A formalisation of the strong normalization of system T

Titouan Leclercq

February 2024

Introduction

This document is a support to detail the work done in the Coq files. It is not mandatory to understand the files, but as a lot of the proofs are long and uninteresting, this document gives a more sensible outline of the arguments.

This project is done for the "programmation fonctionnelle et coq" course of the LMFI master, the aim of which is to discover the proof assistant Coq. The final project is a formalization project of a chosen proof. In my case, I chose to formalize the proof of strong normalization for a typed lambda-calculus with the method of reducibility candidates. The 9 files in the `Source` directory are decomposed in 3 sections:

- The definitions of the lambda-calculus, the type system and the basic tools
- The main structural results on $\mathcal{SN}$
- The definition of saturated sets, the adequacy theorem and the normalization theorem.

The document will thus naturally be composed of 3 sections, one for each section in the files. As all the proofs are formalised in the Coq files, almost no proposition will be proved here, but some important argument or technicality may be outline for some propositions.

The structure of the proof was inspired by a course from Colin Riba, called *Preuves et programmes*, given in the ÉNS de Lyon in 2023 in the M1 Informatique théorique.

1 First definitions

1.1 File 1

The lambda-calculus we will study is based on Gödel's system T, as described in *e.g.* [1], but instead of just using the types `int` and `bool`, we add product and coproduct types, along with their nullary cases: `unit` and `void`. We will use De Bruijn indexes as convention to manage variables. This means that instead of a variable like x, we will use a number telling how many binders one has to cross before finding the one binding the variable. The binders are λ and δ. If there is no binder in a term, for example in the term $\lambda\ 1$ then the variable can be bound in an external context, as will be the case for types.

Definition 1 (Terms)**.** The set Λ of our λ-terms is given by the following grammar:

$$t, u, v ::= n \mid \lambda\ t \mid t\ u \mid \langle t, u\rangle \mid \pi_1\ t \mid \pi_2\ t \mid \langle\rangle \mid \kappa_1\ t \mid \kappa_2\ t \mid \delta(0 \mapsto t \mid 0 \mapsto u)\ v \mid \delta_\bot\ t \\ \mid \mathbf{0} \mid \mathrm{S}\ t \mid \mathrm{rec}\ t\ u\ v \mid \top \mid \bot \mid \text{if } t \text{ then } u \text{ else } v$$

Next, we want to define the reduction on terms, but this needs a prior step, which is the lifting of terms. The idea of lifting is that, if t is interpreted in a context Γ binding its free occurrences, it is possible to interpret it the same way by adding another context in the middle of Γ and changing the variables accordingly: if $\Gamma = \Gamma_0 +\!+ \Gamma_1$, then taking $\Gamma' = \Gamma_0 +\!+ \Delta +\!+ \Gamma_1$ means that we have to add $|\Delta|$ to each variable which is bigger than $|\Gamma_0|$.

Definition 2 (Lift)**.** For all $k, n \in \mathbb{N}$, we define $\uparrow_k^n : \Lambda \to \Lambda$ by induction on its argument:

- if $t = p$ is a variable and $p < k$ then $\uparrow_k^n t = p$
- if $t = p$ is a variable and $p \geq k$ then $\uparrow_k^n t = p + n$
- if $t = \lambda\ u$ then $\uparrow_k^n t = \lambda\ (\uparrow_{k+1}^n u)$
- if $t = \delta(0 \mapsto u \mid 0 \mapsto v)\ w$ then $\uparrow_k^n t = \delta(0 \mapsto (\uparrow_{k+1}^n u) \mid 0 \mapsto (\uparrow_{k+1}^n v))\ (\uparrow_k^n t)$
- for the other cases, we just apply inductively $\uparrow_k^n$ on each subterm.

There is a way to rewrite two liftings, but we use it mostly when one is starting from 0:

Lemma 1.1.1. For all $t \in \Lambda, k, n, m \in \mathbb{N}$, we have

$$\uparrow_0^k (\uparrow_n^m t) = \uparrow_{k+n}^m (\uparrow_0^k t)$$

We can also see that lifting by 0 amount to doing nothing:

Lemma 1.1.2. For all $t \in \Lambda, k \in \mathbb{N}$, $\uparrow_k^0 t = t$.

Lifting behaves well, leading to lemmas in the Coq file which can be summarized by:

Proposition 1.1.3. For terms $t, u \in \Lambda$, if $u = C(u_1, \ldots, u_p)$ for some term constructor C and terms $u_1, \ldots, u_p$ and $\uparrow_k^n t = u$, then $t = C(t_1, \ldots, t_p)$ for some terms $t_1, \ldots, t_p$.

1.2 File 2

This file concentrate on the substitution operation, its properties, and its extension to simultaneous substitution.

1.2.1 Definition

In this file, we define the substitution, essential to define the β-reduction. We use the lift operation here for the substitution under a binder: if we want to substitute k with u in $\lambda\, t$, we will need the free variables in u, which are bound above what we substitute on, to be lifted to add in the context the binder λ.

Definition 3 (Substitution)**.** Let $t, u \in \Lambda$ and $k \in \mathbb{N}$. We define $t[u/k]$ (written $\{k \rightsquigarrow u\}\ t$ in the Coq files) by induction on t :

- if $t = k$, then $t[u/k] = u$.
- if $t = n$ with $n < k$, then $t[u/k] = n$.
- if $t = n$ with $n > k$, then $t[u/k] = n - 1$.
- if $t = \lambda\, v$ then $t[u/k] = \lambda\, (v[\uparrow_0^1 u/k + 1])$.
- if $t = \delta(0 \mapsto v \mid 0 \mapsto w)\ x$ then $t[u/k] = \delta(0 \mapsto v[\uparrow_0^1 u/k + 1] \mid 0 \mapsto w[\uparrow_0^1 u/k + 1])\ x[u/k]$
- for the rest, we just need to substitute under the subterms.

First, we have two commutation lemmas, between substitution and lifting.

Lemma 1.2.1. For all $t, u \in \Lambda, k, n \in \mathbb{N}$, we have the following:

$$\uparrow_k^n (t[u/0]) = (\uparrow_{k+1}^n t)[\uparrow_k^n u/0]$$

Lemma 1.2.2. For all $t, u \in \Lambda, k, n \in \mathbb{N}$, we have the following:

$$\uparrow_0^n (t[u/k]) = \uparrow_0^n t[\uparrow_0^n u/n + k]$$

Lifting can also absorb substitution.

Lemma 1.2.3. For all $t, k \in \Lambda, k \in \mathbb{N}$, we have the following:

$$t = \uparrow_k^1 t[u/k]$$

We also have a commutation property between two substitutions.

Proposition 1.2.4. For all $t, u, v \in \Lambda, n \in \mathbb{N}$, we have the following:

$$t[v/0][u/n] = t[\uparrow_0^1 u/n + 1][v[u/n]]$$

1.2.2 Simultaneous substitution

As seen just before, substituting two terms in some third term gives different result depending on the order of the substitution. To avoid that, we define the new notion of simultaneous substitution, which will take an arbitrary large finite number of terms and make a substitution in a term with all those at once, without the possibility of having a substitution between two substituting terms.

Definition 4 (Permutation and substitution)**.** We call permutation a list of terms, and write $\mathfrak{S}$ the set of permutations. We define the lift operator on $\mathfrak{S}$ by applying it on each term. It preserves length and has the same properties as the lift operator for terms.

For a term $t \in \Lambda$, a permutation $\sigma \in \mathfrak{S}$ and $n \in \mathbb{N}$, we define $t[\sigma/k]$ (written $\{k \mapsto \sigma\}\ t$ in the Coq files) by induction on t :

- if $t = n$ with $n < k$ then $t[\sigma/k] = n$.
- if $t = n$ with $k \leq n < k + |\sigma|$, then $t[\sigma/k] = \sigma_{n-k}$, the $n - k$th term of σ.
- if $t = n$ with $t \geq k + |\sigma|$ then $t[\sigma/k] = n - |\sigma|$.
- if $t = \lambda\ u$ then $t[\sigma/k] = \lambda\ (t[\uparrow_0^1 \sigma/k + 1])$.
- for the other constructor, we just apply the substitution recursively, except in the binders of δ where we lift the permutation and add 1 to the substitution index, as for the classical substitution.

Remark. Substituting by the empty list is the identity function and substituting by a singleton list is just a classical substitution.

As for classical substitution, substituting on a sufficiently high lift just reduces the lift.

Lemma 1.2.5. For all $t \in \Lambda, \sigma \in \mathfrak{S}, k \in \mathbb{N}$ we have the following:

$$(\uparrow_k^{|\sigma|} t)[\sigma/k] = t$$

Given a permutation $\tau = u :: \sigma$, we can decompose $t[\tau/k]$ from the substitution by u and by σ in two ways: first applying σ or first applying u.

Lemma 1.2.6. For all $t, u \in \Lambda, \sigma \in \mathfrak{S}, k \in \mathbb{N}$, we have the following:

$$(t[\uparrow_k^{|\sigma|} u])[\sigma/k] = t[u :: \sigma/k]$$

Lemma 1.2.7. For all $t, u \in \Lambda, \sigma \in \mathfrak{S}, k \in \mathbb{N}$, we have the following:

$$t[u :: \sigma/k] = (t[\uparrow_k^1 \sigma/k + 1])[u/k]$$

1.2.3 Maximal Index

We now define the maximal index of a term, which is the number of the maximal index among all the free variables of a term, ignoring the bound variables (for example, $\lambda\ 1$ has as maximal index 0). As we want to be able to distinguish between a closed term and a term with maximal index 0, we lift all maximal indexes by 1 to keep 0 for the closed terms. The maximal index of a term will be used to prove that, given a term t, we can take the substitution $\sigma_t := [0; 1; \ldots; n]$ for n large enough such that $t[\sigma/0] = t$.

Definition 5 (Maximal index)**.** Let $t \in \Lambda$, we define $\max t$ by induction on t:

- if $t = n$ then $\max t = 1 + n$
- if $t = \lambda\ u$ then $\max t = (\max u) - 1$
- if $t = \delta(0 \mapsto u \mid 0 \mapsto v)\ w$ then $\max t = \max((\max u) - 1, (\max v) - 1, \max w)$
- if $t = C(t_1, \ldots, t_p)$ for a term constructor C, then $\max t = \max_{i=1,\ldots,p}(\max t_i)$

Now, we define the permutation $[0; 1; \ldots; n]$.

Definition 6 (Identity permutation)**.** We define the identity permutation of length n as $\sigma_n := [0; \ldots; n]$.

Proposition 1.2.8. For all $t \in \Lambda$, we have:

$$t = t[\sigma_{\max t}/0]$$

1.3 File 3

In this file, we study the reduction, first from an abstract perspective with rewriting system theory, then we state the basic properties of closure, compatibility and so on, and finally we prove the confluence of the reduction using the parallel reduction presented *e.g.* in [2].

1.3.1 Abstract rewriting system

We define the classical tools of abstract rewriting systems.

Definition 7 (Rewriting rule, closures)**.** A rewriting rule is a subset $\to \subseteq \Lambda^2$.

For a rule $\to$, we define its transitive closure $\to^+$ as the smallest relation closed under the following rules:

$$\frac{t \to u}{t \to^+ u} \qquad \frac{t \to^+ u \quad u \to v}{t \to^+ v}$$

For a rule $\to$, we define its transitive and reflexive closure $\to^*$ as the smallest relation closed under the following rules:

$$\frac{}{t \to^* t} \qquad \frac{t \to^* u \quad u \to v}{t \to^* v}$$

For a rule $\to$, we define its reflexive, symmetric and transitive closure $\to^\sim$ as the smallest relation closed under the following rules:

$$\frac{}{t \to^\sim t} \qquad \frac{t \to^\sim u \quad u \to v}{t \to^\sim v} \qquad \frac{t \to^\sim u \quad v \to u}{t \to^\sim v}$$

Proposition 1.3.1. For all rule $\to$, we have $\to \subseteq \to^+ \subseteq \to^* \subseteq \to^*$, the three last are transitive, the two last are reflexive and the last one is symmetric.

Now, we define the properties of diamond, local confluence, confluence and Church-Rosser.

Definition 8 (Diamond, local confluence, confluence, Church-Rosser)**.** Let $\to$ be a rewriting rule. We say that $\to$ has the diamond property if

$$\forall t, u, v \in \Lambda, (t \to u) \land (t \to v) \implies \exists w \in \Lambda, (u \to w) \land (v \to w)$$

It has the local confluence property if

$$\forall t, u, v \in \Lambda, (t \to u) \land (t \to v) \implies \exists w \in \Lambda, (u \to^* w) \land (v \to^* w)$$

It has the confluence property if

$$\forall t, u, v \in \Lambda, (t \to^* u) \land (t \to^* v) \implies \exists w \in \Lambda, (u \to^* w) \land (v \to^* w)$$

It has the Church-Rosser property if

$$\forall t, u \in \Lambda, t \to^\sim u \implies \exists v \in \Lambda, (t \to^* v) \land (u \to^* v)$$

Lemma 1.3.2. Let $\to$ be a rewriting rule. Then $\to$ is confluent if and only if $\to^*$ has the diamond property.

Lemma 1.3.3. Let $\to$ be a rewriting rule. If $\to$ has the diamond property, then $\to$ has confluence.

Lemma 1.3.4. Confluence and Church-Rosser property are equivalent.

1.3.2 Beta-reduction

Definition 9 (β-reduction)**.** We define $\rhd \subseteq \Lambda^2$ as the smallest compatible relation (*i.e.* the smallest relation for which $C(t_1, \dots, t_n) \rhd C(t'_1, \dots, t'_n)$ for a term constructor C if for all i, $t_i = t'_i$ except for one where $t_i \rhd t'_i$) containing the following rules:

$$\frac{}{(\lambda\, t)\ u \rhd t[u/0]} \qquad \frac{}{\pi_1\ \langle t, u\rangle \rhd t} \qquad \frac{}{\pi_2\ \langle t, u\rangle \rhd u} \qquad \frac{}{\delta(0 \mapsto u \mid 0 \mapsto v)\ \kappa_1\ t \rhd u[t/0]}$$

$$\frac{}{\delta(0 \mapsto u \mid 0 \mapsto v)\ \kappa_2\ t \rhd v[t/0]} \qquad \frac{}{\mathrm{rec}\ u\ v\ \mathbf{0} \rhd u} \qquad \frac{}{\mathrm{rec}\ u\ v\ (\mathrm{S}\ t) \rhd v\ t\ (\mathrm{rec}\ u\ v\ t)} \qquad \frac{}{\text{if } \top \text{ then } u \text{ else } v \rhd u}$$

$$\frac{}{\text{if } \bot \text{ then } u \text{ else } v \rhd v}$$

Proposition 1.3.5. As $\rhd$ is compatible, $\rhd^*$ is also compatible.

The lift operator commutes with the reduction.

Proposition 1.3.6. Let $t, u \in \Lambda, k, n \in \mathbb{N}$, then:

- if $t \rhd u$ then $\uparrow_k^n t \rhd \uparrow_k^n u$
- if $t \rhd^* u$ then $\uparrow_k^n t \rhd^* \uparrow_k^n u$

Substitution commutes with reduction on one side, on the other side a reduction corresponds to (possibly empty) sequence of reductions.

Proposition 1.3.7. Let $t, u, v \in \Lambda, k \in \mathbb{N}$, then:

- if $t \rhd u$ then $t[v/k] \rhd u[v/k]$
- if $t \rhd^* u$ then $t[v/k] \rhd^* u[v/k]$
- if $u \rhd v$ then $t[u/k] \rhd^* t[v/k]$
- if $t \rhd^* u$ then $t[u/k] \rhd^* t[v/k]$

1.3.3 Church-Rosser property

We now prove the Church-Rosser property for $\rhd$.

Definition 10 (Parallel reduction)**.** We define the relation $\rhd_\parallel$ as follows:

- it is reflexive on variables, $\langle\rangle$, $\mathbf{0}$, $\top$ and $\bot$.
- instead of simply adding compatibility, we allow ourselves to use the parallel version with $C(t_1, \ldots, t_n) \rhd_\parallel C(t'_1, \ldots, t'_n)$ with for all i, $t_i \rhd_\parallel t'_i$.

Lemma 1.3.8. $\rhd_\parallel$ is reflexive, and $\rhd \subseteq \rhd_\parallel \subseteq \rhd^*$.

Remark. With the last inclusions, it is straightforward that $\rhd_\parallel^* = \rhd^*$.

Lemma 1.3.9. For all $t, t', u, u' \in \Lambda, k \in \mathbb{N}$, if $t \rhd_\parallel t'$ and $u \rhd_\parallel u'$, then $t[u/k] \rhd_\parallel t'[u'/k]$.

Definition 11 (Total reduction)**.** We define the total reduction $t^\bullet$ of t by induction on t by applying each possible parallel redexes appearing in t.

Lemma 1.3.10. For all $t, u \in \Lambda$, if $t \rhd_\parallel u$ then $u \rhd_\parallel t^\bullet$.

Theorem 1 (Church-Rosser)**.** The relation $\rhd_\parallel$ is diamond, and hence $\rhd$ has the Church-Rosser property.

1.4 File 4

This file focuses on the typing system of our lambda-calculus and proves the subject reduction property.

Definition 12 (Type, context)**.** We define the set Types of types by the following grammar:

$$T, U ::= X \mid \texttt{int} \mid \texttt{bool} \mid \texttt{unit} \mid \texttt{void} \mid T \to U \mid T \times U \mid T + U$$

where X is a countable set of type variable (taken to be int in the Coq file).

A typing context is a list of types, we will write C_T the set of typing contexts. For a typing context Γ, we will write Γ_n for its nth element.

Definition 13 (Typing relation)**.** We define $(\_ \vdash \_ : \_) \subseteq C_T \times \Lambda \times \text{Types}$ by the following rules:

$$\frac{k < |\Gamma|}{\Gamma \vdash k : \Gamma_k} \qquad \frac{\Gamma, T \vdash t : U}{\Gamma \vdash \lambda\, t : T \to U} \qquad \frac{\Gamma \vdash t : T \to U \quad \Gamma \vdash u : T}{\Gamma \vdash t\ u : U} \qquad \frac{}{\Gamma \vdash \mathbf{0} : \texttt{int}}$$

$$\frac{\Gamma \vdash t : \texttt{int}}{\Gamma \vdash \mathrm{S}\, t : \texttt{int}} \qquad \frac{\Gamma \vdash u : T \quad \Gamma \vdash v : \texttt{int} \to T \to T \quad \Gamma \vdash t : \texttt{int}}{\Gamma \vdash \mathrm{rec}\ u\ v\ t : T} \qquad \frac{}{\Gamma \vdash \top : \texttt{bool}}$$

$$\frac{}{\Gamma \vdash \bot : \texttt{bool}} \qquad \frac{\Gamma \vdash t : \texttt{bool} \quad \Gamma \vdash u : T \quad \Gamma \vdash v : T}{\Gamma \vdash \text{if } t \text{ then } u \text{ else } v : T} \qquad \frac{\Gamma \vdash t : T \quad \Gamma \vdash u : U}{\Gamma \vdash \langle t, u \rangle : T \times U} \qquad \frac{\Gamma \vdash t : T \times U}{\Gamma \vdash \pi_1\ t : T}$$

$$\frac{\Gamma \vdash t : T \times U}{\Gamma \vdash \pi_2\ t : U} \qquad \frac{}{\Gamma \vdash \langle\rangle : \texttt{unit}} \qquad \frac{\Gamma \vdash t : T}{\Gamma \vdash \kappa_1\ t : T + U} \qquad \frac{\Gamma \vdash u : U}{\Gamma \vdash \kappa_2\ u : T + U}$$

$$\frac{\Gamma, T \vdash u : V \quad \Gamma, U \vdash v : V \quad \Gamma \vdash t : T + U}{\Gamma \vdash \delta(0 \mapsto u \mid 0 \mapsto v)\ t : V} \qquad \frac{\Gamma \vdash t : \texttt{void}}{\Gamma \vdash \delta_\bot\ t : T}$$

Theorem 2 (Subject reduction)**.** For all $t, u \in \Lambda, T \in \text{Types}, \Gamma \in C_T$, if $\Gamma \vdash t : T$ and $t \rhd u$ (resp $t \rhd^* u$) then $\Gamma \vdash u : T$.

2 The structure of SN

This section focuses on the main results which will be used in the final section, to define the interpretation of types. First, we need to introduce the notion of elimination context.

2.1 File 5

In this file, we define elimination contexts and weak head reduction. The latter is the reduction corresponding to the closure by the former.

2.1.1 Elimination context

An elimination context can be seen as a way to destruct a term: it will be a sequence of elimination procedures with a so-called hole, which can be filled by a term. Eliminators are applications, projections, case disjunction on sum types, nullary case on empty type, recursion on natural numbers and case disjunction for boolean.

Definition 14 (Elimiation context, filling)**.** We define the set of Elimination contexts, Elim, by induction with the following grammar:

$$E ::= [\,] \mid E\ t \mid \pi_1\ E \mid \pi_2\ E \mid \delta(0 \mapsto u \mid 0 \mapsto v)\ E \mid \text{rec}\ u\ v\ E \mid \text{if}\ E\ \text{then}\ u\ \text{else}\ v \mid \delta_\perp\ E$$

For a term $t \in \Lambda$, we define $E[t]$ as E where $[\,]$ is replaced by t.

Proposition 2.1.1. Let $E \in \text{Elim}, t, u \in \Lambda$, then:

- if $t \rhd u$ then $E[t] \rhd E[u]$.
- if $t \rhd^* u$ then $E[t] \rhd^* E[u]$.

We also define composition of context.

Definition 15 (Composition)**.** Let $E, F \in \text{Elim}$, we define $E \circ F$ as E where the hole is replaced by F.

Proposition 2.1.2. Composition of context is associative, and it behaves as composition for the following reason:

$$\forall E, F \in \text{Elim}, \forall t \in \Lambda, (E \circ F)[t] = E[F[t]]$$

We also define $\uparrow_k^n$ on contexts and show that this behaves as the usual lift operator. Now, we define $\rhd_e$ to be the reduction happening between context.

Definition 16 (Context reduction)**.** We define $\rhd_e \subseteq \text{Elim} \times \text{Elim}$, the context reduction, by the fact that $E \rhd_e F$ if and only if E and F differ by only a term (say t_E in E and t_F in F) such that $t_E \rhd t_F$.

$\rhd_e^*$ is defined as the reflexive and transitive closure of $\rhd_e$.

Proposition 2.1.3. $\rhd_e$ commutes with filling: for all $E, F \in \text{Elim}, t \in \Lambda$, if $E \rhd_e F$ (resp $E \rhd_e^* F$) then $E[t] \rhd F[t]$ (resp $E[t] \rhd_e^* F[t]$).

Lemma 2.1.4. If n is a variable and E an elimination context, then $E[n]$ cannot have the shape of a constructor, *i.e.* is not an abstraction nor a pair, nor a coprojection, nor a constant, nor a successor.

Lemma 2.1.5. If $E[n] \rhd t$ where E is an elimination context, n a variable and t a term, then t is of the form $F[n]$ with $E \rhd_e F$.

2.1.2 Flattening

We study here an operation on the terms which just flattens all free variables above some k to k. Its purpose is to be set to $k = 0$ and to show that some properties P are such that $P(t) \iff P(\text{flatten}\ t\ 0)$, meaning that changing the free variables in t will not change the belonging in P (as flattening gives the same result no matter how one changes the free variables).

Definition 17. For a term $t \in \Lambda$, we define $\overline{t}^k$ to be t where, for each free variable i in t, i is replaced by k if $k \leq i$ (taking into account the number of binders) and is left as i if $i < k$.

The lemmas after this just show that if $\overline{t}^k = C(u_1, \ldots, u_p)$ for some $u_1, \ldots, u_p$, then t itself is of the form $C(u'_1, \ldots, u'_p)$.

We define in a similar way $\overline{E}^k$ for a context E.

Proposition 2.1.6. For all $E \in \mathrm{Elim}, t \in \Lambda$, we have :

$$\overline{E[t]}^k = \overline{E}^k[\overline{t}^k]$$

Next, we prove the essential lemma about flattening:

Lemma 2.1.7. For all $t \in \Lambda$, $k, n \in \mathbb{N}$, we have:

$$\overline{\uparrow_k^n t}^0 = \overline{t}^0$$

In the other direction, we also have a commutation equality.

Lemma 2.1.8. For all $t \in \Lambda, k, n, p \in \mathbb{N}$, if $k \leq p$ then:

$$\overline{\uparrow_k^n t}^{p+n} = \uparrow_k^n (\overline{t}^p)$$

Now, we can show that flattening commutes with substitution:

Proposition 2.1.9. For all $k, n \in \mathbb{N}, t, u \in \Lambda$, if $k \leq n$ then:

$$\overline{t[u/k]}^n = \overline{t}^{n+1}[\overline{u}^n/k]$$

This leads to the following lemma:

Lemma 2.1.10. For all $t, u \in \Lambda, k \in \mathbb{N}$, if $t \rhd u$ then $\overline{t}^k \rhd \overline{u}^k$.

Lemma 2.1.11. For all $t, u \in \Lambda, k \in \mathbb{N}$, if $\overline{t}^k \rhd u$, then u is of the form $\overline{v}^k$ with $t \rhd v$.

2.1.3 Weak head reduction

The weak head reduction is a weakening of the usual reduction $\rhd$, but where the compatibility is weakened to include only elimination contexts.

Definition 18 (Weak head reduction)**.** We define the relation $\rhd_0$ by the following cases:

- $(\lambda\ t)\ u \rhd_0 t[u/0]$
- $\pi_1\ \langle t, u\rangle \rhd_0 t$
- $\pi_2\ \langle t, u\rangle \rhd_0 u$
- $\delta(0 \mapsto u \mid 0 \mapsto v)\ (\kappa_1\ t) \rhd_0 u[t/0]$
- $\delta(0 \mapsto u \mid 0 \mapsto v)\ (\kappa_2\ t) \rhd_0 v[t/0]$
- $\mathrm{rec}\ u\ v\ \mathbf{0} \rhd_0 u$
- $\mathrm{rec}\ u\ v\ (\mathrm{S}\ t) \rhd_0 v\ t\ (\mathrm{rec}\ u\ v\ t)$
- $\mathrm{if}\ \top\ \mathrm{then}\ u\ \mathrm{else}\ v \rhd_0 u$
- $\mathrm{if}\ \bot\ \mathrm{then}\ u\ \mathrm{else}\ v \rhd_0 v$

We define the relation $\rhd_h$ as the smallest relation containing $\rhd_0$ and such that for all $t \rhd_h u$ and context E, $E[t] \rhd_h E[u]$.

Remark. The definition in the Coq file is inductive, but just follows this idea. We prove just after that the inductive definition is equivalent to the definition we just gave above.

We have the inclusions $\rhd_0 \subseteq \rhd_h \subseteq \rhd$.

Proposition 2.1.12. $\rhd_h$ and $\rhd_h^*$ both commute with $\overline{-}^k$ and $\uparrow_k^n$ for all $k, n \in \mathbb{N}$.

Proposition 2.1.13. As for $\rhd$, if $\overline{t}^k \rhd_h u$ then u is of the formm $\overline{v}^k$ with $t \rhd_h v$.

Lemma 2.1.14. A term t reduces to some $E[n]$ with E a context and n a variable if and only if $\overline{t}^k$ also reduces to some $E[n]$ (not necessarily the same E and n).

This leads to one of the main results of this subsection: lifting inside the context does not change the property that a term reduces to some $E[n]$ for E a context and n a variable.

Proposition 2.1.15. For all $E, F \in \mathrm{Elim}, t \in \Lambda, k, m, n \in \mathbb{N}$, if $E[t] \rhd_h^* F[n]$ then there exists F' and n' such that $E[\uparrow_k^m t] \rhd_h^* F[n]$.

Proof. From the previous lemma, we know that $E[t] \rhd_h^* F[n]$ for some F, n if and only if it is also the case for $\overline{E[t]}^0$, *i.e.* if it is also the case for $\overline{E}^0[\overline{t}^0]$, but $\overline{t}^0 = \overline{\uparrow_k^m t}^0$ and so, as $\overline{E}^0[\overline{t}^0]$ reduces to some $F[n]$, so do $E[\uparrow_k^m t]$. □

2.2 File 6

This files introduces the set $\mathcal{SN}$ and the notion of normal forms. There is also a part on weakly normalizing terms, but this one is not used anyway.

2.2.1 Normal form

Definition 19 (Normal form)**.** We say that t is a normal form if there is no reduction from t.

Proposition 2.2.1. Let t be a normal form and u be such that $t \rhd^* u$, then $t = u$.

We give the properties of normal forms.

Proposition 2.2.2. For terms t, u, v, we have the following:

- for all variable n, n is a normal form.
- t is a normal form if and only if $\lambda\, t$ is a normal form.
- if $t\ u$ is a normal form, then t and u are both normal forms.
- $\langle t, u\rangle$ is a normal form if and only if both t and u are normal forms.
- if t is a normal form, then $\pi_1\ t$ and $\pi_2\ t$ are normal forms.
- $\top, \bot, \langle\rangle, \mathbf{0}$ are normal forms.
- t is a normal form if and only if $\kappa_1\ t$ is a normal form, if and only if $\kappa_2\ t$ is a normal form.
- if $\delta(0 \mapsto u \mid 0 \mapsto v)\ t$ is a normal form, then so are t, u and v.
- if if t then u else v is a normal form, then so are t, u and v.
- if rec $u\ v\ t$ is a normal form, then so are t, u and v.
- t is a normal form if and only if S t is a normal form.
- t is a normal form if and only if $\delta_\bot\ t$ is a normal form.

We see that, for a constructor, we have an equivalence while it is only an implication for an eliminator, the same phenomenon will happen for $\mathcal{SN}$.

Proposition 2.2.3. There is unicity of a normal form to which reduces a term: if $t \rhd^* u$, $t \rhd^* v$ and both u and v are normal forms, then $u = v$.

2.2.2 Weakly normalizing terms

Definition 20. A term t is weakly normalizing if there exists a normal form u such that $t \rhd^* u$. We write $\mathcal{WN}$ the set of weakly normalizing terms.

Proposition 2.2.4. A normal form is weakly normalizing. $\mathcal{WN}$ is stable by expansion and reduction by $\rhd$.

2.2.3 Strongly normalizing terms

Definition 21 (SN)**.** We define the set $\mathcal{SN}$ of strongly normalizing terms as the smallest set stable under the following rule:

$$\frac{\forall u \in \Lambda, t \rhd u \implies u \in \mathcal{SN}}{t \in \mathcal{SN}}$$

Remark. A normal form is also strongly normalizing, by emptiness of the set of successors.

The classical characterization that there is no infinite sequence of reduction is a consequence of our definition, but is weaker as the equivalence relies on the dependant axiom of choice. The above definition also allow to reason by induction.

Proposition 2.2.5. If $t \in \mathcal{SN}$, then there is no sequence $(t_n)_{n\in\mathbb{N}}$ such that $t_0 = t$ and $t_n \rhd t_{n+1}$.

Proposition 2.2.6. The set $\mathcal{SN}$ is stable by reduction.

The other lemmas of this subsubsection are only proof of stability by constructor and eliminators. As for normal forms, constructors give an equivalence while eliminators only give an implication.

Proposition 2.2.7. If $t[u/k] \in \mathcal{SN}$ for $t, u \in \Lambda, k \in \mathbb{N}$, then $t \in \mathcal{SN}$.

2.2.4 SN contexts

We then study interaction between $\mathcal{SN}$ and elimination contexts.

Definition 22 (SNE)**.** The set $\mathcal{SN}_e$ is the set of contexts made only by terms in $\mathcal{SN}$.

Proposition 2.2.8. If $E \in \mathcal{SN}_e$ and $E \rhd_e F$ (resp $E \rhd_e^* F$) then $F \in \mathcal{SN}_e$.

Proposition 2.2.9. For all $E \in \mathrm{Elim}, t \in \Lambda$, if $E[t] \in \mathcal{SN}$ then $E \in \mathcal{SN}_e$ and $t \in \mathcal{SN}$.

Remark. The converse does not work, for example $\lambda\ (0\ 0) \in \mathcal{SN}$ and $[\,]\ (\lambda\ (0\ 0)) \in \mathcal{SN}_e$, but $(\lambda\ (0\ 0))\ (\lambda\ (0\ 0))$ is not normalizing.

We can also prove one of the important lemmas: $\mathcal{SN}$ is stable by lifting in a restricted way, by changing only the variables inside a context.

Lemma 2.2.10. For all $E \in \mathrm{Elim}, t \in \Lambda, k, n \in \mathbb{N}$, we have

$$E[t] \in \mathcal{SN} \iff E[\uparrow_k^n t] \in \mathcal{SN}$$

Corollary 1. $\forall t \in \Lambda, \forall k, n \in \mathbb{N}, t \in \mathcal{SN} \iff \uparrow_k^n t \in \mathcal{SN}$

The next lemmas are made to prove the following:

Proposition 2.2.11. For all $E \in \mathcal{SN}_e, n \in \mathbb{N}$, we have $E[n] \in \mathcal{SN}$.

2.3 File 7

This file proves the weak standardization and weak head expansion.

Theorem 3 (Weak standardization)**.** Let $E \in \mathrm{Elim}, t, u, v, w \in \Lambda$, then:

- if $E[(\lambda\ t)\ u] \rhd v$ then either $v = E[t[u/0]]$, $v = F[(\lambda\ t)\ u]$ with $E \rhd_e F$, $v = E[(\lambda\ t')\ u]$ with $t \rhd t'$ or $v = E[(\lambda\ t)\ u']$ with $u \rhd u'$.
- if $E[\pi_1\ \langle t, u\rangle] \rhd v$ then either $v = E[t]$, $v = F[\pi_1\ \langle t, u\rangle]$ with $E \rhd_e F$, $v = E[\pi_1\ \langle t', u\rangle]$ with $t \rhd t'$ or $v = E[\pi_1\ \langle t, u'\rangle]$ with $u \rhd u'$.
- if $E[\pi_2\ \langle t, u\rangle] \rhd v$ then either $v = E[u]$, $v = F[\pi_2\ \langle t, u\rangle]$ with $E \rhd_e F$, $v = E[\pi_2\ \langle t', u\rangle]$ with $t \rhd t'$ or $v = E[\pi_2\ \langle t, u'\rangle]$ with $u \rhd u'$.
- if $E[\delta(0 \mapsto u \mid 0 \mapsto v)\ (\kappa_1\ t)] \rhd w$ then either $w = E[u[t/0]]$, $w = F[\delta(0 \mapsto u \mid 0 \mapsto v)\ (\kappa_1\ t)]$ with $E \rhd_e F$, $w = E[\delta(0 \mapsto u \mid 0 \mapsto v)\ (\kappa_1\ t')]$ with $t \rhd t'$, $w = E[\delta(0 \mapsto u' \mid 0 \mapsto v)\ (\kappa_1\ t)]$ with $u \rhd u'$, or $w = E[\delta(0 \mapsto u \mid 0 \mapsto v')\ (\kappa_1\ t)]$ with $v \rhd v'$.
- if $E[\delta(0 \mapsto u \mid 0 \mapsto v)\ (\kappa_2\ t)] \rhd w$ then either $w = E[v[t/0]]$, $w = F[\delta(0 \mapsto u \mid 0 \mapsto v)\ (\kappa_2\ t)]$ with $E \rhd_e F$, $w = E[\delta(0 \mapsto u \mid 0 \mapsto v)\ (\kappa_2\ t')]$ with $t \rhd t'$, $w = E[\delta(0 \mapsto u' \mid 0 \mapsto v)\ (\kappa_2\ t)]$ with $u \rhd u'$, or $w = E[\delta(0 \mapsto u \mid 0 \mapsto v')\ (\kappa_2\ t)]$ with $v \rhd v'$.
- if $E[\mathrm{rec}\ u\ v\ \mathbf{0}] \rhd w$ then either $w = E[u]$, $w = F[\mathrm{rec}\ u\ v\ \mathbf{0}]$ with $E \rhd_e F$, $w = E[\mathrm{rec}\ u\ v\ \mathbf{0}]$ with $u \rhd u'$, or $w = E[\mathrm{rec}\ u\ v\ \mathbf{0}]$ with $v \rhd v'$.
- if $E[\mathrm{rec}\ u\ v\ (\mathrm{S}\ t)] \rhd w$ then either $w = E[v\ t\ (\mathrm{rec}\ t\ v\ t)]$, $w = F[\mathrm{rec}\ u\ v\ (\mathrm{S}\ t)]$ with $E \rhd_e F$, $w = E[\mathrm{rec}\ u\ v\ (\mathrm{S}\ t')]$ with $t \rhd t'$, $w = E[\mathrm{rec}\ u'\ v\ (\mathrm{S}\ t)]$ with $u \rhd u'$, or $w = E[\mathrm{rec}\ u\ v'\ (\mathrm{S}\ t)]$ with $v \rhd v'$.
- if $E[\mathrm{if}\ \top\ \mathrm{then}\ u\ \mathrm{else}\ v] \rhd w$ then either $w = E[u]$, $w = F[\mathrm{if}\ \top\ \mathrm{then}\ u\ \mathrm{else}\ v]$ with $E \rhd_e F$, $w = E[\mathrm{if}\ \top\ \mathrm{then}\ u'\ \mathrm{else}\ v]$ with $u \rhd u'$, $w = E[\mathrm{if}\ \top\ \mathrm{then}\ u'\ \mathrm{else}\ v]$ with $u \rhd u'$.
- if $E[\mathrm{if}\ \bot\ \mathrm{then}\ u\ \mathrm{else}\ v] \rhd w$ then either $w = E[v]$, $w = F[\mathrm{if}\ \bot\ \mathrm{then}\ u\ \mathrm{else}\ v]$ with $E \rhd_e F$, $w = E[\mathrm{if}\ \bot\ \mathrm{then}\ u'\ \mathrm{else}\ v]$ with $u \rhd u'$, $w = E[\mathrm{if}\ \bot\ \mathrm{then}\ u'\ \mathrm{else}\ v]$ with $u \rhd u'$.

Theorem 4 (Weak head expansion)**.** Let $E \in \mathrm{Elim}, t, u, v, w \in \Lambda$. Then:

- if $E[t[u/0]] \in \mathcal{SN}$ and $u \in \mathcal{SN}$, then $E[(\lambda\ t)\ u] \in \mathcal{SN}$.
- if $E[t] \in \mathcal{SN}$ and $u \in \mathcal{SN}$, then $E[\pi_1\ \langle t, u\rangle] \in \mathcal{SN}$.
- if $E[u] \in \mathcal{SN}$ and $t \in \mathcal{SN}$, then $E[\pi_2\ \langle t, u\rangle] \in \mathcal{SN}$.
- if $E[u[t/0]] \in \mathcal{SN}$ and $v \in \mathcal{SN}$, then $E[\delta(0 \mapsto u \mid 0 \mapsto v)\ (\kappa_1\ t)] \in \mathcal{SN}$.

- if $E[v[t/0]] \in \mathcal{SN}$ and $u \in \mathcal{SN}$, then $E[\delta(0 \mapsto u \mid 0 \mapsto v)\ (\kappa_2\ t)] \in \mathcal{SN}$.
- if $E[u] \in \mathcal{SN}$ and $v \in \mathcal{SN}$, then $E[\text{rec}\ u\ v\ \mathbf{0}] \in \mathcal{SN}$.
- if $E[v\ t\ (\text{rec}\ u\ v\ t)] \in \mathcal{SN}$ and $u \in \mathcal{SN}$, then $E[\text{rec}\ u\ v\ (\text{S}\ t)] \in \mathcal{SN}$.
- if $E[u] \in \mathcal{SN}$ and $v \in \mathcal{SN}$, then $E[\text{if}\ \top\ \text{then}\ u\ \text{else}\ v] \in \mathcal{SN}$.
- if $E[v] \in \mathcal{SN}$ and $u \in \mathcal{SN}$, then $E[\text{if}\ \bot\ \text{then}\ u\ \text{else}\ v] \in \mathcal{SN}$.

3 The strong normalization theorem

3.1 File 8

In this file, we define the function $[\![-]\!] : \text{Types} \to \mathcal{P}(\Lambda)$ which, to each type, will associate a set of terms. The purpose of the proof is to show that

$$\Gamma \vdash t : T \implies t \in [\![T]\!]$$

and that for all $T \in \text{Types}, [\![T]\!] \subseteq \mathcal{SN}$. With this, we directly deduce that each typed term is strongly normalizing.

The definition of $[\![-]\!]$ will be by induction on the structure of types, so we need for each type constructor to make a new set of terms. For this, we will use the elimination rule associated to each type. For example,

$$[\![T \to U]\!] := \{t \in \Lambda \mid \forall u \in [\![T]\!], t\ u \in [\![U]\!]\}$$

but to do this, we have two issues:

- when we want to construct such a definition for *e.g.* $T + U$, we need to quantify over all types, as the type of the eliminator if universally quantified over types. As we are defining the interpretation, we can't use $[\![T]\!]$ for any T in it. To avoid that, we define a predetermined set $\mathcal{SAT}_{\mathcal{SN}}$ of adequate parts of Λ, which we call here saturated sets, and for which we will assure that $[\![T]\!] \in \mathcal{SAT}_{\mathcal{SN}}$.
- to construct the definition of $[\![\texttt{int}]\!]$, we furthermore need to be able to call $[\![\texttt{int}]\!]$ itself in the definition. To do so, we will use Knaster-Tarski theorem: we can prove that $\mathcal{SAT}_{\mathcal{SN}}$ is a complete lattice, so each increasing function $f : \mathcal{SAT}_{\mathcal{SN}} \to \mathcal{SAT}_{\mathcal{SN}}$ has a fix point. Now, constructing the correct function, we can define $[\![\texttt{int}]\!]$ to be a fix point.

This is what motivates our choice of defining the set $\mathcal{SAT}_{\mathcal{SN}}$, but now we need to know what is needed for this set to work. The conditions are:

- saturated sets are parts of $\mathcal{SN}$, so $\forall A \in \mathcal{SAT}_{\mathcal{SN}}, A \subseteq \mathcal{SN}$.
- the set $\{E[n] \mid E \in \mathcal{SN}_e, n \in \mathbb{N}\}$ must be contained in all saturated sets.
- each saturated set has the saturation property, which is an analogous of the weak head expansion but with the property of being in the set.
- given a saturated set $A \in \mathcal{SAT}_{\mathcal{SN}}$ and $E \in \text{Elim}, t \in \Lambda, k, n \in \mathbb{N}$, if $E[t] \in A$ then $E[\uparrow_k^n t] \in A$.

The first condition is obvious given the structure of our proof. Now, the other three conditions all work inside of an elimination context. This is done because, when proving that $[\![T]\!] \in \mathcal{SAT}_{\mathcal{SN}}$ for all $T \in \text{Types}$, we precisely enlarge the contexts used in those definitions, because the definition of our interpretations will use eliminators. The second condition, then, just assures that each saturated set is nonempty, and to do so we use variables (thus, all contexts containing variables). The third condition will be used when we prove that $\Gamma \vdash t : T \implies t \in [\![T]\!]$ to ensure, when doing induction on $\Gamma \vdash t : T$, that the induction goes through when t is a constructor (when it is an eliminator, the definition of our interpretations directly give the result). The fourth condition is used when stating the adequacy theorem later, because we need to be able to lift a context to have simultaneous substitutions in our arguments.

Definition 23 (Saturated set)**.** We define the set $\mathcal{SAT}_{\mathcal{SN}} \subseteq \mathcal{P}(\Lambda)$ by the fact that $A \in \mathcal{SAT}_{\mathcal{SN}}$ if all the following conditions are met:

- $A \subseteq \mathcal{SN}$.
- for all $E \in \mathcal{SN}_e$ and $n \in \mathbb{N}$, $E[n] \in A$.
- for all $E \in \text{Elim}, t, u \in \Lambda$, if $E[t[u/0]] \in A$ and $u \in \mathcal{SN}$ then $E[(\lambda\ t)\ u] \in A$.

- for all $E \in \mathrm{Elim}, t, u \in \Lambda$, if $E[t] \in A$ and $u \in \mathcal{SN}$, then $E[\pi_1\ \langle t, u\rangle] \in A$.
- for all $E \in \mathrm{Elim}, t, u \in \Lambda$, if $E[u] \in A$ and $t \in \mathcal{SN}$, then $E[\pi_2\ \langle t, u\rangle] \in A$.
- for all $E \in \mathrm{Elim}, t, u, v \in \Lambda$, if $E[u[t/0]] \in A$ and $v \in \mathcal{SN}$, then $E[\delta(0 \mapsto u \mid 0 \mapsto v)\ (\kappa_1\ t)] \in A$.
- for all $E \in \mathrm{Elim}, t, u, v \in \Lambda$, if $E[v[t/0]] \in A$ and $u \in \mathcal{SN}$, then $E[\delta(0 \mapsto u \mid 0 \mapsto v)\ (\kappa_2\ t)] \in A$.
- for all $E \in \mathrm{Elim}, u, v \in \Lambda$, if $E[u] \in A$ and $v \in \mathcal{SN}$, then $E[\mathrm{rec}\ u\ v\ \mathbf{0}] \in A$.
- for all $E \in \mathrm{Elim}, t, u, v \in \Lambda$, if $E[v\ t\ (\mathrm{rec}\ u\ v\ t)] \in A$ and $u \in \mathcal{SN}$, then $E[\mathrm{rec}\ u\ v\ (\mathrm{S}\ t)] \in A$.
- for all $E \in \mathrm{Elim}, u, v \in \Lambda$, if $E[u] \in A$ and $v \in \mathcal{SN}$, then $E[\mathrm{if}\ \top\ \mathrm{then}\ u\ \mathrm{else}\ v] \in A$.
- for all $E \in \mathrm{Elim}, u, v \in \Lambda$, if $E[v] \in A$ and $u \in \mathcal{SN}$, then $E[\mathrm{if}\ \bot\ \mathrm{then}\ u\ \mathrm{else}\ v] \in A$.
- for all $E \in \mathrm{Elim}, t \in \Lambda k, n \in \mathbb{N}$, if $E[t] \in A$ then $E[\uparrow_k^n t] \in A$.

The first result is the stability by weak head expansion:

Proposition 3.1.1. If $A \in \mathcal{SAT}_{\mathcal{SN}}, t \in A, u \in \mathcal{SN}$ and $u \rhd_h^* t$ then $u \in A$.

Proposition 3.1.2 ($\mathcal{SAT}_{\mathcal{SN}}$ is a complete lattice)**.** The set $\mathcal{SAT}_{\mathcal{SN}}$ is a complete lattice with the order $\subseteq$, with as glb the usual intersection, as lub the usual union, as biggest element $\mathcal{SN}$ and as least element

$$\{t \in \mathcal{SN} \mid \exists E \in \mathcal{SN}_e, \exists n \in \mathbb{N}, t \rhd_h^* E[n]\}$$

After the proofs that $\mathcal{SAT}_{\mathcal{SN}}$ is indeed a complete lattice, we prove the Knaster-Tarski theorem in this particular complet lattice.

Theorem 5 (Knaster-Tarski)**.** Let $f : \mathcal{SAT}_{\mathcal{SN}} \to \mathcal{SAT}_{\mathcal{SN}}$ be a function such that

$$\forall A, B \in \mathcal{SAT}_{\mathcal{SN}}, A \subseteq B \implies f(A) \subseteq f(B)$$

then there exists $A \in \mathcal{SAT}_{\mathcal{SN}}$ such that $f(A) = A$.

To define the interpretation of `int`, we define the adequate function.

Definition 24. We define the function $f_{\mathbb{N}} : \mathcal{SAT}_{\mathcal{SN}} \to \mathcal{SAT}_{\mathcal{SN}}$ by

$$f_{\mathbb{N}}(X) = \{t \in \Lambda \mid \forall A \in \mathcal{SAT}_{\mathcal{SN}}, \forall u \in A, \forall v \in \Lambda, (\forall w \in X, \forall x \in A, v\ w\ x \in A) \implies \mathrm{rec}\ u\ v\ t \in A\}$$

Lemma 3.1.3. The function $f_{\mathbb{N}}$ is an increasing function from $\mathcal{SAT}_{\mathcal{SN}}$ to $\mathcal{SAT}_{\mathcal{SN}}$. It thus have a fix point $A_{f_{\mathbb{N}}}$ such that

$$\forall t \in \Lambda, t \in A_{f_{\mathbb{N}}} \iff \forall A \in \mathcal{SAT}_{\mathcal{SN}}, \forall u \in A, \forall v \in \Lambda, (\forall w \in A_{f_{\mathbb{N}}}, \forall x \in A, v\ w\ x \in A) \implies \mathrm{rec}\ u\ v\ t \in A$$

Now, we define the interpretation of the types.

Definition 25. For all $T \in \mathrm{Types}$, we define $[\![T]\!]$ as follows:

- $[\![\mathtt{unit}]\!] = \mathcal{SN}$
- $[\![\mathtt{void}]\!] = \{t \in \mathcal{SN} \mid \exists E \in \mathcal{SN}_e, \exists n \in \mathbb{N}, t \rhd_h^* E[n]\}$
- $[\![\mathtt{bool}]\!] = \{t \in \mathcal{SN} \mid \forall A \in \mathcal{SAT}_{\mathcal{SN}}, \forall u, v \in A, \mathrm{if}\ t\ \mathrm{then}\ u\ \mathrm{else}\ v \in A\}$
- $[\![\mathtt{int}]\!] = A_{f_{\mathbb{N}}}$
- $[\![T \to U]\!] = \{t \in \Lambda \mid \forall u \in [\![T]\!], t\ u \in [\![U]\!]\}$
- $[\![T \times U]\!] = \{t \in \Lambda \mid \pi_1\ t \in T \wedge \pi_2\ t \in U\}$
- $[\![T + U]\!] = \{t \in \Lambda \mid \forall A \in \mathcal{SAT}_{\mathcal{SN}}, \forall u, v \in \Lambda, (\forall w \in [\![T]\!], u[w/0] \in A) \wedge (\forall w \in [\![U]\!], v[w/0] \in A)$
  $\implies \delta(0 \mapsto u \mid 0 \mapsto v)\ t \in A\}$

Theorem 6 (Saturation)**.**

$$\forall T \in \mathrm{Types}, [\![T]\!] \in \mathcal{SAT}_{\mathcal{SN}}$$

3.2 File 9

The file 9 is the last one and describes the adequacy theorem and its consequence, the strong normalization theorem.

Definition 26 (Valid permutation)**.** For a permutation $\sigma \in \mathfrak{S}$ and a typing context Γ, we say that σ is valid with regard to Γ, written $\sigma \models \Gamma$, if $|\sigma| = |\Gamma|$ and for each $i \in \{0, \dots, |\sigma|\}$, $\sigma_i \in [\![\Gamma_i]\!]$.

Theorem 7 (Adequacy)**.** For all $t \in \Lambda, T \in \text{Types}, \Gamma \in C_T$, if $\sigma \models \Gamma$ and $\Gamma \vdash t : T$ then

$$t[\sigma/0] \in [\![T]\!]$$

But then, as $n \in A$ for all $n \in \mathbb{N}$ and $A \in \mathcal{SAT}_{\mathcal{SN}}$, this means that $\sigma_{|\Gamma|} \models \Gamma$ for all $\Gamma \in C_T$. But as we can easily prove that $\Gamma \vdash t : T \implies \max t \leq |\Gamma|$, we can show that $t[\sigma_{|\Gamma|}/0] \in [\![T]\!]$, which is exactly $t \in [\![T]\!]$. Hence the following theorem:

Theorem 8 (Strong normalization)**.** For all $t \in \Lambda, T \in \text{Types}, \Gamma \in C_T$, if $\Gamma \vdash t : T$ then $t \in \mathcal{SN}$.

References

[1] Jean-Yves Girard, Paul Taylor, and Yves Lafont. *Proofs and Types*. Cambridge University Press, Cambridge, April 1989.

[2] Morten Heine Sørensen and Pawel Urzyczyn. *Lectures on the Curry-Howard isomorphism*. Elsevier, Amsterdam; Oxford, 2006.