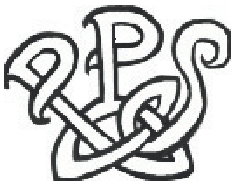
Induction principles as the foundation of the theory of normalisation: Concepts and Techniques

Stéphane Lengrand



Induction principles as the foundation of the theory of normalisation: Concepts & Techniques

Technical report

PPS, Université Denis Diderot, Paris VII
School of Computer Science, University of St Andrews

Stéphane Lengrand

4th March 2005

Contents

Introduction

The first part of this report was originally aimed at defining coherent terminology and notations about reduction relations and their normalisation. The definition of the notions of normalisation are inspired by a thread created by René Vestergaard on the TYPES mailing-list, gathering and comparing the various definitions. Our first purpose here is redefining and re-establishing a theory of normalisation that does not rely on classical logic and double negation.

Negation usually lies in the very definition of strong normalisation already, when it is expressed as "there is no infinite reduction sequence". The most striking example is the use of the definition in order to prove that a reduction relation is strongly normalising. It usually starts with "suppose an infinite reduction sequence" and ends with a contradiction. We believe that the theory of normalisation is not specifically classical, but the habit of using classical logic has been taken because of convenience. Here, we show a theory of normalisation that is just as convenient but constructive.

In this theory, the induction principle is no longer a property of strongly normalising relations, but is its very definition. In other words, instead of basing the notion of strong normalisation on the finiteness of reduction sequences, we base it on the notion on induction: by definition, a relation is strongly normalising if it satisfies the induction principle. The latter should hold for every predicate, so the notion of normqlisqtion is based on second-order quantification rather than double-negation.

We express several induction principles in that setting, then we re-establish some traditional results, especially some techniques to prove strong normalisation. We constructively prove the simulation technique and a few refinements, as well as the termination of the lexicographic reductions and the multi-set reductions. A constructive proof of the latter has already been given by Wilfried Buchholz and is a special case of Coquand's constructive treatment [Coq94] of Ramsey theory.

The second part of this report presents two new techniques for proving strong normalisation. The first one is fundamentally classical but applies to any rewrite system, whereas the second one might hold in intuitionistic logic and applies more specifically to calculi that have some connexion with λ-calculus. When applying the techniques, a major part of the proofs is actually independent from the calculus to which they are applied.

As an example, we show how the former technique can be used to prove the normalisation of the explicit substitution calculus $\lambda\mathsf{x}$ [BR95], which yields a short proof of Preservation of Strong Normalisation (PSN). Since the technique is generic, we also prove those properties for the explicit substitution calculus $\overline{\lambda}$ [Her95], and the proof is shorter than the existing ones in [DU03] and [Kik04]. In both calculi the technique also allows us to easily derive the strong normalisation of typed terms from that of typed λ-terms. Unfortunately, since our technique is fundamentally classical, it cannot draw advantage of the constructive proofs of strong normalisation such as the one in [JM03] for the simply-typed λ-calculus.

We also apply the latter technique to the PSN property of the explicit substitution calculus $\lambda\mathsf{lxr}$ [KL05], a calculus with a full composition of substitutions, for which the standard techniques all failed. This is a new result.

The two techniques can be combined in a fruitful way, for instance for proving cut-elimination in various powerful sequent calculi, including some type theories such as the systems of Barendregt's Cube expressed in sequent calculus.

1 A constructive theory of normalisation

1.1 Relations

We start by establishing some notations about relations and sets.

Definition 1 (Relations) We denote the composition of relations by $\cdot$, the identity relation by Id, and the inverse relation by $^{-1}$, all defined below:

Let $\mathcal{R} : \mathcal{A} \longrightarrow \mathcal{B}$ and $\mathcal{R}' : \mathcal{B} \longrightarrow \mathcal{C}$.

- Composition
  $\mathcal{R} \cdot \mathcal{R}' : \mathcal{A} \longrightarrow \mathcal{C}$ is defined as follows: given $M \in \mathcal{A}$ and $N \in \mathcal{C}$,
  $M(\mathcal{R} \cdot \mathcal{R}')N$ if there exists $P \in \mathcal{B}$ such that $M\mathcal{R}P$ and $P\mathcal{R}'N$

- Identity
  $\mathsf{Id} : \mathcal{A} \longrightarrow \mathcal{A}$ is defined as follows:
  given $M \in \mathcal{A}$ and $N \in \mathcal{A}$, $M\mathsf{Id}N$ if $M = N$
  (Note that for higher-order rewrite systems, the above notion of equality is α-conversion)

- Inverse
  $\mathcal{R}^{-1} : \mathcal{B} \longrightarrow \mathcal{A}$ is defined as follows:
  given $M \in \mathcal{B}$ and $N \in \mathcal{A}$, $M\mathcal{R}^{-1}N$ if $N\mathcal{R}M$

If $\mathcal{D} \subseteq \mathcal{A}$, we write $\mathcal{R}(\mathcal{D})$ for $\{M \in \mathcal{B} |\ \exists N \in \mathcal{D}, N\mathcal{R}M\}$, or equivalently $\bigcup_{N\in\mathcal{D}}\{M \in \mathcal{B} |\ N\mathcal{R}M\}$. When $\mathcal{D}$ is the singleton $\{M\}$, we write $\mathcal{R}(M)$ for $\mathcal{R}(\{M\})$.

Now when $\mathcal{A} = \mathcal{B}$ we define the *relation induced by* $\mathcal{R}$ *through* $\mathcal{R}'$, written $\mathcal{R}'[\mathcal{R}]$, as $\mathcal{R}'^{-1} \cdot \mathcal{R} \cdot \mathcal{R}' : \mathcal{C} \longrightarrow \mathcal{C}$.

We say that a relation $\mathcal{R} : \mathcal{A} \longrightarrow \mathcal{B}$ is *total* if $\mathcal{R}^{-1}(\mathcal{B}) = \mathcal{A}$.

All those notions and notations can be used in the particular case when $\mathcal{R}$ is a function, that is, if $\forall M \in \mathcal{A}$, $\mathcal{R}(M)$ is of the form $\{N\}$ (which we simply write $\mathcal{R}(M) = N$).

Remark 1 Notice that composition is associative, and identity relations are neutral for the composition operation.

Computation in a calculus is described by the notion of reduction relation, defined as follows.

Definition 2 (Reduction relation) A *reduction relation* on $\mathcal{A}$ is a relation from $\mathcal{A}$ to $\mathcal{A}$ (i.e. a subset of $\mathcal{A} \times \mathcal{A}$), which we often write as $\rightarrow$.

Given a reduction relation $\rightarrow$ on $\mathcal{A}$, we define the set of $\rightarrow$-*reducible forms* (or just *reducible forms* when the relation is clear) as $\mathsf{rf}^{\rightarrow} = \{M \in \mathcal{A} |\ \exists N \in \rightarrow(M)\}$. We define the set of *normal forms* as $\mathsf{nf}^{\rightarrow} = \{M \in \mathcal{A} |\ \ \rightarrow(M) = \emptyset\}$.

Given a reduction relation $\rightarrow$ on $\mathcal{A}$, we define $\rightarrow^n$ by induction on the natural number n as follows:
$\rightarrow^0 = \mathsf{Id}$
$\rightarrow^{n+1} = \rightarrow \cdot \rightarrow^n (= \rightarrow^n \cdot \rightarrow)$
$\rightarrow^+$ denotes the transitive closure of $\rightarrow$ (formally, $\rightarrow^+ = \bigcup_{n\geq 1} \rightarrow^n$).
$\rightarrow^*$ denotes the transitive and reflexive closure of $\rightarrow$ (formally, $\rightarrow^* = \bigcup_{n\geq 0} \rightarrow^n$).
$\leftrightarrow^*$ denotes the transitive, reflexive and symmetric closure of $\rightarrow$.

Definition 3 (Finitely branching relations) A reduction relation $\rightarrow$ on $\mathcal{A}$ is *finitely branching* if $\forall M \in \mathcal{A},\ \rightarrow(M)$ is finite.

Definition 4 Given a reduction relation $\rightarrow$ on $\mathcal{A}$, we say that a subset $\mathcal{T}$ of $\mathcal{A}$ is $\rightarrow$-*stable* (or *stable under* $\rightarrow$) if $\rightarrow(\mathcal{T}) \subseteq \mathcal{T}$.

1.2 Normalisation and induction

Proving a universally quantified property by induction consists of verifying that the set of elements having the property is stable, in some sense similar to -yet more subtle than- the one above. Leading to different induction principles, we define two such notions of stability property: being *patriarchal* and being *paternal*.

Definition 5 Given a reduction relation $\rightarrow$ on $\mathcal{A}$, we say that

- a subset $\mathcal{T}$ of $\mathcal{A}$ is $\rightarrow$-*patriarchal* (or just *patriarchal* when the relation is clear) if $\forall N \in \mathcal{A},\ \rightarrow(N) \subseteq \mathcal{T} \Rightarrow N \in \mathcal{T}$.
- a subset $\mathcal{T}$ of $\mathcal{A}$ is $\rightarrow$-*paternal* (or just *paternal* when the relation is clear) if it contains $\mathsf{nf}^{\rightarrow}$ and is stable under $\rightarrow^{-1}$.
- a predicate P on $\mathcal{A}$ is *patriarchal* (resp. *paternal*) if $\{M \in \mathcal{A} |\ P(M)\}$ is *patriarchal* (resp. *paternal*).

Lemma 2 *Suppose that for any N in $\mathcal{A}$, $N \in \mathsf{rf}^{\rightarrow}$ or $N \in \mathsf{nf}^{\rightarrow}$ and suppose $\mathcal{T} \subseteq \mathcal{A}$. If $\mathcal{T}$ is paternal, then it is patriarchal.*

Proof: In order to prove $\forall N \in \mathcal{A},\ \rightarrow(N) \subseteq \mathcal{T} \Rightarrow N \in \mathcal{T}$, a case analysis is needed: either $N \in \mathsf{rf}^{\rightarrow}$ or $N \in \mathsf{nf}^{\rightarrow}$. In both cases $N \in \mathcal{T}$ because $\mathcal{T}$ is paternal. □

Remark 3 Notice that we can obtain from classical logic the hypothesis for all N in $\mathcal{A}$, $N \in \mathsf{rf}^{\rightarrow}$ or $N \in \mathsf{nf}^{\rightarrow}$, because it is an instance of the Law of Excluded Middle. In intuitionistic logic, assuming that amounts to saying that being reducible is decidable, which is very often the case.

We would not require this hypothesis if we defined that $\mathcal{T}$ is paternal whenever $\forall N \in \mathcal{A}, N \in \mathcal{T} \vee (N \in \mathsf{rf}^{\rightarrow} \wedge (\rightarrow(N) \cap T = \emptyset))$. This is classically equivalent to the definition above, but this definition also has some disadvantages as we shall see later.

Typically, if we want to prove that a predicate holds on some set, we actually prove that it is patriarchal or paternal, depending on the induction principle we use.

Hence, we define normalisation so that normalising elements are those captured by an induction principle, which should hold for every predicate satisfying the corresponding stability property. We thus get two notions of normalisation: the *strongly* (resp. *weakly*) *normalising* elements are those in every patriarchal (resp. paternal) set.

Definition 6 (Normalising elements) Given a reduction relation $\rightarrow$ on $\mathcal{A}$:

- The set of $\rightarrow$-*strongly normalising* elements is

$$\mathsf{SN}^{\rightarrow} = \bigcap_{\mathcal{T} \text{ is patriarchal}} \mathcal{T}$$

- The set of $\rightarrow$-*weakly normalising* elements is

$$\mathsf{WN}^{\rightarrow} = \bigcap_{\tau \text{ is paternal}} \mathcal{T}$$

Remark 4 Interestingly enough, $\mathsf{WN}^{\rightarrow}$ can also be captured by an inductive definition:

$$\mathsf{WN}^{\rightarrow} = \bigcup_{n} \mathsf{WN}^{\rightarrow}_{n}$$

where $\mathsf{WN}^{\rightarrow}_{n}$ is defined by induction on the natural number n as follows:

$$\mathsf{WN}^{\rightarrow}_{0} = \mathsf{nf}^{\rightarrow}$$
$$\mathsf{WN}^{\rightarrow}_{n+1} = \{M \in \mathcal{A} |\ \exists n' \leq n, M \in \rightarrow^{-1}(\mathsf{WN}^{\rightarrow}_{n'})\}$$

With the alternative definition of paternal suggested in Remark 3, the inclusion $\mathsf{WN}^{\rightarrow} \subseteq \bigcup_n \mathsf{WN}^{\rightarrow}_{n}$ would require the assumption that being reducible by $\rightarrow$ is decidable. We therefore preferred the first definition because we can then extract from a term M in $\mathsf{WN}^{\rightarrow}$ a natural number n such that $M \in \mathsf{WN}^{\rightarrow}_{n}$, without the hypothesis of decidability.

Such a characterisation gives us the possibility to prove that all weakly normalising elements satisfy some property by induction on n. On the other hand, trying to do so with strong normalisation leads to a different notion, as we shall see below. Hence, we lack for $\mathsf{SN}^{\rightarrow}$ an induction principle based on natural numbers, which is the reason why we built-in a specific induction principle in the definition of $\mathsf{SN}^{\rightarrow}$.

Definition 7 The set of $\rightarrow$-*bounded* elements is defined as

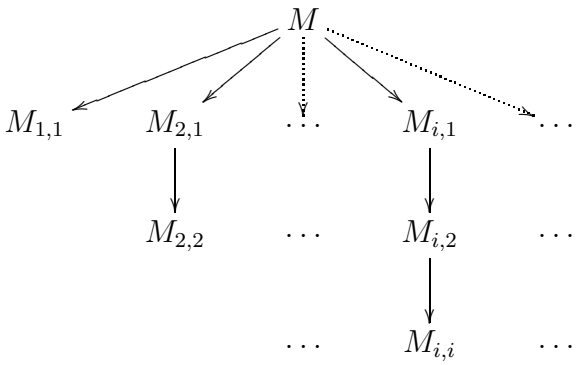
$$\mathsf{BN}^{\rightarrow} = \bigcup_{n} \mathsf{BN}^{\rightarrow}_{n}$$

where $\mathsf{BN}^{\rightarrow}_{n}$ is defined by induction on the natural number n as follows:

$$\mathsf{BN}^{\rightarrow}_{0} = \mathsf{nf}^{\rightarrow}$$
$$\mathsf{BN}^{\rightarrow}_{n+1} = \{M \in \mathcal{A} |\ \exists n' \leq n,\ \rightarrow(M) \subseteq \mathsf{BN}^{\rightarrow}_{n'}\}$$

But we have the following fact:

Remark 5 For some reduction relations $\rightarrow$, $\mathsf{SN}^{\rightarrow} \neq \mathsf{BN}^{\rightarrow}$. For instance, in the following relation, $M \in \mathsf{SN}^{\rightarrow}$ but $M \notin \mathsf{BN}^{\rightarrow}$.



However, suppose that $\rightarrow$ is finitely branching. Then $\mathsf{BN}^{\rightarrow}$ is patriarchal.
As a consequence, $\mathsf{BN}^{\rightarrow} = \mathsf{SN}^{\rightarrow}$ (the counter-example could not be finitely branching).

Proof: Suppose $\rightarrow(M) \subseteq \mathsf{BN}^{\rightarrow}$. Because $\rightarrow$ is finitely branching, there exists a natural number n such that $\rightarrow(M) \subseteq \mathsf{BN}^{\rightarrow}_{n}$. Clearly, $M \in \mathsf{BN}^{\rightarrow}_{n+1} \subseteq \mathsf{BN}^{\rightarrow}$. □

Remark 6 As a trivial example, all the natural numbers are $>$-bounded. Indeed, any natural number n is in $\mathsf{BN}^{>}_{n}$, which can be proved by induction.

A canonical way of proving a statement $\forall M \in \mathsf{BN}^{\to}, P(M)$ is to prove by induction on the natural number n that $\forall M \in \mathsf{BN}^{\to}_{n}, P(M)$. Although we can exhibit no such natural number on which a statement $\forall M \in \mathsf{SN}^{\to}, P(M)$ can be proved by induction, the following induction principles hold by definition of normalisation:

Remark 7 Given a predicate P on $\mathcal{A}$,

1. Suppose P is patriarchal
   (that is, $\forall M \in \mathcal{A}, (\forall N \in {\to}(M), P(N)) \Rightarrow P(M)$).
   Then $\forall M \in \mathsf{SN}^{\to}, P(M)$.

2. Suppose P is paternal
   (that is, $\forall M \in \mathcal{A}, (M \in \mathsf{nf}^{\to} \vee \exists N \in {\to}(M), P(N)) \Rightarrow P(M)$).
   Then $\forall M \in \mathsf{WN}^{\to}, P(M)$.

When we use this remark to prove $\forall M \in \mathsf{SN}^{\to}, P(M)$ (resp. $\forall M \in \mathsf{WN}^{\to}, P(M)$), we say that we prove it *by raw induction in* $\mathsf{SN}^{\to}$ (resp. *in* $\mathsf{WN}^{\to}$).

Definition 8 (Strongly normalising relations) Given a reduction relation $\to$ on $\mathcal{A}$ and a subset $\mathcal{T} \subseteq \mathcal{A}$, we say that the reduction relation is *strongly normalising* or *terminating* on $\mathcal{T}$ if $\mathcal{T} \subseteq \mathsf{SN}^{\to}$. If we do not specify $\mathcal{T}$, it means that we take $\mathcal{T} = \mathcal{A}$. we mean

Remark 8

1. If $n < n'$ then $\mathsf{BN}^{\to}_{n} \subseteq \mathsf{BN}^{\to}_{n'} \subseteq \mathsf{BN}^{\to}$. In particular, $\mathsf{nf}^{\to} \subseteq \mathsf{BN}^{\to}_{n} \subseteq \mathsf{BN}^{\to}$.

2. $\mathsf{BN}^{\to} \subseteq \mathsf{SN}^{\to}$ and $\mathsf{BN}^{\to} \subseteq \mathsf{WN}^{\to}$.
   Hence, all natural numbers are in $\mathsf{SN}^{>}$ and $\mathsf{WN}^{>}$.

3. If being reducible is decidable (or if we work in classical logic), then $\mathsf{SN}^{\to} \subseteq \mathsf{WN}^{\to}$.

Proof:

1. By definition.

2. Both facts can be proved for all $\mathsf{BN}^{\to}_{n}$ by induction on n.

3. This comes from Remark 2 and thus requires either classical logic or the particular instance of the Law of Excluded Middle stating that for all N,

$$N\mathsf{rf}^{\to} \vee N \in \mathsf{nf}^{\to}$$

□

Lemma 9

1. *$\mathsf{SN}^{\to}$ is patriarchal, $\mathsf{WN}^{\to}$ is paternal.*

2. *If $M \in \mathsf{BN}^{\to}$ and $\to(M) \subseteq \mathsf{BN}^{\to}$.*
   If $M \in \mathsf{SN}^{\to}$ then $\to(M) \subseteq \mathsf{SN}^{\to}$.
   If $M \in \mathsf{WN}^{\to}$ then either $M \in \mathsf{nf}^{\to}$ or $M \in \to^{-1}(\mathsf{WN}^{\to})$
   (which implies $M \in \mathsf{rf}^{\to} \Rightarrow M \in \to^{-1}(\mathsf{WN}^{\to})$).

Proof:

1. For the first statement, let $M \in \mathcal{A}$ such that $\to(M) \subseteq \mathsf{SN}^{\to}$ and let $\mathcal{T}$ be patriarchal. We want to prove that $M \in \mathcal{T}$. It suffices to prove that $\to(M) \subseteq \mathcal{T}$. This is the case, because $\to(M) \subseteq \mathsf{SN}^{\to} \subseteq \mathcal{T}$.
   For the second statement, first notice that $\mathsf{nf}^{\to} \subseteq \mathsf{WN}^{\to}$. Now let $M, N \in \mathcal{A}$ such that $M \to N$ and $N \in \mathsf{WN}^{\to}$, and let $\mathcal{T}$ be paternal. We want to prove that $M \in \mathcal{T}$. This is the case because $N \in \mathcal{T}$ and $\mathcal{T}$ is paternal.

2. The first statement is straightforward.
   For the second, we show that $\mathcal{T} = \{P \in \mathcal{A} |\ \to(P) \subseteq \mathsf{SN}^{\to}\}$ is patriarchal:
   Let $P \in \mathcal{A}$ such that $\to(P) \subseteq \mathcal{T}$, that is, $\forall R \in \to(P)$, $\to(R) \subseteq \mathsf{SN}^{\to}$.
   Because $\mathsf{SN}^{\to}$ is patriarchal, $\forall R \in \to(P), R \in \mathsf{SN}^{\to}$.
   Hence, $\to(P) \subseteq \mathsf{SN}^{\to}$, that is, $P \in \mathcal{T}$ as required.
   Now by definition of $\mathsf{SN}^{\to}$, we get $M \in \mathcal{T}$.
   For the third statement, we prove that $\mathcal{T} = \mathsf{nf}^{\to} \cup \to^{-1}(\mathsf{WN}^{\to})$ is paternal:
   Clearly, it suffices to prove that it is stable under $\to^{-1}$. Let $P, Q \in \mathcal{A}$ such that $P \to Q$ and $Q \in \mathcal{T}$. If $Q \in \mathsf{nf}^{\to} \subseteq \mathsf{WN}^{\to}$, then $P \in \to^{-1}(\mathsf{WN}^{\to}) \subseteq \mathcal{T}$. If $Q \in \to^{-1}(\mathsf{WN}^{\to})$, then, because $\mathsf{WN}^{\to}$ is paternal, we get $Q \in \mathsf{WN}^{\to}$, so that $P \in \to^{-1}(\mathsf{WN}^{\to}) \subseteq \mathcal{T}$ as required.
   Now by definition of $M \in \mathsf{WN}^{\to}$, we get $M \in \mathcal{T}$.

□

Notice that this lemma gives the well-known characterisation of $\mathsf{SN}^{\to}$:
$M \in \mathsf{SN}^{\to}$ if and only if $\forall N \in \to(M), N \in \mathsf{SN}^{\to}$.

Now we refine the induction principle immediately contained in the definition of normalisation by relaxing the requirement that the predicate should be patriarchal or paternal:

Theorem 10 (Induction principle) *Given a predicate P on $\mathcal{A}$,*

1. *Suppose $\forall M \in \mathsf{SN}^{\to}, (\forall N \in \to(M), P(N)) \Rightarrow P(M)$.*
   Then $\forall M \in \mathsf{SN}^{\to}, P(M)$.

2. *Suppose $\forall M \in \mathsf{WN}^{\to}, (M \in \mathsf{nf}^{\to} \vee \exists N \in \to(M), P(N)) \Rightarrow P(M)$.*
   Then $\forall M \in \mathsf{WN}^{\to}, P(M)$.

When we use this theorem to prove a statement $P(M)$ for all M in $\mathsf{SN}^{\to}$ (resp. $\mathsf{WN}^{\to}$), we just add $(\forall N \in \to(M), P(N))$ (resp. $M \in \mathsf{nf}^{\to} \vee \exists N \in \to(M), P(N)$) to the assumptions, which we call the induction hypothesis.

We say that we prove the statement by induction in $\mathsf{SN}^{\to}$ *(resp.* in $\mathsf{WN}^{\to}$*).*

Proof:

1. We prove that $\mathcal{T} = \{M \in \mathcal{A} |\ M \in \mathsf{SN}^{\rightarrow} \Rightarrow P(M)\}$ is patriarchal.
   Let $N \in \mathcal{A}$ such that $\rightarrow(N) \subseteq \mathcal{T}$. We want to prove that $N \in \mathcal{T}$:
   Suppose that $N \in \mathsf{SN}^{\rightarrow}$. By Lemma 9 we get that $\forall R \in \rightarrow(N), R \in \mathsf{SN}^{\rightarrow}$. By definition of $\mathcal{T}$ we then get $\forall R \in \rightarrow(N), P(R)$. From the main hypothesis we get $P(N)$. Hence, we have shown $N \in \mathcal{T}$.
   Now by definition of $M \in \mathsf{SN}^{\rightarrow}$, we get $M \in \mathcal{T}$, which can be simplified as $P(M)$ as required.

2. We prove that $\mathcal{T} = \{M \in \mathcal{A} |\ M \in \mathsf{WN}^{\rightarrow} \wedge P(M)\}$ is paternal.
   Let $N \in \mathsf{nf}^{\rightarrow} \subseteq \mathsf{WN}^{\rightarrow}$. By the main hypothesis we get $P(N)$.
   Now let $N \in \rightarrow^{-1}(\mathcal{T})$, that is, there is $R \in \mathcal{T}$ such that $N \rightarrow R$.
   We want to prove that $N \in \mathcal{T}$:
   By definition of $\mathcal{T}$, we have $R \in \mathsf{WN}^{\rightarrow}$, so $N \in \mathsf{WN}^{\rightarrow}$ (because $\mathsf{WN}^{\rightarrow}$ is paternal). We also have $P(R)$, so we can apply the main hypothesis to get $P(N)$. Hence, we have shown $N \in \mathcal{T}$.
   Now by definition of $M \in \mathsf{WN}^{\rightarrow}$, we get $M \in \mathcal{T}$, which can be simplified as $P(M)$ as required.

□

As a first application of the induction principle, we prove the following results:

Remark 11 $M \in \mathsf{SN}^{\rightarrow}$ if and only if there is no infinite reduction sequence starting from R (classically, with the axiom of choice).

Proof:

- *only if*: Consider the predicate $P(M)$ "having no infinite reduction sequence starting from M". We prove it by induction in $\mathsf{SN}^{\rightarrow}$. If M starts an infinite reduction sequence, then there is a $N \in \rightarrow(M)$ that also starts an infinite reduction sequence, which contradicts the induction hypothesis.

- *if*: Suppose $M \notin \mathsf{SN}^{\rightarrow}$. There is a patriarchal set $\mathcal{T}$ in which M is not. Hence, there is a $N \in \rightarrow(M)$ that is not in $\mathcal{T}$, and we re-iterate on it, creating an infinite reduction sequence. This uses the axiom of choice.

□

Remark 12

1. If $\rightarrow \subseteq \rightarrow'$, then $\mathsf{nf}^{\rightarrow} \supseteq \mathsf{nf}^{\rightarrow'}$, $\mathsf{WN}^{\rightarrow} \supseteq \mathsf{WN}^{\rightarrow'}$, $\mathsf{SN}^{\rightarrow} \supseteq \mathsf{SN}^{\rightarrow'}$, and for all n, $\mathsf{BN}_n^{\rightarrow} \supseteq \mathsf{BN}_n^{\rightarrow'}$.

2. $\mathsf{nf}^{\rightarrow} = \mathsf{nf}^{\rightarrow^+}$, $\mathsf{WN}^{\rightarrow} = \mathsf{WN}^{\rightarrow^+}$, $\mathsf{SN}^{\rightarrow} = \mathsf{SN}^{\rightarrow^+}$, and for all n, $\mathsf{BN}_n^{\rightarrow^+} = \mathsf{BN}_n^{\rightarrow}$.

Proof:

1. By expanding the definitions.

2. For each statement, the right-to-left inclusion is a corollary of point 1.
   For the first statement, it remains to prove that $\mathsf{nf}^{\rightarrow} \subseteq \mathsf{nf}^{\rightarrow^+}$.
   Let $M \in \mathsf{nf}^{\rightarrow}$. By definition, $\rightarrow(M) = \emptyset$, so clearly $\rightarrow^+(M) = \emptyset$ as well.
   For the second statement, it remains to prove that $\mathsf{WN}^{\rightarrow} \subseteq \mathsf{WN}^{\rightarrow^+}$ which we do by induction in $\mathsf{WN}^{\rightarrow}$:
   Assume $M \in \mathsf{WN}^{\rightarrow}$ and the induction hypothesis that either $M \in \mathsf{nf}^{\rightarrow}$ or there is $N \in \rightarrow(M)$ such that $N \in \mathsf{WN}^{\rightarrow^+}$. In the former case, we have $M \in \mathsf{nf}^{\rightarrow} = \mathsf{nf}^{\rightarrow^+}$ and $\mathsf{nf}^{\rightarrow^+} \subseteq \mathsf{WN}^{\rightarrow^+}$. In the latter case, we have $N \in \rightarrow^+(M)$. Because of Lemma 9, $\mathsf{WN}^{\rightarrow^+}$ is stable by $\mathsf{WN}^{\rightarrow^{+-1}}$, and hence $M \in \mathsf{WN}^{\rightarrow^+}$.
   For the third statement, it remains to prove that $\mathsf{SN}^{\rightarrow} \subseteq \mathsf{SN}^{\rightarrow^+}$. We prove the stronger statement that $\forall M \in \mathsf{SN}^{\rightarrow}$, $\rightarrow^*(M) \subseteq \mathsf{SN}^{\rightarrow^+}$ by induction in $\mathsf{SN}^{\rightarrow}$: assume $M \in \mathsf{SN}^{\rightarrow}$ and the induction hypothesis $\forall N \in \rightarrow(M)$, $\rightarrow^*(N) \subseteq \mathsf{SN}^{\rightarrow^+}$. Clearly, $\rightarrow^+(M) \subseteq \mathsf{SN}^{\rightarrow^+}$. Because of Lemma 9, $\mathsf{SN}^{\rightarrow^+}$ is $\rightarrow^+$-patriarchal, so $M \in \mathsf{SN}^{\rightarrow^+}$, and hence $\rightarrow^*(M) \subseteq \mathsf{SN}^{\rightarrow^+}$.
   The statement $\mathsf{BN}_n^{\rightarrow} \subseteq \mathsf{BN}_n^{\rightarrow^+}$ can easily be proved by induction on n.

□

Notice that this result enables us to use a stronger induction principle: in order to prove $\forall M \in \mathsf{SN}^{\rightarrow}, P(M)$, it now suffices to prove

$$\forall M \in \mathsf{SN}^{\rightarrow}, (\forall N \in \rightarrow^+(M), P(N)) \Rightarrow P(M)$$

This induction principle is called the *transitive induction in* $\mathsf{SN}^{\rightarrow}$.

Lemma 13 (Strong normalisation of disjoint union) *Suppose that $(\mathcal{A}_i)_{i\in I}$ is a family of sets on some index set I, each being equipped with a reduction relation $\rightarrow_i$.*
Suppose that they are pairwise disjoint ($\forall i, j \in I^2, i \neq j \Rightarrow \mathcal{A}_i \cap \mathcal{A}_j = \emptyset$).
Consider the reduction relation $\rightarrow = \bigcup_{i\in I} \rightarrow_i$ on $\bigcup_{i\in I} \mathcal{A}_i$.
We have $\bigcup_{i\in I} \mathsf{SN}^{\rightarrow_i} \subseteq \mathsf{SN}^{\rightarrow}$.

Proof: It suffices to prove that for all $j \in I$, $\mathsf{SN}^{\rightarrow_j} \subseteq \mathsf{SN}^{\rightarrow}$, which we do by induction in $\mathsf{SN}^{\rightarrow_j}$. Assume $M \in \mathsf{SN}^{\rightarrow_j}$ and assume the induction hypothesis $\rightarrow_j(M) \subseteq \mathsf{SN}^{\rightarrow}$.

We must prove $M \in \mathsf{SN}^{\rightarrow}$, so it suffices to prove that for all N such that $M \rightarrow N$ we have $N \in \mathsf{SN}^{\rightarrow}$.

By definition of the disjoint union, all such N are in $\rightarrow_j(M)$ so we can apply the induction hypothesis. □

1.3 Termination by simulation & lexicographic termination

Now that we have established an induction principle on strongly normalising elements, the question arises of how we can prove strong normalisation. In this subsection we re-establish in our framework the well-known technique of simulation, which can be found for instance in [BN98]. The basic technique to prove that a reduction relation on the set $\mathcal{A}$ terminates consists in mapping the elements of $\mathcal{A}$ to elements of a set $\mathcal{B}$ equipped with its own reduction relation known to be terminating, and proving that the reduction in $\mathcal{A}$ can be simulated by that of $\mathcal{B}$. The mapping is sometimes called the *measure function* or the *weight function*. We generalise here the technique by replacing the weight function by a

relation between $\mathcal{A}$ and $\mathcal{B}$. Oddly enough, we were unable to find this easy generalisation in the literature. But the main point here is that the simulation technique is the typical example where the proof usually starts with "suppose an infinite reduction sequence" and ends with a contradiction. We show how the use of classical logic is completely unnecessary, provided that we use a constructive definition of SN such as ours.

Definition 9 (Strong and Weak Simulation)
Let $\mathcal{R}$ be a relation between two sets $\mathcal{A}$ and $\mathcal{B}$, equipped with the reduction relations $\rightarrow_{\mathcal{A}}$ and $\rightarrow_{\mathcal{B}}$ respectively.

- $\rightarrow_{\mathcal{B}}$ *strongly simulates* $\rightarrow_{\mathcal{A}}$ *through* $\mathcal{R}$ if $(\mathcal{R}^{-1} \cdot \rightarrow_{\mathcal{A}}) \subseteq (\rightarrow_{\mathcal{B}}^{+} \cdot \mathcal{R}^{-1})$.

  In other words, for all $M, M' \in \mathcal{A}$ and for all $N \in \mathcal{B}$, if $M\mathcal{R}N$ and $M \rightarrow_{\mathcal{A}} M'$ then there is $N' \in \mathcal{B}$ such that $M'\mathcal{R}N'$ and $N \rightarrow_{\mathcal{B}}^{+} N'$.

  Notice that when $\mathcal{R}$ is a function, this implies $\mathcal{R}[\rightarrow_{\mathcal{A}}] \subseteq \rightarrow_{\mathcal{B}}^{+}$.

- $\rightarrow_{\mathcal{B}}$ *weakly simulates* $\rightarrow_{\mathcal{A}}$ *through* $\mathcal{R}$ if $(\mathcal{R}^{-1} \cdot \rightarrow_{\mathcal{A}}) \subseteq (\rightarrow_{\mathcal{B}}^{*} \cdot \mathcal{R}^{-1})$.

  In other words, for all $M, M' \in \mathcal{A}$ and for all $N \in \mathcal{B}$, if $M\mathcal{R}N$ and $M \rightarrow_{\mathcal{A}} M'$ then there is $N' \in \mathcal{B}$ such that $M'\mathcal{R}N'$ and $N \rightarrow_{\mathcal{B}}^{*} N'$.

  Notice that when $\mathcal{R}$ is a function, this implies $\mathcal{R}[\rightarrow_{\mathcal{A}}] \subseteq \rightarrow_{\mathcal{B}}^{*}$.

Theorem 14 (Strong normalisation by strong simulation) *Let $\mathcal{R}$ be a relation between $\mathcal{A}$ and $\mathcal{B}$, equipped with the reduction relations $\rightarrow_{\mathcal{A}}$ and $\rightarrow_{\mathcal{B}}$.*
If $\rightarrow_{\mathcal{B}}$ strongly simulates $\rightarrow_{\mathcal{A}}$ through $\mathcal{R}$, then $\mathcal{R}^{-1}(\mathit{SN}^{\rightarrow_{\mathcal{B}}}) \subseteq \mathit{SN}^{\rightarrow_{\mathcal{A}}}$.

Proof: $\mathcal{R}^{-1}(\mathsf{SN}^{\rightarrow_{\mathcal{B}}}) \subseteq \mathsf{SN}^{\rightarrow_{\mathcal{A}}}$ can be reformulated as

$$\forall N \in \mathsf{SN}^{\rightarrow_{\mathcal{B}}}, \forall M \in \mathcal{A}, M\mathcal{R}N \Rightarrow M \in \mathsf{SN}^{\rightarrow_{\mathcal{A}}}$$

which we prove by transitive induction in $\mathsf{SN}^{\rightarrow_{\mathcal{B}}}$. Assume $N \in \mathsf{SN}^{\rightarrow_{\mathcal{B}}}$ and assume the induction hypothesis $\forall N' \in \rightarrow_{\mathcal{B}}^{+}(N), \forall M' \in \mathcal{A}, M'\mathcal{R}N' \Rightarrow M' \in \mathsf{SN}^{\rightarrow_{\mathcal{A}}}$. Now let $M \in \mathcal{A}$ such that $M\mathcal{R}N$. We want to prove that $M \in \mathsf{SN}^{\rightarrow_{\mathcal{A}}}$. It suffices to prove that $\forall M' \in \rightarrow(M), M' \in \mathsf{SN}^{\rightarrow_{\mathcal{A}}}$. Let M' be such that $M \rightarrow_{\mathcal{A}} M'$. The simulation hypothesis provides $N' \in \rightarrow_{\mathcal{B}}^{+}(N)$ such that $M'\mathcal{R}N'$. We apply the induction hypothesis on N', M' and get $M' \in \mathsf{SN}^{\rightarrow_{\mathcal{A}}}$ as required. □

The simulation technique can be improved by another standard method. It consists of splitting the reduction relation into two parts, then proving that the first part is strongly simulated by a first auxiliary terminating relation, and then proving that the second part is weakly simulated by it and strongly simulated by a second auxiliary terminating relation.

In some sense, the two auxiliary terminating relations act as measures that decrease lexicographically.

We express this method in our constructive framework.

Lemma 15 *Given two reduction relations $\rightarrow$, $\rightarrow'$, suppose that $\mathit{SN}^{\rightarrow}$ is stable under $\rightarrow'$. Then $\mathit{SN}^{\rightarrow \cup \rightarrow'} = \mathit{SN}^{\rightarrow^{*} \cdot \rightarrow'} \cap \mathit{SN}^{\rightarrow}$*

Proof: The left-to-right inclusion is an application of Theorem 14: $\rightarrow \cup \rightarrow'$ strongly simulates both $\rightarrow^* \cdot \rightarrow'$ and $\rightarrow$ through Id.

Now we prove the right-to-left inclusion. We first prove the following lemma:

$$\forall M \in \mathsf{SN}^{\rightarrow}, (\rightarrow^* \cdot \rightarrow')(M) \subseteq \mathsf{SN}^{\rightarrow \cup \rightarrow'} \Rightarrow M \in \mathsf{SN}^{\rightarrow \cup \rightarrow'}$$

We do this by induction in $\mathsf{SN}^{\rightarrow}$, so not only assume $(\rightarrow^* \cdot \rightarrow')(M) \subseteq \mathsf{SN}^{\rightarrow \cup \rightarrow'}$, but also assume the induction hypothesis:
$\forall N \in \rightarrow(M),\ (\rightarrow^* \cdot \rightarrow')(N) \subseteq \mathsf{SN}^{\rightarrow \cup \rightarrow'} \Rightarrow N \in \mathsf{SN}^{\rightarrow \cup \rightarrow'}$.
We want to prove that $M \in \mathsf{SN}^{\rightarrow \cup \rightarrow'}$, so it suffices to prove that both $\forall N \in \rightarrow'(M), N \in \mathsf{SN}^{\rightarrow \cup \rightarrow'}$ and $\forall N \in \rightarrow(M), N \in \mathsf{SN}^{\rightarrow \cup \rightarrow'}$. The former case is a particular case of the first hypothesis. The latter case would be provided by the second hypothesis (the induction hypothesis) if only we could prove that $(\rightarrow^* \cdot \rightarrow')(N) \subseteq \mathsf{SN}^{\rightarrow \cup \rightarrow'}$. But this is true because $(\rightarrow^* \cdot \rightarrow')(N) \subseteq (\rightarrow^* \cdot \rightarrow')(M)$ and the first hypothesis reapplies.

Now we prove

$$\forall M \in \mathsf{SN}^{\rightarrow^* \cdot \rightarrow'}, M \in \mathsf{SN}^{\rightarrow} \Rightarrow M \in \mathsf{SN}^{\rightarrow \cup \rightarrow'}$$

We do this by induction in $\mathsf{SN}^{\rightarrow^* \cdot \rightarrow'}$, so not only assume $M \in \mathsf{SN}^{\rightarrow}$, but also assume the induction hypothesis $\forall N \in (\rightarrow^* \cdot \rightarrow')(M), N \in \mathsf{SN}^{\rightarrow} \Rightarrow N \in \mathsf{SN}^{\rightarrow \cup \rightarrow'}$.
Now we can combine those two hypotheses, because we know that $\mathsf{SN}^{\rightarrow}$ is stable under $\rightarrow'$: since $M \in \mathsf{SN}^{\rightarrow}$, we have $(\rightarrow^* \cdot \rightarrow')(M) \subseteq \mathsf{SN}^{\rightarrow}$, so that the induction hypothesis can be simplified in $\forall N \in (\rightarrow^* \cdot \rightarrow')(M), N \in \mathsf{SN}^{\rightarrow \cup \rightarrow'}$.
This gives us exactly the conditions to apply the above lemma to M. □

Corollary 16 (Lexicographic termination)
Let $\mathcal{A}_1, \ldots, \mathcal{A}_n$ be sets, respectively equipped with the reduction relations $\rightarrow_{\mathcal{A}_1}, \ldots, \rightarrow_{\mathcal{A}_n}$. For $1 \leq i \leq n$, let $\rightarrow_i$ be the reduction relation on $\mathcal{A}_1 \times \cdots \times \mathcal{A}_n$ defined as follows:

$$(M_1, \ldots, M_n) \rightarrow_i (N_1, \ldots, N_n)$$

if $M_i \rightarrow_{\mathcal{A}_i} N_i$ and for all $1 \leq j < i$, $M_j = N_j$ and for all $i < j \leq n$, $N_j \in \mathsf{SN}^{\rightarrow_{\mathcal{A}_j}}$
We define the lexicographic reduction $\rightarrow_{\mathsf{lex}}$ as $\rightarrow_1 \cup \ldots \cup \rightarrow_n$. We then have:

$$\mathsf{SN}^{\rightarrow_{\mathcal{A}_1}} \times \cdots \times \mathsf{SN}^{\rightarrow_{\mathcal{A}_n}} \subseteq \mathsf{SN}^{\rightarrow_{\mathsf{lex}}}$$

In particular, if $\rightarrow_{\mathcal{A}_i}$ is terminating on $\mathcal{A}_i$ for all $1 \leq i \leq n$, then $\rightarrow_{\mathsf{lex}}$ is terminating on $\mathcal{A}_1 \times \cdots \times \mathcal{A}_n$.

Proof: By induction on n: for $n = 1$, we conclude from $\rightarrow_{\mathcal{A}_1} = \rightarrow_1$.
Then notice that $\rightarrow_{\mathcal{A}_{n+1}}$ strongly simulates $\rightarrow_{n+1}$ through the $(n+1)^{th}$ projection. Hence, by Theorem 14, if $N_{n+1} \in \mathsf{SN}^{\rightarrow_{\mathcal{A}_{n+1}}}$ then $(N_1, \ldots, N_{n+1}) \in \mathsf{SN}^{\rightarrow_{n+1}}$, which we can also formulate as $\mathcal{A}_1 \times \cdots \times \mathcal{A}_n \times \mathsf{SN}^{\rightarrow_{\mathcal{A}_{n+1}}} \subseteq \mathsf{SN}^{\rightarrow_{n+1}}$.
A first consequence of this is $\mathsf{SN}^{\rightarrow_{\mathcal{A}_1}} \times \cdots \times \mathsf{SN}^{\rightarrow_{\mathcal{A}_{n+1}}} \subseteq \mathsf{SN}^{\rightarrow_{n+1}}$ (1). A second one is that $\mathsf{SN}^{\rightarrow_{n+1}}$ is stable under $\rightarrow_1 \cup \ldots \cup \rightarrow_n$ (2). Now notice that $\rightarrow_1 \cup \ldots \cup \rightarrow_n$ strongly simulates $\rightarrow_{n+1}^* \cdot (\rightarrow_1 \cup \ldots \cup \rightarrow_n)$ through the projection that drops the $(n+1)^{th}$ component. We can thus apply Theorem 14 to get $\mathsf{SN}^{\rightarrow_1 \cup \ldots \cup \rightarrow_n} \times \mathcal{A}_{n+1} \subseteq \mathsf{SN}^{\rightarrow_{n+1}^* \cdot (\rightarrow_1 \cup \ldots \cup \rightarrow_n)}$, which, combined with the induction hypothesis, gives $\mathsf{SN}^{\rightarrow_{\mathcal{A}_1}} \times \cdots \times \mathsf{SN}^{\rightarrow_{\mathcal{A}_{n+1}}} \subseteq \mathsf{SN}^{\rightarrow_{n+1}^* \cdot (\rightarrow_1 \cup \ldots \cup \rightarrow_n)}$ (3). From (1), (2), and (3) we can now conclude by using Lemma 15. □

Corollary 17 *Let $\mathcal{A}$ be a set equipped with a reduction relation $\rightarrow$.*
For each natural number n, let $\rightarrow_{lexn}$ be the lexicographic reduction on $\mathcal{A}^n$.
Consider the reduction relation $\rightarrow_{lex} = \bigcup_n \rightarrow_{lexn}$ on the disjoint union $\bigcup_n \mathcal{A}^n$.

$$\bigcup_n (SN^{\rightarrow})^n \subseteq SN^{\rightarrow_{lex}}$$

Proof: It suffices to combine Corollary 16 with Lemma 13. □

Corollary 18 *Let $\rightarrow_{\mathcal{A}}$ and $\rightarrow'_{\mathcal{A}}$ be two reduction relations on $\mathcal{A}$, and $\rightarrow_{\mathcal{B}}$ be a reduction relation on $\mathcal{B}$. Suppose*

- *$\rightarrow'_{\mathcal{A}}$ is strongly simulated by $\rightarrow_{\mathcal{B}}$ through $\mathcal{R}$*
- *$\rightarrow_{\mathcal{A}}$ is weakly simulated by $\rightarrow_{\mathcal{B}}$ through $\mathcal{R}$*
- *$SN^{\rightarrow_{\mathcal{A}}} = \mathcal{A}$*

Then $\mathcal{R}^{-1}(SN^{\rightarrow_{\mathcal{B}}}) \subseteq SN^{\rightarrow_{\mathcal{A}} \cup \rightarrow'_{\mathcal{A}}}$.
(In other words, if $M \mathcal{R} N$ and $N \in SN^{\rightarrow_{\mathcal{B}}}$ then $M \in SN^{\rightarrow_{\mathcal{A}} \cup \rightarrow'_{\mathcal{A}}}$.)

Proof: Clearly, the reduction relation $\rightarrow^*_{\mathcal{A}} \cdot \rightarrow'_{\mathcal{A}}$ is strongly simulated by $\rightarrow_{\mathcal{B}}$ through $\mathcal{R}$, so that by Theorem 14 we get $\mathcal{R}^{-1}(\mathsf{SN}^{\rightarrow_{\mathcal{B}}}) \subseteq \mathsf{SN}^{\rightarrow^*_{\mathcal{A}} \cdot \rightarrow'_{\mathcal{A}}}$.
But $\mathsf{SN}^{\rightarrow^*_{\mathcal{A}} \cdot \rightarrow'_{\mathcal{A}}} = \mathsf{SN}^{\rightarrow^*_{\mathcal{A}} \cdot \rightarrow'_{\mathcal{A}}} \cap \mathsf{SN}^{\rightarrow_{\mathcal{A}}} = \mathsf{SN}^{\rightarrow_{\mathcal{A}} \cup \rightarrow'_{\mathcal{A}}}$, by the Lemma 15 (since $\mathsf{SN}^{\rightarrow_{\mathcal{A}}} = \mathcal{A}$ is obviously stable by $\rightarrow'_{\mathcal{A}}$). □

The intuitive idea behind this corollary is that after a certain number of $\rightarrow_{\mathcal{A}}$-steps and $\rightarrow'_{\mathcal{A}}$-steps, the only reductions in $\mathcal{A}$ that can take place are those that no longer modify the encoding in $\mathcal{B}$, that is, $\rightarrow_{\mathcal{A}}$-steps. Then it suffices to show that $\rightarrow_{\mathcal{A}}$ terminate, so that no infinite reduction sequence can start from M, as illustrated in Figure 1.

1.4 Multi-set termination

Now we define the notions of multi-sets their reductions. We constructively prove their termination. A classical proof of the result can be found in [Ter03].

Definition 10 (Multi-Sets) Given a set $\mathcal{A}$, a *multi-set on* $\mathcal{A}$ is a total function from $\mathcal{A}$ to the natural numbers such that only a finite subset of elements are not mapped to 0.

Notice that for two such multi-sets f and g, the function $f + g$ mapping any element M of $\mathcal{A}$ to $f(M) + g(M)$ is still a multi-set on $\mathcal{A}$.

We define the multi-set $\{\{N_1, \ldots, N_n\}\}$ as $f_1 + \cdots + f_n$, where for all $1 \leq i \leq n$, f_i maps N_i to 1 and every other element to 0.

We write abusively $M \in f$ if $f(M) \neq 0$.

Definition 11 (Multi-Set reduction relation) Given $\rightarrow$ is a reduction relation on $\mathcal{A}$, we define the multi-set reduction as follows:
if f and g are multi-sets on $\mathcal{A}$, we say that $f \rightarrow_{\mathsf{mul}} g$ if there is a M in $\mathcal{A}$ such that

$$\begin{cases} f(M) = g(M) + 1 \\ \forall N \in \mathcal{A}, f(N) < g(N) \Rightarrow M \rightarrow N \end{cases}$$

In what follows we always assume that $\mathcal{A}$ is a set with a reduction relation $\rightarrow$.

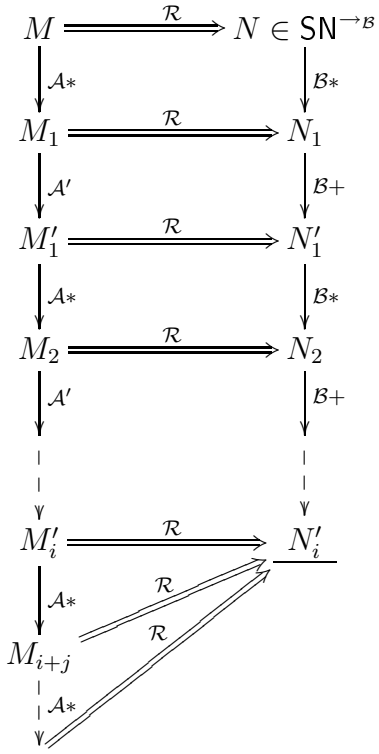


Figure 1: Deriving strong normalisation by simulation

Lemma 19 *If $f_1, \ldots, f_n, g$ are multi-sets on $\mathcal{A}$ and $f_1 + \cdots + f_n \rightarrow_{\mathsf{mul}} g$ then there is $1 \leq i \leq n$ and a multi-set f'_i such that $f_i \rightarrow_{\mathsf{mul}} f'_i$ and $f_1 + \cdots + f_{i-1} + f'_i + f_{i+1} + \cdots + f_n = g$.*

Proof: We know that there is a M in $\mathcal{A}$ such that

$$\begin{cases} f_1(M) + \cdots + f_n(M) = g(M) + 1 \\ \forall N \in \mathcal{A}, f_1(N) + \cdots + f_n(N) < g(N) \Rightarrow M \rightarrow N \end{cases}$$

An easy lexicographic induction on two natural numbers p and q shows that if $p + q > 0$ then $p > 0$ or $q > 0$. By induction on the natural number n, we extend this result: if $p_1 + \cdots + p_n > 0$ then $\exists i, p_i > 0$. We apply this result on $f_1(M) + \cdots + f_n(M)$ and get some $f_i(M) > 0$. Obviously there is a unique f'_i such that $f_1 + \cdots + f_{i-1} + f'_i + f_{i+1} + \cdots + f_n = g$, and we also get $f_i \rightarrow_{\mathsf{mul}} f'_i$. □

Definition 12 Given two sets $\mathcal{N}$ and $\mathcal{N}'$ of multi-sets, we define $\mathcal{N} + \mathcal{N}'$ as $\{f + g |\ f \in \mathcal{N}, g \in \mathcal{N}'\}$.

We define for every M in $\mathcal{A}$ its *relative multi-sets* as all the multi-sets f on $\mathcal{A}$ such that if $N \in f$ then $M \rightarrow^* N$. We denote the set of relative multi-sets as $\mathcal{M}_M$.

Remark 20 Notice that for any $M \in \mathcal{A}$, $\mathcal{M}_M$ is stable under $\rightarrow_{\mathsf{mul}}$.

Lemma 21 *For all $M_1, \ldots, M_n$ in $\mathcal{A}$,*
if $\mathcal{M}_{M_1} \cup \ldots \cup \mathcal{M}_{M_n} \subseteq \mathsf{SN}^{\rightarrow_{\mathsf{mul}}}$ then $\mathcal{M}_{M_1} + \cdots + \mathcal{M}_{M_n} \subseteq \mathsf{SN}^{\rightarrow_{\mathsf{mul}}}$.

Proof: Let $\mathcal{W}$ be the relation between $\mathcal{M}_{M_1} + \cdots + \mathcal{M}_{M_n}$ and $\mathcal{M}_{M_1} \times \cdots \times \mathcal{M}_{M_n}$ defined as: $f_1 + \cdots + f_n \mathcal{W} (f_1, \ldots, f_n)$ for all $f_1, \ldots, f_n$ in $\mathcal{M}_{M_1} \times \cdots \times \mathcal{M}_{M_n}$.

We consider as a reduction relation on $\mathcal{M}_{M_1} \times \cdots \times \mathcal{M}_{M_n}$ the lexicographic composition of $\rightarrow_{\mathsf{mul}}$. We denote this reduction relation as $\rightarrow_{\mathsf{mullex}}$. By Corollary 16, we know that $\mathcal{M}_{M_1} \times \cdots \times \mathcal{M}_{M_n} \subseteq \mathsf{SN}^{\rightarrow_{\mathsf{mullex}}}$. Hence, $\mathcal{W}^{-1}(\mathsf{SN}^{\rightarrow_{\mathsf{mullex}}}) = \mathcal{M}_{M_1} + \cdots + \mathcal{M}_{M_n}$.

Now we prove that $\mathcal{M}_{M_1} + \cdots + \mathcal{M}_{M_n}$ is stable by $\rightarrow_{\mathsf{mul}}$ and that $\rightarrow_{\mathsf{mullex}}$ strongly simulates $\rightarrow_{\mathsf{mul}}$ through $\mathcal{W}$. Suppose $f_1 + \cdots + f_n \rightarrow_{\mathsf{mul}} g$. By Lemma 19 we get a multi-set f_i' such that $f_1 + \cdots + f_{i-1} + f_i' + f_{i+1} + \cdots + f_n = g$ and $f_i \rightarrow_{\mathsf{mul}} f_i'$.
Hence, $f_i' \in \mathcal{M}_{M_i}$, so that $(f_1, \ldots, f_{i-1}, f_i', f_{i+1}, \cdots, f_n) \in \mathcal{M}_{M_1} \times \cdots \times \mathcal{M}_{M_n}$ and even $(f_1, \cdots, f_n) \rightarrow_{\mathsf{mullex}} (f_1, \ldots, f_{i-1}, f_i', f_{i+1}, \cdots, f_n)$.

By Theorem 14 we then get $\mathcal{W}^{-1}(\mathsf{SN}^{\rightarrow_{\mathsf{mullex}}}) \subseteq \mathsf{SN}^{\rightarrow_{\mathsf{mul}}}$, which concludes the proof because $\mathcal{W}^{-1}(\mathsf{SN}^{\rightarrow_{\mathsf{mullex}}}) = \mathcal{M}_{M_1} + \cdots + \mathcal{M}_{M_n}$. □

Lemma 22 $\forall M \in \mathit{SN}^{\rightarrow}, \mathcal{M}_M \subseteq \mathit{SN}^{\rightarrow_{\mathit{mul}}}$

Proof: By transitive induction in $\mathsf{SN}^{\rightarrow}$. Assume that $M \in \mathsf{SN}^{\rightarrow}$ and assume the induction hypothesis $\forall N \in \rightarrow^+(M), \mathcal{M}_N \subseteq \mathsf{SN}^{\rightarrow_{\mathsf{mul}}}$.

Let us split the reduction relation $\rightarrow_{\mathsf{mul}}$: if $f \rightarrow_{\mathsf{mul}} g$, let $f \rightarrow_{\mathsf{mul1}} g$ if $f(M) = g(M)$ and let $f \rightarrow_{\mathsf{mul2}} g$ if $f(M) > g(M)$. Clearly, if $f \rightarrow_{\mathsf{mul}} g$ then either $f \rightarrow_{\mathsf{mul1}} g$ or $f \rightarrow_{\mathsf{mul1}} g$. This is an intuitionistic implication since the equality of two natural numbers can be decided.

Now we prove that $\rightarrow_{\mathsf{mul1}}$ is terminating on $\mathcal{M}_M$.

Let $\mathcal{W}'$ be the following relation (actually, a function) between $\mathcal{M}_M$ to itself: for all f and g in $\mathcal{M}_M$, $f \mathcal{W} g$ if $g(M) = 0$ and for all $N \neq M$, $f(N) = g(N)$.

For a given $f \in \mathcal{M}_M$, let $N_1, \ldots, N_n$ be the elements of $\mathcal{A}$ that are not mapped to 0 by f and that are different from M. Since $f \in \mathcal{M}_M$, for all $1 \leq i \leq n$ we know $M \rightarrow^+ N_i$, and we also know that $\mathcal{W}'(f) \in \mathcal{M}_{N_1} + \cdots + \mathcal{M}_{N_n}$. Hence, we apply the induction hypothesis and Lemma 21 to get $\mathcal{M}_{N_1} + \cdots + \mathcal{M}_{N_n} \subseteq \mathsf{SN}^{\rightarrow_{\mathsf{mul}}}$. Hence, $\mathcal{W}'(f) \in \mathsf{SN}^{\rightarrow_{\mathsf{mul}}}$.

Now notice that $\rightarrow_{\mathsf{mul}}$ strongly simulates $\rightarrow_{\mathsf{mul1}}$ through $\mathcal{W}'$, so by Theorem 14, $f \in \mathsf{SN}^{\rightarrow_{\mathsf{mul1}}}$.

Now that we know that $\rightarrow'_{\mathsf{mul}}$ is terminating on $\mathcal{M}_M$, we notice that the decreasing order on natural numbers strongly simulates $\rightarrow_{\mathsf{mul2}}$ and weakly simulates $\rightarrow_{\mathsf{mul1}}$ through the function that maps every $f \in \mathcal{M}_M$ to the natural number $f(M)$.

Hence, we can apply Corollary 18 to get $\mathcal{M}_M \subseteq \mathsf{SN}^{\rightarrow_{\mathsf{mul}}}$. □

Corollary 23 (Multi-Set termination) *Let f be a multi-set on $\mathcal{A}$.*
If for any $M \in f$, $M \in \mathit{SN}^{\rightarrow}$, then $f \in \mathit{SN}^{\rightarrow_{\mathit{mul}}}$.

Proof: Let $M_1, \ldots, M_n$ be the elements of $\mathcal{A}$ that are not mapped to 0 by f. Clearly, $f \in \mathcal{M}_{M_1} + \cdots + \mathcal{M}_{M_n}$. By Lemma 22, $\mathcal{M}_{M_1} \cup \ldots \mathcal{M}_{M_n} \subseteq \mathsf{SN}^{\rightarrow_{\mathsf{mul}}}$, and by Lemma 21, $\mathcal{M}_{M_1} + \cdots + \mathcal{M}_{M_n} \subseteq \mathsf{SN}^{\rightarrow_{\mathsf{mul}}}$, so $f \in \mathsf{SN}^{\rightarrow_{\mathsf{mul}}}$. □

1.5 Higher-order syntaxes and rewrite systems

We now deal with higher-order syntaxes, where the set $\mathcal{A}$ is recursively defined by a term syntax possibly involving variable binding and the reduction relation $\rightarrow$ is defined as a rewrite system. There are several ways to express those systems in a generic way, among which the Expression Reduction Systems (ERS) [Kha90], the Combinatory Reduction Systems (CRS) [Klo80], and the Higher-Order Systems (HRS) [Nip91]. In the rest of this report, we only use from those formalisms the notions of redex, sub-term and contextual

closure of the rewrite rules, as well as the notion of implicit substitution such as $M\{x = N\}$ (that denotes the term M in which every occurrence of the variable x has been replaced by the term N). All these definitions can be found in [Ter03].

Definition 13 (Conventions)
The symbol $\sqsubseteq$ denotes the sub-term relation and $\sqsubset$ denotes the strict sub-term relation (we also use $\sqsupseteq$ and $\sqsupset$ for the inverse relations).

By definition of terms, $\mathcal{A} = \mathsf{SN}^{\sqsupset}$.

For a rewrite system R, $\longrightarrow_{\mathsf{R}}$ denotes as usual the contextual closure of the relation that contains every instance of the rewrite rules of R.

We identify a rewrite rule h with the rewrite system $\{\mathsf{h}\}$ and for two rewrite systems R and R' we write R, R' for $\mathsf{R} \cup \mathsf{R}'$.

A *congruence* on $\mathcal{A}$ is an equivalence relation that is context-closed.

Lemma 24 $SN^{\longrightarrow_R \cup \sqsupset} = SN^{\longrightarrow_R}$.

Proof: This is a typical theorem that is usually proved classically (using for instance the postponing technique [Ter03]). We prove it constructively here. The left-to-right inclusion is trivial, by Remark 8. Now for the other direction, first notice that $\mathsf{SN}^{\sqsupset} = \mathcal{A}$. Because of the definition of a contextual closure, $\longrightarrow_{\mathsf{R}}$ strongly simulates $\longrightarrow_{\mathsf{R}}$ through $\sqsubseteq$. Also, it weakly simulates $\sqsupset$ through $\sqsubseteq$, so we may apply Corollary 18 and get $\forall N \in \mathsf{SN}^{\rightarrow_{\mathsf{R}}}, \forall M \in \mathcal{A}, M \sqsubseteq N \Rightarrow M \in \mathsf{SN}^{\rightarrow_{\mathsf{R}} \cup \sqsupset}$.
In particular, $\forall N \in \mathsf{SN}^{\rightarrow_{\mathsf{R}}}, M \in \mathsf{SN}^{\rightarrow_{\mathsf{R}} \cup \sqsupset}$. □

Notice that this result enables us to use a stronger induction principle: in order to prove $\forall M \in \mathsf{SN}^{\longrightarrow_{\mathsf{R}}}, P(M)$, it now suffices to prove

$$\forall M \in \mathsf{SN}^{\longrightarrow_{\mathsf{R}}}, (\forall N \in \mathcal{A}, (M \longrightarrow^{+}_{\mathsf{R}} N \vee N \sqsubset M) \Rightarrow P(N)) \Rightarrow P(M)$$

This induction principle is called the *transitive induction in* SN^{R} *with sub-terms* and is used in the following sections.

We briefly recall the various induction principles:
In order to prove $\forall M \in \mathsf{SN}^{\longrightarrow_{\mathsf{R}}}, P(M)$, it suffices to prove

- $\forall M \in \mathcal{A}, (\forall N \in \mathcal{A}, (M \longrightarrow_{\mathsf{R}} N) \Rightarrow P(N)) \Rightarrow P(M)$
  (raw induction in SN^{R}), or just
- $\forall M \in \mathsf{SN}^{\longrightarrow_{\mathsf{R}}}, (\forall N \in \mathcal{A}, (M \longrightarrow_{\mathsf{R}} N) \Rightarrow P(N)) \Rightarrow P(M)$
  (induction in SN^{R}), or just
- $\forall M \in \mathsf{SN}^{\longrightarrow_{\mathsf{R}}}, (\forall N \in \mathcal{A}, (M \longrightarrow^{+}_{\mathsf{R}} N) \Rightarrow P(N)) \Rightarrow P(M)$
  (transitive induction in SN^{R}), or even
- $\forall M \in \mathsf{SN}^{\longrightarrow_{\mathsf{R}}}, (\forall N \in \mathcal{A}, (M \longrightarrow^{+}_{\mathsf{R}} N \vee N \sqsubset M) \Rightarrow P(N)) \Rightarrow P(M)$
  (transitive induction in SN^{R} with sub-terms)

Definition 14 SN^{R} henceforth denotes $\mathsf{SN}^{\longrightarrow_{\mathsf{R}} \cup \sqsupset} = \mathsf{SN}^{\longrightarrow_{\mathsf{R}}}$.

2 Of the difficulty of relating the terminations of λ-calculi

In the rest of this report we develop techniques that were originally designed for deriving strong normalisation results from the strong normalisation of typed λ-calculus [Bar84].

The first one turns out to be more general and can be applied to any rewrite system. It is a useful refinement of the simulation technique, but the main theorem of the technique only holds in classical logic.

The second technique holds in intuitionistic logic, apart maybe from one external result, of which the provability in intuitionitic logic remains to be checked. The technique was originally designed to prove the strong normalisation of calculi with explicit substitutions, such as $\lambda\mathsf{x}$ [BR95].

We call *calculus with explicit substitutions* a calculus that uses a set of variables, denoted $x, y, \ldots$, and one of its constructors is the following one:
If M and N are terms, then $\langle M/x\rangle N$ is a term, where x is bound in N. The construct is called an *explicit substitution* and M is called its *body*.

Of course, the technique is likely to be adapted to other frameworks, which could use De Bruijn indices [Bar84] or explicit substitutions with additional parameters, but the above framework is plainly sufficient for the examples treated hereafter.

Among the calculi with explicit substitutions to which the techniques can be applied are the intuitionistic sequent calculi [Gen35].

The notion of computation in sequent calculi is Cut-elimination: the proof of a sequent may be simplified by eliminating the applications of the Cut-rule, so that a sequent which is provable with the Cut-rule is provable without.

It turns out that the most natural typing rule for an explicit substitution as expressed above is precisely a Cut-rule. From that remark, many techniques aimed at proving normalisation results about calculi of explicit substitutions actually apply to systems with Cut-rules such as sequent calculi. In other words, termination of cut-elimination processes can often be derived from termination of explicit substitution calculi.

Of course, in the case of sequent calculi, termination of Cut-elimination relies only on the strong normalisation of typed terms.

Another notion tackles the strong normalisation of terms with explicit substitutions that are not necessarily typed: the property called *Preservation of Strong Normalisation* (PSN) [BBLRD96]. It concerns syntactic extensions of λ-calculus with their own reduction relations and states that if a λ-term is strongly normalising for the β-reduction, then it is still strongly normalising when considered as a term of the extended calculus undergoing the reductions of the latter. In other words, the reduction relation should not be too big, although it is often required to be big enough to simulate β-reduction. It is typically the case of $\lambda\mathsf{x}$ [BR95], which we shall investigate shortly.

The definition of the PSN property can be slightly generalised for calculi in which λ-calculus can be *embedded* (by a one-to-one translation, say A) rather than just included. In that case PSN states that if a λ-term is strongly normalising, then its encoding is also strongly normalising. This is the case for the explicit substitution calculus $\lambda\mathsf{lxr}$ introduced in [KL05] which requires terms to be linear and hence is not a syntactic extension of λ-calculus. Figure 2 shows the two situations, with the example of $\lambda\mathsf{x}$ and $\lambda\mathsf{lxr}$.

The basic idea in proving that a term M of a calculus with explicit substitutions is SN

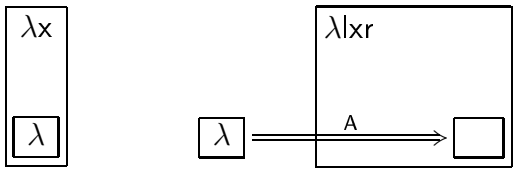


Figure 2: Standard and generalised situations for stating PSN

is to use Corollary 18, that is, simulating M's reductions by β-reductions of a strongly normalising λ-term $\mathsf{H}(M)$.

For PSN, if $M = \mathsf{A}(t)$ where t is the λ-term known to be SN^β by hypothesis, then we would take $\mathsf{H}(M) = t$.

For sequent calculus, it would be a typed (and hence strongly normalising) λ-term that denotes a proof in natural deduction of the same sequent (using Curry-Howard correspondence). The idea of simulating Cut-elimination by β-reductions has been investigated in [Zuc74].

There is one problem in doing so: an encoding into λ-calculus that allows the simulation needs to interpret explicit substitutions by implicit substitutions such as $t\{x = u\}$. But should x not be free in t, all reduction steps taking place within the term of which u is the encoding would not induce any β-reduction in $t\{x = u\}$.

Therefore, the sub-system that is only weakly simulated, i.e. the one consisting of all the reductions that are not necessarily simulated by at least one β-reduction, is too big to be proved terminating (and very often it is not).

The two techniques developed hereafter are designed to overcome this problem, in a somewhat general setting. The two aforementioned calculi with explicit substitutions $\lambda\mathsf{x}$ and $\lambda\mathsf{lxr}$ respectively illustrate how each can be applied and can provide in particular a proof of the PSN property.

In order to compare the examples with λ-calculus, we briefly recall the latter. The syntax is defined as follows:

$$M, N ::= x \mid \lambda x.M \mid M\ N$$

β-reduction is defined as the following rule:

$$(\lambda x.M)\ N \longrightarrow_\beta\ M\{x = N\}$$

The first three inference rules of Figure 3 define the derivable judgements of the simply-typed λ-calculus, which we note as $\Gamma \vdash_{\mathsf{NJ}} M : A$. When the two bottom inference rules are added, we obtain a typing system characterising SN^β, and we note those derivable judgements as $\Gamma \vdash_{\mathsf{NJ}\cap} M : A$.

The following theorem has been proved in [CD78]:

Theorem 25 (Strong Normalisation of λ-calculus)
$\Gamma \vdash_{\mathsf{NJ}\cap} M : A$ if and only if $M \in \mathsf{SN}^\beta$.

A proof of the weaker statement that simply-typed λ-calculus is strongly normalising can be found, for example, in [Bar84].

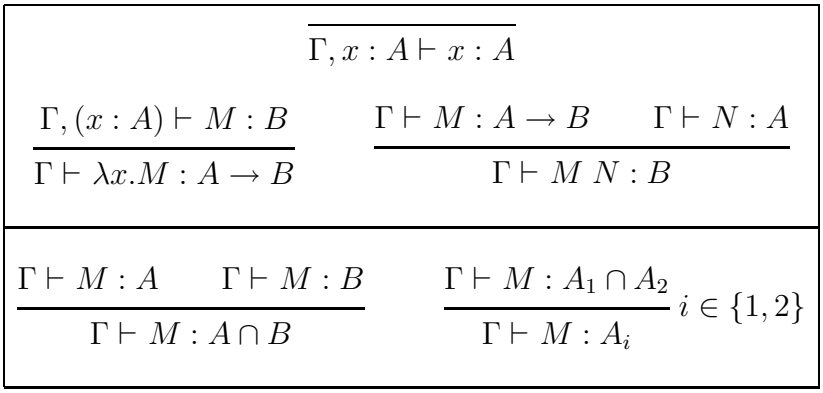


Figure 3: Typing rules for λ-calculus

3 The safeness and minimality technique

Given a rewrite system R on a set of terms $\mathcal{A}$, the *safeness and minimality technique* presents two subsystems minR and safeR satisfying $\longrightarrow_{\mathsf{safeR}} \subseteq \longrightarrow_{\mathsf{minR}} \subseteq \longrightarrow_{\mathsf{R}}$ and $\mathsf{SN}^{\mathsf{minR}} = \mathsf{SN}^{\mathsf{R}}$.

The intuitive idea is that a reduction step is *minimal* if all the (strict) sub-terms of the redex are in SN^{R}. Theorem 27 says that in order to prove that $\longrightarrow_{\mathsf{R}}$ is terminating, we can restrict our attention to minimal reductions only, without loss of generality.

Similarly, a reduction step is *safe* if the redex itself is in SN^{R}, which is a stronger requirement than minimality. Theorem 28 says that, whatever R, safe reductions always terminate.

Those ideas are made precise in the following definition:

Definition 15 (Safe and Minimal reductions) Given two rewrite systems h and R satisfying $\longrightarrow_{\mathsf{h}} \subset \longrightarrow_{\mathsf{R}}$,

- the (R-)*minimal* h-system is given by the following scheme of rules:

$$\mathsf{minh} : M \longrightarrow N \qquad \text{for every } M \longrightarrow_{\mathsf{h}} N \text{ such that for all } P \sqsubset M,\ P \in \mathsf{SN}^{\mathsf{R}}$$

- the (R-)*safe* h-system is given by the following scheme of rules:

$$\mathsf{safeh} : M \longrightarrow N \qquad \text{for every } M \longrightarrow_{\mathsf{h}} N \text{ such that } M \in \mathsf{SN}^{\mathsf{R}}$$

In both rules we could require $M \longrightarrow_{\mathsf{h}} N$ to be a root reduction so that M *is* the redex, but although the rules above seem stronger than that, they have the same contextual closure, so we consider the definition above which is the simplest.

Notice that being safe is stronger than being minimal as we have:
$\longrightarrow_{\mathsf{safeh}} \subseteq \longrightarrow_{\mathsf{minh}} \subseteq \longrightarrow_{\mathsf{h}} \subseteq \longrightarrow_{\mathsf{R}}$.

We also say that a reduction step $M \longrightarrow_{\mathsf{h}} N$ is safe (resp. minimal) if $M \longrightarrow_{\mathsf{safeh}} N$ (resp. $M \longrightarrow_{\mathsf{minh}} N$) and that it is unsafe if not.

Obviously if $\longrightarrow_{\mathsf{h}}$ is finitely branching, then so are $\longrightarrow_{\mathsf{safeh}}$ and $\longrightarrow_{\mathsf{minh}}$.

Remark 26 We shall constantly use the following facts:

1. $\longrightarrow_{\mathsf{min(safeh)}} = \longrightarrow_{\mathsf{safe(minh)}} = \longrightarrow_{\mathsf{safeh}}$

2. $\longrightarrow_{\mathsf{safe}(\mathsf{h},\mathsf{h}')} = \longrightarrow_{\mathsf{safeh},\mathsf{safeh}'}$

3. $\longrightarrow_{\mathsf{min}(\mathsf{h},\mathsf{h}')} = \longrightarrow_{\mathsf{minh},\mathsf{minh}'}$

Theorem 27 $SN^{minR} = SN^{R}$
In other words, in order to prove that a term is strongly normalising, it suffices to prove that it is strongly normalising for minimal reductions only. This theorem holds in intuitionistic logic.

Proof: The right-to-left inclusion is trivial. We now prove that $\mathsf{SN}^{\mathsf{minR}} \subseteq \mathsf{SN}^{\mathsf{R}}$, by transitive induction in $\mathsf{SN}^{\mathsf{minR}}$ with sub-terms.

Let $M \in \mathsf{SN}^{\mathsf{minR}}$, we have the induction hypothesis that
$\forall N, (M \longrightarrow^{+}_{\mathsf{minR}} N \vee N \sqsubset M) \Rightarrow N \in \mathsf{SN}^{\mathsf{R}}$.

We want to prove that $M \in \mathsf{SN}^{\mathsf{R}}$, so it suffices to check that if $M \longrightarrow_{\mathsf{R}} N$, then $N \in \mathsf{SN}^{\mathsf{R}}$.

We first show that in that case $M \longrightarrow_{\mathsf{minR}} N$. Let Q be the R-redex in M, and let $P \sqsubset Q$. We have $P \sqsubset M$. By the induction hypothesis we get $P \in \mathsf{SN}^{\mathsf{R}}$, so Q is a minR-redex. By contextual closure of minimal reduction, $M \longrightarrow_{\mathsf{minR}} N$.

Again by the induction hypothesis, we get $N \in \mathsf{SN}^{\mathsf{R}}$ as required. □

Theorem 28 $SN^{safeR} = \mathcal{A}$
In other words, safe reductions always terminate. This theorem holds in intuitionistic logic.

Proof: Consider the multi-sets of (R)-strongly normalising terms, and consider the multi-set reductions induced by the reductions $(\longrightarrow_{\mathsf{R}} \cup \sqsupset)^{+}$ on strongly normalising terms. By Corollary 23, these multi-set reductions are terminating.

Considering the mapping ϕ of every term to the multi-set of its R-strongly normalising sub-terms, we can check that the multi-set reductions strongly simulate the safe reductions through ϕ. Hence, from Theorem 14, we get that safe reductions are terminating. □

Now the aim of the safeness and minimality technique is to prove the strong normalisation of a system R.

We obtain this by the following theorem, which only holds in classical logic. Indeed, it relies on the fact that for the rewrite system R, for all term M we have either $M \in \mathsf{SN}^{\mathsf{R}}$ or $M \notin \mathsf{SN}^{\mathsf{R}}$. This instance of the Law of Excluded Middle is in general not decidable.

Theorem 29 *Given a system R, if we find a subsystem R' satisfying $\longrightarrow_{safeR} \subseteq \longrightarrow_{R'} \subseteq \longrightarrow_{minR}$, such that we have:*

- *the strong simulation of $\longrightarrow_{minR} \setminus \longrightarrow_{R'}$ in a strongly normalising calculus, through a total relation $\mathcal{Q}$*
- *the weak simulation of $\longrightarrow_{R'}$ through $\mathcal{Q}$*
- *the strong normalisation of $\longrightarrow_{R'}$*

then R is strongly normalising.

Proof: This is a direct corollary of Corollary 18. □

$$\mathsf{x}: \begin{cases} \mathsf{B} & (\lambda x.M)\ N \longrightarrow \langle N/x\rangle M \\ \mathsf{Abs} & \langle N/x\rangle \lambda y.M \longrightarrow \lambda y.\langle N/x\rangle M \\ \mathsf{App} & \langle N/x\rangle M_1\ M_2 \longrightarrow \langle N/x\rangle M_1\ \langle N/x\rangle M_2 \\ \mathsf{VarK} & \langle N/x\rangle y \longrightarrow y \\ \mathsf{VarI} & \langle N/x\rangle x \longrightarrow N \end{cases}$$

Figure 4: Reduction rules for $\lambda\mathsf{x}$

Now notice the particular case of the technique when we take $\mathsf{R}' = \mathsf{safeR}$. By Theorem 28 we would directly have its strong normalisation. Unfortunately, this definition is often too coarse, that is to say, the relation $\longrightarrow_{\mathsf{R}'}$ is to small, so that $\longrightarrow_{\mathsf{minR}} \setminus \longrightarrow_{\mathsf{R}''}$ is often too big to be strongly simulated.

Hence, in order to define R', we use the safeness criterion, but the precise definition depends on the calculus that is being treated. We give the examples of $\lambda\mathsf{x}$ and $\overline{\lambda}$. The proofs in these examples use classical logic.

3.1 Example: $\lambda\mathsf{x}$

$\lambda\mathsf{x}$ [BR95] is the syntactic extension of λ-calculus with the aforementioned explicit substitution operator:

$$M, N ::= x|\ \lambda x.M|\ M\ N|\ M\{x = N\}$$

Its reduction system reduces β-redexes into explicit substitutions which are thence evaluated, as shown in Figure 4.

The first four inference rules of Figure 5 define the derivable judgements of simply-typed $\lambda\mathsf{x}$, which we note as $\Gamma \vdash_{\mathsf{NJCut}} M : A$. When the three bottom inference rules are added, we obtain a typing system characterising $\mathsf{SN}^{B,\mathsf{x}}$ [LLD$^+$04], and we note those derivable judgements as $\Gamma \vdash_{\mathsf{NJCut}\cap} M : A$. The following theorem is proved in [LLD$^+$04]:

$$\frac{}{\Gamma, x : A \vdash x : A} \qquad \frac{\Gamma \vdash P : A \qquad \Gamma, (x : A) \vdash M : C}{\Gamma \vdash \langle P/x\rangle M : C}$$

$$\frac{\Gamma, (x : A) \vdash M : B}{\Gamma \vdash \lambda x.M : A \to B} \qquad \frac{\Gamma \vdash M : A \to B \qquad \Gamma \vdash N : A}{\Gamma \vdash M\ N : B}$$

$$\frac{\Gamma \vdash M : A \qquad \Gamma \vdash M : B}{\Gamma \vdash M : A \cap B} \qquad \frac{\Gamma \vdash M : A_1 \cap A_2}{\Gamma \vdash M : A_i}\ i \in \{1, 2\}$$

$$\frac{\Gamma \vdash M : A \qquad \Delta \vdash N : B \qquad x \notin \Gamma}{\Gamma \vdash \langle N/x\rangle M : A}$$

Figure 5: Typing rules for $\lambda\mathsf{x}$

Theorem 30 (Capturing strongly normalising terms)
If $M \in \mathcal{SN}^{B,x}$ then there is a Γ and a A such that $\Gamma \vdash_{\mathcal{NJCut}\cap} M : A$.

In the same paper, the converse (typed terms are strongly normalising) has been proved by a reducibility technique. We show here that one application of the Safeness and Minimality technique, apart from PSN, is to derive this result from the strong normalisation of λ-calculus with intersection types (Theorem 25).

In this example we take $\mathsf{R}' = \mathsf{safe}B, \mathsf{minx}$.

Lemma 31 *$\longrightarrow_{\mathsf{safe}B,\mathsf{x}}$ is terminating.*

Proof: We use for that a *lexicographic path ordering* [KL80] based on the following infinite first-order signature and its precedence relation:

$$\mathsf{c}^M < \mathsf{succ}(-) < \mathsf{bi}(-,-) < \mathsf{sub}(-,-)$$

where for every $M \in \mathsf{SN}^{B,\mathsf{x}}$ there is a constant c^M. Those constants are all below $\mathsf{succ}()$, and the precedence between them is given by $\mathsf{c}^N < \mathsf{c}^M$ if and only if $M \longrightarrow^+_{B,\mathsf{x}} N$ or $N \sqsubset M$. By Remark 24, the precedence is well-founded (terminating).

Encode $\lambda\mathsf{x}$ as follows:

$\mathcal{P}(M)$	$=$	c^M	if $M \in \mathsf{SN}^{B,\mathsf{x}}$
otherwise			
$\mathcal{P}(\lambda x.M)$	$=$	$\mathsf{succ}(\mathcal{P}(M))$	
$\mathcal{P}(M\ N)$	$=$	$\mathsf{bi}(\mathcal{P}(M), \mathcal{P}(N))$	
$\mathcal{P}(\langle N/x\rangle M)$	$=$	$\mathsf{sub}(\mathcal{P}(N), \mathcal{P}(M))$	

It is quite easy to check that $(\mathsf{safe}B), \mathsf{x}$-reductions decrease $\mathcal{P}()$, so they are terminating. □

Now consider the following encoding in λ:

$\mathsf{H}(x)$	$=$	x	
$\mathsf{H}(\lambda x.M)$	$=$	$\lambda x.\mathsf{H}(M)$	
$\mathsf{H}(M\ N)$	$=$	$\mathsf{H}(M)\ \mathsf{H}(N)$	
$\mathsf{H}(\langle N/x\rangle M)$	$=$	$\mathsf{H}(M)\{x = \mathsf{H}(N)\}$	if $N \in \mathsf{SN}^{B,\mathsf{x}}$
	$=$	$(\lambda x.\mathsf{H}(M))\ \mathsf{H}(N)$	if $N \notin \mathsf{SN}^{B,\mathsf{x}}$

Lemma 32

1. *If $M \longrightarrow_{\mathsf{min}B}$ N is unsafe then $\mathsf{H}(M) \longrightarrow_\beta \mathsf{H}(N)$*
2. *If $M \longrightarrow_{\mathsf{min}B}$ N is safe then $\mathsf{H}(M) \longrightarrow^*_\beta \mathsf{H}(N)$*
3. *If $M \longrightarrow_{\mathsf{min}\mathsf{x}}$ N then $\mathsf{H}(M) = \mathsf{H}(N)$*

Corollary 33 *If $\mathsf{H}(M) \in \mathcal{SN}^\beta$ then $M \in \mathcal{SN}^{B,x}$.*

Proof: Direct application of Theorem 29. □

This results has two obvious corollaries:

Considering that on pure terms (that is, substitution-free terms), the encoding into λ-calculus is the identity, this gives directly the PSN property for $\lambda\mathsf{x}$.

Corollary 34 (Preservation of Strong Normalisation)
If $t \in \mathcal{SN}^{\beta}$ then $t \in \mathcal{SN}^{B,\mathsf{x}}$.

It turns out that the above encoding generally preserves typing. Hence, if the typing system considered in λ-calculus implies strong normalisation, then the original $\lambda\mathsf{x}$-term is also strongly normalising, by Corollary 33. For instance, we have the following theorem:

Theorem 35

1. *If $\Gamma \vdash_{\mathcal{NJ}Cut} M : A$ then $\Gamma \vdash_{\mathcal{NJ}} \mathcal{H}(M) : A$, so $M \in \mathcal{SN}^{B,\mathsf{x}}$.*
2. *If $\Gamma \vdash_{\mathcal{NJ}Cut\cap} M : A$ then $\Gamma \vdash_{\mathcal{NJ}\cap} \mathcal{H}(M) : A$, so $M \in \mathcal{SN}^{B,\mathsf{x}}$.*

Often, that kind of strong normalisation result is derived from the PSN property by lifting the explicit substitutions into β-redexes [Her95], but this is precisely what the encoding does in the necessary places, so that Corollary 33 is a shortcut of Herbelin's technique.

Notice the subtlety of the definition for the encoding of an explicit substitution:

1. As we have already said, always encoding explicit substitutions as implicit substitutions leads to the weak simulation of too many B-steps, so that the system that is only weakly simulated is too big to be proved terminating.
2. On the other hand, always raising $\langle N/x\rangle M$ into a β-redex would be too strong, because the substitution $\langle N/x\rangle$ can be propagated into the sub-terms of M but the β-redex cannot be moved around, so the simulation theorem would not hold.
3. Hence, we needed to define an encoding that is a compromise of those two, and the side-condition $N \in \mathsf{SN}^{B,\mathsf{x}}$ is precisely the criterion we need:
   - First, the satisfiability of the condition may only evolve in one direction, as it may only become satisfied by some reduction within N, and not the other way around. If it does so, we can simulate this step by reducing the β-redex.
   - Now if $N \notin \mathsf{SN}^{B,\mathsf{x}}$, then the substitution is lifted into a β-redex and for the same reason as in point 2 we cannot simulate the propagation of $\langle N/x\rangle$. So we need to prove that we need not consider reduction steps that propagate a substitution of which the body is not strongly normalising. This is *precisely* the point of minimal reduction: Theorem 27 says that in order to prove a strong normalisation result, we may assume that all sub-terms of the redex are strongly normalising.
   - If on the contrary $N \in \mathsf{SN}^{B,\mathsf{x}}$, then we can indeed simulate its propagation, but for the same reason as in point 1, reduction steps within N might only be weakly simulated, but these are precisely what we call safe reductions and we have proved above that they (together with x-reduction) terminate.

3.2 Example: $\overline{\lambda}$

Another example of how this techniques applies is Herbelin's $\overline{\lambda}$, for which PSN has longer proofs in [DU03, Kik04]. Since $\overline{\lambda}$ can be typed by a version called LJT of the intuitionistic sequent calculus and the technique provides again a type-preserving encoding of $\overline{\lambda}$ into the simply-typed λ-calculus, we thus prove the strong normalisation of Cut-elimination in LJT.

The syntax of Herbelin's calculus is defined as follows:

$$\begin{array}{rcl} M, N, A, B & ::= & \lambda x.M \mid x\ l \mid M\ l \mid \langle M/x\rangle N \\ l, l' & ::= & [] \mid M :: l \mid l@l' \mid \langle M/x\rangle l \end{array}$$

$\lambda x.M$ and $\langle N/x\rangle M$ bind x in M, and $\langle M/x\rangle l$ binds x in l, thus defining the free variables of terms and lists as well as α-conversion. We use Barendregt's convention that no variable is free and bound in a term in order to avoid variable capture when reducing it.

The reduction rules of $\overline{\lambda}$ are defined in Figure 6, the typing rules are defined in Figure 7.

$$\mathsf{B}\quad (\lambda x.M)\ (N :: l) \longrightarrow (\langle N/x\rangle M)\ l$$

System x:

$$\left\{\begin{array}{lll}
\mathsf{B1} & M\ [] & \longrightarrow M \\
\mathsf{B2} & (x\ l)\ l' & \longrightarrow x\ (l@l') \\
\mathsf{B3} & (M\ l)\ l' & \longrightarrow M\ (l@l') \\
\\
\mathsf{A1} & (M :: l')@l & \longrightarrow M :: (l'@l) \\
\mathsf{A2} & []@l & \longrightarrow l \\
\mathsf{A3} & (l@l')@l'' & \longrightarrow l@(l'@l'') \\
\\
\mathsf{C1} & \langle P/y\rangle \lambda x.M & \longrightarrow \lambda x.\langle P/y\rangle M \\
\mathsf{C2} & \langle P/y\rangle (y\ l) & \longrightarrow P\ \langle P/y\rangle l \\
\mathsf{C3} & \langle P/y\rangle (x\ l) & \longrightarrow x\ \langle P/y\rangle l \\
\mathsf{C4} & \langle P/y\rangle (M\ l) & \longrightarrow \langle P/y\rangle M\ \langle P/y\rangle l \\
\\
\mathsf{D1} & \langle P/y\rangle [] & \longrightarrow [] \\
\mathsf{D2} & \langle P/y\rangle (M :: l) & \longrightarrow (\langle P/y\rangle M) :: (\langle P/y\rangle l) \\
\mathsf{D3} & \langle P/y\rangle (l@l') & \longrightarrow (\langle P/y\rangle l)@(\langle P/y\rangle l')
\end{array}\right.$$

Figure 6: Reduction Rules for $\overline{\lambda}$

Typically, the case of $\overline{\lambda}$ is one of those where the syntax does not include that of λ-calculus, but the latter can be encoded [Her95]. Indeed, it is well-known that the syntax of λ-calculus can also be described as follows:

$$\begin{array}{l} P := \lambda x.M \\ M, N, A, B := P \mid x\ \overrightarrow{M} \mid P\ N\ \overrightarrow{M} \end{array}$$

where $\overrightarrow{M}$ represents a list of "M-terms" of arbitrary length.

The encoding, given in Figure 8, is threefold, one function $\mathsf{A}_\lambda()$ for the "P-terms", a second one, $\mathsf{A}()$, for the "M-terms", and a third one, $\mathsf{A}_l()$, for lists of "M-terms":

$$\frac{\Gamma; A \vdash_{\mathsf{LJT}} l : B \quad (x:A) \in \Gamma}{\Gamma \vdash_{\mathsf{LJT}} x\ l : B}\ \mathsf{Cont}_x \qquad \frac{\Gamma \vdash_{\mathsf{LJT}} A : s}{\Gamma; A \vdash_{\mathsf{LJT}} [] : A}\ \mathsf{axiom}$$

$$\frac{\Gamma, (x:A) \vdash_{\mathsf{LJT}} M : B}{\Gamma \vdash_{\mathsf{LJT}} \lambda x.M : A \to B} \to \mathsf{r} \qquad \frac{\Gamma \vdash_{\mathsf{LJT}} M : A \qquad \Gamma; B \vdash_{\mathsf{LJT}} l : C}{\Gamma; A \to B \vdash_{\mathsf{LJT}} M :: l : C} \to \mathsf{l}$$

$$\frac{\Gamma \vdash_{\mathsf{LJT}} M : A \qquad \Gamma; A \vdash_{\mathsf{LJT}} l : B}{\Gamma \vdash_{\mathsf{LJT}} M\ l : B}\ \mathsf{Cut}_3 \qquad \frac{\Gamma; C \vdash_{\mathsf{LJT}} l' : A \qquad \Gamma; A \vdash_{\mathsf{LJT}} l : B}{\Gamma; C \vdash_{\mathsf{LJT}} l'@l : B}\ \mathsf{Cut}_1$$

$$\frac{\Gamma \vdash_{\mathsf{LJT}} P : A \qquad \Gamma, (x:A) \vdash_{\mathsf{LJT}} M : C}{\Gamma \vdash_{\mathsf{LJT}} \langle P/x \rangle M : C}\ \mathsf{Cut}_4 \qquad \frac{\Gamma \vdash_{\mathsf{LJT}} P : A \qquad \Gamma, (x:A); B \vdash_{\mathsf{LJT}} l : C}{\Gamma; B \vdash_{\mathsf{LJT}} \langle P/x \rangle l : C}\ \mathsf{Cut}_2$$

Figure 7: Typing rules for $\overline{\lambda}$

$$\begin{array}{|lcl|}
\hline
\mathsf{A}_\lambda(\lambda x.M) & = & \lambda x.\mathsf{A}(M) \\
\hline
\mathsf{A}(P) & = & \mathsf{A}_\lambda(P) \\
\mathsf{A}(x\ \overrightarrow{M}) & = & x\ \mathsf{A}_l(\overrightarrow{M}) \\
\mathsf{A}(P\ N\ \overrightarrow{M}) & = & \mathsf{A}_\lambda(P)\ (\mathsf{A}(N) :: \mathsf{A}_l(\overrightarrow{M})) \\
\hline
\mathsf{A}_l(\overrightarrow{\emptyset}) & = & [] \\
\mathsf{A}_l(\overrightarrow{N_1\ \ldots\ N_i}) & = & \mathsf{A}(N_1) :: \mathsf{A}_l(\overrightarrow{N_2\ \ldots\ N_i}) \\
\hline
\end{array}$$

Figure 8: Encoding λ-calculus into $\overline{\lambda}$

Remark 36 $\mathsf{A}(M)$ is an x-normal form

Lemma 37 $\langle A(M)/x \rangle A(N) \longrightarrow^*_{\mathsf{x}} A(N\{x = M\})$

Proof: By induction on N. □

Finally, we conclude that β-reduction is simulated by B, x, so that λ-calculus can be considered as a sub-calculus of $\overline{\lambda}$.

Theorem 38 *If* $M \longrightarrow_\beta N$ *then* $A(M) \longrightarrow^+_{B,\mathsf{x}} A(N)$

Proof: By induction on M. □

Now we prove PSN (and SN of typed terms) for $\overline{\lambda}$ with the safeness and minimality technique. Again, we consider a first-order syntax equipped with a *lexicographic path ordering* based on the following precedence:

$$\mathsf{c}^M < \mathsf{succ}(-) < \mathsf{bi}(-,-) < \mathsf{sub}(-,-)$$

where for every $M \in \mathsf{SN}^{B,\mathsf{x}}$ (resp. $l \in \mathsf{SN}^{B,\mathsf{x}}$) there is a constant c^M (resp. c^l). Those constants are all below $\mathsf{succ}()$, and the precedence between them is given by $\mathsf{c}^N < \mathsf{c}^M$ if

and only if $M \longrightarrow^{+}_{B,\mathsf{x}} N$ or $N \sqsubset M$ (and similarly for lists). The precedence is hence well-founded.

The encoding goes as follows:

$\mathcal{P}(M)$	$=$	c^M	if $M \in \mathsf{SN}^{B,\mathsf{x}}$
otherwise			
$\mathcal{P}(\lambda x.M)$	$=$	$\mathsf{bi}(\mathcal{P}(A), \mathcal{P}(M))$	
$\mathcal{P}(x\ l)$	$=$	$\mathsf{succ}(\mathcal{Q}(l))$	
$\mathcal{P}(M\ l)$	$=$	$\mathsf{bi}(\mathcal{Q}(l), \mathcal{P}(M))$	
$\mathcal{P}(\langle M/x\rangle N)$	$=$	$\mathsf{sub}(\mathcal{P}(M), \mathcal{P}(N))$	
$\mathcal{Q}(l)$	$=$	c^l	if $l \in \mathsf{SN}^{B,\mathsf{x}}$
otherwise			
$\mathcal{Q}(M :: l)$	$=$	$\mathsf{bi}(\mathcal{P}(M), \mathcal{Q}(l))$	
$\mathcal{Q}(l@l')$	$=$	$\mathsf{bi}(\mathcal{Q}(l), \mathcal{Q}(l'))$	
$\mathcal{Q}(\langle M/x\rangle l)$	$=$	$\mathsf{sub}(\mathcal{P}(M), \mathcal{Q}(l))$	

Lemma 39

1. *If* $M \longrightarrow_{\mathsf{safe}B,\mathsf{x}} N$ *then* $\mathcal{P}(M) > \mathcal{P}(N)$.
2. *If* $l \longrightarrow_{\mathsf{safe}B,\mathsf{x}} l'$ *then* $\mathcal{Q}(l) > \mathcal{Q}(l')$.

Proof: We first check root reductions.
Clearly, if $M, l \in \mathsf{SN}^{B,\mathsf{x}}$ the Lemma holds, and this covers the case of safe reductions.
Also, when $N, l' \in \mathsf{SN}^{B,\mathsf{x}}$ the Lemma holds as well.
The remaining cases are when $\mathcal{P}(M), \mathcal{Q}(l)$ and $\mathcal{P}(N), \mathcal{Q}(l')$ are not constants.

For $\mathsf{B1}, \mathsf{A2}$, the term $\mathcal{P}(N)$ (resp. $\mathcal{Q}(l')$) is a sub-term of $\mathcal{P}(M)$ (resp. $\mathcal{Q}(l)$).

For $\mathsf{B2}, \mathsf{B3}, \mathsf{A1}$, the arguments of $\mathsf{bi}(,)$ decrease in the lexicographic order.

For $\mathsf{C}i's, \mathsf{D}i's$, the symbol at the root of $\mathcal{P}(N)$ (resp. $\mathcal{Q}(l')$) is strictly inferior to that of $\mathcal{P}(M)$ (resp. $\mathcal{Q}(l)$), so we only have to check that the direct sub-terms of $\mathcal{P}(N)$ (resp. $\mathcal{Q}(l')$) are smaller than $\mathcal{P}(M)$ (resp. $\mathcal{Q}(l)$). Clearly, it is the case for all sub-terms that are constants (namely, those encodings of strongly normalising sub-terms of N or l'). For those that are not, it is a routine check on every rule.

The contextual closure is a straightforward induction on M, l:
Again, if $M, l \in \mathsf{SN}^{B,\mathsf{x}}$ or $N, l' \in \mathsf{SN}^{B,\mathsf{x}}$, the Lemma holds;
otherwise, if the reduction is a $\mathsf{safe}B, \mathsf{x}$-reduction in a direct sub-term of M or l, it suffices to use the induction hypothesis on that sub-term. □

Corollary 40 *The reduction relation* $\longrightarrow_{\mathsf{safe}B,\mathsf{x}}$ *is terminating.*

Now we encode $\overline{\lambda}$ in λ-calculus as follows:

$$
\begin{array}{lcll}
\mathsf{H}(\lambda x.M) & = & \lambda x.\mathsf{H}(M) & \\
\mathsf{H}(x\ l) & = & \mathsf{H}^z(l)\{z = x\} & x \text{ fresh} \\
\mathsf{H}(M\ l) & = & \mathsf{H}^z(l)\{z = \mathsf{H}(M)\} & z \text{ fresh} \\
\mathsf{H}(\langle M/x\rangle N) & = & \mathsf{H}(N)\{x = \mathsf{H}(M)\} & \text{if } M \in \mathsf{SN}^{B,\mathsf{x}} \\
\mathsf{H}(\langle M/x\rangle N) & = & (\lambda x.\mathsf{H}(N))\ \mathsf{H}(M) & \text{if } M \notin \mathsf{SN}^{B,\mathsf{x}} \\
\hline
\mathsf{H}^y([]) & = & y & \\
\mathsf{H}^y(M :: l) & = & \mathsf{H}^z(l)\{z = y\ \mathsf{H}(M)\} & z \text{ fresh} \\
\mathsf{H}^y(l@l') & = & \mathsf{H}^z(l')\{z = \mathsf{H}^y(l)\} & z \text{ fresh} \\
\mathsf{H}^y(\langle M/x\rangle l) & = & \mathsf{H}^y(l)\{x = \mathsf{H}(M)\} & \text{if } M \in \mathsf{SN}^{B,\mathsf{x}} \\
\mathsf{H}^y(\langle M/x\rangle l) & = & (\lambda x.\mathsf{H}^y(l))\ \mathsf{H}(M) & \text{if } M \notin \mathsf{SN}^{B,\mathsf{x}}
\end{array}
$$

Remark 41 For all y and l, $y \in FV(\mathsf{H}^y(l))$

Lemma 42

1. *If* $M \longrightarrow_{minB} N$ *is unsafe then* $\mathsf{H}(M) \longrightarrow_\beta \mathsf{H}(N)$
   If $l \longrightarrow_{minB} l'$ *is unsafe then* $\mathsf{H}^y(l) \longrightarrow_\beta \mathsf{H}^y(l')$
2. *If* $M \longrightarrow_{minB} N$ *is safe then* $\mathsf{H}(M) \longrightarrow^*_\beta \mathsf{H}(N)$
   If $l \longrightarrow_{minB} l'$ *is safe then* $\mathsf{H}^y(l) \longrightarrow^*_\beta \mathsf{H}^y(l')$
3. *If* $M \longrightarrow_{minx} N$ *then* $\mathsf{H}(M) = \mathsf{H}(N)$
   If $l \longrightarrow_{minx} l'$ *then* $\mathsf{H}^y(l) = \mathsf{H}^y(l')$

Corollary 43 *If* $\mathsf{H}(M) \in \mathsf{SN}^\beta$ *(resp.* $\mathsf{H}^y(l) \in \mathsf{SN}^\beta$*) then* $M \in \mathsf{SN}^{B,x}$ *(resp.* $l \in \mathsf{SN}^{B,x}$*).*

Proof: Direct application of Theorem 29. □

Now notice that $\mathsf{H} \cdot \mathsf{A} = \mathsf{Id}$, so that we conclude the following:

Corollary 44 (Preservation of Strong Normalisation)
If $t \in \mathsf{SN}^\beta$ *then* $\mathsf{A}(t) \in \mathsf{SN}^{B,x}$.

Notice that the preservation of types can be easily shown:

Remark 45

1. If $\Gamma \vdash_{\mathsf{LJT}} M : A$ then $\Gamma \vdash_{\mathsf{NJ}} \mathsf{H}(M) : A$
2. If $\Gamma; B \vdash_{\mathsf{LJT}} l : A$ then $\Gamma, y : B \vdash_{\mathsf{NJ}} \mathsf{H}^y(l) : A$ if y is fresh

And now by using the fact that typed λ-terms are in SN^β, we directly get:

Corollary 46 (Strong Normalisation of typed terms)

1. *If* $\Gamma \vdash_{LJT} M : A$ *then* $M \in \mathsf{SN}^{B,x}$.
2. *If* $\Gamma; B \vdash_{LJT} l : A$ *then* $l \in \mathsf{SN}^{B,x}$.

Again, this could also be done with any typing system such that the encodings of typed terms by H are typable in a typing system of λ-calculus that entails strong normalisation. This is again the case with intersection types: we could add the three typing rules at the bottom of Figure 5 (as well as three similar rules for lists), and the preservation of typing by the encoding would provide the strong normalisation of the system. We should expect this system to characterise $\mathsf{SN}^{B,\mathsf{x}}$ in $\overline{\lambda}$, but this remains to be checked. Also, since the typing systems of $\overline{\lambda}$ are in the spirit of sequent calculus, it would be better to replace the elimination rules of the intersection by a left-introduction of the intersection, probably in the stoup. This is ongoing work.

4 Simulation in λI

The second technique presented in this section suggests the encoding of a calculus with explicit substitutions in Church-Klop's λI-calculus [Klo80] instead of λ-calculus. We refer the reader to [Sor97, Xi97] for a survey on different techniques based on the λI-calculus to infer normalisation properties.

On the one hand, λI extends the syntax of λ-calculus with a "memory operator" so that, instead of being thrown away, a term N can be retained and carried along in a construct $[\ -\ , N]$. With this operator, those bodies of substitutions are encoded that would otherwise disappear, as explained above. On the other hand, λI restricts λ-abstractions to variables that have at least one free occurrence, so that β-reduction never erases its argument.

Doing so requires the encoding in λI to be non-deterministic, i.e. we define a relation $\mathcal{H}$ between the calculus and λI, and the reason for this is that, since the reductions in λI are non-erasing reductions, we need to add this memory operator at random places in the encoding, using such a rule:

$$\frac{M\ \mathcal{H}\ T}{M\ \mathcal{H}\ [T,U]}\ U \in \lambda I$$

For instance, $\lambda x.x\ \mathcal{H}\ \lambda x.[x,x]$ but also $\lambda x.x\ \mathcal{H}\ [\lambda x.x, \lambda z.z]$, so that both $\lambda x.[x,x]$ and $[\lambda x.x, \lambda z.z]$ (and also $\lambda x.x$) are encodings of $\lambda x.x$.

The reduction relation of the explicit substitution calculus is split into two parts Y and Z that satisfy the following simulation theorem:
$\rightarrow_Y$ is strongly simulated by $\longrightarrow_{\beta,\pi}$
$\rightarrow_Z$ is weakly simulated by $\longrightarrow_{\beta,\pi}$

Now it must be proved that every term M can be encoded into a strongly normalising term of λI. This depends on the calculus that is being treated, but the following method generally works:

1. Encode the term M as a strongly normalising λ-term t, such that no sub-term is lost, i.e. *not* using implicit substitutions. For PSN, the original λ-term would do, because it is strongly normalising by hypothesis; for a proof-term of sequent calculus, t would be a λ-term typed in an appropriate typing system, the typing tree of which is derived from the proof-tree of the sequent (we would get $t \in \mathsf{SN}^{\beta}$ using a theorem stating that typed terms are SN^{β}).

2. Using a translation i() from λ-calculus to λI, introduced in this section, prove that i(t) reduces to one of the non-deterministic encodings of M in λI, that is, that there is a term T such that $M \mathcal{H} T$ and i(t)$\longrightarrow^*_{\beta,\pi} T$.

In this section we prove that if a λ-term t is strongly normalising for β-reductions, then i(t) is weakly normalising in λI. The proof simply consists in simulating an adequate reduction sequence that starts from t and ends with a normal form, the encoding of which is a normal form of λI. What makes this simulation work is the fact that the reduction sequence is provided by a perpetual strategy. Also, weak normalisation implies strong normalisation in λI [Ned73], so that i(t) is strongly normalising, as well as the above λI-term T.

The technique is summarised in Figure 9.

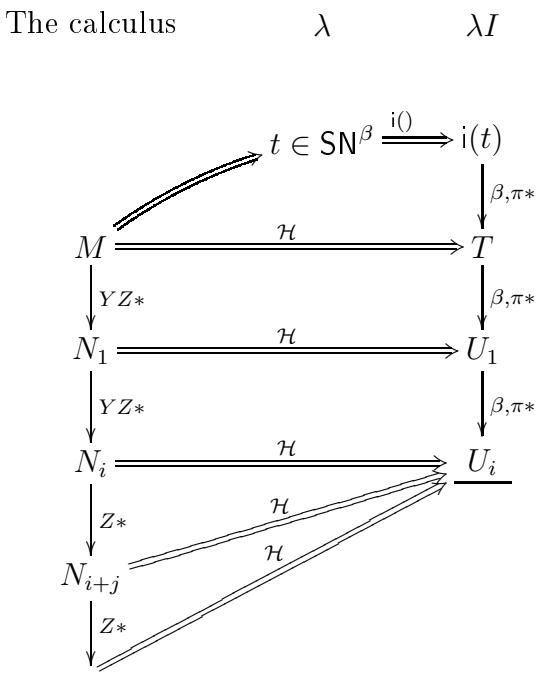


Figure 9: The general technique to prove that $M \in \mathsf{SN}$

Finally, it remains to prove that the relation Z that is only weakly simulated is now small enough to be terminating.

As we shall see, this technique works for proving PSN of the explicit substitution calculus λ_{lxr} of [KL05]. Furthermore, it can be combined with the safeness and minimality technique which provides proofs of strong normalisation for various sequent calculi that range from propositional logic to a logic as expressive as the Calculus of Constructions, and we believe that it can be applied to many other calculi.

4.1 Church-Klop's λI-calculus

Definition 16

$$T, U ::= x | \ \lambda x.T | \ T \ U | \ [T, U]$$

with the condition that $x \in FV(T)$ in $\lambda x.T$.

Lemma 47 (Stability by Substitution) *If $T, U \in \lambda I$, then $T\{x = U\} \in \lambda I$.*

Proof: By induction on T. □

The reduction rules are:

$$\begin{array}{lll} (\beta) & (\lambda x.T)\ U & \to T\{x = U\} \\ (\pi) & [T, V]\ U & \to [T\ U, V] \end{array}$$

We denote lists of λI-terms using vectors, and if $\overrightarrow{T} = T_1, \ldots, T_n$, then $U\ \overrightarrow{T}$ denotes $U\ T_1 \ldots T_n$ and $[U, \overrightarrow{T}]$ denotes $[\ldots[U, T_1], \ldots, T_n]$.

Remark 48 If $T \longrightarrow_{\beta,\pi} U$ then $FV(T) = FV(U)$ and $V\{x = T\} \longrightarrow^{+}_{\beta,\pi} V\{x = U\}$ provided that $x \in FV(V)$.

Lemma 49 (Substitution Lemma)
$T\{x = U\}\{y = V\} = T\{y = V\}\{x = U\{y = V\}\}$ *(with no variable capture)*

Proof: By induction on T. □

4.2 Simulating the perpetual strategy

We may want to use the technique of simulation in λI with calculi that annotate λ-abstractions with types, and others that do not. Indeed, one of the applications is the normalisation of systems in type theory (possibly with dependent types), so we also consider Π-types. In order to express the technique in its most general form, we present it with a mixed syntax as follows.

The *annotated?*-λ-calculus, that we call $\lambda^?$-calculus, uses the following syntax:

$$M, N, A, B ::= x |\ s |\ \Pi x^A.B |\ \lambda x^A.M |\ \lambda x.M |\ M\ N$$

where x ranges over a denumerable set of variables, and s ranges over a set of constants.

The reduction rules are

$$\begin{array}{lll} (\beta^t) & (\lambda x^A.M)\ N & \longrightarrow M\{x = N\} \\ (\beta) & (\lambda x.M)\ N & \longrightarrow M\{x = N\} \end{array}$$

Fully annotated terms are those terms that have no construct $\lambda x.M$. The fragment of fully annotated terms is stable under β^t-reductions, so that β-reductions never apply and hence $\mathsf{SN}^{\beta^t} = \mathsf{SN}^{\beta^t,\beta}$ for that fragment.

We define the notion of *type-annotation* as the smallest transitive, reflexive, context-closed relation $\lhd$ such that $\lambda x.M \lhd \lambda x^A.M$.

Notice that for a fully annotated term N, $N \lhd P$ implies $N = P$.

Lemma 50 *If $M \lhd M'$ and $M \longrightarrow_{\beta^t,\beta} N$ then there is a N' such that $N \lhd N'$ and $M \longrightarrow_{\beta^t,\beta} N$.*

Proof: By induction on M. □

Corollary 51 *If $M \lhd M'$ and $M' \in \mathsf{SN}^{\beta^t,\beta}$ then $M \in \mathsf{SN}^{\beta^t,\beta}$.*

Proof: By Theorem 14 ($\longrightarrow_{\beta^t,\beta}$ strongly simulates itself through $\lhd$). □

Definition 17 We encode the $\lambda^?$-calculus into λI as follows:

$$\begin{array}{lll}
\mathsf{i}(x) & = x & \\
\mathsf{i}(\lambda x.t) & = \lambda x.\mathsf{i}(t) & x \in FV(t) \\
\mathsf{i}(\lambda x.t) & = \lambda x.[\mathsf{i}(t), x] & x \notin FV(t) \\
\mathsf{i}(\lambda x^A.t) & = [\mathsf{i}(\lambda x.t), \mathsf{i}(A)] & \\
\mathsf{i}(t\ u) & = \mathsf{i}(t)\ \mathsf{i}(u) & \\
\mathsf{i}(s) & = \wp & \\
\mathsf{i}(\Pi x^A.B) & = \wp\ [\mathsf{i}(\lambda x.t), \mathsf{i}(A)] &
\end{array}$$

where $\wp$ is a dummy variable that does not appear in the term that is encoded.

Lemma 52 *For any $\lambda^?$-terms t and u,*

1. *$FV(i(t)) = FV(t)$*
2. *$i(t)\{x = i(u)\} = i(t\{x = u\})$*

Proof: Straightforward induction on t. □

Definition 18 The relation $\mathcal{G}$ between $\lambda^?$-terms and λI-terms is given by the following rules:

$$\frac{\forall j \qquad t_j\ \mathcal{G}\ T_j}{(x\ \overrightarrow{t_j})\ \mathcal{G}\ (x\ \overrightarrow{T_j})}\ \mathcal{G}\mathsf{var} \qquad \frac{A\ \mathcal{G}\ T \quad B\ \mathcal{G}\ U \quad x \in FV(U)}{\Pi x^A.B\ \mathcal{G}\ \wp\ [\lambda x.U, T]}\ \mathcal{G}\Pi$$

$$\frac{}{((\lambda x.t)\ t'\ \overrightarrow{t_j})\ \mathcal{G}\ \mathsf{i}((\lambda x.t)\ t'\ \overrightarrow{t_j})}\ \mathcal{G}\beta_1 \qquad \frac{t'\ \mathcal{G}\ T' \qquad x \notin FV(t)}{((\lambda x.t)\ t'\ \overrightarrow{t_j})\ \mathcal{G}\ (\mathsf{i}(\lambda x.t)\ T'\ \overrightarrow{\mathsf{i}(t_j)})}\ \mathcal{G}\beta_2$$

$$\frac{}{((\lambda x^A.t)\ t'\ \overrightarrow{t_j})\ \mathcal{G}\ \mathsf{i}((\lambda x^A.t)\ t'\ \overrightarrow{t_j})}\ \mathcal{G}\beta^t{}_1 \qquad \frac{t'\ \mathcal{G}\ T' \quad A\ \mathcal{G}\ U \quad x \notin FV(t)}{((\lambda x^A.t)\ t'\ \overrightarrow{t_j})\ \mathcal{G}\ ([\mathsf{i}(\lambda x.t), U]\ T'\ \overrightarrow{\mathsf{i}(t_j)})}\ \mathcal{G}\beta^t{}_2$$

$$\frac{}{s\ \mathcal{G}\ \wp}\ \mathcal{G}\mathsf{c} \qquad \frac{t\ \mathcal{G}\ T \qquad N \in \mathsf{nf}^{\beta,\pi}}{t\ \mathcal{G}\ [T, N]}\ \mathcal{G}\mathsf{weak}$$

$$\frac{t\ \mathcal{G}\ T \quad x \in FV(T)}{\lambda x.t\ \mathcal{G}\ \lambda x.T}\ \mathcal{G}\lambda \qquad \frac{t\ \mathcal{G}\ T \quad A\ \mathcal{G}\ U \quad x \in FV(T)}{\lambda x^A.t\ \mathcal{G}\ [\lambda x.T, U]}\ \mathcal{G}\lambda^t$$

Lemma 53

1. *If $t \in \mathsf{nf}^{\beta^t}$ and $t\ \mathcal{G}\ T$, then $T \in \mathsf{nf}^{\beta,\pi}$.*
2. *For any $\lambda^?$-term t, $t\ \mathcal{G}\ i(t)$.*

Proof:

1. By induction on the proof tree associated to $t\ \mathcal{G}\ T$, one can check that no β and no π-redex is introduced, since rules $\mathcal{G}\beta_1$, $\mathcal{G}\beta_2$, $\mathcal{G}\beta^t{}_1$ and $\mathcal{G}\beta^t{}_2$ are forbidden by the hypothesis that t is a β-normal form.

2. By induction on t:

   - If $t = x\ \overrightarrow{t_j}$, then by induction hypothesis $t_j\ \mathcal{G}\ \mathsf{i}(t_j)$ for all j and then we can apply $\mathcal{G}\mathsf{var}$.
   - If $t = (\lambda x.t')\ u\ \overrightarrow{t_j}$, then it suffices to use rules $\mathcal{G}\beta_1$.
   - If $t = (\lambda x^A.t')\ u\ \overrightarrow{t_j}$, then it suffices to use rules $\mathcal{G}\beta^t{}_1$.
   - If $t = \lambda x.u$ then by induction hypothesis $u\ \mathcal{G}\ \mathsf{i}(u)$. If $x \in FV(u)$, then $\mathsf{i}(t) = \lambda x.\mathsf{i}(u)$ and $t\ \mathcal{G}\ \mathsf{i}(t)$ by rule $\mathcal{G}\lambda$. If $x \notin FV(u)$, then $\mathsf{i}(t) = \lambda x.[\mathsf{i}(u), x]$, and thus $u\ \mathcal{G}\ [\mathsf{i}(u), x]$ by rule $\mathcal{G}\mathsf{weak}$ and $t\ \mathcal{G}\ \mathsf{i}(t)$ by rule $\mathcal{G}\lambda$.
   - If $t = \lambda x^A.u$ then by induction hypothesis $u\ \mathcal{G}\ \mathsf{i}(u)$ and $A\ \mathcal{G}\ \mathsf{i}(A)$. If $x \in FV(u)$, then $\mathsf{i}(t) = [\lambda x.\mathsf{i}(u), \mathsf{i}(A)]$ and $t\ \mathcal{G}\ \mathsf{i}(t)$ by rule $\mathcal{G}\lambda^t$. If $x \notin FV(u)$, then $\mathsf{i}(t) = [\lambda x.[\mathsf{i}(u), x], \mathsf{i}(A)]$, and thus $u\ \mathcal{G}\ [\mathsf{i}(u), x]$ by rule $\mathcal{G}\mathsf{weak}$ and $t\ \mathcal{G}\ \mathsf{i}(t)$ by rule $\mathcal{G}\lambda^t$.
   - If $t = s$, then clearly $s\ \mathcal{G}\ \wp$.
   - If $t = \Pi x^A.B$, then by induction hypothesis $A\ \mathcal{G}\ \mathsf{i}(A)$ and $B\ \mathcal{G}\ \mathsf{i}(B)$. If $x \in FV(B)$ then $\mathsf{i}(\Pi x^A.B) = \wp\ [\lambda x.\mathsf{i}(B), \mathsf{i}(A)]$ and $t\ \mathcal{G}\ \mathsf{i}(t)$ by rule $\mathcal{G}\Pi$. If $x \in FV(B)$ then $\mathsf{i}(\Pi x^A.B) = \wp\ [\lambda x.[\mathsf{i}(B), x], \mathsf{i}(A)]$, and thus $B\ \mathcal{G}\ [\mathsf{i}(B), x]$ by rule $\mathcal{G}\mathsf{weak}$ and $t\ \mathcal{G}\ \mathsf{i}(t)$ by rule $\mathcal{G}\Pi$.

□

Definition 19 We define a reduction relation $\rightsquigarrow$ for $\lambda^?$-terms by the following rules:

$$\frac{t \rightsquigarrow t'}{x\ \overrightarrow{t_j}\ t\ \overrightarrow{p_j} \rightsquigarrow x\ \overrightarrow{t_j}\ t'\ \overrightarrow{p_j}}\ \mathsf{perp\text{-}var} \qquad \frac{t \rightsquigarrow t'}{\lambda x.t \rightsquigarrow \lambda x.t'}\ \mathsf{perp}\lambda$$

$$\frac{t \rightsquigarrow t'}{\lambda x^A.t \rightsquigarrow \lambda x^A.t'}\ \mathsf{perp}\lambda^t_1 \qquad \frac{A \rightsquigarrow A'}{\lambda x^A.t \rightsquigarrow \lambda x^{A'}.t}\ \mathsf{perp}\lambda^t_2$$

$$\frac{x \in FV(t) \vee t' \in \mathsf{nf}^{\beta^t\beta}}{(\lambda x.t)\ t'\ \overrightarrow{t_j} \rightsquigarrow t\{x = t'\}\ \overrightarrow{t_j}}\ \mathsf{perp}\beta_1$$

$$\frac{t' \rightsquigarrow t'' \qquad x \notin FV(t)}{(\lambda x.t)\ t'\ \overrightarrow{t_j} \rightsquigarrow (\lambda x.t)\ t''\ \overrightarrow{t_j}}\ \mathsf{perp}\beta_2$$

$$\frac{x \in FV(t) \vee t', A \in \mathsf{nf}^{\beta^t\beta}}{(\lambda x^A.t)\ t'\ \overrightarrow{t_j} \rightsquigarrow t\{x = t'\}\ \overrightarrow{t_j}}\ \mathsf{perp}\beta^t{}_1$$

$$\frac{t' \rightsquigarrow t'' \qquad x \notin FV(t)}{(\lambda x^A.t)\ t'\ \overrightarrow{t_j} \rightsquigarrow (\lambda x^A.t)\ t''\ \overrightarrow{t_j}}\ \mathsf{perp}\beta^t{}_2$$

$$\frac{A \rightsquigarrow A' \qquad x \notin FV(t)}{(\lambda x^A.t)\ t'\ \overrightarrow{t_j} \rightsquigarrow (\lambda x^{A'}.t)\ t'\ \overrightarrow{t_j}}\ \mathsf{perp}\beta^t{}_3$$

$$\frac{A \rightsquigarrow A'}{\Pi x^A.B \rightsquigarrow \Pi x^{A'}.B}\ \mathsf{perp}\Pi_1 \qquad \frac{B \rightsquigarrow B'}{\Pi x^A.B \rightsquigarrow \Pi x^A.B'}\ \mathsf{perp}\Pi_2$$

Remark 54 $\rightsquigarrow \subseteq \longrightarrow_{\beta^t\beta}$
If t is not a $\beta^t\beta$-normal form, then there is a $\lambda^?$-term t' such that $t \rightsquigarrow t'$.

Remark 55 Although we do not need it in the rest of the proof, it is worth mentioning that, at least in the fragment of the untyped λ-calculus, the relation $\rightsquigarrow$ defines a perpetual strategy w.r.t β-reduction, i.e. if M is not β-strongly normalising and $M \rightsquigarrow M'$, then neither is M' [vRSSX99].

Theorem 56 $\longrightarrow_{\beta,\pi}$ *strongly simulates* $\rightsquigarrow$ *through* $\mathcal{G}$.

Proof:

perpβ_1) $(\lambda x.t)\ t'\ \overrightarrow{t_j} \rightsquigarrow t\{x = t'\}\ \overrightarrow{t_j}$

- $x \in FV(t)$:
  The last rule used to prove $u\ \mathcal{G}\ U$ must be $\mathcal{G}\beta_1$ (possibly followed by several steps of $\mathcal{G}$weak), so

$$\begin{array}{lll} U & = & [\lambda x.\mathsf{i}(t)\ \mathsf{i}(t')\ \overrightarrow{\mathsf{i}(t_j)}, \overrightarrow{N}] \\ & \longrightarrow_\beta & [\mathsf{i}(t)\{x = \mathsf{i}(t')\}\ \overrightarrow{\mathsf{i}(t_j)}, \overrightarrow{N}] \\ & =_{Lemma\ 52\ (2)} & [\mathsf{i}(t\{x = t'\}\ \overrightarrow{t_j}), \overrightarrow{N}] \end{array}$$

  Then by Lemma 53 (2), $t\{x = t'\}\ \overrightarrow{t_j}\ \mathcal{G}\ \mathsf{i}(t\{x = t'\}\ \overrightarrow{t_j})$ and by rule $\mathcal{G}$weak, $t\{x = t'\}\ \overrightarrow{t_j}\ \mathcal{G}\ [\mathsf{i}(t\{x = t'\}\ \overrightarrow{t_j}), \overrightarrow{N}]$.
- $x \notin FV(t)$:
  It means that t' is a β-normal form and $t\{x = t'\}\ \overrightarrow{t_j} = t\ \overrightarrow{t_j}$. The last rule used to prove $u\ \mathcal{G}\ U$ must be $\mathcal{G}\beta_1$ or $\mathcal{G}\beta_2$ (possibly followed by several steps of $\mathcal{G}$weak), so in both cases we have $U = [\lambda x.[\mathsf{i}(t), x]\ T'\ \overrightarrow{\mathsf{i}(t_j)}, \overrightarrow{N}]$ with $t'\ \mathcal{G}\ T'$ (using Lemma 53 (2) in the former case where $T' = \mathsf{i}(t')$). By Lemma 53 (1), T' is a β,π-normal form. Now $U \longrightarrow_\beta [[\mathsf{i}(t)\{x = T'\}, T']\ \overrightarrow{\mathsf{i}(t_j)}, \overrightarrow{N}]$. But by Lemma 52 (1), $x \notin FV(\mathsf{i}(t))$ so the above term is $[[\mathsf{i}(t), T']\ \overrightarrow{\mathsf{i}(t_j)}, \overrightarrow{N}]$, which reduces by π to $[\mathsf{i}(t)\ \overrightarrow{\mathsf{i}(t_j)}, T', \overrightarrow{N}] = [\mathsf{i}(t\ \overrightarrow{t_j}), T', \overrightarrow{N}]$. By Lemma 53 (2) and rule $\mathcal{G}$weak, we get $t\ \overrightarrow{t_j}\ \mathcal{G}\ [\mathsf{i}(t\ \overrightarrow{t_j}), T', \overrightarrow{N}]$.

perpβ_2) $(\lambda x.t)\ t'\ \overrightarrow{t_j} \rightsquigarrow (\lambda x.t)\ t''\ \overrightarrow{t_j}$ with $t' \rightsquigarrow t''$ and $x \notin FV(t)$.

The last rule used to prove $u\ \mathcal{G}\ U$ must be $\mathcal{G}\beta_1$ or $\mathcal{G}\beta_2$ (possibly followed by several steps of $\mathcal{G}$weak), so in both cases $U = [\lambda x.[\mathsf{i}(t), x]\ T'\ \overrightarrow{\mathsf{i}(t_j)}, \overrightarrow{N}]$ with $t'\ \mathcal{G}\ T'$ (using Lemma 53 (2) in the former case where $T' = \mathsf{i}(t')$). By induction hypothesis, there is a term T'' such that $T' \longrightarrow^+_{\beta,\pi} T''$ and $t''\ \mathcal{G}\ T''$.
Hence, $U \longrightarrow^+_{\beta,\pi} [\lambda x.[\mathsf{i}(t), x]\ T''\ \overrightarrow{\mathsf{i}(t_j)}, \overrightarrow{N}]$. By application of the rule $\mathcal{G}\beta_2$, $(\lambda x.t)\ t''\ \overrightarrow{t_j}\ \mathcal{G}\ \lambda x.[\mathsf{i}(t), x]\ T''\ \overrightarrow{\mathsf{i}(t_j)}$, and we use rule $\mathcal{G}$weak to conclude.

perp$\beta^t{}_1$) $(\lambda x^A.t)\ t'\ \overrightarrow{t_j} \rightsquigarrow t\{x = t'\}\ \overrightarrow{t_j}$

– $x \in FV(t)$:
The last rule used to prove $u\ \mathcal{G}\ U$ must be $\mathcal{G}\beta^t{}_1$ (possibly followed by several steps of $\mathcal{G}$weak), so

$$\begin{array}{lll} U & = & [[\lambda x.\mathsf{i}(t), \mathsf{i}(A)]\ \mathsf{i}(t')\ \overrightarrow{\mathsf{i}(t_j)}, \overrightarrow{N}] \\ & \longrightarrow^+{}_\pi & [\lambda x.\mathsf{i}(t)\ \mathsf{i}(t')\ \overrightarrow{\mathsf{i}(t_j)}, \mathsf{i}(A), \overrightarrow{N}] \\ & \longrightarrow_\beta & [\mathsf{i}(t)\{x = \mathsf{i}(t')\}\ \overrightarrow{\mathsf{i}(t_j)}, \mathsf{i}(A), \overrightarrow{N}] \\ & =_{Lemma\ 52\ (2)} & [\mathsf{i}(t\{x = t'\}\ \overrightarrow{t_j}), \mathsf{i}(A), \overrightarrow{N}] \end{array}$$

Then by Lemma 53 (2), $t\{x = t'\}\ \overrightarrow{t_j}\ \mathcal{G}\ \mathsf{i}(t\{x = t'\}\ \overrightarrow{t_j})$ and by rule $\mathcal{G}$weak, $t\{x = t'\}\ \overrightarrow{t_j}\ \mathcal{G}\ [\mathsf{i}(t\{x = t'\}\ \overrightarrow{t_j}), \mathsf{i}(A), \overrightarrow{N}]$.

– $x \notin FV(t)$:
It means that t' and A are β-normal forms and $t\{x = t'\}\ \overrightarrow{t_j} = t\ \overrightarrow{t_j}$. The last rule used to prove $u\ \mathcal{G}\ U$ must be $\mathcal{G}\beta^t{}_1$ or $\mathcal{G}\beta^t{}_2$ (possibly followed by several steps of $\mathcal{G}$weak), so in both cases we have $U = [[\lambda x.[\mathsf{i}(t), x], U']\ T'\ \overrightarrow{\mathsf{i}(t_j)}, \overrightarrow{N}]$ with $A\ \mathcal{G}\ U'$ and $t'\ \mathcal{G}\ T'$ (using Lemma 53 (2) in the former case where $U' = \mathsf{i}(A)$ and $T' = \mathsf{i}(t')$). By Lemma 53 (1), U' and T' are β,π-normal forms. Now $U \longrightarrow_\pi\ [\lambda x.[\mathsf{i}(t), x]\ T'\ \overrightarrow{\mathsf{i}(t_j)}, U', \overrightarrow{N}] \longrightarrow_\beta\ [[\mathsf{i}(t)\{x = T'\}, T']\ \overrightarrow{\mathsf{i}(t_j)}, U', \overrightarrow{N}]$. But by Lemma 52 (1), $x \notin FV(\mathsf{i}(t))$ so the above term is $[[\mathsf{i}(t), T']\ \overrightarrow{\mathsf{i}(t_j)}, U', \overrightarrow{N}]$, which reduces by π to $[\mathsf{i}(t)\ \overrightarrow{\mathsf{i}(t_j)}, T', U', \overrightarrow{N}] = [\mathsf{i}(t\ \overrightarrow{t_j}), T', U', \overrightarrow{N}]$. By Lemma 53 (2) and rule $\mathcal{G}$weak, we get $t\ \overrightarrow{t_j}\ \mathcal{G}\ [\mathsf{i}(t\ \overrightarrow{t_j}), T', U', \overrightarrow{N}]$.

perp$\beta^t{}_2$) $(\lambda x^A.t)\ t'\ \overrightarrow{t_j} \rightsquigarrow (\lambda x^A.t)\ t''\ \overrightarrow{t_j}$ with $t' \rightsquigarrow t''$ and $x \notin FV(t)$.
The last rule used to prove $u\ \mathcal{G}\ U$ must be $\mathcal{G}\beta^t{}_1$ or $\mathcal{G}\beta^t{}_2$ (possibly followed by several steps of $\mathcal{G}$weak), so in both cases $U = [[\lambda x.[\mathsf{i}(t), x], U']\ T'\ \overrightarrow{\mathsf{i}(t_j)}, \overrightarrow{N}]$ with $A\ \mathcal{G}\ U'$ and $t'\ \mathcal{G}\ T'$ (using Lemma 53 (2) in the former case where $U' = \mathsf{i}(A)$ and $T' = \mathsf{i}(t')$). By induction hypothesis, there is a term T'' such that $T' \longrightarrow^+{}_{\beta,\pi} T''$ and $t''\ \mathcal{G}\ T''$. Hence, $U \longrightarrow^+{}_{\beta,\pi}\ [[\lambda x.[\mathsf{i}(t), x], U']\ T''\ \overrightarrow{\mathsf{i}(t_j)}, \overrightarrow{N}]$. By application of the rule $\mathcal{G}\beta^t{}_2$, $(\lambda x^A.t)\ t''\ \overrightarrow{t_j}\ \mathcal{G}\ [\lambda x.[\mathsf{i}(t), x], U']\ T''\ \overrightarrow{\mathsf{i}(t_j)}$, and we use rule $\mathcal{G}$weak to conclude.

perp$\beta^t{}_3$) $(\lambda x^A.t)\ t'\ \overrightarrow{t_j} \rightsquigarrow (\lambda x^{A'}.t)\ t'\ \overrightarrow{t_j}$ with $A \rightsquigarrow A'$ and $x \notin FV(t)$.
The last rule used to prove $u\ \mathcal{G}\ U$ must be $\mathcal{G}\beta^t{}_1$ or $\mathcal{G}\beta^t{}_2$ (possibly followed by several steps of $\mathcal{G}$weak), so in both cases $U = [[\lambda x.[\mathsf{i}(t), x], U']\ T'\ \overrightarrow{\mathsf{i}(t_j)}, \overrightarrow{N}]$ with $A\ \mathcal{G}\ U'$ and $t'\ \mathcal{G}\ T'$ (using Lemma 53 (2) in the former case where $U' = \mathsf{i}(A)$ and $T' = \mathsf{i}(t')$). By induction hypothesis, there is a term U'' such that $U' \longrightarrow^+{}_{\beta,\pi} U''$ and $A'\ \mathcal{G}\ U''$. Hence, $U \longrightarrow^+{}_{\beta,\pi}\ [[\lambda x.[\mathsf{i}(t), x], U'']\ T'\ \overrightarrow{\mathsf{i}(t_j)}, \overrightarrow{N}]$. By application of the rule $\mathcal{G}\beta^t{}_2$, $(\lambda x^{A'}.t)\ t'\ \overrightarrow{t_j}\ \mathcal{G}\ [\lambda x.[\mathsf{i}(t), x], U'']\ T'\ \overrightarrow{\mathsf{i}(t_j)}$, and we use rule $\mathcal{G}$weak to conclude.

perpλ) $\lambda x.t \rightsquigarrow \lambda x.t'$ with $t \rightsquigarrow t'$.
The last rule used to prove $u\ \mathcal{G}\ U$ must be $\mathcal{G}\lambda$, so $U = [\lambda x.T, \overrightarrow{N}]$ with $t\ \mathcal{G}\ T$. By induction hypothesis, there is a term T' such that $T \longrightarrow^+{}_{\beta,\pi} T'$ and $t'\ \mathcal{G}\ T'$. Hence, $U \longrightarrow^+{}_{\beta,\pi} [\lambda x.T', \overrightarrow{N}]$ (with $x \in FV(T')$), and we obtain by application of rules $\mathcal{G}\lambda$ and $\mathcal{G}$weak that $\lambda x.t'\ \mathcal{G}\ [\lambda x.T', \overrightarrow{N}]$.

perpλ_1^t) $\lambda x^A.t \rightsquigarrow \lambda x^A.t'$ with $t \rightsquigarrow t'$.

The last rule used to prove $u \mathcal{G} U$ must be $\mathcal{G}\lambda^t$, so $U = [\lambda x.T, U', \overrightarrow{N}]$ with $A \mathcal{G} U'$ and $t \mathcal{G} T$. By induction hypothesis, there is a term T' such that $T \longrightarrow^+_{\beta,\pi} T'$ and $t' \mathcal{G} T'$. Hence, $U \longrightarrow^+_{\beta,\pi} [\lambda x.T', U', \overrightarrow{N}]$ (with $x \in FV(T')$), and we obtain by application of rules $\mathcal{G}\lambda^t$ and $\mathcal{G}\mathsf{weak}$ that $\lambda x^A.t' \mathcal{G} [\lambda x.T', U', \overrightarrow{N}]$.

perpλ_2^t) $\lambda x^A.t \rightsquigarrow \lambda x^{A'}.t$ with $A \rightsquigarrow A'$.

The last rule used to prove $u \mathcal{G} U$ must be $\mathcal{G}\lambda^t$, so $U = [\lambda x.T, U', \overrightarrow{N}]$ with $A \mathcal{G} U'$ and $t \mathcal{G} T$. By induction hypothesis, there is a term U'' such that $U' \longrightarrow^+_{\beta,\pi} U''$ and $A' \mathcal{G} U''$. Hence, $U \longrightarrow^+_{\beta,\pi} [\lambda x.T, U'', \overrightarrow{N}]$ (with $x \in FV(T')$), and we obtain by application of rules $\mathcal{G}\lambda^t$ and $\mathcal{G}\mathsf{weak}$ that $\lambda x^A.t' \mathcal{G} [\lambda x.T, U'', \overrightarrow{N}]$.

perp-var) $x\ \overrightarrow{t_j}\ t\ \overrightarrow{p_j} \rightsquigarrow x\ \overrightarrow{t_j}\ t'\ \overrightarrow{p_j}$ with $t \rightsquigarrow t'$.

The last rule used to prove $u \mathcal{G} U$ must be $\mathcal{G}\mathsf{var}$, so $U = [x\ \overrightarrow{Q_j}\ T\ \overrightarrow{U_j}, \overrightarrow{N}]$ with $t \mathcal{G} T$, $t_j \mathcal{G} Q_j$ and $p_j \mathcal{G} U_j$. By induction hypothesis, there is a term T' such that $T \longrightarrow^+_{\beta,\pi} T'$ and $t' \mathcal{G} T'$. As a consequence we get $U \longrightarrow^+_{\beta,\pi} [x\ \overrightarrow{Q_j}\ T'\ \overrightarrow{U_j}, \overrightarrow{N}]$ and by rules $\mathcal{G}\mathsf{var}$ and $\mathcal{G}\mathsf{weak}$ we obtain $x\ \overrightarrow{t_j}\ t'\ \overrightarrow{p_j} \mathcal{G} [x\ \overrightarrow{Q_j}\ T'\ \overrightarrow{U_j}, \overrightarrow{N}]$.

perpΠ_1) $\Pi x^A.B \rightsquigarrow \Pi x^{A'}.B$ with $A \rightsquigarrow A'$.

The last rule used to prove $u \mathcal{G} U$ must be $\mathcal{G}\Pi$, so $U = [\wp\,[\lambda x.T, V], \overrightarrow{N}]$ with $B \mathcal{G} T$ and $A \mathcal{G} V$. By induction hypothesis, there is a term V' such that $V \longrightarrow^+_{\beta,\pi} V'$ and $A' \mathcal{G} V'$. As a consequence we get $U \longrightarrow^+_{\beta,\pi} [\wp\,[\lambda x.T, V'], \overrightarrow{N}]$ and by application of rules $\mathcal{G}\Pi$ and $\mathcal{G}\mathsf{weak}$ we obtain $\Pi x^{A'}.B \mathcal{G} [\wp\,[\lambda x.T, V'], \overrightarrow{N}]$.

perpΠ_2) $\Pi x^A.B \rightsquigarrow \Pi x^A.B'$ with $B \rightsquigarrow B'$.

The last rule used to prove $u \mathcal{G} U$ must be $\mathcal{G}\Pi$, so $U = [\wp\,[\lambda x.T, V], \overrightarrow{N}]$ with $B \mathcal{G} T$ and $A \mathcal{G} V$. By induction hypothesis, there is a term T' such that $T \longrightarrow^+_{\beta,\pi} T'$ and $B' \mathcal{G} T'$. As a consequence we get $U \longrightarrow^+_{\beta,\pi} [\wp\,[\lambda x.T', V], \overrightarrow{N}]$ and by application of rules $\mathcal{G}\Pi$ and $\mathcal{G}\mathsf{weak}$ we obtain $\Pi x^A.B' \mathcal{G} [\wp\,[\lambda x.T', V], \overrightarrow{N}]$.

□

Corollary 57 *If $t \in \mathcal{WN}^{\rightsquigarrow}$ and $t \mathcal{G} T$ then $T \in \mathcal{WN}^{\beta,\pi}$.*

Proof: By induction in $\mathsf{WN}^{\rightsquigarrow}$, the induction hypothesis is:
$t \in \mathsf{nf}^{\rightsquigarrow} \vee (\exists u \in \rightsquigarrow(t), \forall U, u \mathcal{G} U \Rightarrow U \in \mathsf{WN}^{\beta,\pi})$.

If $t \in \mathsf{nf}^{\rightsquigarrow}$, then Lemma 53 (1) gives $T \in \mathsf{nf}^{\beta,\pi} \subseteq \mathsf{WN}^{\beta,\pi}$.

If $\exists u \in \rightsquigarrow(t), \forall U, u \mathcal{G} U \Rightarrow U \in \mathsf{WN}^{\beta,\pi}$, then by Theorem 56 we get a specific T' such that $u \mathcal{G} T'$ and $T \longrightarrow^+_{\beta,\pi} T'$. We can apply the induction hypothesis by taking $U = T'$ and get $T' \in \mathsf{WN}^{\beta,\pi}$. But because $\mathsf{WN}^{\beta,\pi}$ is patriarchal, $T \in \mathsf{WN}^{\beta,\pi}$ as required. □

Corollary 58 *$i(\mathcal{SN}^{\beta t \beta}) \subseteq \mathcal{WN}^{\beta,\pi}$*

Proof: Notice that $\mathsf{SN}^{\beta t \beta} \subseteq \mathsf{SN}^{\rightsquigarrow} \subseteq \mathsf{WN}^{\rightsquigarrow}$. Then Lemma 53 (2) gives $\forall t \in \mathsf{SN}^{\beta t \beta}, t \mathcal{G} \mathsf{i}(t)$, and thus, by Theorem 56, $\mathsf{i}(t) \in \mathsf{WN}^{\beta,\pi}$. □

Theorem 59 (Nederpelt [Ned73]) $WN^{\beta,\pi} \subseteq SN^{\beta,\pi}$

Corollary 60 *For any $\lambda^?$-term t, if $t \in SN^{\beta t \beta}$, then $i(t) \in SN^{\beta,\pi}$.*

Proof: By Corollary 58 and Theorem 59. □

4.3 Example: λlxr

Inspired by proof-nets and linear logic [Gir87], λlxr is an explicit substitution calculus introduced in [KL05] as the first such calculus having the PSN property and full composition of substitutions. It differs from λx or $\overline{\lambda}$ by the use of explicit resource operators: duplication and erasure, which respectively correspond to contraction and weakening in a typed framework. Binding a variable that has no occurrence or more than one is explicitly expressed by the use of these operators. By the use of erasure operators, the set of free variables is preserved by reduction, which corresponds to the notion of *interface preserving* of Interaction Nets [Laf90]. The rewrite system of λlxr simulates β-reduction, but the techniques used to prove PSN for λx and λlxr all fail, so we use the technique of simulation in λI.

For a full presentation of λlxr, we refer the reader to [KL05]. We only briefly recall here the syntax and the reduction relation.

The syntax of λlxr is given by the following grammar:

$$t ::= x \mid \lambda x.t \mid t\ t \mid t\langle x = t\rangle \mid W_x(t) \mid C_x^{y,z}(t)$$

The abstraction $\lambda x.t$ and the substitution $t\langle x = u\rangle$ bind x in t. The contraction $C_x^{y,z}(t)$ binds y and z in t, whereas x is *free* in the terms x, $C_x^{y,z}(t)$ and $W_x(t)$.

We say that a term is *linear* if it satisfies the following: in every sub-term, every variable has at most one free occurrence, and every binder binds a variable that does have a free occurrence (and hence only one).

For instance, the terms $W_x(x)$ and $\lambda x.xx$ are not linear. However, the latter can be represented in the λlxr-calculus by the linear term $\lambda x.C_x^{y,z}(yz)$. More generally, every λ-term can be translated to a linear λlxr-term.

We use $\Phi, \Delta, \Sigma, \Pi, \ldots$ to denote finite *lists* of variables (with no repetition). We use the notation $W_{x_1,\ldots,x_n}(t)$ for $W_{x_1}(\ldots W_{x_n}(t))$, and $C_{x_1,\ldots,x_n}^{(y_1,\ldots,y_n),(z_1,\ldots,z_n)}(t)$ for $C_{x_1}^{y_1,z_1}(\ldots C_{x_n}^{y_n,z_n}(t))$.

For any term t we define a *renaming operation* $R_{y_1,\ldots,y_n}^{x_1,\ldots,x_n}(t)$ as the result of *simultaneously* substituting y_i for every *free* occurrence x_i in t where $i \in 1\ldots n$. Thus for instance $R_{x',y'}^{x,y}(C_w^{y,z}(x(yz))) = C_w^{y,z}(x'(yz))$.

We introduce in Figure 10 a congruence $\equiv$, which enables us to write "$W_{\mathcal{S}}(u)$", or "$C_{\Phi}^{\Delta,\Pi}(t)$ where $\Phi := \mathcal{S}$", without ordering the variables in $\mathcal{S}$. Besides, we sometimes do not specify what the lists Δ and Π are, assuming them to be two *disjoint* lists of *fresh* variables.

The reduction relation of the calculus, denoted $\longrightarrow_{\lambda\mathsf{lxr}}$, is the relation generated by the reduction rules in Figure 11 modulo the congruence relation in Figure 10. The rules should be understood in the prospect of applying them to linear terms. Indeed, it can be shown that if t is linear and $t \longrightarrow_{\lambda\mathsf{lxr}} t'$, then t' is linear and $FV(t) = FV(t')$. The fact that linearity is preserved is a essential requirement of the system, so that we can henceforth consider linear terms only.

A basic property of the reduction relation is the following:

$C_w^{x,v}(C_x^{z,y}(t))$	$\equiv$	$C_w^{x,y}(C_x^{z,v}(t))$	if $x \neq y, v$
$C_x^{y,z}(t)$	$\equiv$	$C_x^{z,y}(t)$	
$C_{x'}^{y',z'}(C_x^{y,z}(t))$	$\equiv$	$C_x^{y,z}(C_{x'}^{y',z'}(t))$	if $x \neq y', z'$ & $x' \neq y, z$
$W_x(W_y(t))$	$\equiv$	$W_y(W_x(t))$	
$t\langle x = u\rangle\langle y = v\rangle$	$\equiv$	$t\langle y = v\rangle\langle x = u\rangle$	if $y \notin FV(u)$ & $x \notin FV(v)$ & $x \neq y$
$C_w^{y,z}(t)\langle x = u\rangle$	$\equiv$	$C_w^{y,z}(t\langle x = u\rangle)$	if $x \neq w$ & $y, z \notin FV(u)$

Figure 10: Congruence axioms for λlxr-terms

(B)	$(\lambda x.t)\ u$	$\longrightarrow$	$t\langle x = u\rangle$	
System x				
(Abs)	$(\lambda y.t)\langle x = u\rangle$	$\longrightarrow$	$\lambda y.t\langle x = u\rangle$	
$(App1)$	$(t\ v)\langle x = u\rangle$	$\longrightarrow$	$t\langle x = u\rangle\ v$	$x \in FV(t)$
$(App2)$	$(t\ v)\langle x = u\rangle$	$\longrightarrow$	$t\ v\langle x = u\rangle$	$x \in FV(v)$
(Var)	$x\langle x = u\rangle$	$\longrightarrow$	u	
$(Weak1)$	$W_x(t)\langle x = u\rangle$	$\longrightarrow$	$W_{FV(u)}(t)$	
$(Weak2)$	$W_y(t)\langle x = u\rangle$	$\longrightarrow$	$W_y(t\langle x = u\rangle)$	$x \neq y$
$(Cont1)$	$C_x^{y,z}(t)\langle x = u\rangle$	$\longrightarrow$	$C_\Phi^{\Delta,\Pi}(t\langle y = u_1\rangle\langle z = u_2\rangle)$	
			where $\Phi := FV(u)$	
			$u_1 = R_\Delta^\Phi(u)$	
			$u_2 = R_\Pi^\Phi(u)$	
$(Comp)$	$t\langle y = v\rangle\langle x = u\rangle$	$\longrightarrow$	$t\langle y = v\langle x = u\rangle\rangle$	$x \in FV(v)$
System r				
$(WAbs)$	$\lambda x.W_y(t)$	$\longrightarrow$	$W_y(\lambda x.t)$	$x \neq y$
$(WApp1)$	$W_y(u)\ v$	$\longrightarrow$	$W_y(uv)$	
$(WApp2)$	$u\ W_y(v)$	$\longrightarrow$	$W_y(uv)$	
$(WSubs)$	$t\langle x = W_y(u)\rangle$	$\longrightarrow$	$W_y(t\langle x = u\rangle)$	
$(Merge)$	$C_w^{y,z}(W_y(t))$	$\longrightarrow$	$R_w^z(t)$	
$(Cross)$	$C_w^{y,z}(W_x(t))$	$\longrightarrow$	$W_x(C_w^{y,z}(t))$	$x \neq y,\ \ x \neq z$
$(CAbs)$	$C_w^{y,z}(\lambda x.t)$	$\longrightarrow$	$\lambda x.C_w^{y,z}(t)$	
$(CApp1)$	$C_w^{y,z}(t\ u)$	$\longrightarrow$	$C_w^{y,z}(t)\ u$	$y, z \in FV(t)$
$(CApp2)$	$C_w^{y,z}(t\ u)$	$\longrightarrow$	$t\ C_w^{y,z}(u)$	$y, z \in FV(u)$
$(CSubs)$	$C_w^{y,z}(t\langle x = u\rangle)$	$\longrightarrow$	$t\langle x = C_w^{y,z}(u)\rangle$	$y, z \in FV(u)$

Figure 11: Reduction rules for λlxr-terms

Theorem 61 (Lengrand [KL05]) *xr is terminating.*

Now we can encode λ-calculus in λlxr.

Definition 20 The encoding of λ-terms is defined by induction as follows:

$$\begin{array}{lcll} \mathsf{A}(x) & := & x & \\ \mathsf{A}(\lambda x.t) & := & \lambda x.\mathsf{A}(t) & \text{if } x \in FV(t) \\ \mathsf{A}(\lambda x.t) & := & \lambda x.W_x(\mathsf{A}(t)) & \text{if } x \notin FV(t) \\ \mathsf{A}(tu) & := & C_\Phi^{\Delta,\Pi}(R_\Delta^\Phi(\mathsf{A}(t))\ R_\Pi^\Phi(\mathsf{A}(u))) & \text{where } \Phi := FV(t) \cap FV(u) \end{array}$$

In [KL05], the following property has been proved:

Theorem 62 (Simulating β-reduction)
If $t \longrightarrow_\beta t'$, then $\mathsf{A}(t) \longrightarrow^+_{\lambda \mathsf{lxr}} W_{FV(t)\setminus FV(t')}(\mathsf{A}(t'))$.

Now we prove the PSN property in detail.

Definition 21 The relation $\mathcal{H}$ between well-formed $\lambda\mathsf{lxr}$-terms and λI is given by the following rules:

$$\frac{}{x\ \mathcal{H}\ x} \qquad \frac{t\ \mathcal{H}\ T}{\lambda x.t\ \mathcal{H}\ \lambda x.T} \qquad \frac{t\ \mathcal{H}\ T \quad u\ \mathcal{H}\ U}{tu\ \mathcal{H}\ TU} \qquad \frac{t\ \mathcal{H}\ T}{t\ \mathcal{H}\ [M,N]}\ N \in \lambda I$$

$$\frac{t\ \mathcal{H}\ T \quad u\ \mathcal{H}\ U}{t\langle x=u\rangle\ \mathcal{H}\ T\{x=U\}} \qquad \frac{t\ \mathcal{H}\ T}{C_x^{y,z}(t)\ \mathcal{H}\ T\{y=x\}\{z=x\}} \qquad \frac{t\ \mathcal{H}\ T}{W_x(t)\ \mathcal{H}\ T}\ x \in FV(T)$$

The relation $\mathcal{H}$ enjoys the following properties.

Lemma 63 *If $t\ \mathcal{H}\ M$, then*

1. *$FV(t) \subseteq FV(M)$*
2. *$M \in \lambda I$*
3. *$x \notin FV(t)$ and $N \in \lambda I$ implies $t\ \mathcal{H}\ M\{x=N\}$*
4. *$t \equiv t'$ implies $t'\ \mathcal{H}\ M$*
5. *$R_\Delta^\Gamma(t)\ \mathcal{H}\ R_\Delta^\Gamma(M)$*

Proof: Property (1) is a straightforward induction on the proof tree as well as Property (2) which also uses Lemma 47. Properties (3) and (5) are also proved by induction on the tree, using the substitution lemma that holds in λI. For Property (4):

- If $t\langle x=u\rangle\langle y=v\rangle\ \mathcal{H}\ M$ with $y \notin FV(u)$, then $M = [[T\{x=U\}, \overrightarrow{T}]\{y=V\}, \overrightarrow{U}]$ with $t\ \mathcal{H}\ T$, $u\ \mathcal{H}\ U$ and $v\ \mathcal{H}\ V$. We can assume

  $$x \notin FV(T_1) \cup \ldots \cup FV(T_m) \cup FV(V)$$

  so that $M = [[T, \overrightarrow{T}]\{x=U\}\{y=V\}, \overrightarrow{U}] = [[T, \overrightarrow{T}]\{y=V\}\{x=U\{y=V\}\}, \overrightarrow{U}]$. As a consequence $t\langle y=v\rangle\langle x=u\rangle\ \mathcal{H}\ M$, since by (3) we get $u\ \mathcal{H}\ U\{y=V\}$.
- The associativity and commutativity of contraction are very similar.
- If $W_x(W_y(t))\ \mathcal{H}\ M$ then $M = [[T, \overrightarrow{T}], \overrightarrow{U}]$ with $t\ \mathcal{H}\ T$, $y \in FV(T)$ and $x \in FV([T, \overrightarrow{T}])$. Then $W_y(W_x(t))\ \mathcal{H}\ M$.

□

Theorem 64 (Simulation in λI)

1. $\longrightarrow_{\beta,\pi}$ *strongly simulates* $\longrightarrow_B$ *through* $\mathcal{H}$.
2. $\longrightarrow_{\beta,\pi}$ *weakly simulates* $\longrightarrow_{xr}$ *through* $\mathcal{H}$.

Proof:

B) $(\lambda x.p)\ u \longrightarrow p\langle x = u\rangle$.
Then $T = [[\lambda x.P, \overrightarrow{P}]U, \overrightarrow{U}]$ with $p\ \mathcal{H}\ P$ and $u\ \mathcal{H}\ U$. We then obtain the following reduction sequence $T \longrightarrow^*_\pi [(\lambda x.P)U, \overrightarrow{P}, \overrightarrow{U}] \longrightarrow_\beta [P\{x = U\}, \overrightarrow{P}, \overrightarrow{U}] = T'$.

Abs) $(\lambda y.p)\langle x = u\rangle \longrightarrow \lambda y.p\langle x = u\rangle$. Then $T = [[\lambda y.P, \overrightarrow{P}]\{x = U\}, \overrightarrow{U}]$ with $p\ \mathcal{H}\ P$ and $u\ \mathcal{H}\ U$. We have $T = [\lambda y.(P\{x = U\}), \overrightarrow{P\{x = U\}}, \overrightarrow{U}]$.

*App*1,*App*2) Similar to the previous case.

Var) $x\langle x = u\rangle \longrightarrow u$. Then $T = [[x, \overrightarrow{P}]\{x = U\}, \overrightarrow{U}]$ with $u\ \mathcal{H}\ U$.
We have $T = [U, \overrightarrow{P\{x = U\}}, \overrightarrow{U}]$.

*Weak*1) $W_x(p)\langle x = u\rangle \longrightarrow W_{FV(u)}(p)$.
Then $T = [[P, \overrightarrow{P}]\{x = U\}, \overrightarrow{U}]$ with $p\ \mathcal{H}\ P$, $u\ \mathcal{H}\ U$, and $x \in FV(P)$. We have $T = [P\{x = U\}, \overrightarrow{P\{x = U\}}, \overrightarrow{U}]$. Since $x \notin FV(p)$, then $p\ \mathcal{H}\ P\{x = U\}$ by Lemma 63 (3), and since $x \in FV(P)$, $FV(U) \subseteq FV(P\{x = U\})$. By Lemma 63 (1) $FV(u) \subseteq FV(U)$ so that $FV(u) \subseteq FV(P\{x = U\})$ concludes the proof.

*Weak*2) $W_y(p)\langle x = u\rangle \longrightarrow W_y(p\langle x = u\rangle)$.
Then $T = [[P, \overrightarrow{P}]\{x = U\}, \overrightarrow{U}]$ with $p\ \mathcal{H}\ P$, $u\ \mathcal{H}\ U$, and $y \in FV(P)$. We have $T = [P\{x = U\}, \overrightarrow{P\{x = U\}}, \overrightarrow{U}]$ and we still have $y \in FV(P\{x = U\})$.

*Cont*1) $C^{y,z}_x(p)\langle x = u\rangle \longrightarrow C^{\Delta,\Pi}_\Gamma(p\langle y = R^\Gamma_\Delta(u)\rangle\langle z = R^\Gamma_\Pi(u)\rangle)$.
Then $T = [[P\{y = x\}\{z = x\}, \overrightarrow{P}]\{x = U\}, \overrightarrow{U}]$ with $p\ \mathcal{H}\ P$ and $u\ \mathcal{H}\ U$. We obtain the following equality $T = [P\{y = U\}\{z = U\}, \overrightarrow{P\{x = U\}}, \overrightarrow{U}]$ which can be expressed as

$$T = [P\{y = U'\}\{z = U''\}\{\Delta = \Gamma\}\{\Pi = \Gamma\}, \overrightarrow{P\{x = U\}}, \overrightarrow{U}]$$

where $U' = U\{\Gamma = \Delta\}$ and $U'' = U\{\Gamma = \Pi\}$. We obtain $R^\Gamma_\Delta(u)\ \mathcal{H}\ U'$ and $R^\Gamma_\Pi(u)\ \mathcal{H}\ U''$ by Lemma 63 (5).

*Cont*2) $C^{y,z}_w(p)\langle x = u\rangle \longrightarrow C^{y,z}_w(p\langle x = u\rangle)$.
Then $T = [[P\{y = w\}\{z = w\}, \overrightarrow{P}]\{x = U\}, \overrightarrow{U}]$ with $p\ \mathcal{H}\ P$ and $u\ \mathcal{H}\ U$. We then conclude by the following equality $T = [P\{x = U\}\{y = w\}\{z = w\}, \overrightarrow{P\{x = U\}}, \overrightarrow{U}]$.

Comp) $p\langle y = v\rangle\langle x = u\rangle \longrightarrow p\langle y = v\langle x = u\rangle\rangle$ where $x \in FV(v)$.
Then $T = [[P\{y = Q\}, \overrightarrow{P}]\{x = U\}, \overrightarrow{U}]$ with $t\ \mathcal{H}\ P$, $v\ \mathcal{H}\ Q$, and $u\ \mathcal{H}\ U$. We have $T = [P\{x = U\}\{y = Q\{x = U\}\}y, \overrightarrow{P\{x = U\}}, \overrightarrow{U}]$. Notice that we obtain $t\ \mathcal{H}\ P\{x = U\}$ by Lemma 63 (3).

- $WAbs$, $WApp1$, $WApp2$, $Cross$ are straightforward because the condition $x \in FV(P)$ that is checked by $W_x()$ is just changed into a side-condition $x \in FV(Q)$ (checked one step later), where $x \in FV(P)$ implies $x \in FV(Q)$.

Merge) $C_w^{y,z}(W_y(p)) \longrightarrow R_w^z(p)$.
Then $T = [[P, \overrightarrow{P}]\{y = w\}\{z = w\}, \overrightarrow{U}]$ with $t \mathcal{H} P$ and $y \in FV(P)$. We then have the following equality $T = [[P\{z = w\}, \overrightarrow{P\{z = w\}}]\{y = w\}, \overrightarrow{U}]$ and it suffices to use Lemma 63 (3).

CAbs) $C_w^{y,z}(\lambda x.t) \longrightarrow \lambda x.C_w^{y,z}(p)$.
Then $T = [[\lambda x.P, \overrightarrow{P}]\{y = w\}\{z = w\}, \overrightarrow{U}]$ with $t \mathcal{H} P$.
We have $T = [\lambda x.(P\{y = w\}\{z = w\}), \overrightarrow{P\{y = w\}y\{z = w\}}, \overrightarrow{U}]$.

*CApp*1, *CApp*2) Similar to the previous case.

Now for the closure under context, we use the fact that if $P \longrightarrow_{\beta,\pi} P'$ then $P\{x = U\} \longrightarrow_{\beta,\pi} P'\{x = U\}$, and if also $x \in FV(P)$ then $P\{x = U\} \longrightarrow^+_{\beta,\pi} P\{x = U'\}$. The latter is useful for the closure: if $p\langle x = t\rangle \mathcal{H} Q$ and $t \longrightarrow_B t'$, then $Q = [P\{x = T\}, \overrightarrow{U}]$ with $p \mathcal{H} P$, $u \mathcal{H} U$ and by induction hypothesis we get $T \longrightarrow^+_{\beta,\pi} T'$ such that $t' \mathcal{H} T'$. Since $x \in FV(p)$, $x \in FV(P)$ by Lemma 63 (2), and hence $Q \longrightarrow^+_{\beta,\pi} [P\{x = T'\}, \overrightarrow{U}]$. □

Corollary 65 *If $t \mathcal{H} T$ and $T \in \mathcal{SN}^{\beta,\pi}$, then $t \in \mathcal{SN}^{\lambda lxr}$.*

Proof: Application of Corollary 18. □

We can conclude the proof of PSN by stating the following theorem:

Theorem 66 *For any λ-term u, $\mathsf{A}(u) \mathcal{H} \mathsf{i}(u)$.*

Proof: By induction on u:

- $x \mathcal{H} x$ trivially holds.
- If $u = \lambda x.t$, then $\mathsf{A}(t) \mathcal{H} \mathsf{i}(t)$ holds by induction hypothesis. Therefore, we obtain $\lambda x.\mathsf{A}(t) \mathcal{H} \lambda x.\mathsf{i}(t)$ and $\lambda x.W_x(\mathsf{A}(t)) \mathcal{H} \lambda x.[\mathsf{i}(t), x]$.
- If $u = (t\ u)$, then $\mathsf{A}(t) \mathcal{H} \mathsf{i}(t)$ and $\mathsf{A}(u) \mathcal{H} \mathsf{i}(u)$ hold by induction hypothesis and $R^\Gamma_\Pi(\mathsf{A}(t)) \mathcal{H} R^\Gamma_\Pi(\mathsf{i}(t))$ and $R^\Gamma_\Pi(\mathsf{A}(u)) \mathcal{H} R^\Gamma_\Pi(\mathsf{i}(u))$ by Lemma 63 (5). Since $R^\Gamma_\Pi(\mathsf{i}(t))\{\Pi = \Gamma\} = \mathsf{i}(t)$ (and the same for $\mathsf{i}(u)$), we can then conclude $C^{\Delta,\Pi}_\Gamma(R^\Gamma_\Delta(\mathsf{A}(t))\ R^\Gamma_\Pi(\mathsf{A}(u))) \mathcal{H} \mathsf{i}(t)\ \mathsf{i}(u)$.

□

Corollary 67 (PSN) *For any λ-term t, if $t \in \mathcal{SN}^\beta$, then $\mathcal{A}(t) \in \mathcal{SN}^{\lambda lxr}$.*

Proof: If $t \in \mathsf{SN}^\beta$, then $\mathsf{i}(t) \in \mathsf{SN}^{\beta,\pi}$ by Corollary 60. As $\mathsf{A}(t) \mathcal{H} \mathsf{i}(t)$ by Theorem 66, then we conclude $\mathsf{A}(t) \in \mathsf{SN}^{\lambda lxr}$ by Corollary 65.

□

Conclusion

In this report we have developed a constructive theory of normalisation and induction based on an original approach that relies on second-order quantification rather than classical logic. We have re-established a few normalisation results in this framework, including the simulation technique and a few variants.

We have introduced two new developments to the simulation technique. The first one, called the Safeness and Minimality technique, can be applied to any higher-order rewrite system. The second one concerns more specifically systems that can be related to λ-calculus, and uses Church-Klop's λI-calculus.

For the two introduced techniques, which can be combined, examples of applications have been given with the calculi $\lambda\mathsf{x}$ [BR95], $\overline{\lambda}$ [Her95], and $\lambda\mathsf{lxr}$ [KL05].

Normalisation results have been inferred from the techniques, among which the property called Preservation of Strong Normalisation. The latter was known for $\lambda\mathsf{x}$ and $\overline{\lambda}$, but the Safeness and Minimality technique shortens the existing proofs for $\overline{\lambda}$ [DU03, Kik04]. The PSN property in $\lambda\mathsf{lxr}$ is a new result, which makes it the first calculus of explicit substitutions with full composition that satisfies it (together with a calculus in [Pol04] that has been developed simultaneously and independently).

We should check that Nederpelt's result that weak normalisation in λI implies strong normalisation can be proved constructively, so that the whole technique of simulation in λI is constructive.

Also, the examples for the safeness and minimality technique rely on a few external results such as the termination of the *lexicographic path ordering* [KL80], which has been proven in a framework with traditional definitions of normalisation. The latter are classically equivalent to ours, so that we can classically use them.

However, although the Safeness and Minimality technique is classical, it would be interesting to prove the LPO technique in our constructive framework, which is left as future work.

Acknowledgements

The author is grateful to Delia Kesner, Roy Dyckhoff, Alexandre Miquel and Ralph Matthes for their valuable comments and remarks.

References

[Bar84] H. P. Barendregt. *The Lambda-Calculus, its syntax and semantics*. Studies in Logic and the Foundation of Mathematics. Elsevier Science Publishers B. V. (North-Holland), Amsterdam, 1984. Second edition.

[BBLRD96] Z. Benaissa, D. Briaud, P. Lescanne, and J. Rouyer-Degli. $\lambda\upsilon$, a calculus of explicit substitutions which preserves strong normalisation. *Journal of Functional Programming*, 6(5):699–722, Sept. 1996.

[BN98] F. Baader and T. Nipkow. *Term rewriting and all that*. Cambridge University Press, 1998.

[BR95] R. Bloo and K. H. Rose. Preservation of strong normalisation in named lambda calculi with explicit substitution and garbage collection. In *CSN '95 – Computer Science in the Netherlands*, pages 62–72, Koninklijke Jaarbeurs, Utrecht, Nov. 1995.

[CD78] M. Coppo and M. Dezani-Ciancaglini. A new type assignment for lambda-terms. *Archive f. math. Logic u. Grundlagenforschung*, 19:139–156, 1978.

[Coq94] T. Coquand. An analysis of Ramsey's theorem. *Information and Computation*, 110(2):297–304, 1994.

[DU03] R. Dyckhoff and C. Urban. Strong normalization of Herbelin's explicit substitution calculus with substitution propagation. *Journal of Logic and Computation*, 13(5):689–706, 2003.

[Gen35] G. Gentzen. Investigations into logical deduction. In *Gentzen collected works*, pages 68–131. Ed M. E. Szabo, North Holland, (1969), 1935.

[Gir87] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–101, 1987.

[Her95] H. Herbelin. *Séquents qu'on calcule*. PhD thesis, Université Paris 7, 1995.

[JM03] F. Joachimski and R. Matthes. Short proofs of normalization for the simply-typed lambda-calculus, permutative conversions and Gödel's T . *Archive for Mathematical Logic*, 42(1):59–87, 2003.

[Kha90] Z. Khasidashvili. Expression reduction systems. In *Proceedings of IN Vekua Institute of Applied Mathematics*, volume 36, Tbilisi, 1990.

[Kik04] K. Kikuchi. A direct proof of strong normalization for an extended Herbelin's calculus. In Y. Kameyama and P. J. Stuckey, editors, *Proceedings of the 7th International Symposium on Functional and Logic Programming (FLOPS'04)*, volume 2998 of *Lecture Notes in Computer Science*, pages 244–259. Springer-Verlag, Apr. 2004.

[KL80] S. Kamin and J.-J. Lévy. Attempts for generalizing the recursive path orderings. Handwritten paper, University of Illinois, 1980.

[KL05] D. Kesner and S. Lengrand. Extending the explicit substitution paradigm. In J. Giesl, editor, *16th International Conference on Rewriting Techniques and Applications*, volume 3467 of *Lecture Notes in Computer Science*, pages 407–422. Springer-Verlag, Apr. 2005.

[Klo80] J.-W. Klop. *Combinatory Reduction Systems*, volume 127 of *Mathematical Centre Tracts*. CWI, Amsterdam, 1980. PhD Thesis.

[Laf90] Y. Lafont. Interaction nets. In *17th Annual ACM Symposium on Principles of Programming Languages (POPL)*, pages 95–108. ACM, 1990.

[LLD+04] S. Lengrand, P. Lescanne, D. Dougherty, M. Dezani-Ciancaglini, and S. van Bakel. Intersection types for explicit substitutions. *Information and Computation*, 189(1):17–42, 2004.

[Ned73] R. Nederpelt. *Strong Normalization in a Typed Lambda Calculus with Lambda Structured Types*. PhD thesis, Eindhoven University of Technology, 1973.

[Nip91] T. Nipkow. Higher-order critical pairs. In *6th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 342–349. IEEE Computer Society Press, July 1991.

[Pol04] E. Polonovski. *Substitutions explicites, logique et normalisation*. Thèse de doctorat, Université Paris 7, 2004.

[Sor97] M. H. Sorensen. Strong normalization from weak normalization in typed lambda-calculi. *IandC*, 37:35–71, 1997.

[Ter03] Terese. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.

[vRSSX99] F. van Raamsdonk, P. Severi, M. H. B. Sørensen, and H. Xi. Perpetual reductions in λ-calculus. *Information and Computation*, 149(2):173–225, 15 Mar. 1999.

[Xi97] H. Xi. Weak and strong beta normalisations in typed lambda-calculi. In P. de Groote, editor, *Proceedings of the 3th International Conference on Typed Lambda Calculus and Applications*, volume 1210 of *Lecture Notes in Computer Science*, pages 390–404. Springer-Verlag, Apr. 1997.

[Zuc74] J. Zucker. Correspondence between cut-elimination and normalization. *Annals of Mathematical Logic*, 7:1–156, 1974.