN}$ and $t \in \mathsf{SN}$, v is elementary; if instead $u \notin \mathsf{SN}$ or $t \notin \mathsf{SN}$, by induction hypothesis u or t contains an elementary subterm, and hence v.
- If $v = \lambda xu$ or $v = u\pi_i$, then $u \notin \mathsf{SN}$, and by induction hypothesis u contains an elementary subterm, and thus also v.
- If $v = \langle t, u\rangle$, then $t \notin \mathsf{SN}$ or $u \notin \mathsf{SN}$, and by induction hypothesis u contains an elementary subterm, and thus also v.

The next proposition tells that it is always possible to contract the head redex of an elementary term in such a way to preserve its property of being non-strongly normalizable.

Proposition 2 (Saturation). *Suppose that v is elementary. Then:*

1. *If $v = (\lambda xu)tt_1 \dots t_n \notin \mathsf{SN}$, then $u[t/x]t_1 \dots t_n \notin \mathsf{SN}$.*
2. *If $v = [\langle u_0, u_1\rangle]\pi_i t_1 \dots t_n \notin \mathsf{SN}$, then $u_i t_1 \dots t_n \notin \mathsf{SN}$.*
3. *If $v = \mathsf{if}^\star t\, u\, t_1 \dots t_n \notin \mathsf{SN}$, then $ut_1 \dots t_n \notin \mathsf{SN}$ or $tt_1 \dots t_n \notin \mathsf{SN}$*
4. *If $v = \mathsf{It}^\star t\, u\, t_1 \dots t_n \notin \mathsf{SN}$, then for some $m \in \mathbb{N}$, $(\overbrace{(u) \dots (u)}^{m\ times}\, t)t_1 \dots t_n \notin \mathsf{SN}$.*

Proof.

1. By lexicographic induction on the $(n+2)$-tuple $(h(u), h(t), h(t_1), \dots, h(t_n))$. Since by hypothesis $(\lambda xu)tt_1 \dots t_n \notin \mathsf{SN}$, there exists a $w \notin \mathsf{SN}$ such that $(\lambda xu)tt_1 \dots t_n \mapsto w$. There are two cases. First case: w is either $(\lambda xu')tt_1 \dots t_n$ or $(\lambda xu)t't_1 \dots t_n$ or $(\lambda xu)tt_1 \dots t'_i \dots t_n$ with $u \mapsto u'$, $t \mapsto t'$ and $t_i \mapsto t'_i$ $(i = 1, \dots n)$ respectively. We have $h(u') < h(u), h(t') < h(t), h(t'_i) < h(t_i)$. Then, by induction hypothesis, $u[t/x]t_1 \dots t_n \mapsto u'[t/x]t_1 \dots t_n \notin \mathsf{SN}$, $u[t/x]t_1 \dots t_n \mapsto u[t'/x]t_1 \dots t_n \notin \mathsf{SN}$ and
$u[t/x]t_1 \dots t_i \dots t_n \mapsto u[t/x]t_1 \dots t'_i \dots t_n \notin \mathsf{SN}$ $(i = 1 \dots n)$. Second case: $w = u[t/x]t_1 \dots t_n$. We conclude $u[t/x]t_1 \dots t_n \notin \mathsf{SN}$ by hypothesis on w.

2. The other cases are similar.

Let $v \notin \mathsf{SN}$ and s be an elementary subterm of v. Then $s = (\lambda x u)t t_1 \dots t_n$ or $s = \mathsf{if}^{\star} t\, u\, t_1 \dots t_n$ or $s = \mathsf{It}^{\star} t\, u\, t_1 \dots t_n$ or $s = [\langle t, u\rangle]\pi_i t_1 \dots t_n$. By Proposition 2, there exists an $s' \notin \mathsf{SN}$ such that s' is obtained from s by the contraction of its head redex. In particular, either $s' = u[t/x]t_1 \dots t_n$, $s' = u t_1 \dots t_n$ or $s' = t t_1 \dots t_n$ or $s' = ((u) \dots (u)t)t_1 \dots t_n$. This provides the justification for the next definition and proposition.

Definition 3 (Zoom-in Reduction). *Let $t \notin \mathsf{SN}$ and s be an elementary subterm of t. We write $t \overset{\mathsf{z}}{\mapsto} u$ if u has been obtained from t by replacing s with an $s' \notin \mathsf{SN}$ such that s' results from s by a contraction of the head redex of s. A sequence (finite or infinite) of terms $t_1, t_2, \dots, t_n, \dots$ is said to be a* zoom-in reduction *of t if $t = t_1$, and for all i, $t_i \overset{\mathsf{z}}{\mapsto} t_{i+1}$; if $i \leq j$, we write $t_i \overset{\mathsf{z}*}{\longmapsto} t_j$.*

Proposition 3. *Suppose $t \notin \mathsf{SN}$. There is an infinite zoom-in reduction of t.*

The zoom-in reduction strategy was studied in Melliès's PhD Thesis [15]. It is a *perpetual* reduction (see [21]), in the sense it preserves non-strong normalization. The idea is to contract each time a redex which is *essential* in order to produce an infinite reduction. In this way, one concentrates on a minimal amount of resources sufficient to generate non-termination. For example, the reduction $(\lambda y.\, y)(\lambda x.\, xx)\lambda x.\, xx \mapsto (\lambda x.\, xx)\lambda x.\, xx$ is smartly avoided by the relation $\overset{\mathsf{z}}{\mapsto}$, because the reduction of the first redex is not strictly necessary. Instead, one has $(\lambda y.\, y)(\lambda x.\, xx)\lambda x.\, xx \overset{\mathsf{z}}{\mapsto} (\lambda y.\, y)(\lambda x.\, xx)\lambda x.\, xx$ by contraction of the second redex.

We now study what happens when the zoom-in reduction strategy is applied to elementary terms. The goal is to prove an Inversion Property (Proposition 6). That is, starting from an elementary term ut, we want to show that t will necessarily be used in head position as an active function in the future of the zoom-in reduction of ut. In this sense, there will be an *inversion* of the roles of argument and function. We break the result in two steps.

The first observation is that the zoom-in reduction of ut will contract redexes inside u as long as the term is "blocked", i.e. u does not transform into a function.

Proposition 4. *Let ut be elementary. Then one of the following cases occurs:*

1. *There exists a term $(\lambda x v)t$ such that $ut \overset{\mathsf{z}*}{\longmapsto} (\lambda x v)t$.*
2. *There exists a term $\mathsf{It}^{\star} v\, t$ such that $ut \overset{\mathsf{z}*}{\longmapsto} \mathsf{It}^{\star} v\, t$*

Proof. By induction on $h(u)$. There are two cases:

– The head redex of ut is in u. Then $ut \overset{\mathsf{z}}{\mapsto} u't$, with $u \mapsto u'$ and $h(u') < h(u)$. By induction hypothesis, $u't \overset{\mathsf{z}*}{\longmapsto} (\lambda x v)t$ or $u't \overset{\mathsf{z}*}{\longmapsto} \mathsf{It}^{\star} v\, t$, and we are done.

– The head redex of ut is ut itself. If $ut = (\lambda x v)t$ or $ut = \mathsf{It}^{\star} v\, t$, we have the thesis. Moreover, those are the only possible cases, for neither $ut = \mathsf{if}^{\star} v\, t$ nor $ut = [\langle v, v'\rangle]\pi_i$ can hold; otherwise by Proposition 2, $v \notin \mathsf{SN}$ or $v' \notin \mathsf{SN}$ or $t \notin \mathsf{SN}$, but since ut is elementary, $v, v', t \in \mathsf{SN}$.

The second observation is that in a zoom-in reduction of a term $u[t/x] \notin \mathsf{SN}$, with $u, t \in \mathsf{SN}$, t will necessarily be used at some point in head position, because at some point one will run out of redexes in u.

Proposition 5. *Suppose $u, t \in \mathsf{SN}$ and $u[t/x] \notin \mathsf{SN}$. Then there exists v such that $u[t/x] \overset{\mathsf{z}*}{\longmapsto} v$ and v has an elementary subterm of the form $tt_1 \dots t_n$ $(n > 0)$.*

Proof. By induction on $h(u)$. Assume $u[t/x] \overset{\mathsf{z}}{\mapsto} w$; let s be the elementary subterm of $u[t/x]$ whose head redex is contracted in order to obtain w. We have the following possibilities:

- A redex inside u has been contracted, obtaining $u'[t/x]$, with $u \mapsto u'$. Then, $h(u') < h(u)$ and the proposition immediately follows by induction hypothesis.
- A redex inside t has been contracted. Since $t \in \mathsf{SN}$, s is not a subterm of t; moreover, since the head redex of s must have been contracted, $s = tt_1 \dots t_n$.
- A redex which is neither in t nor in u has been contracted. Then, t is a lambda abstraction or a pair or $\mathsf{if}^\star$ or $\mathsf{It}^\star$, u has a subterm of the form $xu_1 \dots u_n$ and $s = (xu_1 \dots u_n)[t/x] = tt_1 \dots t_n$. Of course, $n > 0$, since $t \in \mathsf{SN}$ and $s \notin \mathsf{SN}$.

We are now able to prove the Inversion Property, the most crucial result.

Proposition 6 (Inversion Property). *Let ut be elementary. Then there exists w such that $ut \overset{\mathsf{z}*}{\longmapsto} w$ and w has an elementary subterm of the form $tt_1 \dots t_n$ $(n > 0)$.*

Proof. By Propositions 4 and 5, one of the following cases occur:

1. $ut \overset{\mathsf{z}*}{\longmapsto} (\lambda x.v)t \mapsto v[t/x] \overset{\mathsf{z}*}{\longmapsto} w$, with w containing an elementary subterm of the form $tt_1 \dots t_n$ $(n > 0)$.
2. $ut \overset{\mathsf{z}*}{\longmapsto} \mathsf{It}^\star v\, t \mapsto (t) \dots (t)v = (x) \dots (x)v[t/x] \overset{\mathsf{z}*}{\longmapsto} w$ (for some x not free in v), with w containing an elementary subterm of the form $tt_1 \dots t_n$ $(n > 0)$.

By iteration of the Inversion Property, we finally obtain our characterization of non-strongly normalizable terms.

Theorem 2 (Characterization of non-strongly normalizable terms). *Let $u \notin \mathsf{SN}$. Then there exists an infinite sequence of terms $u_1, u_2, \dots, u_n, \dots$ such that $u_1 = u$, for all i, $u_i \overset{\mathsf{z}*}{\longmapsto} u_{i+1}$ and:*

$$u_1 \trianglerighteq t_1 \dots t_2, \quad u_2 \trianglerighteq t_2 \dots t_3, \quad u_3 \trianglerighteq t_3 \dots t_4, \; \dots, \; u_n \trianglerighteq t_n \dots t_{n+1} \; \dots$$

where for all i, $t_i \dots t_{i+1}$ is an elementary term.

Proof. We set $u_1 = u$. Supposing u_n to have been defined, and that $u_n \trianglerighteq t_n \dots t_{n+1}$ elementary. By Proposition 6, we can set u_{n+1} as the term obtained from u_n by substituting $t_n \dots t_{n+1}$ with a v' such that $t_n \dots t_{n+1} \overset{\mathsf{z}*}{\longmapsto} v'$ and v' contains an elementary subterm of the form $t_{n+1} \dots t_{n+2}$.

3 The System $\mathsf{T}^\star$ and its Strong Normalization

As well as one can consider a simply typed version of the ordinary lambda calculus with pairs, we now introduce a simply typed version of the non-deterministic lambda calculus $\Lambda^\star$. We call it System $\mathsf{T}^\star$, since it will be interpreted as a non-deterministic version of Gödel's System T. $\mathsf{T}^\star$ is formally described in Figure 2. The basic objects of $\mathsf{T}^\star$ are numerals and booleans, its basic computational constructs are primitive iterator at all types, if-then-else and pairs; $\overline{n}$ is the usual encoding $\mathsf{S}\dots\mathsf{S}0$ of the natural number n. The strong normalization of $\mathsf{T}^\star$ can be readily proved from the Characterization Theorem 2.

Types

$$\sigma, \tau ::= \mathtt{N} \mid \mathtt{Bool} \mid \sigma \to \tau \mid \sigma \times \tau$$

Constants

$$\mathsf{c} ::= \mathsf{It}^\star_\tau \mid \mathsf{if}^\star_\tau \mid 0 \mid \mathsf{S} \mid \mathtt{True} \mid \mathtt{False}$$

Terms

$$t, u ::= \ \mathsf{c} \mid x^\tau \mid (t)u \mid \lambda x^\tau u \mid \langle t, u\rangle \mid [t]\pi_0 \mid [t]\pi_1$$

Typing Rules for Variables and Constants

$$x^\tau : \tau$$

$$0 : \mathtt{N}, \mathsf{S} : \mathtt{N} \to \mathtt{N}$$

$$\mathtt{True} : \mathtt{Bool}, \mathtt{False} : \mathtt{Bool}$$

$$\mathsf{if}^\star_\tau : \tau \to \tau \to \tau$$

$$\mathsf{It}^\star_\tau : \tau \to (\tau \to \tau) \to \tau$$

Typing Rules for Composed Terms

$$\frac{t : \sigma \to \tau \qquad u : \sigma}{tu : \tau} \qquad \frac{u : \tau}{\lambda x^\sigma u : \sigma \to \tau}$$

$$\frac{u : \sigma \qquad t : \tau}{\langle u, t\rangle : \sigma \times \tau} \qquad \frac{u : \tau_0 \times \tau_1}{\pi_i u : \tau_i}\ i \in \{0, 1\}$$

Reduction Rules The same reduction rules of $\Lambda^\star$, restricted to the terms of $\mathsf{T}^\star$.

Fig. 2. The system $\mathsf{T}^\star$

Theorem 3 (Strong Normalization Theorem for $\mathsf{T}^\star$). *Every term w of $\mathsf{T}^\star$ is strongly normalizable.*

Proof. Suppose for the sake of contradiction that $w \notin \mathsf{SN}$. By the Characterization Theorem 2 (which can clearly be applied also to the terms of $\mathsf{T}^\star$), we obtain the existence of an infinite sequence of typed elementary terms $t_1 \dots t_2,\ t_2 \dots t_3,\ \dots, t_n \dots t_{n+1} \dots$. which yields a contradiction, since for every i, the type of t_i is strictly greater than the type of t_{i+1}.

4 The System T and its Strong Normalization

In this section we will prove the strong normalization theorem for System T. Syntax and typing rules of T are formally described in Figure 3.

Strong normalization follows as a corollary of Theorem 3. We define a simple translation mapping terms of System T into terms of System $\mathsf{T}^\star$:

Types

$$\sigma, \tau ::= \mathtt{N} \mid \mathtt{Bool} \mid \sigma \to \tau \mid \sigma \times \tau$$

Constants

$$c ::= \mathsf{It}_\tau \mid \mathsf{if}_\tau \mid 0 \mid \mathsf{S} \mid \mathtt{True} \mid \mathtt{False}$$

Terms

$$t, u ::= \ c \mid x^\tau \mid (t)u \mid \lambda x^\tau u \mid \langle t, u\rangle \mid [u]\pi_0 \mid [u]\pi_1$$

Typing Rules for Variables and Constants

$$\begin{aligned} x^\tau &: \tau \\ 0 : \mathtt{N}, \mathsf{S} &: \ \mathtt{N} \to \mathtt{N} \\ \mathtt{True} : \mathtt{Bool}, \mathtt{False} &: \ \mathtt{Bool} \\ \mathsf{if}_\tau &: \mathtt{Bool} \to \tau \to \tau \to \tau \\ \mathsf{It}_\tau &: \tau \to (\tau \to \tau) \to \mathtt{N} \to \tau \end{aligned}$$

Typing Rules for Composed Terms

$$\frac{t : \sigma \to \tau \qquad u : \sigma}{(t)u : \tau} \qquad \frac{u : \tau}{\lambda x^\sigma u : \sigma \to \tau}$$

$$\frac{u : \sigma \qquad t : \tau}{\langle u, t\rangle : \sigma \times \tau} \qquad \frac{u : \tau_0 \times \tau_1}{[u]\pi_i : \tau_i}\ i \in \{0, 1\}$$

Fig. 3. Syntax and Typing Rules for Gödel's system T

Definition 4 (Translation of T into $\mathsf{T}^\star$). *We define a translation $\_^* : \mathsf{T} \to \mathsf{T}^\star$, leaving types unchanged. In the case of constants of the form $\mathsf{if}_\tau, \mathsf{It}_\tau$, we set:*

$$(\mathsf{if}_\tau)^* := \lambda b^{\mathsf{Bool}}.\, \mathsf{if}^\star_\tau \qquad\qquad (\mathsf{It}_\tau)^* := \lambda x^\tau \lambda y^{\tau\to\tau} \lambda z^{\mathtt{N}}.\, \mathsf{It}^\star{}_\tau xy$$

For all other terms t of Gödel's System T, we set t^* as the term of $\mathsf{T}^\star$ obtained from t by replacing all its constants if_τ with $(\mathsf{if}_\tau)^*$ and all its constants It_τ with $(\mathsf{It}_\tau)^*$.

In the following, we will proceed by endowing T with two distinct reduction strategies, respectively dubbed as $\mapsto_{\mathsf{v}}$ and $\mapsto$. Informally, $\mapsto_{\mathsf{v}}$ forces a call-by-value discipline on the datatype $\mathtt{N}$. The second one, $\mapsto$, is the usual strategy T is endowed with. We will prove the strong normalization property in both cases. Whereas the goal is straightforward for $\mapsto_{\mathsf{v}}$, in the second case a bit of work is required.

4.1 Strong Normalization for System T with the strategy $\mapsto_{\mathsf{v}}$

The reduction strategy $\mapsto_{\mathsf{v}}$ is formally defined in Figure 4. Strong normalization theorem for T with $\mapsto_{\mathsf{v}}$ easily follows from Theorem 3. As a matter of fact, each computational step in T (with $\mapsto_{\mathsf{v}}$ reductions' set) can be plainly simulated in $\mathsf{T}^\star$ by a non-deterministic guess. In particular, each reduction step between T terms corresponds to *at least* a step between their translations:

Proposition 7 (Preservation of the Reduction Relation). *Let v be any term of T. Then $v \mapsto_{\mathsf{v}} w \implies v^* \mapsto^+ w^*$*

Proof. It is sufficient to prove the proposition when v is a redex r. We have several possibilities:

Reduction strategy $\mapsto_{\mathsf{v}}$

$$(\lambda x^{\tau} u)t \mapsto_{\mathsf{v}} u[t/x^{\tau}]$$

$$[\langle u_0, u_1\rangle]\pi_i \mapsto_{\mathsf{v}} u_i, \text{ for i=0,1}$$

$$\mathsf{It}_{\tau} u v \overline{n} \mapsto_{\mathsf{v}} \overbrace{(v)\dots(v)}^{n\ times} u$$

$$\mathsf{if}_{\tau}\, \mathtt{True}\, u\, v \mapsto_{\mathsf{v}} u \qquad \mathsf{if}_{\tau}\, \mathtt{False}\, u\, v \mapsto_{\mathsf{v}} v$$

Fig. 4. Reduction strategy $\mapsto_{\mathsf{v}}$ for T

1. $r = (\lambda x^{\tau} u)t \mapsto_{\mathsf{v}} u[t/x^{\tau}]$. We verify indeed that
$$((\lambda x^{\tau} u)t)^* = (\lambda x^{\tau} u^*)t^* \mapsto u^*[t^*/x^{\tau}] = u[t/x^{\tau}]^*$$
2. $r = \langle u_0, u_1\rangle\pi_i \mapsto_{\mathsf{v}} u_i$. We verify indeed that
$$(\langle u_0, u_1\rangle\pi_i)^* = \langle u_0^*, u_1^*\rangle\pi_i \mapsto u_i^*$$
3. $r = \mathsf{if}\, \mathtt{True}\, t\, u \mapsto_{\mathsf{v}} t$ or $r = \mathsf{if}\, \mathtt{False}\, t\, u \mapsto_{\mathsf{v}} u$. We verify indeed – by choosing the appropriate reduction rule for $\mathsf{if}^{\star}$ – that
$$(\mathsf{if}\, \mathtt{True}\, t\, u)^* = (\mathsf{if})^*\, \mathtt{True}\, t^*\, u^* \mapsto \mathsf{if}^{\star} t^* u^* \mapsto t^*$$
$$(\mathsf{if}\, \mathtt{False}\, t\, u)^* = (\mathsf{if})^*\, \mathtt{False}\, t^*\, u^* \mapsto \mathsf{if}^{\star} t^* u^* \mapsto u^*$$
4. $r = \mathsf{It}\, u\, t\, \overline{n} \mapsto_{\mathsf{v}} \overbrace{(t)\dots(t)}^{n\ times} u$. We verify indeed – by choosing the appropriate reduction rule for $\mathsf{It}^{\star}$ – that
$$(\mathsf{It}\, u\, t\, \overline{n})^* = (\mathsf{It})^* u^* t^* \overline{n} \mapsto^* \mathsf{It}^{\star} u^* t^* \mapsto (t^*)\dots(t^*)u^*$$

Theorem 4 (Strong Normalization for System T with $\mapsto_{\mathsf{v}}$ strategy). *Any term t of System T is strongly normalizable with respect to the relation $\mapsto_{\mathsf{v}}$.*

Proof. By Proposition 7, any infinite reduction $t = t_1, t_2, \dots, t_n, \dots$ in System T gives rise to an infinite reduction $t^* = t_1^*, t_2^*, \dots, t_n^*, \dots$ in System $\mathsf{T}^{\star}$. By the strong normalization Theorem 3 for $\mathsf{T}^{\star}$, infinite reductions of the latter kind cannot occur; thus neither of the former.

We have just proved the strong normalization theorem for T with the call-by-value restriction on the datatype $\mathtt{N}$. In any "practical" application (such as realizability, functional interpretation, program extraction from logical proofs), this evaluation discipline is perfectly suitable. From the constructive point of view, the call-by-value evaluation on natural numbers is even *desirable*. In fact, what essentially distinguishes the constructive reading of the iteration from the classical one is that the first requires complete knowledge of the number of times a functional will be iterated *before the actual execution* of the iteration. Call-by-value performs exactly this task: in a term $\mathsf{It}\, u\, v\, t$, it first completely evaluates t to a numeral, so providing a precise account about the number of times the function v will be called. Even if that is constructively satisfying, for the sake of completeness we will prove strong normalization with respect to the most general reduction strategy. This is the aim of the following section.

4.2 Strong Normalization of System T with the strategy $\mapsto$

The reduction strategy $\mapsto$ is formally defined in Figure 5. Notice that the only difference with respect to the call-by-value strategy $\mapsto_{\mathsf{v}}$ is that the term t in the reduction rule for It is not necessarily a numeral. We define SN_{T} to be the set of strongly normalizable terms of T with respect to the strategy $\mapsto$ and E_{T} to be the set of elementary terms of T with respect to the strategy $\mapsto$. We observe that it is still true that each term of T not in SN_{T} contains a term in E_{T}.

Reduction Strategy $\mapsto$

$$(\lambda x^{\tau} u)t \mapsto u[t/x^{\tau}]$$

$$[\langle u_0, u_1\rangle]\pi_i \mapsto u_i, \text{ for i=0,1}$$

$$\mathsf{It}_{\tau} uv0 \mapsto u \qquad \mathsf{It}_{\tau} uv(\mathsf{S}t) \mapsto v(\mathsf{It}_{\tau} uvt)$$

$$\mathsf{if}_{\tau}\,\mathtt{True}\,u\,v \mapsto u \qquad \mathsf{if}_{\tau}\,\mathtt{False}\,u\,v \mapsto v$$

Fig. 5. Reduction Strategy $\mapsto$ for System T

One may be tempted to proceed as in the previous section, by directly simulating $\mapsto$-reduction steps in T with reduction steps in $\mathsf{T}^{\star}$. Unfortunately, this is not possible. On the $\mathsf{T}^{\star}$ side, in order to interpret $\mathsf{It}\,u\,v\,t$, one has to "guess" the value of t by means of $\mathsf{It}^{\star}$. But it can very well happen that t is open, for example, so without value. To solve this issue, we are going to define an "almost" reduction relation $\mathscr{P}$ which can instead be simulated in $\mathsf{T}^{\star}$. In fact, $\mathscr{P}$ turns out to be a version of $\overset{\mathsf{Z}}{\mapsto}$ adapted to System T, which can be proved perpetual (Proposition 10). As a first step, we need to widen the class of numerals:

Definition 5 (Generalized Numerals). *A* generalized numeral *is a term of* T *of the form* $\mathsf{S}\ldots\mathsf{S}t$*, with* $t \in \mathsf{NF}$*,* $t \neq \mathsf{S}u$*;* GN *is the set of generalized numerals. If* $\mathsf{S}\ldots\mathsf{S}t$ *is a generalized numeral and* v *occurs in the head of* $(v)\ldots(v)u$ *as many times as* S *occurs in the prefix of* $\mathsf{SS}\ldots\mathsf{S}t$*, then* $(v)\ldots(v)u$ *is said to be the* expansion *of* $\mathsf{It}\,u\,v\,(\mathsf{S}\ldots\mathsf{S}t)$*.*

We remark that one could have equivalently defined GN as the set of type-$\mathtt{N}$ terms; this latter definition however does not generalized to untyped lambda calculus, while our results probably do, with some adaptation.

As a second step, we need to define a relation $\mathscr{P}$.

Definition 6 (Perpetual Relation $\mathscr{P}$). *Let* $t \notin \mathsf{SN}_{\mathsf{T}}$ *and* $s \in \mathsf{E}_{\mathsf{T}}$ *be a subterm of* t*. We write* $t \mathscr{P} u$ *if* u *has been obtained from* t *by replacing* s *with an* s' *such that:*

- $s = (\lambda x^{\tau} u)tt_1\ldots t_n \implies s' = u[t/x]t_1\ldots t_n$;
- $s = [\langle u_0, u_1\rangle]\pi_i t_1\ldots t_n \implies s' = u_i t_1\ldots t_n$;
- $s = \mathsf{if}\,\mathtt{True}\,t\,u\,t_1\ldots t_n \implies s' = tt_1\ldots t_n$;
- $s = \mathsf{if}\,\mathtt{False}\,t\,u\,t_1\ldots t_n \implies s' = ut_1\ldots t_n$;
- $s = \mathsf{It}\,u\,v\,t\,t_1\ldots t_n$ *and* $t \mapsto^{*} t' \in \mathsf{GN} \implies s' = ((v)(v)\ldots(v)u)t_1\ldots t_n$*, where* $(v)\ldots(v)u$ *is the expansion of* $\mathsf{It}\,u\,v\,t'$*.*

The idea behind $\mathscr{P}$ is to make it behave like a call-by-value strategy on $\mathtt{N}$, even when it should not be possible, by considering a term in GN as a "numeral". In order to show that $\mathscr{P}$ is perpetual, we need some technical but quite simple results.

The following lemma states that the set of non-strongly normalizable terms is closed w.r.t. the substitution of subterms in SN_T with their normal forms.

Lemma 1. *Assume $t_1, \ldots, t_n : \mathtt{N}$ and $t_1, \ldots, t_n \in \mathsf{SN}_\mathsf{T}$. Let $s_1, \ldots, s_n : \mathtt{N}$ be such that, for all $i = 1 \ldots n$, s_i is the normal form of t_i. Then, given any term u of T:*

$$u[t_1/x_1, \ldots, t_n/x_n] \notin \mathsf{SN}_\mathsf{T} \implies u[s_1/x_1, \ldots, s_n/x_n] \notin \mathsf{SN}_\mathsf{T}$$

Proof. It suffices to prove that there exist terms u', $t'_1, \ldots, t'_m \in \mathsf{SN}_\mathsf{T}$ and $s'_1, \ldots, s'_n$ such that for $i = 1, \ldots, m$, and

$$u'[t'_1/x_1, \ldots, t'_m/x_m] \notin \mathsf{SN}_\mathsf{T}$$

and

$$u[s_1/x_1, \ldots, s_n/x_n] \mapsto^+ u'[s'_1/x_1, \ldots, s'_m/x_m]$$

where again each s'_i is the normal form of t'_i. Since the end terms of the two lines above satisfy the hypothesis of the proposition, one may iterate this construction infinitely many times and obtains an infinite reduction of $u[s_1/x_1, \ldots, s_n/x_n]$.

In order to show that, let us consider an infinite reduction of $u[t_1/x_1, \ldots, t_n/x_n]$. Since $t_1, \ldots, t_n \in \mathsf{SN}_\mathsf{T}$, only finitely many reduction steps can be performed inside them. So the infinite reduction has a first segment of the shape:

$$u[t_1/x_1, \ldots, t_n/x_n] \mapsto^* u[t'_1/x_1, \ldots, t'_n/x_n] \mapsto w \notin \mathsf{SN}_\mathsf{T}$$

with $t_i \mapsto^* t'_i$. We have now two possibilities, depending on the kind of redex that has been contracted in order to obtain w:

1. $w = u'[t'_1/x_1, \ldots, t'_n/x_n]$, with $u \mapsto u'$. Then also $u[s_1/x_1, \ldots, s_n/x_n] \mapsto u'[s_1/x_1, \ldots, s_n/x_n]$ and we are done.
2. w has been obtained from $u[t'_1/x_1, \ldots, t'_n/x_n]$ by reduction of a redex created by the substitution t'_i/x_i. In this case, since $t'_1, \ldots, t'_n : \mathtt{N}$, the only possible redex of that kind has the form $(\mathsf{It}uvx_i)[t'_1/x_1, \ldots t'_i/x_i \ldots t'_n/x_n]$, with $\mathsf{It}uvx_i$ subterm of u and $t'_i = \mathsf{S}t_{n+1}$. Then v is obtained by replacing

$$\mathsf{It}uvx_i[t'_1/x_1, \ldots t'_i/x_i \ldots t'_n/x_n] = \mathsf{It}u'v'\mathsf{S}(t_{n+1})$$

with

$$(v')\mathsf{It}u'v't_{n+1} = (v)\mathsf{It}uvx_{n+1}[t'_1/x_1, \ldots t'_i/x_i \ldots t'_n/x_n \; t_{n+1}/x_{n+1}]$$

where x_{n+1} is a fresh variable. If we define $u' := u[(\mathsf{It}uvx_i) := (v)\mathsf{It}uvx_{n+1}]$ (i.e. u' is obtained from u by replacing $\mathsf{It}uvx_i$ with $(v)\mathsf{It}uvx_{n+1}$) we then have

$$v = u'[t'_1/x_1 \ldots t'_n/x_n \; t_{n+1}/x_{n+1}]$$

Since s_i is the normal form of $t'_i = \mathsf{S}t_{n+1}$, we have $s_i = \mathsf{S}s_{n+1}$, where s_{n+1} is the normal form of t_{n+1}. As before,

$$\mathsf{It}uvx_i[s_1/x_1, \dots s_i/x_i \dots s_n/x_n] \mapsto (v)\mathsf{It}uvx_{n+1}[s_1/x_1, \dots s_i/x_i \dots s_n/x_n \; s_{n+1}/x_{n+1}]$$

which implies $u[s_1/x_1 \dots s_n/x_n] \mapsto u'[s_1/x_1, \dots s_i/x_i \dots s_n/x_n \; s_{n+1}/x_{n+1}]$ and we are done.

By means of Lemma 1 it is possible to prove:

Proposition 8. *If* $(\mathsf{It}uvt)t_1 \dots t_n \in \mathsf{E_T}$ *and* $t \mapsto^* t' \in \mathsf{GN}$*, then* $(\mathsf{It}uvt')t_1 \dots t_n \in \mathsf{E_T}$.

Proof. By Lemma 1, applied to the terms $\mathsf{It}\, u\, v\, x\, t_1 \dots t_n[t/x]$ and $\mathsf{It}\, u\, v\, x\, t_1 \dots t_n[t'/x]$ (x fresh).

Lemma 2 is similar to Lemma 1: the set of non-strongly normalizable terms can be proved to be closed w.r.t. the substitution of subterms with their expansions.

Lemma 2. *Let* $t_1, \dots, t_n, s_1, \dots, s_n$ *be a sequence of terms such that for* $i = 1 \dots n$*,* s_i *is the expansion of* t_i *and all the proper subterms of* t_i *are in* $\mathsf{SN_T}$*. Then given any term* u *of* T*,*

$$u[t_1/x_1, \dots, t_n/x_n] \notin \mathsf{SN_T} \implies u[s_1/x_1, \dots, s_n/x_n] \notin \mathsf{SN_T}$$

Proof. It suffices to prove that there exist terms u' and $t'_1, \dots, t'_m, s'_1, \dots, s'_m$ such that for $i = 1, \dots, m$, s'_i is the expansion of t'_i, all the strict subterms of t'_i are in $\mathsf{SN_T}$ and

$$u'[t'_1/x_1, \dots, t'_m/x_m] \notin \mathsf{SN_T} \quad \text{and}$$
$$u[s_1/x_1, \dots, s_n/x_n] \mapsto^+ u'[s'_1/x_1, \dots, s'_m/x_m]$$

Since the end terms of the two lines above satisfy the hypothesis of the proposition, one may iterate this construction infinite times and obtains an infinite reduction of $u[s_1/x_1, \dots, s_n/x_n]$.

In order to show that, let us consider an infinite reduction of $u[t_1/x_1, \dots, t_n/x_n]$. By definition 5, $t_i = \mathsf{It}\, u_i\, v_i\, n_i$, for some u_i, v_i and generalized numeral n_i. Since $u_i, v_i \in \mathsf{SN_T}$, only finitely many reduction steps can be performed inside them. So the infinite reduction has a first segment of the shape:

$$u[t_1/x_1, \dots, t_n/x_n] \mapsto^* u[t'_1/x_1, \dots, t'_n/x_n] \mapsto v \notin \mathsf{SN_T}$$

with $t'_i = \mathsf{It}\, u'_i\, v'_i\, n_i$ and $u_i \mapsto^* u'_i, v_i \mapsto^* v'_i$. We have now two possibilities, depending on the kind of redex that has been contracted in order to obtain v (we notice that it must be already in u or in some t'_i):

1. $v = u'[t'_1/x_1, \dots, t'_n/x_n]$, with $u \mapsto u'$. Let now, for $i = 1, \dots, n$, s'_i be the expansion of t'_i. Then
$$s_i = (v_i) \dots (v_i)u_i \mapsto^* (v'_i) \dots (v'_i)u'_i = s'_i$$

   Therefore $u[s_1/x_1, \dots, s_n/x_n] \mapsto^+ u'[s'_1/x_1, \dots, s'_n/x_n]$ and we are done.

2. v has been obtained from $u[t'_1/x_1, \dots, t'_n/x_n]$ by replacing one of the occurrences of $t'_i = \mathsf{It}\, u'_i\, v'_i\, n_i$ with $(v'_i)\mathsf{It}\, u'_i\, v'_i\, m_i$ (assuming that $n_i = \mathsf{S}m_i$). Let $t'_{n+1} := \mathsf{It}\, u'_i\, v'_i\, m_i$. Then there exists a term u' (obtained from u by replacing a suitable occurrence of x_i with $(v'_i)x_{n+1}$, where x_{n+1} fresh) such that
$$v = u'[t'_1/x_1, \dots, t'_n/x_n\ t'_{n+1}/x_{n+1}]$$
Let now, for $i = 1, \dots, n+1$, s'_i be the expansion of t'_i. We want to show that
$$u[s_1/x_1, \dots, s_n/x_n] \mapsto^+ u'[s'_1/x_1, \dots, s'_n/x_n\ s'_{n+1}/x_{n+1}]$$
As before, $u[s_1/x_1, \dots, s_n/x_n] \mapsto^* u[s'_1/x_1, \dots, s'_n/x_n]$. Moreover, since s'_i is the expansion of $\mathsf{It}\, u'_i\, v'_i\, \mathsf{S}m_i$ and s'_{n+1} is the expansion of $\mathsf{It}\, u'_i\, v'_i\, m_i$, we have $s'_i = (v'_i)s'_{n+1}$. Therefore
$$x_i[s'_i/x_i] = s'_i = (v'_i)s'_{n+1} = (v'_i)x_{n+1}[s'_{n+1}/x_{n+1}]$$
and thus
$$u[s'_1/x_1, \dots, s'_n/x_n] = u'[s'_1/x_1, \dots, s'_n/x_n\ s'_{n+1}/x_{n+1}]$$
which concludes the proof.

The set E_T is closed w.r.t. the expansion of a head It redex of an elementary term:

Proposition 9. *Suppose that s' is the expansion of s. Then*
$$st_1 \dots t_n \in \mathsf{E}_\mathsf{T} \implies s't_1 \dots t_n \in \mathsf{E}_\mathsf{T}$$

Proof. By Lemma 2, applied to $xt_1 \dots t_n[s/x]$ and $xt_1 \dots t_n[s'/x]$ (x fresh).
Finally, the perpetuality of $\mathscr{P}$ follows from Propositions 8 and 9.

Proposition 10 (Perpetuality of $\mathscr{P}$). *If $t \notin \mathsf{SN}_\mathsf{T}$ and $t \mathrel{\mathscr{P}} u$, then $u \notin \mathsf{SN}_\mathsf{T}$.*

Proof. Assume u is obtained from t by replacing an elementary subterm s of u with s'; we show that $s' \notin \mathsf{SN}_\mathsf{T}$. The only case not covered by a straightforward adaptation of Proposition 2 is the one in which $s = \mathsf{It}\, u\, v\, t\, t_1 \dots t_n$ and $t \mapsto^* t' \in \mathsf{GN} \implies s' = ((v)(v)\dots(v)u)t_1 \dots t_n$, where $(v)\dots(v)u$ is the expansion of $\mathsf{It}uvt'$. Now, by Proposition 8, we obtain that $\mathsf{It}uvt'$ is elementary; by Proposition 9, we obtain that $((v)(v)\dots(v)u)t_1 \dots t_n$ is elementary too.

The perpetual relation $\mathscr{P}$ is simulated in $\mathsf{T}^\star$ by means of the translation $\_^*$.

Proposition 11 (Simulation of the Perpetual relation in $\mathsf{T}^\star$). *Let v be any term of T. Then $v \mathrel{\mathscr{P}} w \implies v^* \mapsto^+ w^*$.*

Proof. The proof is the same as that of proposition 7.

We are now able to prove the Strong Normalization Theorem for T:

Theorem 5 (Strong Normalization for System T). *Every term t of Gödel's System T is strongly normalizable with respect to the relation $\mapsto$.*

Proof. Suppose for the sake of contradiction that $t \notin \mathsf{SN}_\mathsf{T}$. By Proposition 10, there is an infinite sequence of terms $t = t_1, t_2, \dots, t_n, \dots$ in System T such that for all i, $t_i \mathrel{\mathscr{P}} t_{i+1}$. By Proposition 11 that gives rise to an infinite reduction $t^* = t^*_1, t^*_2, \dots, t^*_n, \dots$ in System $\mathsf{T}^\star$. By the strong normalization Theorem 3 for $\mathsf{T}^\star$, infinite reductions of the latter kind cannot occur: contradiction.

5 Conclusions and Related Works

Most of the proofs in this paper are intuitionistic. We remark however that our proof of the Characterization Theorem 2 is classical, since the excluded middle is used in a crucial way to prove Proposition 1. But this is not an issue: it is nowadays well-known how to interpret constructively classical proofs, especially when so limited a use of classical reasoning is made. One may thus obtain, by using classical realizabilities [12] or functional interpretations [19], non-trivial programs providing arbitrarily long approximations of the sequence of terms proved to exists in the Characterization Theorem. The same considerations apply to the proofs of the strong normalization theorems: it is possible to extract *directly* from them normalization algorithms (giving a nice case study in the field of program-extraction from classical proofs).

Our proofs of strong normalizations bear similarities with others. In [22], the iterator It_τ is translated as the infinite term

$$\lambda x^\tau \lambda f^{\tau\to\tau} \lambda n^{\mathtt{N}}.\ \langle x, (f)x, (f)(f)x, (f)(f)(f)x, \ldots\rangle\, n$$

and a weak normalization theorem is proven with respect to the new infinite calculus. On our side, the use of the non-deterministic operator $\mathsf{It}^\star$ clearly allows to simulate that infinite term. On a first thought, the move may not seem a big deal, but, surprisingly, the gain is considerable. First, one radically simplifies Tait's calculus by avoiding infinite terms. Secondly, the Characterization Theorem for $\Lambda^\star$ and $\mathsf{T}^\star$ *does not hold* for Tait's infinite calculus, since this latter does not enjoy its main corollary, strong normalization (an infinite term may contain infinite redexes). Last, with our technique we obtain strong normalization for T.

Our work has also some aspects in common with the technique of Joachimski-Matthes [14], which provides an adaptation of the technique in [18] that works for the lambda formulation of System T. For example, our use of generalized numerals is similar to the evaluation function of [14] used to inject Ω in SN. Indeed, we consider our work to be a refinement and an extension to an untyped setting of the methods of [18,14]. In fact, we claim to be also able to prove the strong normalization theorem for System $\mathsf{T}^\star$ *directly*, in a Van Daalen style (see also [4]). In other words, one can simplify both our proof for T (call-by-value) and the one in [14] by avoiding to reason on a inductively defined set of "SN" terms and instead use a triple induction. This is possible since the non-deterministic reduction relation of $\mathsf{T}^\star$ allows to express in a natural way a heavy inductive load, which is performed in [18,14] by defining a set of "regular" terms and the set "SN" by an omega-rule. Indeed, we believe that the idea of using non-determinism to simplify the study of strong normalization can be applied in other situations as well: we shall show that in future papers. Moreover, our technique makes explicit as a perpetual reduction the "reduction" hidden in the family of proofs in [18,20,14]. This enables not only to prove normalization, but also to increase the qualitative understanding of non-termination in lambda calculus with explicit recursion and to explain why it is avoided in the typed version. As for [1], we consider our extension of the Characterization Theorem from lambda calculus to $\mathsf{T}^\star$ as a genuine advancement: for quite a while, such a generalization seemed hopeless for a system which can simulate in a so direct way System T.

References

1. Aschieri, F.: *Una caratterizzazione dei lambda termini non fortemente normalizzabili.* Master Degree Thesis, Università degli Studi di Verona (2007).
2. Biasi, C., Aschieri, F.: *A Term Assignment for Polarized Bi-Intuitionistic Logic and its Strong Normalization.* Fundamenta Informaticae **84(02)** (2008).
3. Dal Lago, U., Zorzi, M.: *Probabilistic Operational Semantics for the Lambda Calculus.* RAIRO-ITA- DOI 10.1051/ita/2012012 **46(03)** (2012) 413–450.
4. David, R., Nour, K: *A short proof of the strong normalization of the simply typed lambda-mu-calculus.* Schedae Informaticae **12** (2003) 27–34.
5. de' Liguoro, U., Piperno, A.: *Non-Deterministic Extensions of Untyped Lambda-Calculus.* Information and Computation **122** (1995) 149–177.
6. Girard, J.-Y., Lafont, Y., Taylor, P.: *Proofs and Types.* Cambridge University Press (1989).
7. Howard, W.A.: *Ordinal analysis of terms of finite type.* The Journal of Symbolic Logic, **45(3)** (1980) 493–504.
8. Gandy, R.O.: *Proofs of Strong Normalization*, Essays on Combinatoriay Logic, Lambda Calculus and Formalism. (1980) 457–477. Academic Press, London.
9. Khasidashvili, Z., Ogawa, M.: *Perpetualilty and Uniform Normalization*, Algebraic and Logic Programming, Springer LNCS, **1298** (1997) 240-255.
10. Kreisel, G.: *Interpretation of Analysis by Means of Constructive Functionals of Finite Types.* Constructivity in Mathematics (1959)101128. North-Holland.
11. Krivine, J.-L.: *Lambda-calcul types et modèles*, Studies in Logic and Foundations of Mathematics (1990) 1–176. Masson, Paris.
12. Krivine, J.-L.: *Classical Realizability.* In Interactive models of computation and program behavior. Panoramas et synthèses **27** (2009) 197–229. Société Mathématique de France.
13. Levy, J.-J.: *Reductions correctes et optimales dans le lambda-calcul.* PhD Thesis, Université Paris 7 (1978).
14. Joachimski, F., Matthes, R.: *Short proofs of normalization for the simply-typed lambda-calculus, permutative conversions and Gödel's T*, Archive of Mathematical Logic, **42(1)** (2003), 49–87.
15. Melliès, P.-A.: *Description Abstraite des Systèmes de Réécriture.* PhD Thesis, Université Paris 7 (1996).
16. Nederpelt, R.-P.: *Strong Normalization in Typed Lambda Calculus with Lambda Structured Types.* PhD Thesis, Eindhoven University of Technology (1973).
17. Prawitz, D.: *Ideas and Results in Proof Theory.* In Proceedings of the Second Scandinavian Logic Symposium (1971).
18. Sanchis, L.-E.: *Functionals Defined by Recursion.* Notre Dame Journal of Formal Logic, **VIII(3)** (1967) 161–174.
19. Spector, C.: *Provably recursive functionals of analysis: a consistency proof of analysis by an extension of principles in current intuitionistic mathematics.* In Proceedings of Symposia in Pure Mathematics, **5** AMS (1962) 1–27.
20. van Daalen, D.: *The language theory of Automath.* PhD Thesis, Eindhoven University of Technology (1977).
21. van Raamsdonk, F., Severi, P., Sørensen, M.H., Xi, H.: *Perpetual Reductions in Lambda-Calculus.* Information and Computation **149** (1999) 173–225.
22. Tait, W.:*Infinitely Long Terms of Transfinite Type.* Formal Systems and Recursive Functions **40** (1965) 176–185.