Metatheoretic Results for a Modal Lambda Calculus

Pierre Leleu

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Metatheoretic results for a modal lambda calculus

Pierre Leleu

N° 3361

Février 1998

THÈME 2

ISSN 0249-6399

INRIA
SOPHIA ANTIPOLIS

Metatheoretic results for a modal lambda calculus

Pierre Leleu

Thème 2 — Génie logiciel
et calcul symbolique
Projet Croap

Rapport de recherche n° 3361 — Février 1998 — 38 pages

Abstract: This paper presents the proofs of the strong normalization, subject reduction, and Church-Rosser theorems for a presentation of the intuitionistic modal lambda calculus S4. It is adapted from Healfdene Goguen's thesis, where these properties are shown for the simply-typed lambda calculus and for UTT. Following this method, we introduce the notion of typed operational semantics for our system. We define a notion of typed substitution for our system, which has context stacks instead of usual contexts. This latter peculiarity leads to the main difficulties and consequently to the main original features in our proofs. Since the original proof was extended to an inductive setting, we expect our proof could also be extended to a calculus with higher order abstract syntax and induction.

Key-words: MODAL LOGIC, LOGICAL FRAMEWORK, TYPE THEORY, STRONG NORMALIZATION, CONFLUENCE

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, B.P. 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : 04 92 38 77 77 - International : +33 4 92 38 77 77 — Fax : 04 92 38 77 65 - International : +33 4 92 38 77 65

Résultats métathéoriques pour un lambda calcul modal

Résumé : Nous présentons dans ce travail les preuves des théorèmes de forte normalisation, conservation des types et Church-Rosser pour une présentation du calcul modal intuitionniste IS4. Nos démonstrations s'inspirent de la thèse de Healfdene Goguen, dans laquelle les mêmes propriétés sont établies pour le lambda calcul simplement typé et UTT. A la suite de H. Goguen, nous définissons les notions de sémantique opérationnelle typée et de substitution typée pour notre système. Ce dernier met en jeu des piles de contextes au lieu des contextes usuels. Cette particularité est à l'origine des principales difficultés et par conséquent des principales nouveautés introduites dans les preuves. Comme la méthode originale a été étendue avec succès aux types inductifs (UTT), nous espérons pouvoir étendre notre travail à un calcul où la modalité permet de mélanger la syntaxe abstraite d'ordre supérieur avec l'induction.

Mots-clés : LOGIQUE MODALE, THEORIE TYPEE, FORTE NORMALISATION, CONFLUENCE

1 Introduction

We present here proofs of metatheoretic results for modal λ-calculus IS4 (see for example [Che90] for a classification of modal logics), in the presentation by Frank Pfenning and Hao-Chi Wong [PW95]. We have chosen this variant because the terms generated by the syntax are simpler than those of [BdP96], [PW95] or [DP96], since we have no 'let' construction.

The proof follows Healfdene Goguen's method [Gog94, Gog95], called 'Typed Operational Semantics'. It appeared to be surprisingly difficult to apply the method to the modal case, because the type system we chose uses context stacks. In particular, we had to extend the notion of typed substitution, which is central to the method, in a non-trivial way.

As a result, we get proofs of the strong normalization, subject reduction, and Church-Rosser theorems for our modal calculus IS4. These results are already well known. They can be proved fairly easily by interpreting modal λ-terms into simply-typed λ-terms. However, we cannot expect to extend this interpretation method to a stronger calculus, since it relies on the existence of a corresponding non modal λ-calculus.

The adaptation of H. Goguen's method to the modal setting proved to be challenging. Yet it is promising : since H. Goguen already extended his proof to an inductive setting (namely UTT), we expect we could also extend this proof to a type system where modality helps mixing higher order abstract syntax and induction (cf. [DPS97]).

The structure of the paper follows the plan outlined by Healfdene Goguen in his method. We first introduce the type system for our calculus, then give the reduction rules and the Typed Operational Semantics (TOS). The next section presents metatheoretical results for the TOS. At this point we prove that if a term has a reduction in the TOS then it verifies the subject reduction, strong normalization and Church-Rosser properties. Then a soundness result, namely that if a term is well-typed then it reduces to a normal form in the TOS, enables us to transfer these properties onto the original type system. Finally, we give two annexes: the first one presents a variant of our system with syntax-driven typing rules, the second one is a technical development of a tricky sub-part of the proof.

2 The type system

We present here F. Pfenning and H-C. Wong's system of modal lambda-calculus IS4 [PW95]. Then, we recall briefly its basic properties.

The sets of terms and contexts are generated by the usual syntax, with an additional definition for context stacks, as follows:

Types	A	$::= c \mid A \to A' \mid \Box A$
Terms	t	$::= x \mid \lambda x : A.t' \mid (t\ t') \mid \uparrow t \mid \downarrow t$
Contexts	Γ	$::= . \mid \Gamma, x : t$
Context stacks	Δ	$::= . \mid \Delta; \Gamma$

We use c for type constants. Instead of being declared in simple contexts, variables are declared in context stacks, i.e. ordered lists of contexts.

Notations A *valid* context (resp. context stack) is a context (resp. context stack) where all the variables are distinct. We call *domain* of a context Γ (resp. a stack Δ), denoted by $\mathrm{dom}(\Gamma)$ (resp. $\mathrm{dom}(\Delta)$), the set of the variables declared in this context (resp. stack). The notation Δ, Γ, where Δ is a stack $.;\Gamma_1;\cdots;\Gamma_n$ and Γ is a context, is the stack $.;\Gamma_1;\cdots;\Gamma_n,\Gamma$. Similarly, the notation Δ, Δ', where Δ is the stack $.;\Gamma_1;\cdots;\Gamma_n$ and Δ' is the stack $.;\Gamma'_1;\cdots;\Gamma'_m$, denotes the stack $.;\Gamma_1;\cdots;\Gamma_n,\Gamma'_1;\cdots;\Gamma'_m$. The notation $\Delta;\Delta'$, where Δ is the stack $.;\Gamma_1;\cdots;\Gamma_n$ and Δ' is the stack $.;\Gamma'_1;\cdots;\Gamma'_m$, denotes the stack $.;\Gamma_1;\cdots;\Gamma_n;\Gamma'_1;\cdots;\Gamma'_m$.

Examples

- $(.;x:A;.);(.;z:C;u:D) = (.;x:A;.;z:C;u:D)$
- $(.;x:A;y:B),(.;z:C;u:D) = (.;x:A;y:B,z:C;u:D)$
- $(.;x:A;.),(.;z:C;u:D) = (.;x:A;z:C;u:D)$
- $(.;x:A;.),(.;.;z:C) = (.;x:A;.;z:C)$

Note Instead of requiring a context stack to be valid in the Var and Pop rules below, in [PW95] it is assumed by default that any variable can only be declared at most once in a context stack.

The $\uparrow$ operator introduces an object of type $\Box A$, while the $\downarrow$ operator marks the elimination of a term of type $\Box A$. The last context Γ_n of a stack $\Delta \equiv .;\Gamma_1;\cdots;\Gamma_n$ is called the *local context* of Δ. The idea is to begin a new segment of context each time we encounter a $\uparrow$ operator during type-checking (rule $\uparrow$). A context can only be popped (i.e. added to the current stack) when type-checking a sub-term of type $\Box$ (rule Pop).

We have one typing judgment : "$\Delta \vdash M : A$", which is taken to mean that the canonical form of M is an element of type A in context stack Δ. In short, we say that "M has type A in stack Δ". The complete system is the following one:

$$(\mathrm{Var})\ \frac{x:A\in\Gamma}{\Delta;\Gamma\vdash x:A}\ \Delta;\Gamma \text{ valid}$$

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

$$(\uparrow)\ \frac{\Delta;.\vdash M:A}{\Delta\vdash\uparrow M:\Box A} \qquad (\downarrow)\ \frac{\Delta\vdash M:\Box A}{\Delta\vdash\downarrow M:A} \qquad (Pop)\ \frac{\Delta\vdash M:\Box A}{\Delta;\Gamma\vdash M:\Box A}\ \Delta;\Gamma \text{ valid}$$

Because of the (Pop) rule, this system is not syntax-driven, i.e. given a judgment $\Delta \vdash M : A$, we cannot guess which typing rule was the last one applied. Indeed if $\Delta \vdash M : \Box B$ and if Δ is made of at least two contexts, the last rule applied can be (Pop) as well as the structural rule corresponding to the form of M. In Annex A, we introduce a variant of our system with an extra operator which marks applications of rule Pop. This system is syntax-driven but the reduction rules become more complicated.

Note It is easy to see that, because of their structure, terms of the form $((\uparrow M)\ N)$ and $\downarrow \lambda x : A.M$ are not well-typed in any valid stack. Indeed, if $((\uparrow M)\ N)$ were well-typed in a stack Δ, $\uparrow M$ could only have a type of the form $\Box A$ but at the same time $\uparrow M$ must have a type of the form $B \to C$ since it is on the left side of an application. This is a contradiction. A similar argument works for $\downarrow \lambda x : A.M$.

2.1 Basic properties

The typing rules enjoy the usual property that if M is well-typed and N is a subterm of M then N is well-typed too.

Lemma 2.1 *If M is well-typed and N is a subterm of M ($N \subseteq M$) then N is well-typed.*

Proof By induction on the proof of $N \subseteq M$.
□

The structure of a context stack is more complex than the structure of a simple context. The following lemmas express the basic stack manipulations that preserve well-typedness. Namely, if $\Delta;\Gamma \vdash M : A$, then M is still well-typed in a context stack where we have permuted declarations in a context, tranformed some semicolons into commas, removed unnecessary variables, etc. These lemmas are all easily proved by induction on the structure of the proof of the hypothesis.

Lemma 2.2 (Swapping)
Two variables of any context in a stack can be swapped arbitrarily:

$$(Swap)\ \frac{.;D_1;\cdots;\Gamma,x:B,\Gamma',y:C,\Gamma'';\cdots;D_n \vdash M : A}{.;D_1;\cdots;\Gamma,y:C,\Gamma',x:B,\Gamma'';\cdots;D_n \vdash M : A}$$

Lemma 2.3 (Thinning)
A fresh variable can be added anywhere in a stack:

$$(Thin)\ \frac{.;D_1;\cdots;D_i\cdots;D_n \vdash M : A}{.;D_1;\cdots;(D_i,x:B);\cdots;D_n \vdash M : A}\ for\ 1 \le i \le n,\quad x \notin dom(.;D_1;\cdots;D_n)$$

Lemma 2.4 (Modal weakening)
A "fresh" context (i.e. a context whose domain only contains fresh variables) can be added anywhere in a stack, except behind the local context:

$$(Weak)\ \frac{.;D_1;\cdots;D_i;D_{i+1};\cdots;D_n \vdash M : A}{.;D_1;\cdots;D_i;D;D_{i+1};\cdots;D_n \vdash M : A}\ .;D_1;\cdots;D_i;D;D_{i+1};\cdots;D_n\ valid\ (1 \leq i < n)$$

The declaration of a variable which does not appear as a fresh variable in the term to type is somewhat superfluous. The following lemma tells us that we are allowed to remove it from the stack, whatever its location in the stack:

Lemma 2.5 (Strengthening)
The following rule is admissible:

$$(Strengthening)\ \frac{.;D_1;\cdots;\Gamma,x:B,\Gamma';\cdots;D_n \vdash M : A}{.;D_1;\cdots;\Gamma,\Gamma';\cdots;D_n \vdash M : A}\ x \notin FV(M)$$

Lemma 2.6 (Fusion)
Two successive contexts of a stack can be merged:

$$(Fus)\ \frac{.;D_1;\cdots;D_i;D_{i+1};\cdots;D_n \vdash M : A}{.;D_1;\cdots;(D_i,D_{i+1});\cdots;D_n \vdash M : A}\ for\ 1 \leq i < n$$

In particular if $\Delta;\Gamma;. \vdash M : A$ then $\Delta;\Gamma \vdash M : A$. On the contrary, $.;x : A \vdash x : A$ but $.;x : A;. \not\vdash x : A$ in general (actually $.;x : A;. \vdash x : A$ iff A is of the form $\Box B$).

Note that, in general, splitting a context of a stack into two separate contexts does not preserve typing. For example, it is true that $.;f : c \to \Box c, x : c;. \vdash (f\ x) : \Box c$. However we do not have $.;f : c \to \Box c;x : c;. \vdash (f\ x) : \Box c$.

2.2 Substitution

We denote the substitution of N for the free variable x in M by $M[N/x]$. It is defined as usual to avoid the capture of free variables. The rules for the modal operators are as expected:

- $(\downarrow M)[P/x] \equiv \downarrow (M[P/x])$
- $(\uparrow M)[P/x] \equiv \uparrow (M[P/x])$

Lemma 2.7 (Admissibility of Subst)
The following rule is admissible:

$$(Subst)\ \frac{.;D_1;\cdots;\Gamma,x:B,\Gamma';\cdots;D_n \vdash M : A \qquad .;D_1;\cdots;\Gamma \vdash N : B}{.;D_1;\cdots;\Gamma,\Gamma';\cdots;D_n \vdash M[N/x] : A}$$

Proof By induction on the derivation of the first premise.
□

2.3 Inversion lemmas

We end this section by giving the inversion lemmas for our typing rules. The inversion lemmas tell us how to type the immediate subterms of a well-typed term. Because of the non-determinism of the typing system, we cannot always find with certainty which typing rule was the last one applied.

First, we give a definition for truncated context stacks.

Definition 2.8 *For a context stack Δ with $n+1$ contexts, we define the context stacks Δ^i and the contexts $\delta^i\Delta$ ($i \in I\!N$, $0 \leq i \leq n$) as follows: Δ^0 is the stack Δ itself and if $\Delta^i = \Psi;\Gamma$ then $\Delta^{i+1} = \Psi$ and $\delta^i = \Gamma$, so that:*

$$\Delta = (\Delta^1; \delta^0\Delta) = (\Delta^2; \delta^1\Delta; \delta^0\Delta) = \ldots = (.; \delta^n\Delta; \cdots; \delta^1\Delta; \delta^0\Delta)$$

Lemma 2.9 (Inversion lemmas) *(also called Generation lemmas):*

1. $\Delta \vdash x : \Box A \Rightarrow x : A \in \Delta$.
2. $\Delta;\Gamma \vdash x : A \;\&\; A \not\equiv \Box A' \Rightarrow x : A \in \Gamma$.
3. $(\Delta \vdash \lambda x : A.M \;:\; A \to B) \Rightarrow (\Delta, x : A \vdash M : B)$, *where the variable x has possibly be renamed so that $\Delta, x : A$ is valid.*
4. $(\Delta \vdash\uparrow M : \Box A) \Rightarrow (\Delta; . \vdash M : A)$
5. $(\Delta \vdash\downarrow N : A) \Rightarrow (\Delta \vdash N : \Box A)$
6. $(\Delta \vdash (M\ N) : B) \Rightarrow (\exists n \in N. \Delta^n \vdash M : A \to B \;\&\; \Delta^n \vdash N : A)$

Proof By induction on the derivation of the hypothesis. We use the basic lemmas of Section 2.1 to simplify the result in the fourth and fifth cases.

□

Note In spite of the non determinism of the typing system, nearly all the inversion rules are written as if the typing system was deterministic. Only the application rule is affected, and not too badly: if B is not of the form $\Box C$, the rule yielding $\Delta \vdash (M\ N) : B$ is (App) and n is equal to 0 in the sixth inversion rule. Otherwise, rule (App) is eventually reached after a certain number of applications of the (Pop) rule.

3 Untyped reduction

After describing the syntax of the system, we turn to its semantics. We begin our study by introducing the untyped reduction rules.

Definition 3.1 (Untyped reduction) *We introduce the following one-step reduction relations:*

(β) $(\lambda x : A.M\ N)$ β $M[N/x]$
(η) $\lambda x : A.(M\ x)$ η M *if* $x \notin FV(M)$
$(\beta_{\Box})$ $\downarrow\uparrow M$ $\beta_{\Box}$ M
$(\eta_{\Box})$ $\uparrow\downarrow M$ $\eta_{\Box}$ M

Note The $(\beta_{\Box})$ and $(\eta_{\Box})$ rules correspond to the elimination of the following patterns in the derivation tree:

$$\frac{\dfrac{\Delta;. \vdash M : A}{\Delta \vdash \uparrow M : \Box A}}{\Delta \vdash \downarrow\uparrow M : A} \qquad \frac{\dfrac{\Delta;. \vdash M : \Box A}{\Delta;. \vdash \downarrow M : A}}{\Delta \vdash \uparrow\downarrow M : \Box A}$$

Notice that if $\Delta;. \vdash M : A$ then $\Delta \vdash M : A$ (by Lemma 2.6).

The rules $(\beta_{\Box})$ and $(\eta_{\Box})$ are named after the rules (β) and (η). Indeed, just as the abstraction and the application are the constructor and the destructor of the arrow, $\uparrow$ and $\downarrow$ are the constructor and the destructor of $\Box$.

Definition 3.2 (Compatible Closure) *Let R be a relation on terms. Then the* compatible closure *of R, notation $M \hookrightarrow_R N$, is the least relation satisfying the following rules:*

$$(R_i)\ \frac{M\ R\ N}{M \hookrightarrow_R N} \qquad (\xi)\ \frac{M \hookrightarrow_R N}{\lambda x : A.M \hookrightarrow_R \lambda x : A.N}$$

$$(App_l)\ \frac{M \hookrightarrow_R P}{(M\ N) \hookrightarrow_R (P\ N)} \qquad (App_r)\ \frac{N \hookrightarrow_R P}{(M\ N) \hookrightarrow_R (M\ P)}$$

$$(\downarrow)\ \frac{M \hookrightarrow_R N}{\downarrow M \hookrightarrow_{R\downarrow} N} \qquad (\uparrow)\ \frac{M \hookrightarrow_R N}{\uparrow M \hookrightarrow_{R\uparrow} N}$$

Let the untyped relation $M \hookrightarrow N$ be the compatible closure of the reduction relations defined above. The reflexive and transitive closure (resp. reflexive, symmetrical and transitive closure) will be denoted by $\hookrightarrow_*$ (resp. by $=$).

Definition 3.3 (Normal form) *A term is normal iff it has no reduction for $\hookrightarrow$.*

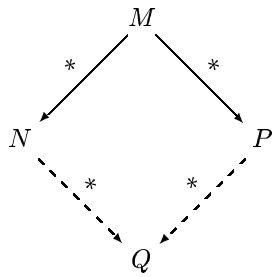
Lemma 3.4 (Forms of normal terms)
The normal forms can be characterized by induction:

- *Variables are normal,*
- *$\lambda x : A.M$ is normal if M is normal and not of the form $(N\ x)$ with $x \notin FV(N)$,*
- *$(M\ N)$ is normal if M and N are normal and M is not of the form $\lambda x : A.P$,*
- *$\uparrow M$ is normal if M is normal and not of the form $\downarrow N$,*
- *$\downarrow M$ is normal if M is normal and not of the form $\uparrow N$.*

Note As usual, if M is normal then all its subterms are normal too (obvious since reduction is compatible with all the operations).

Definition 3.5 (Strongly normalizing) *A term is* strongly normalizing *if all the reduction sequences starting from that term terminate.*

Definition 3.6 (Diamond property) *We say that a term satisfies the* diamond property *if whenever $M \hookrightarrow_* N$ and $M \hookrightarrow_* P$ then there exists a term Q such that $N \hookrightarrow_* Q$ and $P \hookrightarrow_* Q$.*



Lemma 3.7 (Substitution and reduction)

1. $M \hookrightarrow_* M' \Rightarrow M[N/x] \hookrightarrow_* M'[N/x]$.
2. $N \hookrightarrow_* N' \Rightarrow M[N/x] \hookrightarrow_* M[N'/x]$.
3. $(M \hookrightarrow_* M' \text{ and } N \hookrightarrow_* N') \Rightarrow M[N/x] \hookrightarrow_* M'[N'/x]$.

Proof The first two cases are proved by induction on M. The third one follows from the previous two results.
□

Now we define a typed judgment $\Delta \vdash M = N : A$, which means that M and N are equal objects of type A in stack Δ. More formally, this means that the canonical forms of M and N exist and that these canonical forms are equal terms of type A.

Definition 3.8 (Equality) *The judgment $\Delta \vdash M = N : A$ is defined as follows:*

$$(Refl)\ \frac{\Delta \vdash M : A}{\Delta \vdash M = M : A} \qquad (Sym)\ \frac{\Delta \vdash M = N : A}{\Delta \vdash N = M : A}$$

$$(Trans)\ \frac{\Delta \vdash M = N : A \qquad \Delta \vdash N = P : A}{\Delta \vdash M = P : A}$$

$$(\beta)\ \frac{\Delta, x : A \vdash M : B \qquad \Delta \vdash N : A}{\Delta \vdash (\lambda x : A.M\ \ N)\ =\ M[N/x]\ :\ B} \qquad (\eta)\ \frac{\Delta \vdash M : A \to B}{\Delta \vdash \lambda x : A.(M\ x) = M\ :\ A \to B}$$

$$(Eq\lambda)\ \frac{\Delta, x : A \vdash M = N : B}{\Delta \vdash \lambda x : A.M \ = \ \lambda x : A.N \ : \ A \to B}$$

$$(EqApp)\ \frac{\Delta \vdash M = P : A \to B \qquad \Delta \vdash N = Q : A}{\Delta \vdash (M\ N) \ = \ (P\ Q) : B}$$

$$(Eq\beta_{\Box})\ \frac{\Delta; . \vdash M : A}{\Delta \vdash \downarrow\uparrow M = M : A} \qquad (Eq\eta_{\Box})\ \frac{\Delta; . \vdash M : \Box A}{\Delta \vdash \uparrow\downarrow M = M : \Box A}$$

$$(Eq\downarrow)\ \frac{\Delta \vdash M = N : \Box A}{\Delta \vdash \downarrow M = \downarrow N : A} \qquad (Eq\uparrow)\ \frac{\Delta; . \vdash M = N : A}{\Delta \vdash \uparrow M = \uparrow N : \Box A}$$

$$(EqPop)\ \frac{\Delta \vdash M = N : \Box A}{\Delta; \Gamma \vdash M = N : \Box A}\ \Delta; \Gamma \ \textit{valid}$$

As expected, this equality expresses conversion between two well-typed terms.

Lemma 3.9 *$\Delta \vdash M = N : A$ iff ($\Delta \vdash M : A$, $\Delta \vdash N : A$ and $M = N$)*

Proof
($\Rightarrow$) By induction on the derivation of "$\Delta \vdash M = N : A$".
($\Leftarrow$) By induction on the derivation of "$M = N$", using inversion lemmas on "$\Delta \vdash M : A$" and "$\Delta \vdash N : A$".
□

4 Typed operational semantics

Following Healfdene Goguen ([Gog94]), we define a typed operational semantics based on standard reduction (or left-most reduction). Like him, we will not only give a reduction path from any term M to a normal form but we will also require that all the subterms of M have a normal form as well. Our operational system has the same form as H. Goguen's with additional rules for modal operators.

4.1 Weak head normal forms

Before introducing the system we need some preliminary definitions in order to state properly the side-conditions of some inference rules. These definitions are quite similar to the ones found in [Gog94]. We have used the similarities between application and $\downarrow$, and between abstraction and $\uparrow$ to extend the definitions to our modal setting.

Definition 4.1 (Redex) *We call* redex *either a β-redex or a $\beta_{\Box}$-redex.*

Definition 4.2 (Base term) *A term is a* base term *if it is a variable or if it is an application $(M\ N)$ and M is a base term or if it is a term of the form $\downarrow M$ and M is a base term.*

Definition 4.3 (Weak head normal) *We say that a term is* weak head normal (whn) *or in weak head normal form if it is not a redex and if it is of the form* $(M\ N)$ *or* $\downarrow M$ *then* M *is whn.*

So a term in weak head normal form is either a variable or an abstraction or a term of the form $\uparrow M$ or an application $(M\ N)$ where M is whn and not an abstraction, or a term $\downarrow M$ where M is whn and not of the form $\uparrow M'$.

Note A base term is always weak head normal.

The notion of weak head normal terms will be used later in the side-conditions of the typed operational semantics (see Section 4). The following results will be useful when proving metatheoretical results about the typed operational semantics.

Lemma 4.4

- *If* M *is normal then* M *is weak head normal.*
- *If* $(M\ N)$ *is weak head normal and* M *and* N *are normal then* $(M\ N)$ *is normal.*
- *If* $\downarrow M$ *is weak head normal and* M *is normal then* $\downarrow M$ *is normal.*
- *If* M *is weak head normal and* $M \hookrightarrow_{\beta\beta\Box *} N$ *then* N *is weak head normal.*

Proof Straightforward.
□

Note The proposition *"If* M *is weak head normal and* $M \hookrightarrow_* N$ *then* N *is weak head normal"*, which was true in the simply-typed λ-calculus, is false here. For example, $\uparrow (\lambda x : A.x\ \downarrow M)$ is weak head normal and reduces to M even if M is not weak head normal. Nevertheless, if we refine the hypotheses, we get similar results:

Lemma 4.5

- *If* $(M\ N)$ *is well-typed,* $(M\ N) \hookrightarrow_* R$ *and* $(M\ N)$ *is weak head normal, then* R *is weak head normal.*
- *If* $\downarrow M$ *is well-typed,* $\downarrow M \hookrightarrow_* R$ *and* $\downarrow M$ *is weak head normal, then* R *is weak head normal.*

Proof By induction on M.
□

4.2 The inference rules

Now we come to the actual inference rules of the typed operational semantics. They are strongly inspired by the original system [Gog94]. The main changes are the new rules for the modal operators, which are inspired by the analogies between $\downarrow$ and the application and between $\uparrow$ and λ.

$\Delta \vdash M \rightarrow_{nf} N : A$ means that M has canonical form N which is a canonical term of type A in stack Δ.
$\Delta \vdash M \rightarrow_{wh} N : A$ means that M weak head reduces to N of type A in stack Δ.

Normal forms:

$$(SVar)\ \frac{x : A \in \Gamma}{\Delta;\Gamma \vdash x \rightarrow_{nf} x : A}\ \Delta;\Gamma \text{ valid}$$

$$(S\lambda)\ \frac{\Delta, x : A \vdash M \rightarrow_{nf} P : B}{\Delta \vdash \lambda x : A.M \rightarrow_{nf} \lambda x : A.P : A \rightarrow B}\ \text{if } P \equiv (Q\ x)\ \Rightarrow\ x \in FV(Q)$$

$$(S\eta)\ \frac{\Delta, x : A \vdash M \rightarrow_{nf} (P\ x) : B \qquad \Delta \vdash P \rightarrow_{nf} Q : A \rightarrow B}{\Delta \vdash \lambda x : A.M \rightarrow_{nf} Q\ :\ A \rightarrow B}$$

$$(SApp)\ \frac{\Delta \vdash M \rightarrow_{nf} P : A \rightarrow B \qquad \Delta \vdash N \rightarrow_{nf} Q : A}{\Delta \vdash (M\ N) \rightarrow_{nf} (P\ Q) : B}\ \text{if } (M\ N) \text{ is whn}$$

$$(S\downarrow)\ \frac{\Delta \vdash M \rightarrow_{nf} N : \Box A}{\Delta \vdash \downarrow M \rightarrow_{nf} \downarrow N : A}\ \text{if } \downarrow M \text{ is whn}$$

$$(S\uparrow)\ \frac{\Delta;. \vdash M \rightarrow_{nf} N : A}{\Delta \vdash \uparrow M \rightarrow_{nf} \uparrow N : \Box A}\ \text{if } N \neq \downarrow P \qquad (S\eta_{\Box})\ \frac{\Delta;. \vdash M \rightarrow_{nf} \downarrow N : A}{\Delta \vdash \uparrow M \rightarrow_{nf} N : \Box A}$$

$$(SPop)\ \frac{\Delta \vdash M \rightarrow_{nf} N : \Box A}{\Delta;\Gamma \vdash M \rightarrow_{nf} N : \Box A}\ \Delta;\Gamma \text{ valid}$$

$$(SW)\ \frac{\Delta \vdash M \rightarrow_{wh} N : A \qquad \Delta \vdash N \rightarrow_{nf} P : A}{\Delta \vdash M \rightarrow_{nf} P : A}$$

Weak head reduction:

$$(W\beta)\ \frac{\Delta \vdash \lambda x : A.M \rightarrow_{nf} M' : A \rightarrow B \qquad \Delta \vdash N \rightarrow_{nf} N' : A}{\Delta \vdash (\lambda x : A.M \ \ N) \rightarrow_{wh} M[N/x] \ : \ B}$$

$$(WApp)\ \frac{\Delta \vdash M \rightarrow_{wh} P : A \rightarrow B \qquad \Delta \vdash N \rightarrow_{nf} N' : A}{\Delta \vdash (M\ N) \rightarrow_{wh} (P\ N) : B}$$

$$(W\beta_{\Box})\ \frac{\Delta \vdash\uparrow M \rightarrow_{nf} M' : \Box A}{\Delta \vdash\downarrow\uparrow M \rightarrow_{wh} M : A} \qquad (W\downarrow)\ \frac{\Delta \vdash M \rightarrow_{wh} N : \Box A}{\Delta \vdash\downarrow M \rightarrow_{wh}\downarrow N : A}$$

$$(WPop)\ \frac{\Delta \vdash M \rightarrow_{wh} N : \Box A}{\Delta;\Gamma \vdash M \rightarrow_{wh} N : \Box A}\ \Delta;\Gamma \text{ valid}$$

5 Metatheoretical results for the typed operational semantics

Now we study metatheoretical results for the Typed Operational Semantics (TOS). We first introduce the notion of typed substitutions between two stacks. It stems from H. Goguen's definition but it is modified to take modality into account. Then we present basic results about the TOS (stack manipulations, inversion lemmas). Finally, we prove that the reductions in the TOS correspond to actual untyped reductions (adequacy of reduction), and we show Subject Reduction, Strong Normalization and the diamond property for all the terms that have an $\rightarrow_{nf}$ reduction in the TOS. These latter results are interesting in themselves if our aim is to study the TOS but above all, they are intermediate results in our proof of the Subject Reduction, Strong Normalization and Church-Rosser properties for the untyped reduction.

5.1 Typed substitutions

Definition 5.1 (Pre-substitution) *A* pre-substitution *for a finite set of variables S is a function from S to terms.*

Definition 5.2 (Pre-renaming) *A* pre-renaming δ *for a finite set of variables S is a pre-substitution for S such that for each x in S, $(\delta\ x)$ is a variable.*

Notations Suppose Δ and Φ are context stacks, D_i and Γ are contexts, δ is a pre-substitution for $\mathrm{dom}(\Delta)$, ϕ is a pre-substitution for $\mathrm{dom}(\Phi)$, and ρ is a pre-substitution for $\mathrm{dom}(\Gamma)$. Then:

- We write $(\overline{\delta}\ M)$ for the result of simultaneously substituting the values for the variables in the domain of Δ:
$$(\overline{\delta}\ M) =_{def} M[(\delta\ x_1), \cdots, (\delta\ x_n)/x_1, \cdots, x_n]$$
- We write $\delta[x := M]$ for the extended pre-substitution for $\mathrm{dom}(\Delta, x : A)$ such that:
  - $(\delta[x := M]\ y) = (\delta\ y)$ if $y \in \mathrm{dom}(\Delta)$

  - $(\delta[x := M]\ x) = M$

- We write $\delta;\rho$ for the extended pre-substitution for $\mathrm{dom}(\Delta;\Gamma)$ such that:
  - $(\delta;\rho\ x) = (\delta\ x)$ if $x \in \Delta$
  - $(\delta;\rho\ x) = (\rho\ x)$ if $x \in \Gamma$
- The composition of δ and ϕ, $\delta \circ \phi$, is $(\delta \circ \phi\ \ x) = (\overline{\delta}(\phi\ x))$.
- If Δ' is a stack such that $\Delta' \subseteq \Delta$ (i.e. all the declarations of Δ' appear in Δ), then $\delta|\Delta'$ is the pre-substitution for $\mathrm{dom}(\Delta')$ such that $\forall x \in \mathrm{dom}(\Delta'), (\delta|\Delta'\ x) = (\delta\ x)$.

Defining a proper notion of typed substitution for our modal system is not obvious because we manipulate context stacks instead of contexts. A first attempt, too restrictive, would be to define a substitution ρ from Φ to Δ as a pre-substitution for $\mathrm{dom}(\Phi)$ such that for each $x : A \in \delta^i\Phi$, we have $\Delta^i \vdash (\rho\ x) : A$ (Besides, it would implicitly mean that Δ is necessarily a stack with more contexts than Φ). The following definition is more flexible:

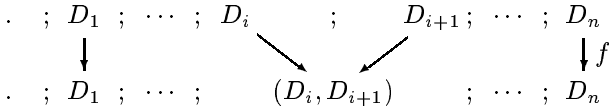
Definition 5.3 (Substitution) *A* substitution ρ *from* Φ *to* Δ*, where* Δ *and* Φ *are context stacks, is a pre-substitution for* $dom(\Phi)$ *such that there exists a non decreasing function* f *such that for each* $x : A \in \delta^i\Phi$ *we have* $\Delta^{f(i)} \vdash (\rho\ x) : A$*.*

Definition 5.4 (Renaming) *A* renaming δ *from* Φ *to* Δ *is a substitution from* Φ *to* Δ *such that* $(\delta\ x) = y$*, where* $y \in dom(\Delta)$*, for each* $x : A \in \Phi$*.*

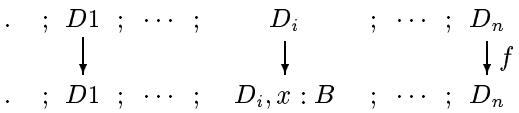
Definition 5.5 (Compatible context stacks) *A context stack* Δ *is said to be* compatible *with a context stack* Φ*, if for each context* $\delta^i\Phi$ *there exists a context* $\delta^j\Delta$ *such that* $\delta^j\Delta$ *has all declarations of* $\delta^i\Phi$*, and the function* $i \mapsto j$ *is a non decreasing function* f *s.t.* $f(0) = 0$*.*

Examples

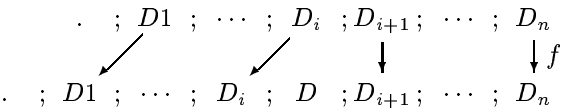
- $.; D_1; \cdots; (D_i, D_{i+1}); \cdots; D_n$ is compatible with $.; D_1; \cdots; D_n$ $\quad (f(n-i-1) = f(n-i) = n-i)$



- $.; D_1; \cdots; (D_i, x : B); \cdots; D_n$ is compatible with $.; D_1; \cdots; D_n$ $\quad (f(k) = k, \forall k \in \{1, \ldots, n\})$



- $.; D_1; \cdots; D_i; D; D_{i+1}; \cdots; D_n$ is compatible with $.; D_1; \cdots; D_n$ $\quad (f(n-i-1) = n-i-1, f(n-i) = n-i+1)$.



Notations

- If Δ is compatible with Φ then we write $weak^{\Delta}_{\Phi}$ for the pre-substitution $(weak^{\Delta}_{\Phi}\ x) = x$ for $dom(\Phi)$.

  Examples:

  - $weak^{.;D_1;\cdots;(D_i,D_{i+1});\cdots;D_n}_{.;D_1;\cdots;D_n}$ for $i \in [1..n[$
  - $weak^{.;D_1;\cdots;(D_i,x:B);\cdots;D_n}_{.;D_1;\cdots;D_n}$ for $i \in [1..n]$
  - $weak^{.;D_1;\cdots;D_i;D;D_{i+1};\cdots;D_n}_{.;D_1;\cdots;D_n}$ for $i \in [1..n[$
  - $weak^{\Delta}_{\Delta;.}$

- $id_{\Delta} =_{def} weak^{\Delta}_{\Delta}$

Lemma 5.6 *If Δ is a context stack compatible with Φ, then $weak^{\Delta}_{\Phi}$ is a renaming from Φ to Δ.*

Proof By definition of renamings.
□

Lemma 5.7 *Suppose δ is a substitution from Φ to Δ^n (for $n \geq 0$) and ρ is a substitution from $.;\Gamma$ to Δ. Then $\delta;\rho$ is a substitution from $\Phi;\Gamma$ to Δ.*

Proof By definition of substitutions and $\delta;\rho$.
□

In particular $\delta; weak^{\Delta;.}_{.;.}$ is a substitution from $\Phi;.$ to $\Delta;.$ that we shall use in several places in our proofs. Note that for any $x \in \Phi$ we have $(\delta; weak^{\Delta;.}_{.;.}\ x) = (\delta\ x)$.

Lemma 5.8 *For pre-renamings and pre-substitutions, we have:*

- $\delta \circ (\phi[x := N]) \equiv (\delta \circ \phi)[x := (\overline{\delta}\ N)]$
- $(\overline{\delta \circ \phi}\ M) \equiv (\overline{\delta}(\overline{\phi}\ M))$.
- $\delta \circ (\phi_1; \phi_2) \equiv (\delta \circ \phi_1); (\delta \circ \phi_2)$.

- $(\overline{\rho}\ \downarrow M) \equiv \downarrow (\overline{\rho}\ M)$ *and* $(\overline{\rho}\ \uparrow M) \equiv \uparrow (\overline{\rho}\ M)$.
- *Let* δ *be a pre-renaming from* Φ *to* Δ*, and let us assume that* $x \notin dom(\Phi)$, $y \notin dom(\Delta)$ *and* y *is not free in* M*, then* $(\overline{\delta}\ \lambda x : A.M) \equiv \lambda y : A.(\overline{\delta[x := y]}\ M)$.

Moreover, for renamings and substitutions, we have:

- *If* ϕ *is a substitution from* Φ *to* Δ *and* δ *is a substitution from* Δ *to* Θ *then* $\delta \circ \phi$ *is a substitution from* Φ *to* Θ.
- *If* δ *is a substitution from* Φ *to* Δ *and* f *the corresponding non decreasing function,* $\delta|\Phi^i$ *is a substitution from* Φ^i *to* $\Delta^{f(i)}$.
- *If* δ *is a substitution from* $\Phi;\Gamma;\Phi'$ *to* Δ *then* $\delta|\Phi;\Phi'$ *is a substitution from* $\Phi;\Phi'$ *to* Δ.
- *If* δ *is a renaming from* $\Phi;\Gamma$ *to* Δ *then* δ *is of the form* $\delta_1;\delta_2$*, where* δ_1 *is a renaming from* Φ *to* Δ^n *(*$n \in I\!N$ *) and* δ_2 *is a renaming from* $.;\Gamma$ *to* Δ.

Proof Straightforward.
□

5.2 Basic lemmas

Now, we state basic results about the typed operational semantics.

Lemma 5.9 (Free variables) *The following rule is valid for reductions* $\rightarrow_{nf}$ *and* $\rightarrow_{wh}$*:*
If $\Delta \vdash M \rightarrow N : A$ *then* $FV(M) \subseteq dom(\Delta)$ *and* $FV(N) \subseteq dom(\Delta)$.

Lemma 5.10 (Contexts) *If* $\Delta \vdash M \rightarrow_{nf} N : A$ *or* $\Delta \vdash M \rightarrow_{wh} N : B$ *then* Δ *is a valid context stack.*

Proof By induction on derivations.
□

Lemma 5.11 (Substitution preserves typing)
If $\Delta \vdash M : A$ *and* δ *is a renaming from* Δ *to* Φ*, then* $\Phi \vdash (\overline{\delta}\ M) : A$.

Proof By induction on derivations of $\Delta \vdash M : A$.
□

Lemma 5.12 (Renaming) *The following rule is valid for reductions* $\rightarrow_{nf}$ *and* $\rightarrow_{wh}$*:*
If δ *is a renaming from* Φ *to* Δ *and if* $\Phi \vdash M \rightarrow N : A$ *then* $\Delta \vdash (\overline{\delta}\ M) \rightarrow (\overline{\delta}\ N) : A$.

Proof By induction on derivations. The new difficult cases are rules $(S\ \uparrow)$, $(SPop)$ and $(WPop)$:

- $(S\uparrow)\ \dfrac{\Phi;. \vdash M \rightarrow_{nf} N : A}{\Phi \vdash \uparrow M \rightarrow_{nf} \uparrow N : \Box A}$ if $M \neq \downarrow P$

Let ϕ be $\delta; weak^{\Delta;.}_{.;.}$. ϕ is a renaming from $(\Phi;.)$ to $(\Delta;.)$. Thus by induction hypothesis,

$$\Delta;. \vdash (\overline{\phi}\ \ M) \rightarrow_{nf} (\overline{\phi}\ \ N) : A$$

By applying $(S\uparrow)$, we find that

$$\Delta \vdash (\overline{\phi}\ \ \uparrow M) \rightarrow_{nf} (\overline{\phi}\ \ \uparrow N) : \Box A$$

Since $(\overline{\phi}\ \ \uparrow M) = (\overline{\delta}\ \ \uparrow M)$ and $(\overline{\phi}\ \ \uparrow N) = (\overline{\delta}\ \ \uparrow N)$, we have proved that $\Delta \vdash (\overline{\delta}\ \ \uparrow M) \rightarrow_{nf} (\overline{\delta}\ \ \uparrow N) : \Box A$

- $(SPop)\ \dfrac{\Delta \vdash M \rightarrow_{nf} N : \Box A}{\Delta;\Gamma \vdash M \rightarrow_{nf} N : \Box A}\ \Delta;\Gamma$ valid

Let δ be a renaming from $\Phi;\Gamma$ to Δ. δ is of the form $\delta_1;\delta_2$ where δ_1 is a renaming from Φ to $\Delta^n (n \in \mathbb{N})$ and δ_2 is a renaming from $.;\Gamma$ to Δ. Since the free variables of M belong to $\mathrm{dom}(\Phi)$, one has $(\overline{\delta}\ \ M) = (\overline{\delta_1}\ \ M)$. Thus

$$\Delta^n \vdash (\overline{\delta}\ \ M) \rightarrow_{nf} (\overline{\delta}\ \ N) : \Box A$$

and after n successive applications of the $(SPop)$ rule :

$$\Delta \vdash (\overline{\delta}\ \ M) \rightarrow_{nf} (\overline{\delta}\ \ N) : \Box A$$

- The case $(WPop)$ is similar to the case $(SPop)$.

□

Thanks to our flexible definition of renaming, the renaming lemma can be seen as a generalization of several lemmas that deal with the preservation of $\rightarrow_{nf}$ and $\rightarrow_{wh}$ judgments by usual manipulations of stacks:

Corollary 5.13 (Thinning) *The following rule is valid for reductions $\rightarrow_{nf}$ and $\rightarrow_{wh}$:*

$$(Thinning)\ \frac{.; D_1; \cdots; D_i \cdots; D_n \vdash M \rightarrow N : A}{.; D_1; \cdots; (D_i, x : B); \cdots; D_n \vdash M \rightarrow N : A}\ x \notin dom(.; D_1; \cdots; D_n)\ (1 \leq i \leq n)$$

Proof Apply Lemma 5.12 with $\delta = \mathrm{weak}^{.;D_1;\cdots;(D_i,x:B);\cdots;D_n}_{.;D_1;\cdots;D_i\cdots;D_n}$.
□

Corollary 5.14 (Fusion) *The following rule is valid for reductions $\rightarrow_{nf}$ and $\rightarrow_{wh}$:*

$$(Fusion)\ \frac{.;D_1;\cdots;D_i;D_{i+1};\cdots;D_n \vdash M \to N : A}{.;D_1;\cdots;(D_i,D_{i+1});\cdots;D_n \vdash M \to N : A}\ for\ 1 \leq i < n$$

Proof Apply Lemma 5.12 with $\delta = \mathrm{weak}^{.;D_1;\cdots;(D_i,D_{i+1});\cdots;D_n}_{.;D_1;\cdots;D_i;D_{i+1};\cdots;D_n}$.
□

Corollary 5.15 (Weakening) *The following rule is valid for reductions $\to_{nf}$ and $\to_{wh}$:*

$$(Weakening)\ \frac{.;D_1;\cdots;D_i;D_{i+1};\cdots;D_n \vdash M \to N : A}{.;D_1;\cdots;D_i;D;D_{i+1};\cdots;D_n \vdash M \to N : A}\ for\ 1 \leq i < n$$

Proof Apply Lemma 5.12 with $\delta = \mathrm{weak}^{.;D_1;\cdots;D_i;D;D_{i+1};\cdots;D_n}_{.;D_1;\cdots;D_i;D_{i+1};\cdots;D_n}$.
□

Lemma 5.16 (Strengthening) *The following rule is valid for reductions $\to_{nf}$ and $\to_{wh}$:*

$$(Strengthening)\ \frac{.;D_1;\cdots;\Gamma,x:C,\Gamma';\cdots;D_n \vdash M \to N : A}{.;D_1;\cdots;\Gamma,\Gamma';\cdots;D_n \vdash M \to N : A}\ x \notin FV(M)$$

Proof By induction on derivations.
□

The following results analyse how a judgment $\Delta \vdash M \to_{nf} P : A$ can be obtained, according to the form of term M. As before, the inversion lemmas are affected by the non-determinism introduced by rule (Pop).

Lemma 5.17 (Inversion lemmas for $\to_{nf}$)

1. $\Delta \vdash x \to_{nf} x : \Box A \Rightarrow x : A \in \Delta$.
2. $\Delta;\Gamma \vdash x \to_{nf} x : A\ \&\ A \not\equiv \Box A' \Rightarrow x : A \in \Gamma$.
3. $\Delta \vdash \lambda x : A.M \to_{nf} Q : A \to B \Rightarrow (\Delta, x : A \vdash M \to_{nf} P : B$ *with either* $P \equiv (R\ x)$ *with* $x \notin FV(R)$, $\Delta \vdash R \to_{nf} S : A \to B\ \&\ Q \equiv S$, *or* $Q \equiv \lambda x : A.P$ *).*
4. $\Delta \vdash (M\ N) \to_{nf} R : B\ \&\ (M\ N)$ *is whn* $\Rightarrow \exists P,Q.R \equiv (P\ Q)\ \&\ \exists n \in I\!N.\Delta^n \vdash M \to_{nf} P : A \to B\ \&\ \Delta^n \vdash N \to_{nf} Q : A$ *where* $n = 0$ *if* $B \not\equiv \Box B'$.
5. $\Delta \vdash (M\ N) \to_{nf} P : A\ \&\ (M\ N)$ *is not whn* $\Rightarrow \Delta \vdash (M\ N) \to_{wh} M' : A\ \&\ \Delta \vdash M' \to_{nf} P : A$.
6. $\Delta \vdash\downarrow M \to_{nf} R : A\ \&\ \downarrow M$ *is whn* $\Rightarrow \exists N.R \equiv\downarrow N\ \&\ \Delta \vdash M \to_{nf} N : \Box A$
7. $\Delta \vdash\downarrow M \to_{nf} R : A\ \&\ \downarrow M$ *is not whn* $\Rightarrow \Delta \vdash\downarrow M \to_{wh} M' : A\ \&\ \Delta \vdash M' \to_{nf} R : A$.
8. $\Delta \vdash\uparrow M \to_{nf} Q : \Box A \Rightarrow \Delta;. \vdash M \to_{nf} P : A$ *with either* $Q \equiv\uparrow P\ \&\ P \not\equiv\downarrow P'$, *or* $P \equiv\downarrow Q$.

Proof By induction on derivations.
□

We also have inversion lemmas for judgment $\rightarrow_{wh}$.

Lemma 5.18 (Inversion lemmas for $\rightarrow_{wh}$)

1. $\Delta \vdash (M\ N) \rightarrow_{wh} Q : B \Rightarrow \exists n \in I\!N\ .\Delta^n \vdash N \rightarrow_{nf} N' : A$ *with either* $M \equiv \lambda x : A.P,\ Q \equiv P[N/x],\ \Delta^n \vdash \lambda x : A.P \rightarrow_{nf} M' : A \rightarrow B$, *or* $\Delta^n \vdash M \rightarrow_{wh} P : A \rightarrow B\ \&\ Q \equiv (P\ N)$.
2. $\Delta \vdash\downarrow M \rightarrow_{wh} N : A \Rightarrow$ *either* $M \equiv\uparrow P,\ N \equiv P,\ \Delta;. \vdash P \rightarrow_{nf} P' : A$, *or* $N \equiv\downarrow P,\ \Delta \vdash M \rightarrow_{wh} P : \Box A$.

Proof By induction on derivations.
□

Lemma 5.19 (Uniqueness of normal forms) *The following result is valid for reductions* $\rightarrow_{nf}$ *and* $\rightarrow_{wh}$*: if* $\Delta \vdash M \rightarrow P : A$ *and* $\Delta \vdash M \rightarrow Q : B$ *then* $P \equiv Q$ *and* $A \equiv B$.

Proof By induction on the proof of the first hypothesis, using the inversion lemmas on the second hypothesis.
□

Lemma 5.20 (Completeness)

- *If* $\Delta \vdash M \rightarrow_{nf} N : A$ *then* $\Delta \vdash M = N : A$.
- *If* $\Delta \vdash M \rightarrow_{wh} N : A$ *then* $\Delta \vdash M = N : A$.

Proof By induction on derivations.
□

5.3 Typed operational system and untyped reduction

In this section we establish the links between the typed operational semantics and the untyped reduction. First, we prove that the judgment $\Delta \vdash M \rightarrow_{nf} P : A$ actually computes a reduct of M which is normal. Similarly, the $\rightarrow_{wh}$ judgment corresponds to β and $\beta_\Box$ reductions.

Lemma 5.21 (Adequacy for reduction)

- *If* $\Delta \vdash M \rightarrow_{nf} P : A$ *then* $M \hookrightarrow_* P$ *and* P *is normal, and furthermore there is an* N *such that* $M \hookrightarrow_{\beta\beta_{\Box *}} N \hookrightarrow_{\eta\eta_{\Box *}} P$.
- *If* $\Delta \vdash M \rightarrow_{wh} P : A$ *then* $M \hookrightarrow_{\beta\beta_\Box} P$.

Proof By induction on derivations.
□

Lemma 5.22 *If* $\Delta \vdash M \rightarrow_{nf} N : A$ *and* M *is normal then* $M \equiv N$.

Proof By adequacy for reduction $M \hookrightarrow_* N$. Furthermore M has no reduction. Thus $M \equiv N$.
□

In the rest of the section, we prove important properties about judgment $\rightarrow_{nf}$, namely that if $\Delta \vdash M \rightarrow_{nf} P : A$ then M is strongly normalizing, and that if $\Delta \vdash M \rightarrow_{nf} P : A$ and $M \hookrightarrow N$ then $\Delta \vdash N \rightarrow_{nf} P : A$.

Unlike H. Goguen, we do not use an intermediate predicate (the logical meaning of which was unclear to us) but directly prove the properties. Also we do not need to define the stack in which a subterm of a well-typed term is itself typable (this would not have been as easy as in the simply typed λ-calculus).

We need some preliminary lemmas. The first two are necessary because of (η) and $(\eta_\Box)$ reductions.

Lemma 5.23 (Subject reduction for η) *If $\Delta \vdash \lambda x : A.(M\ x) \rightarrow_{nf} P : A \rightarrow B$ and $x \notin FV(M)$ then $\Delta \vdash M \rightarrow_{nf} P : A \rightarrow B$.*

This lemma is proved in Healfdene Goguen's thesis ([Gog94]) using inversion lemmas. We only give here the proof of its modal counterpart:

Lemma 5.24 (Subject reduction for $\eta_\Box$) *If $\Delta \vdash \uparrow\downarrow M \rightarrow_{nf} P : A$ then $\Delta \vdash M \rightarrow_{nf} P : A$.*

Proof Using the inversion lemmas, we obtain $\Delta;. \vdash\downarrow M \rightarrow_{nf} N : B$ (where $B \equiv \Box A$) with either $P \equiv\uparrow N$ (and N not of the form $\downarrow X$) or $N \equiv\downarrow P$. We now proceed by induction on the length of the proof of $\Delta;. \vdash\downarrow M \rightarrow_{nf} N : B$. By applying the inversion lemmas once again, we have two cases :

- $(S\downarrow)$ $\Delta \vdash M \rightarrow_{nf} N' : \Box B$ where $\downarrow M$ is whn and $N \equiv\downarrow N'$.
  - If $N \equiv\downarrow P$ then $P \equiv N'$ and the lemma is proved.
  - Otherwise N is of the form $\downarrow N'$, which is not permitted.
- (SW) $\dfrac{\Delta;. \vdash\downarrow M \rightarrow_{wh} R : B \qquad \Delta;. \vdash R \rightarrow_{nf} N : B}{\Delta;. \vdash\downarrow M \rightarrow_{nf} N : B}$

  We apply the inversion lemmas to the first premise:
  - $(W\downarrow\uparrow)$ $\Delta;. \vdash M' \rightarrow_{nf} M'' : B$, where $M \equiv\uparrow M'$ and $P \equiv M'$. Thus the second premise of (SW) gives us that $\Delta;. \vdash M' \rightarrow_{nf} N : B$. Depending on whether $P \equiv\uparrow N$ or $N \equiv\downarrow P$, $(S\uparrow)$ or $(S\eta_\Box)$ leads us to the result.
  - $(W\downarrow)$ $\Delta \vdash M \rightarrow_{wh} Q : \Box B$ where $\downarrow Q \equiv R$. In that case, the second premise of (SW) becomes $\Delta;. \vdash\downarrow Q \rightarrow_{nf} N : B$ and by induction hypothesis $\Delta \vdash Q \rightarrow_{nf} P : A$. Applying (SW), we finally obtain that $\Delta \vdash M \rightarrow_{nf} P : A$.

□

Now, we prove the important results of this section, using the technical developments of Annex B in the proofs.

Lemma 5.25 (Strong Normalization for $\rightarrow_{nf}$)

- *If $\Delta \vdash M \rightarrow_{nf} P : A$ then M is strongly normalizing.*
- *If $\Delta \vdash M \rightarrow_{wh} N : A$ and N is strongly normalizing then M is strongly normalizing.*

Proof By induction on the length of the derivations of the $\rightarrow_{nf}$ and $\rightarrow_{wh}$ judgments.

- (S Var) (S Pop) (SW) (W Pop) Easy.
- $(S\lambda)$ $\dfrac{\Delta, x : A \vdash M \rightarrow_{nf} P : B}{\Delta \vdash \lambda x : A.M \rightarrow_{nf} \lambda x : A.P : A \rightarrow B}$ if $P \equiv (Q\ x) \ \Rightarrow \ x \in \mathrm{FV}(Q)$

  We consider a sequence of reductions starting from $\lambda x : A.M$.

  - As long as there is no η-reduction of the top abstraction, we always reduce under it:

    $$\lambda x : A.M \hookrightarrow_* \lambda x : A.M'$$

    where $M \hookrightarrow_* M'$. By induction hypothesis M is strongly normalizing and we cannot obtain an infinite sequence of reductions this way.
  - Otherwise, we have a sequence of reductions with an eta-reduction at some point:

    $$\lambda x : A.M \hookrightarrow_* \lambda x : A.({M_1}'\ x) \hookrightarrow {M_1}' \hookrightarrow_* {M_1}''$$

    This means that we also have $M \hookrightarrow_* ({M_1}'\ x)$. Since M is strongly normalizing by induction hypothesis, so is ${M_1}'$ and we cannot build any infinite sequence of reductions this way either.
- $(S\eta)$ Similar to the previous case.
- $(S App)$ $\dfrac{\Delta \vdash M \rightarrow_{nf} P : A \rightarrow B \qquad \Delta \vdash N \rightarrow_{nf} Q : A}{\Delta \vdash (M\ N) \rightarrow_{nf} (P\ Q) : B}$ if $(M\ N)$ is whn

  We know that $(M\ N)$ is well-typed and weak head normal. By Lemma 4.5, if $(M\ N) \hookrightarrow_* R$ then R is still a weak head normal term $(M'\ N')$ with $M \hookrightarrow_* M'$ and $N \hookrightarrow_* N'$. By induction hypothesis M and N are strongly normalizing. Thus $(M\ N)$ is strongly normalizing.
- $(S\downarrow)$ Like above, using Lemma 4.5 and induction hypothesis to prove the result.
- $(S\uparrow)$ and $(S\eta_{\Box})$ Like $(S\lambda)$ and $(S\eta)$.
- $(W\beta)$ $\dfrac{\Delta \vdash \lambda x : A.M \rightarrow_{nf} M' : A \rightarrow B \qquad \Delta \vdash N \rightarrow_{nf} N' : A}{\Delta \vdash (\lambda x : A.M\ \ N) \rightarrow_{wh} M[N/x]\ :\ B}$

  We assume that $M[N/x]$ is strongly normalizing and we examine the possible sequences of reductions starting from $(\lambda x : A.M\ \ N)$:

- If $(\lambda x : A.M\ \ N) \hookrightarrow_* (\lambda x : A.M'\ \ N')$, where $M \hookrightarrow_* M'$ and $N \hookrightarrow_* N'$, we cannot have an infinite sequence of reductions because $\lambda x : A.M$ and N are strongly normalizing by induction hypothesis.
- Otherwise, $(\lambda x : A.M\ \ N) \hookrightarrow_* (\lambda x : A.M'\ \ N') \hookrightarrow M'[N'/x] \hookrightarrow_* R$. We notice that $M'[N'/x]$ is strongly normalizing since it is obtained by reducing $M[N/x]$, which is strongly normalizing. Thus we cannot have an infinite sequence of reductions this way either.

- $(W\beta_\Box)$ Similar to the previous case $(W\beta)$.
- $(W App)$ $\dfrac{\Delta \vdash M \rightarrow_{wh} P : A \rightarrow B \qquad \Delta \vdash N \rightarrow_{nf} N' : A}{\Delta \vdash (M\ N) \rightarrow_{wh} (P\ N) : B}$

  We assume that $(P\ N)$ is strongly normalizing. Thus P and N are strongly normalizing and by induction hypothesis M enjoys this property too. Like above we examine the possible sequences of reductions starting from $(M\ N)$:

  - If $(M\ N) \hookrightarrow_* (M'\ N')$, we cannot have an infinite sequence because M and N are strongly normalizing.
  - Otherwise, the left argument of the application is weak head reduced at some point :

    $$(M\ N) \hookrightarrow_* (M'\ N') \hookrightarrow_{wh} (M''\ N') \hookrightarrow_* R$$

    In this case we use the results of Annex B, Lemma 9.10 tells us that $(P\ N) \hookrightarrow_* (M''\ N')$. Thus $(M''\ N')$ is strongly normalizing and we cannot have any infinite sequence.

- $(W \downarrow)$ Similar to the previous case (W App).

□

Lemma 5.26 (Subject Reduction for $\rightarrow_{nf}$)

- *If* $\Delta \vdash M \rightarrow_{nf} P : A$ *and* $M \hookrightarrow_* N$ *then* $\Delta \vdash N \rightarrow_{nf} P : A$.
- *If* $\Delta \vdash M \rightarrow_{wh} N : A$, $\Delta \vdash N \rightarrow_{nf} P : A$ *and* $(N \hookrightarrow_* N' \Rightarrow \Delta \vdash N' \rightarrow_{nf} P : A)$ *then* $(M \hookrightarrow_* M' \Rightarrow \Delta \vdash M' \rightarrow_{nf} P : A)$

Proof By induction on the length of the derivations of the $\rightarrow_{nf}$ and $\rightarrow_{wh}$ judgments.

- (S Var), (S Pop), (SW) and (W Pop) are straightforward.
- $(S\lambda)$ $\dfrac{\Delta, x : A \vdash M \rightarrow_{nf} P : B}{\Delta \vdash \lambda x : A.M \rightarrow_{nf} \lambda x : A.P : A \rightarrow B}$ if $P \equiv (Q\ x) \ \Rightarrow\ x \in \mathrm{FV}(Q)$

  Like in the proof of the previous lemma we examine the possible sequences of reductions.

- If $\lambda x : A.M \hookrightarrow_* \lambda x : A.M'$ with $M \hookrightarrow_* M'$ then by induction hypothesis $\Delta, x : A \vdash M' \rightarrow_{nf} P : B$. Thus $\Delta \vdash \lambda x : A.M' \rightarrow_{nf} \lambda x : A.P : A \rightarrow B$
- Otherwise we have $\lambda x : A.M \hookrightarrow_* \lambda x : A.({M_1}' \ x) \hookrightarrow_\eta {M_1}' \hookrightarrow_* {M_1}''$ with an η-reduction at some point. Then $M \hookrightarrow_* ({M_1}'' \ x)$ and by induction hypothesis $\Delta, x : A \vdash ({M_1}'' \ x) \rightarrow_{nf} P : B$. Since $x \notin \mathrm{FV}({M_1}')$, we also have $x \notin \mathrm{FV}({M_1}'')$ and we can apply rule $(S\lambda)$, which gives $\Delta \vdash \lambda x : A.({M_1}'' \ x) \rightarrow_{nf} \lambda x : A.P : A \rightarrow B$. Finally by Lemma 5.23 (Subject reduction for η), $\Delta \vdash {M_1}'' \rightarrow_{nf} \lambda x : A.P : A \rightarrow B$.

- $(S\eta)$ Similar to the previous case $(S\lambda)$.
- $(SApp)$ $\dfrac{\Delta \vdash M \rightarrow_{nf} P : A \rightarrow B \qquad \Delta \vdash N \rightarrow_{nf} Q : A}{\Delta \vdash (M \ N) \rightarrow_{nf} (P \ Q) : B}$ if $(M \ N)$ is whn

  We know that $(M \ N)$ is well-typed and weak head normal. By Lemma 4.5, if $(M \ N) \hookrightarrow_* R$ then R is still a weak head normal application $(M' \ N')$ with $M \hookrightarrow_* M'$ and $N \hookrightarrow_* N'$. By induction hypothesis $\Delta \vdash M' \rightarrow_{nf} P : A \rightarrow B$ and $\Delta \vdash N' \rightarrow_{nf} Q : A$. By $(SApp)$, $\Delta \vdash (M' \ N') \rightarrow_{nf} (P \ Q) : B$.
- $(S \downarrow)$ Like $(SApp)$, using Lemma 4.5, induction hypothesis and $(S \downarrow)$ to prove the result.
- $(S \uparrow)$ and $(S\eta_\Box)$ Like $(S\lambda)$ and $(S\eta)$.
- $(W\beta)$ $\dfrac{\Delta \vdash \lambda x : A.M \rightarrow_{nf} M' : A \rightarrow B \qquad \Delta \vdash N \rightarrow_{nf} N' : A}{\Delta \vdash (\lambda x : A.M \ \ N) \rightarrow_{wh} M[N/x] \ : \ B}$

  We assume that $\Delta \vdash M[N/x] \rightarrow_{nf} R : B$ and that $(M[N/x] \hookrightarrow_* P' \Rightarrow \Delta \vdash P' \rightarrow_{nf} R : B)$. We examine the possible sequences of reductions starting from $(\lambda x : A.M \ \ N)$

  - If $(\lambda x : A.M \ \ N) \hookrightarrow_* (\lambda x : A.M' \ \ N')$, where $M \hookrightarrow_* M'$ and $N \hookrightarrow_* N'$, then by induction hypothesis, $\Delta \vdash \lambda x : A.M' \rightarrow_{nf} P'' : A \rightarrow B$ and $\Delta \vdash N' \rightarrow_{nf} Q'' : A$. Thus by $(W\beta)$, $\Delta \vdash (\lambda x : A.M' \ N') \rightarrow_{wh} M'[N'/x] : B$. Now, since $M[N/x] \hookrightarrow_* M'[N'/x]$, we have $\Delta \vdash M'[N'/x] \rightarrow_{nf} R : B$. By (S W), $\Delta \vdash (\lambda x : A.M' \ N') \rightarrow_{nf} R : B$
  - Otherwise, $(\lambda x : A.M \ \ N) \hookrightarrow_* (\lambda x : A.M' \ \ N') \hookrightarrow M'[N'/x] \hookrightarrow_* T$. In that case, $M[N/x] \hookrightarrow_* T$ and $\Delta \vdash T \rightarrow_{nf} R : B$.

- $(W\beta_\Box)$ Similar to the previous case $(W\beta)$.
- $(WApp)$ $\dfrac{\Delta \vdash M \rightarrow_{wh} P : A \rightarrow B \qquad \Delta \vdash N \rightarrow_{nf} N'' : A}{\Delta \vdash (M \ N) \rightarrow_{wh} (P \ N) : B}$

  We assume that $\Delta \vdash (P \ N) \rightarrow_{nf} R : B$ and that $(P \ N) \hookrightarrow_* P' \Rightarrow \Delta \vdash P' \rightarrow_{nf} R : B$. We examine the possible sequences of reductions starting from $(M \ N)$:

  - If $(M \ N) \hookrightarrow_* (M' \ N')$ without any weak head reduction step then $M \hookrightarrow_* M'$ without any weak head reduction step. Once again, we use the results of Annex B. By Lemma 9.10, there is a term Q such that $M' \hookrightarrow_{wh} Q$ and $P \hookrightarrow_* Q$. By Lemma 9.6, all the elements of M have a derivation $\rightarrow_{nf}$. By Lemma 9.9 and induction hypothesis, it is also

true for M'. Thus, by Lemma 9.11, $\Delta \vdash M' \rightarrow_{wh} Q : A \rightarrow B$. Now, since $(P\ N) \hookrightarrow_* (Q\ N')$, we have $\Delta \vdash (Q\ N') \rightarrow_{nf} R : B$. By (W App) and induction hypothesis for N, we have $\Delta \vdash (M'\ N') \rightarrow_{wh} (Q\ N') : B$. Finally by (S W), $\Delta \vdash (M'\ N') \rightarrow_{nf} R : B$.

– Otherwise, the left argument of the application is weak head reduced at some point :

$$(M\ N) \hookrightarrow_* (M'\ N') \hookrightarrow_{wh} (M''\ N') \hookrightarrow_* T$$

By Lemma 9.10, $P \hookrightarrow_* M''$. Thus we have $(P\ N) \hookrightarrow_* (M''\ N') \hookrightarrow_* T$, which implies by hypothesis $\Delta \vdash T \rightarrow_{nf} R : B$.

- $(W \downarrow)$ Similar to the previous case (W App).

□

Corollary 5.27 (Diamond Property for $\rightarrow_{nf}$) *If $\Delta \vdash M \rightarrow_{nf} Q : A$, $M \hookrightarrow_* N$ and $M \hookrightarrow_* P$ then $N \hookrightarrow_* Q$ and $P \hookrightarrow_* Q$.*

The following property will be useful when showing the Soundness lemma 6.8.

Proposition 5.28 (Admissibility of $S\eta'$)
The following rule is admissible:

$$(S\eta') \ \frac{\Delta, x : A \vdash M \rightarrow_{nf} (P\ x) : B \qquad x \notin FV(P)}{\Delta \vdash \lambda x : A.M \rightarrow_{nf} P \ : \ A \rightarrow B}$$

Proof We assume that $\Delta, x : A \vdash M \rightarrow_{nf} (P\ x) : B$. By Lemma Adequacy for Reduction, $(P\ x)$ is normal. Thus P is normal as well. We can show by an easy induction that for any normal term Q well-typed of type A in the stack Φ, we have $\Phi \vdash Q \rightarrow_{nf} Q : A$. Thus $\Delta \vdash P \rightarrow_{nf} P : B$. By applying $(S\ \eta)$, we find that $\Delta \vdash \lambda x : A.M \rightarrow_{nf} P \ : \ A \rightarrow B$.
□

6 Soundness

We have shown in the previous section the links between the typed operational semantics and the untyped reduction. Now it is time to clarify the links between the typing rules and the typed operational semantics. We will prove a result of strong normalization, namely that for any well-typed term M of type A in stack Δ, there is a normal form P such that $\Delta \vdash M \rightarrow_{nf} P : A$. The proof will be carried out quite classically, "à la Tait". Finally, the results of last section will enable us to deduce the classic properties of subject reduction, strong normalization and Church-Rosser for our system.

Definition 6.1 (Semantic Object) *A* semantic object *for Δ and A is a term M such that $\Delta \vdash M \rightarrow_{nf} P : A$ for some term P.*

Definition 6.2 (Interpretation of types) *Let Δ be a context stack. The* interpretation *of a type A in Δ, denoted by $[\![A]\!]_\Delta$, is given by induction on the structure of the type A:*

- *$[\![c]\!]_\Delta$ is the set of semantic objects for Δ and c.*
- *$[\![A \to B]\!]_\Delta$ is the set of semantic objects M for Δ and $A \to B$ such that, for any context Γ such that Δ, Γ is valid, and any $N \in [\![A]\!]_{\Delta,\Gamma}$ we know that $(M\ N) \in [\![B]\!]_{\Delta,\Gamma}$.*
- *$[\![\Box A]\!]_\Delta$ is the set of semantic objects M for Δ and $\Box A$ such that, for any stack Δ' such that Δ, Δ' is valid, we know that $(\downarrow M) \in [\![A]\!]_{\Delta,\Delta'}$.*

In his thesis, Healfdene Goguen uses renamings to define $[\![A \to B]\!]_\Delta$. Namely, $[\![A \to B]\!]_\Delta$ was the set of semantic objects M for Δ and $A \to B$ such that, for any renaming δ from Δ' to Δ and any $N \in [\![A]\!]_{\Delta'}$, $((\overline{\delta}\ M)\ N) \in [\![B]\!]_{\Delta'}$. Here we are implicitly using weakenings instead of renamings. Actually what we want is to be able to extend the contexts with fresh variables, this is precisely what weakenings do.

Lemma 6.3 (Weakening for $[\![A]\!]_\Delta$)

1. *If $M \in [\![A]\!]_\Delta$ and if Γ is a context such that Δ, Γ is valid then $M \in [\![A]\!]_{\Delta,\Gamma}$.*
2. *If $M \in [\![\Box A]\!]_\Delta$ and if Δ' is a stack such that $\Delta; \Delta'$ is valid then $M \in [\![\Box A]\!]_{\Delta;\Delta'}$. In particular if Γ is a context such that $\Delta; \Gamma$ is valid then $M \in [\![\Box A]\!]_{\Delta;\Gamma}$.*

Proof

1. By case analysis on the structure of type A. We consider the case $A \equiv \Box B$. If a context Γ is such that Δ, Γ is valid then let Δ' be a stack such that Δ, Γ, Δ' is valid. Δ, Γ, Δ' is of the form Δ, Δ'' with $\Delta'' = \Gamma, \Delta'$. By definition of $[\![\Box B]\!]_\Delta$, $\downarrow M \in [\![B]\!]_{\Delta,\Gamma,\Delta'}$ and thus by definition of $[\![\Box B]\!]_{\Delta,\Gamma}$ $M \in [\![\Box B]\!]_{\Delta,\Gamma}$.
2. Let Δ'' a stack such that $(\Delta, \Delta'), \Delta''$ is valid. $(\Delta, \Delta'), \Delta'' = \Delta, (\Delta', \Delta'')$. Thus by definition of $[\![\Box A]\!]_\Delta$, $\downarrow M \in [\![A]\!]_{\Delta,\Delta',\Delta''}$ and by definition of $[\![\Box A]\!]_{\Delta,\Delta'}$, $M \in [\![\Box A]\!]_{\Delta,\Delta'}$. If Γ is a context such that $\Delta; \Gamma$ is valid, we note that $\Delta; \Gamma = \Delta, (.; .; \Gamma)$. Thus $M \in [\![\Box A]\!]_{\Delta;\Gamma}$.

□

Roughly speaking, the interpretation of a context stack Φ in Δ is the set of the substitutions that replace variables $x : A$ declared in Φ by terms belonging to $[\![A]\!]_\Delta$. The exact definition is a bit more complex because we deal with context stacks, instead of contexts:

Definition 6.4 (Interpretation of context stacks) *The* interpretation *of a context stack Φ in Δ, $[\![\Phi]\!]_\Delta$, is the set of substitutions from Φ to Δ defined by induction on the structure of Φ:*

- *$[\![.]\!]_\Delta =_{def} \{weak^\Delta_.\}$.*

- $[\![.;.]\!]_\Delta =_{def} \{weak^{\Delta}_{.;.}\}$.
- $[\![.;\Gamma, x : A]\!]_\Delta =_{def} \{\rho[x := M] \mid \rho \in [\![.;\Gamma]\!]_\Delta \ \&\ M \in [\![A]\!]_\Delta\}$.
- $[\![\Phi;\Gamma]\!]_\Delta =_{def} \{\delta;\rho \mid \exists n \in I\!N\ .\delta \in [\![\Phi]\!]_{\Delta^n} \ \&\ \rho \in [\![.;\Gamma]\!]_\Delta\}$.

As a consequence of the last item, $[\![\Phi;.]\!]_{\Delta;.} = \{\delta;\rho \mid \exists n \in \mathbb{N}\ .\delta \in [\![\Phi]\!]_{(\Delta;.)^n}$ and $\rho \in [\![.;.]\!]_{\Delta;.}\} = \{\delta; weak^{\Delta;.}_{.;.} \mid \delta \in [\![\Phi]\!]_{(\Delta;.)^n}\}$.

The difficulty of the definition lies in the expression of $[\![\Phi;\Gamma]\!]_\Delta$. The simpler choice $[\![\Phi;\Gamma]\!]_\Delta =_{def} \{\delta;\rho \mid \delta \in [\![\Phi]\!]_{\Delta^1} \ \&\ \rho \in [\![.;\Gamma]\!]_\Delta\}$ would not have been flexible enough to meet our needs.

Example Let us assume that $\Phi = (.; x : A; y : B)$ and $\Delta = (.; f : B \to A, z : B; .; u : B)$, then the substitution δ from Φ to Δ defined by $\delta(x) = (f\ z)$ and $\delta(y) = u$ belongs to $[\![\Phi]\!]_\Delta$ if $u \in [\![B]\!]_\Delta$ and $(f\ z) \in [\![A]\!]_{.;f:B\to A,z:B}$.

Lemma 6.5 *If $\rho \in [\![\Phi]\!]_\Delta$ and if Γ is a context such that Δ, Γ is valid then $\rho \in [\![\Phi]\!]_{\Delta,\Gamma}$.*

Proof By induction on the proof of $\rho \in [\![\Phi]\!]_\Delta$ using Lemma 6.3.
□

Definition 6.6 (Saturated set) *A set $\mathcal{S}$ of semantic objects for Δ and A is a* saturated set *for Δ and A if:*

(S1) *If M is a base term and a semantic object for Δ and A, then $M \in \mathcal{S}$.*

(S2) *If $N \in \mathcal{S}$ and $\Delta \vdash M \to_{wh} N : A$ then $M \in \mathcal{S}$.*

Lemma 6.7 ($[\![A]\!]_\Delta$ is a saturated set)
$[\![A]\!]_\Delta$ is a saturated set for any valid stack Δ and type A.

Proof By induction on the structure of A. We consider the cases $A = B \to C$ and $A = \Box B$.

- $A = B \to C$

  (S1) Let M be a base term and a semantic object for Δ and $B \to C$, Γ a context such that Δ, Γ is valid and N an element of $[\![B]\!]_{\Delta,\Gamma}$. We have to prove that $M \in [\![B \to C]\!]_\Delta$, i.e. $(M\ N) \in [\![C]\!]_{\Delta,\Gamma}$. Actually we will show that $(M\ N)$ is a base term and a semantic object for Δ, Γ and C and applying the induction hypothesis will give us the result.

  - $(M\ N)$ is a base term because M is a base term.
  - Thanks to the definition of a semantic object and Thinning lemma (5.13), $\exists P.\Delta, \Gamma \vdash M \to_{nf} P : B \to C$. Because $N \in [\![B]\!]_{\Delta,\Gamma}$, we also have $\exists Q.\Delta, \Gamma \vdash N \to_{nf} Q : B$. Since any base term is also weak head normal, we have $\exists R.\Delta, \Gamma \vdash (M\ N) \to_{nf} R : C$.

(S2) Let us assume that $N \in [\![B \to C]\!]_\Delta$ and that $\Delta \vdash M \to_{wh} N : B \to C$. We want to prove that $M \in [\![B \to C]\!]_\Delta$. Let Γ be a context such that Δ, Γ is valid and $P \in [\![B]\!]_{\Delta,\Gamma}$, we will prove that $(M\ P) \in [\![C]\!]_{\Delta,\Gamma}$. Thanks to Thinning lemma (5.13), we have $\Delta, \Gamma \vdash M \to_{wh} N : B \to C$. Since $P \in [\![B]\!]_{\Delta,\Gamma}$ we also know that $\exists Q.\Delta, \Gamma \vdash P \to_{nf} Q : B$. Applying the rule $(W\ App)$ we find that $\Delta, \Gamma \vdash (M\ P) \to_{wh} (N\ P) : C$. Finally, $(N\ P) \in [\![C]\!]_{\Delta,\Gamma}$ and by induction hypothesis $(M\ P) \in [\![C]\!]_{\Delta,\Gamma}$.

- $A = \Box B$

  (S1) Let M be a base term and a semantic object for Δ and $\Box B$, and Δ' a stack such that Δ, Δ' is valid. $\downarrow M$ is also a base term and $\exists N.\Delta, \Delta' \vdash \downarrow M \to_{nf} \downarrow N : B$ (by definition of a semantic object, rules $(S\ Pop)$ and $(S\downarrow)$). By inductive hypothesis $\downarrow M \in [\![B]\!]_{\Delta,\Delta'}$. Therefore $M \in [\![\Box B]\!]_{\Delta,\Delta'}$.

  (S2) Let us assume that $N \in [\![\Box B]\!]_\Delta$ and that $\Delta \vdash M \to_{wh} N : \Box B$. We want to prove that $M \in [\![\Box B]\!]_\Delta$. Let Δ' be a stack such that Δ, Δ' is valid. After applying rules $(W\ Pop)$ and $(W\downarrow)$, we obtain $\Delta, \Delta' \vdash \downarrow M \to_{wh} \downarrow N : B$. By inductive hypothesis $\downarrow M \in [\![B]\!]_{\Delta,\Delta'}$ and therefore $M \in [\![\Box B]\!]_\Delta$.

□

Now we prove the key lemma of this section: soundness of the typed operational semantics for our calculus.

Lemma 6.8 (Soundness) *If* $\Phi \vdash M : A$ *and* $\rho \in [\![\Phi]\!]_\Delta$ *then* $(\overline{\rho}\ M) \in [\![A]\!]_\Delta$.

Proof By induction on derivations of $\Delta \vdash M : A$. We show here some cases.

- (λ) We have $\Phi \vdash \lambda x : A.M : A \to B$ by rule λ. Let $\rho \in [\![\Phi]\!]_\Delta$, we have to prove that $(\overline{\rho}\ \lambda x : A.M) \in [\![A \to B]\!]_\Delta$. By inductive hypothesis, we know that $(\overline{\rho'}\ M) \in [\![B]\!]_{\Delta'}$ for any $\rho' \in [\![\Phi, x : A]\!]_{\Delta'}$.

  We first need to show that there exists a Q such that $\Delta \vdash (\overline{\rho}\ \lambda x : A.M) \to_{nf} Q : A \to B$. Let y be fresh in Δ. Then $\rho \in [\![\Phi]\!]_{\Delta,y:A}$, and $y \in [\![A]\!]_{\Delta,y:A}$ by Lemma 6.7 and (S1), so $\rho[x := y] \in [\![\Phi, x : A]\!]_{\Delta,y:A}$ by definition of $[\![\Phi, x : A]\!]_{\Delta,y:A}$. Therefore we know that $(\overline{\rho[x := y]}\ M) \in [\![B]\!]_{\Delta,y:A}$ which implies that $\Delta, y : A \vdash (\overline{\rho[x := y]}\ M) \to_{nf} P : B$ for some P. Hence if $P \equiv (P'\ y)$ with $y \notin FV(P')$ then $\Delta \vdash (\overline{\rho}\ \lambda x : A.M) \to_{nf} P' : A \to B$ by $(S\eta')$, and otherwise $\Delta \vdash (\overline{\rho}\ \lambda x : A.M) \equiv (\overline{\rho}\ \lambda y : A.M[y/x]) \to_{nf} \lambda y : A.P : A \to B$ by $(S\lambda)$. Let Q be the normal form in either case.

  Then we need to show that $(\lambda x : A.M\ N) \in [\![B]\!]_{\Delta,\Gamma}$ for any context Γ such that Δ, Γ is valid and any N belonging to $[\![A]\!]_{\Delta,\Gamma}$. We can show by reasoning like above that $\rho[x := N] \in [\![\Phi, x : A]\!]_{\Delta,\Gamma}$. This means that $(\overline{\rho[x := N]}\ M) \in [\![B]\!]_{\Delta,\Gamma}$. So $\Delta, \Gamma \vdash (\overline{\rho[x := N]}\ M) \to_{nf} R : B$ for some R. Furthermore we know that $\Delta, \Gamma \vdash (\overline{\rho}\ \lambda x : A.M) \to_{nf} Q : A \to B$ by Thinning lemma, and also $\Delta, \Gamma \vdash N \to_{nf} N' : A$ for some N' by definition of semantic objects, so $\Delta, \Gamma \vdash ((\overline{\rho}\ \lambda x : A.M)\ N) \equiv (\lambda z : A.(\overline{\rho[x := z]}\ M)\ N) \to_{wh} (\overline{\rho[x := z]}\ M)[N/z] \equiv$

$(\overline{\rho[x := N]}\ M) : B$ by $(W\beta)$ for any z fresh in Δ, Γ. Hence $((\overline{\rho}\ \lambda x : A.M)\ N) \in [\![B]\!]_{\Delta,\Gamma}$ by (S2). So $(\overline{\rho}\ \lambda x : A.M) \in [\![A \to B]\!]_\Delta$ by definition.

- $(\downarrow)$ straightforward.
- $(\uparrow)$ We have $\Phi \vdash \uparrow M : \Box A$ by rule $(\uparrow)$. Suppose $\rho \in [\![\Phi]\!]_\Delta$, we have to prove that $(\overline{\rho}\ \ \uparrow M) \in [\![\Box A]\!]_\Delta$. We first need to show that there exists a Q such that $\Delta \vdash (\overline{\rho}\ \uparrow M) \to_{nf} Q : \Box A$. By induction hypothesis, we know that there exists a P such that $\Delta; . \vdash (\overline{\rho'}\ M) \to_{nf} P : A$, for any $\rho' \in [\![\Phi; .]\!]_{\Delta;.}$. Hence if $P \neq \downarrow N$, then $\Delta \vdash (\overline{\rho'}\ \uparrow M) \to_{nf} \uparrow P : \Box A$ by rule $(S\uparrow)$, and otherwise $P \equiv \downarrow N$ and $\Delta \vdash (\overline{\rho'}\ \uparrow M) \to_{nf} N : \Box A$ by rule $(S\eta_\Box)$. Let Q be the normal form in either case. We take $\rho' = \rho; weak^{\Delta;\cdot}_{\cdot;\cdot}$.

  We then need to show that $(\overline{\rho}\ \ \downarrow\uparrow M) \in [\![A]\!]_{\Delta,\Delta'}$ for any stack Δ' such that Δ, Δ' is valid. By induction hypothesis, we know that $(\overline{\rho'}\ M) \in [\![A]\!]_{\Delta,\Delta'}$ for any $\rho' \in [\![\Phi; .]\!]_{\Delta,\Delta'}$. So by definition of semantic objects, we know that $\Delta, \Delta' \vdash (\overline{\rho'}\ M) \to_{nf} M' : A$ for some M'. So $\Delta, \Delta' \vdash (\overline{\rho'}\ \ \downarrow\uparrow M) = \downarrow\uparrow (\overline{\rho'}\ M) \to_{wh} (\overline{\rho'}\ M) : A$ by rule $(W\beta_\Box)$. So, by $(S2)$, $(\overline{\rho'}\ \downarrow\uparrow M) = \overline{\rho}(M) \in [\![A]\!]_{\Delta,\Delta'}$. To conclude, we take $\rho' = \rho; weak^{\Delta,\Delta'}_{\cdot;\cdot}$. So $(\overline{\rho}\ \uparrow M) \in [\![\Box A]\!]_\Delta$ by definition.
- (Pop) Let us assume that ρ belongs to $[\![\Phi; \Gamma]\!]_\Delta$. We write ρ as $\rho_1; \rho_2$, where $\rho_1 \in [\![\Phi]\!]_{\Delta^n}$ and $\rho_2 \in [\![.; \Gamma]\!]_\Delta$. Since $\mathrm{FV}(M) \subset \mathrm{dom}(\Phi)$ we have $\overline{\rho}(M) = \overline{\rho_1}(M)$. By induction hypothesis $\overline{\rho_1}(M) \in [\![\Box A]\!]_{\Delta^n}$ and therefore $\overline{\rho}(M) \in [\![\Box A]\!]_\Delta$.

□

Since the identity substitution from Δ to Δ belongs to $[\![\Delta]\!]_\Delta$, we are able to link typing judgments with $\to_{nf}$ judgments.

Lemma 6.9 *If Δ is a context stack then $id_\Delta \in [\![\Delta]\!]_\Delta$.*

Proof Straightforward, by induction on the structure of Δ.
□

Corollary 6.10 *If $\Delta \vdash M : A$ then there is a term P such that $\Delta \vdash M \to_{nf} P : A$.*

Proof By Soundness, Lemma 6.9 and the definition of $[\![A]\!]_\Delta$.
□

Lemma 6.11 (Soundness for $\Delta \vdash M = N : A$) *If $\Delta \vdash M = N : A$ then there is a P such that $\Delta \vdash M \to_{nf} P : A$ and $\Delta \vdash N \to_{nf} P : A$.*

Proof By induction on derivations that $\Delta \vdash M = N : A$.
□

Hence the three expected results of Strong Normalization, Subject Reduction and Church-Rosser:

Corollary 6.12 (Strong Normalization) *If $\Delta \vdash M : A$ then M is strongly normalizing.*

Corollary 6.13 (Subject Reduction) *If $\Delta \vdash M : A$ and $M \hookrightarrow N$ then $\Delta \vdash N : A$.*

Corollary 6.14 (Church-Rosser) *If $\Delta \vdash M = N : A$ then there is a P such that $M \hookrightarrow_* P$ and $N \hookrightarrow_* P$.*

7 Conclusion

We have presented in this paper the proofs of Strong Normalization, Subject Reduction, and Church-Rosser theorems for a modal λ-calculus $S4$ [PW95]. The proof followed the 'Typed Operational Semantics' method introduced by Healfdene Goguen [Gog94, Gog95] for the simply-typed λ-calculus. We have succeeded in adapting his method although our typing rules are not syntax-driven and we had to deal with context stacks instead of simple contexts. In the course of the proof, we have extracted an interesting notion of typed substitution for our modal system. Moreover some definitions we have used are different, maybe clearer and more general than the original ones. For instance in the definitions of the interpretations of types, we have used weakenings instead of renamings.

It has been shown recently how modality IS4 can be used to built a simple type system which allows primitive recursion on terms of a higher order abstract syntax ([DPS97]). Using the developments presented in this paper (in particular the definitions and lemmas of Section 6), we have been able to propose a variant of this type system, with simpler reduction rules and a much shorter proof ([Lel97]). We hope that the results presented there will serve as a basis for the proofs for an extension of the type system to a richer calculus, including polymorphic and dependent types.

Acknowledgements I gratefully acknowledge discussions with Healfdene Goguen and André Hirschowitz. I am endebted to my advisor, Joëlle Despeyroux for her numerous suggestions on drafts of this work. Thanks are also due to Paul Taylor for his TeX macros for drawing category-theoretic diagrams.

References

[BdP96] Gavin Bierman and Valeria de Paiva. Intuitionistic necessity revisited. In *Technical Report CSRP-96-10*, School of Computer Science, University of Birmingham, 1996.

[Che90] Brian F. Chellas. *Modal logic : an introduction.* Cambridge University Press, 1990.

[DP96] Rowan Davies and Frank Pfenning. A modal analysis of staged computation. In Jr. Guy Steele, editor, *Proceedings of the 23rd Annual Symposium on Principles of Programming Languages*, pages 258–270, St. Petersburg Beach, Florida, January 1996. ACM Press.

[DPS97] Joëlle Despeyroux, Frank Pfenning, and Carsten Schürmann. Primitive Recursion for Higher-Order Abstract Syntax. In J.R. Hindley and P. de Groote, editors, *Int. Conf. on Typed lambda calculi and applications - TLCA '97*, pages 147–163, Nancy, France, April 1997. Springer-Verlag LNCS 1210.

[Gog94] Healfdene Goguen. *A Typed Operational Semantics for Type Theory.* PhD thesis, University of Edinburgh, August 1994.

[Gog95] Healfdene Goguen. Typed operational semantics. In *Proceedings of the International Conference on Typed Lambda Calculi and Applications*, volume 902 of *Lecture Notes in Computer Science*, pages 186–200. Springer–Verlag, 1995.

[Lel97] Pierre Leleu. A Modal Lambda Calculus with Iteration and Case Constructs. Technical Report RR-3322, INRIA, France, 1997.

[PW95] Frank Pfenning and Hao-Chi Wong. On a modal λ-calculus for S4. In S. Brookes and M. Main, editors, *Proceedings of the Eleventh Conference on Mathematical Foundations of Programming Sematics*, New Orleans, Louisiana, March 1995. To appear in *Electronic Notes in Theoretical Computer Science*, Volume 1, Elsevier.

8 Annex A: variant with Pop rules

In Section 2, the typing rules are not syntax-driven because of rule (Pop) which keeps the term unchanged. In this annex we present a variant of our system where we add to our syntax an operator 'Pop' which marks the application of rule (Pop).

8.1 The type system

The sets of types, terms and contexts are generated by the same syntaxes as before, with an additional definition for *Pop* terms, as follows:

$$\begin{array}{lll} \mathit{Types} & A & ::= c \mid A \to A' \mid \Box A \\ \mathit{Terms} & t & ::= x \mid \lambda x : A.t' \mid (t\ t') \mid \uparrow t \mid \downarrow t \mid \mathit{Pop}(t) \\ \mathit{Contexts} & \Gamma & ::= . \mid \Gamma, x : t \\ \mathit{Context\ stacks} & \Delta & ::= . \mid \Delta;\Gamma \end{array}$$

The complete type system is the following one, where only the (*Pop*) rule has been modified:

$$\text{(Var)}\ \frac{x : A \in \Gamma}{\Delta;\Gamma \vdash x : A}\ \Delta;\Gamma \text{ valid}$$

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

$$(\uparrow)\ \frac{\Delta;. \vdash M : A}{\Delta \vdash \uparrow M : \Box A} \qquad (\downarrow)\ \frac{\Delta \vdash M : \Box A}{\Delta \vdash \downarrow M : A} \qquad (Pop')\ \frac{\Delta \vdash M : \Box A}{\Delta;\Gamma \vdash Pop(M) : \Box A}\ \Delta;\Gamma \text{ valid}$$

Note Terms of the form $((\uparrow M)\ N)$, $\downarrow \lambda x : A.M$, $Pop(\lambda x : A.M)$ or $(Pop(M)\ N)$ are not well-typed in this system.

8.1.1 Lemmas

We do not need here the modal weakening lemmas of Section 2.1 (without the *Pop* terms), as the type system is now trivially syntax-driven, i.e. there is only one applicable rule at each stage of the construction of a proof. Nevertheless, we can prove the following modal weakening lemmas.

Lemma 8.1 (Weakening)

$$(Weak)\ \frac{.; D_1; \cdots; D_i; D_{i+1}; \cdots; D_n \vdash M : A}{.; D_1; \cdots; D_i; D; D_{i+1}; \cdots; D_n \vdash POP(M, n-(i+1)) : A}\ \textit{for } 1 \leq i < n$$

where D is such that $.; D_1; \cdots; D_i; D; D_{i+1}; \cdots; D_n$ *is valid and POP(M,k) is defined by induction on M:*

- $POP(x, n) = x$
- $POP(Pop(M), n) = \begin{cases} Pop(POP(M, n-1)) & \textit{if } n > 1 \\ Pop(Pop(M)) & \textit{if } n = 0 \end{cases}$
- $POP(\uparrow M, n) = \uparrow POP(M, n+1)$
- $POP(\downarrow M, n) = \downarrow POP(M, n)$
- $POP((MN), n) = (POP(M, n)\ POP(N, n))$
- $POP(\lambda x : A.M, n) = \lambda x : A.POP(M, n)$

Note Other terms built from M with an extra Pop could verify the Modal Weakening rule. They are not given by this algorithm.

For instance, if $.; x : \Box\Box A; \Gamma \vdash \downarrow Pop(x) : \Box A$, then $POP(\downarrow Pop(x), 0) = \downarrow Pop(Pop(x))$ and $.; x : \Box\Box A; \Phi; \Gamma \vdash \downarrow Pop(Pop(x)) : \Box A$. But we also have $.; x : \Box\Box A; \Phi; \Gamma \vdash Pop(\downarrow Pop(x)) : \Box A$ and $.; x : \Box\Box A; \Phi; \Gamma \vdash Pop(Pop(\downarrow x)) : \Box A$.

Lemma 8.2 (Strengthening)

$$(Strengthening)\ \frac{.; D_1; \cdots; \Gamma, x : B, \Gamma'; \cdots; D_n \vdash M : A}{.; D_1; \cdots; \Gamma, \Gamma'; \cdots; D_n \vdash M : A}\ x \notin FV(M)$$

The fusion lemma is more complicated than before. It uses the $UNPOP$ function which appears also later in the reduction rules.

Lemma 8.3 (Fusion)

$$(Fusion)\ \frac{.; D_1; \cdots; D_i; D_{i+1}; \cdots; D_n \vdash M : A}{.; D_1; \cdots; (D_i, D_{i+1}); \cdots; D_n \vdash UNPOP(M, n-(i+1)) : A}\ \textit{for } 1 \leq i < n$$

where $UNPOP(M,n)$ *(*$n \in \mathbb{N}$*) is defined by induction on* M *:*

- $UNPOP(x,n) = x$
- $UNPOP(Pop(M),n) = \begin{cases} Pop(UNPOP(M,n-1)) & \textit{if } n > 1 \\ M & \textit{if } n = 0 \end{cases}$
- $UNPOP(\uparrow M, n) = \uparrow UNPOP(M, n+1)$
- $UNPOP(\downarrow M, n) = \downarrow UNPOP(M, n)$
- $UNPOP((MN), n) = (UNPOP(M,n)\ UNPOP(N,n))$
- $UNPOP(\lambda x : A.M, n) = \lambda x : A.UNPOP(M,n)$

8.1.2 Substitution

Substitution is defined like in Section 2.2 with the following additional rule:

$$(Pop(M))[N/x] \equiv Pop(M[N/x])$$

8.1.3 Inversion lemmas

The inversion (generation) lemmas are obvious:

1. $(\Delta; \Gamma \vdash x : A) \Rightarrow x : A \in \Gamma$.
2. $(\Delta \vdash \lambda x : A.M\ :\ A \to B) \Rightarrow (\Delta, x : A \vdash M : B)$
3. $(\Delta \vdash \uparrow M : \Box A) \Rightarrow (\Delta; . \vdash M : A)$
4. $(\Delta \vdash \downarrow N : A) \Rightarrow (\Delta \vdash N : \Box A)$
5. $(\Delta \vdash (M\ N) : B) \Rightarrow (\Delta \vdash M : A \to B\ \&\ \Delta \vdash N : A)$
6. $(\Delta; \Gamma \vdash Pop(M) : \Box A) \Rightarrow (\Delta \vdash M : \Box A)$

Proof By induction on the derivation of the hypothesis.
□

8.1.4 Equality

The rules are the same as before, except for rules $(EqPop)$, $(Eq\beta_{\Box})$ and $(Eq\eta_{\Box})$, that we give here.

$$(EqPop') \ \frac{\Delta \vdash M = N : \Box A}{\Delta;\Gamma \vdash Pop(M) = Pop(N) : \Box A} \ \Delta;\Gamma \text{ valid}$$

$$(Eq\beta'_{\Box}) \ \frac{\Delta;. \vdash M : A}{\Delta;\Gamma_1;\cdots;\Gamma_n \vdash \downarrow Pop^n(\uparrow M) = UNPOP(Pop^n(M),0) : A} \ \Delta;\Gamma_1;\cdots;\Gamma_n \text{ valid}$$

$$(Eq\eta'_{\Box}) \ \frac{\Delta;. \vdash M : \Box A}{\Delta;\Gamma_1;\cdots;\Gamma_n \vdash \uparrow Pop^n(\downarrow M) = UNPOP(Pop^n(M),0) : \Box A} \ \Delta;\Gamma_1;\cdots;\Gamma_n \text{ valid}$$

8.2 A typed operational system

The definitions for normal forms and weak head reduction are the same as before, except for the (Pop), $(S\eta_{\Box})$ and $(W\beta_{\Box})$ rules. The side condition in $(S\uparrow)$ is also different.

Normal forms:

$$(SVar) \ \frac{x : A \in \Gamma}{\Delta;\Gamma \vdash x \rightarrow_{nf} x : A} \ \Delta;\Gamma \text{ valid}$$

$$(S\lambda) \ \frac{\Delta, x : A \vdash M \rightarrow_{nf} P : B}{\Delta \vdash \lambda x : A.M \rightarrow_{nf} \lambda x : A.P : A \rightarrow B} \text{ if } P \equiv (Q\ x) \ \Rightarrow \ x \in FV(Q)$$

$$(S\eta) \ \frac{\Delta, x : A \vdash M \rightarrow_{nf} (P\ x) : B \qquad \Delta \vdash P \rightarrow_{nf} Q : A \rightarrow B}{\Delta \vdash \lambda x : A.M \rightarrow_{nf} Q \ : \ A \rightarrow B}$$

$$(SApp) \ \frac{\Delta \vdash M \rightarrow_{nf} P : A \rightarrow B \qquad \Delta \vdash N \rightarrow_{nf} Q : A}{\Delta \vdash (M\ N) \rightarrow_{nf} (P\ Q) : B} \text{ if } (M\ N) \text{ is whn}$$

$$(S\downarrow) \ \frac{\Delta \vdash M \rightarrow_{nf} N : \Box A}{\Delta \vdash \downarrow M \rightarrow_{nf} \downarrow N : A} \text{ if } \downarrow M \text{ is whn}$$

$$(S\uparrow) \ \frac{\Delta;. \vdash M \rightarrow_{nf} N : A}{\Delta \vdash \uparrow M \rightarrow_{nf} \uparrow N : \Box A} \text{if } N \neq Pop^n(\downarrow P)$$

$$(S\eta'_{\Box}) \ \frac{\Delta;. \vdash M \rightarrow_{nf} Pop^n(\downarrow N) : A}{\Delta \vdash \uparrow M \rightarrow_{nf} UNPOP(Pop^n(N),0) : \Box A}$$

$$(SPop)\ \frac{\Delta \vdash M \rightarrow_{nf} N : \Box A}{\Delta;\Gamma \vdash Pop(M) \rightarrow_{nf} Pop(N) : \Box A}\ \Delta;\Gamma \text{ valid}$$

$$(SW)\ \frac{\Delta \vdash M \rightarrow_{wh} N : A \qquad \Delta \vdash N \rightarrow_{nf} P : A}{\Delta \vdash M \rightarrow_{nf} P : A}$$

Weak head reduction:

$$(W\beta)\ \frac{\Delta \vdash \lambda x : A.M \rightarrow_{nf} M' : A \rightarrow B \qquad \Delta \vdash N \rightarrow_{nf} N' : A}{\Delta \vdash (\lambda x : A.M\ \ N) \rightarrow_{wh} M[N/x]\ :\ B}$$

$$(WApp)\ \frac{\Delta \vdash M \rightarrow_{wh} P : A \rightarrow B \qquad \Delta \vdash N \rightarrow_{nf} N' : A}{\Delta \vdash (M\ N) \rightarrow_{wh} (P\ N) : B}$$

$$(W\beta'_{\Box})\ \frac{\Delta;. \vdash M \rightarrow_{nf} M' : A}{\Delta;\Gamma_1;\cdots;\Gamma_n \vdash \downarrow Pop^n(\uparrow M) \rightarrow_{wh} UNPOP(Pop^n(M),0) : A}\ \Delta;\Gamma_1;\cdots;\Gamma_n \text{ valid}$$

$$(W\downarrow)\ \frac{\Delta \vdash M \rightarrow_{wh} N : \Box A}{\Delta \vdash \downarrow M \rightarrow_{wh} \downarrow N : A}$$

$$(WPop')\ \frac{\Delta \vdash M \rightarrow_{wh} N : \Box A}{\Delta;\Gamma \vdash Pop(M) \rightarrow_{wh} Pop(N) : \Box A}\ \Delta;\Gamma \text{ valid}$$

8.3 Metatheory

$UNPOP$ now appears in the $(\beta'_{\Box})$ and $(\eta'_{\Box})$ reduction rules.

Definition 8.4 (Untyped reduction)

(β) $(\lambda x : A.M\ N)\ \ \beta\ \ M[N/x]$
(η) $\lambda x : A.(M\ x)\ \ \eta\ \ M\ \ if\ x \notin FV(M)$
$(\beta'_{\Box})$ $\downarrow Pop^n(\uparrow M)\ \ \beta'_{\Box}\ \ UNPOP(Pop^n(M),0)$
$(\eta'_{\Box})$ $\uparrow Pop^n(\downarrow M)\ \ \eta'_{\Box}\ \ UNPOP(Pop^n(M),0)$

The following definition is the same as before, except for the $\uparrow$, $\downarrow$ and Pop items, which remain fairly simple.

Lemma 8.5 (Forms of normal terms)
The normal forms can be characterized by induction:

- *Variables are normal,*
- $\lambda x : A.M$ *is normal if* M *is normal and not of the form* $(N\ x)$ *with* $x \notin FV(N)$*,*

- $(M\ N)$ *is normal if* M *and* N *are normal and* M *is not of the form* $\lambda x : A.P$,
- $\uparrow M$ *is normal if* M *is normal, and not of the form* $Pop^n(\downarrow M)$ $(n \in \mathbb{N})$.
- $\downarrow M$ *is normal if* M *is normal, and not of the form* $Pop^n(\uparrow M)$ $(n \in \mathbb{N})$.
- $Pop(M)$ *is normal iff* M *is normal.*

A partial attempt to prove the subject reduction, Church-Rosser and strong normalization properties has shown that this seems feasible with some changes. For instance we could adopt a simpler definition of substitutions and the inversion lemmas would be straightforward. Nevertheless, since the reduction rules are more complex, other results would be more difficult to prove (for instance the subject reduction property).

9 Annex B: technical development

The results proved in this annex are rather technical. The key result (Lemma 9.10) is a lemma of confluence between weak head reduction and arbitrary reduction. It is used in the proofs of Lemmas 5.25 and 5.26.

Definition 9.1 (weak head reducible)
We call weak head reducible *(whr) terms the terms that are not weak head normal.*

Lemma 9.2 *The weak head reducible terms (whr) are of the form:*

$$whr = (\lambda x : A.P\ N) \mid \ \downarrow\uparrow M \mid (whr\ N) \mid \ \downarrow whr.$$

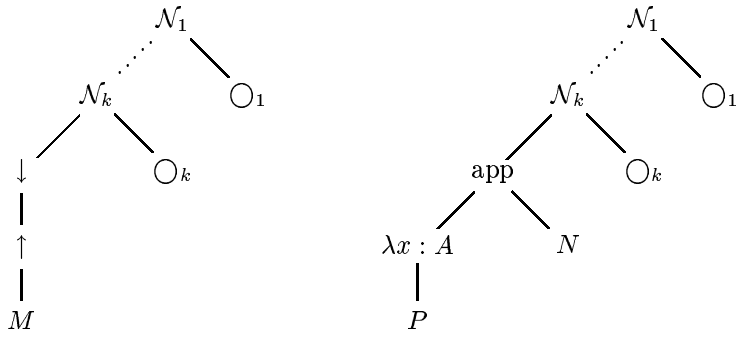
Definition 9.3 (elements) *We recursively define the function 'elem' which maps a term whr to a list of terms (its* 'elements'*) by the following rules:*

- $elem(\lambda x : A.P\ N) = (N, \lambda x : A.P)$
- $elem(\downarrow\uparrow M) = \uparrow M$
- $elem(whr\ N) = (N, elem(whr))$
- $elem(\downarrow whr) = elem(whr)$

We also define the function '*head*' which maps a *whr* term to a term:

Definition 9.4 (head)
If M *is whr and* $elem(M) = (M_1, \ldots, M_k)$, *we define* $head(M) := M_k$.

To summarize, the possible forms of weak head reducible terms are the following ones:



where $\bigcirc_1, \ldots, \bigcirc_k$ are subtrees, all the operators $\mathcal{N}_1, \ldots, \mathcal{N}_k$ are either app (i.e. application) or $\downarrow$ (in which case the corresponding subtree $\bigcirc_j$ is empty).

For the first term, the '*elem*' function returns the list $(\bigcirc_1, \ldots, \bigcirc_k, \uparrow M)$ and the '*head*' function returns $\uparrow M$. For the second term, the '*elem*' function returns the list $(\bigcirc_1, \ldots, \bigcirc_k, N, \lambda x : A.P)$ and the '*head*' function returns $\lambda x : A.P$.

We call *structure* of a whr term the list $(\mathcal{N}_1, \ldots, \mathcal{N}_k, \downarrow)$ (or $(\mathcal{N}_1, \ldots, \mathcal{N}_k, \text{app})$). A whr term is entirely defined by its structure and the list of its elements.

Lemma 9.5

$$(\Delta \vdash M \rightarrow_{wh} P : B) \Rightarrow (M \textit{ is whr})$$

Proof Easy. By induction on the proof of the hypothesis.
□

Lemma 9.6

$$(\Delta \vdash M \rightarrow_{wh} P : B) \Rightarrow \left(\begin{array}{l} \forall N \in \textit{elem}(M), \\ \exists n \in I\!N\,, N' \textit{ and } C \textit{ such that } \Delta^n \vdash N \rightarrow_{nf} N' : C. \end{array} \right)$$

Proof Easy. By induction on the proof of the hypothesis.
□

The following lemma allows us to solve some difficulties associated with truncated stacks. It says that if a term is well-typed in a stack Δ^n of type A and has a $\rightarrow_{nf}$ evaluation in another stack Δ^m of type B then it has the same $\rightarrow_{nf}$ evaluation in Δ^n and $A = B$.

Lemma 9.7 *If* $\Delta^n \vdash M : A$ $(n \in I\!N)$ *then*

- *if* $\Delta^m \vdash M \rightarrow_{nf} P : B$ $(m \in I\!N)$ *then* $\Delta^n \vdash M \rightarrow_{nf} P : A$
- *if* $\Delta^p \vdash M \rightarrow_{wh} N : C$ $(p \in I\!N)$ *then* $\Delta^n \vdash M \rightarrow_{wh} N : A$

Proof By induction on the proofs of the hypotheses "$\Delta^m \vdash M \rightarrow_{nf} P : B$" and "$\Delta^p \vdash M \rightarrow_{wh} N : C$", using the inversion lemmas on the typing hypothesis. We show here two significant cases:

- (SVar) $\Delta^m \vdash x \rightarrow_{nf} x : B$ because $x : B$ belongs to the local context of Δ^m. By the inversion lemmas, it is clear that $A = B$. If the typing judgment $\Delta^n \vdash x : A$ comes from rule (Pop), then $n < m$ and A is of the form $\Box C$. Thus by rule (SPop), $\Delta^n \vdash x \rightarrow_{nf} x : A$. Otherwise, if the typing judgment $\Delta^n \vdash x : A$ comes from rule (Var) then $n = m$.
- (SApp) $\dfrac{\Delta^m \vdash M \rightarrow_{nf} P : C \rightarrow B \qquad \Delta^m \vdash N \rightarrow_{nf} Q : C}{\Delta^m \vdash (M\ N) \rightarrow_{nf} (P\ Q) : B}$ if $(M\ N)$ is whn

  By hypothesis, we have $\Delta^n \vdash (M\ N) : A$. The inversion lemmas tell us that $\Delta^{n+k} \vdash M : D \rightarrow A$ and $\Delta^{n+k} \vdash N : D$, where $k = 0$ if A is not of the form $\Box E$. By induction hypothesis, $\Delta^{n+k} \vdash M \rightarrow_{nf} P : D \rightarrow A$ and $\Delta^{n+k} \vdash N \rightarrow_{nf} Q : D$. Thus, by (SApp), $\Delta^{n+k} \vdash (M\ N) \rightarrow_{nf} (P\ Q) : A$. If A is of the form $\Box E$, we apply (SPop) and we are done.

□

Definition 9.8 *For M a whr term such that each N belonging to $elem(M)$ is strongly normalizing, we define $\mu(M)$ as $\mu(M) = \sum_{N \in elem(M)} \nu(N)$, where $\nu(N)$ is the maximum number of reduction steps starting from N.*

Lemma 9.9 *If $\Delta \vdash M : A$, M whr and $M \hookrightarrow_* N$ (without weak head reduction steps) then N is whr, has the same structure as M and its elements are obtained from those of M by reductions. Moreover if all the elements of M are strongly normalizing, then $\mu(M)$ and $\mu(N)$ are defined and $\mu(N) < \mu(M)$.*

Proof By induction on the proof of "M whr". We show here two interesting cases:

- If $M \equiv{\downarrow\uparrow} M_1$, we necessarily have $N \equiv{\downarrow\uparrow} N_1$ with $M_1 \hookrightarrow_* N_1$.
- If $M \equiv (P_1\ Q)$ with P_1 whr, we have $N \equiv (P_1'\ Q')$ where $P_1 \hookrightarrow_* P_1'$ and $Q \hookrightarrow_* Q'$. By induction hypothesis, P_1' is whr, its structure is the same as the one of P_1 and its elements are obtained from those of P_1 by reductions. Thus N is whr, has the same structure as M and its elements are obtained from those of M by reductions.

  Moreover if all the elements of M are strongly normalizing, all the elements of P_1 are also strongly normalizing and by induction hypothesis $\mu(P_1') < \mu(P_1)$. Thus $\mu(M) = \mu(P_1) + \nu(Q) < \mu(P_1') + \nu(Q') = \mu(N)$.

□

Lemma 9.10 *If $\Delta \vdash M : A$, $M \hookrightarrow_{wh} N$, $M \hookrightarrow_* P$ (without weak head reduction steps) then there is a term Q such that $P \hookrightarrow_{wh} Q$ and $N \hookrightarrow_* Q$.*

Proof By induction on the proof of "$M \hookrightarrow_{wh} N$".

For instance, if $M \equiv{\downarrow} M_1$ with M_1 whr, then $N \equiv{\downarrow} N_1$ where $M_1 \hookrightarrow_{wh} N_1$ and $P \equiv{\downarrow} P_1$ where $M_1 \hookrightarrow_* P_1$. By induction hypothesis, there is a term Q_1 such that $P_1 \hookrightarrow_{wh} Q_1$ and $N_1 \hookrightarrow_* Q_1$. We take $Q :=_{\downarrow} Q_1$.

□

Lemma 9.11 *If* $\Delta \vdash M : A$, $M \hookrightarrow_{wh} N$ *and for each element* M_k *of* M, $\Delta^{n_k} \vdash M_k \rightarrow_{nf} P_k : B_k$ *then* $\Delta \vdash M \rightarrow_{wh} N : A$.

Proof By induction on the proof of "$M \hookrightarrow_{wh} N$".

For instance, if $M \equiv\downarrow M_1$, then $N \equiv\downarrow N_1$ where $M_1 \hookrightarrow_{wh} N_1$. Then by induction hypothesis, $\Delta \vdash M_1 \rightarrow_{wh} N_1 : \Box A$ and we apply rule $(W\downarrow)$.

If $M \equiv\downarrow\uparrow M_1$, then $N \equiv M_1$. By hypothesis $\Delta \vdash M : A$ and $\Delta^n \vdash\uparrow M_1 \rightarrow_{nf} P : B$. By Lemma 9.7, $\Delta \vdash\uparrow M_1 \rightarrow_{nf} P : \Box A$. We apply rule $(W\beta_{\Box})$ and we are done.

□

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - B.P. 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Lorraine : Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 Villers lès Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot St Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, B.P. 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399