Introduction to Proof Theory

Gilles Dowek

École polytechnique and INRIA
LIX, École polytechnique
91128 Palaiseau Cedex, France
Gilles.Dowek@polytechnique.fr
http://www.lix.polytechnique.fr/~dowek

Contents

Introduction

There is something special about the mathematical discourse : each assertion must be justified by a proof. A proof is a sequence of assertions produced from the previous ones by deduction rules. The deduction rules are thus the "rules of the game" that mathematicians play.

Euclid's *Elements* (III$^{\text{rd}}$ century B.C.) are usually considered as the first systematic development where each assertion is given a proof, however, the precise definition of the notion of proof has only been formulated at the beginning of the XX$^{\text{th}}$ century. Having a definition, and not just an informal idea of what a correct proof is, is important in several areas. First, since the middle of the XX$^{\text{th}}$ century, proofs have been used not only by mathematicians, but also by computerized proof processing systems such as proof checkers and proof search systems, and designing such a system requires a precise definition.

Having a definition is also necessary to solve some problems about proofs. This is what proof theory is about. A first type of results proof theory permits to prove is independence results: results asserting that some proposition cannot be proved in some theory, for instance that the axiom of parallels cannot be proved from the other axioms of geometry.

However, proof theory is not concerned only with the provable propositions but also with the structure of proofs themselves, for instance with the comparison of different proofs of the same theorem. One key notion in proof theory is that of *canonical*, *direct* or *cut free* proof. For instance, if we first prove two propositions A and B, to deduce the proposition $A \wedge B$ (A *and* B) and at last the proposition A, we build a proof that is not canonical, because it contains an unnecessary detour by the proposition $A \wedge B$, that has nothing to do with the problem. Such a detour is called a *cut*. The main results we prove in these course notes are that in some cases, such cuts can be eliminated and thus that all provable propositions have canonical proofs. Moreover non canonical proofs can be transformed into canonical ones in an algorithmic way.

From a philosophical point of view, these results show that proving a theorem does not require to use ideas external to the statement of the theorem, or more precisely, that the use of such external ideas is only required in some specific cases, depending on the theory. Another application of cut elimination is that studying the structure of canonical proofs permits to show that some propositions have no canonical proofs. Hence, from the cut elimination theorem, we can

deduce that they have no proof at all. We get this way independence results. Cut elimination is also used to reduce dramatically the search space of proof search algorithms, by restricting to canonical proofs. Finally, cut elimination permits to prove the witness property for constructive proofs, *i.e.* that each time we have a proof of a special form of the existence of an object verifying a property P, there is also a mathematical object, called *a witness*, for which the property P can be proved to hold. Moreover, with the cut elimination algorithm, a description of this object can be computed from the proof. This allows to use mathematics as a programming language: the cut elimination process is the execution process of this programming language.

Very often, a proof is defined as a succession of reasonning steps starting from the axioms and ending at a conclusion. With such a definition, deduction rules are just reasonning rules. This definition hides the fact that, in mathematics, proofs are not only formed with reasonning steps but also with computation steps. *Deduction modulo* is a reformulation of the axiomatic method where reasonning and computation are both fully taken into account. We can, for instance, take advantage of this distinction between reasonning and computation when designing proof seach methods. More surprisingly, we can also take advantage of this distinction in proof theory. In particular, several cut elimination theorems can then be seen as corollaries of a single general cut elimination theorem for deduction modulo. Thus deduction modulo can be used as a unifying framework to present the basic results of proof theory. This is the point of view we have taken in these course notes.

Chapter 1

Predicate Logic

1.1 Languages

A *language* permits to designate things (The Moon, the number 2, the set of even numbers, ...) and to express facts (The Moon is a satellite of the Earth, the number 2 is a member of the set of even numbers, the set of even numbers is infinite, ...). A phrase that designates a thing is called a *term*, one that expresses a fact is called a *proposition*.

The easiest way to designate a thing is to use an *individual symbol* (also called a *proper name*) such as "2". Thus, a language contains individual symbols and individual symbols are terms. But, if we want to be able to designate an infinite number of objects with a finite number of symbols, we cannot give a proper name to each object. Thus, a language must contain an other kind of symbols, called *function symbols*. A function symbol alone is not a term, but it permits to construct a term when it is applied to already constructed terms. For instance, with the individual symbol 0 and the function symbol Su (for "successor") we can designate all the natural numbers. The number zero is designated by the term 0, the number one by the term $Su(0)$ obtained by applying the function symbol Su to the term 0, the number two by the term $Su(Su(0))$, ... Some function symbols must be applied to several arguments to construct a term, for instance the symbol + must be applied to two arguments. The function symbol + is said to have two arguments, while the symbol Su is said to have one argument. Individual symbols can be seen as special function symbols that have zero arguments.

The simplest way to form a proposition is to apply a *predicate symbol* to one or several terms. For instance, we can form this way the proposition

satellite(Moon,Earth)

that expresses that the Moon is a satellite of the Earth. Thus, a language contains predicate symbols. The predicate symbol *satellite* that must be applied to two terms to form a proposition is said to have two arguments. A proposition

formed by application of a predicate symbol to terms is called *atomic*. More propositions can be formed with the connectors $\neg$ (not), $\wedge$ (and), $\vee$ (or) and $\Rightarrow$ (implies). It is also convenient to consider propositions $\top$ (truth) and $\bot$ (falsity). We can for instance form this way the proposition

$$prime(Su(Su(0))) \wedge \neg prime(Su(Su(Su(Su(0)))))$$

that expresses that the number two is prime and that the number four is not.

A last construction is needed for propositions such as "all men are mortal" or "some number is prime", where we express that all objects verify some predicate or that some object verify some predicate without expliciting this object. We could introduce symbols *all* and *some* and let them replace a term as an argument of a predicate symbol or a function symbol. For instance we would write

$$prime(some)$$

to express that some number is prime, in the same way that we write

$$prime(Su(Su(0)))$$

to express that the number two is prime. But, such a construction is ambiguous. Indeed, the proposition

$$some \geq all$$

may express that for all numbers there is some greater number (which is true) but also that there is some number greater than all numbers (which is false). A more precise construction is to apply the predicate symbol to a variable and indicate in a second step if this variable is universal or existential with a quantifier $\forall$ (for all) or $\exists$ (there exists). The fact that some number is prime is then expressed

$$\exists x \; prime(x)$$

The order in which these quantifiers are applied permits to resolve the ambiguities. The fact that for all numbers there is some greater number is expressed by the proposition

$$\forall x \; \exists y \; y \geq x$$

while the fact that some number is greater than all numbers (which is false) is expressed by the proposition

$$\exists y \; \forall x \; y \geq x$$

Among all the symbols used to form terms and propositions, some are the same in all languages: the connectors $\top$, $\bot$, $\neg$, $\wedge$, $\vee$ and $\Rightarrow$, the quantifiers $\forall$ and $\exists$ and the variables, while the function symbols (including the individual symbols) and the predicate symbols are specific to a given language. For instance the symbol *Moon* is used in the language of astronomy, but not in the language of geometry.

1.1.1 Terms and propositions

Definition 1.1.1 (Language) *A* language *is a set of function symbols and a set of predicate symbols. To each symbol is associated a number, called its* number of arguments.

Definition 1.1.2 (Term) *Let $\mathcal{L}$ be a language and $\mathcal{V}$ be an infinite set whose elements are called* variables. *The* terms *of the language $\mathcal{L}$ with variables $\mathcal{V}$ are defined by the following rules*

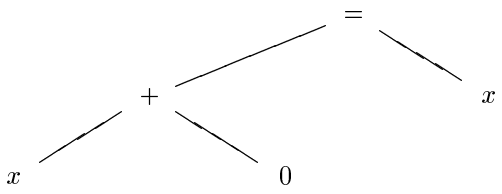
- *if x is a variable then the tree whose root is labeled by x and that has no sub-tree is a term,*
- *if f is a function symbol of n arguments and $t_1, ..., t_n$ are terms then the tree whose root is labeled by f and whose sub-trees are $t_1, ..., t_n$ is a term.*

Definition 1.1.3 (Proposition) *Let $\mathcal{L}$ be a language and $\mathcal{V}$ be an infinite set. The* propositions *of the language $\mathcal{L}$ with variables $\mathcal{V}$ are defined by the following rules*

- *if P is a predicate symbol of n arguments and $t_1, ..., t_n$ are terms then the tree whose root is labeled by P and whose sub-trees are $t_1, ..., t_n$ is a proposition,*
- *the trees whose root are labeled by $\top$ and $\bot$ and that have no sub-tree are propositions,*
- *if A is a proposition then the tree whose root is labeled by $\neg$ and whose sub-tree is A is a proposition,*
- *if A and B are propositions then the trees whose root are labeled by $\wedge$, $\vee$ or $\Rightarrow$ and whose sub-trees are A and B are propositions,*
- *if A is a proposition and x a variable then the trees whose root are labeled $\forall x$ and $\exists x$ and whose sub-tree is A are propositions.*

Remark. In several places, we shall use the notation $A \Leftrightarrow B$. There is no connector $\Leftrightarrow$ in our definition of the notion of proposition. Thus the proposition $A \Leftrightarrow B$ is just a notation for the proposition $(A \Rightarrow B) \wedge (B \Rightarrow A)$.

Example 1.1.1 *If $=$ is a predicate symbol of two arguments, $+$ a function symbol of two arguments, 0 a function symbol of zero arguments (*i.e. *an individual symbol) and x a variable then the tree*



is a proposition.

Remark. Terms and propositions have been defined as trees whose nodes are labeled by symbols. Some authors prefer to define terms and propositions as strings, *i.e.* as sequences of symbols. The proposition of example 1.1.1 would then be written

$$= (+(x, 0), x)$$

or

$$x + 0 = x$$

This is difference is just a matter of taste.

However, the advantage of considering trees instead of strings is that this permits to disregard the shallow properties of expressions: whether + is written before, between or after its arguments, whether parentheses or brackets are used, ... and to focus on the logical structure of expressions.

1.1.2 Variables and substitutions

Definition 1.1.4 (Variables) *The set of variables of a term (resp. proposition) is defined by induction over its height as follows*

- $Var(x) = \{x\}$,
- $Var(f(t_1, ..., t_n)) = Var(t_1) \cup ... \cup Var(t_n)$,
- $Var(P(t_1, ..., t_n)) = Var(t_1) \cup ... \cup Var(t_n)$,
- $Var(\top) = Var(\bot) = \emptyset$,
- $Var(\neg A) = Var(A)$,
- $Var(A \wedge B) = Var(A \vee B) = Var(A \Rightarrow B) = Var(A) \cup Var(B)$,
- $Var(\forall x\ A) = Var(\exists x\ A) = Var(A) \cup \{x\}$.

The set of free variables of a term (resp. a proposition) is defined by induction over its height as follows

- $FV(x) = \{x\}$,
- $FV(f(t_1, ..., t_n)) = FV(t_1) \cup ... \cup FV(t_n)$,
- $FV(P(t_1, ..., t_n)) = FV(t_1) \cup ... \cup FV(t_n)$,
- $FV(\top) = FV(\bot) = \emptyset$,
- $FV(\neg A) = FV(A)$,
- $FV(A \wedge B) = FV(A \vee B) = FV(A \Rightarrow B) = FV(A) \cup FV(B)$,
- $FV(\forall x\ A) = FV(\exists x\ A) = FV(A) \setminus \{x\}$.

Definition 1.1.5 (Closed and open) *A term (resp. a proposition) that contain no free variables is said to be* closed*, otherwise it is said to be* open*.*

We now want to define the operation of substitution. For instance, substituting the term $y+2$ for the variable x in the proposition $x \times 2 = 4$ yields the proposition $(y+2) \times 2 = 4$. The result of the substitution of the term u for the variable x in the term or proposition t is written $(u/x)t$. When we substitute a term u for a variable x in a term or a proposition t, we want to substitute only the free occurrences of x. A first attempt to define substitution is the following.

Definition 1.1.6 (Replacement) *Let t be a term (resp. a proposition), x be a variable and u be a term. The term (resp. the proposition) $\langle u/x\rangle t$ is defined by induction over the height of t as follows.*

- $\langle u/x\rangle x = u$,
  if y is a variable different from x, then $\langle u/x\rangle y = y$,
  $\langle u/x\rangle f(t_1, ..., t_n) = f(\langle u/x\rangle t_1, ..., \langle u/x\rangle t_n)$,
- $\langle u/x\rangle P(t_1, ..., t_n) = P(\langle u/x\rangle t_1, ..., \langle u/x\rangle t_n)$,
  $\langle u/x\rangle \top = \top$,
  $\langle u/x\rangle \bot = \bot$,
  $\langle u/x\rangle (\neg A) = \neg \langle u/x\rangle A$,
  $\langle u/x\rangle (A \wedge B) = \langle u/x\rangle A \wedge \langle u/x\rangle B$,
  $\langle u/x\rangle (A \vee B) = \langle u/x\rangle A \vee \langle u/x\rangle B$,
  $\langle u/x\rangle (A \Rightarrow B) = \langle u/x\rangle A \Rightarrow \langle u/x\rangle B$,
  $\langle u/x\rangle (\forall x\ A) = \forall x\ A$,
  $\langle u/x\rangle (\forall y\ A) = \forall y\ \langle u/x\rangle A$ *if* $y \neq x$,
  $\langle u/x\rangle (\exists x\ A) = \exists x\ A$,
  $\langle u/x\rangle (\exists y\ A) = \exists y\ \langle u/x\rangle A$ *if* $y \neq x$.

But there is still a problem with this definition : when we replace $y+0$ for x in $\forall y\ P(x, y)$ we obtain $\forall y\ P(y+0, y)$ where the variable y in $y+0$ is now quantified, while originally, this variable y had nothing to do with the variable y quantified in $\forall y\ P(x, y)$. To perform a correct substitution, we must first rename the variable y quantified in $\forall y\ P(x, y)$ to get, for instance, $\forall z\ P(x, z)$ and then substitute the variable x by $y+0$ to get $\forall z\ P(y+0, z)$. The choice of the variable z is arbitrary, and we could also have obtained $\forall w\ P(y+0, w)$.

Thus, to define the substitution operation, we must first define the equivalence of two propositions modulo bound variable renaming and define substitution on the quotient of the set of propositions modulo this relation.

Definition 1.1.7 (Alphabetic equivalence) *The alphabetic equivalence between propositions is defined as follows*

- *if A and B are atomic propositions then $A \sim B$ if and only if $A = B$,*
  $\top \sim \top$,
  $\bot \sim \bot$,
  $(\neg A) \sim (\neg A')$ *if and only if* $A \sim A'$,

$(A \wedge B) \sim (A' \wedge B')$ *if and only if* $A \sim A'$ *and* $B \sim B'$,
$(A \vee B) \sim (A' \vee B')$ *if and only if* $A \sim A'$ *and* $B \sim B'$,
$(A \Rightarrow B) \sim (A' \Rightarrow B')$ *if and only if* $A \sim A'$ *and* $B \sim B'$,
$(\forall x\ A) \sim (\forall y\ A')$ *if and only if for some variable* z *not appearing in* $\forall x\ A$ *nor in* $\forall y\ A'$ $\langle z/x\rangle A \sim \langle z/y\rangle A'$,
$(\exists x\ A) \sim (\exists y\ A')$ *if and only if for some variable* z *not appearing in* $\exists x\ A$ *nor in* $\exists y\ A'$ $\langle z/x\rangle A \sim \langle z/y\rangle A'$.

From now on, propositions will be considered up to alphabetic equivalence, *i.e.* we consider only classes of propositions modulo alphabetic equivalence. So the proposition $\forall x\ (0 \leq x)$ and $\forall y\ (0 \leq y)$ are equal.

Definition 1.1.8 (Substitution) *Let* t *be a term (resp. a proposition),* x *be a variable and* u *be a term. The term (resp. the proposition)* $(u/x)t$ *is defined by induction over the height of* t *as follows*

- $(u/x)x = u$,
  if y *is a variable different from* x, *then* $(u/x)y = y$,
  $(u/x)f(t_1, ..., t_n) = f((u/x)t_1, ..., (u/x)t_n)$,
- $(u/x)P(t_1, ..., t_n) = P((u/x)t_1, ..., (u/x)t_n)$,
  $(u/x)\top = \top$,
  $(u/x)\bot = \bot$,
  $(u/x)(\neg A) = \neg(u/x)A$,
  $(u/x)(A \wedge B) = (u/x)A \wedge (u/x)B$,
  $(u/x)(A \vee B) = (u/x)A \vee (u/x)B$,
  $(u/x)(A \Rightarrow B) = (u/x)A \Rightarrow (u/x)B$,
  $(u/x)(\forall y\ A) = \forall z\ (u/x)(z/y)A$ *where* z *is a variable not appearing in* $\forall y\ A$, *not appearing in* u *and distinct from* x,
  $(u/x)(\exists y\ A) = \exists z\ (u/x)(z/y)A$ *where* z *is a variable not appearing in* $\exists y\ A$, *not appearing in* u *and distinct from* x.

We can in the same way define *simultaneous substitution.*

Definition 1.1.9 (Simultaneous substitution) *Let* t *be a term (resp. a proposition),* $x_1, ..., x_n$ *be variables and* $u_1, ..., u_n$ *be terms. Let* σ *be the finite function mapping* x_i *to* u_i. *The term (resp. the proposition)* σt *is defined by induction over the height of* t *as follows*

- $\sigma x_i = u_i$,
  if y *is a variable different from the* x_i*'s, then* $\sigma y = y$,
  $\sigma f(t_1, ..., t_n) = f(\sigma t_1, ..., \sigma t_n)$,
- $\sigma P(t_1, ..., t_n) = P(\sigma t_1, ..., \sigma t_n)$,
  $\sigma\top = \top$,
  $\sigma\bot = \bot$,
  $\sigma(\neg A) = \neg\sigma A$,
  $\sigma(A \wedge B) = \sigma A \wedge \sigma B$,

$\sigma(A \vee B) = \sigma A \vee \sigma B$,
$\sigma(A \Rightarrow B) = \sigma A \Rightarrow \sigma B$,
$\sigma(\forall y\ A) = \forall z\ \sigma(z/y)A$ *where* z *is a variable not appearing in* $\forall y\ A$ *and not appearing in* σ,
$\sigma(\exists y\ A) = \exists z\ \sigma(z/y)A$ *where* z *is a variable not appearing in* $\exists y\ A$ *and not appearing in* σ.

1.2 Proofs

We are now ready to define the tools that permit to prove propositions.

1.2.1 Proofs *à la* Hilbert

Definition 1.2.1 (Theory) *A* theory *is a set of propositions, called* axioms, *such that the membership of some proposition to this set can be decided in an algorithmic way.*

Definition 1.2.2 (Deduction rule) *A* Deduction rule *is a set of* $n+1$*-uples of propositions, such that the membership of some* $n+1$*-uples of propositions to this set can be decided in an algorithmic way. The* $n+1$*-uple* $\langle A_1, ..., A_n, B\rangle$ *is written*

$$\frac{A_1\ \ ...\ \ A_n}{B}$$

The propositions $A_1, ..., A_n$ *are called* the premises *and the proposition* B the conclusion *of the* $n+1$*-uple.*

Definition 1.2.3 (Proof) *Let* D *a set of deduction rules. A* proof *of a proposition* B *in* D *is a tree whose root is labeled by the proposition* B*, whose sub-trees are proofs of propositions* $A_1, ..., A_n$ *and such that the* $n+1$*-uple*

$$\frac{A_1\ \ ...\ \ A_n}{B}$$

is an element of one of the deduction rules of D.

Definition 1.2.4 (Logical axioms) *A* logical axiom *is a proposition of the following form where* A, B, C *are arbitrary propositions and* x *an arbitrary variable.*

$$A \Rightarrow (B \Rightarrow A)$$

$$(A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))$$

$$(\forall x\ (A \Rightarrow B)) \Rightarrow (A \Rightarrow \forall x\ B) \quad \textit{(if } x \notin FV(A)\textit{)}$$

$$\top$$

$$\bot \Rightarrow A$$

$$A \Rightarrow (\neg A \Rightarrow \bot)$$

$$(A \Rightarrow \bot) \Rightarrow \neg A$$

$$(A \wedge B) \Rightarrow A$$

$$(A \wedge B) \Rightarrow B$$

$$A \Rightarrow B \Rightarrow (A \wedge B)$$

$$A \Rightarrow (A \vee B)$$

$$B \Rightarrow (A \vee B)$$

$$(A \vee B) \Rightarrow ((A \Rightarrow C) \Rightarrow ((B \Rightarrow C) \Rightarrow C))$$

$$\forall x\ A \Rightarrow (t/x)A$$

$$(t/x)A \Rightarrow \exists x\ A$$

$$\exists x\ A \Rightarrow ((\forall x\ (A \Rightarrow B)) \Rightarrow B) \quad \textit{(if } x \notin FV(B)\textit{)}$$

$$A \vee \neg A$$

Definition 1.2.5 (Deduction rules *à la* Hilbert) *Given a theory* Γ*, the deduction rules* à la *Hilbert for* Γ *are the following:*

- *the rule* Axiom *containing all the* 1*-uples*

$$\frac{}{A}$$

  where A *is an element of* Γ *or a logical axiom,*

- *the rule* Modus ponens *containing all the* 3*-uples*

$$\frac{A \Rightarrow B \quad A}{B}$$

- *the rule* Generalization *containing all the* 2*-uples*

$$\frac{A}{\forall x\ A}$$

  where x *does not appear free in* Γ*.*

These rules should be understood as follows: axioms have trivial proofs, if we have already proved $A \Rightarrow B$ and A we can deduce B, if we have already proved A with no assumption on x, we can deduce $\forall x\ A$.

Example 1.2.1 *Consider the language formed with the four proposition symbols (*i.e. *predicate symbol of zero arguments) P, Q, R and S. Consider the theory formed with the propositions*

$$P$$

$$Q$$

$$Q \Rightarrow R$$

$$P \Rightarrow (R \Rightarrow S)$$

we have the following proof of the proposition S

$$\dfrac{\dfrac{\dfrac{}{P \Rightarrow (R \Rightarrow S)}\ \textit{Axiom} \quad \dfrac{}{P}\ \textit{Axiom}}{R \Rightarrow S}\ \textit{Modus ponens} \qquad \dfrac{\dfrac{}{Q \Rightarrow R}\ \textit{Axiom} \quad \dfrac{}{Q}\ \textit{Axiom}}{R}\ \textit{Modus ponens}}{S}\ \textit{Modus ponens}$$

Remark. Some authors prefer to define proofs as sequences of propositions rather than as trees. Again, this is just a matter of taste.

1.2.2 The deduction lemma

We now want to prove that a proposition $A \Rightarrow B$ has a proof in the theory Γ if and only if the proposition B has a proof in the theory Γ, A.

Proposition 1.2.1 *Let A be a proposition, the proposition $A \Rightarrow A$ has a proof in the empty theory.*

Proof. The propositions

$$(A \Rightarrow ((A \Rightarrow A) \Rightarrow A)) \Rightarrow ((A \Rightarrow (A \Rightarrow A)) \Rightarrow (A \Rightarrow A))$$

$$A \Rightarrow ((A \Rightarrow A) \Rightarrow A)$$

$$A \Rightarrow (A \Rightarrow A)$$

are logical axioms. Hence, the proposition $A \Rightarrow A$ has the proof

$$\dfrac{\dfrac{B \qquad A \Rightarrow ((A \Rightarrow A) \Rightarrow A)}{(A \Rightarrow (A \Rightarrow A)) \Rightarrow (A \Rightarrow A)}\ \text{Modus ponens} \qquad A \Rightarrow (A \Rightarrow A)}{A \Rightarrow A}\ \text{Modus ponens}$$

where B is $(A \Rightarrow ((A \Rightarrow A) \Rightarrow A)) \Rightarrow ((A \Rightarrow (A \Rightarrow A)) \Rightarrow (A \Rightarrow A))$.

Proposition 1.2.2 (Deduction lemma) *The proposition $A \Rightarrow B$ has a proof in the theory Γ is and only if the proposition B has a proof in the theory Γ, A.*

Proof. If the proposition $A \Rightarrow B$ has a proof in the theory Γ, then it has a proof in the theory Γ, A. So does the proposition A. Thus, the proposition B has a proof built with the *Modus ponens* rule.

Conversely, we prove by induction over the height of the proof of B in Γ, A that there is a proof of $A \Rightarrow B$ in Γ.

- If the root of the proof is a *Axiom*, then either $B = A$ and we have a proof of $A \Rightarrow B$ by the proposition 1.2.1, or B an element of Γ and we have the proof

$$\frac{B \Rightarrow (A \Rightarrow B) \quad B}{A \Rightarrow B}\,\text{Modus ponens}$$

- If the root of the proof is a *Modus ponens* then B is deduced from $C \Rightarrow B$ and C, that have smaller proofs. By induction hypothesis, there are proofs π_1 and π_2 of $A \Rightarrow (C \Rightarrow B)$ and $A \Rightarrow C$ in Γ and we take the proof

$$\dfrac{\dfrac{(A \Rightarrow (C \Rightarrow B)) \Rightarrow ((A \Rightarrow C) \Rightarrow (A \Rightarrow B)) \quad \overline{A \Rightarrow (C \Rightarrow B)}^{\,\pi_1}}{(A \Rightarrow C) \Rightarrow (A \Rightarrow B)}\,\text{Modus p.} \quad \overline{A \Rightarrow C}^{\,\pi_2}}{A \Rightarrow B}\,\text{Modus p.}$$

- If the root of the proof is a *Generalization* then we have $B = \forall x\ C$, x does not appear in Γ nor in A and C has a smaller proof. By induction hypothesis, there is a proof π of $A \Rightarrow C$ in Γ and we take the proof

$$\dfrac{(\forall x\ (A \Rightarrow C)) \Rightarrow (A \Rightarrow \forall x\ C) \quad \dfrac{\overline{A \Rightarrow C}^{\,\pi}}{\forall x\ (A \Rightarrow C)}\,\text{Generalization}}{A \Rightarrow \forall x\ C}\,\text{Modus ponens}$$

1.2.3 Natural deduction

Introducing an hypothesis seems to be a natural step in a proof. To prove, for instance, the proposition $(n = 0) \Rightarrow (n + 1 = 1)$ we want to assume that $n = 0$ and then to prove that $n + 1 = 1$.

Proofs *à la* Hilbert do not permit to do that directly: if we have a proof of the proposition $n + 1 = 1$ using the hypothesis $n = 0$, the deduction lemma permits to transform this proof into one of the proposition $(n = 0) \Rightarrow (n + 1 = 1)$, but this proof is much longer than the proof we started with and it is not very natural.

Natural deduction is an alternative definition of the notion of proof where the introduction of an hypothesis is deduction rule. In Natural deduction, a deduction step can modify not only the proved proposition but also the theory Γ, hence a proof is not a tree of propositions, but a tree of ordered pairs $\langle \Gamma, A \rangle$ where Γ is a theory and A a proposition. Such an ordered pair is called a *sequent*

and is written $\Gamma \vdash A$ (read "Γ *entails* A"). The *Introduction* rule that permits to introduce an hypothesis transforms the sequent $\Gamma, A \vdash B$ into the sequent $\Gamma \vdash A \Rightarrow B$.

The notions of deduction rule and proof adapt straightforwardly to sequents.

Definition 1.2.6 (Deduction rule on sequents) *A* Deduction rule *is a set of $n+1$-uples of sequents, such that the membership of some $n+1$-uples of sequents to this set can be decided in an algorithmic way. The $n+1$-uple $\langle \Gamma_1 \vdash A_1, ..., \Gamma_n \vdash A_n, \Delta \vdash B \rangle$ is written*

$$\frac{\Gamma_1 \vdash A_1 \ ... \ \Gamma_n \vdash A_n}{\Delta \vdash B}$$

The sequents $\Gamma_1 \vdash A_1, ..., \Gamma_n \vdash A_n$ are called the premises *and the sequent $\Delta \vdash B$* the conclusion *of the $n+1$-uple.*

Definition 1.2.7 (Proof on sequents) *Let D a set of deduction rules. A* proof *of a sequent $\Delta \vdash B$ in D is a tree whose root is labeled by the sequent $\Delta \vdash B$, whose sub-trees are proofs of sequents $\Gamma_1 \vdash A_1, ..., \Gamma_n \vdash A_n$ and such that the $n+1$-uple*

$$\frac{\Gamma_1 \vdash A_1 \ ... \ \Gamma_n \vdash A_n}{\Delta \vdash B}$$

is an element of one of the deduction rule of D.

With the introduction rule, the three first logical axioms are now redundant, indeed the sequent $\Gamma \vdash A \Rightarrow (B \Rightarrow A)$ can be proved as follows

$$\dfrac{\dfrac{\Gamma, A, B \vdash A}{\Gamma, A \vdash B \Rightarrow A}\,\text{Intro}}{\Gamma \vdash A \Rightarrow (B \Rightarrow A)}\,\text{Intro}$$

The sequent $\Gamma \vdash (A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))$ can be proved as follows

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Delta \vdash A \Rightarrow (B \Rightarrow C) \quad \Delta \vdash A}{\Delta \vdash B \Rightarrow C}\,\text{Modus p.} \quad \dfrac{\Delta \vdash A \Rightarrow B \quad \Delta \vdash A}{\Delta \vdash B}\,\text{Modus p.}}{\Gamma, A \Rightarrow (B \Rightarrow C), A \Rightarrow B, A \vdash C}\,\text{Intro}}{\Gamma, A \Rightarrow (B \Rightarrow C), A \Rightarrow B \vdash A \Rightarrow C}\,\text{Intro}}{\Gamma, A \Rightarrow (B \Rightarrow C) \vdash (A \Rightarrow B) \Rightarrow (A \Rightarrow C)}\,\text{Intro}}{\Gamma \vdash (A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))}\,\text{Intro}$$

where $\Delta = \Gamma, A \Rightarrow (B \Rightarrow C), A \Rightarrow B, A$. And, if the variable x appears free neither in Γ nor in A, the sequent $\Gamma \vdash (\forall x \ (A \Rightarrow B)) \Rightarrow (A \Rightarrow \forall x \ B)$ can be proved as follows

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Delta \vdash (\forall x \ (A \Rightarrow B)) \Rightarrow (A \Rightarrow B) \quad \Delta \vdash \forall x \ (A \Rightarrow B)}{\Delta \vdash A \Rightarrow B}\,\text{Modus p.} \quad \Delta \vdash A}{\Delta \vdash B}\,\text{Modus p.}}{\Gamma, \forall x \ (A \Rightarrow B), A \vdash \forall x \ B}\,\text{Generalization}}{\Gamma, \forall x \ (A \Rightarrow B) \vdash A \Rightarrow \forall x \ B}\,\text{Intro}}{\Gamma \vdash (\forall x \ (A \Rightarrow B)) \Rightarrow (A \Rightarrow \forall x \ B)}\,\text{Intro}$$

where $\Delta = \Gamma, \forall x \ (A \Rightarrow B), A$.

Using proof *à la* Hilbert, when we have proved the propositions A and B and we want to deduce the proposition $A \wedge B$, we must use the logical axiom $A \Rightarrow (B \Rightarrow (A \wedge B))$ and deduce $B \Rightarrow (A \wedge B)$ and then $A \wedge B$ with the *Modus ponens* rule. It is more natural to take a rule allowing to deduce directly $\Gamma \vdash A \wedge B$ from $\Gamma \vdash A$ and $\Gamma \vdash B$. As we have the rule *Introduction* this logical axiom and this rule are equivalent. As we have just seen, in a system where we have the logical axiom, we can simulate any instance of the rule and conversely, in a system where we have the rule, the axiom can be proved as follows

$$\dfrac{\dfrac{\dfrac{\Gamma, A, B \vdash A \quad \Gamma, A, B \vdash B}{\Gamma, A, B \vdash A \wedge B}\text{ New rule}}{\Gamma, A \vdash B \Rightarrow (A \wedge B)}\text{ Intro}}{\Gamma \vdash A \Rightarrow (B \Rightarrow (A \wedge B))}\text{ Intro}$$

Excercise 1.2.1 *With proof à la Hilbert, are the logical axiom and the rule equivalent ? Hint: try to prove the Deduction lemma.*

We can suppress in a similar way all the logical axioms and replace them by deduction rules. Let us take another example. The logical axiom

$$(A \vee B) \Rightarrow ((A \Rightarrow C) \Rightarrow ((B \Rightarrow C) \Rightarrow C))$$

can be replaced by the rule

$$\dfrac{\Gamma \vdash A \vee B \quad \Gamma \vdash A \Rightarrow C \quad \Gamma \vdash B \Rightarrow C}{\Gamma \vdash C}$$

But, as it is equivalent to prove the sequent $\Gamma \vdash A \Rightarrow C$ or the sequent $\Gamma, A \vdash C$ we can transform this rule further into

$$\dfrac{\Gamma \vdash A \vee B \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C}$$

In this rule, $\vee$ is the only connector or quantifier that appears explicitly. In most rules, only one connector or quantifier occurs. This permits to classify the rules according to the connector or quantifier that appears in this rule. The rules of a connector or quantifier can further be classified according to the position of this connector or quantifier. If it appears in the conclusion of the rule, then the rules is called an *introduction rule*, if it appears in a premise, then the rule is an *elimination rule*. For instance, the connector $\vee$ has two introduction rules

$$\dfrac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\ \vee\text{-intro}$$

$$\dfrac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\ \vee\text{-intro}$$

and one elimination rule

$$\dfrac{\Gamma \vdash A \vee B \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C}\ \vee\text{-elim}$$

The *Modus ponens*

$$\frac{\Gamma \vdash A \Rightarrow B \quad \Gamma \vdash A}{\Gamma \vdash B}$$

is the elimination rule of implication. The *Generalization*

$$\frac{\Gamma \vdash A}{\Gamma \vdash \forall x\ A} \text{ if } x \notin FV(\Gamma)$$

is the introduction rule of the universal quantifier $\forall$. And the rule *Introduction*

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}$$

is the introduction rule of the implication.

The system obtained this way is called *Natural Deduction*.

Definition 1.2.8 (Natural deduction)

$$\frac{}{\Gamma \vdash A}\ Axiom \quad if\ A \in \Gamma$$

$$\frac{}{\Gamma \vdash \top}\ \top\text{-}intro$$

$$\frac{\Gamma \vdash \bot}{\Gamma \vdash A}\ \bot\text{-}elim$$

$$\frac{\Gamma, A \vdash \bot}{\Gamma \vdash \neg A}\ \neg\text{-}intro$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash \neg A}{\Gamma \vdash \bot}\ \neg\text{-}elim$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\ \wedge\text{-}intro$$

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}\ \wedge\text{-}elim$$

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}\ \wedge\text{-}elim$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\ \vee\text{-}intro$$

$$\frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\ \vee\text{-}intro$$

$$\frac{\Gamma \vdash A \vee B \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C}\ \vee\text{-}elim$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}\ \Rightarrow\text{-}intro$$

$$\frac{\Gamma \vdash A \Rightarrow B \quad \Gamma \vdash A}{\Gamma \vdash B} \Rightarrow\text{-}elim$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash \forall x\ A} \forall\text{-}intro \quad \text{if } x \notin FV(\Gamma)$$

$$\frac{\Gamma \vdash \forall x\ A}{\Gamma \vdash (t/x)A} \forall\text{-}elim$$

$$\frac{\Gamma \vdash (t/x)A}{\Gamma \vdash \exists x\ A} \exists\text{-}intro$$

$$\frac{\Gamma \vdash \exists x\ A \quad \Gamma, A \vdash B}{\Gamma \vdash B} \exists\text{-}elim \quad \text{if } x \notin FV(\Gamma, B)$$

$$\frac{}{\Gamma \vdash A \vee \neg A}\ \textit{Excluded middle}$$

Proposition 1.2.3 *A proposition A has a proof* à la *Hilbert in the theory Γ if and only if the sequent $\Gamma \vdash A$ has a proof in natural deduction.*

Proof. By induction on the height of proofs.

Definition 1.2.9 (Contradictory, consistent) *A theory Γ is* contradictory *if all propositions have a proof in Γ. It is* consistent *otherwise.*

Excercise 1.2.2 *Prove that a theory Γ is contradictory if and only the proposition $\bot$ has a proof. Prove that a theory Γ is contradictory if and only there is a proposition A such that A and $\neg A$ have a proof.*

Excercise 1.2.3 *Let A be a proposition, prove that a theory that proves the proposition $A \Leftrightarrow \neg A$ is contradictory.*

Example 1.2.2 (Equality) *Given a language $\mathcal{L}$ containing a predicate symbol $=$ of two arguments, the theory of equality in this language is formed with the following axioms.*
Identity axiom:

$$\forall x\ (x = x)$$

Leibniz' axiom scheme: for each proposition A, the axiom

$$\forall x\ \forall y\ ((x = y) \Rightarrow ((x/z)A \Rightarrow (y/z)A))$$

Excercise 1.2.4 *In the theory of equality, give a proof of the proposition*

$$\forall x\ \forall y\ (x = y \Rightarrow y = x)$$

Example 1.2.3 (Arithmetic) *The language of arithmetic is formed with*

- *an individual symbol 0, a function symbol Su of one argument and two function symbols $+$ and $\times$ of two arguments*

- *a predicate symbol* $=$ *of two arguments.*

The axioms of arithmetic are the axioms of equality and the axioms:

$$\forall x\ \forall y\ (Su(x) = Su(y) \Rightarrow x = y)$$

$$\forall x\ \neg(0 = Su(x))$$

induction scheme: for each proposition A *the axiom*

$$((0/z)A \wedge (\forall x\ ((x/z)A \Rightarrow (Su(x)/z)A))) \Rightarrow \forall y\ (y/z)A$$

and the axioms

$$\forall y\ (0 + y = y)$$

$$\forall x\ \forall y\ (Su(x) + y = Su(x + y))$$

$$\forall y\ (0 \times y = 0)$$

$$\forall x\ \forall y\ (Su(x) \times y = (x \times y) + y)$$

Excercise 1.2.5 *Write a proof in arithmetic of the propositions*

$$Su(0) + Su(0) = Su(Su(0))$$

$$\forall x\ (x + 0 = x)$$

1.2.4 Constructive proofs

Definition 1.2.10 (Constructive proof) *A proof is* constructive *if it does not use the* excluded middle *rule.*

We want to prove that constructive provability and general provability are equivalent. This does not mean, of course, that all propositions that have a proof have a constructive proof, but that for each proposition A we can compute a proposition A' such that the proposition A has a proof if and only if the proposition A' has a constructive proof.

Definition 1.2.11 (Negative translation) *Let* A *be a proposition, the proposition* A' *is defined by induction over the height of* A *as follows.*

- $A' = \neg\neg A$ *if* A *is atomic,*
- $\top' = \neg\neg\top$,
- $\bot' = \neg\neg\bot$,
- $(\neg A)' = \neg\neg\neg A'$,
- $(A \wedge B)' = \neg\neg(A' \wedge B')$,
- $(A \vee B)' = \neg\neg(A' \vee B')$,

- $(A \Rightarrow B)' = \neg\neg(A' \Rightarrow B')$,
- $(\forall x\ A)' = \neg\neg(\forall x\ A')$,
- $(\exists x\ A)' = \neg\neg(\exists x\ A')$.

Proposition 1.2.4 *The proposition A has a proof if and only if A' has a constructive proof.*

Proof. (1) If a sequent $\Gamma \vdash A$ has a constructive proof π, then the sequent $\Gamma \vdash \neg\neg A$ has a constructive proof. First, we can add the hypothesis $\neg A$ to all sequents of the proof π, we obtain a proof π' of the sequent $\Gamma, \neg A \vdash A$. Then we have the following proof.

$$\dfrac{\dfrac{\Gamma, \neg A \vdash \neg A \qquad \overset{\pi'}{\overline{\Gamma, \neg A \vdash A}}}{\Gamma, \neg A \vdash \bot}\,\neg\text{-elim}}{\Gamma \vdash \neg\neg A}\,\neg\text{-intro}$$

Thus, we can build a constructive proof of $\neg\neg\top$. From a constructive proof of $\Gamma, A \vdash \bot$ we can build a constructive proof of $\Gamma \vdash \neg\neg\neg A$. From constructive proofs of $\Gamma \vdash A$ and $\Gamma \vdash B$, we can build a constructive proof of $\Gamma \vdash \neg\neg(A \wedge B)$. From a constructive proof of $\Gamma \vdash A$, we can build a constructive proof of $\Gamma \vdash \neg\neg(A \vee B)$. From a constructive proof of $\Gamma \vdash B$, we can build a constructive proof of $\Gamma \vdash \neg\neg(A \vee B)$. From a constructive proof of $\Gamma, A \vdash B$, we can build a constructive proof of $\Gamma \vdash \neg\neg(A \Rightarrow B)$. From a constructive proof of $\Gamma \vdash A$, we can build a constructive proof of $\Gamma \vdash \neg\neg\forall x\ A$ provided x does not appear free in Γ. From a constructive proof of $\Gamma \vdash (t/x)A$, we can build a constructive proof of $\Gamma \vdash \neg\neg\exists x\ A$.

(2) Then, we check that from a constructive proofs of $\Gamma \vdash \neg\neg\bot$, we can build a constructive proof of $\Gamma \vdash \neg\neg A$. From constructive proofs of $\Gamma \vdash \neg\neg\neg A$ and $\Gamma \vdash \neg\neg A$, we can build a constructive proof of $\Gamma \vdash \neg\neg\bot$. From a constructive proof of $\Gamma \vdash \neg\neg(\neg\neg A \wedge \neg\neg B)$, we can build a constructive proof of $\Gamma \vdash \neg\neg A$ and a constructive proof of $\Gamma \vdash \neg\neg B$. From a constructive proofs of $\Gamma \vdash \neg\neg(\neg\neg A \vee \neg\neg B)$, $\Gamma, \neg\neg A \vdash \neg\neg C$ and $\Gamma, \neg\neg B \vdash \neg\neg C$ we can build a constructive proof of $\Gamma \vdash \neg\neg C$. From constructive proofs of $\Gamma \vdash \neg\neg(\neg\neg A \Rightarrow \neg\neg B)$ and $\Gamma \vdash \neg\neg A$, we can build a constructive proof of $\Gamma \vdash \neg\neg B$. From constructive proofs of $\Gamma \vdash \neg\neg(\forall x\ \neg\neg A)$, we can build a constructive proof of $\Gamma \vdash \neg\neg(t/x)A$. From constructive proofs of $\Gamma \vdash \neg\neg\exists x\ A$ and $\Gamma, \neg\neg A \vdash \neg\neg B$ we can build a constructive proof of $\Gamma \vdash \neg\neg B$ provided that x does not appear free in Γ nor in B.

As an example we show that from constructive proofs of $\Gamma \vdash \neg\neg(\neg\neg A \Rightarrow \neg\neg B)$ and $\Gamma \vdash \neg\neg A$, we can build a constructive proof of $\Gamma \vdash \neg\neg B$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma, \neg B, \neg\neg A \Rightarrow \neg\neg B \vdash \neg\neg A \Rightarrow \neg\neg B \qquad \overset{\pi'}{\overline{\Gamma, \neg B, \neg\neg A \Rightarrow \neg\neg B \vdash \neg\neg A}}}{\Gamma, \neg B, \neg\neg A \Rightarrow \neg\neg B \vdash \neg\neg B}\,\Rightarrow\text{-elim} \qquad \Gamma, \neg B, \neg\neg A \Rightarrow \neg\neg B \vdash \neg B}{\dfrac{\overset{\pi}{\overline{\Gamma, \neg B \vdash \neg\neg(\neg\neg A \Rightarrow \neg\neg B)}} \qquad \dfrac{\Gamma, \neg B, \neg\neg A \Rightarrow \neg\neg B \vdash \bot}{\Gamma, \neg B \vdash \neg(\neg\neg A \Rightarrow \neg\neg B)}\,\neg\text{-intro}}{\Gamma, \neg B \vdash \bot}\,\neg\text{-elim}}\,\neg\text{-elim}}{\Gamma \vdash \neg\neg B}\,\neg\text{-intro}$$

(3) We check that if A is a proposition, then the proposition $\neg\neg(A \vee \neg A)$ has a constructive proof.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\neg(A \vee \neg A), A \vdash A}}{\neg(A \vee \neg A), A \vdash \neg(A \vee \neg A)} \quad \dfrac{}{\neg(A \vee \neg A), A \vdash A \vee \neg A}}{\neg(A \vee \neg A), A \vdash \bot}\,\neg\text{-e.}}{\neg(A \vee \neg A) \vdash \neg A}\,\neg\text{-intro} \quad \dfrac{}{\neg(A \vee \neg A) \vdash A \vee \neg A}}{\neg(A \vee \neg A) \vdash \bot}\,\neg\text{-elim}}{\vdash \neg\neg(A \vee \neg A)}\,\neg\text{-intro}$$

(4) Then, we show that if $\Gamma \vdash A$ has a proof π then $\Gamma' \vdash A'$ has a constructive proof, by induction over the height of π. If the last rule of π is an axiom then we use the axiom rule, if the last rule is an introduction rule then we use lemma (1), if it is an elimination rule then we use lemma (2), if it the excluded middle rule, we use lemma (3).

(5) Conversely, we show that the proposition $A \Leftrightarrow \neg\neg A$ has a (not necessarily constructive) proof and we deduce that $A \Leftrightarrow A'$ has a (non necessarily constructive) proof and that if $\Gamma' \vdash A'$ has a constructive proof then $\Gamma \vdash A$ has a (not necessarily constructive) proof.

Remark. In these course notes, we shall mainly focus on constructive proofs. This does not mean that we renounce the non constructive proofs, but that non constructive proofs of a proposition A are understood as constructive proofs of its negative translation.

1.3 Models

Definition 1.3.1 (Structure) *Let $\mathcal{L}$ be a language formed with the function symbols $f_0, f_1, \ldots$ of number or arguments $n_0, n_1, \ldots$ and the predicate symbols $P_0, P_1, \ldots$ of number of arguments $m_0, m_1, \ldots$. A* structure *$\mathcal{M}$ built on $\mathcal{L}$ is a n-uple formed with*

- *a non empty set M,*
- *a function $\hat{f}_0$ from M^{n_0} to M, a function $\hat{f}_1$ from M^{n_1} to M, ...*
- *a function $\hat{P}_0$ from M^{m_0} to $\{0,1\}$, a function $\hat{P}_1$ from M^{m_1} to $\{0,1\}$, ...*

Definition 1.3.2 (Assignment) *An* assignment *over the set of variables $\mathcal{V}$ is a function from $\mathcal{V}$ to M. If ϕ is an assignment, x a variable and a an element of M, then $\phi + \langle x, a\rangle$ is the assignment mapping x to a and y to $\phi(y)$ when y is distinct from x.*

Definition 1.3.3 (Denotation) *Let $\mathcal{L}$ be a language, $\mathcal{V}$ be a set of variables and $\mathcal{M}$ be a structure built on $\mathcal{L}$. Let ϕ be an assignment and t be a term (resp. a proposition), the denotation of t in $\mathcal{M}$ modulo ϕ is defined by induction over the height of t.*

- $|x|_\phi = \phi(x)$,
  $|f_i(t_1, ..., t_{n_i})|_\phi = \hat{f}_i(|t_1|_\phi, ..., |t_{n_i}|_\phi)$,

- $|P_i(t_1, ..., t_{n_i})|_\phi = \hat{P}_i(|t_1|_\phi, ..., |t_{n_i}|_\phi)$,
  $|\top|_\phi = 1$,
  $|\bot|_\phi = 0$,
  $|\neg A|_\phi = 1$ *if* $|A|_\phi = 0$, *and* 0 *otherwise,*
  $|A \wedge B|_\phi = 1$ *if* $|A|_\phi = 1$ *and* $|B|_\phi = 1$, *and* 0 *otherwise,*
  $|A \vee B|_\phi = 1$ *if* $|A|_\phi = 1$ *or* $|B|_\phi = 1$, *and* 0 *otherwise,*
  $|A \Rightarrow B|_\phi = 1$ *if* $|A|_\phi = 0$ *or* $|B|_\phi = 1$, *and* 0 *otherwise,*
  $|\forall x\ A|_\phi = 1$ *if for all elements* a *of* M, $|A|_{\phi + \langle x,a \rangle} = 1$, *and* 0 *otherwise*
  $|\exists x\ A|_\phi = 1$ *if there is an element* a *of* M *such that* $|A|_{\phi + \langle x,a \rangle} = 1$, *and* 0 *otherwise.*

Definition 1.3.4 (Validity, model) *Let* $\mathcal{L}$ *be a language,* $\mathcal{V}$ *be a set of variables and* $\mathcal{M}$ *be a structure built on* $\mathcal{L}$. *A proposition* P *is* valid *in* $\mathcal{M}$ *is for all assignments* ϕ, $|P|_\phi = 1$. *A theory* Γ *is valid in* $\mathcal{M}$ *if all its axioms are valid. The structure* $\mathcal{M}$ *is a* model *of* Γ *if* Γ *is valid in* $\mathcal{M}$.

Proposition 1.3.1 (Soundness) *Let* Γ *be a theory. If the proposition* P *has a proof in* Γ, *then it is valid in all the models of* Γ.

Proof. By induction over the height of a proof of P in Γ.

Corollary 1.3.2 *If the theory* Γ *has a model in which* P *is not valid then* P *has no proof in* Γ.

Corollary 1.3.3 *If* Γ *has a model then* Γ *is consistent.*

Example 1.3.1 *Consider the language containing two predicate symbol* $=$ *and* $\leq$ *of two arguments. Consider the theory* $\mathcal{O}$ *formed with the axioms of equality and*

$$\forall x\ (x \leq x)$$

$$\forall x\ \forall y\ ((x \leq y \wedge y \leq x) \Rightarrow x = y)$$

$$\forall x\ \forall y\ \forall z\ ((x \leq y \wedge y \leq z) \Rightarrow x \leq z)$$

From these axiom we cannot deduce the proposition

$$\forall x\ \forall y\ (x \leq y \vee y \leq x)$$

Indeed, consider the structure $\mathcal{M} = \langle \mathbb{N}, I, | \rangle$ *where* $I(n, m) = 1$ *if* $n = m$ *and* 0 *otherwise,* $|(n, m) = 1$ *if* n *is a divisor of* m *and* 0 *otherwise. The structure* $\mathcal{M}$ *is a model of* $\mathcal{O}$. *But it is not a model of the proposition* $\forall x\ \forall y\ (x \leq y \vee y \leq x)$, *because* 2 *is not a divisor of* 3 *and* 3 *is not a divisor of* 2.

Remark. The first use of the notion of model to prove that some proposition has no proof in a theory is probably that of F. Klein who has built in 1871 a model of all the axioms of Euclid's geometry except the axiom of parallels, showing that the axiom of parallels cannot be deduced from the other axioms of Euclid's geometry. (However the notion of model has only been defined by A. Tarski, more than fifty years later, in 1936).

The soundness theorem has a converse we shall not prove here.

Proposition 1.3.4 (Gödel's completeness theorem) *Let Γ be a theory. If the proposition P is valid in all the models of Γ then it has a proof in Γ.*

Remark. The soundness theorem holds also for constructive proofs. But not the completeness theorem. For instance, let P be a proposition symbol (*i.e.* a predicate symbol of zero arguments). We shall see (exercise 4.1.1) that the proposition $P \vee \neg P$ has no constructive proof, but it is valid in all models. The notion of model needs to be adapted for constructive proofs.

Remark. In proof theory, the notion of model is mostly used to prove independence results, *i.e.* that some propositions have no proof in some theories. The notion of model is also used in algebra. For instance, ordered sets can be defined as the models of the theory $\mathcal{O}$ of example 1.3.1. Groups can also be defined as the models of some theory, but it can be shown that Archimedian complete ordered fields cannot be defined as the models of some theory. This fact may be used to prove, for instance, that there are ordered sets or groups of all infinite cardinals, while it is known that all Archimedian complete ordered fields are isomorphic to $\mathbb{R}$ and thus that they all have cardinal $2^{\aleph_0}$. The branch of mathematics that studies these applications of logic to algebra is called *model theory*.

Remark. A common misconception is that the notion of model can be used, as an alternative to the notion of proof, to define the notion of mathematical truth, *i.e.* that instead of saying that a proposition is true if it has a proof, we could say that it is true if it is valid in all models. The problem with such a definition of truth is that, unlike the fact that a tree is a proof of some proposition, the fact that a proposition is valid in all models is not self evident, *i.e.* it cannot be checked in an algorithmic way. Thus, the fact that some proposition is valid in all models must itself be justified by some argument. Thus, such a definition of truth reduces the question of the truth of the proposition "P" to that of the proposition "the proposition P is valid in all models" and trying to justify some proposition we enter into an infinite regression.

Remark. (Many-valued model) In the definition 1.3.1, the truth value 0 is used as denotation of non valid propositions, and the truth value 1 as denotation of valid propositions. This definition can be extended by adding other truth values. A common extension is to take a third value for propositions whose validity is unknown in this model.

Chapter 2

Extensions of predicate logic

2.1 Many-sorted predicate logic

In some theories, we want to distinguish several sorts of objects. For instance, in a language with the individual symbols $German$, $English$, $French$, $Germany$, $United-Kingdom$, $Ireland$, $France$ and a predicate L, we can form the propositions

$$L(German, Germany)$$

$$L(English, United - Kingdom)$$

$$L(English, Ireland)$$

$$L(French, France)$$

expressing that $German$ is an official language of $Germany$, ... In this theory, we can also form the unwanted proposition

$$L(Germany, Germany)$$

An extension of predicate logic permits to restrict the term and proposition formation rules, in such a way that such unwanted propositions are avoided.

Definition 2.1.1 (Many-sorted language) *A* language *is a set of sorts, a set of function symbols and a set of predicate symbols. To each function symbol is associated a $n+1$-uple of sorts $\langle s_1, ..., s_n, s_{n+1} \rangle$ called its* rank *and to each predicate symbol is associated a n-uple of sorts $\langle s_1, ..., s_n \rangle$ called its* rank.

Definition 2.1.2 (Term in a many-sorted language) *Let $\mathcal{L}$ be a many-sorted language and $\mathcal{V}_s$ be a a family of disjoint infinite sets indexed by sorts. The* terms *of the language $\mathcal{L}$ with variables $\mathcal{V}_s$ are defined by the following rules*

- *if x is a variable of $\mathcal{V}_s$ then the tree whose root is labeled by x and that has no sub-tree is a term of sort s,*
- *if f is a function symbol of rank $\langle s_1, ..., s_n, s_{n+1}\rangle$ and $t_1, ..., t_n$ are terms of sort $s_1, ..., s_n$ then the tree whose root is labeled by f and whose sub-trees are $t_1, ..., t_n$ is a term of sort s_{n+1}.*

Definition 2.1.3 (Proposition in a many-sorted language) *Let $\mathcal{L}$ be a many-sorted language and $\mathcal{V}_s$ be a a family of disjoint infinite sets indexed by sorts. The* propositions *of the language $\mathcal{L}$ with variables $\mathcal{V}_s$ are defined by the following rules*

- *if P is a predicate symbol of rank $\langle s_1, ..., s_n\rangle$ and $t_1, ..., t_n$ are terms of sort $s_1, ..., s_n$, then the tree whose root is labeled by P and whose sub-trees are $t_1, ..., t_n$ is a proposition,*
- *the trees whose root are labeled by $\top$ and $\bot$ and that have no sub-tree are propositions,*
- *if A is a proposition then the tree whose root is labeled by $\neg$ and whose sub-tree is A is a proposition,*
- *if A and B are propositions then the trees whose root are labeled by $\wedge$, $\vee$ or $\Rightarrow$ and whose sub-trees are A and B are propositions,*
- *if A is a proposition and x a variable then the trees whose root are labeled $\forall x$ and $\exists x$ and whose sub-tree is A are propositions.*

The definition of a substitution is restricted in such a way that a variable of sort s can only be substituted by a term of sort s. The proof rules are the same than in ordinary predicate logic.

Definition 2.1.4 (Structure in a many-sorted language) *Let $\mathcal{L}$ be a language formed with the sorts $s_0, s_1, ...$, the function symbols $f_0, f_1, ...$ of number or arguments and the predicate symbols $P_0, P_1, ...$ A* structure *$\mathcal{M}$ built on $\mathcal{L}$ is a n-uple formed with*

- *a family of non empty sets $M_{s_0}, M_{s_1}, ...,$*
- *a function $\hat{f}_0$ from $M_{s_1} \times ... \times M_{s_n}$ to $M_{s_{n+1}}$ where $\langle s_1, ..., s_n, s_{n+1}\rangle$ is the rank of f_0, a function $\hat{f}_1$ ...*
- *a function $\hat{P}_0$ from $M_{s_1} \times ... \times M_{s_n}$ to $\{0, 1\}$ where $\langle s_1, ..., s_n\rangle$ is the rank of P_0, a function $\hat{P}_1$ ...*

The denotation of a term and a proposition is defined in the same way as in ordinary predicate logic, with the extra condition that in the case of quantifiers, the object a belongs to M_s where s is the sort of the quantified variable.

Proposition 2.1.1 (Soundness and completeness) *A proposition has a proof in a theory if and only if it is valid in all the models of this theory.*

Remark. Predicate logic is a particular case of many-sorted predicate logic with a single sort.

2.2 Predicate logic modulo

In predicate logic, proofs are sequences of deduction steps. The idea of predicate logic modulo is that a proof is not a sequence of deduction steps, but a sequence of deduction steps *and of computation steps*. For instance, in arithmetic, to prove the proposition

$$\exists x\ (2 \times x = 4)$$

we use the $\exists$-intro rule and we are reduced to prove the proposition $2 \times 2 = 4$. Then, we have to use the axioms of addition and multiplication to prove this proposition. In predicate logic modulo, we can simply compute the term 2×2 and obtain the proposition $4 = 4$ that can easily be proved with the identity axiom.

2.2.1 Deduction rules

Definition 2.2.1 *A relation $\equiv$ defined on terms and propositions of a language is a* congruence *if*

- *it is an equivalence relation,*
- *it is compatible with all function symbols, predicate symbols, connectors and quantifiers,* i.e. *if $t \equiv u$ then $f(t) \equiv f(u)$, if $A \equiv B$ and $A' \equiv B'$ then $A \wedge A' \equiv B \wedge B'$, if $A \equiv B$ then $\forall x\ A \equiv \forall x\ B$, ...*

In predicate logic modulo a *theory* is formed with a set of axioms Γ such that the membership of some proposition to this set can be decided in an algorithmic way *and* a congruence $\equiv$ on terms and propositions such that the equivalence of two propositions can be decided in an algorithmic way. Before or after each deduction step, we can transform the proved proposition into any equivalent one. The deduction rules are thus modified to take these computations into account. These rules permit to prove sequents of the form $\Gamma \vdash_{\equiv} A$. A proposition is said to have a proof in the theory $\Gamma, \equiv$ if the sequent $\Gamma \vdash_{\equiv} A$ has a proof with the following deduction rules.

Definition 2.2.2 (Deduction rules modulo)

$$\frac{}{\Gamma \vdash_{\equiv} B}\ \textit{Axiom if } A \in \Gamma \textit{ and } A \equiv B$$

$$\frac{}{\Gamma \vdash_{\equiv} A}\ \top\textit{-intro if } A \equiv \top$$

$$\frac{\Gamma \vdash_{\equiv} B}{\Gamma \vdash_{\equiv} A}\ \bot\textit{-elim if } B \equiv \bot$$

$$\frac{\Gamma, A \vdash_{\equiv} B}{\Gamma \vdash_{\equiv} C}\ \neg\textit{-intro if } B \equiv \bot \textit{ and } C \equiv \neg A$$

$$\frac{\Gamma \vdash_{\equiv} C \quad \Gamma \vdash_{\equiv} A}{\Gamma \vdash_{\equiv} B}\ \neg\textit{-elim if } C \equiv \neg A \textit{ and } B \equiv \bot$$

$$\frac{\Gamma \vdash_{\equiv} A \quad \Gamma \vdash_{\equiv} B}{\Gamma \vdash_{\equiv} C}\ \wedge\textit{-intro if } C \equiv (A \wedge B)$$

$$\frac{\Gamma \vdash_{\equiv} C}{\Gamma \vdash_{\equiv} A}\ \wedge\textit{-elim if } C \equiv (A \wedge B)$$

$$\frac{\Gamma \vdash_{\equiv} C}{\Gamma \vdash_{\equiv} B}\ \wedge\textit{-elim if } C \equiv (A \wedge B)$$

$$\frac{\Gamma \vdash_{\equiv} A}{\Gamma \vdash_{\equiv} C}\ \vee\textit{-intro if } C \equiv (A \vee B)$$

$$\frac{\Gamma \vdash_{\equiv} B}{\Gamma \vdash_{\equiv} C}\ \vee\textit{-intro if } C \equiv (A \vee B)$$

$$\frac{\Gamma \vdash_{\equiv} D \quad \Gamma, A \vdash_{\equiv} C \quad \Gamma, B \vdash_{\equiv} C}{\Gamma \vdash_{\equiv} C}\ \vee\textit{-elim if } D \equiv (A \vee B)$$

$$\frac{\Gamma, A \vdash_{\equiv} B}{\Gamma \vdash_{\equiv} C}\ \Rightarrow\textit{-intro if } C \equiv (A \Rightarrow B)$$

$$\frac{\Gamma \vdash_{\equiv} C \quad \Gamma \vdash_{\equiv} A}{\Gamma \vdash_{\equiv} B}\ \Rightarrow\textit{-elim if } C \equiv (A \Rightarrow B)$$

$$\frac{\Gamma \vdash_{\equiv} A}{\Gamma \vdash_{\equiv} B}\ \langle x, A\rangle\ \forall\textit{-intro if } B \equiv (\forall x\ A) \textit{ and } x \notin FV(\Gamma)$$

$$\frac{\Gamma \vdash_{\equiv} B}{\Gamma \vdash_{\equiv} C}\ \langle x, A, t\rangle\ \forall\textit{-elim if } B \equiv (\forall x\ A) \textit{ and } C \equiv (t/x)A$$

$$\frac{\Gamma \vdash_{\equiv} C}{\Gamma \vdash_{\equiv} B}\ \langle x, A, t\rangle\ \exists\textit{-intro if } B \equiv (\exists x\ A) \textit{ and } C \equiv (t/x)A$$

$$\frac{\Gamma \vdash_{\equiv} C \quad \Gamma, A \vdash_{\equiv} B}{\Gamma \vdash_{\equiv} B}\ \langle x, A\rangle\ \exists\textit{-elim if } C \equiv (\exists x\ A) \textit{ and } x \notin FV(\Gamma, B)$$

$$\frac{}{\Gamma \vdash_{\equiv} A}\ B \textit{ Excluded middle if } A \equiv (B \vee \neg B)$$

Proposition 2.2.1 (Equivalence) *For every congruence $\equiv$ there is a theory $\mathcal{T}$ such that $\Gamma \vdash_{\equiv} A$ if and only if $\mathcal{T}\Gamma \vdash A$.*

Proof. We take, for instance, all the axioms of the form $\forall x_1 \ldots \forall x_n\ (A \Leftrightarrow B)$ where $A \equiv B$.

Definition 2.2.3 (Model of a theory modulo) *A structure $\mathcal{M}$ is a model of a theory modulo $\Gamma, \equiv$ if all the axioms of Γ are valid in $\mathcal{M}$ and each time two terms (resp. propositions) are congruent they have the same denotation in $\mathcal{M}$.*

Proposition 2.2.2 (Soundness and completeness) *A proposition has a proof in a theory if and only if it is valid in all the models of this theory.*

2.2.2 Congruences defined by rewrite rules

Congruences used in predicate logic modulo are often defined by rewrite systems.

Definition 2.2.4 (Rewrite rule, rewrite system) *A* rewrite rule *is an ordered pair of terms or an ordered pair of propositions* $\langle l, r\rangle$ *written* $l \longrightarrow r$. *A* rewrite system *is a set of rewrite rules.*

Definition 2.2.5 (Redex) *Let* $\mathcal{R}$ *be a rewrite system and* t *be a term. The term* t *is a* redex *(reducible expression) if there exists a rule* $l \longrightarrow r$ *in* $\mathcal{R}$ *and a substitution* σ *such that* $t = \sigma l$. *A term* t *is said to contain a redex if one of its sub-terms is a redex.*

Definition 2.2.6 (One step reduction) *Let* $\mathcal{R}$ *be a rewrite system. A term (resp. a proposition)* t *reduces to a term (resp. a proposition)* u *in one step* ($t \longrightarrow^1 u$) *if there is a sub-term* t' *of* t *and a substitution* σ *such that* $t' = \sigma l$ *and* u *is obtained by replacing in* t *the sub-term* t' *by the term* σu.

Definition 2.2.7 (Reduction sequence) *Let* $\mathcal{R}$ *be a rewrite system. A* reduction sequence *is a finite or infinite sequence of terms (resp. propositions)* $t_0, t_1, \ldots$ *such that for every* i, $t_i \longrightarrow^1 t_{i+1}$.

Definition 2.2.8 (Reduction) *Let* $\mathcal{R}$ *be a rewrite system. A term (resp. a proposition)* t *reduces to a term (resp. a proposition)* u ($t \longrightarrow u$) *if there is a finite reduction sequence starting on* t *and ending on* u.

Definition 2.2.9 (Congruence sequence) *Let* $\mathcal{R}$ *be a rewrite system. A* congruence sequence *is a finite or infinite sequence of terms (resp. propositions)* $t_0, t_1, \ldots$ *such that for every* i, $t_i \longrightarrow^1 t_{i+1}$ *or* $t_{i+1} \longrightarrow^1 t_i$.

Definition 2.2.10 (Congruence) *Let* $\mathcal{R}$ *be a rewrite system. Two terms (resp. two propositions)* t *and* u *are congruent if there is a finite congruence sequence starting on* t *and ending on* u.

Definition 2.2.11 (Normal term) *A term (resp. a proposition) is* normal *if it contains no redex. A term (resp. a proposition)* u *is a* normal form *of a term (resp. a proposition)* t *if* $t \longrightarrow u$ *and* u *is normal.*

Definition 2.2.12 (Terminating) *A term (resp. a proposition) is* terminating *if it has a normal form,* i.e. *if there exists a finite reduction sequence starting on this term and ending on a normal term. It is* strongly terminating *if all reduction sequences issued from this term are finite.*

A rewrite system is terminating (resp. strongly terminating) if all terms and all propositions are terminating (resp. strongly terminating).

Definition 2.2.13 (Confluent) *A rewrite system is* confluent *if whenever a term (resp. proposition)* t *reduces to two terms (resp. proposition)* u_1 *and* u_2, *then there exists a term (resp. proposition)* v *such that* u_1 *reduces to* v *and* u_2 *reduces to* v.

Proposition 2.2.3 *In a confluent rewrite system, two terms (resp. two propositions) are congruent if and only if they reduce to a common term.*

Proof. By induction on the length of the congruence sequence.

Proposition 2.2.4 *In a confluent rewrite system a term has at most one normal form.*

Proof. If u_1 and u_2 are normal forms of t, then $t \longrightarrow u_1$ and $t \longrightarrow u_2$. By confluence, there exists a term v such that $u_1 \longrightarrow v$ and $u_2 \longrightarrow v$. As u_1 and u_2 are normal $u_1 = v = u_2$.

Proposition 2.2.5 *In a terminating and confluent rewrite system a term has exactly one normal form. And this normal form can be computed form the term.*

Proof. Termination yields existence and confluence unicity. To compute the normal form, it is sufficient to reduce the term until a normal form is reached.

Proposition 2.2.6 *In a terminating and confluent rewrite system two terms (resp. propositions) are congruent if they have the same normal form.*

Proof. If the two terms have the same normal form, then they are congruent. If they are congruent, so are their normal forms and these two normal forms reduce to a common term. Hence they are equal.

Proposition 2.2.7 *In a terminating and confluent rewrite system, the congruence can be checked in an algorithmic way.*

Proof. Congruence can be checked by computing the normal forms and checking their identity.

Example 2.2.1 (A presentation of arithmetic in predicate logic modulo) *To formulate arithmetic in predicate logic modulo, we can keep the axioms of equality and the axioms*

$$\forall x\ \forall y\ (Su(x) = Su(y) \Rightarrow x = y)$$

$$\forall x\ \neg(0 = Su(x))$$

$$((0/z)A \wedge (\forall x\ ((x/z)A \Rightarrow (Su(x)/z)A))) \Rightarrow \forall y\ (y/z)A$$

and replace the axioms

$$\forall y\ (0 + y = y)$$

$$\forall x\ \forall y\ (Su(x) + y = Su(x + y))$$

$$\forall y\ (0 \times y = 0)$$

$$\forall x\ \forall y\ (Su(x) \times y = (x \times y) + y)$$

by the rewrite rules

$$0 + y \longrightarrow y$$
$$Su(x) + y \longrightarrow Su(x + y)$$
$$0 \times y \longrightarrow 0$$
$$Su(x) \times y \longrightarrow x \times y + y$$

Excercise 2.2.1 *Give a proof of the proposition* $\exists x\ (2 \times x = 4)$.

2.3 Binding logic

In mathematics, we use the notation $x \mapsto x + 2$ to designate the function that maps x to $x + 2$. Such a symbol is said to be a *binder*, because the variable x that is free in $x + 2$ is bound in $x \mapsto x + 2$. In predicate logic the only binders are the quantifiers $\forall$ and $\exists$ that bind variables in propositions, but there is no way to bind variables in terms and so, there is no way to form a term such as $x \mapsto t$.

Binding logic is an extension of predicate logic where function symbols and predicate symbols can bind variables in their arguments. To each function symbol or predicate symbol of n arguments is associated a *rank* $\langle k_1, ..., k_n \rangle$ where $k_1, ..., k_n$ are natural numbers. Then, if f has the rank $\langle k_1, ..., k_n \rangle$ and $t_1, ..., t_n$ are terms, we can form the term

$$f(x_1^1 ... x_{k_1}^1\ t_1, ..., x_1^n ... x_{k_n}^n\ t_n)$$

where $x_1^1, ..., x_{k_1}^1$ are bound in the term t_1, ..., $x_1^n, ..., x_{k_n}^n$ are bound in the term t_n.

In many-sorted binding logic a rank is a sequence of sequences of sorts. Then, when a function symbol f has the rank

$$\langle \langle s_1^1, ..., s_{k_1}^1, s_{k_1+1}^1 \rangle, ... \langle s_1^n, ..., s_{k_n}^n, s_{k_n+1}^n \rangle, s^{n+1} \rangle$$

$x_1^1, ..., x_{k_1}^1$ are variables of sorts $s_1^1, ..., s_{k_1}^1$, ..., $x_{k_n}^n, ..., x_{k_n}^n$ are variables of sorts $s_1^n, ..., s_{k_n}^n$ and $t_1, ..., t_n$ are terms or sorts $s_{k_1+1}^1, ..., s_{k_n+1}^n$ then the sort of the term $f(x_1^1 ... x_{k_1}^1\ t_1, ..., x_1^n ... x_{k_n}^n\ t_n)$ is s^{n+1}.

Substitution is modified in such a way that bound variables are renamed to avoid capture. Proof rules are the same than in predicate logic or predicate logic modulo. A notion of model can also be defined for binding logic, but we shall not present it here.

Chapter 3

Type theory

In arithmetic, (example 1.2.3), we can speak about the natural numbers but not about the functions mapping natural numbers to natural numbers nor about the sets of natural numbers. Thus, arithmetic is not sufficient to express mathematics and we need to build more expressive theories. Set theory and type theory (also called higher-order logic) are such theories.

3.1 Naive set theory

In the language of arithmetic, the symbol Su is a function symbol, thus, it may be used to form terms, such as $Su(0)$, but it is not itself a term. If we want to be able to speak about the function Su, we need the symbol Su to be a term and hence an individual symbol. When Su is an individual symbol, we cannot form the term $Su(0)$ anymore. Hence, we need to introduce a new function symbol α for the application of a function to its argument and write this term $\alpha(Su, 0)$.

We could also introduce a function symbol α_2 for functions of two arguments, but this is not needed. Indeed, a function f of two arguments can always be seen as a function of one argument that maps x to the function that maps y to $f(x, y)$. Thus instead of writing $\alpha_2(f, x, y)$ we can write $\alpha(\alpha(f, x), y)$.

To ease notations we shall write $(f\ x)$ for the term $\alpha(f, x)$ and $(f\ x_1\ ...\ x_n)$ for the term $(...(f\ x_1)...x_n)$.

In the same way, we want the symbols designating predicates (sets), to be terms and hence individual symbols, for instance if the individual symbol *prime* designates the set of prime numbers, to express that the number 2 is prime, we cannot write $prime(2)$, but we need to introduce a new predicate symbol $\in$ and write this proposition $2 \in prime$.

For terms expressing predicates of several arguments to be terms, we must also introduce symbols $\in_2$, $\in_3$, ... For predicates of zero arguments (*i.e.* propositions) to be terms, we must introduce a predicate symbol $\in_0$, also written ε. The proposition $\in_2 (R, x, y)$ expresses that x and y are related by the predicate of two arguments (relation) R. The proposition $\varepsilon(E)$ expresses that the pred-

icate of zero argument E is true. The only difference between E and $\varepsilon(E)$ is that E is a term (designating an object) while $\varepsilon(E)$ is a proposition (expressing a fact). The object E may be called the *propositional content* of the proposition $\varepsilon(E)$.

The notions of function and set are redundant. We can express a function as a functional relation (its graph), *i.e.* as a set of ordered pairs. In this case, we just need the symbol $\in$.

Conversely, we can define a set as its characteristic function, *i.e.* as the function mapping its argument to the propositional content of the fact that x belongs to the set. In this case, we just need the symbols α and ε. If E is a set and x an object, the propositional content of the fact that x belongs to E is designated by the term $(E\ x)$ and the fact that x belongs to E is expressed by the proposition $\varepsilon(E\ x)$. Thus, the proposition $x \in E$ is thus written $\varepsilon(E\ x)$. In the same way, the proposition $\in_2 (R, x, y)$ is written $\varepsilon(R\ x\ y)$, ...

Let us now turn to the making of functions and sets. Whenever we have a term t and variables $x_1, ..., x_n$, we want to consider the function $x_1, ..., x_n \mapsto t$, for instance the function $x \mapsto (3 \times x)$. This function is such that we get back t when we apply it to $x_1, ..., x_n$. Whenever we have a proposition P and variables $x_1, ..., x_n$, we want to build the predicate $\{x_1, ..., x_n \mid P\}$, for instance the set $\{x \mid \exists y\ (x = 2 \times y)\}$. This predicate is such that we get back P when we apply it to $x_1, ..., x_n$.

A solution would be to introduce for each term t and sequence of variables $x_1, ..., x_n$ an individual symbol $C_{x_1,...,x_n,t}$ and an axiom

$$(C_{x_1,...,x_n,t}\ x_1\ ...\ x_n) = t$$

and for each proposition P and sequence of variables $x_1, ..., x_n$ an individual symbol $E_{x_1,...,x_n,P}$ and an axiom

$$\varepsilon(E_{x_1,...,x_n,P}\ x_1\ ...\ x_n) \Leftrightarrow P$$

In predicate logic modulo, these axioms can be transformed into rewrite rules

$$(C_{x_1,...,x_n,t}\ u_1\ ...\ u_n) \longrightarrow (u_1/x_1, ..., u_n/x_n)t$$

$$\varepsilon(E_{x_1,...,x_n,P}\ u_1\ ...\ u_n) \longrightarrow (u_1/x_1, ..., u_n/x_n)P$$

But, not all these symbols are necessary, and we can restrict to a much smaller language.

Definition 3.1.1 (Naive set theory) *The language of naive set theory is formed with*

- *a predicate symbol ε of one argument.*
- *a function symbol α of two arguments,*
- *individual symbols S, K, $\dot{\top}$, $\dot{\bot}$, $\dot{\neg}$, $\dot{\wedge}$, $\dot{\vee}$, $\dot{\Rightarrow}$, $\dot{\forall}$ and $\dot{\exists}$.*

and the congruence defined by the rewrite rules

$$(S\ x\ y\ z) \longrightarrow ((x\ z)\ (y\ z))$$

$$(K\ x\ y) \longrightarrow x$$

$$\varepsilon(\dot{\top}) \longrightarrow \top$$

$$\varepsilon(\dot{\bot}) \longrightarrow \bot$$

$$\varepsilon(\dot{\neg}\ x) \longrightarrow \neg\varepsilon(x)$$

$$\varepsilon(\dot{\wedge}\ x\ y) \longrightarrow (\varepsilon(x) \wedge \varepsilon(y))$$

$$\varepsilon(\dot{\vee}\ x\ y) \longrightarrow (\varepsilon(x) \vee \varepsilon(y))$$

$$\varepsilon(\dot{\Rightarrow}\ x\ y) \longrightarrow (\varepsilon(x) \Rightarrow \varepsilon(y))$$

$$\varepsilon(\dot{\forall}\ x) \longrightarrow \forall y\ \varepsilon(x\ y)$$

$$\varepsilon(\dot{\exists}\ x) \longrightarrow \exists y\ \varepsilon(x\ y)$$

Proposition 3.1.1 (Comprehension) *For each term t and sequence of variables $x_1, ..., x_n$ there is a term u such that*

$$(u\ x_1\ ...\ x_n) \equiv t$$

and for each proposition P and sequence of variables $x_1, ..., x_n$ there is a term u such that

$$\varepsilon(u\ x_1\ ...\ x_n) \equiv P$$

Proof. By induction over the height of t (resp. P).

Many variants of this theory have been proposed in the History of mathematics: Cantor's set theory (1872), Frege's *Begriffschrift* (1879), Church's pure λ-calculus (1932), ... Unfortunately, all these systems are contradictory. A contradiction is given by Russell's paradox.

By proposition 3.1.1 there exists a term R such that

$$\forall x\ (\varepsilon(R\ x) \Leftrightarrow \neg\varepsilon(x\ x))$$

(take for instance $R = (S\ (K\ \dot{\neg})\ (S\ (S\ K\ K)\ (S\ K\ K)))$). The set R is the set of all sets that do not contain themselves. By definition, this set contains itself if and only if it does not, which is contradictory. More precisely, with the elimination rule of the universal quantifier $\forall$, we can deduce from this proposition the proposition

$$\varepsilon(R\ R) \Leftrightarrow \neg\varepsilon(R\ R)$$

and we have seen (exercise 1.2.3) that from such a proposition, we can prove a contradiction.

3.2 Set theory

In naive set theory, it is possible to construct functions defined on all the universe and to construct sets in comprehension with any property P. To restrict naive set theory and avoid paradoxes, we may restrict function construction in such a way that functions are defined with a domain of definition and, similarly, only subsets of already constructed sets are constructed in comprehension. Such ideas are exploited in several theories, including set theory and simple type theory.

In Zermelo's set theory and in its extension Zermelo-Fraenkel set theory, the basic notion is that of set and functions are defined as relations. Thus the language does not contain symbols α and ε, but a symbol $\in$.

When P is a proposition, it is not always possible to form the set of objects verifying the property P. This is only allowed in four cases.

- If x and y are two sets, we can form the set $\{x, y\}$ containing exactly x and y (the symbol $\{,\}$ is a function symbol),
- If x is a set we can form the set $\bigcup(x)$ containing the elements of the elements of x,
- If x is a set, we can form a set $\wp(x)$ containing the subsets of x.
- If x is a set and P is a proposition containing variables $y, z_1, ..., z_n$, we can form the subset of x of the elements y verifying P. This set can be written $f_{y,z_1,...,z_n,P}(x, z_1, ..., z_n)$ where $f_{y,z_1,...,z_n,P}$ is a function symbol.

The axioms are

$$z \in \{x, y\} \Leftrightarrow (z = x \vee z = y)$$

$$y \in \bigcup(x) \Leftrightarrow (\exists z\ (y \in z \wedge z \in x))$$

$$y \in \wp(x) \Leftrightarrow (\forall z\ (z \in y \Rightarrow z \in x))$$

$$y \in f_{y,z_1,...,z_n,P}(x, z_1, ..., z_n) \Leftrightarrow (y \in x \wedge P)$$

There is no way to construct the set of sets that do not belong to themselves and Russell's paradox is avoided.

In predicate logic modulo, these axioms may be transformed into rewrite rules

$$t \in \{u, v\} \longrightarrow t = u \vee t = v$$

$$t \in \bigcup(u) \longrightarrow \exists z\ (t \in z \wedge z \in u)$$

$$t \in \wp(u) \longrightarrow \forall z\ (z \in t \Rightarrow z \in u)$$

$$t \in f_{y,z_1,...,z_n,P}(u, v_1, ..., v_n) \longrightarrow t \in u \wedge (t/y, v_1/z_1, ..., v_n/z_n)P$$

This system does not terminate as the proposition $f_{y,\neg y\in y}(x) \in f_{y,\neg y\in y}(x)$ reduces to $f_{y,\neg y\in y}(x) \in x \wedge \neg f_{y,\neg y\in y}(x) \in f_{y,\neg y\in y}(x)$. Thus, if we call A the

proposition $f_{y,\neg y\in y}(x) \in f_{y,\neg y\in y}(x)$ and B the proposition $f_{y,\neg y\in y}(x) \in x$ we have

$$A \longrightarrow B \wedge \neg A$$

The decidability of the congruence relation generated by these rule is an open problem.

3.3 Simple type theory

Simple type theory originates from the work of A.N. Whitehead and B. Russell. It is another way to restrict naive set theory to avoid paradoxes. In this theory, the basic notion is that of function. Each function has a domain of definition and the application $(f\ t)$ can be constructed only when t belongs to the domain of the function f, otherwise it is prohibited by the syntax. Hence simple type theory is a many-sorted theory. Taking all sets as possible function domains, *i.e.* all sets as sorts, makes it difficult to decide if a term $(f\ t)$ is well-formed or not because we need to decide if the term t designates an object that belongs to the domain of f or not. Moreover as an object can belong to several set, it should have several sorts. In type theory, an object has only one sort that is the maximal set it belongs to. It is called the *type* of this object. There is one type ι for atoms and one type o for propositional contents, then each time we have two types T and U, we can form the type $T \to U$ of functions mapping objects of sort T to objects of sort U.

Definition 3.3.1 (Simple types) Simple types *are closed terms formed with the individual symbols ι and o and the function symbol $\to$ of two arguments.*

To ease notation, we write $T_1 \to T_2 \to ... \to T_n \to U$ for the type $(T_1 \to (T_2 ... \to (T_n \to U)...))$.

Definition 3.3.2 (Language of type theory) *The language of simple type theory in predicate logic modulo is formed with*

- *a predicate symbol ε of rank $\langle o\rangle$,*
- *for each pair of type T, U, a function symbol $\alpha_{T,U}$ of rank $\langle T \to U, T, U\rangle$,*
- *for each triple of types T, U, V an individual symbol $S_{T,U,V}$ of sort $(T \to U \to V) \to (T \to U) \to T \to V$,*
  for each pair of types T, U an individual symbol $K_{T,U}$ of sort $T \to U \to T$,
  individual symbols $\dot\top$ and $\dot\bot$ of sort o,
  an individual symbol $\dot\neg$ of sort $o \to o$,
  individual symbols $\dot\wedge$, $\dot\vee$, $\dot\Rightarrow$ of sort $o \to o \to o$,
  for each type T, individual symbols $\dot\forall_T$ and $\dot\exists_T$ of type $(T \to o) \to o$.

Definition 3.3.3 (Rewrite system of type theory) *The rewrite system $\mathcal{T}$ is defined by the rules*

$$(S_{T,U,V}\ x\ y\ z) \longrightarrow ((x\ z)\ (y\ z))$$

$$(K_{T,U}\ x\ y) \longrightarrow x$$

$$\varepsilon(\dot{\top}) \longrightarrow \top$$

$$\varepsilon(\dot{\bot}) \longrightarrow \bot$$

$$\varepsilon(\dot{\neg}\ x) \longrightarrow \neg\varepsilon(x)$$

$$\varepsilon(\dot{\wedge}\ x\ y) \longrightarrow \varepsilon(x) \wedge \varepsilon(y)$$

$$\varepsilon(\dot{\vee}\ x\ y) \longrightarrow \varepsilon(x) \vee \varepsilon(y)$$

$$\varepsilon(\dot{\Rightarrow}\ x\ y) \longrightarrow \varepsilon(x) \Rightarrow \varepsilon(y)$$

$$\varepsilon(\dot{\forall}_T\ x) \longrightarrow \forall y\ \varepsilon(x\ y)$$

$$\varepsilon(\dot{\exists}_T\ x) \longrightarrow \exists y\ \varepsilon(x\ y)$$

Proposition 3.3.1 (Comprehension) *For each term t there is a term u not containing the variable x such that $(u\ x) \equiv t$. For each proposition P there is a term u such that $\varepsilon(u) \equiv A$.*

Proof. By induction over the height of t.

- If $t = x$ then we take $u = (S\ K\ K)$, we have $(u\ x) = (S\ K\ K\ x) \equiv (K\ x\ (K\ x)) \equiv x$.
- If t is a variable different from x or an individual symbol, we take $u = (K\ t)$, we have $(u\ x) = (K\ t\ x) \equiv t$.
- If $t = (t_1\ t_2)$, then by induction hypothesis, there are terms u_1 and u_2 such that $(u_1\ x) \equiv t_1$ and $(u_2\ x) \equiv t_2$. We take $u = (S\ u_1\ u_2)$. We have $(u\ x) = (S\ u_1\ u_2\ x) \equiv ((u_1\ x)\ (u_2\ x)) \equiv (t_1\ t_2) = t$.

By induction over the height of A.

- If $A = \varepsilon(t)$, we take $u = t$.
- If $A = B \wedge C$, then by induction hypothesis, there are terms v and w such that $\varepsilon(v) \equiv B$ and $\varepsilon(w) \equiv C$. We take $u = (\dot{\wedge}\ v\ w)$. We proceed the same way if $A = \top, \bot, \neg B, B \vee C$ or $B \Rightarrow C$.
- If $A = \forall x\ B$, then by induction hypothesis, there is a term v such that $\varepsilon(v) \equiv B$ and there is a term w not containing x such that $(w\ x) \equiv v$ and hence $\varepsilon(w\ x) \equiv \varepsilon(v) \equiv B$. We take $u = (\dot{\forall}\ w)$. We have $\varepsilon(u) \equiv \forall x\ \varepsilon(w\ x) \equiv \forall x\ B$. We proceed the same way if $A = \exists x\ B$.

Definition 3.3.4 (Leibniz' Equality) *By the proposition 3.3.1 there is a term $\doteq$ such that*

$$\varepsilon(\doteq\ x\ y) \equiv \forall p\ (\varepsilon(p\ x) \Rightarrow \varepsilon(p\ y))$$

Excercise 3.3.1 *Prove*

$$\forall x\ \varepsilon(x \doteq x)$$

and for each proposition A

$$\forall x\ \forall y\ (\varepsilon(x \doteq y) \Rightarrow ((x/z)A \Rightarrow (y/z)A))$$

To prove that the rewrite system $\mathcal{T}$ is terminating, we first focus on the two first rules.

Proposition 3.3.2 (Tait's theorem) *The rewrite system*

$$(S_{T,U,V}\ x\ y\ z) \longrightarrow ((x\ z)\ (y\ z))$$

$$(K_{T,U}\ x\ y) \longrightarrow x$$

is strongly terminating.

Proof. The set of *reducible* terms of type T is defined by induction over the height of T.

- If T is ι or o then t is reducible of type T if and only if it is strongly terminating.
- If $T = T_1 \to T_2$ then t is reducible of type T if and only if for every reducible term u of type T_1, the term $(t\ u)$ is reducible of type T_2.

We prove by induction over the height of T that

- (1) all reducible terms are strongly terminating and
- (2) variables and individual symbols other than S and K are reducible terms.

Let $T = U_1 \to ... \to U_n \to V$ ($V = \iota$ or $V = o$). (1) If t is a reducible term of type T, then let x_1, ..., x_n be variables of types $U_1, ..., U_n$. By induction hypothesis, the variables $x_1, ..., x_n$ are reducible. Hence, the term $(t\ x_1\ ...\ x_n)$ is reducible and its type is either ι or o. Hence it is strongly terminating and so is t. (2) If x is a variable of type T or an individual symbol of type T different from S and K, then let u_1, ..., u_n be reducible terms of types $U_1, ..., U_n$. By induction hypothesis the terms $u_1, ..., u_n$ are strongly terminating. A reduction sequence starting from $(x\ u_1\ ...\ u_n)$ reduces redexes in the terms $u_1, ..., u_n$. Hence, it is finite. The term $(x\ u_1\ ...\ x_n)$ is strongly terminating and its type is ι or o, hence it is reducible. Thus, x is reducible.

Then, we prove by induction over the height of t that every term is reducible.

- If t is a variable or an individual symbol different from S and K then it is reducible.
- If $t = (u\ v)$, then the terms u and v are reducible by induction hypothesis, and the term t is reducible.

- If $t = K$ (resp. $t = S$) then let $U_1 \to ... \to U_n \to V$ ($V = \iota$ or $V = o$) be the type of t and let $u_1, ..., u_n$ be reducible terms of types $U_1, ..., U_n$. We have to prove that the term $(K\ u_1\ ...\ u_n)$ (resp. $(S\ u_1\ ...\ u_n)$) is strongly terminating. Consider a reduction sequence $t_0, t_1, t_2, ...$ starting from the term $(K\ u_1\ ...\ u_n)$ (resp. $(S\ u_1\ ...\ u_n)$). We have to prove that this reduction sequence is finite. If the root redex is never reduced, all reductions take place in $u_1, ..., u_n$, these terms are reducible and hence strongly terminating and the reduction sequence is finite. If the root redex is reduced at step m, then the term t_m has the form $(K\ u'_1\ u'_2\ u'_3 ...\ u'_n)$ (resp. $(S\ u'_1\ u'_2\ u'_3 ...\ u'_n)$) and the term t_{m+1} is $(u'_1\ u'_3\ ...\ u'_n)$ (resp. $(u'_1\ u'_3\ (u'_2\ u'_3)\ u'_4\ ...\ u'_n)$) where u'_1 is a reduct of u_1, ..., u'_n is a reduct of u_n. The term $(u_1\ u_3\ ...\ u_n)$ (resp. $(u_1\ u_3\ (u_2\ u_3)\ u_4\ ...\ u_n)$) is reducible, hence it is strongly terminating and the term $(u'_1\ u'_3\ ...\ u'_n)$ (resp. $(u'_1\ u'_3\ (u'_2\ u'_3)\ u'_4\ ...\ u'_n)$) is strongly terminating, thus the reduction sequence $t_0, t_1, t_2, ...$ is finite. Therefore, the term K (resp. S) is reducible.

All terms are reducible, hence all terms are strongly terminating.

Proposition 3.3.3 *The rewrite system $\mathcal{T}$ is strongly terminating.*

Proof. We reduce termination in $\mathcal{T}$ to termination in the system SK. We define a translation $\|\ \|$ of the terms and the propositions of type theory into terms of type theory. In each type T, we choose a variable z_T.

- $\|x\| = x$,
- $\|S_{T,U,V}\| = S_{T,U,V}$,
  $\|K_{T,U}\| = K_{T,U}$,
- $\|(t\ u)\| = (\|t\|\ \|u\|)$,
- $\|\dot{\top}\| = \|\dot{\bot}\| = ((S\ K\ K)\ z_o)$,
  $\|\dot{\neg}\| = (S\ K\ K)$,
  $\|\dot{\wedge}\| = \|\dot{\vee}\| = \|\dot{\Rightarrow}\| = ((S\ K\ K)\ z_{o\to o\to o})$,
  $\|\dot{\forall}_T\| = \|\dot{\exists}_T\| = (S\ (S\ K\ K)\ (K\ z_T))$,
- $\|\varepsilon(t)\| = \|t\|$,
  $\|\top\| = \|\bot\| = z_o$,
  $\|\neg A\| = \|A\|$,
  $\|A \wedge B\| = \|A \vee B\| = \|A \Rightarrow B\| = (z_{o\to o\to o}\ \|A\|\ \|B\|)$,
  $\|\forall x\ A\| = \|\exists x\ A\| = \|(z_T/x)A\|$.

We check that if A rewrites in one step to B in $\mathcal{T}$, then $\|A\|$ rewrites in at least one step to $\|B\|$ in SK. If $A_0, A_1, A_2, ...$ is a reduction sequence in $\mathcal{T}$, then the sequence $\|A_0\|, \|A_1\|, \|A_2\|, ...$ is a reduction sequence in SK, thus it is finite.

Proposition 3.3.4 *The rewrite system $\mathcal{T}$ is confluent.*

Proposition 3.3.5 *Each term (resp. proposition) has a unique normal form for the rewrite system $\mathcal{T}$ and the congruence generated by this system can be checked in an algorithmic way.*

Proof. It is terminating and confluent.

Proposition 3.3.6 *Type theory has a model.*

Proof. Consider the model

$$\begin{array}{rcl} M_\iota & = & \{0\} \\ M_o & = & \{0,1\} \\ M_{T\to U} & = & M_U^{M_T} \end{array}$$

$$\begin{array}{rcl} \hat{S}_{T,U,V} & = & a \mapsto (b \mapsto (c \mapsto a(c)(b(c)))) \\ \hat{K}_{T,U} & = & a \mapsto (b \mapsto a) \\ \hat{\alpha}(a,b) & = & a(b) \\ \hat{\varepsilon}(a) & = & a \end{array}$$

$$\begin{array}{rcl} \hat{\dot{\top}} & = & 1 \\ \hat{\dot{\bot}} & = & 0 \\ \hat{\dot{\neg}}(a) & = & 1 \text{ if } a = 0 \text{ and } 0 \text{ otherwise} \\ \hat{\dot{\wedge}}(a,b) & = & 1 \text{ if } a = 1 \text{ and } b = 1 \text{ and } 0 \text{ otherwise} \\ \hat{\dot{\vee}}(a,b) & = & 1 \text{ if } a = 1 \text{ or } b = 1 \text{ and } 0 \text{ otherwise} \\ \hat{\dot{\Rightarrow}}(a,b) & = & 1 \text{ if } a = 0 \text{ or } b = 1 \text{ and } 0 \text{ otherwise} \\ \hat{\dot{\forall}}_T(a) & = & 1 \text{ if for all } b \text{ in } M_T \ a(b) = 1 \text{ and } 0 \text{ otherwise} \\ \hat{\dot{\exists}}_T(a) & = & 1 \text{ if there exists a } b \text{ in } M_T \text{ such that } a(b) = 1 \text{ and } 0 \text{ otherwise} \end{array}$$

It is easy to check that $|A|_\phi = |B|_\phi$ when $A \equiv B$.

3.4 Infinity

A set is said E to be infinite if there is function f mapping elements of E to elements of E that is injective, but not surjective. In type theory this proposition $Infinite(E)$ is expressed as follows.

$$\begin{array}{l} \exists a\ \exists f\ \forall x\ (\varepsilon(E\ x) \Rightarrow (E\ (f\ x))) \wedge \forall x\ \forall y\ ((\varepsilon(E\ x) \wedge \varepsilon(E\ y) \\ \qquad\qquad \wedge \varepsilon((f\ x) \doteq (f\ y))) \Rightarrow \varepsilon(x \doteq y)) \wedge (\forall x\ (\varepsilon(E\ x) \Rightarrow \neg\varepsilon(a \doteq (f\ x)))) \end{array}$$

Notice that the proposition $\exists E\ Infinite(E)$ is not valid in the model of proposition 3.3.6, hence it is not provable. If we replace M_ι by the set $\mathbb{N}$ in the model of proposition 3.3.6, we keep a model of type theory and the proposition $\exists E\ Infinite(E)$ is valid in this model. Thus, the proposition $\neg\exists E\ Infinite(E)$ is not valid in this model and therefore it is not provable either. Indeed, so far neither in type theory nor in set theory we have given an axiom that permits to

construct an infinite set. To be able to formalize mathematics we need to add such an axiom.

In type theory, we add an axiom expressing that the set of objects of type ι is infinite. Thus, the set E is such that $\varepsilon(E\ x) \equiv \top$ and we can formulate the axiom

$$\exists a\ \exists f\ \forall x\ \forall y\ (\varepsilon((f\ x)\doteq(f\ y)) \Rightarrow \varepsilon(x\doteq y)) \wedge (\forall x\ \neg\varepsilon(a\doteq(f\ x)))$$

Instead of taking an existential axiom, we can give a name to the function and to the element that is not in its image. For instance, we can call them Su and 0 and we get the two axioms

$$\forall x\ \forall y\ (\varepsilon((Su\ x)\doteq(Su\ y)) \Rightarrow \varepsilon(x\doteq y))$$

$$\forall x\ \neg\varepsilon(0\doteq(Su\ x))$$

that are two of Peano's axioms.

These axioms become theorems if we add some symbols and rewrite rules.

Definition 3.4.1 (Type theory with infinity) Type theory with infinity *is the extension of type theory with individual symbols* 0 *of type* ι, Su *and* $Pred$ *of type* $\iota \to \iota$, *an individual symbol* $Null$ *of type* $\iota \to o$ *and the rules*

$$(Pred\ (Su\ x)) \longrightarrow x$$

$$(Null\ 0) \longrightarrow \dot{\top}$$

$$(Null\ (Su\ 0)) \longrightarrow \dot{\bot}$$

Excercise 3.4.1 *In simple type theory with infinity, prove the propositions*

$$\forall x\ \forall y\ (\varepsilon((Su\ x)\doteq(Su\ y)) \Rightarrow \varepsilon(x\doteq y))$$

$$\forall x\ \neg\varepsilon(0\doteq(Su\ x))$$

Proposition 3.4.1 *Type theory with infinity has a model.*

Proof. Consider the model

$$\begin{aligned}
M_\iota &= \mathbb{N} \\
M_o &= \{0,1\} \\
M_{T\to U} &= M_U^{M_T} \\
\\
\hat{0} &= 0, \\
\hat{Su} &= n \mapsto n+1, \\
\hat{Pred} &= n \mapsto \text{if } n=0 \text{ then } 0 \text{ else } n-1, \\
\hat{Null} &= n \mapsto \text{if } n=0 \text{ then } 1 \text{ else } 0, \\
\hat{S}_{T,U,V} &= a \mapsto (b \mapsto (c \mapsto a(c)(b(c)))) \\
\hat{K}_{T,U} &= a \mapsto (b \mapsto a) \\
\hat{\alpha}(a,b) &= a(b) \\
\hat{\varepsilon}(a) &= a
\end{aligned}$$

$$\begin{aligned}
\hat{\top} &= 1 \\
\hat{\bot} &= 0 \\
\hat{\neg}(a) &= 1 \text{ if } a = 0 \text{ and } 0 \text{ otherwise} \\
\hat{\wedge}(a,b) &= 1 \text{ if } a = 1 \text{ and } b = 1 \text{ and } 0 \text{ otherwise} \\
\hat{\vee}(a,b) &= 1 \text{ if } a = 1 \text{ or } b = 1 \text{ and } 0 \text{ otherwise} \\
\hat{\Rightarrow}(a,b) &= 1 \text{ if } a = 0 \text{ or } b = 1 \text{ and } 0 \text{ otherwise} \\
\hat{\forall}_T(a) &= 1 \text{ if for all } b \text{ in } M_T \ a(b) = 1 \text{ and } 0 \text{ otherwise} \\
\hat{\exists}_T(a) &= 1 \text{ if there exists a } b \text{ in } M_T \text{ such that } a(b) = 1 \text{ and } 0 \text{ otherwise}
\end{aligned}$$

It is easy to check that $|A|_\phi = |B|_\phi$ when $A \equiv B$.

There are many ways to construct the natural numbers in type theory with infinity (as finite cardinals, ...). An easy way is simply to take 0 for zero and $(Su\ n)$ for the successor of n.

Then the type ι contains all the natural numbers, but possibly also other objects. The set of natural numbers can be defined as the smallest set containing 0 and closed by successor, *i.e.* as the intersection of all such sets. An object is a member of $\mathbb{N}$ if it is a member of all sets E containing 0 and closed by successor. Thus

$$\varepsilon(\mathbb{N}\ n) = \forall E\ ((\varepsilon(E\ 0) \wedge (\forall x\ (\varepsilon(E\ x) \Rightarrow \varepsilon(E\ (Su\ x))))) \Rightarrow \varepsilon(E\ n))$$

The existence of such an object given by proposition 3.3.1.

Excercise 3.4.2 *Prove the induction theorem*

$$\forall E\ (\varepsilon(E\ 0) \wedge \forall x\ (\varepsilon(E\ x) \Rightarrow \varepsilon(E\ (Su\ x)))) \Rightarrow \forall n\ (\varepsilon(\mathbb{N}\ n) \Rightarrow \varepsilon(E\ n))$$

3.5 More axioms

3.5.1 Extensionality

In mathematics, it is usual to consider that two sets that have the same elements are equal and that two functions that are point-wise equal are equal. This leads, both in set theory and in type theory to the *axiom of extensionality*. In type theory, this axiom is stated

$$\forall f\ \forall g\ ((\forall x\ \varepsilon((f\ x) \doteq (g\ x))) \Rightarrow \varepsilon(f \doteq g))$$

$$\forall x\ \forall y\ (\varepsilon(x) \Leftrightarrow \varepsilon(y)) \Rightarrow \varepsilon(x \doteq y)$$

3.5.2 Descriptions

The proposition 3.3.1 permits for instance to prove the existence of a function that adds two to its arguments, *i.e.* the proposition

$$\exists f\ \forall x\ \varepsilon((f\ x) \doteq (Su\ (Su\ x)))$$

but, it does not permit to prove the existence of a function that takes the value 1 on 1 and the value 0 anywhere else. Indeed, it can be proved that the proposition

$$\exists f\ \forall x\ ((\varepsilon(x\doteq(S u\ 0))\Rightarrow\varepsilon((f\ x)\doteq(S u\ 0)))\wedge(\neg\varepsilon(x\doteq(S u\ 0))\Rightarrow\varepsilon((f\ x)\doteq 0)))$$

has no proof in type theory.

In contrast, with the proposition 3.3.1, it is easy to prove the existence of the graph of this function, *i.e.* the proposition

$$\exists R\ \forall x\ \forall y\ (\varepsilon(R\ x\ y)\Leftrightarrow((\varepsilon(x\doteq 1)\Rightarrow\varepsilon(y\doteq 1))\wedge(\neg\varepsilon(x\doteq 1)\Rightarrow\varepsilon(y\doteq 0))))$$

and we can also prove, for instance by induction, that this relation is functional, *i.e.* that

$$\forall x\ (\varepsilon(\mathbb{N}\ x)\Rightarrow\exists^1 y\ \varepsilon(R\ x\ y))$$

But to conclude to the existence of the function we need the following axiom (descriptions axiom)

$$\forall P\ \forall Q\ (\forall x\ (\varepsilon(P\ x)\Rightarrow\exists^1 y\ \varepsilon(Q\ x\ y))\Rightarrow\exists f\ \forall x\ (\varepsilon(P\ x)\Rightarrow\varepsilon(Q\ x\ (f\ x)))$$

that relates functions and functional relations.

In set theory, functions are functional relations, thus they need no axiom to be related.

3.6 Type theory with a binder

We have seen in proposition 3.3.1 that to have a language containing the function symbols $\alpha_{T,U}$ and the individual symbols $S_{T,U,V}$ and $K_{T,U}$ and the related rewrite rules is sufficient to prove that, for each term t and variable x there is a term u not containing the variable x such that $(u\ x)\equiv t$. But, the term u is sometimes cumbersome to compute. It is more comfortable to have a symbol $\mapsto$ such that the function mapping x to t can simply be written $x\mapsto t$. The symbol $\mapsto$ is a function symbol of one argument binding one variable in its argument. When we take the symbol $\mapsto$, the symbols S and K become superfluous ($S=x\mapsto y\mapsto z\mapsto((x\ z)\ (y\ z))$, $K=x\mapsto y\mapsto x$). We thus get the following theory.

Definition 3.6.1 (Language of type theory with a binder) *The language of simple type theory with a binder is formed with*

- *a predicate symbol ε of rank $\langle o\rangle$,*
- *for each pair of type T,U, a function symbol $\alpha_{T,U}$ of rank $\langle T\to U,T,U\rangle$,*
  for each pair of types T,U a function symbol $\mapsto$ of rank $\langle\langle T,U\rangle,T\to U\rangle$,
- *individual symbols $\dot\top$ and $\dot\bot$ of sort o,*
  an individual symbol $\dot\neg$ of sort $o\to o$,
  individual symbols $\dot\wedge$, $\dot\vee$, $\dot\Rightarrow$ of sort $o\to o\to o$,
  for each type T, individual symbols $\dot\forall_T$ and $\dot\exists_T$ of type $(T\to o)\to o$.

Definition 3.6.2 (Rewrite system of type theory with a binder) *The rewrite system $\mathcal{T}'$ is defined by the rules*

$$((x \mapsto t)\ u) \longrightarrow (u/x)t$$

$$\varepsilon(\dot{\top}) \longrightarrow \top$$

$$\varepsilon(\dot{\bot}) \longrightarrow \bot$$

$$\varepsilon(\dot{\neg}\ x) \longrightarrow \neg\varepsilon(x)$$

$$\varepsilon(\dot{\wedge}\ x\ y) \longrightarrow \varepsilon(x) \wedge \varepsilon(y)$$

$$\varepsilon(\dot{\vee}\ x\ y) \longrightarrow \varepsilon(x) \vee \varepsilon(y)$$

$$\varepsilon(\dot{\Rightarrow}\ x\ y) \longrightarrow \varepsilon(x) \Rightarrow \varepsilon(y)$$

$$\varepsilon(\dot{\forall}_T\ x) \longrightarrow \forall y\ \varepsilon(x\ y)$$

$$\varepsilon(\dot{\exists}_T\ x) \longrightarrow \exists y\ \varepsilon(x\ y)$$

To prove that the rewrite system $\mathcal{T}'$ is terminating, we first focus on the first rule.

Proposition 3.6.1 (Tait's theorem with a binder) *The rewrite system*

$$((x \mapsto t)\ u) \longrightarrow (u/x)t$$

is strongly terminating.

Proof. The set $|T|$ of *reducible* terms of type T is defined by induction over the height of T.

- If T is ι or o then t is in $|T|$ if and only if it is strongly terminating.
- If $T = T_1 \to T_2$ then t is in $|T|$ if and only if it is strongly terminating and when its reduces to a term of the form $x \mapsto t'$ then for every term u in $|T_1|$, $(u/x)t'$ is in $|T_2|$.

To prove that all terms of type T are strongly terminating, we prove that all terms of type T are in $|T|$. More generally, we prove, by induction over the height of t, that if t is a term of type T, σ a substitution mapping variables of type U to elements of $|U|$, then σt is in $|T|$.

- If $t = y$, then if y is in the domain of σ then σt is in $|T|$. Otherwise, $\sigma t = y$, the variable y is normal, hence it is strongly terminating and it cannot reduce to a term of the form $x \mapsto t'$, hence it is in $|T|$.

- If $t = x \mapsto u$, then $T = T_1 \to T_2$. Modulo alphabetic equivalence, we can chose the variable x not appearing in σ, thus $\sigma t = x \mapsto \sigma u$. This term is strongly terminating because a reduction sequence issued from it can only reduce the term σu and, by induction hypothesis, this term is in $|T_2|$ and thus it is strongly terminating. Then, if σt reduces to the term $x \mapsto t'$, then t' is a reduct of σu. Let v be a term of $|T_2|$, the term $(v/x)t'$ is a reduct of $((v/x) \circ \sigma)u$, that is in $|T_2|$ by induction hypothesis. It is easy to check that $|T_2|$ is closed by reduction. Thus the term $(v/x)t'$ is in $|T_2|$.

  Hence, the term σt is in $|T|$.

- If $t = (t_1\ t_2)$ and t_1 is a term of type $U \to T$ and t_2 a term of type U. We have $\sigma t = (\sigma t_1\ \sigma t_2)$. By induction hypothesis σt_1 and σt_2 are in the sets $|U \to T|$ and $|U|$. To prove that σt is in $|T|$, we prove that if u_1 is in $|U \to T|$ and u_2 is in U then $(u_1\ u_2)$ is in $|T|$.

  The terms u_1 and u_2 are strongly terminating. Let n be the maximum length of a reduction sequence issued from u_1 and n' the maximum length of a reduction sequence issued from u_2. We prove that $(u_1\ u_2)$ is in $|T|$ by induction on $n + n'$.

  First we prove that $(u_1\ u_2)$ is strongly terminating. Consider a reduction sequence issued from this term. If the first redex is in u_1 or u_2 then we apply the induction hypothesis, otherwise the redex is at the root of the term $(u_1\ u_2)$, u_1 has the form $x \mapsto u'$ and the first step of the reduction sequence reduces $(u_1\ u_2)$ to $(u_2/x)u'$. This term is in $|T|$, hence it is strongly terminating and the reduction sequence is finite. Then, we prove that if $T = U_1 \to U_2$ and $(u_1\ u_2)$ reduces to a term of the form $y \mapsto v$, then for every term w in $|U_1|$, $(w/y)v$ is in $|U_2|$. As $(u_1\ u_2)$ is an application, the reduction sequence is not empty. If the first redex is in u_1 or u_2, we apply the induction hypothesis, otherwise the redex is at the root of the term $(u_1\ u_2)$, u_1 has the form $x \mapsto u'$ and the first step of the reduction sequence reduces $(u_1\ u_2)$ to $(u_2/x)u'$. This term is in $|T|$ and it reduces to $y \mapsto v$, hence for every term w in $|U_1|$, $(w/y)v$ is in $|U_2|$. Thus the term $(u_1\ u_2)$ is in $|T|$.

Proposition 3.6.2 *The rewrite system $\mathcal{T}'$ is strongly terminating.*

Proof. We follow the lines of the proof of proposition 3.3.3 and reduce termination in $\mathcal{T}'$ to termination in the system formed with the first rule. We define a translation $\| \ \|$ of the terms and the propositions of type theory into terms of type theory. In each type T, we choose a variable z_T.

- $\|x\| = x$,
- $\|x \mapsto t\| = x \mapsto \|t\|$,
- $\|(t\ u)\| = (\|t\|\ \|u\|)$,

- $\|\dot{\top}\| = \|\dot{\bot}\| = ((x \mapsto x)\ z_o)$,
  $\|\dot{\neg}\| = x \mapsto x$,
  $\|\dot{\wedge}\| = \|\dot{\vee}\| = \|\dot{\Rightarrow}\| = ((x \mapsto x)\ z_{o\to o\to o})$,
  $\|\dot{\forall}_T\| = \|\dot{\exists}_T\| = x \mapsto (x\ z_T)$,

- $\|\varepsilon(t)\| = \|t\|$,
  $\|\top\| = \|\bot\| = z_o$,
  $\|\neg A\| = \|A\|$,
  $\|A \wedge B\| = \|A \vee B\| = \|A \Rightarrow B\| = (z_{o\to o\to o}\ \|A\|\ \|B\|)$,
  $\|\forall x\ A\| = \|\exists x\ A\| = \|(z_T/x)A\|$.

We check that if A rewrites in one step to B in $\mathcal{T}$, then $\|A\|$ rewrites in at least one step to $\|B\|$ in the system formed with the first rule. If $A_0, A_1, A_2, ...$ is a reduction sequence in $\mathcal{T}$, then the sequence $\|A_0\|, \|A_1\|, \|A_2\|, ...$ is a reduction sequence in the system formed with the first rule, thus it is finite.

Proposition 3.6.3 *The rewrite system $\mathcal{T}'$ is confluent.*

Remark. If we add the axiom of extensionality to both formulations of type theory we get equivalent theories, *i.e.* each language can be translated into the other preserving provability. When we do not take the extensionality axioms, there are subtle differences between these theories, we shall not discuss here.

Remark. Some authors use the notation $\lambda x\ t$ for $x \mapsto t$, hence the name *λ-calculus* for this language.

Chapter 4

Cut elimination in predicate logic

4.1 Uniform proofs

A natural deduction proof built without the excluded middle rule is said to be *constructive*. The choice of this name comes from the fact that, as we shall see, from a constructive proof in the empty theory of a proposition of the form $\exists x\ A$, it is possible to compute a term t and a proof of the proposition $(t/x)A$. Such a term t is called a *witness* of the proposition $\exists x\ A$. Thus, explicitly or implicitly, a constructive existence proof contains a witness.

Conversely, from a term t and a proof of $(t/x)A$, the rule $\exists$-intro permits to build a proof of the proposition $\exists x\ A$. A proof ended by an introduction rule is said to be *uniform*. Witnesses are explicit in uniform existence proofs. Thus, it is equivalent to have a term t and a proof of $(t/x)A$ or a uniform proof of the proposition $\exists x\ A$. To prove that from a constructive proof of a proposition of the form $\exists x\ A$ we can compute a witness, we shall prove that all proofs can be transformed into uniform ones. For instance, the non uniform proof of the proposition $\exists x\ (P(x) \Rightarrow P(x))$

$$\dfrac{\dfrac{\dfrac{}{\exists x\ (P(x) \Rightarrow P(x)) \vdash \exists x\ (P(x) \Rightarrow P(x))}}{\vdash \exists x\ (P(x) \Rightarrow P(x)) \Rightarrow \exists x\ (P(x) \Rightarrow P(x))}\Rightarrow\text{-intro} \quad \dfrac{\dfrac{\dfrac{}{P(c) \vdash P(c)}}{\vdash P(c) \Rightarrow P(c)}\Rightarrow\text{-intro}}{\vdash \exists x\ (P(x) \Rightarrow P(x))}\exists\text{-intro}}{\vdash \exists x\ (P(x) \Rightarrow P(x))}\Rightarrow\text{-elim}$$

will be transformed into

$$\dfrac{\dfrac{\dfrac{}{P(c) \vdash P(c)}}{\vdash P(c) \Rightarrow P(c)}\Rightarrow\text{-intro}}{\vdash \exists x\ (P(x) \Rightarrow P(x))}\exists\text{-intro}$$

From the fact that all proofs can be transformed into uniform ones, we will deduce that

- if A is an atomic proposition then it has no proof,

- $\bot$ has no proof,
- if $\neg A$ has a proof then $\bot$ has a proof from the axiom A,
- if $A \wedge B$ has a proof then A has a proof and B has a proof,
- if $A \vee B$ has a proof then A has a proof or B has a proof,
- if $A \Rightarrow B$ has a proof then B has a proof from the axiom A,
- if $\forall x\ A$ has a proof then A has a proof,
- if $\exists x\ A$ has a proof then there is a term t such that $(t/x)A$ has a proof.

The results obtained for the case of $\top$, $\neg$, $\wedge$, $\Rightarrow$ and $\forall$ are trivial, they can simply be established with the elimination rules. The interesting results are thus for $\bot$, $\vee$ and $\exists$. The result in the case of the existential quantifier $\exists$ is the witness property. The result obtained in the case of the disjunction $\vee$ is called the *disjunction property*. The result obtained in the case of the contradiction $\bot$ is the consistency of the empty theory. Thus, like model constructions, proof transformation results permit to prove consistency and independence results.

Excercise 4.1.1 (Independence of the Excluded middle rule) *Consider a language formed with a proposition symbol P and a theory containing no axioms and no rewrite rules. Construct a model where the proposition P is not valid. Does this proposition have a proof ? Construct a model where the proposition $\neg P$ is not valid. Does this proposition have a proof ? Does the proposition $P \vee \neg P$ have a constructive proof ?*

Excercise 4.1.2 *Consider a language formed with a proposition symbol P and a theory containing no axioms and no rewrite rules. Construct a model where the proposition P is not valid. Does this proposition have a proof ? Construct a model where the proposition $\neg P$ is not valid. Does this proposition have a proof ? Does the proposition $P \vee \neg P$ have a proof (possibly using the excluded middle rule) ? Does natural deduction with the excluded middle have the disjunction property ?*

Excercise 4.1.3 *Consider a language formed with a proposition symbol P, a predicate symbol Q of one argument and two individual symbols 0 and 1 and a theory containing no axioms and no rewrite rules. Construct a model where the proposition*

$$(((Q(0) \Rightarrow Q(0)) \wedge P) \vee (Q(1) \Rightarrow Q(0) \wedge \neg P))$$

is not valid. Does this proposition have a proof ? Construct a model where the proposition

$$(((Q(0) \Rightarrow Q(1)) \wedge P) \vee (Q(1) \Rightarrow Q(1) \wedge \neg P))$$

is not valid. Does this proposition have a proof ? Does the proposition

$$\exists x\ (((Q(0) \Rightarrow Q(x)) \wedge P) \vee (Q(1) \Rightarrow Q(x) \wedge \neg P))$$

have a proof (possibly using the excluded middle) ? Does natural deduction with the excludes middle rule have the witness property ?

Remark. Some problems in mathematics have the form "Find an object x such that A". One way to solve such a problem is to prove constructively the proposition $\exists x\ A$, to transform this proof into a uniform one and to read the witness in the proof. For instance, finding the quotient of the division of 9 by 2 can be done in the following way: fist prove constructively the proposition

$$\exists q\ \exists r\ (9 = 2 \times q + r \wedge r < 2)$$

then transform this proof into a uniform one and read the witness in the proof. One advantage of proceeding this way, compared to other division algorithms, is that the result cannot be wrong. Indeed, a uniform proof of

$$\exists q\ \exists r\ (9 = 2 \times q + r \wedge r < 2)$$

not only contains the witness 4 but also a proof of the proposition

$$\exists r\ (9 = 2 \times 4 + r \wedge r < 2)$$

Of course, finding a proof of the proposition

$$\exists q\ \exists r\ (9 = 2 \times q + r \wedge r < 2)$$

may be tedious, but it is not if we prove once for all the proposition

$$\forall n\ \forall p\ (\neg(p = 0) \Rightarrow \exists q\ \exists r\ (n = p \times q + r \wedge r < p))$$

Notice that when we apply this theorem to 9 and 2 and to a proof of $\neg 2 = 0$ we get a proof of

$$\exists q\ \exists r\ (9 = 2 \times q + r \wedge r < 2)$$

that is not uniform. Thus, this proof needs to be transformed before the witness can be read. The quotient 4 is computed during this transformation. Thus cut elimination is the execution process of mathematics seen as a programming language.

4.2 Cuts and cut elimination

Definition 4.2.1 (Cut, cut free) *A cut is a proof ended with an elimination rule whose left premise is proved by an introduction rule on the same symbol. Here are the different cases*

$$\dfrac{\dfrac{\dfrac{\pi}{\Gamma, A \vdash \bot}}{\Gamma \vdash \neg A}\neg\text{-}intro \quad \dfrac{\pi'}{\Gamma \vdash A}}{\Gamma \vdash \bot}\neg\text{-}elim$$

$$\dfrac{\dfrac{\dfrac{\pi}{\Gamma \vdash A} \quad \dfrac{\pi'}{\Gamma \vdash B}}{\Gamma \vdash A \wedge B}\wedge\text{-}intro}{\Gamma \vdash A}\wedge\text{-}elim$$

$$\dfrac{\dfrac{\dfrac{\pi}{\Gamma \vdash A} \quad \dfrac{\pi'}{\Gamma \vdash B}}{\Gamma \vdash A \wedge B}\wedge\text{-}intro}{\Gamma \vdash B}\wedge\text{-}elim$$

$$\dfrac{\dfrac{\dfrac{\pi}{\Gamma \vdash A}}{\Gamma \vdash A \vee B}\vee\text{-}intro \quad \dfrac{\pi'}{\Gamma, A \vdash C} \quad \dfrac{\pi''}{\Gamma, B \vdash C}}{\Gamma \vdash C}\vee\text{-}elim$$

$$\dfrac{\dfrac{\dfrac{\pi}{\Gamma \vdash B}}{\Gamma \vdash A \vee B}\vee\text{-}intro \quad \dfrac{\pi'}{\Gamma, A \vdash C} \quad \dfrac{\pi''}{\Gamma, B \vdash C}}{\Gamma \vdash C}\vee\text{-}elim$$

$$\dfrac{\dfrac{\dfrac{\pi}{\Gamma, A \vdash B}}{\Gamma \vdash A \Rightarrow B}\Rightarrow\text{-}intro \quad \dfrac{\pi'}{\Gamma \vdash A}}{\Gamma \vdash B}\Rightarrow\text{-}elim$$

$$\dfrac{\dfrac{\dfrac{\pi}{\Gamma \vdash A}}{\Gamma \vdash \forall x\ A}\forall\text{-}intro}{\Gamma \vdash (t/x)A}\forall\text{-}elim$$

$$\dfrac{\dfrac{\dfrac{\pi}{\Gamma \vdash (t/x)A}}{\Gamma \vdash \exists x\ A}\exists\text{-}intro \quad \dfrac{\pi'}{\Gamma, A \vdash B}}{\Gamma \vdash B}\exists\text{-}elim$$

A proof contains a cut if one of its sub-trees is a cut. Otherwise it is cut free.

It is easy to check that cut free proofs in the empty theory are uniform.

Proposition 4.2.1 *In the empty theory, a cut free proof ends with an introduction rule.*

Proof. By induction over the height of the proof. The last rule cannot be an axiom rule, because the theory contains no axioms. If the last rule is an elimination, then the left premise of the elimination is proved with a cut free proof. Hence it ends by an introduction and the proof is a cut contradicting the fact that it is cut free.

Thus to prove that all proofs can be transformed into uniform ones we will prove that all proofs can be transformed into cut free ones. To do so, we define a process that eliminates cuts step by step. A cut of the form

$$\dfrac{\dfrac{\dfrac{\pi}{\Gamma, A \vdash \bot}}{\Gamma \vdash \neg A}\neg\text{-intro} \quad \dfrac{\pi'}{\Gamma \vdash A}}{\Gamma \vdash \bot}\neg\text{-elim}$$

is replaced by the proof obtained this way: in the proof π we suppress the hypothesis A in all sequents, then each time the axiom rule is used with this

proposition, we replace it with the proof π'. A cut of the form

$$\dfrac{\dfrac{\dfrac{\pi}{\Gamma \vdash A} \quad \dfrac{\pi'}{\Gamma \vdash B}}{\Gamma \vdash A \wedge B}\,\wedge\text{-intro}}{\Gamma \vdash A}\,\wedge\text{-elim}$$

is replaced by the proof π. A cut of the form

$$\dfrac{\dfrac{\dfrac{\pi}{\Gamma \vdash A} \quad \dfrac{\pi'}{\Gamma \vdash B}}{\Gamma \vdash A \wedge B}\,\wedge\text{-intro}}{\Gamma \vdash B}\,\wedge\text{-elim}$$

is replaced by the proof π'. A cut of the form

$$\dfrac{\dfrac{\dfrac{\pi}{\Gamma \vdash A}}{\Gamma \vdash A \vee B}\,\vee\text{-intro} \quad \dfrac{\pi'}{\Gamma, A \vdash C} \quad \dfrac{\pi''}{\Gamma, B \vdash C}}{\Gamma \vdash C}\,\vee\text{-elim}$$

is replaced by the proof obtained this way: in the proof π' we suppress the hypothesis A in all sequents, then each time the axiom rule is used with this proposition, we replace it by the proof π. A cut of the form

$$\dfrac{\dfrac{\dfrac{\pi}{\Gamma \vdash B}}{\Gamma \vdash A \vee B}\,\vee\text{-intro} \quad \dfrac{\pi'}{\Gamma, A \vdash C} \quad \dfrac{\pi''}{\Gamma, B \vdash C}}{\Gamma \vdash C}\,\vee\text{-elim}$$

is replaced by the proof obtained this way: in the proof π'' we suppress the hypothesis B in all sequents, then each time the axiom rule is used with this proposition, we replace it by the proof π. A cut of the form

$$\dfrac{\dfrac{\dfrac{\pi}{\Gamma, A \vdash B}}{\Gamma \vdash A \Rightarrow B}\,\Rightarrow\text{-intro} \quad \dfrac{\pi'}{\Gamma \vdash A}}{\Gamma \vdash B}\,\Rightarrow\text{-elim}$$

is replaced by the proof obtained this way: in the proof π we suppress the hypothesis A in all sequents, then each time the axiom rule is used with this proposition, we replace it with the proof π'. A cut of the form

$$\dfrac{\dfrac{\dfrac{\pi}{\Gamma \vdash A}}{\Gamma \vdash \forall x\ A}\,\forall\text{-intro}}{\Gamma \vdash (t/x)A}\,\forall\text{-elim}$$

is replaced by the proof π where the variable x is substituted by the term t everywhere. A cut of the form

$$\dfrac{\dfrac{\dfrac{\pi}{\Gamma \vdash (t/x)A}}{\Gamma \vdash \exists x\ A}\,\exists\text{-intro} \quad \dfrac{\pi'}{\Gamma, A \vdash B}}{\Gamma \vdash B}\,\exists\text{-elim}$$

is replaced by the proof obtained this way: in the proof π', we substitute the variable x by the term t everywhere, then we suppress the hypothesis $(t/x)A$ in all sequents and each time the axiom rule is used with this proposition, we replace it with the proof π.

Excercise 4.2.1 *Eliminate the cuts in the proof*

$$\dfrac{\dfrac{\exists x\ (P(x) \Rightarrow P(x)) \vdash \exists x\ (P(x) \Rightarrow P(x))}{\vdash \exists x\ (P(x) \Rightarrow P(x)) \Rightarrow \exists x\ (P(x) \Rightarrow P(x))}\Rightarrow\text{-}intro \quad \dfrac{\dfrac{P(c) \vdash P(c)}{\vdash P(c) \Rightarrow P(c)}\Rightarrow\text{-}intro}{\vdash \exists x\ (P(x) \Rightarrow P(x))}\exists\text{-}intro}{\vdash \exists x\ (P(x) \Rightarrow P(x))}\Rightarrow\text{-}elim$$

When a proof contains a cut, it is always simple to remove it, thus the cut elimination process is not difficult to define. But removing a cut may create new cuts, so the main question is that of the termination of this process.

4.3 Proofs as terms

The cut elimination process of the previous section is still cumbersome to express. This is due to the fact that we use a too cumbersome notation for natural deduction proof. The goal of this section is to introduce another notation for these proofs.

As we have seen, one of the key operations in this proof transformation process is the substitution of a variable by a term. Another key operation is the following: in a proof π of the sequent $\Gamma, A \vdash B$, remove the hypothesis A in all sequents and replace the axiom rules on this proposition by a proof π' of the sequent $\Gamma \vdash A$. To be able to express smoothly this operation, it is better to use a notation where proofs are expressed by terms containing special variables standing for proofs of the hypotheses. Thus to express a proof of a sequent $A_1, ..., A_n \vdash B$ we shall first introduce variables $\xi_1, ..., \xi_n$ standing for proofs of the propositions $A_1, ..., A_n$. If B is the proposition A_i and the sequent $A_1, ..., A_n \vdash A_i$ is proved with the axiom rule, we shall write this proof ξ_i.

Now a proof π of the sequent $\Gamma, A \vdash B$ is expressed by a term containing one variable for each proposition of Γ and a variable ξ for A and the proof obtained by removing the hypothesis A in all sequents of π and replacing the axiom rules on this proposition by a proof π' of the sequent $\Gamma \vdash A$ is simply obtained by substituting the proof π' for the variable ξ in π.

For each natural deduction rule, we introduce a function symbol. To express a proof such as

$$\dfrac{\dfrac{\pi}{\Gamma \vdash A} \quad \dfrac{\pi'}{\Gamma \vdash B}}{\Gamma \vdash A \wedge B}\wedge\text{-intro}$$

we express first the proofs π and π' as terms, then we apply the function symbol of two arguments associated to the rule $\wedge$-intro to π and π'.

In the case of the rule $\Rightarrow$-intro, we transform a proof π of the sequent $\Gamma, A \vdash B$ into one of the sequent $\Gamma \vdash A \Rightarrow B$ containing less hypotheses. The proof π

is expressed by a term containing a variable ξ standing for a proof of A. This variable must not appear in the proof of $\Gamma \vdash A \Rightarrow B$. Thus the function symbol associated to the rule $\Rightarrow$-intro must be a binder.

From now on, to simplify proofs, we shall drop the negation symbol $\neg$. Everything works for the proposition $\neg A$ as for the proposition $A \Rightarrow \bot$.

Definition 4.3.1 (Term notation for proofs) *We express proofs as terms in a language with two sorts: one for terms of the theory and the other for proof-terms. Terms of the theory will be written with Latin letters (t, u, ...) while proof-terms will be written with Greek letters (π, ...).*

- *The proof*

$$\frac{}{A_1, ..., A_n \vdash A_i}\ Axiom$$

is expressed by the term ξ_i.

- *The proof*

$$\frac{}{\Gamma \vdash \top}\ \top\text{-}intro$$

is expressed by the term I, where I is an individual symbol.

- *The proof*

$$\frac{\dfrac{\pi}{\Gamma \vdash \bot}}{\Gamma \vdash A}\ \bot\text{-}elim$$

is expressed by the term $\delta_\bot(\pi)$, where $\delta_\bot$ is a function symbol of one argument.

- *The proof*

$$\frac{\dfrac{\pi}{\Gamma \vdash A} \quad \dfrac{\pi'}{\Gamma \vdash B}}{\Gamma \vdash A \wedge B}\ \wedge\text{-}intro$$

is expressed by the term $\langle \pi, \pi' \rangle$, where $\langle , \rangle$ is a function symbol of two arguments.

- *The proof*

$$\frac{\dfrac{\pi}{\Gamma \vdash A \wedge B}}{\Gamma \vdash A}\ \wedge\text{-}elim$$

is expressed by the term $fst(\pi)$ and the proof

$$\frac{\dfrac{\pi}{\Gamma \vdash A \wedge B}}{\Gamma \vdash A}\ \wedge\text{-}elim$$

is expressed by the term $snd(\pi)$ where fst and snd are function symbols of one argument.

- *The proof*

$$\dfrac{\dfrac{\pi}{\Gamma \vdash A}}{\Gamma \vdash A \vee B} \vee\text{-}intro$$

is expressed by the term $i(\pi)$ and the proof

$$\dfrac{\dfrac{\pi}{\Gamma \vdash B}}{\Gamma \vdash A \vee B} \vee\text{-}intro$$

is expressed by the term $j(\pi)$, where i and j are function symbols of one argument.

- *The proof*

$$\dfrac{\dfrac{\pi}{\Gamma \vdash A \vee B} \quad \dfrac{\pi'}{\Gamma, A \vdash C} \quad \dfrac{\pi''}{\Gamma, B \vdash C}}{\Gamma \vdash C} \vee\text{-}elim$$

is expressed by the term $\delta(\pi, \xi\ \pi', \chi\ \pi'')$, where δ is a function symbol of three arguments binding one variable in its second argument and one in its third.

- *The proof*

$$\dfrac{\dfrac{\pi}{\Gamma, A \vdash B}}{\Gamma \vdash A \Rightarrow B} \Rightarrow\text{-}intro$$

is expressed by the term $\xi \mapsto \pi$, where $\mapsto$ is a function symbol of one argument binding one variable in its argument.

- *The proof*

$$\dfrac{\dfrac{\pi}{\Gamma \vdash A \Rightarrow B} \quad \dfrac{\pi'}{\Gamma \vdash A}}{\Gamma \vdash B} \Rightarrow\text{-}elim$$

is expressed by the term $\alpha(\pi, \pi')$, where α is a function symbol of two arguments. This term is also simply written $(\pi\ \pi')$.

- *The proof*

$$\dfrac{\dfrac{\pi}{\Gamma \vdash A}}{\Gamma \vdash \forall x\ A} \forall\text{-}intro$$

is expressed by the term $x \mapsto \pi$, where $\mapsto$ is a function symbol of one argument binding one variable in its argument.

- *The proof*

$$\dfrac{\dfrac{\pi}{\Gamma \vdash \forall x\ A}}{\Gamma \vdash (t/x)A} \forall\text{-}elim$$

is expressed by the term $\alpha(\pi, t)$ where α is a function symbol of two arguments. This term is also simply written $(\pi\ t)$.

- *The proof*

$$\dfrac{\dfrac{\pi}{\Gamma \vdash (t/x)A}}{\Gamma \vdash \exists x\ A}\ \exists\text{-}intro$$

is expressed by the term $\langle t, \pi \rangle$ *where* $\langle , \rangle$ *is a function symbol of two arguments.*

- *The proof*

$$\dfrac{\dfrac{\pi}{\Gamma \vdash \exists x\ A} \quad \dfrac{\pi'}{\Gamma, A \vdash B}}{\Gamma \vdash B}\ \exists\text{-}elim$$

is expressed by the term $\delta_\exists(\pi, x\xi\ \pi')$ *where* $\delta_\exists$ *is a function symbol of two arguments binding two variables in its second argument.*

Excercise 4.3.1 *Write the term associated to the proof*

$$\dfrac{\dfrac{\dfrac{}{\exists x\ (P(x) \Rightarrow P(x)) \vdash \exists x\ (P(x) \Rightarrow P(x))}}{\vdash \exists x\ (P(x) \Rightarrow P(x)) \Rightarrow \exists x\ (P(x) \Rightarrow P(x))}\ \Rightarrow\text{-}intro \quad \dfrac{\dfrac{\dfrac{}{P(c) \vdash P(c)}}{\vdash P(c) \Rightarrow P(c)}\ \Rightarrow\text{-}intro}{\vdash \exists x\ (P(x) \Rightarrow P(x))}\ \exists\text{-}intro}{\vdash \exists x\ (P(x) \Rightarrow P(x))}\ \Rightarrow\text{-}elim$$

Remark.(An historical note on the choice of symbols) The choice of these symbols comes from a tradition due to Brouwer, Heyting and Kolmogorov, according to which

- there is only one proof of $\top$,
- there is no proof of $\bot$,
- a proof of $A \wedge B$ is an ordered pair formed with a proof of A and a proof of B,
- a proof of $A \vee B$ is a boolean value together with a proof of A or B according to the value of the boolean,
- a proof of $A \Rightarrow B$ is a function mapping proofs of A to proofs of B,
- a proof of $\forall x\ A$ is a function mapping any object t to a proof of $(t/x)A$,
- a proof of $\exists x\ A$ is an ordered pair formed with a term t and a proof of $(t/x)A$.

Remark. (Types of proofs) If π is a proof of B under the hypothesis A then $\xi \mapsto \pi$ is a proof of $A \Rightarrow B$. As all proofs have the same sort, the proof-term $\xi \mapsto \pi$ does not have a type, but if we wanted to give a type to it, it would get the type $A' \to B'$ where A' is the type of proofs of A and B' the type of proofs of B. Thus the type of a proof would be isomorphic to the proposition proved by the proof-term. This isomorphism is called Curry-de Bruijn-Howard isomorphism. In particular it can be proved that a type contains a closed term

in the language of definition 3.3.2 or 3.6.1 if and only if this type is isomorphic to proposition that has a constructive proof.

As proof-terms have no type, there are proof-terms that are proof of no proposition. For instance, if P is a proposition symbol and ξ a variable standing for a proof of P then the proof-term $(\xi\ \xi)$ does not corresponds to any proof. The natural deduction rules are now used to express which proof-terms is a a proof of which proposition. We use a notation $\xi_1 : A_1, ..., \xi_n : A_n \vdash \pi : B$ to express that π is a proof of the sequent $A_1, ..., A_n \vdash B$ where $\xi_1, ..., \xi_n$ are the names given to the variables of standing for proofs of the propositions $A_1, ..., A_n$. The rules are the following.

Definition 4.3.2 (Deduction rules with proofs)

$$\frac{}{\Gamma \vdash \xi : A}\ \textit{Axiom if } \xi : A \in \Gamma$$

$$\frac{}{\Gamma \vdash I : \bot}\ \top\textit{-intro}$$

$$\frac{\Gamma \vdash \pi : \bot}{\Gamma \vdash \delta_\bot(\pi) : A}\ \bot\textit{-elim}$$

$$\frac{\Gamma \vdash \pi : A \quad \Gamma \vdash \pi' : B}{\Gamma \vdash \langle \pi, \pi' \rangle : A \wedge B}\ \wedge\textit{-intro}$$

$$\frac{\Gamma \vdash \pi : A \wedge B}{\Gamma \vdash \mathit{fst}(\pi) : A}\ \wedge\textit{-elim}$$

$$\frac{\Gamma \vdash \pi : A \wedge B}{\Gamma \vdash \mathit{snd}(\pi) : B}\ \wedge\textit{-elim}$$

$$\frac{\Gamma \vdash \pi : A}{\Gamma \vdash i(\pi) : A \vee B}\ \vee\textit{-intro}$$

$$\frac{\Gamma \vdash \pi : B}{\Gamma \vdash j(\pi) : A \vee B}\ \vee\textit{-intro}$$

$$\frac{\Gamma \vdash \pi : A \vee B \quad \Gamma, \xi : A \vdash \pi' : C \quad \Gamma, \chi : B \vdash \pi'' : C}{\Gamma \vdash \delta(\pi, \xi\pi', \chi\pi'') : C}\ \vee\textit{-elim}$$

$$\frac{\Gamma, \xi : A \vdash \pi : B}{\Gamma \vdash \xi \mapsto \pi : A \Rightarrow B}\ \Rightarrow\textit{-intro}$$

$$\frac{\Gamma \vdash \pi : A \Rightarrow B \quad \Gamma \vdash \pi' : A}{\Gamma \vdash (\pi\ \pi') : B}\ \Rightarrow\textit{-elim}$$

$$\frac{\Gamma \vdash \pi : A}{\Gamma \vdash x \mapsto \pi : \forall x\ A}\ \forall\textit{-intro if } x \notin FV(\Gamma)$$

$$\frac{\Gamma \vdash \pi : \forall x\ A}{\Gamma \vdash (\pi\ t) : (t/x)A}\ \forall\textit{-elim}$$

$$\frac{\Gamma \vdash \pi : (t/x)A}{\Gamma \vdash \langle t, \pi \rangle : \exists x\ A}\ \exists\text{-}intro$$

$$\frac{\Gamma \vdash \pi : \exists x\ A \quad \Gamma, \xi : A \vdash \pi' : B}{\Gamma \vdash \delta_\exists(\pi, x\xi\pi') : B}\ \exists\text{-}elim \text{ if } x \notin FV(\Gamma, B)$$

Proposition 4.3.1 *A sequent $A_1, \ldots, A_n \vdash B$ is derivable in natural deduction if and only if there exists a term π such that the judgment $\xi_1 : A_1, \ldots, \xi_n : A_n \vdash \pi : B$ is derivable in this system.*

The cut elimination rules can now be rephrased on the proof-terms

Definition 4.3.3 (Cut elimination rules)

$$fst(\langle \pi_1, \pi_2 \rangle) \longrightarrow \pi_1$$

$$snd(\langle \pi_1, \pi_2 \rangle) \longrightarrow \pi_2$$

$$\delta(i(\pi_1), \xi\pi_2, \chi\pi_3) \longrightarrow (\pi_1/\xi)\pi_2$$

$$\delta(j(\pi_1), \xi\pi_2, \chi\pi_3) \longrightarrow (\pi_1/\chi)\pi_3$$

$$((\xi \mapsto \pi_1)\ \pi_2) \longrightarrow (\pi_2/\xi)\pi_1$$

$$((x \mapsto \pi)\ t) \longrightarrow (t/x)\pi$$

$$\delta_\exists(\langle t, \pi_1 \rangle, \xi x\pi_2) \longrightarrow (t/x, \pi_1/\xi)\pi_2$$

Proposition 4.3.2 (Subject reduction) *If $\Gamma \vdash \pi : P$ and $\pi \longrightarrow \pi'$ then $\Gamma \vdash \pi' : P$.*

4.4 Cut elimination

We now want to prove that if a proof-term is a proof of some proposition then it is strongly terminating. Following the idea of Curry-de Bruijn-Howard isomorphism, this proof extends that of proposition 3.6.1.

Definition 4.4.1 (Reducible proof-terms) *Let A be a proposition. We define the set $|A|$ of reducible proof-terms of A by induction over the height of A.*

- *If A is an atomic proposition then a proof-term π is an element of $|A|$ if it is strongly terminating.*
- *A proof-term π is an element of $|\top|$ if it is strongly terminating.*
- *A proof-term π is an element of $|\bot|$ if it is strongly terminating.*
- *A proof-term π is an element of $|A \wedge B|$ if it is strongly terminating and when π reduces to a proof-term of the form $\langle \pi_1, \pi_2 \rangle$ then π_1 is an element of $|A|$ and π_2 is an element of $|B|$.*

- *A proof-term π is an element of $|A \vee B|$ if it is strongly terminating and when π reduces to a proof-term of the form $i(\pi_1)$ (resp. $j(\pi_2)$) then π_1 (resp. π_2) is an element of $|A|$ (resp. $|B|$).*

- *A proof-term π is element of $|A \Rightarrow B|$ if it is strongly terminating and when π reduces to a proof-term of the form $\xi \mapsto \pi_1$ then for every π' in $|A|$, $(\pi'/\xi)\pi_1$ is an element of $|B|$.*

- *A proof-term π is an element of $|\forall x\ A|$ if it is strongly terminating and when π reduces to a proof-term of the form $x \mapsto \pi_1$ then for every term t $(t/x)\pi_1$ is an element of $|(t/x)A|$ (which is equal to $|A|$).*

- *A proof-term π is an element of $|\exists x\ A|$ if it is strongly terminating and when π reduces to a proof-term of the form $\langle t, \pi_1 \rangle$ then π_1 is an element of $|(t/x)A|$ (which is equal to $|A|$).*

Lemma 4.4.1 *Elements of $|A|$ are strongly terminating.*

Proof. By definition.

Lemma 4.4.2 *If π is an element of $|A|$ and $\pi \longrightarrow \pi'$ then π' is an element of $|A|$.*

Proof. By definition.

Lemma 4.4.3 *All variables are members of $|A|$.*

Proof. By definition.

Lemma 4.4.4 *If π is an elimination and if for every π' such that $\pi \longrightarrow^1 \pi'$, $\pi' \in |A|$ then $\pi \in |A|$.*

Proof. We first prove that π is strongly terminating. Let $\pi = \pi_1, \pi_2, \ldots$ be a reduction sequence issued from π. If this sequence is empty it is finite. Otherwise we have $\pi \longrightarrow^1 \pi_2$ and hence π_2 is an element of $|A|$ thus it is strongly terminating and the reduction sequence is finite.

Then, we prove that if π reduces to an introduction then the sub-terms belong to the appropriate sets. Let $\pi = \pi_1, \pi_2, \ldots \pi_n$ be a reduction sequence issued from π and such that π_n is an introduction. This sequence cannot be empty because π is an elimination. Thus $\pi \longrightarrow^1 \pi_2 \longrightarrow \pi_n$. We have $\pi_2 \in |A|$ and thus if π_n is an introduction the sub-terms belong to the appropriate sets.

Proposition 4.4.5 (Gentzen-Prawitz theorem) *If $\Gamma \vdash \pi : A$ then the proof-term π is strongly terminating.*

Proof. By lemma 4.4.1, it is sufficient to prove that if $\Gamma \vdash \pi : A$ then the proof-term π is an element of $|A|$. More generally, we prove, by induction over the height of the proof-assignment tree, that if $\Gamma \vdash \pi : A$, θ is a substitution mapping the term variable to terms and σ is a substitution mapping some proof variables associated to a proposition B in Γ to an element of $|B|$, then $\sigma\theta\pi$ is an element of $|A|$.

- Axiom. If π is a variable ξ, we have $(\xi : A) \in \Gamma$. If ξ is in the domain of definition of σ, then $\sigma\theta\xi = \sigma\xi$ is an element of $|A|$, otherwise $\sigma\theta\xi = \sigma\xi = \xi$ is an element of $|A|$ by proposition 4.4.3.

- $\top$-intro. The proof-term π has the form I. We have $\sigma\theta\pi = I$. This proof-term is normal and thus it is strongly terminating. Hence, the proof-term $\sigma\theta I$ is in $|A|$.

- $\wedge$-intro. The proof-term π has the form $\langle\rho_1, \rho_2\rangle$ where ρ_1 is a proof of some proposition B and ρ_2 a proof of some proposition C. We have $\sigma\theta\pi = \langle\sigma\theta\rho_1, \sigma\theta\rho_2\rangle$. Consider a reduction sequence issued from this proof-term. This sequence can only reduce the proof-terms $\sigma\theta\rho_1$ and $\sigma\theta\rho_2$. By induction hypothesis these proof-terms are in $|B|$ and $|C|$. Thus the reduction sequence is finite.

  Furthermore, all reducts of $\sigma\theta\pi$ have the form $\langle\rho_1', \rho_2'\rangle$ where ρ_1' is a reduct of $\sigma\theta\rho_1$ and ρ_2' one of $\sigma\theta\rho_2$. The proof-terms ρ_1' and ρ_2' are in $|B|$ and $|C|$ by proposition 4.4.2.

  Hence, the proof-term $\sigma\theta\langle\rho_1, \rho_2\rangle$ is in $|A|$.

- $\vee$-intro. The proof-term π has the form $i(\rho)$ (resp. $j(\rho)$) and ρ is a proof of some proposition B. We have $\sigma\theta\pi = i(\sigma\theta\rho)$ (resp. $j(\sigma\theta\rho)$). Consider a reduction sequence issued from this proof-term. This sequence can only reduce the proof-terms $\sigma\theta\rho$. By induction hypothesis this proof-term is an element of $|B|$. Thus the reduction sequence is finite.

  Furthermore, all reducts of $\sigma\theta\pi$ have the form $i(\rho')$ (resp. $j(\rho')$) where ρ' is a reduct of $\sigma\theta\rho$. The proof-term ρ' is an element of $|B|$ by proposition 4.4.2.

  Hence, the proof-term $\sigma\theta i(\rho)$ (respectively $\sigma\theta j(\rho)$) is an element of $|A|$.

- $\Rightarrow$-intro. The proof-term π has the form $\xi \mapsto \rho$ where ξ is a proof variable of some proposition B and ρ a proof of some proposition C. We have $\sigma\theta\pi = \xi \mapsto \sigma\theta\rho$, consider a reduction sequence issued from this proof-term. This sequence can only reduce the proof-term $\sigma\theta\rho$. By induction hypothesis, the proof-term $\sigma\theta\rho$ is an element of $|C|$, thus the reduction sequence is finite.

  Furthermore, all reducts of $\sigma\theta\pi$ have the form $\xi \mapsto \rho'$ where ρ' is a reduct of $\sigma\theta\rho$. Let τ be any proof of $|B|$, the proof-term $(\tau/\xi)\rho'$ can be obtained by reduction from $((\tau/\xi) \circ \sigma)\theta\rho$. By induction hypothesis, the proof-term

$((\tau/\xi) \circ \sigma)\theta\rho$ is an element of $|C|$. The proof term $(\tau/\xi)\rho'$ is an element of $|C|$, by proposition 4.4.2.

Hence, the proof-term $\sigma\theta(\xi \mapsto \rho)$ is an element of $|A|$.

- $\forall$-intro. The proof-term π has the form $x \mapsto \rho$ where ρ is a proof of some proposition B. We have $\sigma\theta\pi = x \mapsto \sigma\theta\rho$. Consider a reduction sequence issued from the proof-term $\sigma\theta\pi = x \mapsto \sigma\theta\rho$. This sequence can only reduce the proof-term $\sigma\theta\rho$. By induction hypothesis, the proof-term $\sigma\theta\rho$ is an element of $|B|$, thus the reduction sequence is finite.

  Furthermore, all reducts of $\sigma\theta\pi$ have the form $x \mapsto \rho'$ where ρ' is a reduct of $\sigma\theta\rho$. The proof-term $(t/x)\rho'$ is obtained by reducing the proof-term $((t/x)\sigma)((t/x) \circ \theta)\rho$. By induction hypothesis again, the proof-term $((t/x)\sigma)((t/x) \circ \theta)\rho$ is an element of $|B|$. The proof-term $(t/x)\rho'$ is an element of $|B|$, by proposition 4.4.2.

  Hence $\sigma\theta(x \mapsto \rho)$ is an element of $|A|$.

- $\exists$-intro. The proof-term π has the form $\langle t, \rho\rangle$, where ρ is a proof of some proposition B. We have $\sigma\theta\pi = \langle \theta t, \sigma\theta\rho\rangle$. Consider a reduction sequence issued from this proof-term. This sequence can only reduce the proof-term $\sigma\theta\rho$. By induction hypothesis this proof-term is in $|B|$. Thus the reduction sequence is finite.

  Furthermore, all reducts of $\sigma\theta\pi$ have the form $\langle \theta t, \rho'\rangle$ where ρ' is a reduct of $\sigma\theta\rho$. The proof-term ρ' is an element of $|B|$, by proposition 4.4.2.

  Hence, the proof-term $\sigma\theta\langle t, \rho\rangle$ is an element of $|A|$.

- $\bot$-elim. The proof-term π has the form $\delta_\bot(\rho)$ where ρ is a proof of $\bot$. We have $\sigma\theta\pi = \delta_\bot(\sigma\theta\rho)$. By induction hypothesis, the proof-term $\sigma\theta\rho$ is an element of $|\bot|$. Hence, it is strongly terminating. Let n be the maximum length of reduction sequences issued from this proof-term. We prove by induction on n that $\delta_\bot(\sigma\theta\rho)$ is in $|A|$. Since this proof-term is an elimination, by proposition 4.4.4, we only need to prove that every of its one step reducts is in $|A|$. The reduction can only take place in $\sigma\theta\rho$ and we apply the induction hypothesis.

  Hence, the proof-term $\sigma\theta\delta_\bot(\rho)$ is an element of $|A|$.

- $\wedge$-elim. We only detail the case of left elimination. The proof-term π has the form $fst(\rho)$ where ρ is a proof of some proposition $A \wedge B$. We have $\sigma\theta\pi = fst(\sigma\theta\rho)$. By induction hypothesis the proof-term $\sigma\theta\rho$ is in $|A \wedge B|$. Hence, it is strongly terminating. Let n be the maximum length of a reduction sequence issued from this proof-term. We prove by induction on n that $fst(\sigma\theta\rho)$ is in the set $|A|$. Since this proof-term is a elimination, by proposition 4.4.4, we only need to prove that every of its one step reducts is in $|B|$. If the reduction takes place in $\sigma\theta\rho$ then we apply the induction hypothesis. Otherwise $\sigma\theta\rho$ has the form $\langle \rho'_1, \rho'_2\rangle$ and the reduct is ρ'_1. By the definition of $|A \wedge B|$ this proof-term is in $|A|$.

  Hence, the proof-term $\sigma\theta fst(\rho)$ is an element of $|A|$.

- $\vee$-elim. The proof-term π has the form $\delta(\rho_1, \xi\rho_2\ \chi\rho_3)$ where ρ_1 is a proof of some proposition $B \vee C$ and ρ_2 and ρ_3 are proofs of A. We have $\sigma\theta\pi = \delta(\sigma\theta\rho_1, \xi\sigma\theta\rho_2, \chi\sigma\theta\rho_3)$. By induction hypothesis, the proof-term $\sigma\theta\rho_1$ is in the set $|B \vee C|$, and the proof-terms $\sigma\theta\rho_2$ and $\sigma\theta\rho_3$ are in the set $|A|$. Hence, these proof-terms are strongly terminating. Let n, n' and n'' be the maximum length of reduction sequences issued from these proof-terms. We prove by induction on $n + n' + n''$ that $\delta(\sigma\theta\rho_1, \sigma\theta\xi\rho_2, \chi\sigma\theta\rho_3)$ is in $|A|$. Since this proof-term is an elimination, by proposition 4.4.4, we only need to prove that every of its one step reducts is in $|A|$. If the reduction takes place in $\sigma\theta\rho_1$, $\sigma\theta\rho_2$ or $\sigma\theta\rho_3$ then we apply the induction hypothesis. Otherwise, if $\sigma\theta\rho_1$ has the form $i(\rho')$ (resp. $j(\rho')$) and the reduct is $((\rho'/\xi)\circ\sigma)\theta\rho_2$ (resp. $((\rho'/\chi)\circ\sigma)\theta\rho_3$). By the definition of $|B \vee C|$ the proof-term ρ' is in $|B|$ (resp. $|C|$). Hence by induction hypothesis $((\rho'/\xi) \circ \sigma)\theta\rho_2$ (resp. $((\rho'/\chi) \circ \sigma)\theta\rho_3$) is in $|A|$.

  Hence, the proof-term $\sigma\theta\delta(\rho_1, \xi\rho_2, \chi\rho_3)$ is an element of $|A|$.

- $\Rightarrow$-elim. The proof-term π has the form $(\rho_1\ \rho_2)$ and ρ_1 is a proof of some proposition $B \Rightarrow A$ and ρ_2 a proof of the proposition B. We have $\sigma\theta\pi = (\sigma\theta\rho_1\ \sigma\theta\rho_2)$. By induction hypothesis $\sigma\theta\rho_1$ and $\sigma\theta\rho_2$ are in the sets $|B \Rightarrow A|$ and $|B|$. Hence these proof-terms are strongly terminating. Let n be the maximum length of a reduction sequence issued from $\sigma\theta\rho_1$ and n' the maximum length of a reduction sequence issued from $\sigma\theta\rho_2$. We prove by induction on $n + n'$ that $(\sigma\theta\rho_1\ \sigma\theta\rho_2)$ is in the set $|A|$. Since this proof-term is an elimination, by proposition 4.4.4, we only need to prove that every of its one step reducts is in $|A|$. If the reduction takes place in $\sigma\theta\rho_1$ or in $\sigma\theta\rho_2$ then we apply the induction hypothesis. Otherwise $\sigma\theta\rho_1$ has the form $\xi \mapsto \rho'$ and the reduct is $(\sigma\theta\rho_2/\xi)\rho'$. By the definition of $|B \Rightarrow A|$ this proof-term is in $|A|$.

  Hence, the proof-term $\sigma\theta(\rho_1\ \rho_2)$ is an element of $|A|$.

- $\forall$-elim. The proof-term π has the form $(\rho\ t)$ where ρ is a proof of some proposition $\forall x\ B$ and $A = (t/x)B$. We have $\sigma\theta\pi = (\sigma\theta\rho\ \theta t)$. By induction hypothesis, the proof-term $\sigma\theta\rho$ is in $|\forall x\ B|$. Hence, it is strongly terminating. Let n be the maximum length of a reduction sequence issued from this proof-term. We prove by induction on n that $(\sigma\theta\rho\ \theta t)$ is in the set $|A|$. As this proof-term is an elimination, by proposition 4.4.4, we only need to prove that every of its one step reducts is in $|A|$. If the reduction takes place in $\sigma\theta\rho$ then we apply the induction hypothesis. Otherwise $\sigma\theta\rho$ has the form $x \mapsto \rho'$ and the reduct is $(\theta t/x)\rho'$. By the definition of $|\forall x\ B|$ this proof-term is in $|A|$.

  Hence, the proof-term $\sigma\theta(\rho\ t)$ is an element of $|A|$.

- $\exists$-elim. The proof-term π has the form $\delta_\exists(\rho_1, x\xi\rho_2)$ where ρ_1 is a proof of some proposition $\exists x\ B$ and ρ_2 is a proof of A. We have $\sigma\theta\pi = \delta_\exists(\sigma\theta\rho_1, x\xi\sigma\theta\rho_2)$. By induction hypothesis, the proof-term $\sigma\theta\rho_1$ is in the

set $|\exists x\ B|$ and the proof-term $\sigma\theta\rho_2$ is in the set $|A|$. Hence, these proof-terms are strongly terminating. Let n and n' be the maximum length of reduction sequences issued from these proof-terms. We prove by induction on $n+n'$ that $\delta_\exists(\sigma\theta\rho_1, x\xi\sigma\theta\rho_2)$ is in $|A|$. As this proof-term is an elimination, by proposition 4.4.4, we only need to prove that every of its one step reducts is in $|A|$. If the reduction takes place in $\sigma\theta\rho_1$ or $\sigma\theta\rho_2$ then we apply the induction hypothesis. Otherwise, $\sigma\theta\rho_1$ has the form $\langle t, \rho'\rangle$ and the reduct is $(\rho'/\xi)(t/x)\sigma\theta\rho_2 = ((\rho'/\xi)\circ(t/x)\sigma)((t/x)\circ\theta)\rho_2$. By the definition of $|\exists x\ B|$, the proof-term ρ' is in $|B|$. Thus, by induction hypothesis, the proof-term $((\rho'/\xi)\circ(t/x)\sigma)((t/x)\circ\theta)\rho_2$ is in $|A|$.

Hence, the proof-term $\sigma\theta\delta_\exists(\rho_1, \xi x\rho_2)$ is an element of $|A|$.

4.5 Harrop theories

We have seen that constructive cut free proofs in the empty theory are uniform, and we have deduced the disjunction property and the witness property for the empty theory. Of course these properties do not extend to all theories, but they extended to *Harrop theories.*

Definition 4.5.1 (Harrop theory) *The set of* Harrop propositions *is inductively defined as follows:*

- *atomic propositions,* $\top$ *and* $\bot$ *are Harrop propositions,*
- $\neg A$ *is a Harrop proposition,*
- $A \wedge B$ *is a Harrop proposition if* A *and* B *are Harrop propositions,*
- $A \Rightarrow B$ *is a Harrop proposition if* B *is a Harrop proposition,*
- $\forall x\ A$ *is a Harrop proposition if* A *is a Harrop proposition,*

A Harrop theory *is a theory whose axioms are all Harrop propositions.*

Proposition 4.5.1 *Let* Γ *be a Harrop theory. If* $A \vee B$ *has a constructive proof in* Γ*, then* A *or* B *has a proof in* Γ *and this proof is constructive. If* $\exists x\ A$ *has a constructive proof in* Γ*, then there is a term* t *such that* $(t/x)A$ *has a proof in* Γ *and this proof is constructive.*

Proof. By induction over the height of the proof.

If the proofs ends with an introduction, then the result is trivial.

The proof cannot end with an axiom because Γ contains only Harrop propositions and the conclusion is not a Harrop proposition.

We prove now that if the proof ends with an elimination then the theory Γ is contradictory and hence the result is trivial.

Let C_1 be the conclusion of the proof and C_2 be the left premise of this elimination, the proof of C_2 cannot end with an introduction because the proof

is cut free, hence it ends with an axiom rule or an elimination, if it ends with an elimination rule, then let C_3 be the left premise of this rule, ... Thus the rule ends with a sequence of elimination rules on propositions $C_1, ..., C_n$ and C_n is an axiom.

We prove that at least one of the propositions $C_1, ..., C_n$ is $\bot$. Assume this is not the case. Then the proposition C_n is a Harrop proposition because it is an element of Γ. Let us prove that the proposition C_{n-1} is also a Harrop proposition. The proposition C_{n-1} has been produced from C_n with an elimination rule. This elimination rule cannot be $\vee$-elim or $\exists$-elim because C_n is a Harrop proposition, it cannot be $\bot$-elim, because none of the propositions $C_1, ..., C_n$ is $\bot$. Hence it is either $\wedge$-elim, $\Rightarrow$-elim or $\forall$-elim, thus C_{n-1} is a Harrop proposition. We prove this way by induction that all the propositions $C_n, ..., C_1$ are Harrop propositions. Hence C_1 is a Harrop proposition which is contradictory.

Thus one of the propositions $C_1, ..., C_n$ is $\bot$, thus the theory Γ is contradictory, it proves all propositions and the result is trivial.

Excercise 4.5.1 *Show that proofs of propositions of the form $A \vee B$ and $\exists x\ A$ in consistent Harrop theories end with an introduction rule.*

Corollary 4.5.2 *Let P and Q be two proposition symbols, the proposition*

$$\neg\neg(P \vee Q) \Rightarrow (P \vee Q)$$

does not have a constructive proof in the empty theory.

Proof. Assume that the proposition $\neg\neg(P \vee Q) \Rightarrow (P \vee Q)$ has a proof. Let Γ be the Harrop theory formed with the axiom $\neg\neg(P \vee Q)$, the proposition $P \vee Q$ has a proof in Γ. Thus either the proposition P or the proposition Q has proof in Γ and it is easy to construct a model of Γ where P is not valid and a model of Γ where Q is not valid.

Corollary 4.5.3 *Let P be a proposition symbol, the proposition*

$$\neg\neg P \Rightarrow P$$

does not have a constructive proof in the empty theory.

Proof. If it had, so would the proposition. $\neg\neg(P \vee Q) \Rightarrow (P \vee Q)$.

Corollary 4.5.4 *Let P be a predicate symbol of one argument, the proposition*

$$(\neg\forall x\ P(x)) \Rightarrow \exists x\ \neg P(x)$$

does not have a constructive proof in the empty theory.

Proof. Assume that the proposition $(\neg\forall x\ P(x)) \Rightarrow \exists x\ \neg P(x)$ has a proof. Let Γ be the Harrop theory formed with the axiom $\neg\forall x\ P(x)$. Then the proposition $\exists x\ \neg P(x)$ has a proof in Γ. Thus there is a term t such that the proposition $\neg P(t)$ has a proof in Γ. Consider a model $\mathcal{M}$ with two elements and let $\hat{P}$ hold form the denotation of t but not for the other element. This model is a model of Γ but not of $\neg P(t)$. Thus, the proposition $\neg P(t)$ does not have a proof in Γ which is contradictory.

Chapter 5

Cut elimination in predicate logic modulo

We have seen that from the cut elimination theorem we could deduce the consistency, the disjunction property and the witness property for the empty theory. Of course, not many theorems can be proved in the empty theory. When we add axioms, cut free proofs need not be uniform anymore. For instance adding the axiom $\exists x\ P(x)$, allows a non uniform proof of the proposition $\exists x\ P(x)$. We have already seen that the disjunction property and the witness property extended to Harrop theories. We are now interested in other theories: theories modulo with no axioms, such as simple type theory and simple type theory with infinity.

5.1 Congruences defined by a system rewriting atomic propositions

Proposition 5.1.1 *Consider a congruence $\equiv$ defined by a confluent rewrite system rewriting terms to terms and atomic propositions to arbitrary propositions. If A and B are not atomic and $A \equiv B$ then A and B have the same root connector or quantifier.*

Proposition 5.1.2 *Consider a congruence $\equiv$ defined by a confluent rewrite system rewriting terms to terms and atomic propositions to arbitrary propositions. Consider the theory modulo formed with no axioms and the congruence $\equiv$. A cut free proof in this theory ends with an introduction rule.*

Proof. By induction over the height of the proof. The last rule cannot be an axiom rule, because there is no axiom. If the last rule is an elimination, then the left premise of the elimination is proved with a cut free proof. Hence it ends by an introduction. By proposition 5.1.2, this introduction concerns

the same connector or quantifier as the elimination rule and the proof is a cut contradicting the fact that it is cut free.

Thus, if cut elimination holds for such a theory, then consistency, the disjunction property and the witness property also.

5.2 Proof as terms

Proof-terms are defined as in predicate logic and the reduction rules are the same. But the proof assignments rules have to be modified to take the congruence into account.

Definition 5.2.1 (Deduction rules with proofs)

$$\frac{}{\Gamma \vdash_{\equiv} \xi : B}\ \textit{Axiom if } \xi : A \in \Gamma \textit{ and } A \equiv B$$

$$\frac{}{\Gamma \vdash_{\equiv} I : A}\ \top\textit{-intro if } A \equiv \top$$

$$\frac{\Gamma \vdash_{\equiv} \pi : B}{\Gamma \vdash_{\equiv} \delta_{\bot}(\pi) : A}\ \bot\textit{-elim if } B \equiv \bot$$

$$\frac{\Gamma \vdash_{\equiv} \pi : A \quad \Gamma \vdash_{\equiv} \pi' : B}{\Gamma \vdash_{\equiv} \langle \pi, \pi' \rangle : C}\ \wedge\textit{-intro if } C \equiv (A \wedge B)$$

$$\frac{\Gamma \vdash_{\equiv} \pi : C}{\Gamma \vdash_{\equiv} \mathit{fst}(\pi) : A}\ \wedge\textit{-elim if } C \equiv (A \wedge B)$$

$$\frac{\Gamma \vdash_{\equiv} \pi : C}{\Gamma \vdash_{\equiv} \mathit{snd}(\pi) : B}\ \wedge\textit{-elim if } C \equiv (A \wedge B)$$

$$\frac{\Gamma \vdash_{\equiv} \pi : A}{\Gamma \vdash_{\equiv} i(\pi) : C}\ \vee\textit{-intro if } C \equiv (A \vee B)$$

$$\frac{\Gamma \vdash_{\equiv} \pi : B}{\Gamma \vdash_{\equiv} j(\pi) : C}\ \vee\textit{-intro if } C \equiv (A \vee B)$$

$$\frac{\Gamma \vdash_{\equiv} \pi : D \quad \Gamma, \xi : A \vdash_{\equiv} \pi' : C \quad \Gamma, \chi : B \vdash_{\equiv} \pi'' : C}{\Gamma \vdash_{\equiv} \delta(\pi, \xi\pi', \chi\pi'') : C}\ \vee\textit{-elim if } D \equiv (A \vee B)$$

$$\frac{\Gamma, \xi : A \vdash_{\equiv} \pi : B}{\Gamma \vdash_{\equiv} \xi \mapsto \pi : C}\ \Rightarrow\textit{-intro if } C \equiv (A \Rightarrow B)$$

$$\frac{\Gamma \vdash_{\equiv} \pi : C \quad \Gamma \vdash_{\equiv} \pi' : A}{\Gamma \vdash_{\equiv} (\pi\ \pi') : B}\ \Rightarrow\textit{-elim if } C \equiv (A \Rightarrow B)$$

$$\frac{\Gamma \vdash_{\equiv} \pi : A}{\Gamma \vdash_{\equiv} x \mapsto \pi : B}\ \langle x, A \rangle\ \forall\textit{-intro if } B \equiv (\forall x\ A) \textit{ and } x \notin FV(\Gamma)$$

$$\frac{\Gamma \vdash_{\equiv} \pi : B}{\Gamma \vdash_{\equiv} (\pi\ t) : C}\ \langle x, A, t \rangle\ \forall\textit{-elim if } B \equiv (\forall x\ A) \textit{ and } C \equiv (t/x)A$$

$$\frac{\Gamma \vdash_{\equiv} \pi : C}{\Gamma \vdash_{\equiv} \langle t, \pi \rangle : B}\ \langle x, A, t\rangle\ \exists\text{-}intro\ if\ B \equiv (\exists x\ A)\ and\ C \equiv (t/x)A$$

$$\frac{\Gamma \vdash_{\equiv} \pi : C \quad \Gamma, \xi : A \vdash_{\equiv} \pi' : B}{\Gamma \vdash_{\equiv} \delta_{\exists}(\pi, x\xi\pi') : B}\ \langle x, A\rangle\ \exists\text{-}elim\ if\ C \equiv (\exists x\ A)\ and\ x \notin FV(\Gamma, B)$$

Proposition 5.2.1 *A sequent $A_1, \ldots, A_n \vdash_{\equiv} B$ is derivable in natural deduction modulo if and only if there exists a term π such that the judgment $\xi_1 : A_1, \ldots, \xi_n : A_n \vdash_{\equiv} \pi : B$ is derivable in this system.*

Proposition 5.2.2 (Subject reduction) *If $\Gamma \vdash_{\equiv} \pi : P$ and $\pi \longrightarrow \pi'$ then $\Gamma \vdash \pi' : P$.*

5.3 Counterexamples

Cut elimination fails for very simple rewrite systems.

Example 5.3.1 (Russell's counterexample) *We have seen that in naive set theory, if we call A the proposition $\varepsilon(R\ R)$ (or $R \in R$) we have*

$$A \longrightarrow \neg A$$

Modulo this rule, the proposition $\neg A$ has the proof

$$\xi \mapsto (\xi\ \xi)$$

and the proposition A also thus the proposition $\bot$ has the proof

$$((\xi \mapsto (\xi\ \xi))\ (\xi \mapsto (\xi\ \xi)))$$

This proof only reduces to itself and thus it does not terminate. It is easy to check that more generally, there are no cut free proofs of $\bot$ because there no uniform proofs of this proposition.

Example 5.3.2 (Crabbé's counterexample) *Set theory is an example of a theory modulo that does not have the cut elimination property. We have seen that there are two propositions A and B in set theory such that*

$$A \longrightarrow B \wedge \neg A$$

Thus under the assumption $\chi : B$, the proposition $\neg A$ has the proof

$$\xi \mapsto (snd(\xi)\ \xi)$$

and the proposition A has the proof

$$\langle \chi, \xi \mapsto (snd(\xi)\ \xi)\rangle$$

thus the proposition $\bot$ has the proof

$$((\xi \mapsto (snd(\xi)\ \xi))\ \langle \chi, \xi \mapsto (snd(\xi)\ \xi)\rangle)$$

and the proposition $\neg B$ has the proof

$$\chi \mapsto ((\xi \mapsto (snd(\xi)\ \xi))\ \langle \chi, \xi \mapsto (snd(\xi)\ \xi)))$$

It is easy to check that this proof does not terminate and more generally that the proposition $\neg B$ has no cut free proof.

Example 5.3.3 (A terminating counterexample) *Cut elimination may be lost even with a confluent and terminating rewrite system. The example is a refined version of Russell's counterexample. Instead of taking the non terminating rule $R \in R \longrightarrow \neg R \in R$, we take the terminating rule*

$$R \in R \longrightarrow \forall y\ (y \simeq R \Rightarrow \neg y \in R)$$

where $y \simeq z$ stands for $\forall x\ (y \in x \Rightarrow z \in x)$. Modulo this rule, the proposition $\neg R \in R$ has the proof

$$\pi = \xi \mapsto (\xi\ R\ (x \mapsto (\chi \mapsto \chi))\ \xi)$$

and the proposition $R \in R$ has the proof

$$\pi' = y \mapsto (\xi \mapsto (\chi \mapsto (\pi\ (\xi\ R\ \chi))))$$

The proposition $\bot$ has the proof

$$(\pi\ \pi')$$

This proof only reduces to itself and thus it does not terminate. It is easy to check that more generally, there are no cut free proofs of $\bot$ because there no uniform proofs of this proposition.

5.4 Reducibility candidates

Let us try to characterize some congruences for which cut elimination holds.

We wish to use a cut elimination proof similar to that of predicate logic. The main problem is that we cannot take the set of all strongly terminating proof-terms for the set of reducible proof-terms of an atomic proposition. For instance if P, Q and R are three proposition symbol and we have the rule

$$P \longrightarrow Q \Rightarrow R$$

then a proof of P is also a proof of $Q \Rightarrow R$ and thus, to belong to $|P|$, besides being strongly terminating, a proof-term must be such that whenever it reduces to an introduction $\xi \mapsto \pi'$ for all proof π'' of $|Q|$, the proof $(\pi''/\xi)\pi'$ belongs to $|R|$. In this case we can take the set of all strongly terminating proofs for $|Q|$ and $|R|$ and the set $|Q \Rightarrow R|$ for $|P|$ and a proof similar to that of predicate logic permits to establish cut elimination modulo this rule.

However, generalizing this method may be difficult when we have non terminating rules or rules introducing quantifiers. For instance consider the proposition symbols P and Q and the rule

$$Q \longrightarrow P \wedge Q$$

defining $|Q|$ as $|P \wedge Q|$ would be circular, as to know $|P \wedge Q|$ we need to know $|P|$ and $|Q|$. In the same way, consider a predicate symbol P of one argument, an individual symbol c and the rule

$$P(c) \longrightarrow \forall x\ P(x)$$

Defining $|P(c)|$ as the set $|\forall x\ P(x)|$ would be circular as to know $|\forall x\ P(x)|$ we need to know $|P(t)|$ for all terms t, including c.

Thus we shall prove in a first step that cut elimination holds provided we know how to assign a set of proofs $|A|$ to each atomic proposition A in such a way that the sets of reducible proofs - defined relatively to these sets - of two equivalent propositions are identical. In a second step we shall give examples where such sets can be constructed including the two examples above and simple type theory.

Not any set of proof-terms is a good candidate for $|A|$. Indeed, we have seen that to let the cut elimination proof go through we needed the sets of reducible proofs to verify the properties of propositions 4.4.1, 4.4.2, 4.4.3 and 4.4.4 that are used in the cut elimination proof. Thus, at least, the sets of reducible proofs of atomic propositions must verify these properties. This leads to the following definition.

Definition 5.4.1 (Girard's reducibility candidate) *A set R of proof-terms is a* reducibility candidate *if*

- *if $\pi \in R$, then π is strongly terminating,*
- *if $\pi \in R$ and $\pi \longrightarrow \pi'$ then $\pi' \in R$,*
- *all variables belong to R,*
- *if π is an elimination and if for every π' such that $\pi \longrightarrow^1 \pi'$, $\pi' \in R$ then $\pi \in R$.*

Let $\mathcal{C}$ be the set of all reducibility candidates.

Assigning a reducibility candidate to each atomic proposition A, is equivalent to assign to each predicate symbol P of n arguments a function $\hat{P}$ that maps n-uples of terms to reducibility candidates. Then, we define the set $|P(t_1, ..., t_n)|$ as $\hat{P}(t_1, ..., t_n)$. Thus we want to prove that if we know how to assign such a function to each predicate symbol, in such a way that the sets of reducible proofs defined relatively to these functions are such that two equivalent propositions have the same set of reducible proofs, then cut elimination holds modulo this congruence.

This can be generalized: to have cut elimination it is sufficient to assign, to each predicate symbol P of n arguments, a function $\hat{P}$ that maps n-uples of elements of an arbitrary set M to reducibility candidates and to associate to each term t an element $|t|$ of M. Then we define $|P(t_1, ..., t_n)|$ as $\hat{P}(|t_1|, ..., |t_n|)$. If the sets of reducible proofs defined relatively to these functions are such that two equivalent propositions have the same set of reducible proofs, then cut elimination holds modulo this congruence.

There are many similarities between this definition and the definition of a model. In particular the fact that if $A \equiv B$ then $|A| = |B|$ can be read as the validity of the congruence in this structure. The only difference with the notion of model is that the functions $\hat{P}$ do not map n-uples of elements of M to truth values 0 or 1, but to reducibility candidates. Hence such structures are many-valued models where truth values are reducibility candidates. We shall call them *pre-models*. As we want to apply this result to many-sorted theories also, we directly give the definition for many-sorted predicate logic modulo.

5.5 Pre-model

Definition 5.5.1 (Pre-model) *Let $\mathcal{L}$ be a many sorted first-order language. A* pre-model *for $\mathcal{L}$ is given by:*

- *for every sort T, a set M_T,*
- *for every function symbol f of rank $\langle T_1, \ldots, T_n, U\rangle$, a function $\hat{f}$ from $M_{T_1} \times \ldots \times M_{T_n}$ to M_U,*
- *for every predicate symbol P of rank $\langle T_1, \ldots, T_n\rangle$, a function $\hat{P}$ from $M_{T_1} \times \ldots \times M_{T_n}$ to $\mathcal{C}$.*

Definition 5.5.2 *Let t be a term and ϕ an assignment mapping all the free variables of t of sort T to elements of M_T. We define the object $|t|_\phi$ by induction over the height of t.*

- $|x|_\phi = \phi(x)$,
- $|f(t_1, \ldots, t_n)|_\phi = \hat{f}(|t_1|_\phi, \ldots, |t_n|_\phi)$.

Definition 5.5.3 *Let A be a proposition and ϕ an assignment mapping all the free variables of A of sort T to elements of M_T. We define the set $|A|_\phi$ of proof-terms by induction over the height of A.*

- *A proof-term π is an element of $|P(t_1, \ldots, t_n)|_\phi$ if it is in $\hat{P}(|t_1|_\phi, \ldots, |t_n|_\phi)$.*
- *A proof-term π is an element of $|\top|_\phi$ if π is strongly terminating.*
- *A proof-term π is an element of $|\bot|_\phi$ if π is strongly terminating.*

- *A proof-term π is an element of $|A \wedge B|_\phi$ if π is strongly terminating and when π reduces to a proof-term of the form $\langle \pi_1, \pi_2 \rangle$ then π_1 and π_2 are elements of $|A|_\phi$ and $|B|_\phi$.*

- *A proof-term π is an element of $|A \vee B|_\phi$ if π is strongly terminating and when π reduces to a proof-term of the form $i(\pi_1)$ (resp. $j(\pi_2)$) then π_1 (resp. π_2) is an element of $|A|_\phi$ (resp. $|B|_\phi$).*

- *A proof-term π is element of $|A \Rightarrow B|_\phi$ if it is strongly terminating and when π reduces to a proof-term of the form $\xi \mapsto \pi_1$ then for every π' in $|A|_\phi$, $(\pi'/\xi)\pi_1$ is an element of $|B|_\phi$.*

- *A proof-term π is an element of $|\forall x\ A|_\phi$ if it is strongly terminating and when π reduces to a proof-term of the form $x \mapsto \pi_1$ then for every term t of sort T (where T is the sort of x) and every element E of M_T, $(t/x)\pi_1$ is an element of $|A|_{\phi+\langle x,E\rangle}$.*

- *A proof-term π is an element of $|\exists x\ A|_\phi$ if π is strongly terminating and whenever π reduces to a proof-term of the form $\langle t, \pi_1 \rangle$ there exists an element E of M_T (where T is the sort of x) such that π_1 is an element of $|A|_{\phi+\langle x,E\rangle}$.*

Definition 5.5.4 *A pre-model is a pre-model of a congruence $\equiv$ if, whenever $A \equiv B$, then for every assignment ϕ, $|A|_\phi = |B|_\phi$.*

Proposition 5.5.1 *For every proposition A and assignment ϕ, $|A|_\phi$ is a reducibility candidate*

Proof. By induction over the height of A.

If A is an atomic proposition, $|A|_\phi$ is a reducibility candidate by definition.

If A is a composed proposition, then, by definition, $|A|_\phi$ contains only terminating proof-terms. It is routine to prove closure by reduction. It is also routine to check that all variables are members of $|A|_\phi$.

Now, we assume that π is a an elimination and that for every π' such that $\pi \longrightarrow^1 \pi'$, $\pi' \in |A|_\phi$. We want to prove that π is in $|A|_\phi$. Following the definition of $|A|_\phi$, we first prove that π is strongly terminating and then that if it reduces to an introduction, the sub-proofs belong to the appropriate sets.

We first prove that π is strongly terminating. Let $\pi = \pi_1, \pi_2, \ldots$ be a reduction sequence issued from π. If this sequence is empty it is finite. Otherwise we have $\pi \longrightarrow^1 \pi_2$ and hence π_2 is an element of $|A|_\phi$ thus it is strongly terminating and the reduction sequence is finite.

Then we prove that if π reduces to an introduction then the sub-proofs belong to the appropriate sets. Let $\pi = \pi_1, \pi_2, \ldots \pi_n$ be a reduction sequence issued from π and such that π_n is an introduction. This sequence cannot be empty because π is an elimination and hence not an introduction. Thus $\pi \longrightarrow^1 \pi_2 \longrightarrow \pi_n$. We have $\pi_2 \in |A|_\phi$ and thus if π_n is an introduction the sub-proofs belong to the appropriate sets.

Proposition 5.5.2 (Substitution) *Given any proposition A, term t and variable x we have*

$$|(t/x)A|_\phi = |A|_{\phi+\langle x,|t|_\phi\rangle}$$

Proof. By induction on the height of A.

We can now prove the main theorem of this chapter: if a system has a pre-model then proof-terms modulo this system terminate.

Proposition 5.5.3 *Let $\equiv$ be a congruence and $\mathcal{M}$ be a pre-model of $\equiv$. If $\Gamma \vdash_\equiv \pi : A$ then the proof-term π is strongly terminating.*

Proof. As $|A|_\emptyset$ is a reducibility candidate, it is sufficient to prove that if $\Gamma \vdash \pi : A$ then the proof-term π is an element of $|A|_\emptyset$. More generally, we prove, by induction over the height of the proof-assignment tree, that if $\Gamma \vdash \pi : A$,

- θ is a substitution mapping term variables to terms,
- ϕ is an assignment mapping variables to elements of the model,
- σ is a substitution mapping some proof variables associated to proposition B in Γ to an element of $|B|_\phi$,

then $\sigma\theta\pi$ is an element of $|A|_\phi$.

- Axiom. If π is a variable ξ, we have $(\xi : B) \in \Gamma$ with $B \equiv A$. If ξ is in the domain of definition of σ, then $\sigma\theta\xi = \sigma\xi$ is an element of $|B|_\phi = |A|_\phi$, otherwise $\sigma\theta\xi = \sigma\xi = \xi$ is an element of $|A|_\phi$ because $|A|_\phi$ is a candidate.
- $\top$-intro. The proof-term π has the form I. We have $\sigma\theta\pi = I$. This proof-term is normal, hence it is strongly terminating. Hence, the proof-term $\sigma\theta I$ is in $|A|_\phi$.
- $\wedge$-intro. The proof-term π has the form $\langle\rho_1, \rho_2\rangle$ where ρ_1 is a proof of some proposition B and ρ_2 a proof of some proposition C. We have $\sigma\theta\pi = \langle\sigma\theta\rho_1, \sigma\theta\rho_2\rangle$. Consider a reduction sequence issued from this proof-term. This sequence can only reduce the proof-terms $\sigma\theta\rho_1$ and $\sigma\theta\rho_2$. By induction hypothesis these proof-terms are in $|B|_\phi$ and $|C|_\phi$. Thus the reduction sequence is finite.

  Furthermore, all reducts of $\sigma\theta\pi$ have the form $\langle\rho'_1, \rho'_2\rangle$ where ρ'_1 is a reduct of $\sigma\theta\rho_1$ and ρ'_2 one of $\sigma\theta\rho_2$. The proof-terms ρ'_1 and ρ'_2 are in $|B|_\phi$ and $|C|_\phi$ because these sets are candidates.

  Hence, the proof-term $\sigma\theta\langle\rho_1, \rho_2\rangle$ is in $|A|_\phi$.
- $\vee$-intro. The proof-term π has the form $i(\rho)$ (resp. $j(\rho)$) and ρ is a proof of some proposition B. We have $\sigma\theta\pi = i(\sigma\theta\rho)$ (resp. $j(\sigma\theta\rho)$). Consider a reduction sequence issued from this proof-term. This sequence can only reduce the proof-terms $\sigma\theta\rho$. By induction hypothesis this proof-term is an element of $|B|_\phi$. Thus the reduction sequence is finite.

Furthermore, all reducts of $\sigma\theta\pi$ have the form $i(\rho')$ (resp. $j(\rho')$) where ρ' is a reduct of $\sigma\theta\rho$. The proof-term ρ' is an element of $|B|_\phi$ because this set is a candidate.

Hence, the proof-term $\sigma\theta i(\rho)$ (respectively $\sigma\theta j(\rho)$) is an element of $|A|_\phi$.

- $\Rightarrow$-intro. The proof-term π has the form $\xi \mapsto \rho$ where ξ is a proof variable of some proposition B and ρ a proof of some proposition C. We have $\sigma\theta\pi = \xi \mapsto \sigma\theta\rho$, consider a reduction sequence issued from this proof-term. This sequence can only reduce the proof-term $\sigma\theta\rho$. By induction hypothesis, the proof-term $\sigma\theta\rho$ is an element of $|C|_\phi$, thus the reduction sequence is finite.

  Furthermore, all reducts of $\sigma\theta\pi$ have the form $\xi \mapsto \rho'$ where ρ' is a reduct of $\sigma\theta\rho$. Let τ be any proof of $|B|_\phi$, the proof-term $(\tau/\xi)\rho'$ can be obtained by reduction from $((\tau/\xi) \circ \sigma)\theta\rho$. By induction hypothesis, the proof-term $((\tau/\xi) \circ \sigma)\theta\rho$ is an element of $|C|_\phi$. The proof-term $(\tau/\xi)\rho'$ is an element of $|C|_\phi$ because this set is a candidate.

  Hence, the proof-term $\sigma\theta\xi \mapsto \rho$ is an element of $|A|_\phi$.

- $\forall$-intro. The proof-term π has the form $x \mapsto \rho$ where ρ is a proof of some proposition B. We have $\sigma\theta\pi = x \mapsto \sigma\theta\rho$.

  Consider a reduction sequence issued from the proof-term $\sigma\theta\pi = x \mapsto \sigma\theta\rho$. This sequence can only reduce the proof-term $\sigma\theta\rho$. Let E be an element of M_T (where T is the sort of x). By induction hypothesis, the proof-term $\sigma\theta\rho$ is an element of $|B|_{\phi+\langle x,E\rangle}$, thus the reduction sequence is finite.

  Furthermore, all reducts of $\sigma\theta\pi$ have the form $x \mapsto \rho'$ where ρ' is a reduct of $\sigma\theta\rho$. The proof-term $(t/x)\rho'$ is obtained by reducing the proof-term $((t/x)\sigma)((t/x) \circ \theta)\rho$. By induction hypothesis again, the proof-term $((t/x)\sigma)((t/x) \circ \theta)\rho$ is an element of $|B|_{\phi+\langle x,E\rangle}$. The proof-term $(t/x)\rho'$ is an element of $|B|_{\phi+\langle x,E\rangle}$, because this set is a candidate.

  Hence $\sigma\theta(x \mapsto \rho)$ is an element of $|A|_\phi$.

- $\exists$-intro. The proof-term π has the form $\langle t, \rho\rangle$, $A \equiv \exists x\ B$ and ρ is a proof of $(t/x)B$. We have $\sigma\theta\pi = \langle\theta t, \sigma\theta\rho\rangle$. Consider a reduction sequence issued from this proof-term. This sequence can only reduce the proof-term $\sigma\theta\rho$. By induction hypothesis this proof-term is in $|(t/x)B|_\phi$. Thus the reduction sequence is finite.

  Furthermore, let $E = |t|_\phi$. Any reduct of $\sigma\theta\pi$ has the form $\langle\theta t, \rho'\rangle$ where ρ' is a reduct of $\sigma\theta\rho$. The proof-term ρ' is an element of $|(t/x)B|_\phi$, *i.e.* of $|B|_{\phi+\langle x,E\rangle}$, because $|B|_{\phi+\langle x,E\rangle}$ is a candidate.

  Hence, the proof-term $\sigma\theta\langle t, \rho\rangle$ is an element of $|A|_\phi$.

- $\bot$-elim. The proof-term π has the form $\delta_\bot(\rho)$ where ρ is a proof of $\bot$. We have $\sigma\theta\pi = \delta_\bot(\sigma\theta\rho)$. By induction hypothesis, the proof-term $\sigma\theta\rho$ is an element of $|\bot|_\phi$. Hence, it is strongly terminating. Let n be the maximum length of reduction sequences issued from this proof-term. We

prove by induction on n that $\delta_\perp(\sigma\theta\rho)$ is in $|A|_\phi$. Since this proof-term is an elimination, we only need to prove that every of its one step reducts is in $|A|_\phi$. The reduction can only take place in $\sigma\theta\rho$ and we apply the induction hypothesis.

Hence, the proof-term $\sigma\theta\delta_\perp(\rho)$ is an element of $|A|_\phi$.

- $\wedge$-elim. We only detail the case of left elimination. The proof-term π has the form $fst(\rho)$ where ρ is a proof of some proposition $A \wedge B$. We have $\sigma\theta\pi = fst(\sigma\theta\rho)$. By induction hypothesis the proof-term $\sigma\theta\rho$ is in $|A \wedge B|_\phi$. Hence, it is strongly terminating. Let n be the maximum length of a reduction sequence issued from this proof-term. We prove by induction on n that $fst(\sigma\theta\rho)$ is in the set $|A|_\phi$. Since this proof-term is a elimination we only need to prove that every of its one step reducts is in $|B|_\phi$. If the reduction takes place in $\sigma\theta\rho$ then we apply the induction hypothesis. Otherwise $\sigma\theta\rho$ has the form $\langle\rho'_1, \rho'_2\rangle$ and the reduct is ρ'_1. By the definition of $|A \wedge B|_\phi$ this proof-term is in $|A|_\phi$.

  Hence, the proof-term $\sigma\theta fst(\rho)$ is an element of $|A|_\phi$.

- $\vee$-elim. The proof-term π has the form $\delta(\rho_1, \xi\rho_2\ \chi\rho_3)$ where ρ_1 is a proof of some proposition $B \vee C$ and ρ_2 and ρ_3 are proofs of A. We have $\sigma\theta\pi = \delta(\sigma\theta\rho_1, \xi\sigma\theta\rho_2, \chi\sigma\theta\rho_3)$. By induction hypothesis, the proof-term $\sigma\theta\rho_1$ is in the set $|B \vee C|_\phi$, and the proof-terms $\sigma\theta\rho_2$ and $\sigma\theta\rho_3$ are in the set $|A|_\phi$. Hence, these proof-terms are strongly terminating. Let n, n' and n'' be the maximum length of reduction sequences issued from these proof-terms. We prove by induction on $n + n' + n''$ that $\delta(\sigma\theta\rho_1, \sigma\theta\xi\rho_2, \chi\sigma\theta\rho_3)$ is in $|A|_\phi$. Since this proof-term is an elimination we only need to prove that every of its one step reducts is in $|A|_\phi$. If the reduction takes place in $\sigma\theta\rho_1$, $\sigma\theta\rho_2$ or $\sigma\theta\rho_3$ then we apply the induction hypothesis. Otherwise, if $\sigma\theta\rho_1$ has the form $i(\rho')$ (resp. $j(\rho')$) and the reduct is $(\rho'/\xi)\sigma\theta\rho_2$ (resp. $(\rho'/\chi)\sigma\theta\rho_3$). By the definition of $|B \vee C|_\phi$ the proof-term ρ' is in $|B|_\phi$ (resp. $|C|_\phi$). Hence by induction hypothesis $((\rho'/\xi)\circ\sigma)\theta\rho_2$ (resp. $((\rho'/\chi)\circ\sigma)\theta\rho_3$) is in $|A|_\phi$.

  Hence, the proof-term $\sigma\theta\delta(\rho_1, \xi\rho_2, \chi\rho_3)$ is an element of $|A|_\phi$.

- $\Rightarrow$-elim. The proof-term π has the form $(\rho_1\ \rho_2)$ and ρ_1 is a proof of some proposition $B \Rightarrow A$ and ρ_2 a proof of the proposition B. We have $\sigma\theta\pi = (\sigma\theta\rho_1\ \sigma\theta\rho_2)$. By induction hypothesis $\sigma\theta\rho_1$ and $\sigma\theta\rho_2$ are in the sets $|B \Rightarrow A|_\phi$ and $|B|_\phi$. Hence these proof-terms are strongly terminating. Let n be the maximum length of a reduction sequence issued from $\sigma\theta\rho_1$ and n' the maximum length of a reduction sequence issued from $\sigma\theta\rho_2$. We prove by induction on $n + n'$ that $(\sigma\theta\rho_1\ \sigma\theta\rho_2)$ is in the set $|A|_\phi$. Since this proof-term is an elimination we only need to prove that every of its one step reducts is in $|A|_\phi$. If the reduction takes place in $\sigma\theta\rho_1$ or in $\sigma\theta\rho_2$ then we apply the induction hypothesis. Otherwise $\sigma\theta\rho_1$ has the form $\xi \mapsto \rho'$ and the reduct is $(\sigma\theta\rho_2/\xi)\rho'$. By the definition of $|B \Rightarrow A|_\phi$ this proof-term is in $|A|_\phi$.

Hence, the proof-term $\sigma\theta(\rho_1\ \rho_2)$ is an element of $|A|_\phi$.

- $\forall$-elim. The proof-term π has the form $(\rho\ t)$ where ρ is a proof of some proposition $\forall x\ B$ and $A = (t/x)B$. We have $\sigma\theta\pi = (\sigma\theta\rho\ \theta t)$. By induction hypothesis, the proof-term $\sigma\theta\rho$ is in $|\forall x\ B|_\phi$. Hence, it is strongly terminating. Let n be the maximum length of a reduction sequence issued from this proof-term. We prove by induction on n that $(\sigma\theta\rho\ \theta t)$ is in the set $|A|_\phi$. As this proof-term is an elimination, we only need to prove that every of its one step reducts is in $|A|_\phi$. If the reduction takes place in $\sigma\theta\rho$ then we apply the induction hypothesis. Otherwise $\sigma\theta\rho$ has the form $x \mapsto \rho'$ and the reduct is $\mathrm{n}(\theta t/x)\rho'$. By the definition of $|\forall x\ B|_\phi$ this proof-term is in $|B|_{\phi+\langle x,E\rangle}$ for all E. Thus, it is in is in $|B|_{\phi+\langle x,|t|_\phi\rangle} = |(t/x)B|_\phi = |A|_\phi$.

  Hence, the proof-term $\sigma\theta(\rho\ t)$ is an element of $|A|_\phi$.

- $\exists$-elim. The proof-term π has the form $\delta_\exists(\rho_1, x\xi\rho_2)$ where ρ_1 is a proof of some proposition $\exists x\ B$ and ρ_2 is a proof of A. We have $\sigma\theta\pi = \delta_\exists(\sigma\theta\rho_1, x\xi\sigma\theta\rho_2)$. By induction hypothesis, the proof-term $\sigma\theta\rho_1$ is in the set $|\exists x\ B|_\phi$ and the proof-term $\sigma\theta\rho_2$ is in the set $|A|_\phi$. Hence, these proof-terms are strongly terminating. Let n and n' be the maximum length of reduction sequences issued from these proof-terms. We prove by induction on $n + n'$ that $\delta_\exists(\sigma\theta\rho_1, x\xi\sigma\theta\rho_2)$ is in $|A|_\phi$. As this proof-term is an elimination, we only need to prove that every of its one step reducts is in $|A|_\phi$. If the reduction takes place in $\sigma\theta\rho_1$ or $\sigma\theta\rho_2$ then we apply the induction hypothesis. Otherwise, $\sigma\theta\rho_1$ has the form $\langle t, \rho'\rangle$ and the reduct is $(\rho'/\xi)(t/x)\sigma\theta\rho_2 = ((\rho'/\xi) \circ (t/x)\sigma)((t/x) \circ \theta)\rho_2$. By the definition of $|\exists x\ B|_\phi$, there exists an element E of such that the proof-term ρ' is in $|B|_{\phi+\langle x,E\rangle}$. Thus, by induction hypothesis, the proof-term $((\rho'/\xi) \circ (t/x)\sigma)((t/x) \circ \theta)\rho_2$ is in $|A|_{\phi+\langle x,E\rangle}$, *i.e.* in $|A|_\phi$.

  Hence, the proof-term $\sigma\theta\delta_\exists(\rho_1, \xi x\rho_2)$ is an element of $|A|_\phi$.

5.6 Pre-model construction

5.6.1 The term case

Proposition 5.6.1 *If a congruence is defined by a rewrite system or a set of equalities on terms, but not on propositions, then it has a pre-model and hence proof reduction terminates modulo this congruence.*

Proof. We associate the set of strongly terminating proofs for all atomic propositions.

Corollary 5.6.2 *All equational theories are consistent, have the disjunction property and the witness property.*

5.6.2 Quantifier free rewrite systems

Definition 5.6.1 (Quantifier free) *A rewrite system is* quantifier free *if no quantifier appears on the right hand side of any of its rules.*

Proposition 5.6.3 *A quantifier free, confluent, and terminating rewrite systems has a pre-model, hence proof reduction terminates modulo such a rewrite system.*

Proof. By induction over proposition height, we associate a set of proof-terms to each each normal closed quantifier free proposition.

$$\begin{array}{rcl}
\Psi(A) & = & \{\pi \mid \pi \text{ st. ter.}\} \quad \text{if } A \text{ is atomic} \\
\Psi(\top) & = & \{\pi \mid \pi \text{ st. ter.}\} \\
\Psi(\bot) & = & \{\pi \mid \pi \text{ st. ter.}\} \\
\Psi(A \wedge B) & = & \{\pi \mid \pi \text{ st. ter.} \wedge \pi \longrightarrow \langle \pi_1, \pi_2 \rangle \Rightarrow \pi_1 \in \Psi(A) \wedge \pi_2 \in \Psi(B)\} \\
\Psi(A \vee B) & = & \{\pi \mid \pi \text{ st. ter.} \wedge \pi \longrightarrow i(\pi_1) \Rightarrow \pi_1 \in \Psi(A) \wedge \pi \longrightarrow i(\pi_2) \Rightarrow \pi_2 \in \Psi(B)\} \\
\Psi(A \Rightarrow B) & = & \{\pi \mid \pi \text{ st. ter.} \wedge \pi \longrightarrow \xi \mapsto \pi_1 \Rightarrow \forall \pi' \in \Psi(A)\ (\pi'/\xi)\pi_1 \in \Psi(B)\}
\end{array}$$

We define a pre-model as follows. Let M_T be the set of normal closed terms of sort T.

$$\begin{array}{rcl}
\hat{f}(t_1, \ldots, t_n) & = & f(t_1, \ldots, t_n)\downarrow \\
\hat{P}(t_1, \ldots, t_n) & = & \Psi((P(t_1, \ldots, t_n))\downarrow).
\end{array}$$

where $A\downarrow$ (resp. $t\downarrow$) is the normal form of the proposition A (resp. term t).

We prove, by an easy induction, that $|A|_\phi = |B|_\phi$ when $A \equiv B$.

5.6.3 Positive rewrite systems

For some rewrite systems, pre-models can be built by a fixed point construction.

Definition 5.6.2 *A rewrite system is* positive *if it rewrites atomic propositions to propositions containing only positive occurrences of atomic propositions.*

Definition 5.6.3 *A pre-model is* syntactical *if*

- *$M_T = \mathcal{T}_T/\equiv$ where $\mathcal{T}_T$ is the set of closed terms of sort T,*
- *if f is a function symbol, $\hat{f}$ is the function that maps the classes $e_1, ..., e_n$ to the class of the term $f(t_1, \ldots, t_n)$ where $t_1, ..., t_n$ are elements of $e_1, ..., e_n$ (since the relation $\equiv$ is a congruence, this does not depend of the choice of representatives).*

A syntactical pre-model is defined solely by the interpretation of predicate variables.

Definition 5.6.4 *Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two syntactical pre-models. We write $\hat{P}_1$ for the denotation of P in $\mathcal{M}_1$ and $\hat{P}_2$ for the denotation of P in $\mathcal{M}_2$*

We say that $\mathcal{M}_1 \leq \mathcal{M}_2$ if and only if for any predicate symbol P and closed terms $t_1, \ldots, t_n$ we have

$$\hat{P}_1(t_1, \ldots, t_n) \subseteq \hat{P}_2(t_1, \ldots, t_n)$$

The set of syntactical pre-models is a complete lattice for the order $\leq$.

Proposition 5.6.4 *Let $\mathcal{R}$ be a confluent and terminating rewrite system. If the system $\mathcal{R}$ is positive then it has a pre-model, hence proof reduction terminates modulo $\mathcal{R}$.*

Proof. Let $\mathcal{F}$ be the function mapping syntactical pre-models to syntactical pre-models defined by

$$\mathcal{F}(\mathcal{M})(P)(t_1, \ldots, t_n) = |P(t_1, \ldots, t_n)\downarrow|_{\mathcal{M},\emptyset}.$$

As the system $\mathcal{R}$ is positive the function $\mathcal{F}$ is monotone. Hence, as the set of syntactical pre-models is a complete lattice, it has a fixed point. This fixed point is a pre-model of the rewrite system.

Proposition 5.6.5 *Let $\mathcal{R}$ be a rewrite system such that any atomic proposition has at most one one-step reduct. If the system $\mathcal{R}$ is positive then it has a pre-model, hence proof reduction terminates modulo $\mathcal{R}$.*

Proof. Let $\mathcal{F}$ be the function mapping syntactical pre-models to syntactical pre-models defined by

$$\mathcal{F}(\mathcal{M})(P)(t_1, \ldots, t_n) = |P(t_1, \ldots, t_n)+|_{\mathcal{M},\emptyset}$$

where $A+$ is the unique one-step reduct of A if it exists and A otherwise. Again, since the system $\mathcal{R}$ is positive the function $\mathcal{F}$ is monotone and again, since the set of syntactical pre-models is a complete lattice, it has a fixed point. This fixed point is a pre-model of the rewrite system.

5.6.4 Type theory and type theory with infinity

Proposition 5.6.6 (Girard's theorem) *Simple type theory has a pre-model, hence proof reduction terminate in simple type theory.*

Proof. We construct a pre-model as follows. The essential point is that we anticipate the fact that objects of sort o actually represent propositions, by interpreting them as reducibility candidates.

$$\begin{aligned} M_\iota &= \{0\} \\ M_o &= \mathcal{C} \\ M_{T\to U} &= M_U^{M_T} \end{aligned}$$

$$
\begin{array}{rcl}
\hat{S}_{T,U,V} & = & a \mapsto (b \mapsto (c \mapsto a(c)(b(c)))) \\
\hat{K}_{T,U} & = & a \mapsto (b \mapsto a) \\
\hat{\alpha}(a,b) & = & a(b) \\
\hat{\varepsilon}(a) & = & a
\end{array}
$$

$$
\begin{array}{rcl}
\hat{\dot{\top}} & = & \{\pi \mid \pi \text{ st. ter.}\} \\
\hat{\dot{\bot}} & = & \{\pi \mid \pi \text{ st. ter.}\} \\
\hat{\dot{\wedge}}(a,b) & = & \{\pi \mid \pi \text{ st. ter.} \wedge\ \pi \longrightarrow \langle \pi_1, \pi_2 \rangle \Rightarrow \pi_1 \in a \wedge \pi_2 \in b\} \\
\hat{\dot{\vee}}(a,b) & = & \{\pi \mid \pi \text{ st. ter.} \wedge\ (\pi \longrightarrow i(\pi_1) \Rightarrow \pi_1 \in a) \wedge (\pi \longrightarrow i(\pi_2) \Rightarrow \pi_2 \in b)\} \\
\hat{\dot{\Rightarrow}}(a,b) & = & \{\pi \mid \pi \text{ st. ter.} \wedge\ \pi \longrightarrow \xi \mapsto \pi_1 \Rightarrow \forall \pi' \in a\ (\pi'/\xi)\pi_1 \in b\} \\
\hat{\dot{\forall}}_T(a) & = & \{\pi \mid \pi \text{ st. ter.} \wedge\ \pi \longrightarrow x \mapsto \pi_1 \Rightarrow \forall t \text{ of type } T\ \forall E \in M_T\ (t/x)\pi_1 \in a(E)\} \\
\hat{\dot{\exists}}_T(a) & = & \{\pi \mid \pi \text{ st. ter.} \wedge\ \pi \longrightarrow \langle t, \pi_2 \rangle \Rightarrow \exists E \in M_T\ \pi_2 \in a(E)\}
\end{array}
$$

It is easy to check that $|A|_\phi = |B|_\phi$ when $A \equiv B$.

Proposition 5.6.7 *Simple type theory with infinity has a pre-model, hence proof reduction terminates in simple type theory with infinity.*

Proof.

$$
\begin{array}{rcl}
M_\iota & = & \mathbb{N} \\
M_o & = & \mathcal{C} \\
M_{T \to U} & = & M_U^{M_T}
\end{array}
$$

$$
\begin{array}{rcl}
\hat{0} & = & 0, \\
\hat{Su} & = & n \mapsto n+1, \\
\hat{Pred} & = & n \mapsto \text{if } n = 0 \text{ then } 0 \text{ else } n-1, \\
\hat{Null} & = & n \mapsto \{\pi \mid \pi \text{ st. ter.}\}, \\
\hat{S}_{T,U,V} & = & a \mapsto (b \mapsto (c \mapsto a(c)(b(c)))) \\
\hat{K}_{T,U} & = & a \mapsto (b \mapsto a) \\
\hat{\alpha}(a,b) & = & a(b) \\
\hat{\varepsilon}(a) & = & a
\end{array}
$$

$$
\begin{array}{rcl}
\hat{\dot{\top}} & = & \{\pi \mid \pi \text{ st. ter.}\} \\
\hat{\dot{\bot}} & = & \{\pi \mid \pi \text{ st. ter.}\} \\
\hat{\dot{\wedge}}(a,b) & = & \{\pi \mid \pi \text{ st. ter.} \wedge\ \pi \longrightarrow \langle \pi_1, \pi_2 \rangle \Rightarrow \pi_1 \in a \wedge \pi_2 \in b\} \\
\hat{\dot{\vee}}(a,b) & = & \{\pi \mid \pi \text{ st. ter.} \wedge\ (\pi \longrightarrow i(\pi_1) \Rightarrow \pi_1 \in a) \wedge (\pi \longrightarrow i(\pi_2) \Rightarrow \pi_2 \in b)\} \\
\hat{\dot{\Rightarrow}}(a,b) & = & \{\pi \mid \pi \text{ st. ter.} \wedge\ \pi \longrightarrow \xi \mapsto \pi_1 \Rightarrow \forall \pi' \in a\ (\pi'/\xi)\pi_1 \in b\} \\
\hat{\dot{\forall}}_T(a) & = & \{\pi \mid \pi \text{ st. ter.} \wedge\ \pi \longrightarrow x \mapsto \pi_1 \Rightarrow \forall t \text{ of type } T\ \forall E \in M_T\ (t/x)\pi_1 \in a(E)\} \\
\hat{\dot{\exists}}_T(a) & = & \{\pi \mid \pi \text{ st. ter.} \wedge\ \pi \longrightarrow \langle t, \pi_2 \rangle \Rightarrow \exists E \in M_T\ \pi_2 \in a(E)\}
\end{array}
$$

It is easy to check that $|A|_\phi = |B|_\phi$ when $A \equiv B$.

Remark. In the pre-model above $\dot{\top}$ and $\dot{\bot}$ are interpreted by the same reducibility candidate (while in a model they are interpreted by a different truth value) hence the interpretation of *Null* is simply the constant function equal to this

candidate. Thus it is not necessary to interpret the type ι as $\mathbb{N}$ and we could also take $M_\iota = \{0\}$.