University of Warsaw
Faculty of Mathematics, Informatics and Mechanics

Agnieszka Kozubek

A Formalization of the Naive Type Theory

PhD dissertation

Supervisor
prof. dr hab. Paweł Urzyczyn
Institute of Informatics
University of Warsaw

December 2011

Author's declaration:

Aware of legal responsibility I hereby declare that I have written this dissertation myself and all the contents of the dissertation have been obtained by legal means.

Date Author's signature

Supervisor's declaration:

the dissertation is ready to be reviewed

Date Supervisor's signature

Abstract

The language of contemporary mathematics is set theory. However, the axiomatic set theory was built to establish the consistency of mathematics, not to be the language of it. Set theory identifies two very basic ideas into one notion of a "set": set as a universe and set as a predicate. Type theory is a language which do not exhibit many of the drawbacks of set theory. In this thesis we build a type theory in which there is a clear distinction between universes and predicates. Our theory is a certain Pure Type System extended with inductive types. We give the definition of the system and prove its basic properties. We establish consistency of the system by giving the strong normalization proof.

Streszczenie

Językiem współczesnej matematyki jest teoria mnogości. Jednak aksjomatyczna teoria mnogości powstała w zupełnie innym celu — aby uzasadnić niesprzeczność matematyki. Teoria mnogości utożsamia dwa podstawowe pojęcia w jedno pojęcie zbioru: zbiór jako uniwersum i zbiór jako materializacja predykatu. Teoria typów jest językiem, który jest pozbawiony wielu wad teorii mnogości. W tej pracy proponujemy teorię typów, w której jest wyraźne rozróżnienie między uniwersum i predykatem. Nasza teoria to pewien Pure Type System rozszerzony o typy indukcyjne. Podajemy definicję systemu i pokazujemy jego podstawowe własności. Dowodzimy własność silnej normalizacji, z której wynika niesprzeczność systemu.

Słowa kluczowe

rachunek lambda, systemy typów, pure type systems, normalizacja

Keywords

lambda calculus, type systems, pure type systems, normalization

ACM Computing Classification

F. Theory of Computation
F.4 Mathematical Logic and Formal Languages
F.4.1 Mathematical Logic
Lambda calculus and related systems

Acknowledgements

Most of all, I would like to thank my supervisor Professor Paweł Urzyczyn. He first taught me about foundations of mathematics, logic and typed lambda calculus. He encouraged me to work in this area and provided many valuable ideas. I am very grateful for his concern and support, I could always count on his help and advice.

I would like to thank Daria Walukiewicz-Chrząszcz for her valuable comments on early versions of this thesis. Her suggestions to use typed candidates and an alternative interpretation for inductive types helped me to complete the strong normalization proof.

I thank Zhaohui Luo, Alexandre Miquel and Christine Paulin-Mohring for their insightful remarks and helpful suggestions on how to improve this work.

I would like to thank my family for their love and support during all these years it took me to complete this thesis. I know they never understood why I started this work but they supported me nonetheless. I thank my friend Andrzej for putting up with my moods, especially on days when I had doubts about my work.

I gratefully acknowledge the support of the Polish Government Grant N N206 358536.

Contents

Chapter 1

Introduction

1.1. The problem

The language of contemporary mathematics is set theory. Virtually all maths is developed within the framework of set theory, and all books and papers are written with the silent assumption of ZF or ZFC axioms behind the back. We even use this framework for teaching mathematics at all levels, from university to the kindergarten level.

However, axiomatic set theory was introduced for a different purpose. It was built to establish the consistency of mathematics, not to be the language of it. All complex constructions are built from elementary notions like "set" and "being an element" of a set. We are so used to this idea that we do not see its drawbacks. Yet the drawbacks are easily visible when we try to teach set theory to students. Instead of explaining properties of mathematical objects like ordered pairs, set unions or natural numbers we end up explaining details of their encodings.

Moreover, in set theory two very basic ideas are glued into one notion of a "set":

- Set as a domain or universe;
- Set as a predicate.

We are used to treat this identification as natural and obvious. But perhaps only because we were taught to do so. These two ideas are in fact different and this confusion is responsible for Russell's paradox.

A language which do not exhibit many of the drawbacks of set theory is type theory. Mathematicians have been classifying objects according to their domain, kind, sort or *type* since the antiquity [4, 32, 33]. An empty set of numbers and an empty set of apples are intuitively *not* the same. In everyday mathematical practice we very often informally use the concept of a type. Think of a function mapping elements of set A to elements of set B. Such a function can only be applied to elements of set ("type") A. Similarly, a union $\bigcup A$ of a family A of sets is typically of the same "type" as members of A rather than as A itself.

We want to build a type theory that would capture the naive understanding of a type. We believe that such theory would give a chance to build a framework for "naive" mathematics that would not exhibit many of the drawbacks mentioned above. In particular, it could be free from artificial formalizations and encodings and as such it would be more suitable as a framework for teaching mathematics to students.

The basic idea is to separate the two roles played by sets, to put apart domains (types) and predicates (selection criteria for objects of a given type). We want to bring back the idea which dates back to Cantor and his naive set theory: we want to identify predicates and subsets. We abandoned this approach in axiomatic set theory with the discovery of Russell's paradox. In type theory this identification is possible. For any type A we have a powerset type $P(A)$, identified with the function space $A \to *$, where $*$ is the sort of propositions.

1.2. Related systems

Simple type theory: In Church's simple type theory [9, 32] there are two base types: the type **i** of individuals and the type **b** of truth values. Expressions have types and formulas are simply expressions of type **b**. There is no built-in notion of a proof and formulas are not types. In addition to lambda-abstraction, there is another binding operator that can be used to build expressions, namely the *definite description* $\iota x.\, \varphi(x)$, meaning "the only object x that satisfies $\varphi(x)$". While various forms of definite description are often used in the informal language of mathematics, the construct does not occur in most contemporary logical systems. As argued by William Farmer in a series of papers [19, 20, 21, 22], simple type theory could be efficiently used in mathematical practice and teaching. Also the textbook [4] by P.B. Andrews develops a version of simple type theory as a basis for everyday mathematics. This is very much in line with our way of thinking. We choose a slightly different approach, mostly to avoid the inherently two-valued Boolean logic built in Church's type theory.

Quine's New Foundations: Quine's type theory [31, 48] is based on an implicit *linear* hierarchy of universes. Full comprehension is possible at each level, but a set always lives at a higher floor than its elements. The idea of a linear hierarchy is of course convenient from a foundational point of view, but is not very intuitive. Also implementing "ordinary" mathematics requires a similar effort as in the usual set theory. The restriction to stratified constructs does not help either: one encounters difficulties when trying to define functions between objects belonging to different levels of the hierarchy.

Constable's computational naive type theory: We have to admit that the title of Halmos' book has already been rephrased by R. Constable [10]. But Constable's idea of a "naive type theory" is quite different than ours. It is inspired by Martin-Löf's theory and based on the idea of a *setoid* type, determined by a domain of objects plus an appropriate notion of equality. (In other words, quotient becomes a basic notion.) For instance, the field $\mathbb{Z}_3$ has the same domain as the set of integers $\mathbb{Z}$, but a different equality. And $\mathbb{Z}_6$ is defined by taking an "intersection" of $\mathbb{Z}_2$ and $\mathbb{Z}_3$. This is very convenient and natural way of dealing with quotient constructions. However (even putting aside the little counterintuitivity of the "contravariant" intersection) we still believe that a "naive" notion of equality should be more strict: two objects should not be considered the same in one context but different in another.

Weyl's predicative mathematics and Luo's logic-enriched theories: Zhaohui Luo in [37] considers „logic-enriched type theories" where the logical aspect is separated by design from the data-type aspect (in particular a separate kind *Prf*(P) is used for proofs of any proposition P). Within that framework one can introduce both predicative and impredicative notion of a set, so that the kind *Type* is closed under the powerset construction. This approach is used by Adams and Luo [3] to formalize the predicative mathematics of Weyl [56], who long ago made an explicit distinction between "categories" and sets, understood respectively as universes and predicates. Weyl's theory is strictly predicative, and this certainly departs from our "naive" understanding of sets, but the impredicative version mentioned in [37] is very much consistent with it.

Maietti and Sambin's Minimalist Foundation: Maietti and Sambin in [40] propose to build a foundation for constructive mathematics which could be a common core among relevant existing foundations in axiomatic set theory, such as Aczel-Myhill's CZF theory [2], or in type theory, such as Martin-Löf type theory [44] and Coquand's Calculus of Inductive Constructions [12]. First steps toward implementation of the theory were shown in [38]. In [39] they present a two-level theory to formalize constructive mathematics. One level is an intentional type theory, called Minimal type theory. The other level is an extensional set theory that is interpreted in the first one by means of a quotient model. This two-level theory has two main features: it is minimal among the most relevant foundations for constructive mathematics; it is constructive thanks to the way the extensional level is linked to the intentional one which fulfills the "proofs-as-programs" paradigm and acts as a programming language. However, their theory is predicative and we believe that "naive" type theory has to be impredicative.

Luo and Goguen's UTT: A system similar to ours was proposed by Luo [35, 36] and Goguen [29]. The system UTT is the Calculus of Constructions together with Martin-Löf predicative type theory. The logic of the system is higher order logic. In the system there is an impredicative universe of propositions. At the datatype level, there are predicative universes and inductive types. There are no inductive predicates. The inductive types in UTT are syntactically very similar to inductive types in our system and inductive types in Werner's Calculus of Inductive Construction [55]. As a system formulated in Martin-Löf type theory, UTT is a system with judgemental equality. There is no conversion rule and formally in the system there is no reduction. The equality in UTT is beta-, eta- and iota-equality, thus the equality is different from our notion of conversion.

1.3. Pure Type Systems

Pure Type Systems (PTS) is a framework for defining type systems. It was introduced independently by Berardi [8] and Terlouw [52]. The framework is a generalization of the well-known Barendregt cube [5]. The main properties of Pure Type Systems are discussed in Barendregt [5]. We chose a certain Pure Type System (PTS) as a basis for our theory.

A Pure Type System is specified by three sets $(\mathcal{S}, \mathcal{A}, \mathcal{R})$ where

- $\mathcal{S}$ is the set of *s*orts;
- $\mathcal{A} \subseteq \mathcal{S} \times \mathcal{S}$ the set of *axioms*;
- $\mathcal{R} \subseteq \mathcal{S} \times \mathcal{S} \times \mathcal{S}$ is the set of *rules.*

The terms of the system are defined by the following grammar

$$T := s \mid x \mid (\lambda x : T.T) \mid (TT) \mid (\Pi x : T.T)$$

where x is a variable and s is a sort. As a convention we omit the outermost parentheses. Application associates to the left: $((PQ)R)$ is abbreviated PQR; abstraction and product associate to the right: $(\lambda x : T_1(\lambda x : T_2.P))$ is abbreviated $\lambda x : T_1 \lambda x : T_2.P$. If x does not occur in B then $\Pi x : A.B$ is sometimes denoted by $A \to B$.

For PTSs we assume the usual β-reduction. The relation $\to_\beta$ is described by the rule

$$(\lambda x : T_1.A)B \to_\beta A[x := B]$$

and the usual compatibility rules: whenever $A \to_\beta A'$ then

$$\begin{array}{cc} AB \to_\beta A'B, & BA \to_\beta BA, \\ \lambda x : A.B \to_\beta \lambda x : A'.B, & \lambda x : B.A \to_\beta \lambda x : B.A', \\ \Pi x : A.B \to_\beta \Pi x : A'.B, & \Pi x : B.A \to_\beta \Pi x : B.A'. \end{array}$$

We write $A =_\beta B$ if there exists a sequence $A = A_0, A_1, \ldots, A_n = B$ such that for every $i = 0, \ldots, n-1$ we have $A_i \to_\beta A_{i+1}$ or $A_{i+1} \to_\beta A_i$. A term M is *normalizing* if and only if there is a reduction sequence from M ending in a normal form N. A term M is *strongly normalizing* if all reduction sequences beginning in M are finite.

A *context* is a finite (possibly empty) list of variable declarations $x_1 : A_1, \ldots, x_n : A_n$. We use Γ, Δ, Σ as meta-variables for contexts. We call $\{x_1, \ldots, x_n\}$ the *domain* of the context $\Gamma = (x_1 : A_1, \ldots, x_n : A_n)$ and we denote it by $dom(\Gamma)$.

A Pure Type System derives judgements (often called *sequents*) of the form $\Gamma \vdash A : B$. An assertion $\Gamma \vdash A : B$ states that A has type B in context Γ. The typing rules of the PTS specified by the triple $(\mathcal{S}, \mathcal{A}, \mathcal{R})$ are as follows:

$$\text{(Ax)} \quad \vdash s_1 : s_2 \qquad s_1 : s_2 \in \mathcal{A}$$

$$\text{(Var)}\ \frac{\Gamma \vdash A : s}{\Gamma, x : A \vdash x : A} \qquad x \notin dom(\Gamma)$$

$$\text{(Weak)}\ \frac{\Gamma \vdash A : B \qquad \Gamma \vdash C : s}{\Gamma, x : C \vdash A : B} \qquad x \notin dom(\Gamma)$$

$$\text{(App)}\ \frac{\Gamma \vdash M : (\Pi x : A.B) \qquad \Gamma \vdash N : A}{\Gamma \vdash MN : B[x := N]}$$

$$\text{(Abs)}\ \frac{\Gamma, x : A \vdash M : B \qquad \Gamma \vdash (\Pi x : A.B) : s}{\Gamma \vdash (\lambda x : A.M) : (\Pi x : A.B)}$$

$$\text{(Prod)}\ \frac{\Gamma \vdash A : s_1 \qquad \Gamma, x : A \vdash B : s_2}{\Gamma \vdash (\Pi x : A.B) : s_3} \qquad (s_1, s_2, s_3) \in \mathcal{R}$$

$$\text{(Conv)}\ \frac{\Gamma \vdash A : B \qquad \Gamma \vdash B' : s \qquad B =_\beta B'}{\Gamma \vdash A : B'}$$

A type system is *weakly normalizing* if and only if every term typable in the system is normalizing. A type system is *strongly normalizing* if and only if every term typable in the system is strongly normalizing.

A Pure Type System is *logical* [14] if and only if it is functional (see [26]) and contains two distinguished sorts $*$ and $\Box$ such that $* : \Box$ is an axiom, $(*, *, *)$ is a rule and there are no sorts of type $*$. A logical Pure Type System is *inconsistent* if there exists a proof of T in the context $T : *$. A type system is *consistent* if it is not inconsistent.

It is known that a strongly normalizing logical Pure Type System is consistent. There is no beta normal term of the type x, where x is a variable, in the context $x : *$. Thus a strongly normalizing system is consistent — if there was a term of type x then there would also be a normal one.

1.4. Calculus of Constructions and inductive types

The Calculus of Constructions (CC) was introduced by Coquand and Huet in [16]. They showed the consistency of the system by proving the strong normalization property. The Calculus of Constructions is the richest system in the Barendregt cube [5]. The system was also the basis for the preliminary version [15] of the Coq proof assistant [1]. The Calculus of Constructions is a Pure Type System with the following specification:

$$\begin{aligned}
\mathcal{S} &= \{*, \square\} \\
\mathcal{A} &= \{* : \square\} \\
\mathcal{R} &= \{(*, *, *), (*, \square, \square), (\square, *, *), (\square, \square, \square)\}
\end{aligned}$$

Even though the syntax of Pure Type System is homogeneous, terms typable in a context Γ may be divided into categories. We have *kinds*, that is, terms typable with $\square$; *types*, that is, terms typable with $*$; *objects*, that is, terms typable with a certain type t and *type constructors*, that is, terms typable with a certain kind t.

The four product rules correspond to various products.

- The rule $(*, *, *)$ introduces the usual function space, i.e. that terms may depend on terms.
- The rule $(\square, *, *)$ introduces type polymorphism, which allows to create terms depending on types.
- The rule $(*, \square, \square)$ represents dependent types (types that depend on terms).
- The rule $(\square, \square, \square)$ expresses that types may depend on types.

The Calculus of Constructions is a powerful system. It is possible to define natural numbers, lists, booleans and other inductive types by using the so-called impredicative encoding. There is a systematic procedure that, given a set of typed constructors for some inductive type produces a CC term representing this inductive type [47]. However, this coding has some important drawbacks, for example induction principles are not provable.

This is why the Calculus of Constructions was extended with inductive types. The system was introduced by Coquand and Paulin-Mohring in [17], followed by [46]. The strong normalization proof was done by Werner in [55].

We extend the syntax of the system with the following constructions:

$$\mathrm{Ind}(X : T)\{\vec{T}\} \mid \mathrm{Constr}(n, T) \mid \mathrm{Elim}(T, T, \vec{T}, T)\{\vec{T}\}$$

where n is a natural number. We will explain the meaning of those terms by using an example of natural numbers. The inductive definition of the type of natural numbers becomes

$$Nat = \mathrm{Ind}(X : *)\{X \mid X \to X\}$$

meaning that Nat is an inductive type with two constructors of type Nat and $Nat \to Nat$. The types of constructors are subject to the strict positivity condition: for every constructor of an inductive type I its recursive arguments must be of the form $\Pi\vec{x} : \vec{T}.I$ where I does not appear in $\vec{T}$. Without the condition the system is inconsistent and the details of the proof can be found in Chapter 2 section 2.3.

A term Constr(n, I) represents the n-th constructor of an inductive type I. The term Constr$(0, Nat)$ represents the natural number 0 and the term Constr$(1, Nat)$ represents successor function. Thus

- Constr$(1, Nat)$Constr$(0, Nat)$ is the natural number 1,
- Constr$(1, Nat)$(Constr$(1, Nat)$Constr$(0, Nat)$) is the natural number 2.

The terms corresponding to the elimination schemes are more complicated. We have different variants of elimination schemes: non-dependent, dependent, weak, strong. The simplest elimination scheme is non-dependent weak elimination. Suppose in a fixed context Γ we have the following typing judgements

$$P : *, \quad f_0 : P, \quad f_1 : Nat \to P \to P, \quad m : Nat.$$

Then the typing rules state that

$$\mathrm{Elim}(Nat, P, \epsilon, m)\{f_0 \mid f_1\} : P.$$

If we abstract over P, f_0, f_1 and m then for natural numbers we get the elimination term of the form

$$NatElim_{nodep} : \Pi P : *(P \to (Nat \to P \to P) \to Nat \to P).$$

The elimination term is representing the recursor on natural numbers.

The dependent elimination allows to create objects of type depending on an eliminated term. Suppose in a fixed context Γ we have the following typing judgements

$$P : Nat \to *, \quad f_0 : P0, \quad f_1 : (\Pi k : Nat(Pk \to P(Sk))), \quad m : Nat.$$

Then the typing rules state that

$$\mathrm{Elim}(Nat, P, \epsilon, m)\{f_0 \mid f_1\} : (Pm).$$

If we abstract over P, f_0, f_1 and m then for natural numbers we get the elimination term

$$NatElim_{dep} : \Pi P : Nat \to *(P0 \to (\Pi k : Nat(Pk \to P(Sk))) \to \Pi n : Nat.Pn).$$

The elimination term represents the induction scheme on natural numbers.

We can also have an elimination scheme like

$$NatElim_{Tnodep} : \Pi P : \Box^t(P \to (Nat \to P \to P) \to Nat \to P)$$

representing the possibility to create types. This variant of elimination scheme is called *strong elimination*. Strictly speaking the type above is not a valid type in the Calculus of Constructions but it shows well the principle of the elimination scheme. It is only allowed for the so called *small inductive types*, i.e. types that do not take types as arguments. Otherwise it leads to an inconsistent type system. The paradox in the system with strong elimination over large inductive types is studied in Chapter 2, section 2.4.

The reduction rules for natural numbers are

$$\begin{aligned} \mathrm{Elim}(Nat, P, \epsilon, 0)\{f_0 \mid f_1\} &\to_\iota f_0\ 0 \\ \mathrm{Elim}(Nat, P, \epsilon, Sn)\{f_0 \mid f_1\} &\to_\iota f_1\ n\ \mathrm{Elim}(Nat, P, \epsilon, n)\{f_0 \mid f_1\}. \end{aligned}$$

They are called ι-reduction and become part of the conversion rule of the system.

1.5. Less Naive Type Theory

The first attempt to formalize the naive type theory was the system called Naive Type Theory (NTT). It is a PTS with the following specification:

$$\mathcal{S} = \{*, \Box\}$$
$$\mathcal{A} = \{* : \Box\}$$
$$\mathcal{R} = \{(*, *, *), (*, \Box, *), (*, \Box, \Box)\}$$

As in the Calculus of Constructions the rule $(*, *, *)$ introduces the usual function space and the rule $(*, \Box, \Box)$ introduces dependent types. The rule $(*, \Box, *)$ expresses the slogans "subsets are objects": if τ is a type then the powerset $\tau \to *$ is also a type. However, the system turned out to be inconsistent. See Chapter 2 section 2.2 for a detailed description of the paradox.

Less Naive Type Theory (LNTT) is a refined version of the previous system. We split every sort in NTT into a t-version and a p-version. Thus in LNTT we have four sorts: $*^t$, $*^p$, $\Box^t$, $\Box^p$. The t-sorts correspond to object (datatype) part of the system and the p-sorts correspond to the logical part of the system. This is similar to the sorts Set and $Prop$ in the Coq proof assistant [1]. The full specification of the system is as follows:

$$\mathcal{S} = \{*^t, *^p, \Box^t, \Box^p\}$$
$$\mathcal{A} = \{*^t : \Box^t, *^p : \Box\}p$$
$$\mathcal{R} = \{(*^t, *^t, *^t), (*^p, *^p, *^p), (*^t, *^p, *^p), (*^t, \Box^p, *^t), (*^t, \Box^t, \Box^t), (\Box^p, *^p, *^p)\}.$$

The rules are now more fine-grained taking into account the distinction between the two parts of the system.

- The rule $(*^t, *^t, *^t)$ introduces the usual function space.
- The rule $(*^p, *^p, *^p)$ introduces implication (i.e. logical function space).
- The rule $(*^t, *^p, *^p)$ expresses universal quantification.
- The rule $(*^t, \Box^t, \Box^t)$ adds dependent types.
- The rule $(\Box^p, *^p, *^p)$ adds formula polymorphism.

Finally, the rule $(*^t, \Box^p, *^t)$ is the new version of the rule $(*, \Box, *)$ in NTT. It says that products of the form $\tau \to *^p$, where τ is a type, are types themselves. Remember that $*^p$ is the sort of formulas, thus $\tau \to *^p$ is a powerset. The rule expresses the fact that powersets are types or, reading it at the object level, that subsets are objects. One may easily note that LNTT is a logical Pure Type System.

The strong normalization property of the system was proved in [34]. This implies that the system is consistent. The proof technique used is a translation to the Calculus of Constructions. Some parts of the translation are used in this work.

As pointed out by A. Miquel, LNTT can be embedded in his system called $F\omega.2$ [42]. However, in LNTT we distinguish between sorts $*^t$, introducing object terms, and $*^p$, introducing proof terms. Our classification of terms is thus more fine-grained.

1.6. Overview

In Chapter 2 we discuss paradoxes in type theories. We define Girard's system U, perhaps the most know paradoxical type system. We analyze a paradox in Naive Type Theory, our first attempt to build a framework for naive type theory. The paradox is essentially the same as Girard's paradox so we omit the details of the latter. Then we present paradoxes in type system with inductive types: a paradox in the system with non-positive constructors and a paradox in a system with strong elimination over large inductive types.

In Chapter 3 we define Less Naive Type Theory with inductive types. We present the syntax of the system, the reduction and the typing rules. We introduce some terminology we will use in later parts of this work. Then we prove basic properties of the system.

In Chapter 4 we define a translation from LNTT with inductive types to the Calculus of Inductive Constructions. The translation only deals with non-proofs of the system. It preserves the reduction relation and thus proves the strong normalization property for non-proofs.

Chapter 5 is the strong normalization proof for the full system. We use Girard's candidates [28] but in a typed setting, first introduced in [13]. The proof combines ideas from [23] as well as [54].

Chapter 2

Paradoxes in type theories

When creating a new type system one has to be careful, it is very easy to define a system which is inconsistent. Sometimes one may encounter a contradiction. This is the case for the type system with the sort $*$ and the axiom $* : *$. The axiom expresses the slogan "Type is a type" and is indeed very similar to the naive set theory concept of the set of all sets. But often the inconsistency is not visible at first glance.

In this chapter we present a few well known paradoxical systems. We begin with the Pure Type Systems: Girard's System U and Naive Type Theory. NTT was our first attempt to implement the slogan "subsets are objects" and it turned out to be wrong. We present the proof that NTT is inconsistent. Then we proceed to systems with inductive types. We show paradoxes in the system with non-positive constructors and in the system with strong elimination on large inductive constructors.

2.1. Girard's Paradox

Girard's System U is perhaps the most known paradoxical type system. It was introduced and proved to be inconsistent by Girard in 1972 in [28]. The paradox also showed that the first version of Martin-Löf type theory [41] was inconsistent [45]. System U is a Pure Type System with the sorts $*$, $\Box$, $\triangle$, the axioms $* : \Box$ and $\Box : \triangle$ and the rules

$$(*, *, *), (\Box, *, *), (\Box, \Box, \Box), (\triangle, *, *), (\triangle, \Box, \Box).$$

This example shows that the circularity provided by the axiom $* : *$ is not necessary to get a contradictory system. Girard's paradox was analysed and discussed, for instance in Coquand [11], Hurkens [30] and Barendregt [5].

In the next section we present an inconsistency proof for Naive Type Theory. This proof is essentially the same as the proof of Girard's paradox.

2.2. Naive Type Theory

Naive Type Theory was our first attempt to formalize the system with powersets as types. It is a Pure Type System with the sorts $*$ and $\Box$, the axiom $* : \Box$ and the rules $(*, *, *)$, $(*, \Box, \Box)$ and $(*, \Box, *)$. However, as observed by H. Geuvers [25], this system is inconsistent. The proof is essentially the same as for Girard's paradox. The proof we present below is based on the formalization of the Burali-Forti paradox in [30]. Precisely, this is the formalization in [49].

We shall think of the sort $*$ as the sort of propositions. Then the type

$$\mathsf{P}(\tau) = \tau \to *$$

is the set of all predicates on type τ, i.e. its powerset.

We will use the notation $\forall x : T_1.T_2$ to denote $\Pi x : T_2.T_2$ if the product is a proposition. Logical connectives $\bot$, $\wedge$, $\neg$, $\leftrightarrow$, $\exists$ will also be used. We define them in a similar way as in the system F. The only problem is that in Naive Type Theory there is no type polymorphism. However, it may be simulated using the powerset rule. Assume we have an arbitrary type T and its inhabitant $a : T$. Instead of the variable $p : *$ we will use the variable $p : T \to *$. Thus we have the following definitions:

$$\begin{aligned}
\bot &\equiv \forall p : T \to *.pa,\\
\neg\alpha &\equiv \alpha \to \bot,\\
\alpha \wedge \beta &\equiv \forall p : T \to *.(\alpha \to \beta \to pa) \to pa,\\
\alpha \leftrightarrow \beta &\equiv (\alpha \to \beta) \wedge (\beta \to \alpha),\\
\exists x : \tau.\varphi(x) &\equiv \forall p : T \to *.(\forall x : \tau.\varphi(x) \to pa) \to pa.
\end{aligned}$$

We introduce the following abbreviations:

$$\begin{aligned}
\langle M, N\rangle &\equiv \lambda p : T \to *.\lambda u : \alpha \to \beta \to pa.uMN,\\
\pi_i(M) &\equiv M(\lambda x : T.\alpha_i)(\lambda x_1 : \alpha_1 x_2 : \alpha_2.x_i),\\
[b, q]_{\exists x:\tau.\varphi(x)} &\equiv \lambda p : T \to *\lambda u : (\forall x : \tau.(\varphi(x) \to pa)).ubq,\\
\text{let } [b, q] = y \text{ in } N^\tau &\equiv yT(\lambda b : \tau\lambda u : \varphi(b).N).
\end{aligned}$$

One may observe that the types of the terms are as expected:

$$\begin{gathered}
\Gamma, M : \alpha, N : \beta \vdash \langle M, N\rangle : \alpha \wedge \beta,\\
\Gamma, M : \alpha_1 \wedge \alpha_2 \vdash \pi_i(M) : \alpha_i,\\
\Gamma, b : \tau, q : \varphi(b) \vdash [b, q]_{\exists x:\tau.\varphi(x)} : \exists x : \tau.\varphi(x),
\end{gathered}$$

and that the following rule is admissible

$$\frac{\Gamma \vdash y : (\exists x : \tau.\varphi(x)) \qquad \Gamma, b : \tau, q : \varphi(b) \vdash N : T}{\Gamma \vdash \text{let } [b, q] = y \text{ in } N : T}$$

Moreover

$$\pi_i(\langle M_1, M_2\rangle) \to^*_\beta M_i.$$

and

$$\text{let } [a, q] = [b, p]_{\exists x:\tau.\varphi(x)} \text{ in } N \to^*_\beta N[x := a][x := b].$$

We define Leibniz equality in type κ in the usual way:

$$\alpha =^\kappa \beta \quad \equiv \quad \forall \gamma : \kappa \to *.\gamma\alpha \to \gamma\beta.$$

For brevity, we will write $\lambda x^\tau.B$ and $\Pi x^\tau.B$ instead of $\lambda x : \tau.B$ and $\Pi x : \tau.B$.

First we show that the context

$$\Gamma = \{k : *, el : \mathsf{P}(k) \to k, set : k \to \mathsf{P}(k),$$
$$V : \forall X^{\mathsf{P}(k)} \forall \alpha^k (set(elX)\alpha \leftrightarrow \exists \beta^k (X\beta \wedge \alpha =^k el(set\beta)))\}$$

is inconsistent. We derive a contradiction similar to Russell's paradox: define the abbreviations

$$\alpha' \quad \text{for} \quad el(set\alpha) \qquad \text{and} \qquad \alpha \simeq \beta \quad \text{for} \quad \forall R^{k\to k\to *}(EqvR \to \forall \gamma^k R\gamma\gamma' \to R\alpha\beta)$$

where Eqv is the property of being an equivalence relation:

$$Eqv = \lambda R : k \to k \to *. \forall \alpha^k \beta^k \gamma^k (R\alpha\alpha \wedge (R\alpha\beta \to R\beta\alpha)) \wedge (R\alpha\beta \to R\beta\gamma \to R\alpha\gamma).$$

Then the relation $\simeq$ is the least equivalence relation on k such that $x \simeq el(set(x))$. Define a relation $\alpha\epsilon\beta$ as $\exists\gamma^k(\alpha \simeq \gamma \wedge set\beta\gamma)$ and $\alpha \not\epsilon \beta$ as $\neg(\alpha\epsilon\beta)$. Let $\Delta = el(\lambda x.x \not\epsilon x)$. We will prove that for each y in k we have $y\epsilon\Delta$ if and only if $y \not\epsilon y$.

We show that $\simeq$ is an equivalence relation.

Lemma 1. *In the context* Γ *we can prove* $Eqv(\lambda\alpha^k\beta^k.\alpha \simeq \beta)$.

Proof. We have to find terms A_1, A_2 and A_3 such that

$$\Gamma, \alpha : k, \beta : k, \gamma : k \vdash A_1 : \alpha \simeq \alpha,$$
$$\Gamma, \alpha : k, \beta : k, \gamma : k \vdash A_2 : \alpha \simeq \beta \to \beta \simeq \alpha,$$
$$\Gamma, \alpha : k, \beta : k, \gamma : k \vdash A_3 : \alpha \simeq \beta \to \beta \simeq \gamma \to \alpha \simeq \gamma.$$

Take

$$A_1 = \lambda R^{k\to k\to *} \lambda p^{EqvR} \lambda z^{\forall\gamma^k.R\gamma\gamma'}.\pi_1(\pi_1(p\alpha\alpha\alpha)),$$
$$A_2 = \lambda x^{\alpha\simeq\beta} \lambda R^{k\to k\to *} \lambda p^{EqvR} \lambda z^{\forall\gamma^k.R\gamma\gamma'}.\pi_1(\pi_2(p\alpha\beta\alpha))(xRpz),$$
$$A_3 = \lambda x^{\alpha\simeq\beta} \lambda y^{\beta\simeq\gamma}.\lambda R^{k\to k\to *} \lambda p^{EqvR} \lambda z^{\forall\gamma^k.R\gamma\gamma'}.\pi_2(p\alpha\beta\gamma)(xRpz)(yRpz).$$

Then $M = \lambda\alpha^k\beta^k\gamma^k.\langle\langle A_1, A_2\rangle, A_3\rangle$ is the proof we want. □

We will show that $\simeq$ is a congruence with respect to relation ϵ. We will need two auxiliary lemmas.

Lemma 2. *In the context* $\Gamma, \alpha : k, \beta : k$ *we can prove that* $set\alpha\beta \to set\alpha'\beta'$.

Proof. Observe that

$$\Gamma, \alpha : k, \beta : k \vdash V(set\alpha)\beta' : set\alpha'\beta' \leftrightarrow \exists\gamma^k(set\alpha\gamma \wedge \beta' =^k \gamma').$$

Recall that $\leftrightarrow$ is a conjunction of two implications. If I is a proof of $\beta' =^k \beta'$ then

$$\Gamma, \alpha : k, \beta : k, u : set\alpha\beta \vdash \pi_2(V(set\alpha)\beta')[\beta, \langle u, I\rangle]_{\exists\gamma(set\alpha\gamma\wedge\beta'=^k\gamma')} : set\alpha'\beta'.$$

From this we easily get the desired proof. □

Lemma 3. *In the context* $\Gamma, \alpha : k, \beta : k$ *we can prove that* $\beta\epsilon\alpha \leftrightarrow \beta\epsilon\alpha'$.

Proof. Observe that Lemma 2 implies that there is N_1 such that

$$\Gamma, \alpha : k, \beta : k, \gamma : k \vdash N_1 : (\beta \simeq \gamma \wedge set\alpha\gamma) \rightarrow (\beta \simeq \gamma' \wedge set\alpha'\gamma')$$

and thus there exists N_2 such that

$$\Gamma, \alpha : k, \beta : k \vdash N_2 : \exists\gamma^k(\beta \simeq \gamma \wedge set\alpha\gamma) \rightarrow \exists\gamma^k(\beta \simeq \gamma \wedge set\alpha'\gamma).$$

We now prove the other implication. Assume that $\exists\gamma^k(\beta \simeq \gamma \wedge set\alpha'\gamma)$. Observe that $set\alpha'\gamma \equiv set(el(set\alpha))\gamma$ is equivalent to $\exists\delta^k(set\alpha\delta \wedge \gamma =^k \delta')$. We have δ such that $set\alpha\delta$. Moreover $\gamma =^k \delta'$ and $\beta \simeq \gamma$. This implies $\beta \simeq \delta'$. But $\simeq$ is an equivalence relation and $\delta \simeq \delta'$ thus $\beta \simeq \delta$. Hence indeed $\exists\delta^k(\beta \simeq \delta \wedge set\alpha\delta)$. □

Now we prove that $\simeq$ is a congruence with respect to the relation ϵ.

Lemma 4. *In the context* $\Gamma, \alpha : k, \beta : k$ *we can prove that*

1. $\alpha \simeq \beta \rightarrow \forall\delta^k(\alpha\epsilon\delta \rightarrow \beta\epsilon\delta)$;
2. $\alpha \simeq \beta \rightarrow \forall\gamma^k(\gamma\epsilon\alpha \rightarrow \gamma\epsilon\beta)$.

Proof. 1. Recall that $\alpha\epsilon\delta$ is an existential type. It is easy to observe that

$$\Gamma, \alpha : k, \beta : k, x : \alpha \simeq \beta, \delta : k, y : \alpha\epsilon\delta \vdash B : \beta\epsilon\delta$$

where

$$B = \text{let } [\gamma, z : (\alpha \simeq \gamma \wedge set\delta\gamma)] = y \text{ in } [\gamma, \langle A_3(A_2x)(\pi_1(z)), \pi_2(z)\rangle]_{\beta\epsilon\delta},$$

and A_2 and A_3 are defined in the proof of Lemma 1. Then $\lambda x^{\alpha\simeq\beta}\lambda\delta^k.B$ is the desired proof.

2. Take $R = \lambda\alpha^k\beta^k.\forall\gamma^k(\gamma\epsilon\alpha \leftrightarrow \gamma\epsilon\beta)$. From Lemma 3 we know that $\Gamma, \alpha : k \vdash R\alpha\alpha'$. It suffices to show $Eqv(R)$ and apply the definition of $\simeq$. This is easy.

□

Lemma 5. *Let* $\Delta = el(\lambda\alpha^k.\alpha \not\epsilon \alpha)$. *Then in the context* $\Gamma, \beta : k$ *we have* $\beta\epsilon\Delta \leftrightarrow \beta \not\epsilon \beta$.

Proof. In one direction, note that $\beta \not\epsilon \beta$ implies $\exists\gamma^k(\gamma \not\epsilon \gamma \wedge \beta' =^k \gamma')$ which is equivalent to $set\Delta\beta'$. Since $\beta \simeq \beta'$ we conclude $\beta\epsilon\Delta$. In the other direction: we want to prove that the assumptions $\beta \simeq \gamma$ and $set\Delta\gamma$ contradict $\beta\epsilon\beta$. But $set\Delta\gamma$ implies $\exists\delta^k(\gamma \simeq \delta' \wedge \delta \not\epsilon \delta)$ so it suffices to infer $\bot$ from the set $\{\beta \simeq \gamma, \beta\epsilon\beta, \gamma \simeq \delta', \delta \not\epsilon \delta\}$. This is a consequence of Lemma 4 because $\beta \simeq \gamma \simeq \delta'$ implies $\beta \simeq \delta$. □

If we take $\beta = \Delta$ then we get that in the context Γ the equivalence $\Delta\epsilon\Delta \leftrightarrow \Delta \not\epsilon \Delta$ holds which leads to a contradiction.

We now implement the context Γ in NTT. Take

$$k = \forall\kappa : T \rightarrow *.(\forall\iota : T \rightarrow *((\iota a \rightarrow \kappa a) \rightarrow \mathsf{P}(\iota a) \rightarrow \kappa a)) \rightarrow \kappa a.$$

and the functions el and set:

$$el = \lambda X^{\mathsf{P}(k)}.\lambda\kappa^{T\rightarrow *}\lambda y^{\forall\iota:T\rightarrow *((\iota a\rightarrow\kappa a)\rightarrow\mathsf{P}(\iota a)\rightarrow\kappa a))}.y(\lambda x : T.k)(\lambda\beta^k.\beta\kappa y)X,$$

$$set = \lambda\beta^k.\beta(\lambda x : T.\mathsf{P}(k))\psi,$$

where

$$\psi = \lambda\iota^{T\to *}\lambda f^{\iota a\to \mathsf{P}(\kappa)}\lambda X : \mathsf{P}(\iota a).\lambda\alpha^{\kappa}.\exists\beta^{\iota a}.(X\beta \wedge \alpha =^{\kappa} el(f\beta)).$$

Then

$$\begin{aligned} set(elX)\alpha &=_\beta (\lambda\beta^k.\beta(\lambda x : T.(\mathsf{P}(k))\psi))(elX)\alpha =_\beta elX(\lambda x : T.\mathsf{P}(k))\psi\alpha \\ &=_\beta \psi(\lambda x : T.k)(\lambda\beta^k.\beta(\lambda x : T.\mathsf{P}(k))\psi)X\alpha = \psi(\lambda x : T.k)setX\alpha \\ &=_\beta \exists\beta^{\iota a}(X\beta \wedge \alpha =^k el(set\beta)). \end{aligned}$$

Then we have

$$set(elX)\alpha =_\beta \exists\beta^{\iota a}(X\beta \wedge \alpha =^k el(set\beta))$$

which is even more then the equivalence we wanted.

2.3. A paradox with non-positive constructors

The positivity condition says that for every constructor of an inductive type I its recursive arguments must be of the form $\Pi\vec{x} : \vec{T}.I$ where I does not appear in $\vec{T}$. We will show that lifting this restriction can lead to inconsistency. This result may be found e.g. in [55].

Consider the Calculus of Inductive Constructions without the positivity restriction. Consider a type $\mathrm{Empty} = \mathrm{Ind}(X : *)\{(X \to X) \to X\}$. Let lam denote $\mathrm{Constr}(0, \mathrm{Empty})$. We define a term $A : \mathrm{Empty} \to \mathrm{Empty}$:

$$\begin{aligned} A = \lambda x : \mathrm{Empty}.\,&\mathrm{Elim}(\mathrm{Empty}, \mathrm{Empty}, x) \\ &\{\lambda f : \mathrm{Empty} \to \mathrm{Empty}.\lambda q : \mathrm{Empty} \to \mathrm{Empty}.f(lam\ f)\}. \end{aligned}$$

Consider the object $a = lam\ A$. Then

$$Aa \to^*_{\beta\iota} (\lambda f\lambda q.f(lam\ f))AR \to^*_\beta A(lam\ A) = Aa$$

where R is the recursive call

$$\begin{aligned} R = \lambda x : \mathrm{Empty}.\,&\mathrm{Elim}(\mathrm{Empty}, \mathrm{Empty}, Ax) \\ &\{\lambda f : \mathrm{Empty} \to \mathrm{Empty}.\lambda q : \mathrm{Empty} \to \mathrm{Empty}.f(lam\ f)\}. \end{aligned}$$

Hence the system is not strongly normalizing. It is also easy to derive an arbitrary predicate P using the type Empty:

$$x : \mathrm{Empty}, P : * \vdash \mathrm{Elim}(\mathrm{Empty}, P, x)\{\lambda f : \mathrm{Empty} \to \mathrm{Empty}.\lambda q : \mathrm{Empty} \to P.q(fx)\} : P.$$

If we take $x = lam\ (\lambda y : \mathrm{Empty}.y)$ then we get the term of type P.

2.4. A paradox with strong elimination on large constructors

We say that an inductive constructor is *small* if its type is of the form $T = \Pi\vec{x} : \vec{\tau}.X$ and every τ_i is a type. An inductive constructor is *large* if it is not small. An elimination of an inductive object is *strong* if it defines a type or a type constructor. Consider an extension of the Calculus of Inductive Construction in which we do not restrict strong elimination to small constructors. It is know that this system is inconsistent [11]. The proof we present here is a refined version of the proof by H. Geuvers [24], which again is a formalization of the proof by A. Hurkens [30].

We will first show that the following context is inconsistent:

$$\begin{aligned} B &: *, \\ b2p &: B \to *, \\ p2b &: * \to B, \\ H &: \forall A : *(A \leftrightarrow b2p(p2b(A))). \end{aligned}$$

As before, we will use the notation $\forall x^{\tau}.\varphi$ for $\forall x : \tau.\varphi$ and $\lambda x^{\tau}.\varphi$ for $\lambda x : \tau.\varphi$. Take the following proposition

$$U = \Pi A : *(((A \to B) \to A) \to (A \to B)).$$

We will define two terms $el : \mathsf{P}(U) \to U$ and $set : U \to \mathsf{P}(U)$ so that

$$\forall X^{U\to *} \forall \alpha^{U} (set(elX)\alpha \leftrightarrow \exists \beta (X\beta \wedge \alpha = (el(set\beta)))).$$

The definition of el follows

$$el = \lambda X^{\mathsf{P}(U)} \lambda A : *.\lambda c^{((A\to B)\to A)} \lambda a^{A}.p2b(\forall P^{A\to *}((\forall x^{U}(Xx \to P(c(xAc)))) \to Pa)).$$

We also define an auxiliary term $el' : (U \to B) \to U$:

$$el' = \lambda X^{U\to B} \lambda A : *.\lambda c^{((A\to B)\to A)} \lambda a^{A}.p2b(\forall P^{A\to *}((\forall x^{U}(b2p(Xx) \to P(c(xAc)))) \to Pa))$$

and the term set is as follows:

$$set = \lambda x^{U}.\lambda b^{U}.b2p(xUel'b).$$

Note that $el'(xUel') = el(set\ x)$:

$$\begin{aligned} &el'(xUel') \\ &\quad = \lambda A : *.\lambda c^{((A\to B)\to A)} \lambda a^{A}.p2b(\forall P^{A\to *}((\forall u^{U}(b2p(xUel'u) \to P(c(uAc)))) \to Pa)) \end{aligned}$$

and

$$\begin{aligned} &el(set\ x) \\ &\quad = \lambda A : *.\lambda c^{((A\to B)\to A)} \lambda a^{A}.p2b(\forall P^{A\to *}((\forall u^{U}((set\ x)u \to P(c(uAc)))) \to Pa)) \\ &\quad = \lambda A : *.\lambda c^{((A\to B)\to A)} \lambda a^{A}.p2b(\forall P^{A\to *}((\forall u^{U}((b2p(xUel'u) \to P(c(uAc)))) \to Pa)). \end{aligned}$$

Let α be a term of type U. We will prove that $set(elX)\alpha \leftrightarrow \exists \beta(X\beta \wedge \alpha = el(set\beta))$. Observe that

$$\begin{aligned} set(elX)\alpha &= b2p(elXUel'\alpha) \\ &= b2p(p2b(\forall P^{U\to *}((\forall x^{U}(Xx \to P(el(set\ x)))) \to P\alpha))). \end{aligned}$$

We have to prove two implications. First we prove $set(elX)\alpha \to \exists \beta(X\beta \wedge \alpha = el(set\beta))$. Assume $set(elX)\alpha$. We want to prove $\exists \beta(X\beta \wedge \alpha = el(set\beta))$. The assumption is equal to

$b2p(p2b(\forall P^{U\to *}((\forall x^U(Xx \to P(el(set\ x)))) \to P\alpha)))$. We use the assumption H to extract the proposition

$$\forall P^{U\to *}((\forall x^U(Xx \to P(el(set\ x)))) \to P\alpha).$$

We choose P so that the target of the proposition is the formula we want to get. Take

$$P = \lambda u^U.\exists\beta(X\beta \wedge u = el(set\beta)).$$

We now have to prove that for P as above we have

$$\forall x^U(Xx \to P(el(set\ x))).$$

This is easy: take x of type U and assume Xx. We want to prove that $\exists\beta(X\beta \wedge el(set\ x) = el(set\beta))$. Take $\beta = x$ and apply the assumption and the equality $el(set\ x) = el(set\beta)$. The proof term we get is:

$$\lambda z^{set(elX)\alpha}.(\pi_2(H(\forall P^{U\to *}((\forall x^U(Xx \to P(el'(xUel')))) \to P\alpha)))z)$$
$$(\lambda u^U.\exists\beta(X\beta \wedge u = el(set\beta)))(\lambda x^U.\lambda H^{Xx}.[x, \langle H, I\rangle])$$

Then we prove $\exists\beta(X\beta \wedge \alpha = el(set\beta)) \to set(elX)\alpha$. Assume $\exists\beta(X\beta \wedge \alpha = el(set\beta))$. We want to prove $set(elX)\alpha$, i.e. $b2p(p2b(\forall P^{U\to *}((\forall x^U(Xx \to P(el(set\ x)))) \to P\alpha)))$. We will prove the internal proposition $\forall P^{U\to *}((\forall x^U(Xx \to P(el(set\ x)))) \to P\alpha)$ and use the assumption H. Take $P : U \to *$ and assume $\forall x^U(Xx \to P(el(set\ x)))$. We want to prove $P\alpha$. Take β' such that $(X\beta' \wedge \alpha = el(set\beta'))$ and apply our assumption to β'. We know that $X\beta'$, thus by the assumption $P(el(set\ \beta'))$. But $el(set\ \beta') = \alpha$. Hence conclusion. The proof term we get with this reasoning is:

$$\lambda z^{\varphi_2}.(\pi_1(H(\forall P^{U\to *}((\forall x^U(Xx \to P(el(set\ x)))) \to P\alpha))))$$
$$(\lambda P^{U\to *}\lambda H^{\forall x^U(Xx\to P(el(set\ x)))}.\text{let } [b, H2] = z \text{ in } \pi_2(H2)(Hb\pi_1(H2)).$$

We can now implement the inconsistent context. Take

$$B = \text{Ind}(X : *)\{\Pi A : *.((A \to A) \to X)\}.$$

Define $p2b : * \to B$ as

$$p2b = \lambda D : *.\text{Constr}(0, B)D(\lambda x : D.x)$$

and $b2p : B \to *$ as

$$b2p = \lambda x : B.\pi_1(x)$$

where π_1 is defined using the strong elimination:

$$\pi_1 = \lambda x : B.\text{Elim}(B, \lambda x : B.*, x)\{\lambda a : *\lambda p : a \to a.a\}.$$

Then for every $D : *$ we have

$$b2p(p2bD) = \pi_1(\text{Constr}(0, B)D(\lambda x : D.x)) = (\lambda a : *\lambda p : a \to a.a)D(\lambda x : D.x) \to_{\beta\iota} D.$$

We have constructed the contradictory context.

Chapter 3

Less Naive Type Theory with inductive types

3.1. Notation

We introduce the notation which will be used in the rest of this work. First, $\lambda\!\!\lambda x : T.A(x)$ denotes a function with the domain T which takes an argument x and returns the value $A(x)$.

We will often deal with sequences, in particular with sequences of terms. We use the notation $\langle a_1, \ldots, a_n\rangle$ to denote a sequence of length n. As usual, we write $\vec{T}$ to highlight that $\vec{T}$ is a sequence of terms and the empty sequence is denoted by ϵ. We write T_i to denote the i-th element of the sequence $\vec{T}$. If $g = \vec{N}$ is a sequence of terms then a term of the form $M\vec{N}$ will sometimes be denoted by $M \cdot g$. Furthermore, we use the symbol $\mid$ to separate elements in the sequence, for example $(T_0 \mid T_1 \mid T_2)$ is a sequence of terms of length 3. If T_0 is an element and $\vec{T}$ is a sequence then we write $T_0 :: \vec{T}$ to denote the sequence which has T_0 as its first element and the elements of $\vec{T}$ in the following places. If $\vec{f}$ is a vector of functions then $\vec{f}(x)$ will denote the sequence $\langle f_0(x), f_1(x), \ldots, f_n(x)\rangle$. If f is a function and $\vec{x} = \langle x_0, x_1, \ldots, x_n\rangle$ is a sequence then $f(\vec{x})$ will denote $f(x_0, x_1, \ldots, x_n)$.

3.2. The terms

We have four sorts $*^t$, $*^p$, $\Box^t$ and $\Box^p$. The terms of the system are defined by the following grammar, where x stands for a variable and s stands for a sort.

$$T := s \mid x \mid (TT) \mid (\lambda x : T.T) \mid (\Pi x : T.T) \mid \mathrm{Ind}(x : T)\{\vec{T}\} \mid \mathrm{Constr}(n, T) \mid \mathrm{Elim}(T, T, \vec{T}, T)\{\vec{T}\}.$$

We use the same conventions for parentheses as for Pure Type System (compare page 9). When compared to PTSs the syntax is extended with the following constructions:

- $\mathrm{Ind}(x : T)\{\vec{T}\}$,
- $\mathrm{Constr}(n, T)$,
- $\mathrm{Elim}(T, T, \vec{T}, T)\{\vec{T}\}$.

In $\mathrm{Ind}(x : T_1)\{\vec{T_2}\}$ the expression T_1 is a type of the bound variable x and $\vec{T_2}$ is a sequence of types of the inductive constructors. We use the Coq inspired syntax, elements in a sequence are separated with |. Let us see some examples. We will refer to them through the rest of this work.

Example 6.

1. $\mathrm{Ind}(X : *^t)\{X \mid X \to X\}$ is a type of natural numbers. We denote it by Nat.
2. $\mathrm{Ind}(X : *^t)\{X \mid \tau \to X \to X\}$ is a type of lists over the type τ. We denote it by $List(\tau)$.
3. $\mathrm{Ind}(X : *^t)\{X \mid (Nat \to X) \to X\}$ is a type of trees in which every internal node is of degree ω. We denote it by $Tree$.
4. If S denotes the successor function then $\mathrm{Ind}(X : Nat \to *^p)\{X0 \mid \Pi n : Nat(Xn \to X(S(Sn)))\}$ is the predicate "even". We denote it by $Even$.
5. If S denotes the successor function, *leaf* denotes the empty tree and *node* denotes the tree node constructor then

$$\begin{aligned} FullTree = \mathrm{Ind}(X : Nat \to Tree \to *^p)\{&X\ 0\ leaf \mid \\ &\Pi f : Nat \to Tree.\Pi n : Nat.\Pi p : (\Pi m : Nat.Xn(fm)).X(S\ n)(node\ f)\} \end{aligned}$$

   is a binary inductive predicate which holds, for a number n and a tree t, when every path from root to leaf in t has the same length n.

A term $\mathrm{Constr}(n, I)$ is a constructor of an inductive object. Here I is an inductive type and n is a natural number indicating which inductive constructor is meant.

Example 7.

1. $\mathrm{Constr}(0, Nat)$ denotes the natural number 0;
2. $\mathrm{Constr}(1, Nat)\mathrm{Constr}(0, Nat)$ denotes the natural number 1;
3. $\mathrm{Constr}(0, List(Nat))$ denotes the empty list of natural numbers;
4. $\mathrm{Constr}(1, List(Nat))\ \mathrm{Constr}(0, Nat)\ \mathrm{Constr}(0, List(Nat))$ denotes a list of natural numbers of length 1.

A term $\mathrm{Elim}(I, Q, \vec{u}, c)\{\vec{t}\}$ is an eliminator of an inductive type or predicate I. If I is an inductive predicate then c is a term being eliminated, $\vec{u}$ is a sequence of parameters, $Q\vec{u}$ is a type of the result, and $\vec{t}$ is a vector of definitions corresponding to the inductive constructors of I (i.e. cases possible for c). If I is an inductive type then c is a term being eliminated, Q is such that Qc is a type of the result, the vector of parameters $\vec{u}$ is always empty, and $\vec{t}$ is a vector of definitions corresponding to the inductive constructors of I.

Notation 8. For an inductive type I we will use the notation $\mathrm{Elim}(I, Q, c)\{\vec{t}\}$.

Example 9.

1. $\mathrm{Elim}(Nat, \lambda x{:}Nat.Nat, n)\{0 \mid \lambda m : Nat.\lambda p : Nat.m\}$ is the predecessor of n.

2. If $\cup$ is the union operator and $\bot$ is the constant "false" then the following term defines the union of the sets in the list l

$$\mathrm{Elim}(List(\tau \to *^p), \lambda x : List(\tau \to *^p).(\tau \to *^p), l)$$
$$\{\lambda x : \tau.\bot \mid \lambda h : \tau \to *^p.\lambda t : List(\tau \to *^p)\lambda p : \tau \to *^p.h \cup p\}.$$

Recall that we use the Coq inspired syntax and we separate the elimination branches with $|$.

Example 10. Consider the inductive predicate *Even*. Suppose n is of type *Nat* and we want to prove that $Even(n) \to Even(S(S(S(S(n)))))$. Of course, we can use the constructor $\mathrm{Constr}(1, Even)$. For the sake of example we will show a proof with the use of the eliminator. Assume p is of type $Even(n)$. We will use abbreviations

$$E_4 = \mathrm{Constr}(1, Even)\ \underbrace{(S(S(0))}_{2}\ \underbrace{(\mathrm{Constr}(1, Even)\ 0\ \mathrm{Constr}(0, Even))}_{\text{proof of } Even(2)}$$

and

$$S^4(k) = S(S(S(S(k)))), \quad \text{for } k \text{ of type } Nat.$$

We thus get the following proof term:

$$\lambda p : Even(n).\mathrm{Elim}(Even, \lambda m : Nat.Even(S^4(m)), n, p)\{E_4 \mid$$
$$\lambda m : Nat \lambda r : Even(m)\lambda q : Even(S^4(m)).\mathrm{Constr}(1, Even)(S^4(m))q\}.$$

3.3. Additional definitions

For a term M we define the set of *free variables* of M (notation $FV(M)$) by induction with respect to the structure of M:

- $FV(x) = \{x\}$,
- $FV(\Pi x : A.B) = FV(A) \cup (FV(B) - \{x\})$,
- $FV(\lambda x : A.B) = FV(A) \cup (FV(B) - \{x\})$,
- $FV(AB) = FV(A) \cup FV(B)$,
- $FV(\mathrm{Ind}(X : A)\{\vec{C}\}) = FV(A) \cup (FV(\vec{C}) - \{X\})$,
- $FV(\mathrm{Constr}(n, I)) = FV(I)$,
- $FV(\mathrm{Elim}(I, Q, \vec{u}, M)\{\vec{f}\}) = FV(I) \cup FV(Q) \cup FV(\vec{u}) \cup FV(M) \cup FV(\vec{f})$.

The set $FV(\vec{t})$ of free variables of a sequence $\vec{t}$ of terms is defined as follows

- $FV(\epsilon) = \emptyset$,
- $FV(t_0 :: \vec{t}) = FV(t_0) \cup FV(\vec{t})$.

If M, N are terms and x is a variable then we define the term $M[x := N]$ (sometimes written $M[N/x]$) in the usual way. If M is a term, $\vec{x} = x_0, \ldots, x_n$ is a sequence of variables and $\vec{N} = N_0, \ldots, N_n$ is a sequence of terms then we define

$$M[\vec{x} := \vec{N}] = M[x_0 := N_0][x_1 := N_1] \ldots [x_n := N_n].$$

Let X be a variable. A term t is *strictly positive in* X if $t \equiv \Pi\vec{x} : \vec{t}.X\vec{t'}$, there are no free occurrences of X in $\vec{t}$ or $\vec{t'}$ and X does not occur among the variables of $\vec{x}$.

We say that a term $C(X)$ is a *type of constructor* in X if

- $C(X) = X\vec{t}$, or
- $C(X) = \Pi x : t.D(X)$, where $D(X)$ is a type of constructor in X, the variable X does not have free occurrences in t and $X \neq x$, or
- $C(X) = P \to D(X)$, where $D(X)$ is a type of constructor in X, and P is strictly positive in X.

Note that a type of constructor does not have to be a type. We say that a type of constructor $C(X)$ is *simple* if

- $C(X) = X$, or
- $C(X) = \Pi x : t.D(X)$ and $D(X)$ is a simple type of constructor in X,
- $C(X) = P \to D(X)$ and $D(X)$ is a simple type of constructor in X.

If $C(X)$ is a simple type of constructor then we define the type $\triangle\{C(X), Q, c\}$ by induction with respect to the structure of $C(X)$:

- $\triangle\{X, Q, c\} = Qc$,
- $\triangle\{\Pi x : t.D(X), Q, c\} = \Pi x : t.\triangle\{D(X), Q, (cx)\}$,
- $\triangle\{(\Pi\vec{x} : \vec{t}.X) \to D(X), Q, c\} = \Pi p : (\Pi\vec{x} : \vec{t}.X).\,\big(\Pi\vec{x} : \vec{t}.Q(p\vec{x})\big) \to \triangle\{D(X), Q, (cp)\}$.

The type $\triangle\{C(X), Q, c\}$ is used in rule $(Elim_{*^t})$ (see page 30). It helps to define a type of an elimination branch in a term $\mathrm{Elim}(I, Q, m)\{\vec{f}\}$, where $I = \mathrm{Ind}(X : *^t)\{\vec{C}(X)\}$. Every elimination branch f_i corresponds to a certain type of constructor $C_i(X)$. This type of constructor is a basis for the type of f_i. The variable c is auxiliary, it represents a partially constructed inductive object. We use the notation: $\triangle\{C(t), Q, c\}$ for $\triangle\{C(X), Q, c\}[X := t]$. We could as well define the term $\triangle\{C(X), Q, c\}$ for every type of constructor $C(X)$. However, we will only use this notion for simple types of constructor.

Example 11. Consider the type of natural numbers Nat. Recall the predecessor function we have seen in Example 9:

$$\mathrm{Elim}(Nat, \lambda x : Nat.Nat, n)\{0 \mid \lambda m : Nat.\lambda p : Nat.m\}.$$

Here $Q = \lambda x : Nat.Nat$. The type of the first constructor is Nat and the type of the first branch is

$$Nat = \triangle\{X, Q, 0\}[X := Nat, Q := \lambda x : Nat.Nat].$$

The type of the second constructor is $Nat \to Nat$ and the type of the second branch is

$$Nat \to Nat \to Nat = \triangle\{X \to X, Q, S\}[X := Nat, Q := (\lambda x : Nat.Nat)].$$

The first argument is the recursive argument and the second argument is the result of the recursive call of the function on the first argument.

Example 12. If we consider the type of lists $List(\tau \to *^p)$ and the example of the union function from Example 9 then the type of the first branch is $\tau \to *^p$ and the type of the second branch is

$$\underbrace{(\tau \to *^p)}_{\text{head}} \to \underbrace{List(\tau \to *^p)}_{\text{tail}} \to \underbrace{(\tau \to *^p)}_{\text{recursive call}} \to (\tau \to *^p).$$

The branch takes the head and tail of the list, and the result of the recursive call of the function and it returns the result of the function.

Let $C(X)$ be a type of constructor and Q a term. We define a type of the nondependent elimination branch for the inductive constructor of type $C(X)$, denoted $\triangle\{C(X), Q\}$, by induction with respect to the structure of $C(X)$:

- $\triangle\{X\vec{t'}, Q\} = Q\vec{t'}$,
- $\triangle\{\Pi x : t.D(X), Q\} = \Pi x : t.\triangle\{D(X), Q\}$,
- $\triangle\{\left(\Pi\vec{x} : \vec{t}.X\vec{t'}\right) \to D(X), Q\} = (\Pi\vec{x} : \vec{t}.X\vec{t'}) \to (\Pi\vec{x} : \vec{t}.Q\vec{t'}) \to \triangle\{D(X), Q\}$.

We use the notation $\triangle\{C(t), Q\}$ for $\triangle\{C(X), Q\}[X := t]$.

Example 13. Recall the proof of $Even(n) \to Even(S^4(n))$ we have seen in Example 10:

$$\begin{aligned}&\lambda p : Even(n).\mathrm{Elim}(Even, \lambda m : Nat.Even(S^4(m)), n, p)\{E_4 \mid \\ &\qquad \lambda m : Nat \lambda r : Even(m) \lambda q : Even(S^4(m)).\mathrm{Constr}(1, Even)(S^4(m))q\}.\end{aligned}$$

Observe that the term p is of type $Even(n)$ and $Q = \lambda m : Nat.Even(S^4(m))$. The vector of parameters has only one element: the term n.

- The first constructor $\mathrm{Constr}(0, Even)$ has type $Even(0)$ and the corresponding first branch has the type

  $$\begin{aligned}\triangle\{X0, Q\} = \triangle\{X0, (\lambda m : Nat.Even(S^4(m))\} &= (\lambda m : Nat.Even(S^4(m))0 \\ &= Even(S^4(0)) = Q0.\end{aligned}$$

- The second branch corresponds to the constructor $\mathrm{Constr}(1, Even)$ which has the type $\Pi m : Nat(Even(m) \to Even(S(S(m)))$. The type of the branch is

  $$\begin{aligned}&\triangle\{\Pi m : Nat(X(m) \to X(S(S(m)))), Q\}[X := Even, Q := \lambda m : Nat.Even(S^4(m))] \\ &\qquad = \Pi m : Nat\left(Even(m) \to Even(S^4(m)) \to Even(S^6(m))\right).\end{aligned}$$

  It takes three arguments: the first (non-recursive) argument of the constructor, the second (recursive) argument of type $Even(m)$ and the result of the recursive call which has the type $Even(S^4(m)) = Qm$. It returns a term of type $Even(S^6(m)) = Q(S^2(m))$.

3.4. The reduction rules

Let $C(X)$ be a type of constructor in X, let f, I, Q, c be terms and $\vec{N}$, $\vec{f}$ sequences of terms. Suppose $C(X) = \Pi\vec{x} : \vec{T}.X\vec{t'}$ and the vectors $\vec{T}$ and $\vec{N}$ have the same length. We define a term $\triangle[C(X), f, \vec{N}, I, Q, \vec{f}]$. The intended use is to define a reduction rule as follows

$$\mathrm{Elim}(I, Q, \vec{u}, \mathrm{Constr}(n, I')\vec{N})\{\vec{f}\} \rightarrow_\iota \triangle[C_n(I), f_n, \vec{N}, I, Q, \vec{f}].$$

An expression $\mathrm{Elim}(I, Q, \vec{u}, \mathrm{Constr}(n, I')\vec{N})\{\vec{f}\}$ reduces to the application of the term f_n to the sequence consisting of arguments $\vec{N}$ and appropriate calls to the operator $\mathrm{Elim}(I, Q, \vec{t}, N_i)\{\vec{f}\}$ for the recursive arguments N_i. The definition is by induction with respect to the structure of $C(X)$.

- $\triangle[X\vec{t'}, f, \epsilon, I, Q, \vec{f}] = f$,
- $\triangle[\Pi x : t.D(X), f, (N_0 :: \vec{N}), I, Q, \vec{f}] = \triangle[D(X), fN_0, \vec{N}, I, Q, \vec{f}]$,
- $\triangle[(\Pi\vec{x} : \vec{t}.X\vec{t'}) \rightarrow D(X), f, (N_0 :: \vec{N}), I, Q, \vec{f}] =$
  $\triangle[D(X), fN_0(\lambda\vec{x} : \vec{t}.\mathrm{Elim}(I, Q, \vec{t'}, N_0\vec{x})\{\vec{f}\}), \vec{N}, I, Q, \vec{f}]$.

The reduction relation is the context closure of the following base cases:

- $(\lambda x : T.t_1)t_2 \rightarrow_\beta t_1[x := t_2]$.
- $\mathrm{Elim}(I, Q, \vec{u}, \mathrm{Constr}(n, I')\vec{N})\{\vec{f}\} \rightarrow_\iota \triangle[C_n(I), f_n, \vec{N}, I, Q, \vec{f}]$.

In the last rule, I and I' may be different. However, the typing rules ensure that for well typed terms it holds that $I =_{\beta\iota} I'$.

We use common notational conventions. The one-step reduction will be denoted by $\rightarrow_{\beta\iota}$. The transitive closure of the relation will be denoted by $\rightarrow^+_{\beta\iota}$ and the transitive-reflexive closure will be denoted by $\rightarrow^*_{\beta\iota}$.

Example 14. We compute the predecessor of $S(0)$:

$$\begin{aligned}
&\mathrm{Elim}(Nat, \lambda x : Nat.Nat, S(0))\{0 \mid \lambda m : Nat.\lambda p : Nat.m\}\\
&\qquad \rightarrow_\iota (\lambda m : Nat.\lambda p : Nat.m)\ 0\ \mathrm{Elim}(Nat, \lambda x : Nat.Nat, 0)\{0 \mid \lambda m : Nat.\lambda p : Nat.m\}\\
&\qquad \rightarrow_\iota (\lambda m : Nat.\lambda p : Nat.m)\ 0\ 0\\
&\qquad \rightarrow_\iota (\lambda p : Nat.0)\ 0\\
&\qquad \rightarrow_\beta\ 0.
\end{aligned}$$

Example 15. Using the notation from Example 10 we compute the proof of $Even(S^4(0))$:

$$\begin{aligned}
&\mathrm{Elim}(Even, \lambda m : Nat.Even(S^4(m)), 0, \mathrm{Constr}(0, Even))\{E_4 \mid\\
&\qquad\qquad \lambda m : Nat\lambda r : Even(m)\lambda q : Even(S^4(m)).\mathrm{Constr}(1, Even)(S^4(m))q\} \rightarrow_\iota E_4.
\end{aligned}$$

Observe that $\triangle[C(X), f, \vec{N}, I, Q, \vec{f}]$ is always of the form $f\vec{e}$ where $\vec{e}$ is a vector. For convenience, we use the notation $\vec{e}[C(X), \vec{N}, I, Q, \vec{f}]$ for this vector. The elements of the sequence are either elements of the sequence $\vec{N}$ (we use the notation $(\vec{e}[C(X), \vec{N}, I, Q, \vec{f}])_m$ to denote N_m) or recursive calls for those elements (we use the notation $(\vec{e}[C(X), \vec{N}, I, Q, \vec{f}])^R_m$ to denote the recursive call associated with N_m). The variable X does not occur in the sequence $\vec{e}[C(X), \vec{N}, I, Q, \vec{f}]$, the type $C(X)$ is used only as induction parameter.

For $C(X)$ a type of constructor in X, a sequence of terms $\vec{N}$, and terms I, Q, $\vec{f}$ we define

$$\vec{e}[C(X), \vec{N}, I, Q, \vec{f}] = \vec{e}[C(X), \vec{N}, I, Q, \vec{f}, 0]$$

where $\vec{e}[C(X), \vec{N}, I, Q, \vec{f}, k]$ is defined by induction with respect to the structure of $C(X)$:

- $\vec{e}[X\vec{t'}, \epsilon, I, Q, \vec{f}, k] = \epsilon$
- $\vec{e}[\Pi x : \tau.D(X), N_0 :: \vec{N}, I, Q, \vec{f}, k] = N_0 :: \vec{e}[D(X), \vec{N}, I, Q, \vec{f}, k+1]$ if $X \notin FV(\tau)$. In this case,
$$(\vec{e}[\Pi x : \tau.D(X), N_0 :: \vec{N}, I, Q, \vec{f}])_k = N_0.$$
- $\vec{e}[\Pi x : \tau.D(X), N_0 :: \vec{N}, I, Q, \vec{f}, k] = N_0 :: (\lambda\vec{y} : \vec{\sigma}.\mathrm{Elim}(I, Q, \vec{t'}, N_0\vec{y})\{\vec{f}\}) ::$ $\vec{e}[D(X), \vec{N}, I, Q, \vec{f}, k+1]$ if $\tau = \Pi\vec{y} : \vec{\sigma}.X\vec{t'}$. In this case,
$$(\vec{e}[\Pi x : \tau.D(X), N_0 :: \vec{N}, I, Q, \vec{f}])_k = N_0,$$
$$(\vec{e}[\Pi x : \tau.D(X), N_0 :: \vec{N}, I, Q, \vec{f}])^R_k = \lambda\vec{y} : \vec{\sigma}.\mathrm{Elim}(I, Q, \vec{t'}, N_0\vec{y})\{\vec{f}\}.$$

3.5. The typing rules

A *context* is a sequence of pairs of the form $x : T$ where x is a variable and T is a term. Contexts will be denoted using Greek letters Γ, Δ, Σ with appropriate subscripts and superscripts, where necessary. Moreover, Γ_1, Γ_2 denotes the concatenation of two contexts. In the following we consider contexts where every variable occurs at most once. Then $\Gamma(x)$ denotes the term associated with the variable x in Γ, that is if $\Gamma = \Gamma_1, x : A, \Gamma_2$ then $\Gamma(x) = A$. The set of variables in the context is called the *domain of the context* (notation $dom(\Gamma)$). We define a relation $\subseteq$ for contexts: we write $\Gamma \subseteq \Gamma'$ if Γ is a subsequence of Γ'. Note that a subsequence is not necessarily a prefix.

In addition to PTS rules we have new rules for inductive types.

$$(Ind_{*^t})\ \frac{\Gamma, X : *^t \vdash C_i(X) : *^t}{\Gamma \vdash \mathrm{Ind}(X : *^t)\{\vec{C}(X)\} : *^t}$$

If $A = \Pi\vec{x} : \vec{T}.*^p$ and $s \in \{\Box^p, *^t\}$ then we have the rule

$$(Ind_{*^p})\ \frac{\Gamma \vdash A : s \qquad \Gamma, X : A \vdash C_i(X) : *^p}{\Gamma \vdash \mathrm{Ind}(X : A)\{\vec{C}(X)\} : A}$$

In the rules (Ind_{*^t}) and (Ind_{*^p}) we additionally assume that every $C_i(X)$ is a type of constructor in X.
In the rule $(Intro_{*^t})$ the term I denotes $\mathrm{Ind}(X : *^t)\{\vec{C}(X)\}$.

$$(Intro_{*^t})\ \frac{\Gamma \vdash I : *^t}{\Gamma \vdash \mathrm{Constr}(n, I) : C_n(I)}$$

In the rule $(Intro_{*^p})$ the term I denotes $\mathrm{Ind}(X : A)\{\vec{C}(X)\}$ where $A = \Pi\vec{x} : \vec{\tau}.*^p$.

$$(Intro_{*^p})\ \frac{\Gamma \vdash I : A}{\Gamma \vdash \mathrm{Constr}(n, I) : C_n(I)}$$

In the rule $(Elim_{*^t})$ the term I denotes $\mathrm{Ind}(X : *^t)\{\vec{C}(X)\}$.

$$(Elim_{*^t})\ \frac{\Gamma \vdash t : I \qquad \Gamma \vdash Q : I \to s \qquad \Gamma \vdash f_n : \triangle\{C_n(I), Q, \mathrm{Constr}(n, I)\}}{\Gamma \vdash \mathrm{Elim}(I, Q, t)\{\vec{f}\} : Qt}$$

We give the typing rule for elimination. In this rule the term I denotes $\mathrm{Ind}(X : A)\{\vec{C}(X)\}$ and $A = \Pi\vec{x} : \vec{T}.*^p$.

$$(Elim_{*^p})\ \frac{\Gamma \vdash I\vec{u} : *^p \qquad \Gamma \vdash t : I\vec{u} \qquad \Gamma \vdash Q : A \qquad \Gamma \vdash f_n : \triangle\{C_n(I), Q\}}{\Gamma \vdash \mathrm{Elim}(I, Q, \vec{u}, t)\{\vec{f}\} : Q\vec{u}}$$

We have introduced the new reduction rule. Thus the Conversion rule has to be changed accordingly:

$$(\mathrm{Conv})\ \frac{\Gamma \vdash A : B \qquad \Gamma \vdash B' : s \qquad B =_{\beta\iota} B'}{\Gamma \vdash A : B'}$$

All rules of the system, including the PTS rules, are shown in Figure 3.5 on page 31.

We say that a sequent $\Gamma' \vdash A' : B'$ is *structurally smaller than sequent* $\Gamma \vdash A : B$ if it occurs in a derivation tree of $\Gamma \vdash A : B$.

3.6. The classification of terms

Let Γ be a context and M be a term. We say that

- M is *typable in the context* Γ if there exists T such that $\Gamma \vdash M : T$.
- M is a *kind in the context* Γ if $\Gamma \vdash M : \Box^t$.
- M is a *type in the context* Γ if $\Gamma \vdash M : *^t$.
- M is a *formula in the context* Γ if $\Gamma \vdash M : *^p$.
- M is a *type constructor in the context* Γ if there exists a term T such that $\Gamma \vdash M : T$ and $\Gamma \vdash T : \Box^t$.
- M is an *object in the context* Γ if there exists a term T such that $\Gamma \vdash M : T$ and $\Gamma \vdash T : *^t$.
- M is a *proof in the context* Γ if there exists a term T such that $\Gamma \vdash M : T$ and $\Gamma \vdash T : *^p$.
- M is a *powerset* if $M =_{\beta\iota} \Pi\vec{x} : \vec{\tau}.*^p$.
- M is a *subset in the context* Γ if there exists a powerset T such that $\Gamma \vdash M : T$.

$$(\mathrm{Ax}) \quad \vdash s_1 : s_2 \qquad s_1 : s_2 \in \mathcal{A}$$

$$(\mathrm{Var}) \ \frac{\Gamma \vdash A : s}{\Gamma, x : A \vdash x : A} \quad x \notin dom(\Gamma)$$

$$(\mathrm{Weak}) \ \frac{\Gamma \vdash A : B \qquad \Gamma \vdash C : s}{\Gamma, x : C \vdash A : B} \quad x \notin dom(\Gamma)$$

$$(\mathrm{App}) \ \frac{\Gamma \vdash M : (\Pi x : A.B) \qquad \Gamma \vdash N : A}{\Gamma \vdash MN : B[x := N]}$$

$$(\mathrm{Abs}) \ \frac{\Gamma, x : A \vdash M : B \qquad \Gamma \vdash (\Pi x : A.B) : s}{\Gamma \vdash (\lambda x : A.M) : (\Pi x : A.B)}$$

$$(\mathrm{Prod}) \ \frac{\Gamma \vdash A : s_1 \qquad \Gamma, x : A \vdash B : s_2}{\Gamma \vdash (\Pi x : A.B) : s_3} \quad (s_1, s_2, s_3) \in \mathcal{R}$$

$$(Ind_{*^t}) \ \frac{\Gamma, X : *^t \vdash C_i(X) : *^t}{\Gamma \vdash \mathrm{Ind}(X : *^t)\{\vec{C}(X)\} : *^t}$$

$$(Ind_{*^p}) \ \frac{\Gamma \vdash A : s \qquad \Gamma, X : A \vdash C_i(X) : *^p}{\Gamma \vdash \mathrm{Ind}(X : A)\{\vec{C}(X)\} : A}$$

$$(Intro_{*^t}) \ \frac{\Gamma \vdash I : *^t}{\Gamma \vdash \mathrm{Constr}(n, I) : C_n(I)}$$

$$(Intro_{*^p}) \ \frac{\Gamma \vdash I : A}{\Gamma \vdash \mathrm{Constr}(n, I) : C_n(I)}$$

$$(Elim_{*^t}) \ \frac{\Gamma \vdash t : I \qquad \Gamma \vdash Q : I \to s \qquad \Gamma \vdash f_n : \triangle\{C_n(I), Q, \mathrm{Constr}(n, I)\}}{\Gamma \vdash \mathrm{Elim}(I, Q, t)\{\vec{f}\} : Qt}$$

$$(Elim_{*^p}) \ \frac{\Gamma \vdash I\vec{u} : *^p \qquad \Gamma \vdash t : I\vec{u} \qquad \Gamma \vdash Q : (\Pi\vec{x} : \vec{T}.*^p) \qquad \Gamma \vdash f_n : \triangle\{C_n(I), Q\}}{\Gamma \vdash \mathrm{Elim}(I, Q, \vec{u}, t)\{\vec{f}\} : Q\vec{u}}$$

$$(\mathrm{Conv}) \ \frac{\Gamma \vdash A : B \qquad \Gamma \vdash B' : s \qquad B =_{\beta\iota} B'}{\Gamma \vdash A : B'}$$

Figure 3.1: Rules of the system

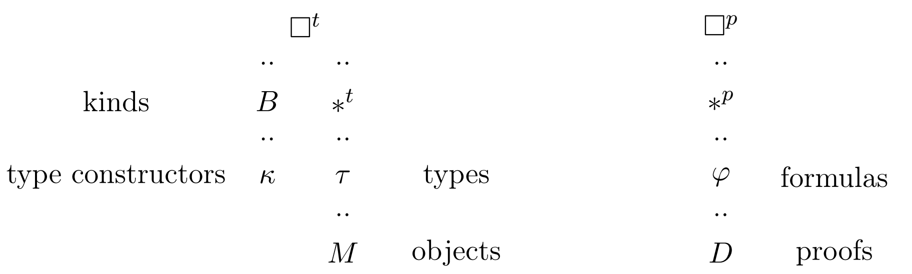

Figure 3.2: Structure of the system

We say that M is a kind, a type, a type constructor, etc. if the context Γ is known. We use the notation $Kind^\Gamma$, $Type^\Gamma$, $Formula^\Gamma$, $TConstr^\Gamma$, Obj^Γ, $Proof^\Gamma$, $Powerset^\Gamma$, $Subset^\Gamma$ to denote respectively kinds, types, formulas, etc. in Γ. We denote the set of all terms typable in Γ by $Term^\Gamma$.

Figure 3.2 shows the structure of the system and illustrates the basic terminology introduced above.

As we see, there are two hierarchies in the system, the type hierarchy (kinds, types, type constructors and objects) and the logical hierarchy (formulas and proofs). However, those standard notions are not precise enough to describe the system, we need a more fine-grained terminology. First we want to distinguish inductive types and predicates:

- M is an *inductive type in the context* Γ if $M = \mathrm{Ind}(X : *^t)\{\vec{C}\}$.
- M is an *inductive predicate in the context* Γ if M is is of the form $M = \mathrm{Ind}(X : A)\{\vec{C}\}\vec{t}$ and $A = \Pi\vec{x} : \vec{\tau}.*^p$.

Then we divide the terms in the type hierarchy (i.e. those below $*^t$ in Figure 3.2) into large and small. We introduce *large inductive types* by induction as follows:

- M is a *large inductive type in the context* Γ if it has a constructor with type $C(X) = \Pi\vec{x} : \vec{\tau}.X$ such that there is an element τ_i in the sequence $\vec{\tau}$ which is a powerset or a type of the form $\tau_i = \Pi\vec{x} : \vec{\sigma}.I$ where I is a large inductive type.

We say that

- M is a *type with large inductive target in the context* Γ if there exists a large inductive type I such that $M =_{\beta\iota} \Pi\vec{x} : \vec{\tau}.I$. In particular, a large inductive type is a type with a large inductive target.
- M is a *large inductive object in the context* Γ if there exists a large inductive type T such that $\Gamma \vdash M : T$.
- M is *a generator of a large inductive object in the context* Γ if there exists a large inductive type I such that $\Gamma \vdash M : (\Pi\vec{x} : \vec{\tau}.I)$.
- M is a *large type* in Γ if M is a powerset or a type with large inductive target in Γ. A type M which is not large is a *small type.*

- M is a *large object* in Γ if M is a subset or a generator of large inductive object. An object M which is not large is a *small objects*. The set of all small objects is denoted by $SmallObj^{\Gamma}$.
- M is a *large term* in Γ if M is a sort, a kind, a type, a formula, a type constructor, a subset or a large object in Γ. A variable $x \in dom(\Gamma)$ is *large* if it is a large term in Γ.

Figure 3.3 illustrates the above notions. Types are divided into large and small. Large types are powersets or types with large inductive targets. As a consequence, objects are also divided into large and small objects. We also have large terms: those are all terms but small objects and proofs.

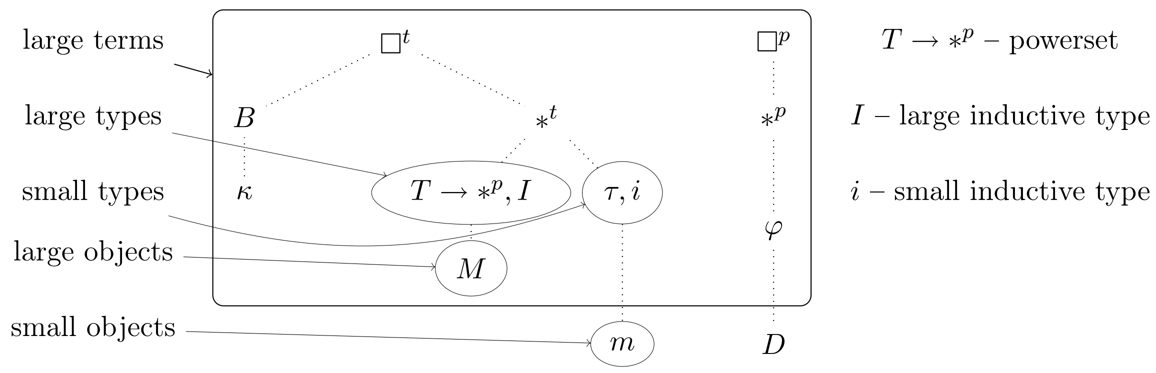

Figure 3.3: Large and small terms

Classification into large and small terms is based on the target of the type: for types, the target of the type itself, for objects, the target of its type. We also need a classification based on the argument that a term accepts. Thus we have the following definitions

- M is an *acceptor of large argument of type* τ in Γ if M is a type constructor or a subset, $\Gamma \vdash M : (\Pi x : \tau.B)$ and τ is a large type.
- M is an *acceptor of small argument of type* τ in Γ if M is a type constructor or a subset, $\Gamma \vdash M : (\Pi x : \tau.B)$ and τ is a small type.

3.7. Basic properties of the system

In this section we prove basic properties of the system.

Lemma 16 (Substitution Lemma)**.** *If* $\Gamma_1, x : A, \Gamma_2 \vdash B : C$ *and* $\Gamma_1 \vdash D : A$ *then*

$$\Gamma_1, \Gamma_2[x := D] \vdash B[x := D] : C[x := D].$$

Proof. Induction with respect to the structure of the derivation $\Gamma_1, x : A, \Gamma_2 \vdash B : C$. □

Lemma 17 (Generation Lemma)**.** *Let* $\Gamma \vdash M : T$.

- *If* $M = x$ *then* $x \in dom(\Gamma)$, $T =_{\beta\iota} \Gamma(x)$ *and* $\Gamma \vdash T : s$ *where* s *is a sort.*

- *If* $M = *^t$ *then* $T =_{\beta\iota} \Box^t$.
- *If* $M = *^p$ *then* $T =_{\beta\iota} \Box^p$.
- *If* $M = \Pi x : T_1.T_2$ *then there exists* $(s_1, s_2, s_3) \in \mathcal{R}$ *such that*

$$\Gamma \vdash T_1 : s_1, \quad \Gamma, x : T_1 \vdash T_2 : s_2, \quad T =_{\beta\iota} s_3.$$

- *If* $M = \lambda x : T_1.M_2$ *then there exists* $(s_1, s_2, s_3) \in \mathcal{R}$ *and term* T_2 *such that*

$$\Gamma \vdash T_1 : s_1, \quad \Gamma, x : T_1 \vdash M_2 : T_2, \quad \Gamma, x : T_1 \vdash T_2 : s_2, \quad \Gamma \vdash T : s_3,$$
$$\text{and } T =_{\beta\iota} \Pi x : T_1.T_2.$$

- *If* $M = M_1 M_2$ *then there exist* T_1, T_2 *such that*

$$\Gamma \vdash M_1 : (\Pi x : T_1.T_2), \quad \Gamma \vdash M_2 : T_1, \quad T =_{\beta\iota} T_2[x := M_2];$$

- *If* $M = \mathrm{Ind}(X : A)\{\vec{C}\}$ *then* $A =_{\beta\iota} T$, *every* C_i *is a type of constructor in* X, *and either there exist* $\vec{t}$ *such that*

$$A =_{\beta\iota} *^t, \quad \Gamma \vdash A : \Box^t, \text{ and for every } i \text{ we have } \Gamma, X : A \vdash C_i : *^t$$

or there exist $\vec{t}$ *and* $s \in \{*^t, \Box^p\}$ *such that*

$$A =_{\beta\iota} \Pi \vec{x} : \vec{t}.*^p, \quad \Gamma \vdash A : s, \text{ and for every } i \text{ we have } \Gamma, X : A \vdash C_i : *^p.$$

- *If* $M = \mathrm{Constr}(n, I)$ *then* $I =_{\beta\iota} \mathrm{Ind}(X : A)\{\vec{C}\}$ *and* $C_n(I) =_{\beta\iota} T$.
- *If* $M = \mathrm{Elim}(I, Q, \vec{u}, m)\{\vec{f}\}$ *then either there exist* $\vec{C}$, s *such that*

$$I =_{\beta\iota} \mathrm{Ind}(X : *^t)\{\vec{C}\}, \quad \vec{u} = \epsilon, \quad \Gamma \vdash Q : I \to s, \quad \Gamma \vdash m : I, \quad T =_{\beta\iota} Qm$$
$$\text{and for all } i \text{ we have } \Gamma \vdash f_i : \triangle\{C_i(I), Q, \mathrm{Constr}(i, I)\},$$

or there exist $\vec{C}$, s_1, A, A_1 *such that*

$$I =_{\beta\iota} \mathrm{Ind}(X : A)\{\vec{C}\}, \quad \Gamma \vdash Q : A_1, \quad A =_{\beta\iota} A_1, \quad \Gamma \vdash m : I\vec{u}, \quad T =_{\beta\iota} Q\vec{u},$$
$$\text{and for all } i \text{ we have } \Gamma \vdash f_i : \triangle\{C_i(I), Q\}.$$

Proof. Induction with respect to the structure of the derivation of $\Gamma \vdash M : T$. □

Lemma 18 (Uniqueness of types)**.** *If* $\Gamma \vdash M : T_1$ *and* $\Gamma \vdash M : T_2$ *then* $T_1 =_{\beta\iota} T_2$.

Proof. Induction with respect to the structure of the term M, using Lemma 17. □

We will now prove the Church-Rosser property for the system. We could almost use the theorem stating that for higher order term rewriting systems which are left-linear and non-overlapping have the Church-Rosser property [51]. However, the right hand side of the iota reduction rule is not a pattern in the sense of Definition 11.2.18 in [51]. It has to be computed by a simple recursive algorithm. We cannot use the theorem so we have to prove the property on our own. We use a Takahashi variant of Tait's parallel reduction method [50]. Our proof is inspired by [27].

First we introduce the notion of parallel computation. The relation $\rightrightarrows$ is defined as follows:

- For every term M we have $M \rightrightarrows M'$.
- If $M \rightrightarrows M'$ and $N \rightrightarrows N'$, then $(\lambda x : A.M)N \rightrightarrows M'[x := N']$.
- If $I \rightrightarrows I'$, $Q \rightrightarrows Q'$, $\vec{u} \rightrightarrows \vec{u'}$, $\vec{N} \rightrightarrows \vec{N'}$ and $\vec{f} \rightrightarrows \vec{f'}$ then
$$\mathrm{Elim}(I, Q, \vec{u}, \mathrm{Constr}(n, J)\vec{N})\{\vec{f}\} \rightrightarrows \triangle[C_n(I'), f'_n, \vec{N'}, I', Q', \vec{f'}].$$
- If $M \rightrightarrows M'$ and $N \rightrightarrows N'$ then $MN \rightrightarrows M'N'$.
- If $M \rightrightarrows M'$ and $N \rightrightarrows N'$ then $\Pi x : M.N \rightrightarrows \Pi x : M'.N'$.
- If $M \rightrightarrows M'$ and $N \rightrightarrows N'$ then $\lambda x : M.N \rightrightarrows \lambda x : M'.N'$.
- If $I \rightrightarrows I'$ then $\mathrm{Constr}(n, I) \rightrightarrows \mathrm{Constr}(n, I')$;
- If $A \rightrightarrows A'$ and $\vec{C} \rightrightarrows \vec{C'}$ then $\mathrm{Ind}(X : A)\{\vec{C}\} \rightrightarrows \mathrm{Ind}(X : A')\{\vec{C'}\}$.
- If $I \rightrightarrows I'$, $Q \rightrightarrows Q'$, $\vec{u} \rightrightarrows \vec{u'}$, $\vec{M} \rightrightarrows \vec{M'}$ and $\vec{f} \rightrightarrows \vec{f'}$ then
$$\mathrm{Elim}(I, Q, \vec{u})\{\overrightarrow{M}\}\vec{f} \rightrightarrows \mathrm{Elim}(I', Q', \vec{u'})\{\overrightarrow{M'}\}\vec{f'}$$

The transitive closure of the relation $\rightrightarrows$ is the same as the transitive closure of the relation $\rightarrow_{\beta\iota}$. This is a consequence of the following lemma.

Lemma 19. *1. If $M \rightarrow_{\beta\iota} N$ then $M \rightrightarrows N$.*

2. If $M \rightrightarrows N$ then $M \rightarrow^*_{\beta\iota} N$.

Proof. 1. We only show the proof in the case when M is a beta redex and N is a beta reduct. Suppose $M = (\lambda x : A.B)C$ and $N = B[x := C]$. Then we have $B \rightrightarrows B$ and $C \rightrightarrows C$ because the relation $\rightrightarrows$ is reflexive. Thus $(\lambda x : A.B)C \rightrightarrows B[x := C]$. The proof in the remaining cases is similar: we often use the fact that the relation $\rightrightarrows$ is reflexive.

2. The proof is by induction with respect to the definition of the relation $\rightrightarrows$. □

For every term M we define a term M^*. The idea is that the term M^* is a term where all redexes present in M have been contracted. We can define such term because the redexes in M are not overlapping. The definition of M^* follows.

- $x^* = x$;
- $(\lambda x : A.B)^* = \lambda x : A^*.B^*$;
- $(\Pi x : A.B)^* = \Pi x : A^*.B^*$;
- $(AB)^* = N^*[x := B^*]$, if $A = \lambda x : M.N$;
- $(AB)^* = A^*B^*$, if $A \neq \lambda x : M.N$;
- $(\mathrm{Constr}(n, I))^* = \mathrm{Constr}(n, I^*)$;
- $(\mathrm{Ind}(X : A)\{\vec{C}\})^* = \mathrm{Ind}(X : A^*)\{\vec{C^*}\}$;

- $(\text{Elim}(I,Q,\vec{u},M)\{\vec{f}\})^* = \triangle[C_n(I^*), f_n^*, \vec{N}^*, I^*, Q^*, \vec{f^*}]$ if $M = \text{Constr}(n,I')\vec{N}$;
- $(\text{Elim}(I,Q,\vec{u},M)\{\vec{f}\})^* = (\text{Elim}(I^*,Q^*,\vec{u^*},M^*)\{\vec{f^*}\})$ if $M \neq \text{Constr}(n,I')\vec{N}$.

Lemma 20. *If $M \rightrightarrows M'$ and $N \rightrightarrows N'$ then $M[x := N] \rightrightarrows M'[x := N']$;*

Proof. Induction with respect to the structure of the term M. □

Lemma 21. *If $C(X)$ is a type of constructor in X and $I \rightrightarrows I^*$, $Q \rightrightarrows Q^*$, $\vec{M} \rightrightarrows \vec{M}^*$, $\vec{u} \rightrightarrows \vec{u^*}$, $\vec{A} \rightrightarrows \vec{A}^*$ and $\vec{f} \rightrightarrows \vec{f^*}$ then*

$$\triangle[C(X), M, \vec{A}, I, Q, \vec{f}] \rightrightarrows \triangle[(C(X))^*, M^*, \vec{A}^*, I^*, Q^*, \vec{f^*}].$$

Proof. We proceed by induction with respect to the structure of $C_n(I)$.

- $C(X) = X\vec{t}$. Then $(C(X))^* = X\vec{t^*}$. Then

$$\triangle[C(X), M, \vec{A}, I, Q, \vec{f}] = M \rightrightarrows M^* = \triangle[(C(X))^*, M^*, \vec{A}^*, I^*, Q^*, \vec{f^*}].$$

- $C(X) = \Pi x : T.D(X)$ where $X \notin FV(T)$. Then $(C(X))^* = \Pi x : T^*.(D(X))^*$ and $X \notin FV(T^*)$. Note that then $\vec{A} = A_0 :: \vec{A'}$ and

$$\triangle[C(X), M, \vec{A}, I, Q, \vec{f}] = \triangle[D(X), MA_0, \vec{A'}, I, Q, \vec{f}].$$

  By assumption we have $M \rightrightarrows M^*$ and $A_0 \rightrightarrows A_0^*$ and thus $MA_0 \rightrightarrows M^*A_o^*$. By the induction hypothesis we have

$$\triangle[D(X), MA_0, \vec{A'}, I, Q, \vec{f}] \rightrightarrows \triangle[(D(X))^*, M^*A_0^*, \vec{A'^*}, I^*, Q^*, \vec{f^*}].$$

  But

$$\triangle[(D(X))^*, M^*A_0^*, \vec{A'^*}, I^*, Q^*, \vec{f^*}] = \triangle[(C(X))^*, M^*, \vec{A}^*, I^*, Q^*, \vec{f^*}].$$

- $C(X) = \Pi x : T.D(X)$ where $T = \Pi\vec{y} : \vec{\tau}.X\vec{t}$. Then $C(X)^* = \Pi x : T^*.D(X)^*$ and $T^* = \Pi\vec{y} : \vec{\tau}^*.X\vec{t^*}$. Then $\vec{A} = A_0 :: \vec{A'}$ and

$$\triangle[C(X), M, \vec{A}, I, Q, \vec{f}] = \triangle[D(X), MA_0(\lambda\vec{y} : \vec{\tau}.\text{Elim}(I,Q,\vec{t},A_0\vec{y})\{\vec{f}\}), \vec{A'}, I, Q, \vec{f}].$$

  By assumption we have $M \rightrightarrows M^*$ and $A_0 \rightrightarrows A_0^*$ and $\vec{\tau} \rightrightarrows \vec{\tau^*}$. Thus

$$(\lambda\vec{y} : \vec{\tau}.\text{Elim}(I,Q,\vec{t},A_0\vec{y})\{\vec{f}\}) \rightrightarrows (\lambda\vec{y} : \vec{\tau^*}.\text{Elim}(I^*,Q^*,\vec{t^*},A_0^*\vec{y})\{\vec{f^*}\}.$$

  and

$$MA_0(\lambda\vec{y} : \vec{\tau}.\text{Elim}(I,Q,\vec{t},A_0\vec{y})\{\vec{f}\}) \rightrightarrows M^*A_0^*(\lambda\vec{y} : \vec{\tau^*}.\text{Elim}(I^*,Q^*,\vec{t^*},A_0^*\vec{y})\{\vec{f^*}\}.$$

  By the induction hypothesis we have

$$\begin{aligned}&\triangle[D(X), MA_0(\lambda\vec{y} : \vec{\tau}.\text{Elim}(I,Q,\vec{t},A_0\vec{y})\{\vec{f}\}), \vec{A'}, I, Q, \vec{f}] \\ &\qquad \rightrightarrows \triangle[D(X)^*, M^*A_0^*(\lambda\vec{y} : \vec{\tau}.\text{Elim}(I^*,Q^*,\vec{t^*},A_0^*\vec{y})\{\vec{f^*}\}), \vec{A'^*}, I^*, Q^*, \vec{f^*}]\end{aligned}$$

  But

$$\begin{aligned}&\triangle[D(X)^*, M^*A_0^*(\lambda\vec{y} : \vec{\tau}.\text{Elim}(I^*,Q^*,\vec{t^*},A_0^*\vec{y})\{\vec{f^*}\}), \vec{A'^*}, I^*, Q^*, \vec{f^*}] \\ &\qquad = \triangle[C(X)^*, M^*, \vec{A}^*, I^*, Q^*, \vec{f^*}].\end{aligned}$$ □

Lemma 22. *For every term M we have $M \rightrightarrows M^*$.*

Proof. Induction with respect to the structure of M. □

We now prove the main lemma of this section.

Lemma 23. *If $M \rightrightarrows N$ then $N \rightrightarrows M^*$.*

Proof. We proceed by induction with respect to the definition of the relation $\rightrightarrows$.

- If $M \rightrightarrows M$ then $M \rightrightarrows M^*$ by Lemma 22.
- Suppose $M = (\lambda x : A.B)C$ and $N = B'[x := C']$ where $B \rightrightarrows B'$ and $C \rightrightarrows C'$. By the induction hypothesis we have $B' \rightrightarrows B^*$ and $C \rightrightarrows C^*$. By Lemma 20 we have $N = B'[x := C'] \rightrightarrows B^*[x := C^*] = M^*$.
- Suppose $M = \mathrm{Elim}(I, Q, \vec{u}, \mathrm{Constr}(n, J)\vec{A})\{\vec{f}\}$ and $N = \triangle[C_n(I'), f'_n, \vec{A'}, I', Q', \vec{f'}]$ where

$$I \rightrightarrows I', \quad Q \rightrightarrows Q', \quad \vec{u} \rightrightarrows \vec{u'}, \quad \vec{A} \rightrightarrows \vec{A'} \quad \text{and} \quad \vec{f} \rightrightarrows \vec{f'}.$$

  By the induction hypothesis we have

$$I' \rightrightarrows I^*, \quad Q' \rightrightarrows Q^*, \quad \vec{u'} \rightrightarrows \vec{u^*}, \quad \vec{A'} \rightrightarrows \vec{A^*} \quad \text{and} \quad \vec{f'} \rightrightarrows \vec{f^*}.$$

  By Lemma 21 we have

$$N = \triangle[C_n(I'), f'_n, \vec{A'}, I', Q', \vec{f'}] \rightrightarrows \triangle[C_n(I^*), f^*_n, \vec{A^*}, I^*, Q^*, \vec{f^*}] = M^*.$$

- Suppose $M = AB$ and $N = A'B'$ and $A \rightrightarrows A'$ and $B \rightrightarrows B'$. By the induction hypothesis $A' \rightrightarrows A^*$ and $B' \rightrightarrows B^*$. Then $N = A'B' \rightrightarrows A^*B^*$. If $A \neq (\lambda x : D.E)$ then $M^* = A^*B^*$ and we are done. If $A = (\lambda x : D.E)$ then $A^* = (\lambda x : D^*.E^*)$ and $M^* = E^*[x := B^*]$. We have $A' \rightrightarrows (\lambda x : D^*.E^*)$ and $B' \rightrightarrows B^*$ and thus by Lemma 20 we have $N \rightrightarrows E^*[x := B^*] = M^*$.
- Suppose $M = \mathrm{Elim}(I, Q, \vec{u}, A)\{\vec{f}\}$ and $N = \mathrm{Elim}(I', Q', \vec{u'}, A')\{\vec{f'}\}$ where

$$I \rightrightarrows I', \quad Q \rightrightarrows Q', \quad \vec{u} \rightrightarrows \vec{u'}, \quad A \rightrightarrows A' \quad \text{and} \quad \vec{f} \rightrightarrows \vec{f'}.$$

  By the induction hypothesis we have

$$I' \rightrightarrows I^*, \quad Q' \rightrightarrows Q^*, \quad \vec{u'} \rightrightarrows \vec{u^*}, \quad A' \rightrightarrows A^* \quad \text{and} \quad \vec{f'} \rightrightarrows \vec{f^*}.$$

  By Lemma 21 we have $N \rightrightarrows \mathrm{Elim}(I^*, Q^*, \vec{u}^*, A^*)\{\vec{f^*}\}$. If $A \neq \mathrm{Constr}(n, J)\vec{B}$ then $M^* = \mathrm{Elim}(I^*, Q^*, \vec{u}^*, A^*)\{\vec{f^*}\}$ and we are done. If $A = \mathrm{Constr}(n, J)\vec{B}$ then $M^* = \triangle[C_n(I^*), f^*_n, \vec{B^*}, I^*, Q^*, \vec{f^*}]$. The conclusion follows from Lemma 21.
- Suppose $M = \Pi x : A.B$ and $N = \Pi x : A'.B'$ and $A \rightrightarrows A'$ and $B \rightrightarrows B'$. By the induction hypothesis we have $A' \rightrightarrows A^*$ and $B' \rightrightarrows B^*$. Then $N = \Pi x : A'.B' \rightrightarrows \Pi x : A^*.B^* = M$.
- The remaining cases are similar.

□

Lemma 24. *If $M \to_{\beta\iota} N_1$ and $M \to_{\beta\iota} N_2$ then there exists a term P such that $N_1 \to^*_{\beta\iota} P$ and $N_2 \to^*_{\beta\iota} P$.*

Proof. Assume that $M \to_{\beta\iota} N_1$ and $M \to_{\beta\iota} N_2$. By Lemma 19 we have $M \rightrightarrows N_1$ and $M \rightrightarrows N_2$. By Lemma 23, $N_1 \rightrightarrows M^*$ and $N_2 \rightrightarrows M^*$. Using again Lemma 19 we get $N_1 \to^*_{\beta\iota} M^*$ and $N_2 \to^*_{\beta\iota} M^*$. □

Theorem 25 (Church-Rosser Property)**.** *If $M =_{\beta\iota} N$ then there exists a term P such that $M \to^*_{\beta\iota} P$ and $N \to^*_{\beta\iota} P$.*

Proof. Induction with respect to the definition of $=_{\beta\iota}$. □

Lemma 26. *If $(\Pi x : A.B) =_{\beta\iota} (\Pi x : A'.B')$ then $A =_{\beta\iota} A'$ and $B =_{\beta\iota} B'$.*

Proof. By Theorem 25 there exists a term P such that

$$(\Pi x : A.B) \to_{\beta\iota} P \quad \text{ and } (\Pi x : A'.B') \to_{\beta\iota} P.$$

Neither the beta reduction nor the iota reduction may destroy the product. Thus we have $P = \Pi x : A''.B''$, and $A \to_{\beta\iota} A''$, and $A' \to_{\beta\iota} A''$, $B \to_{\beta\iota} B''$, and $B' \to_{\beta\iota} B''$. Then indeed $A =_{\beta\iota} A'$ and $B =_{\beta\iota} B'$. □

Theorem 27 (Subject Reduction)**.** *If $\Gamma \vdash M : T$ and $M \to_{\beta\iota} N$ then $\Gamma \vdash N : T$.*

Proof. Induction with respect to the structure of the derivation of $\Gamma \vdash M : T$, using Theorem 25 and Lemma 26.

□

Chapter 4

A translation for non-proofs

In this chapter we prove that the non-proof terms are strongly normalizing. This result will be needed in the next section for the full proof of strong normalization. We prove it using a translation to the Calculus of Inductive Constructions.

4.1. Calculus of Inductive Constructions

By Calculus of Inductive Construction (CIC) [55] we mean here an extension of the Calculus of Constructions (CC) [16] with inductive types. The Calculus of Constructions is the most powerful system in Barendregt cube. Recall that CC is a Pure Type System where

$$\begin{aligned}
&\mathcal{S} = \{*, \Box\}, \\
&\mathcal{A} = \{* : \Box\}, \\
&\mathcal{R} = \{(*, *, *), (*, \Box, \Box), (\Box, *, *), (\Box, \Box, \Box)\}.
\end{aligned}$$

We extend the syntax with the following constructions.

$$T := \mathrm{Ind}(x : T)\{\vec{T}\} \mid \mathrm{Constr}(n, T) \mid \mathrm{Elim}(T, T, \vec{T}, T)\{\vec{T}\}.$$

The meaning of constructions is similar as in LNTT with inductive types. A term $\mathrm{Ind}(x : A)\{\vec{C}\}$ is an inductive type, the expression A is a type of the bound variable x and $\vec{C}$ is a sequence of types of the inductive constructors. A term $\mathrm{Constr}(n, I)$ is a constructor of an inductive object, I is an inductive type and n is a natural number indicating which inductive constructor is meant. Finally, $\mathrm{Elim}(I, Q, \vec{u}, m)\{\vec{f}\}$ is an eliminator of an inductive type I. The expression I is an inductive type, m is the term being eliminated, $\vec{u}$ is the vector of inductive type parameters, Q is such that $Q\vec{u}m$ is the type of the result, and $\vec{f}$ is the vector of definitions corresponding to the inductive constructors of I (i.e. cases possible for m).

We also add typing rules. For technical reasons, we extend CC with a sort $\triangle$, an axiom rule

$$(Ax2) \vdash \Box : \triangle,$$

product rules

$$(\mathrm{Prod1})\ \frac{\Gamma \vdash A : * \qquad \Gamma, x : A \vdash B : \triangle}{\Gamma \vdash (\Pi x : A.B) : \triangle} \qquad (\mathrm{Prod2})\ \frac{\Gamma \vdash A : \Box \qquad \Gamma, x : A \vdash B : \triangle}{\Gamma \vdash (\Pi x : A.B) : \triangle}$$

and the corresponding (Abs) rules. However, do not add (Var) or (Weak) rules for terms of the sort $\triangle$. Thus there are no kind variables (i.e. such that $\Gamma \vdash x : \Box$) in CIC.

If $A = \Pi\vec{x} : \vec{T}.*$ and every $C_i(X)$ is a type of constructor in X then we have the rule

$$(Ind)\ \frac{\Gamma \vdash A : \Box \qquad \Gamma, X : A \vdash C_i(X) : *}{\Gamma \vdash \mathrm{Ind}(X : A)\{\vec{C}(X)\} : A}$$

In the rule $(Intro)$ the term I denotes $\mathrm{Ind}(X : A)\{\vec{C}(X)\}$.

$$(Intro)\ \frac{\Gamma \vdash I : T}{\Gamma \vdash \mathrm{Constr}(n, I) : C_n(I)}$$

In the rule $(Elim)$ we will use the notation $\Gamma \vdash \vec{u} : (\vec{x} : \vec{T})$. Here $\vec{u}$, $\vec{T}$ and $\vec{x}$ are respectively two sequences of terms and one sequence of variables of the same length. The typing rules for judgements of this form are as follows:

$$\text{(Nil)}\ \Gamma \vdash \epsilon : (\epsilon : \epsilon)$$

$$\text{(Cons)}\ \frac{\Gamma \vdash \vec{u} : (\vec{x} : \vec{T}) \qquad \Gamma, \vec{x} : \vec{T}[\vec{x} := \vec{u}] \vdash u : T[\vec{x} := \vec{u}]}{\Gamma \vdash u :: \vec{u} : (x :: \vec{x} : T :: \vec{T})}$$

We give the typing rules for elimination. In this rule the term I denotes $\mathrm{Ind}(X : A)\{\vec{C}(X)\}$, $A = \Pi\vec{x} : \vec{T}.*$ and the operator $\triangle\{C(X), Q, c\}$ is defined as on page 26.

$$\text{(Elim-W)}\ \frac{\begin{array}{c}\Gamma \vdash \vec{u} : (\vec{x} : \vec{T}) \qquad \Gamma \vdash t : I\vec{u} \qquad \Gamma \vdash Q : \Pi\vec{x} : \vec{T}.I\vec{x} \to * \\ \Gamma \vdash f_n : \triangle\{C_n(I), Q, \mathrm{Constr}(n, I)\}\end{array}}{\Gamma \vdash \mathrm{Elim}(I, Q, \vec{u}, t)\{\vec{f}\} : Q\vec{u}t}$$

As already mentioned, in CIC we have strong elimination rule. The type of constructor $C(X) = \Pi\vec{x} : \vec{\tau}.X\vec{x'}$ is *small in* Γ if every τ_i is a type, i.e. $\Gamma \vdash \tau_i : *$. An inductive type $I = \mathrm{Ind}(X : A)\{\vec{C}(X)\}$ is *small* if every type of constructor $C_i(X)$ is small. Strong elimination is only allowed for small inductive types. If the type I is small then we have strong elimination rule.

$$\text{(Elim-S)}\ \frac{\begin{array}{c}\Gamma \vdash \vec{u} : (\vec{x} : \vec{T}) \qquad \Gamma \vdash t : I\vec{u} \qquad \Gamma \vdash Q : \Pi\vec{x} : \vec{T}.I\vec{x} \to \Box \\ \Gamma \vdash f_n : \triangle\{C_n(I), Q, \mathrm{Constr}(n, I)\}\end{array}}{\Gamma \vdash \mathrm{Elim}(I, Q, \vec{u}, t)\{\vec{f}\} : Q\vec{u}t}$$

4.2. The definition of the translation

For a context Γ and a term M typable in Γ which is not a proof we define a term $T_\Gamma(M)$ typable in the Calculus of Inductive Constructions. In the translation we use the following variables

- $\mathrm{Bool} : *$,
- $\mathrm{Impl} : Bool \to Bool \to Bool$,

- Forall : $\Pi x : *.(x \to Bool) \to Bool$,
- Forall2 : $(Bool \to Bool) \to Bool$.
- IND : $\Pi x : *.((x \to Bool) \to x)$,

Moreover, for every natural number n we consider a variable

$$\text{IND}_n : \underbrace{Bool \to \ldots \to Bool}_{n \text{ times}} \to Bool.$$

Note that the types of the variables are correct types in the Calculus of Inductive Constructions in the context $Bool : *$. The impredicativity of the sort $*$ is essential to type the variables Forall and IND.

We define the translation $T_\Gamma(M)$ by induction with respect to the structure of M as follows:

- $T_\Gamma(\Box^p) = *$,
- $T_\Gamma(*^p) = Bool$,
- $T_\Gamma(\Box^t) = \Box$,
- $T_\Gamma(*^t) = *$,
- $T_\Gamma(\Pi x : \varphi.\psi) = \text{Impl } T_\Gamma(\varphi)\ T_{\Gamma,x:\varphi}(\psi)$, if φ, ψ are formulas,
- $T_\Gamma(\Pi x : \tau.\varphi) = \text{Forall } T_\Gamma(\tau)\ (\lambda x : T_\Gamma(\tau).T_{\Gamma,x:\tau}(\varphi))$, if τ is a type and φ is a formula,
- $T_\Gamma(\Pi x : *^p.\varphi) = \text{Forall2 } (\lambda x : Bool.T_{\Gamma,x:*^p}(\varphi))$, if φ is a formula,
- $T_\Gamma(\Pi x : A.B) = \Pi x : T_\Gamma(A).T_{\Gamma,x:A}(B)$ in all other cases,
- $T_\Gamma(\lambda x : A.B) = \lambda x : T_\Gamma(A).T_{\Gamma,x:A}(B)$,
- $T_\Gamma(AB) = T_\Gamma(A)T_\Gamma(B)$,
- $T_\Gamma(x) = x$,
- $T_\Gamma(\text{Ind}(X : *^t)\{\vec{C}\}) = \text{Ind}(X : *)\{\vec{T}_{\Gamma,X:*^t}(C\})$,
- $T_\Gamma(\text{Ind}(X : A)\{\vec{C}\}) = \text{IND } T_\Gamma(A)\ (\lambda X : T_\Gamma(A).\text{IND}_n \cdot \vec{T}_{\Gamma,X:A}(C))$, if $\vec{C}$ has length n and $A = \Pi \vec{x} : \vec{\tau}.*^p$; (recall the notation introduced on page 23: if $g = \vec{N}$ is a sequence of terms then a term of the form $M\vec{N}$ can be denoted by $M \cdot g$),
- $T_\Gamma(\text{Constr}(n, I)) = \text{Constr}(n, T_\Gamma(I))$,
- $T_\Gamma(\text{Elim}(I, Q, M)\{\vec{f}\}) = \text{Elim}(T_\Gamma(I), T_\Gamma(Q), \epsilon, T_\Gamma(M))\{T_\Gamma(\vec{f})\}$.

For a context Γ we define $T^n(\Gamma)$:

- $T^n(\epsilon) = \{\text{Bool} : *, \text{Impl} : Bool \to Bool \to Bool,$
  $\text{Forall} : \Pi x : *.(x \to Bool) \to Bool,$
  $\text{Forall2} : (Bool \to Bool) \to Bool,$
  $\text{IND} : \Pi x : *.((x \to Bool) \to x),$
  $\text{IND}_0 : Bool, \text{IND}_1 : Bool \to Bool, \ldots, \text{IND}_n : \overrightarrow{Bool} \to Bool, \}$

- $T^n(\Gamma, x : A) = T^n(\Gamma), x : T^n_\Gamma(A)$, if A is not a formula,
- $T^n(\Gamma, x : A) = T^n(\Gamma)$, if A is a formula.

We will prove that the translation is correct that is if $\Gamma \vdash M : A$ in LNTT with inductive types then there exists n such that $T^n(\Gamma) \vdash T_\Gamma(M) : T_\Gamma(A)$ in the Calculus of Inductive Constructions. We will first state some auxiliary lemmas.

Lemma 28. *Suppose Γ is a context and M, N are non-proofs such that*

$$\Gamma, x : \tau_2 \vdash M : \tau_1, \quad \Gamma \vdash N : \tau_2.$$

Then

$$T_\Gamma(M[x := N]) = T_{\Gamma,x:\tau_2}(M)[x := T_\Gamma(N)].$$

Proof. Note that $\Gamma \vdash M[x := N] : \tau_1[x := N]$ by Lemma 16. The proof is by routine induction with respect to the structure of M. □

Lemma 29. *Suppose Γ is a context and M, N are two non-proofs in Γ. If $M \to_{\beta\iota} N$ then $T_\Gamma(M) \to^+_{\beta\iota} T_\Gamma(N)$.*

Proof. We proceed by induction with respect to the definition of $M \to_{\beta\iota} N$.

Suppose M is a beta redex and N is its reduct, namely

$$M = (\lambda x : A.B)C \quad \text{and} \quad N = B[x := C].$$

Then

$$T_\Gamma(M) = T_\Gamma((\lambda x : A.B)C) = (\lambda x : T_\Gamma(A).T_{\Gamma,x:A}(B))\ T_\Gamma(C)$$

and by Lemma 28

$$T_\Gamma(N) = T_{\Gamma,x:A}(B)[x := T_\Gamma(C)].$$

Thus $T_\Gamma(M) \to_{\beta\iota} T_\Gamma(N)$.

If M is a iota redex and N is its reduct then $M = \mathrm{Elim}(I, Q, \mathrm{Constr}(n, I')\vec{m})\{\vec{f}\}$ and $N = \triangle[C_n(I), f_n, \vec{m}, I, Q, \vec{f}]$. Note that if M is not a proof then I is an inductive type (and not predicate). Thus the vector of parameters is empty. Then

$$T_\Gamma(M) = \mathrm{Elim}(T_\Gamma(I), T_\Gamma(Q), \epsilon, \mathrm{Constr}(n, T_\Gamma(I))T_\Gamma(\vec{m}))\{T_\Gamma(\vec{f})\}$$

so $T_\Gamma(M)$ is still a redex. Moreover,

$$T_\Gamma(N) = T_\Gamma(\triangle[C_n(I), f_n, \vec{m}, I, Q, \vec{f}]).$$

We will prove that

$$T_\Gamma(\triangle[C, f, \vec{m}, I, Q, \vec{f}]) = \triangle[T_\Gamma(C, T_\Gamma(f), T_\Gamma(\vec{m}), T_\Gamma(I), T_\Gamma(Q), T_\Gamma(\vec{f})].$$

The proof is by induction with respect to the structure of C.

- If $C = I$ then $T_\Gamma(C) = T_\Gamma(I)$. We have

  $$\triangle[C, f, \vec{m}, I, Q, \vec{f}] = f,$$

  and

  $$T_\Gamma(\triangle[C, f, \vec{m}, I, Q, \vec{f}]) = T_\Gamma(f) = \triangle[T_\Gamma(C), T_\Gamma(f), T_\Gamma(\vec{m}), T_\Gamma(I), T_\Gamma(Q), T_\Gamma(\vec{f})].$$

- If $C = \Pi x : t.D$ then D is not a formula and thus $T_\Gamma(C) = \Pi x : T_\Gamma(t).T_\Gamma(D)$ and

$$\triangle[C, f, m_0 :: \vec{m}, I, Q, \vec{f}] = \triangle[C, fm_0, \vec{m}, I, Q, \vec{f}].$$

  Thus

$$\begin{aligned} T(\triangle[C, f, m_0 :: \vec{m}, I, Q, \vec{f}] &= T_\Gamma(\triangle[C, fm_0, \vec{m}, I, Q, \vec{f}]) \\ &= \triangle[T_\Gamma(C), T_\Gamma(f)T_\Gamma(m_0), T_\Gamma(\vec{m}), T_\Gamma(I), T_\Gamma(Q), T_\Gamma(\vec{f})]. \end{aligned}$$

- If $C = (\Pi\vec{x} : \vec{t}.I) \to D$ then D is not a formula and thus

$$T_\Gamma(C) = (\Pi\vec{x} : T_\Gamma(\vec{t}).T_\Gamma(I)) \to T_\Gamma(D).$$

  Moreover

$$\triangle[C, f, m_0 :: \vec{m}, I, Q, \vec{f}] = \triangle[D, (fm_0(\lambda\vec{x} : \vec{t}.\text{Elim}(I, Q, m_0\vec{x})\{\vec{f}\})), \vec{m}, I, Q, \vec{f}].$$

  and

$$\begin{aligned} &T(\Delta[C, f, m_0 :: \vec{m}, I, Q, \vec{f}]) = T_\Gamma(\triangle[D, (fm_0(\lambda\vec{x} : \vec{t}.\text{Elim}(I, Q, m_0\vec{x})\{\vec{f}\})), I, Q, \vec{f}]) \\ &\quad = \triangle[T_\Gamma(D), (T_\Gamma(f)T_\Gamma(m_0)(\lambda\vec{x} : T_\Gamma(\vec{t}).T_\Gamma(\text{Elim}(I, Q, m_0\vec{x})\{\vec{f}\}))), T_\Gamma(I), T_\Gamma(Q), T_\Gamma(\vec{f})] \\ &\quad = \triangle[T_\Gamma(C), T_\Gamma(f), T_\Gamma(I), T_\Gamma(Q), T_\Gamma(\vec{f})]. \end{aligned}$$

If $M \to_{\beta\iota} N$ by context closure then the proof is immediate. As an example we consider the case when $M = (\Pi x : A.B)$ and $N = (\Pi x : A.B')$, where A is a type and B is a formula. By Lemma 27, the term A' is a type and B' is a formula. Then

$$T_\Gamma(M) = \text{Forall } T_\Gamma(A)\ (\lambda x : T_\Gamma(A).T_{\Gamma,x:A}(B))$$

and

$$T_\Gamma(N) = \text{Forall } T_\Gamma(A')\ (\lambda x : T_\Gamma(A').T_{\Gamma,x:A'}(B')).$$

By the induction hypothesis $T_\Gamma(M) \to^+_{\beta\iota} T_\Gamma(N)$.

□

Lemma 30. *Suppose $\Gamma \vdash M : A$. Suppose all inductive types and predicates occurring in the derivation of $\Gamma \vdash M : A$ have at most n constructors. Then $T^n(\Gamma) \vdash T_\Gamma(M) : T_\Gamma(A)$.*

Proof. We proceed by induction with respect to the derivation of $\Gamma \vdash M : A$. We consider the last rule used in the derivation.

- (Ax) We have either

$$\vdash *^t : \square^t \qquad \text{or} \qquad \vdash *^p : \square^p$$

  In both cases the conclusion is trivial. We get

$$T^0(\epsilon) \vdash * : \square \qquad \text{and} \qquad T^0(\epsilon) \vdash \mathit{Bool} : *$$

  respectively.

- (Var) We have

$$\frac{\Gamma \vdash A : s}{\Gamma, x : A \vdash x : A}$$

If $s = *^t$, or $s = \Box^t$, or $s = \Box^p$ the conclusion is trivial. If $s = *^p$ then x is a proof and the translation for x is undefined.

- (Weak) We have

$$\frac{\Gamma \vdash B : C \qquad \Gamma \vdash A : s}{\Gamma, x : A \vdash B : C}$$

If s is one of $*^t$, $\Box^t$, $\Box^p$ the conclusion follows easily from induction hypothesis. If s is $*^p$, we have $T^n(\Gamma, x : A) = T^n(\Gamma)$ the conclusion is equivalent to the induction hypothesis $T^n(\Gamma) \vdash T_\Gamma(B) : T_\Gamma(C)$.

- (Conv) We have

$$\frac{\Gamma \vdash M : B \qquad \Gamma \vdash C : s \qquad B =_{\beta\iota} C}{\Gamma \vdash M : C}$$

The conclusion is a consequence of Lemma 29.

- (Prod) We have

$$\frac{\Gamma \vdash A : s_1 \qquad \Gamma, x : A \vdash B : s_2}{\Gamma \vdash (\Pi x : A.B) : s_3}$$

We deal here with the rules: $(*^t, *^t, *^t)$, $(*^p, *^p, *^p)$, $(*^t, *^p, *^p)$, $(*^t, \Box^p, *^t)$, $(*^t, \Box^t, \Box^t)$, $(\Box^p, *^p, *^p)$. We consider separate cases:

  - The product was created using the rule $(*^p, *^p, *^p)$. By the induction hypothesis we get
  $$T^n(\Gamma) \vdash T_\Gamma(A) : \mathit{Bool} \text{ and } T^n(\Gamma) \vdash T_\Gamma(B) : \mathit{Bool}.$$
  We know that
  $$T^n(\Pi x : A.B) = \text{Impl } T_\Gamma(A)\ T_{\Gamma,x:A}(B).$$
  Since Impl : $\mathit{Bool} \to \mathit{Bool} \to \mathit{Bool}$, we get
  $$T^n(\Gamma) \vdash \text{Impl } T_\Gamma(A)\ T_{\Gamma,x:A}(B) : \mathit{Bool}$$
  as it ought to be.
  - The product was created using the rule $(*^t, *^p, *^p)$. By the induction hypothesis we get
  $$T^n(\Gamma) \vdash T_\Gamma(A) : * \text{ and } T^n(\Gamma), x : T_\Gamma(A) \vdash T_{\Gamma,x:A}(B) : \mathit{Bool}.$$
  We know that
  $$T_\Gamma(\Pi x : A.B) = \text{Forall } T_\Gamma(A)\ (\lambda x : T_\Gamma(A).T_{\Gamma,x:A}(B)).$$
  The constant Forall has the type $\Pi x : *.(x \to \mathit{Bool}) \to \mathit{Bool}$ so the whole expression has type Bool.

– The product was created using the rule $(\Box^p, *^p, *^p)$. By induction hypothesis we have

$$T^n(\Gamma) \vdash T_\Gamma(A) : * \text{ and } T^n(\Gamma), x : T_\Gamma(A) \vdash T_{\Gamma,x:A}(B) : \mathit{Bool}.$$

Moreover, we know by the Generation Lemma and Theorem 25 that in that case $A = *^p$ and $T_\Gamma(A) = \mathit{Bool}$. We know that

$$T_\Gamma(\Pi x : A.B) = \text{Forall2 } (\lambda x : \mathit{Bool}.T_{\Gamma,x:A}(B))$$

and Forall2 : $(\mathit{Bool} \to \mathit{Bool}) \to \mathit{Bool}$. Thus the application indeed is of type *Bool* as it ought to be.

– The product was created using one of the rules $(*^t, *^t, *^t)$, $(*^t, \Box^p, *^t)$, $(*^t, \Box^t, \Box^t)$. By induction hypothesis we have

$$T^n(\Gamma) \vdash T_\Gamma(A) : T_\Gamma(s_1) \text{ and } T_\Gamma(s_1) = T_\Gamma(*^t) = *.$$

Moreover

$$T^n(\Gamma), x : T_\Gamma(A) \vdash T_{\Gamma,x:A}(B) : T_{\Gamma,x:A}(s_2).$$

Here, $T_{\Gamma,x:A}(s_2) = *$ or $T_{\Gamma,x:A}(s_2) = \Box$. In any case, $T_{\Gamma,x:A}(s_2)$ is a sort. We may apply the rule $(*, *, *)$ or $(*, \Box, \Box)$ and get the desired conclusion.

- (App) Routine application of the inductive hypothesis.
- (Abs) Routine application of the inductive hypothesis.
- (Ind_{*^t}). We have

$$(\mathit{Ind}_{*^t})\ \frac{\Gamma, X : *^t \vdash C_i : *^t}{\Gamma \vdash \text{Ind}(X : *^t)\{\vec{C}\}}$$

Note that $T_\Gamma(*^t)$ is $*$ and by the induction hypothesis

$$T^n(\Gamma) \vdash T_\Gamma(*^t) : \Box$$

and for all i

$$T^n(\Gamma), X : T_\Gamma(*^t) \vdash T_{\Gamma,X:*^t}(C_i) : *.$$

Moreover, every $T_{\Gamma,X:*^t}(C_i)$ is a type of constructor in X. We may thus apply the rule (Ind) and get the conclusion.

- (Ind_{*^p}). We have

$$(\mathit{Ind}_{*^p})\ \frac{\Gamma \vdash A : s \qquad \Gamma, X : A \vdash C_i : *^p}{\Gamma \vdash \text{Ind}(X : A)\{\vec{C}\} : A}$$

and $s \in \{\Box^p, *^t\}$. Here,

$$T_\Gamma(\text{Ind}(X : A)\{\vec{C}\}) = \textsc{ind}\ T_\Gamma(A)\ (\lambda X : T_\Gamma(A).\textsc{ind}_n \cdot T_\Gamma(\vec{C})).$$

The constant IND has type $(\Pi x : *.(x \to Bool) \to x)$. The constant IND_n has type $\underbrace{Bool \to \ldots \to Bool}_{n \text{ times}} \to Bool$. By the induction hypothesis for the premises of the rule we have

$$T^n(\Gamma) \vdash T_\Gamma(A) : * \quad \text{and} \quad T^n(\Gamma), X : T_\Gamma(A) \vdash T_\Gamma(C_i(X)) : Bool.$$

It follows that the application IND $T_\Gamma(A)$ $(\lambda X : T_\Gamma(A).\text{IND}_n T_\Gamma(\vec{C}(X)))$ is correct and the type of it indeed is $T_\Gamma(A)$.

- $(Intro_{*^t})$ We only deal with inductive types. In the other case, the term constructed is a proof. We have $I = \text{Ind}(X : *^t)\{\vec{C}\}$.

$$(Intro)\ \frac{\Gamma \vdash I : *^t}{\Gamma \vdash \text{Constr}(i, I) : C_i(I)}$$

By the induction hypothesis we have

$$T^n(\Gamma) \vdash T_\Gamma(\text{Ind}(X : *^t)\{\vec{C}\}) : *.$$

Moreover, $T_\Gamma(\text{Ind}(X : *^t)\{\vec{C})\}) = \text{Ind}(X : *)\{\vec{T_\Gamma}(C\})$ so indeed we may apply the rule (Intro). Now,

$$T_\Gamma(\text{Constr}(i, I)) = \text{Constr}(i, T_\Gamma(I))$$

and

$$T_\Gamma(C_i(I)) = T_\Gamma(C_i)(T_\Gamma(I))$$

and we get the conclusion.

- $(Elim_{*^t})$ We only deal with the following case

$$\frac{\Gamma \vdash t : I \qquad \Gamma \vdash Q : I \to *^t \qquad \Gamma \vdash f_i : \triangle\{C_i(I), Q, \text{Constr}(i, I)\}}{\Gamma \vdash \text{Elim}(I, Q, t)\{\vec{f}\} : (Qt)}$$

By induction hypothesis we have

$$\begin{aligned} &T^n(\Gamma) \vdash T_\Gamma(t) : T_\Gamma(I), \\ &T^n(\Gamma) \vdash T_\Gamma(Q) : T_\Gamma(I) \to *, \\ &T^n_\Gamma(\Gamma) \vdash T_\Gamma(f_i) : T_\Gamma(\triangle\{C_i(I), Q, \text{Constr}(i, I)\}). \end{aligned}$$

By induction with respect to the structure of $C_i(I)$ it is easy to prove that

$$T_\Gamma(\triangle\{C_i(I), Q, \text{Constr}(i, I)\}) = \triangle\{T_\Gamma(C_i(I)), T_\Gamma(Q), \text{Constr}(i, T_\Gamma(I))\}.$$

From this we get the conclusion. □

Lemma 31. *If* $\Gamma \vdash M : A$ *and* M *is not a proof then* M *is strongly normalizing.*

Proof. If M is not a proof then by Lemma 30 we have $T_\Gamma(\Gamma) \vdash T_\Gamma(M) : T_\Gamma(A)$. If there is an infinite reduction beginning in M:

$$M \to_{\beta\iota} M_1 \to_{\beta\iota} M_2 \to_{\beta\iota} \ldots$$

then there is an infinite reduction in CIC

$$T_\Gamma(M) \to^+_{\beta\iota} T_\Gamma(M_1) \to^+_{\beta\iota} T_\Gamma(M_2) \to^+_{\beta\iota} \ldots$$

The latter is not possible as $T_\Gamma(M)$ is strongly normalizing. □

Chapter 5

Strong normalization

The proof uses a variant of Girard's candidates of reducibility. There are two main differences in comparison with Girard's proof. First, we use saturated sets instead of candidates. Second, we use a typed version of saturated sets, a technique first introduced by J. Gallier and T. Coquand in [13]. In the commonly used untyped version, one deals with sets of terms. In the typed version, we deal with sets of pairs of the form $(\Gamma \vdash M)$ such that Γ is a context, M is a term and for some T the assertion $\Gamma \vdash M : T$ is valid.

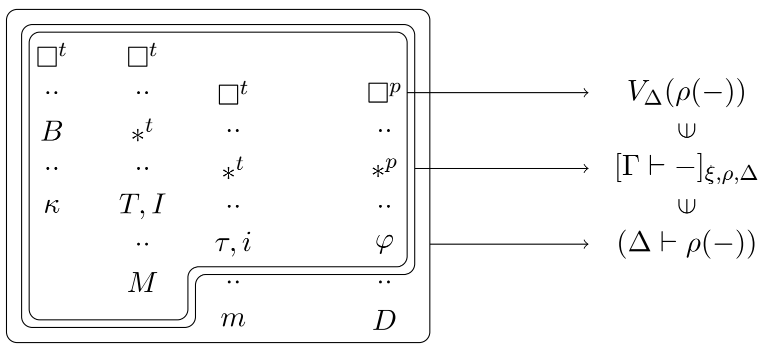

Figure 5.1: Operators in the proof

There are two basic notions in the proof. One is the set $V_\Delta(M)$ of possible interpretations of a term M. The other is the interpretation of $(\Gamma \vdash M)$. The interpretation operator $[\Gamma \vdash M]_{\xi,\rho,\Delta}$ takes five arguments: a context Γ, a term M which is being interpreted, a set valuation ξ and term substitution ρ (precise definitions are given later), and a context Δ. As usual, we give interpretations for type-like terms, i.e. types, formulas, kinds and sorts and for terms which may be eliminated so that one obtains type-like terms, namely type constructors, subsets and (via the elimination operator) large inductive objects. Recall that we refer to those terms as *large* terms. The main lemmas state in particular that for all (appropriate) Γ, Δ, M, T, ξ and ρ

$$\text{if } \Gamma \vdash M : T \text{ then } [\Gamma \vdash M]_{\xi,\rho,\Delta} \in V_\Delta(\rho(M)),$$

and

$$\text{if } \Gamma \vdash M : T \text{ then } (\Delta \vdash \rho(M)) \in [\Gamma \vdash T]_{\xi,\rho,\Delta}.$$

The relations between the operators are shown in Figure 5.1.

5.1. Saturated sets

A *simple sequent* is any pair $(\Delta \vdash M)$ where Δ is a context, M is a term and there exists a term σ such that $\Delta \vdash M : \sigma$. We say that such a pair is a simple sequent *of type* σ. For simplicity we will sometimes use the name *term* to denote a simple sequent.
Let T_σ^Δ be the set of simple sequents of type σ in the context Δ:

$$T_\sigma^\Delta = \{(\Delta' \vdash M) \mid \Delta \subseteq \Delta' \wedge \Delta' \vdash M : \sigma\}.$$

Let SN_σ^Δ be the subset of T_σ^Δ consisting of strongly normalizing terms:

$$SN_\sigma^\Delta = \{(\Delta' \vdash M) \mid \Delta \subseteq \Delta' \wedge \Delta' \vdash M : \sigma \wedge M \in SN\}.$$

We write $(\Delta' \vdash M) \in SN$ if M is strongly normalizing and there exists σ such that $\Delta' \vdash M : \sigma$.
The family of *base terms* is defined by induction:

- every variable x is a base term;
- if M is a base term and $N \in SN$ then MN is a base term;
- if M is a base term and $I, Q, \vec{u}, \vec{f} \in SN$ then $\mathrm{Elim}(I, Q, \vec{u}, M)\{\vec{f}\}$ is a base term.

Let Δ be a context and σ be a type or a kind, or a formula in the context Δ. The family of *base sequents* $B_\sigma^\Delta \subseteq T_\sigma^\Delta$ is the family of simple sequents $(\Delta' \vdash M) \in T_\sigma^\Delta$ where M is a base term.

We define the *key reduction* $\to_k$ by induction:

- $(\lambda x : A.B)C \to_k B[x := C]$;
- $\mathrm{Elim}(I, Q, \vec{u}, \mathrm{Constr}(n, I')\vec{N})\{\vec{f}\} \to_k \triangle[C_n(I), f_n, \vec{N}, I, Q, \vec{f}]$;
- if $M \to_k M'$ then $MN \to_k M'N$;
- if $M \to_k M'$ then $\mathrm{Elim}(I, Q, \vec{u}, M)\{\vec{f}\} \to_k \mathrm{Elim}(I, Q, \vec{u}, M')\{\vec{f}\}$.

The beta-reduction equivalent of the key reduction is sometimes referred to as weak head reduction. A reduction which is not key reduction will be called an *internal reduction*. We will denote it by $\to_i$.
A set of simple sequents $U \subseteq T_\sigma^\Delta$ is *saturated* (denoted by $U \in SAT_\sigma^\Delta$) if it satisfies the following conditions:

(SAT1) $U \subseteq SN_\sigma^\Delta$;

(SAT2) $B_\sigma^\Delta \subseteq U$;

(SAT3) if $(\Delta' \vdash M) \in U$ and $\Delta' \subseteq \Delta''$ then $(\Delta'' \vdash M) \in U$;

(SAT4) if $(\Delta' \vdash M_1) \in U$, $M \to_k M_1$ and $(\Delta' \vdash M) \in SN_\sigma^\Delta$ then $(\Delta' \vdash M) \in U$.

Note that in the condition **(SAT4)** the reduction $\to_k$ can be equivalently replaced by $\to_k^*$.

Lemma 32. *The set SN_σ^Δ is saturated.*

Proof. Immediate. □

Lemma 33. *If $\mathcal{S} \subseteq SAT_\sigma^\Delta$ is a non-empty family then*

$$\bigcap \mathcal{S} \in SAT_\sigma^\Delta \text{ and } \bigcup \mathcal{S} \in SAT_\sigma^\Delta.$$

Proof. Immediate from the definition of a saturated set. □

Lemma 34. *The set $\bigcap SAT_\sigma^\Delta$ is saturated.*

Proof. It is a consequence of Lemma 33. □

Lemma 35.

$$\bigcap SAT_\sigma^\Delta = \{(\Delta' \vdash M) \in SN_\sigma^\Delta \mid \text{there is } M' \text{ such that } (\Delta' \vdash M') \in B_\sigma^\Delta \text{ and } M \to_k^* M'\}.$$

Proof. Let R denote the right hand side of the equation above. It is easy to observe that R is a saturated set. Thus

$$\bigcap SAT_\sigma^\Delta \subseteq R.$$

We will now prove

$$R \subseteq \bigcap SAT_\sigma^\Delta.$$

Let U be an arbitrary set in SAT_σ^Δ. We will prove $R \subseteq U$. Let $(\Delta' \vdash M) \in R$. By the definition of R there exists $(\Delta' \vdash M') \in B_\sigma^\Delta$ such that

$$\Delta' \supseteq \Delta, \quad (\Delta' \vdash M) \in SN_\sigma^\Delta, \quad M \to_k^* M'.$$

We proceed by induction with respect to the length of the reduction sequence $M \to_k^* M'$:

- The reduction sequence has zero steps. In this case $(\Delta' \vdash M) \in B_\sigma^\Delta$. Thus by the definition of a saturated set $(\Delta' \vdash M) \in U$.
- The reduction sequence has $n+1$ steps. Then $M \to_k M_1 \to_k^n M'$. By induction hypothesis $(\Delta' \vdash M_1) \in U$. By the definition of a saturated set, is must be the case that $(\Delta' \vdash M) \in U$.

Hence indeed $R \subseteq U$.

We have proved that $\bigcap SAT_\sigma^\Delta \subseteq R$ and $R \subseteq \bigcap SAT_\sigma^\Delta$. Thus

$$\bigcap SAT_\sigma^\Delta = R.$$

□

Lemma 36. *The set SAT_σ^Δ is a complete lattice with respect to inclusion.*

Proof. It is a consequence of Lemma 33. □

Lemma 37. *Let M be a key redex and $M \to_k N$. If $M \to_i M'$ then M' is a key redex and if $M \to_i M' \to_k L$ then there exists a sequence of reductions $M \to_k N \to^* L$.*

Proof. We proceed by induction with respect to the definition of $M \to_k N$. There are four cases.

Case 1: We have $M = (\lambda x : A.B)C$ and $N = B[x := C]$. If $M \to_i M'$ then

$$M' = (\lambda x : A'.B')C' \text{ and } A \to A' \text{ or } B \to B' \text{ or } C \to C'.$$

Of course M' is a key redex. It is easy to observe that if $M \to_i M' \to_k B'[x := C']$ then there exists a sequence of reductions $M \to_k N \to^* B'[x := C']$.

Case 2: We have $M = \mathrm{Elim}(I, Q, \vec{u}, \mathrm{Constr}(n, J)\vec{A})\{\vec{f}\}$ and $N = \triangle[C_n(I), f_n, \vec{A}, I, Q, \vec{f}]$. If $M \to_i M'$ then

$$M' = \mathrm{Elim}(I', Q', \vec{u'}, \mathrm{Constr}(n, J')\vec{A'})\{\vec{f'}\}$$

and

$$I \to I', \text{ or } Q \to Q', \text{ or } u_i \to u'_i, \text{ or } J \to J', \text{ or } A_i \to A'_i, \text{ or } f_i \to f'_i.$$

Of course M' is a key redex. It is easy to observe that if

$$M \to_i M' \to_k \triangle[C_n(I'), f'_n, \vec{A'}, I', Q', \vec{f'}]$$

then there exists a sequence of reductions $M \to_k N \to^* \triangle[C_n(I'), f'_n, \vec{A'}, I', Q', \vec{f'}]$.

Case 3: We have $M = AB$ and $N = A'B'$ and $A \to_k A'$. Then the conclusion follow easily from induction hypothesis. Note that A cannot be an abstraction as it is a key redex.

Case 4: We have $M = \mathrm{Elim}(I, Q, \vec{u}, A)\{\vec{f}\}$ and $N = \mathrm{Elim}(I, Q, \vec{u}, A')\{\vec{f}\}'$ and $A \to_k A'$. The conclusion follows from induction hypothesis. Note that A cannot be a constructor as it is a key redex. □

Corollary 38. *Let M be a key redex. Suppose $M \to_k N$ and every sequence of internal reductions beginning in M is finite. If N is strongly normalizing then M is strongly normalizing.*

Proof. Consequence of Lemma 37. □

5.2. Families of saturated sets

We begin by defining a measure m. If T is a sort, a kind, a type or a type constructor we define the measure $m(T)$ by induction as follows

- $m(\Box^t) = 1$,
- $m(\Box^p) = 1$,
- $m(*^t) = 1$,
- $m(*^p) = 1$,
- $m(p) = 1$, if p is a variable,
- $m(\Pi x : A.B) = \max(m(A), m(B)) + 1$,
- $m(\kappa M) = m(\kappa)$,

- $m(\lambda x : A.\kappa) = m(\kappa)$,
- $m(\mathrm{Ind}(X : *^t)\{\vec{C}\}) = \max_i(m(C_i(X))) + 1$.

Lemma 39. *Let A be a sort, a kind, a type or a type constructor. If M is an object then $m(A) = m(A[x := M])$.*

Proof. Easy induction with respect to the structure of A. □

Lemma 40. *Let A, B be two sorts, kinds, types or type constructors such that $A =_{\beta\iota} B$. Then $m(A) = m(B)$.*

Proof. If A is a sort then by Theorem 25 and Generation Lemma the term B is also a sort and $m(A) = 1 = m(B)$. If A is a kind then by Lemma 27 the term B is also a kind. By Theorem 25 we have $A = \Pi\vec{x} : \vec{T_1}.s$ and $B = \Pi\vec{x} : \vec{T_2}.s$. We proceed by induction with respect to the length of $\vec{T_1}$. Let n be the length of the vector $\vec{T_1}$. By Theorem 25 the vector $\vec{T_2}$ is also of length n. If $n = 0$ then A and B are sorts and $m(A) = 1 = m(B)$. If $n > 0$ then $A = \Pi x : T_1^1.A_1$ and $B = \Pi x : T_2^1.B_1$ and by Theorem 25 we have $T_1^1 =_{\beta\iota} T_2^1$. By the induction hypothesis

$$m(T_1^1) = m(T_2^1),$$
$$m(A_1) = m(B_1)$$

and thus $m(A) = m(B)$.

If A is a type or a type constructor then B is also a type or a type constructor. By Theorem 25 there exists a term C such that $A \to_{\beta\iota}^* C$ and $B \to_{\beta\iota}^* C$. Using Lemma 39 we note that if $M \to_{\beta\iota} M'$ then $m(M) = m(M')$. Thus

$$m(A) = m(B) = m(C).$$ □

Let Γ be a context. If $\Gamma \vdash A : T$ we define the domain of interpretation $V_\Gamma(A)$. Simultaneously, for $C \in V_\Gamma(A)$ and $\Gamma' \supseteq \Gamma$, we define the *restriction of C to the context* Γ' denoted $C|_{\Gamma'}$ such that $C|_{\Gamma'} \in V_{\Gamma'}(A)$. The restriction $C|_{\Gamma'}$ is the part of interpretation relevant to the context Γ'. If A is a small term then we define

$$V_\Gamma(A) = \{\emptyset\} \quad \text{and} \quad \emptyset|_{\Gamma'} = \emptyset.$$

If A is a large term then we define $V_\Gamma(A)$ by induction with respect to $m(T)$. In the definition we will use the abbreviation

$$\overline{T}_\tau^\Gamma = \{(\Gamma' \vdash M, C) \mid \Gamma \subseteq \Gamma', \Gamma' \vdash M : \tau, C \in V_{\Gamma'}(M)\}.$$

The definition of $V_\Gamma(A)$ follows:

- If A is a type, a formula, a kind or a sort then $V_\Gamma(A) = SAT_A^\Gamma$. In this case if $C \in V_\Gamma(A)$ then $C|_{\Gamma'} = \{(\Gamma'' \vdash M) \in C \mid \Gamma' \subseteq \Gamma''\}$.
- If A is an acceptor of an argument of type τ then $V_\Gamma(A)$ is the set consisting of functions f with the domain $\overline{T}_\tau^\Gamma$ such that $f(\Gamma' \vdash M, C) \in V_{\Gamma'}(AM)$ and
  - $f(\Gamma' \vdash M_1, C) = f(\Gamma' \vdash M_2, C)$, if $M_1 =_{\beta\iota} M_2$,

– $f(\hat\Gamma \vdash M, C)|_{\Gamma'} = f(\Gamma' \vdash M, C|_{\Gamma'})$, if $\Gamma \subseteq \hat\Gamma \subseteq \Gamma'$, $\hat\Gamma \vdash M : \tau$ and $C \in V_{\hat\Gamma}(M)$.

In this case, if $f \in V_\Gamma(A)$ then $f|_{\Gamma'}$ is a function with the domain $\overline{T}_\tau^{\Gamma'}$ such that for $\Gamma'' \supseteq \Gamma'$

$$f|_{\Gamma'}(\Gamma'' \vdash M, C) = f(\Gamma'' \vdash M, C).$$

Suppose $\Gamma \vdash M : \mathrm{Ind}(X : *^t)\{\vec{C}\}$ and $\mathrm{Ind}(X : *^t)\{\vec{C}\}$ is a large inductive type with n constructors.

- If $M =_{\beta\iota} \mathrm{Constr}(k, J)\vec{N}$ where $\vec{N} = (N_1, \ldots, N_r)$ then $V_\Gamma(M) = \{k\} \times \Pi_{i=1}^r V_\Gamma(N_i)$.

  In this case, for $C = \langle k, U_1, \ldots, U_j\rangle \in V_\Gamma(M)$ we define $C|_{\Gamma'} = \langle k, U_1|_{\Gamma'}, \ldots, U_j|_{\Gamma'}\rangle$.

- Otherwise $V_\Gamma(M) = \{0\}$. In this case, if $C \in V_\Gamma(M)$ then $C|_{\Gamma'} = \{0\}$.

Note that the definition of $V_\Gamma(A)$ is correct. If $\Gamma \vdash A : T_A$ is an acceptor of an argument of type τ, a term M is an argument of type τ in the context $\Gamma' \supseteq \Gamma$ and $\Gamma' \vdash AM : T_{AM}$ then we have $m(T_{AM}) < m(T_A)$. Hence $V_{\Gamma'}(AM)$ is defined before $V_\Gamma(A)$. Similarly, if I is an inductive type then $m(I)$ is greater than $m(C_i)$ for every type of constructor C_i of I. By the Church-Rosser property for non-proofs (Theorem 25) if $M =_{\beta\iota} \mathrm{Constr}(n, I)\vec{N}$ and $M =_{\beta\iota} \mathrm{Constr}(k, J)\vec{P}$ then $n = k$ and $I =_{\beta\iota} J$, and the sequences $\vec{N}$, $\vec{P}$ have the same length and for every i we have $N_i =_{\beta\iota} P_i$. The correctness of the definition for inductive objects follows from the following lemma.

Lemma 41. *If $\Gamma \vdash A : T$ and $\Gamma \vdash B : T_1$ and $A =_{\beta\iota} B$ then*

$$V_\Gamma(A) = V_\Gamma(B).$$

Proof. Induction with respect to $m(T)$.

- If A is a sort, a kind, a type or a formula, and $A =_{\beta\iota} B$ then

$$V_\Gamma(A) = SAT_A^\Gamma = SAT_B^\Gamma = V_\Gamma(B).$$

- Otherwise the conclusion follows from the induction hypothesis. □

For every term A in Γ we define a relation $\leq$ in $V_\Gamma(A)$. Let $f, f' \in V_\Gamma(A)$. If $V_\Gamma(A) = \{\emptyset\}$ then $f \leq f'$. If $V_\Gamma(A)$ is a family of saturated sets then $f \leq f'$ if and only if $f \subseteq f'$. If $V_\Gamma(A)$ is a family of functions then every function in $V_\Gamma(A)$ has the same domain. We say that $f \leq f'$ if for every $a \in dom(f)$ we have $f(a) \leq f'(a)$. If $V_\Gamma(A)$ is a family of n-tuples then we say that $f \leq f'$ if $f_0 = f'_0$ and for every $i = 1, \ldots, n-1$ we have $f_i \leq f'_i$.

Lemma 42. *Suppose $\Gamma \vdash A : T$. Then the set $V_\Gamma(A)$ is a lattice with respect to $\leq$.*

Proof. We proceed by induction with respect to $m(T)$.

- If A is a small term then the conclusion is obvious.

- If A is a type, or a formula, a kind then the conclusion follows from Lemma 36.

- Suppose A is an acceptor of an argument of type τ. Let $D \subseteq V_\Gamma(A)$. We will find the supremum of the set D. Define a function f with the domain $\overline{T}_\tau^\Gamma$ in the following way:

$$f(\Delta \vdash a, C) = \sup\{g(\Delta \vdash a, C) \mid g \in D\}$$

  where $\sup X$ denotes the supremum of a set X. Note that the supremum exists by the induction hypothesis and thus the function is well defined. It is easy to observe that $f \in V_\Gamma(A)$ and that $f = \sup D$.

- If A is an inductive object and $D \subseteq V_\Gamma(A)$ then every $C \in D$ is of the form $C = \langle k, C_1, \ldots, C_j \rangle$. Then it is easy to prove that $\sup D = \langle k, \sup D_1, \ldots, \sup D_j \rangle$ where $D_i = \{C_i \mid \langle k, C_1, \ldots, C_j \rangle \in D\}$. □

Lemma 43. *If τ is a kind, a type, a formula or a sort and $C \in V_\Gamma(\tau)$, and $\Gamma' \supseteq \Gamma$ then $C|_{\Gamma'} \in V_{\Gamma'}(\tau)$.*

Proof. If τ is a type, a formula, a kind, or a sort then $V_\Gamma(\tau) = SAT_\tau^\Delta$. Recall that then $C|_{\Gamma'} = \{(\Gamma'' \vdash M) \mid \Gamma'' \supseteq \Gamma' \text{ and } (\Gamma'' \vdash M) \in C\}$. We prove that $C|_{\Gamma'}$ is a saturated set. We check the four conditions in the definition of a saturated set.

(SAT1) $B_A^{\Gamma'} \subseteq C|_{\Gamma'}$. It is obvious.

(SAT2) $C|_{\Gamma'} \subseteq SN_A^{\Gamma'}$. It is obvious.

(SAT3) Let $(\Gamma'' \vdash M) \in C|_{\Gamma'}$ and $\Gamma'' \subseteq \Gamma'''$. Then $(\Gamma'' \vdash M) \in C$. Because C is a saturated set it also holds that $(\Gamma''' \vdash M) \in C$. But $\Gamma''' \supseteq \Gamma'$ and thus $(\Gamma''' \vdash M) \in C|_{\Gamma'}$.

(SAT4) Let $(\Gamma'' \vdash M') \in C|_{\Gamma'}$ and $\Gamma'' \vdash M \in SN_A^{\Gamma'}$ and $M \to_k M'$. Then $(\Gamma'' \vdash M) \in C$. As C is a saturated set it also holds that $(\Gamma'' \vdash M') \in C$. Thus $(\Gamma'' \vdash M) \in C|_{\Gamma'}$. □

Lemma 44. *Let $\Gamma \vdash A : T$ and $\Gamma' \supseteq \Gamma$. If $C \in V_\Gamma(A)$ then*

$$C|_{\Gamma'}(A) \in V_{\Gamma'}(A).$$

Proof. The proof is by induction with respect to $m(T)$.

- If A is a type, or a formula, a kind then the conclusion follows from Lemma 43.

- If A is an acceptor of an argument of type τ and $f \in V_\Gamma(A)$ then $f|_{\Gamma'}$ is the restriction of f to the domain

$$\{(\Gamma'' \vdash M, C) \mid \Gamma'' \supseteq \Gamma', \Gamma'' \vdash M : \tau, C \in V_{\Gamma''}(M)\}.$$

  We have to check the three conditions:

  - $f|_{\Gamma'}(\Gamma'' \vdash M, C) \in V_{\Gamma''}(AM)$, which is obvious;
  - if $M_1 =_{\beta\iota} M_2$ then $f|_{\Gamma'}(\Gamma'' \vdash M_1, C) = f(\Gamma'' \vdash M_1, C) = f(\Gamma'' \vdash M_2, C) = f|_{\Gamma'}(\Gamma'' \vdash M_2, C)$;
  - if $\Gamma \subseteq \hat{\Gamma} \subseteq \Gamma'$, $\hat{\Gamma} \vdash M : \tau$ then $f|_{\Gamma'}(\hat{\Gamma} \vdash M, C)|_{\Gamma''} = f(\hat{\Gamma} \vdash M, C)|_{\Gamma''} = f(\Gamma'' \vdash M, C|\Gamma'') = f|_{\Gamma'}(\Gamma'' \vdash M, C|_{\Gamma''})$.

- If A is an inductive object and $C \in V_\Gamma(A)$ then $C = \langle k, C_1, \ldots, C_j \rangle$. Then $C|_{\Gamma'} = \langle k, C_1|_{\Gamma'}, \ldots, C_j|_{\Gamma'}$ and the conclusion follows from the induction hypothesis. □

Lemma 45. *Let $\Gamma \vdash A : T$ and $\Gamma'' \supseteq \Gamma' \supseteq \Gamma$, and $C \in V_\Gamma(A)$. Then*

$$C|_{\Gamma'}|_{\Gamma''} = C|_{\Gamma''}.$$

Proof. Induction with respect to $m(T)$. □

Special elements

If A is a term we define the *canonical element* $Can_\Gamma(A) \in V_\Gamma(A)$:

- $Can_\Gamma(A) = \emptyset$ if, A is a small term;
- $Can_\Gamma(*^t) = SN^\Gamma_{*^t}$;
- $Can_\Gamma(*^p) = SN^\Gamma_{*^p}$;
- $Can_\Gamma(\tau) = SN^\Gamma_\tau$, if τ is a type, or a formula, or a kind;
- $Can_\Gamma(A) = \lambda\!\!\lambda(\Gamma' \vdash M, C) \in \overline{T}^\Gamma_\tau.Can_{\Gamma'}(AM)$ if A is an acceptor of an argument of type τ;
- $Can_\Gamma(A) = \langle n, Can_\Gamma(\vec{N})\rangle$ if A is a large inductive object and $A =_{\beta\iota} \mathrm{Constr}(n, J)\vec{N}$,
- $Can_\Gamma(A) = 0$ if A is a large inductive object and A is not $\beta\iota$-equal to a term of the form $\mathrm{Constr}(n, J)\vec{N}$.

Lemma 46. *If A and A' are terms in the context Γ and $\Gamma \vdash A : T$, and $\Gamma \vdash A' : T$, and $A =_{\beta\iota} A'$ then $Can_\Gamma(A) = Can_\Gamma(A')$.*

Proof. Easy induction with respect to $m(T)$. □

Lemma 47. *If A is a term in the context Γ and $\Gamma \vdash A : T$ then*

- $Can_\Gamma(A) \in V_\Gamma(A)$,
- *if* $\Gamma \subseteq \Gamma'$ *then* $Can_\Gamma(A)|_{\Gamma'} = Can_{\Gamma'}(A)$.

Proof. Easy induction with respect to $m(T)$. □

If A is a term we define the *minimal element* $Min_\Gamma(A) \in V_\Gamma(A)$:

- $Min_\Gamma(A) = \emptyset$, if A is a small term;
- $Min_\Gamma(*^t) = SN^\Gamma_{*^t}$;
- $Min_\Gamma(*^p) = SN^\Gamma_{*^p}$;
- $Min_\Gamma(\tau) = \bigcap SAT^\Gamma_\tau$, if τ is a type, or a formula, or a kind;
- $Min_\Gamma(A) = \lambda\!\!\lambda(\Gamma' \vdash M, C) : \overline{T}^\Gamma_\tau.Min_{\Gamma'}(AM)$ if A is an acceptor of an argument of type τ;
- $Min_\Gamma(A) = \langle n, (Min_\Gamma(\vec{N}))\rangle$ if A is a large inductive object and $A =_{\beta\iota} \mathrm{Constr}(n, J)\vec{N}$;
- $Min_\Gamma(A) = 0$ if A is a large inductive object and A is not $\beta\iota$-equal to a term of the form $\mathrm{Constr}(n, J)\vec{N}$.

Lemma 48. *If A and A' are terms in the context Γ and $\Gamma \vdash A : T$, and $\Gamma \vdash A' : T$, and $A =_{\beta\iota} A'$ then $Min_\Gamma(A) = Min_\Gamma(A')$.*

Proof. Easy induction with respect to $m(T)$. □

Lemma 49. *If A is a term in the context Γ and $\Gamma \vdash A : T$ then*

- $Min_\Gamma(A) \in V_\Gamma(A)$,
- *if* $\Gamma \subseteq \Gamma'$ *then* $Min_\Gamma(A)|_{\Gamma'} = Min_{\Gamma'}(A)$.

Proof. Easy induction with respect to $m(T)$. □

5.2.1. Valuations and appropriate sequences

Let Γ be a context. A *constructor substitution of* Γ is a function ξ such that for each variable $x \in dom(\Gamma)$ there exist a context Γ' and terms M, T such that $\Gamma' \vdash M : T$ and $\xi(x) \in V_{\Gamma'}(M)$. An *object substitution* is a function ρ such that if $(x : T) \in \Gamma$ then $\Gamma' \vdash \rho(x) : T'$ for a certain context Γ' and a certain term T'.

By $\rho(A)$ we denote the term obtained from A by replacing each free variable x in $dom(\rho)$ with $\rho(x)$.

A *constructor valuation* is a pair $\langle \xi, \rho \rangle$ where ξ is a constructor substitution and ρ is an object substitution. We say that $\langle \xi, \rho \rangle$ *satisfies* Γ *at* Δ if for every pair $(x : T) \in \Gamma$ we have $\Delta \vdash \rho(x) : \rho(T)$ and $\xi(x) \in V_\Delta(\rho(x))$. A *constructor valuation for* Δ is a constructor valuation $\langle \xi, \rho \rangle$ such that $\xi(p) = Can_\Delta(\rho(p))$ for every type constructor variable p.

If ξ is a constructor substitution then by $\xi|_{\Delta'}$ we denote the substitution such that $\xi|_{\Delta'}(x) = \xi(x)|_{\Delta'}$ for every $x \in dom(\xi)$. If ρ is an object substitution then by $\rho; x{:=}A$ we denote an object substitution such that

$$(\rho; x{:=}A)(y) = \begin{cases} \rho(y), & \text{if } y \neq x, \\ A, & \text{if } y = x. \end{cases}$$

We use a similar notation for constructor substitutions.

Lemma 50. *If* $\Gamma \vdash M : T$ *and* $\langle \xi, \rho \rangle$ *is a constructor valuation satisfying* Γ *at* Δ *then* $\Delta \vdash \rho(M) : \rho(T)$.

Proof. The proof is by induction with respect to the structure of the derivation $\Gamma \vdash M : T$. We proceed by cases depending on the the last rule in the derivation.

(Ax) The conclusion is obvious.

(Var) The conclusion is obvious by the assumption.

(Weak) If $\Gamma = (\Gamma', x : A)$ then $\langle \xi, \rho \rangle$ satisfies Γ' at Δ and the conclusion follows from the induction hypothesis.

(Conv) The conclusion is obvious.

(Abs) We have

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

By the induction hypothesis

$$\Delta \vdash (\Pi x : \rho(A).\rho(B)) : \rho(s).$$

Moreover $\langle (\xi; x{:=}Can_\Delta(\rho(A))), (\rho; x{:=}x) \rangle$ is a constructor valuation which satisfies $(\Gamma, x : A)$ at $(\Delta, x : \rho(A))$ and thus

$$\Delta, x : \rho(A) \vdash \rho(M) : \rho(B).$$

Hence we get the conclusion.

(App) We have

$$\frac{\Gamma \vdash M : \Pi x : A.B \qquad \Gamma \vdash N : A}{\Gamma \vdash MN : B[x := N]}$$

By the induction hypothesis

$$\Delta \vdash \rho(M) : (\Pi x{:}\rho(A).\rho(B)) \text{ and } \Delta \vdash \rho(N) : \rho(A).$$

Then $\Delta \vdash \rho(M)\rho(N) : \rho(B)[x := \rho(N)]$. But $\rho(B)[x := \rho(N)] = \rho(B[x := N])$ and thus we get the conclusion.

(Prod) We have

$$\frac{\Gamma \vdash A : s_1 \qquad \Gamma, x : A \vdash B : s_2}{\Gamma \vdash (\Pi x{:}A.B) : s_3}$$

By the induction hypothesis $\Delta \vdash \rho(A) : s_1$. Let

$$\rho_1 = \rho; x{:=}x \quad \text{ and } \quad \xi_1 = \xi; x{:=}Can_\Delta(\rho(A)).$$

Note that $\langle \xi_1, \rho_1 \rangle$ is a constructor valuation which satisfies $(\Gamma, x : A)$ at $(\Delta, x : \rho(A))$. Thus $\Delta, x : \rho(A) \vdash \rho(B) : s_2$ and we get the conclusion.

In the remaining cases the conclusion follows immediately from the induction hypothesis. □

Let Γ and Δ be contexts and $\langle \xi, \rho \rangle$ be a constructor valuation which satisfies Γ at Δ. Let $\vec{\tau}$ be a sequence of types, kinds, formulas or $*^p$ in Γ and $\vec{x}$ be a sequence of variables of the same length. We define an auxiliary notion of an *appropriate sequence of arguments for* $(\vec{x} : \vec{\tau})$ *at* $\langle \xi, \rho \rangle$ *in* Δ by induction with respect to the length p of the sequence $\vec{\tau}$. Simultaneously, we define a sequence $(\langle \xi_i, \rho_i \rangle)_{i=0}^{p-1}$ of constructor valuations which we will call a *sequence associated with the appropriate sequence of arguments*. In the definition we take $\xi_{-1} = \xi$, $\rho_{-1} = \rho$ and $\Delta_{-1} = \Delta$. An appropriate sequence of arguments is a sequence $(A_i)_{i=0}^{p-1}$ of triples of the form (Δ_i, N_i, C_i) such that each Δ_i is a context, each N_i is a term and each C_i is a set. Note that we have $C_i = \emptyset$, if N_i is small.

- The empty sequence is an appropriate sequence of arguments for the empty sequence ϵ at $\langle \xi, \rho \rangle$ in Δ.
- The sequence $(A_i)_{i=0}^{p-1}$ is an appropriate sequence of arguments for $(x_i : \tau_i)_{i=0}^{p-1}$ at $\langle \xi, \rho \rangle$ in Δ if and only if $(A_i)_{i=0}^{p-2}$ is an appropriate sequence of arguments for $(x_i : \tau_i)_{i=0}^{p-2}$ at $\langle \xi, \rho \rangle$ in Δ and $A_{p-1} = (\Delta_{p-1}, N_{p-1}, C_{p-1})$ where

  $$\Delta_{p-1} \supseteq \Delta_{p-2}, \qquad \Delta_{p-1} \vdash N_{p-1} : \rho_{p-2}(\tau_{p-1}) \qquad \text{and} \qquad C_{p-1} \in V_{\Delta_{p-1}}(N_{p-1}).$$

  In this case

  $$\xi_{p-1} = \xi_{p-2}; x_{p-1} := C_{p-1}, \qquad \text{and} \qquad \rho_{p-1} = \rho_{p-2}; x_{p-1} := N_{p-1}.$$

5.3. The interpretation of terms

For two contexts Γ, Δ and a constructor valuation $\langle\xi,\rho\rangle$ which satisfies Γ at Δ, and a large term A we define the interpretation of A, denoted $[\Gamma\vdash A]_{\xi,\rho,\Delta}$, by induction. Keep in mind that we want to have the property

$$[\Gamma\vdash A]_{\xi,\rho,\Delta}\in V_\Delta(\rho(A)).$$

The definition follows:

- $[\Gamma\vdash\Box^t]_{\xi,\rho,\Delta}=SN^\Delta_{\Box^t}$,
- $[\Gamma\vdash\Box^p]_{\xi,\rho,\Delta}=SN^\Delta_{\Box^p}$,
- $[\Gamma\vdash *^t]_{\xi,\rho,\Delta}=SN^\Delta_{*^t}$,
- $[\Gamma\vdash *^p]_{\xi,\rho,\Delta}=SN^\Delta_{*^p}$,
- $[\Gamma\vdash\alpha]_{\xi,\rho,\Delta}=\xi(\alpha)$, if α is a large variable,
- $[\Gamma\vdash PQ]_{\xi,\rho,\Delta}=[\Gamma\vdash P]_{\xi,\rho,\Delta}(\Delta\vdash\rho(Q),[\Gamma\vdash Q]_{\xi,\rho,\Delta})$ if P and Q are large objects or type constructors,
- $[\Gamma\vdash PQ]_{\xi,\rho,\Delta}=[\Gamma\vdash P]_{\xi,\rho,\Delta}(\Delta\vdash\rho(Q),\emptyset)$ if P is a large object or a type constructor and Q is a small object,
- $[\Gamma\vdash\lambda x:\tau.A]_{\xi,\rho,\Delta}=\lambda\!\!\lambda(\Delta'\vdash M,C):\overline{T}^\Delta_{\rho(\tau)}.[\Gamma,x:\tau\vdash A]_{(\xi|_{\Delta'};x:=C),(\rho;x:=M),\Delta'}$
- $[\Gamma\vdash\Pi x:\tau.B]_{\xi,\rho,\Delta}=\{(\Delta'\vdash M)\mid\Delta\subseteq\Delta'$ and $\Delta'\vdash M:\rho(\Pi x:\tau.B)$ and
  for every $\Delta''\supseteq\Delta'$, for every a such that $(\Delta''\vdash a)\in[\Gamma\vdash\tau]_{\xi|_{\Delta''},\rho,\Delta''}$,
  for every $C\in V_{\Delta''}(a)$ we have $(\Delta''\vdash Ma)\in[\Gamma,x:\tau\vdash B]_{(\xi|_{\Delta''};x:=C),(\rho;x:=a),\Delta''})\}$
- $[\Gamma\vdash\mathrm{Constr}(n,I)]_{\xi,\rho,\Delta}=\lambda\!\!\lambda\vec{X}.\langle n,U_1,\ldots,U_k\rangle$ where it is assumed that $C_n(I)=\Pi\vec{x}{:}\vec{T}.I$ and $\vec{X}=(\Delta_i,a_i,U_i)_{i=1}^p$ is an appropriate sequence of arguments for $(\vec{x}:\vec{T})$ at $\langle\xi,\rho\rangle$ in Δ

5.3.1. Interpretation of inductive types

The definition of interpretation of inductive types is more complicated than the interpretations given so far. It is a set of simple sequents which reduce "well", i.e. if the term reduces by the key reduction to a term of the form $\mathrm{Constr}(n,X)\vec{N}$ then its arguments $\vec{N}$ already belong to the interpretations of their types. Suppose that the n-th constructor of I is of type $C_n(I)$. Then $C_n(I)=\Pi\vec{x}:\vec{\tau}.I$ and every τ_j is a type. We would like to have the property:

$$(\Delta'\vdash M)\in[\Gamma\vdash I]_{\xi,\rho,\Delta}\Leftrightarrow(\text{if } M\to^*_k\mathrm{Constr}(n,I)\vec{N}\text{ then }(\Delta'\vdash N_j)\in[\Gamma\vdash\tau_j]_{\xi_j,\rho_j,\Delta'}).$$

However, if we used that property directly then our definition of interpretation would not be well-founded. Thus we introduce an auxiliary set $Interp(\Gamma'\vdash T)_{\xi,\rho,\Delta,X,S}$, where Γ' and Δ are contexts, X is a variable, S is a saturated set, T is a type in Γ' and $\langle\xi,\rho\rangle$ is a constructor valuation such that $\langle(\xi;X{:=}S),(\rho;X{:=}\rho(I))\rangle$ satisfies Γ' at Δ. The set S is an intended interpretation for the inductive type I and X is a variable representing the type I. The operator $Interp$ computes the interpretation of $(\Gamma'\vdash T)$ in an appropriate context and constructor valuation without referring to the interpretation of I and thus avoiding the vicious circle. It is defined by induction with respect to the structure of T:

- if $X \notin FV(T)$ then $Interp(\Gamma' \vdash T)_{\xi,\rho,\Delta,X,S} = [\Gamma' \vdash T]_{(\xi;X:=S),(\rho;X:=\rho(I)),\Delta}$,
- if $T = X$ then $Interp(\Gamma' \vdash T)_{\xi,\rho,\Delta,X,S} = S$
- if $T = \Pi x : A.B$ and $X \notin A$ then

$$\begin{aligned} Interp(\Gamma' \vdash T)_{\xi,\rho,\Delta,X,S} = \{(\Delta' \vdash M) \mid {} & \Delta \subseteq \Delta' \wedge \Delta' \vdash M : \rho(\Pi x : A.B) \wedge \\ & \text{for every } \Delta'' \supseteq \Delta', \text{ for every } a \text{ such that } (\Delta'' \vdash a) \in [\Gamma' \vdash A]_{\xi|_{\Delta''},\rho,\Delta''}, \\ & \text{for every } C \in V_{\Delta''}(a) \\ & \text{we have} (\Delta'' \vdash Ma) \in Interp(\Gamma', x : A \vdash B)_{(\xi|_{\Delta''};x:=C),(\rho;x:=a;\Delta''),X,S|_{\Delta''}})\}, \end{aligned}$$

The definition of $Interp$ is partial but it suffices to define the interpretation for the inductive type. Indeed, if $T = \Pi\vec{x} : \vec{\tau}.X$ is a type of constructor then X occurs strictly positively in every τ_i.

We can now define the interpretation of an inductive type. Let $I = \text{Ind}(X : *^t)\{\vec{C}\}$. Recall that then $V_\Delta(\rho(I)) = SAT^{\Delta}_{\rho(I)}$. We define the interpretation of I in the following way.

$$[\Gamma \vdash I]_{\xi,\rho,\Delta} = lfp(F_{\Gamma,I,\xi,\rho,\Delta})$$

where lfp is the least fixpoint operator and $F_{\Gamma,I,\xi,\rho,\Delta} : V_\Delta(\rho(I)) \to V_\Delta(\rho(I))$ is such that

$$\begin{aligned} F_{\Gamma,I,\xi,\rho,\Delta}(S) = \Big(\bigcap SAT^{\Delta}_{\rho(I)}\Big) \cup \{(\Delta' \vdash u) \in SN^{\Delta}_{\rho(I)} \mid {} & \\ \text{if } \Delta'' \supseteq \Delta' \text{ and } \Delta'' \vdash u \to^*_k \text{Constr}(n, X)\vec{N}, & \text{ and } C_n(X) = \Pi\vec{x} : \vec{T}.X \\ \text{then for every } j \text{ we have } (\Delta'' \vdash N_j) \in Interp(\Gamma^j \vdash T_j)_{\xi^j,\rho^j,\Delta'',X,S|_{\Delta''}}\} & \end{aligned}$$

where

$$\begin{aligned} \Gamma^j &= \Gamma, X : *^t, (x_i : T_i)_{i=1}^{j-1} \\ \xi^j &= \xi|_{\Delta''}; (x_i := Can_{\Delta''}(N_i))_{i=1}^{j-1} \\ \rho^j &= \rho; (x_i := N_i)_{i=1}^{j-1}. \end{aligned}$$

We have to prove the correctness of the above definition. We will first state some auxiliary and rather technical lemmas which will later be used to establish correctness of the definition of interpretation. As we may expect, the correctness proof will be done by induction with respect to the structure of an interpreted term. The hypotheses in the following lemmas imitate the induction hypothesis. In the following we say that the interpretation of a term M in the context Γ *depends only on the values of a constructor valuation for the free variables of* M if and only if for every context Δ, for every pair of constructor valuations $\langle\xi, \rho\rangle$, $\langle\xi', \rho'\rangle$ which satisfy Γ at Δ such that for every $x \in FV(M)$ we have $\rho(x) = \rho'(x)$ and $\xi(x) = \xi'(x)$ it holds that

$$[\Gamma \vdash M]_{\xi,\rho,\Delta} = [\Gamma \vdash M]_{\xi',\rho',\Delta}.$$

Lemma 51. *Let $\Gamma' \vdash I : *^t$ and let I be an inductive type. Suppose that*

1. *for each $\hat{\Gamma} \vdash \hat{A} : \hat{T}$ structurally smaller than $\Gamma' \vdash I : *^t$, where $\hat{A}$ is large, and every constructor valuation $\langle\hat{\xi}, \hat{\rho}\rangle$ satisfying $\hat{\Gamma}$ at $\hat{\Delta}$ we have $[\hat{\Gamma} \vdash \hat{A}]_{\hat{\xi},\hat{\rho},\hat{\Delta}} \in V_{\hat{\Delta}}(\hat{\rho}(A))$;*
2. *$C(X) = \Pi\vec{x} : \vec{\tau}.X$ is a type of constructor of I;*

3. *τ_i is a large type;*

4. *$\Gamma \vdash \tau_i : *^t$ is structurally smaller than the sequent $\Gamma' \vdash I : *^t$;*

5. *Δ is a context, $S \in V_\Delta(\rho(I))$ and $\langle(\xi; X := S), (\rho; X : \rho(I)\rangle$ is a constructor valuation which satisfies Γ at Δ.*

Then $Interp(\Gamma \vdash \tau_i)_{\xi,\rho,\Delta,X,S}$ is well defined.

Proof. Induction with respect to the structure of the derivation of $\Gamma \vdash \tau_i : *^t$. □

Lemma 52. *Let $\Gamma \vdash I : *^t$ and let I be an inductive type. Suppose that*

1. *for each $\hat{\Gamma} \vdash \hat{A} : \hat{T}$ structurally smaller than $\Gamma' \vdash I : *^t$, where $\hat{A}$ is large, and a constructor valuation $\langle\hat{\xi}, \hat{\rho}\rangle$ satisfying $\hat{\Gamma}$ at $\hat{\Delta}$ we have $[\hat{\Gamma} \vdash \hat{A}]_{\hat{\xi},\hat{\rho},\hat{\Delta}} \in V_{\hat{\Delta}}(\hat{\rho}(A))$;*

2. *$C(X) = \Pi\vec{x} : \vec{\tau}.X$ is a type of constructor of I;*

3. *τ_j is a large type, $\Gamma^j = \Gamma, X : *^t, (x_i : \tau_i)_{i=1}^{j-1}$;*

4. *$\Gamma^j \vdash \tau^j : *^t$ is structurally smaller than $\Gamma \vdash I : *^t$;*

5. *Δ is a context and $\langle\xi, \rho\rangle$ is a constructor valuation which satisfies Γ at Δ and $S \in V_\Delta(\rho(I))$;*

6. *$\Delta' \supseteq \Delta$, $\vec{N}$ is a vector of terms, for every $k \leq j$ the pair $\langle\xi^k, \rho^k\rangle$ is as follows:*

$$\xi^k = \xi|_{\Delta'}; (x_i := Can_{\Delta'}(N_i))_{i=1}^{k-1}$$
$$\rho^k = \rho; (x_i := N_i)_{i=1}^{k-1}$$

   and for each $k < j$ we have $\Gamma \vdash N_k : \rho^k(\tau_k)$.

Then $Interp(\Gamma^j \vdash \tau_j)_{\xi^j,\rho^j,\Delta',X,S|_{\Delta'}}$ is well defined.

Proof. By Lemma 51 it is enough to prove that $\langle(\xi^j; X := S|_{\Delta'}), (\rho^j; X := \rho(I))\rangle$ is a constructor valuation which satisfies Γ^j at Δ'.

Let x be a variable in $dom(\Gamma^j)$.

- If $x \in Dom(\Gamma)$ then by assumption $\xi^j(x) = \xi|_{\Delta'}(x) \in V_{\Delta'}(\rho(x))$ and

$$\Delta \vdash \rho(x) : \rho(\Gamma(x)).$$

- If $x = X$, then $\Delta \vdash \rho(I) : *^t$ by Lemma 50 and

$$(\rho^j; X := \rho(I))(X) = \rho(I) \text{ and } S|_{\Delta'} \in V_{\Delta'}(\rho(I)).$$

- If $x = x_i$ then

$$(\rho^j; X := I)(x_i) = N_i \text{ and } \Delta \vdash N_i : (\rho^j; X := \rho(I))(\tau_i), \text{ and}$$
$$\xi^j(x_i) = Can_{\Delta'}(N_i) \in V_{\Delta'}(N_i).$$

Thus the pair $\langle(\xi^j; X := S|_{\Delta'}), (\rho^j; X := \rho(I))\rangle$ is a constructor valuation which satisfies Γ^j at Δ'. □

Lemma 53. *Suppose* $\Gamma \vdash I : *^t$ *where* I *is an inductive type,* Δ *is a context,* $\langle \xi, \rho \rangle$ *is a constructor valuation which satisfies* Γ *at* Δ *and* $S \in SAT^{\Delta}_{\rho(I)}$*. Then*

$$F_{\Gamma,I,\xi,\rho,\Delta}(S) \in SAT^{\Delta}_{\rho(I)}.$$

Proof. The conclusion follows from the definition of $F_{\Gamma,I,\xi,\rho,\Delta}(S)$. □

Note that if α is a limit ordinal then $F^{\alpha}(\bigcap SAT^{\Delta}_{\rho(I)}) = \bigcup_{\alpha'<\alpha} F^{\alpha'}(\bigcap SAT^{\Delta}_{\rho(I)})$ is also a saturated set because by Lemma 33 the set of saturated sets is closed on arbitrary unions.

Lemma 54. *Suppose* Γ, Δ *are contexts,* $I = \mathrm{Ind}(X : *^t)\{\vec{C}\}$ *is an inductive type,* $\langle \xi, \rho \rangle$ *is a constructor valuation which satisfies* Γ *at* Δ*, the interpretation* S *is in* $V_{\Delta}(\rho(I))$*,* $C_n(X) = \Pi\vec{x}{:}\vec{\tau}.X$ *is a type of the n-th constructor of* I*, and* τ_j *is a large type, and* $\langle \xi^j, \rho^j \rangle$ *is a constructor valuation such that* $\langle (\xi^j; X := S), (\rho^j; X := \rho(I)) \rangle$ *satisfies* $(\Gamma, X : *^t, (x_i : \tau_i)_{i=0}^{j-1})$ *at* Δ*. Then*

$$Interp(\Gamma, X : *^t, (x_i : \tau_i)_{i=0}^{j-1} \vdash \tau_j)_{\xi^j,\rho^j,\Delta,X,S}$$

is defined and

$$Interp(\Gamma, X : *^t, (x_i : \tau_i)_{i=0}^{j-1} \vdash \tau_j)_{\xi^j,\rho^j,\Delta,X,S} = [\Gamma, X : *^t, (x_i : \tau_i)_{i=0}^{j-1} \vdash \tau_j]_{(\xi^j;X:=S),(\rho^j;X:=\rho(I)),\Delta}.$$

Proof. Easy induction with respect to the structure of τ_j. □

The following Lemma says that the values $\xi(x)$ for a formula or object variable x (i.e. subset or large inductive object variable) are irrelevant to the value of $[\Gamma \vdash \tau]_{\xi,\rho,\Delta}$ for a type or a type constructor τ. This justifies the choice of values for arguments in the definition of the interpretation of an inductive type: any choice is equally good.

Lemma 55. *Let* Γ, Δ *be two contexts and* $\langle \xi, \rho \rangle$, $\langle \xi', \rho \rangle$ *be two constructor valuations which satisfy* Γ *at* Δ *and differ only in formula or object variables (i.e.* $\xi(x) = \xi'(x)$ *for all non-formula and non-object variables). Moreover, suppose for every type constructor variable* p *such that* $\Gamma \vdash p : (\Pi\vec{x} : \vec{\tau}.*^t)$ *and for any two sequences of arguments*

$$\vec{A_1} = (\Delta_i, m_i, c_i^1)_{i=0}^r, \qquad \vec{A_2} = (\Delta_i, m_i, c_i^2)_{i=0}^r$$

appropriate for $(\vec{x} : \vec{\tau})$ *at* $\langle \xi, \rho \rangle$ *in* Δ*, we have* $\xi(p)(\vec{A_1}) = \xi'(p)(\vec{A_2})$*. If* $\Gamma \vdash \kappa : (\Pi\vec{x} : \vec{\tau}.*^t)$ *is a type constructor then for any two sequences of arguments*

$$\vec{B^1} = (\Delta_i, M_i, C_i^1)_{i=0}^s, \qquad \vec{B^2} = (\Delta_i, M_i, C_i^2)_{i=0}^s,$$

appropriate for $(\vec{x} : \vec{\tau})$ *at* $\langle \xi, \rho \rangle$ *in* Δ*, we have*

$$[\Gamma \vdash \kappa]_{\xi,\rho,\Delta}(\vec{B^1}) = [\Gamma \vdash \kappa]_{\xi',\rho,\Delta}(\vec{B^2}).$$

Note that if κ is a type then the lemma states that $[\Gamma \vdash \kappa]_{\xi,\rho,\Delta} = [\Gamma \vdash \kappa]_{\xi',\rho,\Delta}$.

Proof. Induction with respect to the structure of the term κ.
If $\kappa = p$ then

$$[\Gamma \vdash \kappa]_{\xi,\rho,\Delta} = \xi(p) = \xi'(p) = [\Gamma \vdash \kappa]_{\xi',\rho,\Delta}.$$

By the assumption for every two appropriate sequences of arguments

$$\vec{B}^1 = (\Delta_i, M_i, C_i^1)_{i=0}^s, \qquad \vec{B}^2 = (\Delta_i, M_i, C_i^2)_{i=0}^s$$

for $(\vec{x} : \vec{\tau})$ at $\langle \xi, \rho \rangle$ we have

$$[\Gamma \vdash \kappa]_{\xi,\rho,\Delta}(\vec{B}^1) = [\Gamma \vdash \kappa]_{\xi',\rho,\Delta}(\vec{B}^2).$$

If $\kappa = \Pi x : \tau'.\sigma$ then the conclusion follows from the induction hypothesis.

If $\kappa = \mathrm{Ind}(X : *^t)\{\vec{C}\}$. then

$$[\Gamma \vdash \kappa]_{\xi,\rho,\Delta} = \mathit{lfp}(F_{\Gamma,I,\xi,\rho,\Delta})$$

and

$$[\Gamma \vdash \kappa]_{\xi',\rho,\Delta} = \mathit{lfp}(F_{\Gamma,I,\xi',\rho,\Delta}).$$

The conclusion follows from the induction hypothesis and Lemma 54.
If $\kappa = \lambda x : \sigma.\kappa'$ then

$$\begin{aligned}
&[\Gamma \vdash \kappa]_{\xi,\rho,\Delta} = \\
&\quad \boldsymbol{\lambda\!\!\lambda}(\Delta' \vdash a, C) : \overline{T}^{\Delta}_{\rho(\sigma)}.[\Gamma, x : \sigma \vdash \kappa']_{(\xi|_{\Delta'};x:=C),(\rho;x:=a),\Delta'} = \\
&\quad \boldsymbol{\lambda\!\!\lambda}(\Delta' \vdash a, C) : \overline{T}^{\Delta}_{\rho(\sigma)}.[\Gamma, x : \sigma \vdash \kappa']_{(\xi'|_{\Delta'};x:=C),(\rho;x:=a),\Delta'} = \\
&[\Gamma \vdash \kappa]_{\xi',\rho,\Delta}.
\end{aligned}$$

By the induction hypothesis and the fact that x is an object variable for every two sequences of arguments

$$\vec{B}^1 = (\Delta_i, M_i, C_i^1)_{i=0}^s, \qquad \vec{B}^2 = (\Delta_i, M_i, C_i^2)_{i=0}^s$$

appropriate for $(\vec{x} : \vec{\tau})$ at $\langle \xi, \rho \rangle$ in Δ we have

$$\begin{aligned}
&[\Gamma \vdash \kappa]_{\xi,\rho,\Delta}(\vec{B}^1) = \\
&\quad [\Gamma, x : \tau \vdash \kappa']_{(\xi|_{\Delta_1};x:=C_1^1),(\rho;x:=M_1),\Delta_1}((B_i^1)_{i=1}^s) = \\
&\quad [\Gamma, x : \tau \vdash \kappa']_{(\xi'|_{\Delta_1};x:=C_1^2),(\rho;x:=M_1),\Delta_1}((B_i^1)_{i=1}^s) = \\
&\quad [\Gamma, x : \tau \vdash \kappa']_{(\xi'|_{\Delta_1};x:=C_1^2),(\rho;x:=M_1),\Delta_1}((B_i^2)_{i=1}^s) = \\
&[\Gamma \vdash \kappa]_{\xi',\rho,\Delta}(\vec{B}^2).
\end{aligned}$$

If $\kappa = \kappa' N$ then we have

$$\begin{aligned}
&[\Gamma \vdash \kappa]_{\xi,\rho,\Delta} = \\
&\quad [\Gamma \vdash \kappa']_{\xi,\rho,\Delta}(\Delta \vdash \rho(N), [\Gamma \vdash N]_{\xi,\rho,\Delta}) = \\
&\quad [\Gamma \vdash \kappa']_{\xi',\rho,\Delta}(\Delta \vdash \rho(N), [\Gamma \vdash N]_{\xi',\rho,\Delta}) = \\
&[\Gamma \vdash \kappa]_{\xi,\rho,\Delta}.
\end{aligned}$$

Then by the induction hypothesis

$$\begin{aligned}
&[\Gamma \vdash \kappa]_{\xi,\rho,\Delta}(\vec{B}^1) = \\
&\quad [\Gamma \vdash \kappa']_{\xi,\rho,\Delta}(\Delta \vdash \rho(N), [\Gamma \vdash N]_{\xi,\rho,\Delta})(\vec{B}^1) = \\
&\quad [\Gamma \vdash \kappa']_{\xi',\rho,\Delta}(\Delta \vdash \rho(N), [\Gamma \vdash N]_{\xi',\rho,\Delta})(\vec{B}^2) = \\
&[\Gamma \vdash \kappa]_{\xi,\rho,\Delta}(\vec{B}^2).
\end{aligned}$$

□

Lemma 56. *Suppose that for every* $\Gamma' \vdash \tau' : T$ *structurally smaller than* $\Gamma \vdash \tau : *^t$ *and every constructor valuation* $\langle \xi', \rho' \rangle$ *which satisfies* Γ' *at* Δ' *the value of* $[\Gamma' \vdash \tau']_{\xi',\rho',\Delta'}$ *depends only on the values of* ξ' *and* ρ' *for variables in* $FV(\tau')$. *If* $S \subseteq S'$ *then*

$$Interp(\Gamma \vdash \tau)_{\xi,\rho,\Delta,X,S} \subseteq Interp(\Gamma \vdash \tau)_{\xi,\rho,\Delta,X,S'}.$$

Proof. Easy consequence of the definition of the operator $Interp$. □

Lemma 57. *Suppose* $\Gamma \vdash I : *^t$ *and* I *is an inductive type. Assume that for each* $\Gamma' \vdash \tau' : T$ *structurally smaller than* $\Gamma \vdash I : *^t$ *and each constructor valuation* $\langle \xi', \rho' \rangle$ *which satisfies* Γ' *at* Δ' *the value of* $[\Gamma' \vdash \tau']_{\xi',\rho',\Delta'}$ *depends only on the values of* ξ' *and* ρ' *for variables in* $FV(\tau')$. *If* $S \subseteq S'$ *then* $F_{\Gamma,I,\xi,\rho,\Delta}(S) \subseteq F_{\Gamma,I,\xi,\rho,\Delta}(S')$.

Proof. Immediate from Lemma 56. □

Lemma 58. *Let* I *be an inductive type. Let* Γ, Δ *be two contexts. Suppose for each sequent* $\Gamma' \vdash N' : T$ *in the derivation of* $\Gamma \vdash I : *^t$, *for each context* Δ', *and each constructor valuation* $\langle \xi', \rho' \rangle$ *which satisfies* Γ' *at* Δ' *it holds that if* $\Delta' \subseteq \Delta''$ *then*

$$([\Gamma' \vdash N']_{\xi',\rho',\Delta'})|_{\Delta''} = [\Gamma' \vdash N']_{\xi'|_{\Delta''},\rho',\Delta''}.$$

If $\langle \xi, \rho \rangle$ *is a constructor valuation which satisfies* Γ *at* Δ *and* $\Delta \subseteq \Delta'$ *and* $S \in V_\Delta(\rho(I))$ *then*

$$F_{\Gamma,I,\xi,\rho,\Delta}(S)|_{\Delta'} = F_{\Gamma,I,\xi|_{\Delta'},\rho,\Delta'}(S|_{\Delta'}).$$

Proof. Immediate from Lemma 45: if $\Delta' \subseteq \Delta'''$ then $\xi|_{\Delta'}|_{\Delta'''} = \xi|_{\Delta'''}$ and $S|_{\Delta'}|_{\Delta'''} = S|_{\Delta'''}$. □

Lemma 59. *Let* I *be an inductive type. Let* Γ, Δ *be two contexts. Suppose that for each sequent* $\Gamma' \vdash N' : T$ *in the derivation of* $\Gamma \vdash I : *^t$, *for each context* Δ', *and for each constructor valuation* $\langle \xi', \rho' \rangle$ *which satisfies* Γ' *at* Δ' *if* $\Delta' \subseteq \Delta''$ *then*

$$([\Gamma' \vdash N']_{\xi',\rho',\Delta'})|_{\Delta''} = [\Gamma' \vdash N']_{\xi'|_{\Delta''},\rho',\Delta''}.$$

If $\langle \xi, \rho \rangle$ *is a constructor valuation which satisfies* Γ *at* Δ *and* $\Delta \subseteq \Delta'$ *then*

$$[\Gamma \vdash I]_{\xi,\rho,\Delta}|_{\Delta'} = [\Gamma \vdash I]_{\xi|_{\Delta'},\rho,\Delta'}.$$

Proof. If $\Delta \supseteq \Delta'$ then $[\Gamma \vdash I]_{\xi,\rho,\Delta}|_{\Delta'} = \mathit{lfp}(F_{\Gamma,I,\xi,\rho,\Delta})|_{\Delta'}$. But

$$\mathit{lfp}(F_{\Gamma,I,\xi,\rho,\Delta}) = \bigcup_{\alpha'<\alpha} F^{\alpha'}_{\Gamma,I,\xi,\rho,\Delta}(\bigcap SAT^{\Delta}_{\rho(I)})$$

for a certain α. To get the conclusion it is enough to prove that for every $S \in V_\Delta(\rho(I))$ we have

$$F_{\Gamma,I,\xi,\rho,\Delta}(S)|_{\Delta'} = F_{\Gamma,I,\xi|_{\Delta'},\rho,\Delta'}(S|_{\Delta'}).$$

This follows from Lemma 58. □

Lemma 60. *Let* I *be an inductive type. Let* Γ, Δ *be two contexts. Suppose for each sequent* $\Gamma' \vdash N' : T$ *in the derivation of* $\Gamma \vdash I : *^t$, *for each context* Δ', *and for each constructor valuation* $\langle \xi', \rho' \rangle$ *which satisfies* Γ' *at* Δ' *we have*

- $[\Gamma' \vdash N']_{\xi',\rho',\Delta'} \in V_{\Delta'}(\rho'(T))$,

- *if* $\Delta' \subseteq \Delta''$ *then*
$$([\Gamma' \vdash N']_{\xi',\rho',\Delta'})|_{\Delta''} = [\Gamma' \vdash N']_{\xi'|_{\Delta''},\rho',\Delta''}$$

and the value of $[\Gamma' \vdash N']_{\xi',\rho',\Delta'}$ *depends only on the values of* ξ' *and* ρ' *for variables in* $FV(N')$*. If* $\langle \xi, \rho \rangle$ *is a constructor valuation which satisfies* Γ *at* Δ *then*

- $[\Gamma \vdash I]_{\xi,\rho,\Delta} \in V_\Delta(\rho(I))$,
- *if* $\Delta \subseteq \Delta'$ *then* $[\Gamma \vdash I]_{\xi,\rho,\Delta}|_{\Delta'} = [\Gamma \vdash I]_{\xi|_{\Delta'},\rho,\Delta'}$.

Proof. By Lemma 53 and Lemma 57 the operator $F_{\Gamma,I,\xi,\rho,\Delta}$ is a well defined monotone operator on the complete lattice (see Lemma 36) $SAT^{\Delta}_{\rho(I)} = V_\Delta(\rho(I))$. Hence $lfp(F_{\Gamma,I,\xi,\rho,\Delta})$ exists and is a saturated set. Thus
$$[\Gamma \vdash I]_{\xi,\rho,\Delta} \in SAT^{\Delta}_{\rho(I)} = V_\Delta(\rho(I)).$$
□

If $(\Delta' \vdash M) \in lfp(F_{\Gamma,I,\xi,\rho,\Delta})$ then there exists the least number α such that $(\Delta' \vdash M) \in F^{\alpha}_{\Gamma,I,\xi,\rho,\Delta}(\bigcap SAT^{\Delta}_{\rho(I)})$. Note that α is never a limit ordinal. If $\alpha > 0$ then by
$$pred_{\Gamma,I,\xi,\rho,\Delta}(\Delta' \vdash M)$$
we denote the set $F^{\alpha'}_{\Gamma,I,\xi,\rho,\Delta}(\bigcap SAT^{\Delta}_{\rho(I)})$ where α' is the predecessor of α.

We define the set $\mathcal{D}_{\Gamma,I,\xi,\rho,\Delta}$ of all approximations of $lfp(F_{\Gamma,I,\xi,\rho,\Delta})$. Let β be the ordinal number such that $F^{\beta}_{\Gamma,I,\xi,\rho,\Delta}(\bigcap SAT^{\Delta}_{\rho(I)}) = lfp(F_{\Gamma,I,\xi,\rho,\Delta})$. Then
$$\mathcal{D}_{\Gamma,I,\xi,\rho,\Delta} = \{F^{\alpha}_{\Gamma,I,\xi,\rho,\Delta}(\bigcap SAT^{\Delta}_{\rho(I)}) \mid \alpha \leq \beta\}.$$

5.3.2. Interpretation of elimination terms

In this section we define the interpretation for elimination terms. We have to give this interpretation because we may create large terms using elimination operation. In most type theories with inductive types one only has to give this interpretation for small inductive types, because creating large terms from large inductive objects is not allowed. In LNTT with inductive types we can create large objects by eliminating large inductive objects. It seems that the definition of interpretation would be simpler if our inductive types were in predicative universe.

In the definition we use an auxiliary notion of smooth union of a set. We begin the section with the definition and we prove basic properties of this notion.

The smooth union

Let Δ be a context and A be a large term. We say that the set $\mathcal{F}$ is *consistent for* A *at* Δ if for each $f \in \mathcal{F}$ there exists $\Delta' \supseteq \Delta$ such that $f \in V_{\Delta'}(A)$ and there exists $g \in \mathcal{F}$ such that $g \in V_\Delta(A)$.

For a set $\mathcal{F}$ consistent for A at Δ we define the *smooth union* $\bigsqcup \mathcal{F}$. The smooth union of a family of sets is the usual union of sets. If A is an acceptor of an argument of type τ then $\bigsqcup \mathcal{F}$ is a function with the domain $\overline{T}^{\Delta}_{\tau}$ such that
$$(\bigsqcup \mathcal{F})(\Delta \vdash a, C) = \bigsqcup \{f(\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \mid f \in \mathcal{F}, \hat{\Delta} \supseteq \Delta', (\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \in dom(f)\}.$$

The smooth union of a set of tuples of the form $\langle n, U_1, \dots, U_k \rangle$ is the tuple $\langle n, \mathcal{U}_1, \dots, \mathcal{U}_k \rangle$ where $\mathcal{U}_i = \bigsqcup\{U_i \mid \langle n, U_1, \dots, U_k \rangle \in \mathcal{F}\}$, for every i.
We will prove some technical lemmas about the smooth unions. The lemmas will later be used to show the correctness of the definition of interpretation.

Lemma 61. *Let Δ be a context and A be a large term. Let $\mathcal{F}$ be a set consistent for A at Δ. Then*

$$\bigsqcup \mathcal{F} \in V_\Delta(A).$$

Proof. We proceed by induction with respect to the definition of $V_\Delta(A)$.

- If A is a sort, a type, a kind or a formula then for every context Δ it is the case that $V_\Delta(A) = SAT_A^\Delta$. We have to prove that

$$\bigsqcup \mathcal{F} \in SAT_A^\Delta.$$

  The proof is a routine check of the four conditions.

- If A is an acceptor of an argument of type τ then $\bigsqcup \mathcal{F}$ is the function with the domain $\overline{T}_\tau^\Delta$. By the definition

$$\bigsqcup \mathcal{F}(\Delta' \vdash a, C) = \bigsqcup\{f(\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \mid f \in \mathcal{F}, \hat{\Delta} \supseteq \Delta', (\Delta' \vdash a, C|_{\hat{\Delta}}) \in dom(f)\}.$$

  Note that if $f \in \mathcal{F}$, $\hat{\Delta} \supseteq \Delta'$, $(\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \in dom(f)$ then $f(\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \in V_{\hat{\Delta}}(Aa)$. Moreover by the assumption there exists $f \in \mathcal{F}$ such that $f \in V_\Delta(A)$. Thus $f(\Delta \vdash a, C) \in V_\Delta(Aa)$. By the induction hypothesis

$$\bigsqcup\{f(\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \mid f \in \mathcal{F}, \hat{\Delta} \supseteq \Delta', (\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \in dom(f)\} \in V_\Delta(Aa).$$

  If $a =_{\beta\iota} b$ then

$$f(\Delta' \vdash a, C|_{\hat{\Delta}}) = f(\Delta' \vdash b, C|_{\hat{\Delta}}).$$

  Thus

$$(\bigsqcup \mathcal{F})(\Delta' \vdash a, C|_{\Delta'}) = (\bigsqcup \mathcal{F})(\Delta' \vdash b, C|_{\Delta'}).$$

  Now

$$\begin{aligned}
&(\bigsqcup \mathcal{F})(\Delta' \vdash a, C|_{\Delta'})|_{\Delta''} \\
&= (\bigsqcup\{f(\hat{\Delta} \vdash a, C|_{\Delta'}|_{\hat{\Delta}}) \mid f \in \mathcal{F}, \hat{\Delta} \supseteq \Delta', (\hat{\Delta} \vdash a, C|_{\Delta'}|_{\hat{\Delta}}) \in dom(f)\})|_{\Delta''} \\
&= (\bigsqcup\{f(\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \mid f \in \mathcal{F}, \hat{\Delta} \supseteq \Delta', (\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \in dom(f)\})|_{\Delta''} \\
&= \bigsqcup(\{f(\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \mid f \in \mathcal{F}, \hat{\Delta} \supseteq \Delta'', (\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \in dom(f)\} \\
&\qquad \cup \{f(\hat{\Delta} \vdash a, C|_{\hat{\Delta}})|_{\Delta''} \mid f \in \mathcal{F}, \Delta'' \supseteq \hat{\Delta} \supseteq \Delta', (\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \in dom(f)\}) \\
&= \bigsqcup(\{f(\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \mid f \in \mathcal{F}, \hat{\Delta} \supseteq \Delta'', (\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \in dom(f)\} \\
&\qquad \cup \{f(\Delta'' \vdash a, C_{\hat{\Delta}}|_{\Delta''}) \mid f \in \mathcal{F}, \Delta'' \supseteq \hat{\Delta} \supseteq \Delta', (\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \in dom(f)\}) \\
&= \bigsqcup(\{f(\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \mid f \in \mathcal{F}, \hat{\Delta} \supseteq \Delta'', (\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \in dom(f)\} \\
&\qquad \cup \{f(\Delta'' \vdash a, C|_{\Delta''}) \mid f \in \mathcal{F}, \Delta'' \supseteq \hat{\Delta} \supseteq \Delta', (\Delta'' \vdash a, C|_{\Delta''}) \in dom(f)\}) \\
&= \bigsqcup\{f(\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \mid f \in \mathcal{F}, \hat{\Delta} \supseteq \Delta'', (\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \in dom(f)\} \\
&= (\bigsqcup \mathcal{F})(\Delta'' \vdash a, C|_{\Delta''}).
\end{aligned}$$

- A is a large inductive object and $A =_{\beta\iota} \mathrm{Constr}(n, J)\vec{N}$. Then every $f \in \mathcal{F}$ has the form $f = \langle n, U_1, \ldots, U_k \rangle$ and

  $$\bigsqcup \mathcal{F} = \langle n, \bigsqcup \{\pi_1(f) \mid f \in \mathcal{F}\}, \ldots, \bigsqcup \{\pi_k(f) \mid f \in \mathcal{F}\} \rangle.$$

  By the induction hypothesis

  $$\bigsqcup \{\pi_i(f) \mid f \in \mathcal{F}\} \in V_\Delta(N_i)$$

  and thus

  $$\bigsqcup \mathcal{F} \in V_\Delta(A).$$ □

Note that

$$(\bigsqcup \mathcal{F})|_{\Delta''} = \bigsqcup \left(\{f \in \mathcal{F} \mid f \in V_{\Delta''}(A)\} \cup \{f|_{\Delta''} \mid f \in \mathcal{F}, f \in V_{\Delta'}(A), \Delta'' \supseteq \Delta' \supseteq \Delta\} \right).$$

Lemma 62. *Let $f \in V_\Delta(A)$ and let $\mathcal{F} = \{f|_{\Delta'} \mid \Delta' \supseteq \Delta\}$. Then*

$$\bigsqcup \mathcal{F} = f.$$

Proof. We proceed by induction with respect to the definition of $\bigsqcup \mathcal{F}$.

- If $V_\Delta(A) = SAT_A^\Delta$ then for every $\Delta' \supseteq \Delta$ it holds that $f|_{\Delta'} \subseteq f$. Then

  $$\bigsqcup \mathcal{F} = \bigcup \mathcal{F} = f.$$

- If A is an acceptor of an argument of type τ then

  $$\begin{aligned}
  &(\bigsqcup \mathcal{F})(\Delta' \vdash a, C) \\
  &\quad = \bigsqcup \{g(\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \mid g \in \mathcal{F}, \hat{\Delta} \supseteq \Delta', (\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \in \mathit{dom}(g)\} \\
  &\quad = \bigsqcup \{f|_{\Delta''}(\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \mid \hat{\Delta} \supseteq \Delta'' \supseteq \Delta'\} \\
  &\quad = \bigsqcup \{f(\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \mid \hat{\Delta} \supseteq \Delta'\} \\
  &\quad = \bigsqcup \{f(\Delta' \vdash a, C|_{\Delta'})|_{\hat{\Delta}} \mid \hat{\Delta} \supseteq \Delta'\} \\
  &\quad = f(\Delta' \vdash a, C|_{\Delta'}).
  \end{aligned}$$

- If A is a large inductive object and $A =_{\beta\iota} \mathrm{Constr}(n, J)\vec{N}$ then for every i

  $$\bigsqcup \{\pi_i(f|_{\Delta'}) \mid \Delta' \supseteq \Delta\} = \bigsqcup \{\pi_i(f)|_{\Delta'} \mid \Delta' \supseteq \Delta\} = \pi_i(f).$$

  Thus

  $$\bigsqcup \mathcal{F} = f.$$ □

Recall the definition of the relation $\leq$ introduced on page 52.

Lemma 63. *Let Δ and Δ' be contexts such that $\Delta \subseteq \Delta'$. Let A be a large term in Δ. Suppose $\mathcal{F}$, $\mathcal{F}'$ are two sets consistent for A at Δ such that $\mathcal{F}' \subseteq \mathcal{F}$. Suppose that for every $f \in \mathcal{F} - \mathcal{F}'$*

- *if* $f \in V_{\hat{\Delta}}(A)$ *and* $\Delta' \subseteq \hat{\Delta}$ *then there exists* $g \in \mathcal{F}'$ *such that* $f \leq g|_{\hat{\Delta}}$*;*
- *if* $f \in V_{\hat{\Delta}}(A)$ *and* $\hat{\Delta} \subseteq \Delta'$ *then there exists* $g \in \mathcal{F}'$ *such that* $f|_{\Delta'} \leq g$*.*

Then

$$(\bigsqcup \mathcal{F})|_{\Delta'} = (\bigsqcup \mathcal{F}')|_{\Delta'}.$$

Proof. We proceed by induction with respect to the definition of the smooth sum $\bigsqcup \mathcal{F}$. If A is a sort, a type, a formula or a kind then $\bigsqcup \mathcal{F}$ is the usual union of sets and the conclusion is obvious.

Suppose A is an acceptor of an argument of type τ and $\Delta \subseteq \Delta'$. The domains of $(\bigsqcup \mathcal{F})|_{\Delta'}$ and $(\bigsqcup \mathcal{F}')|_{\Delta'}$ are the same. Let $(\Delta' \vdash a, C)$ be an arbitrary argument in their domain. Then

$$\begin{aligned}(\bigsqcup \mathcal{F})(\Delta' \vdash a, C) = \bigsqcup F_1 \text{ where}\\ F_1 = \{f(\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \mid f \in \mathcal{F}, (\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \in dom(f), \hat{\Delta} \supseteq \Delta'\}\end{aligned}$$

and

$$\begin{aligned}(\bigsqcup \mathcal{F}')(\Delta' \vdash a, C) = \bigsqcup F_2 \text{ where}\\ F_2 = \{g(\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \mid g \in \mathcal{F}', (\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \in dom(g), \hat{\Delta} \supseteq \Delta'\}.\end{aligned}$$

Note that $F_2 \subseteq F_1$. If $f(\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \in F_1 - F_2$ then $f \in \mathcal{F} - \mathcal{F}'$. There exists $g \in \mathcal{F}'$ such that $(\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \in dom(g)$ and $f(\hat{\Delta} \vdash a, C|_{\hat{\Delta}}) \leq g(\hat{\Delta} \vdash a, C|_{\hat{\Delta}})$. By the induction hypothesis

$$(\bigsqcup F_1)|_{\Delta'} = (\bigsqcup F_2)|_{\Delta'}$$

and thus

$$(\bigsqcup \mathcal{F})(\Delta' \vdash a, C,) = (\bigsqcup \mathcal{F}')(\Delta' \vdash a, C).$$ □

Discussion

We now return to the definition of interpretation for inductive types. The basic property we want to achieve is preserving iota reduction:

$$[\Gamma \vdash \text{Elim}(I, Q, \text{Constr}(n, I')\vec{N})\{\vec{f}\}]_{\xi,\rho,\Delta} = [\Gamma \vdash f_n\vec{e}\,[C(X), \vec{N}, I, Q, \vec{f}\,]\,]_{\xi,\rho,\Delta}.$$

One should have this property in mind when reading the contents of this section. We cannot use it directly as the definition of interpretation for elimination terms for two reasons. First, the definition would be incomplete, we have to define $[\Gamma \vdash \text{Elim}(I, Q, m)\{\vec{f}\}]_{\xi,\rho,\Delta}$ for arbitrary object m of type I and not only for $m = \text{Constr}(n, I')\vec{N}$. Second, the definition would not be well founded. The term $f_n\vec{e}\,[C(X), \vec{N}, I, Q, \vec{f}\,]$ is not necessarily smaller than the term $\text{Elim}(I, Q, m)\{\vec{f}\}$. Moreover, if the constructor $C_n(I)$ has a recursive argument, say N_l, then in the right hand-side of the definition we would again refer to the value $[\Gamma \vdash \text{Elim}(I, Q, N_l)\{\vec{f}\}]_{\xi,\rho,\Delta}$, which at this point is not yet defined.

The exact definition of interpretation depends on whether the eliminated term is a large or a small inductive object. For small inductive objects we use an operator G':

$$[\Gamma \vdash \text{Elim}(I, Q, m)\{\vec{f}\}]_{\xi,\rho,\Delta} = G'_{\Gamma,Q,\vec{f},I,\xi,\rho,\Delta}([\Gamma \vdash I]_{\xi,\rho,\Delta})(\Delta \vdash \rho(m), \emptyset).$$

For large inductive objects we use an operator G:

$$[\Gamma \vdash \mathrm{Elim}(I, Q, m)\{\vec{f}\}]_{\xi,\rho,\Delta} = G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}([\Gamma \vdash I]_{\xi,\rho,\Delta})(\Delta \vdash \rho(m), [\Gamma \vdash m]_{\xi,\rho,\Delta}).$$

Both operators take two arguments: an approximation $S \in \mathcal{D}_{\Gamma,I,\xi,\rho,\Delta}$ of the value $[\Gamma \vdash I]_{\xi,\rho,\Delta}$ and a pair $(\Delta' \vdash a, U) \in \overline{T}^{\Delta}_{\rho(I)}$. The sequent is a result of applying the object substitution ρ to the eliminated term. In case of the operator $G'_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}$ the value U is always equal to $\emptyset$ because the interpreted term is small. In case of the operator $G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}$ the value U is an interpretation of the eliminated term under $[- \vdash -]_{\xi,\rho,\Delta}$ operator. Apart from that difference, the operators G and G' work similarly. For simplicity, in the following we only give proofs for the slightly more complicated operator G. The properties and proofs for the operator G' are similar.

We want to define the interpretation $[\Gamma \vdash \mathrm{Elim}(I, Q, m)\{\vec{f}\}]_{\xi,\rho,\Delta}$. The important case is when $\rho(m) =_{\beta\iota} \mathrm{Constr}(n, I')\vec{N}$. As already said, we want the interpretation to be equal to

$$[\Gamma \vdash f_n \vec{e}\,[C(X), \vec{N}, I, Q, \vec{f}\,]\,]_{\xi,\rho,\Delta}.$$

Observe that this interpretation is of the form

$$[\Gamma \vdash f_n]_{\xi,\rho,\Delta} \cdot \vec{g} \tag{5.1}$$

where $\vec{g}$ is a certain sequence of arguments. In the actual definition of the interpretation $[\Gamma \vdash \mathrm{Elim}(I, Q, m)\{\vec{f}\}]_{\xi,\rho,\Delta}$ we use the notation

$$g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[M, U, C(X), \vec{N}] \quad \text{or} \quad g'_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[M, C(X), \vec{N}]$$

to denote the sequence $\vec{g}$. We explain the exact meaning of the parameters later. We need two different notions because the sequences are slightly different if we interpret large or small inductive type. The intended use of the sequence is to define the operator G' for arguments m such that $m =_{\beta\iota} \mathrm{Constr}(n, I)\vec{N}$ as

$$G'_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta' \vdash m, U) = [\Gamma \vdash f_n] \cdot g'_{\Gamma,I,Q,\xi,\rho,\vec{f},\Delta'}[\mathrm{Constr}(n, I)\vec{N}, C_n(X), \vec{N}].$$

The operator G will be defined as

$$G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta' \vdash m, U) = [\Gamma \vdash f_n] \cdot g_{\Gamma,I,Q,\xi,\rho,\vec{f},\Delta'}[\mathrm{Constr}(n, I)\vec{N}, U, C_n(X), \vec{N}].$$

Let us first see some examples of what a sequence $\vec{g}$ in (5.1) looks like. The simplest case is an inductive type with only small, non-recursive arguments. Suppose the types T and S are small and OR is the disjoint union of T and S:

$$OR = \mathrm{Ind}(X : *^t)\{T \to X \mid S \to X\}.$$

Consider an inductive object $m = \mathrm{Constr}(0, OR)N$ and its elimination $\mathrm{Elim}(OR, Q, m)\{f_0 \mid f_1\}$. Then we want

$$[\Gamma \vdash \mathrm{Elim}(OR, Q, m)\{f_0 \mid f_1\}]_{\xi,\rho,\Delta} = [\Gamma \vdash f_0 N]_{\xi,\rho,\Delta} = [\Gamma \vdash f_0]_{\xi,\rho,\Delta} \underbrace{(\Delta \vdash \rho(N), \emptyset)}_{\vec{g}},$$

where the sequence of arguments $\vec{g}$ is as in (5.1). A more complicated example is the type of natural numbers:

$$Nat = \mathrm{Ind}(X : *^t)\{X \mid X \to X\}.$$

We focus our attention on the second constructor because it has a recursive argument. Consider an inductive object $m = \mathrm{Constr}(1, Nat)N$ and its elimination $\mathrm{Elim}(Nat, Q, m)\{f_0 \mid f_1\}$. The interpretation of branch f_1 is a function which takes an interpretation of the argument and then the interpretation of the recursive call on this argument, $R = \mathrm{Elim}(Nat, Q, N)\{f_0 \mid f_1\}$. We want

$$\begin{aligned}[\Gamma \vdash \mathrm{Elim}(Nat, Q, m)\{f_0 \mid f_1\}]_{\xi,\rho,\Delta} &= [\Gamma \vdash f_1 N R]_{\xi,\rho,\Delta} \\ &= [\Gamma \vdash f_1]_{\xi,\rho,\Delta} \underbrace{(\Delta \vdash \rho(N), \emptyset)(\Delta \vdash \rho(R), [\Gamma \vdash R]_{\xi,\rho,\Delta})}_{\vec{g}},\end{aligned}$$

where

$$[\Gamma \vdash R]_{\xi,\rho,\Delta} = [\Gamma \vdash \mathrm{Elim}(Nat, Q, N)\{f_0 \mid f_1\}]_{\xi,(\rho;x:=a),\Delta'}.$$

Finally, let us see the most complicated case, with argument which is both recursive and functional:

$$Tree = \mathrm{Ind}(X : *^t)\{X \mid (Nat \to X) \to X\}.$$

Let us consider an elimination term $\mathrm{Elim}(Tree, Q, m)\{f_0 \mid f_1\}$ with an inductive object $m = \mathrm{Constr}(1, Tree)N$. Then the recursive call is $R = (\lambda x : Nat.\mathrm{Elim}(Tree, Q, Nx)\{f_0 \mid f_1\})$ and

$$\begin{aligned}[\Gamma \vdash \mathrm{Elim}(Tree, Q, m)\{f_0 \mid f_1\}]_{\xi,\rho,\Delta} &= [\Gamma \vdash f_1 N R]_{\xi,\rho,\Delta} \\ &= [\Gamma \vdash f_1]_{\xi,\rho,\Delta} \underbrace{(\Delta \vdash \rho(N), \emptyset)(\Delta \vdash \rho(R), [\Gamma \vdash R]_{\xi,\rho,\Delta})}_{\vec{g}}\end{aligned}$$

and the interpretation of R is a function which applies the interpretation operator for an elimination term as follows:

$$[\Gamma \vdash R]_{\xi,\rho,\Delta} = \lambda\!\!\lambda(\Delta' \vdash a, C) : \overline{T}^{\Delta}_{Nat}.[\Gamma \vdash \mathrm{Elim}(Tree, Q, Nx)\{f_0 \mid f_1\})]_{(\xi;x:=C),(\rho;x:=a),\Delta'}.$$

We see that if $C_n(I)$ has no recursive arguments then $\vec{g}$ is simply the sequence of evaluations of $\vec{N}$: under object substitution ρ only (if N is small) and under both object substitution ρ and the operator $[- \vdash -]_{-,-,-}$ (if N is large). In fact, in this situation we could have defined the interpretation as $[\Gamma \vdash f_n \vec{e}\,[C(X), \vec{N}, I, Q, \vec{f}]\,]_{\xi,\rho,\Delta}$. If $C_n(I)$ has a recursive argument N_i then in $\vec{g}$ we have an evaluation for the recursive call of the elimination operation on N_i. We will apply the operator G (G') to compute it. If the argument N_i is both recursive and functional (i.e. $\Gamma \vdash N_i : (\Pi\vec{x} : \vec{t}.I)$) then its interpretation is also a function.

As already mentioned, in the definition of $[\Gamma \vdash \mathrm{Elim}(I, Q, m)\{\vec{f}\}]_{\xi,\rho,\Delta}$ we use the notation

$$g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[M, U, C(X), \vec{N}] \quad \text{or} \quad g'_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[M, C(X), \vec{N}]$$

to denote the sequence $\vec{g}$. In the notation M is a term and $\vec{N}$ is a sequence of terms. We should think that $M = \rho(m) = \mathrm{Constr}(n, X)\vec{N}$. Then $C(X)$ is a type of the n-th constructor of I. Finally, U is the interpretation of m.

Consider the examples given above. For the type OR we have

$$\rho(m) = \text{Constr}(0, OR)\rho(N),$$
$$\vec{g} = g'_{\Gamma,OR,Q,\vec{f},\xi,\rho,\Delta}[\rho(m), \underbrace{T \to OR}_{C_0(OR)}, \rho(N)] = (\Delta \vdash \rho(N), \emptyset).$$

For the type Nat of natural numbers we have

$$\rho(m) = \text{Constr}(1, Nat)\rho(N),$$
$$\vec{g} = g'_{\Gamma,Nat,Q,\vec{f},\xi,\rho,\Delta}[\rho(m), \underbrace{Nat \to Nat}_{C_1(Nat)}, \rho(N)] = (\Delta \vdash \rho(N), \emptyset), (\Delta \vdash \text{Elim}(I, Q, \rho(N))\{\vec{f}\}, R)$$

where R is an interpretation for the recursive call $\text{Elim}(I, Q, \rho(N))\{\vec{f}\}$. We take

$$R = G'_{\Gamma,Nat,Q,\vec{f},\xi,\rho,\Delta}(pred_{\Gamma,I,\xi,\rho,\Delta}(\Delta \vdash \rho(m)))(\Delta \vdash \rho(N))$$

Observe that the operator G' is used to obtain the interpretation instead of the operator $[- \vdash -]_{\xi,\rho,\Delta}$. Why we choose $pred_{\Gamma,I,\xi,\rho,\Delta}(\Delta \vdash \rho(m))$ as its first argument will be explained later.

For the type $Tree$ of trees we have

$$\rho(m) = \text{Constr}(1, Tree)\rho(N),$$
$$\vec{g} = g'_{\Gamma,Tree,Q,\vec{f},\xi,\rho,\Delta}[\rho(m), \underbrace{(Nat \to Tree) \to Tree}_{C_1(Tree)}, \rho(N)]$$
$$= (\Delta \vdash \rho(N), \emptyset), (\Delta \vdash \lambda x : Nat.\text{Elim}(I, Q, \rho(N)x)\{\vec{f}\}, R)$$

where R is an interpretation of the recursive call $(\lambda x : Nat.\text{Elim}(I, Q, \rho(N)x)\{\vec{f}\})$. We take

$$R = \lambda\!\!\lambda(\Delta' \vdash a', C) : \overline{T}^{\Delta}_{Nat}.G'_{(\Gamma,x:Nat),Tree,Q,\vec{f},\xi,(\rho;x:=a'),\Delta}$$
$$\underbrace{(pred_{(\Gamma,x:Nat),I,(\xi;x:=C),(\rho;x:=a'),\Delta}(\Delta \vdash \rho(m)))}_{\text{approximation of } [\Gamma \vdash Tree]_{\xi,\rho,\Delta}} \underbrace{(\Delta' \vdash \rho(N)a', \emptyset)}_{\text{recursive argument}}.$$

Observe how we chose the arguments for the operator G'.

We will now define the sequence $g'_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[M, C(X), \vec{N}]$ of arguments for the interpretation $[\Gamma \vdash f]_{\xi,\rho,\Delta}$ of the branch f. The sequence is defined by simultaneous induction with respect to the definition of the operator G' (see page 71). Recall that the type I is a small inductive type so every type of constructor of I is small, the sequence of arguments $\vec{N}$ consists of small objects and an interpretation of every N_i is equal to $\emptyset$.

- If $C(X) = X$ then
$$g'_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[M, C(X), \vec{N}] = \epsilon$$
where ϵ is an empty sequence,

- If $C(X) = \Pi x : T.D(X)$ and X does not occur in T, and T is a small type, and $\vec{N} = N_0 :: \vec{N}'$ then
$$g'_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[M, C(X), \vec{N}] = (\Delta \vdash N_0, \emptyset) :: g'_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[M, D(X), \vec{N}'].$$

- If $C(X) = \Pi x : T.D(X)$ and $T = \Pi \vec{y} : \vec{t}.X$, and $\vec{N} = N_0 :: \vec{N}'$ then

$$g'_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[M, C(X), \vec{N}] = (\Delta \vdash N_0, \emptyset) :: (\Delta \vdash e, R) :: g'_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[M, D(X), \vec{N}']$$

with

$$e = \lambda \vec{x} : \rho(\vec{t}).\text{Elim}(\rho(I), \rho(Q), N_0\vec{x})\{\rho(\vec{f})\}$$

and R is a function which for an appropriate sequence of arguments $(\Delta_i, a_i, C_i)_{i=1}^k$ for $(\vec{x} : \rho(\vec{t}))$ at $\langle \xi, \rho \rangle$ in Δ is defined as follows

$$R((\Delta_i, a_i, C_i)_{i=1}^k) = G'_{(\Gamma,\vec{x}:\vec{t}),I,Q,\vec{f},\xi_k,\rho_k,\Delta_k}(pred_{(\Gamma,\vec{x}:\vec{t}),I,\xi_k,\rho_k,\Delta_k}(\Delta_k \vdash M))(\Delta_k \vdash N_0\vec{a}, \emptyset).$$

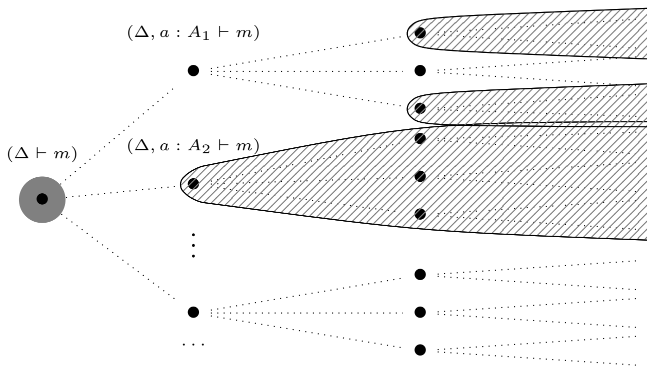

Figure 5.2: Simple sequents and the interpretation of an inductive type (incorrect)

We would like to define the operator G' so that if $m =_{\beta\iota} \text{Constr}(n, I)\vec{N}$ and $(\Delta' \vdash m) \in S$ then

$$G'_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta' \vdash m) = [\Gamma \vdash f_n]_{\xi,\rho,\Delta} \cdot g'_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta'}[\text{Constr}(n, I)\vec{N}, C_n(I), \vec{N}] \tag{5.2}$$

and in all other cases we have

$$G'_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta' \vdash m) = Min_{\Delta'}(\text{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\}.$$

However, there is one problem with this definition. It does not satisfy the property that for $\Delta \subseteq \Delta'$ we have

$$([\Gamma \vdash \text{Elim}(I, Q, m)\{\vec{f}\}]_{\xi,\rho,\Delta})|_{\Delta'} = [\Gamma \vdash \text{Elim}(I, Q, m)\{\vec{f}\}]_{\xi,\rho,\Delta'}. \tag{5.3}$$

Suppose $m = \text{Constr}(n, I)\vec{N}$ and $(\Delta' \vdash m) \in [\Gamma \vdash I]_{\xi,\rho,\Delta'}$. Then

$$\begin{aligned}[\Gamma \vdash \text{Elim}(I, Q, m)\{\vec{f}\}]_{\xi,\rho,\Delta'} &= G'_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}([\Gamma \vdash I]_{\xi,\rho,\Delta'})(\Delta' \vdash m)\\ &= [\Gamma \vdash f_n]_{\xi,\rho,\Delta} \cdot g'_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta'}[\text{Constr}(n, I)\vec{N}, C_n(I), \vec{N}].\end{aligned}$$

However, we cannot prove that $(\Delta \vdash m) \in [\Gamma \vdash I]_{\xi,\rho,\Delta}$. It is possible that

$$\begin{aligned}([\Gamma \vdash \text{Elim}(I, Q, m)\{\vec{f}\}]_{\xi,\rho,\Delta})|_{\Delta'} &= (Min_{\Delta'}(\text{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\})|_{\Delta'}\\ &= Min_{\Delta'}(\text{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\}.\end{aligned}$$

The two sets are not equal. This behaviour of the operator G' is illustrated in Figure 5.2. Suppose we want to compute the value $G'_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta \vdash m)$. The picture shows the set of simple sequents of the form $(\Delta' \vdash m)$ where $\Delta' \supseteq \Delta$. The sequents are naturally ordered by the relation $\subseteq$ on contexts, this order is represented in the picture by dotted lines. The dashed area is the set S. It is a saturated set so it satisfies the property that if $(\Delta' \vdash m)$ is in the set then so is every $(\Delta'' \vdash m)$ for $\Delta'' \supseteq \Delta'$. Thus a dashed area is a union of "cones" generated by some sequents of the form $(\Delta' \vdash m)$ (i.e. sets of sequents $(\Delta'' \vdash m)$ such that $\Delta'' \supseteq \Delta'$). In the above definition (5.2) the value $G'_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta \vdash m)$ depends only on whether the sequent $(\Delta \vdash m)$ is in the set S or not. In the picture this is the question whether a gray circle is inside the dashed area or not. In order to satisfy property (5.3) the

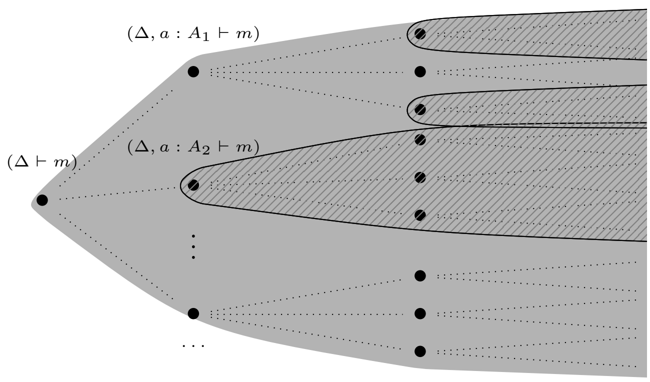

Figure 5.3: Simple sequents and the interpretation of an inductive type (correct)

value $G'_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta \vdash m)$ should depend on the relation between the set S and the whole "cone" generated by $(\Delta \vdash m)$ as illustrated in Figure 5.3.

We have to change a little the definition of interpretation. We will make sure that the interpretation changes smoothly as the context grows. This is why we need the notion of a *smooth union* of a set.

The definition of interpretation for elimination terms

We may now give the definition of interpretation for elimination of a small inductive type. Recall the abbreviation $g'_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[M, C(I), \vec{N}]$ which was introduced on page 69. This definition depends on the operator $G'_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}$ to be defined below. Recall that the operator takes two arguments: an approximation $S \in \mathcal{D}_{\Gamma,I,\xi,\rho,\Delta}$ and a pair $(\Delta' \vdash m, U) \in \overline{T}^{\Delta}_{\rho(I)}$. The value

$$G'_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta' \vdash m, U)$$

is defined as follows.

- If $m =_{\beta\iota} \mathrm{Constr}(j, J)\vec{N}$ and there exists Δ'' such that

$$\Delta'' \supseteq \Delta' \text{ and } (\Delta'' \vdash \mathrm{Constr}(j, J)\vec{N}) \in S$$

  then

$$G'_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta' \vdash m, U) = \bigsqcup(Base \cup Min)$$

where

- *Base* consists of all values of the form

$$[\Gamma \vdash f_j]_{\xi|_{\Delta''},\rho,\Delta''} \cdot g'_{\Gamma,I,Q,\vec{f},\xi|_{\Delta''},\rho,\Delta''}[\mathrm{Constr}(j,J)\vec{M}, C_j(I), \vec{M}]$$

  such that $\Delta'' \supseteq \Delta'$, $m =_{\beta\iota} \mathrm{Constr}(j,J)\vec{M}$ and $(\Delta'' \vdash \mathrm{Constr}(j,J)\vec{M}) \in S$.

- *Min* consists of all values of the form

$$Min_{\Delta''}(\mathrm{Elim}(\rho(I),\rho(Q),m)\{\rho(\vec{f})\})$$

  such that $\Delta'' \supseteq \Delta'$, $m =_{\beta\iota} \mathrm{Constr}(j,J)\vec{M}$ and $(\Delta'' \vdash \mathrm{Constr}(j,J)\vec{M}) \in T^{\Delta}_{\rho(I)} - S$.

- Otherwise

$$G'_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta' \vdash m, U) = Min_{\Delta'}(\mathrm{Elim}(\rho(I),\rho(Q),m)\{\rho(\vec{f})\}.$$

The definition of the operator $G'_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}$ is sound. We will later prove that the set $(Base \cup Min)$ is consistent. In every recursive call to the operator (compare the definition of the sequence g') the first argument is smaller than the argument for which we are defining the value. Thus the operator is in fact defined by induction with respect to the ordering in $\mathcal{D}_{\Gamma,I,\xi,\rho,\Delta}$.

If I is a small inductive type then we define $[\Gamma \vdash \mathrm{Elim}(I,Q,M)\{\vec{f}\}]_{\xi,\rho,\Delta}$ in the following way

$$[\Gamma \vdash \mathrm{Elim}(I,Q,M)\{\vec{f}\}]_{\xi,\rho,\Delta} = G'_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}([\Gamma \vdash I]_{\xi,\rho,\Delta})(\Delta \vdash \rho(M), \emptyset).$$

The definition of interpretation for elimination for large inductive types

If I is a large inductive type the definition has to be adjusted accordingly. The only thing that really changes is the definition of the sequence $\vec{g'}$. We take into account the fact that a large inductive object has its own set interpretation which has to be passed to the interpretation of the branch.

Let us first see an example. Consider the type

$$List = \mathrm{Ind}(X : *^t)\{X \mid (T \to *^t) \to X \to X\}.$$

Suppose we want to find the interpretation for the term $\mathrm{Elim}(List, Q, \mathrm{cons}\ A\ \mathrm{nil})\{\vec{f}\}$. Observe that

$$\mathrm{Elim}(List, Q, \mathrm{cons}\ A\ \mathrm{nil})\{\vec{f}\} \to_\iota f_1\ A\ \mathrm{Elim}(List, Q, \mathrm{nil})\{\vec{f}\}.$$

Suppose that $\rho(\mathrm{cons}\ A\ \mathrm{nil}) = \mathrm{cons}\ A'\ \mathrm{nil}$ and The interpretation of the term $(\mathrm{cons}\ A\ \mathrm{nil})$ is a triple $U = \langle 1, a, b\rangle$. The sequence of arguments is equal to $A' :: \mathrm{nil}$. Then

$$\begin{aligned}
&g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta''}[\mathrm{cons}\ A'\ \mathrm{nil}, \langle 1,a,b\rangle, C_j(I), A' :: \mathrm{nil}] = \\
&\qquad (\Delta \vdash A', a), (\Delta \vdash \mathrm{nil}, b), \\
&\qquad (\Delta \vdash \mathrm{Elim}(I,Q,\mathrm{nil})\{\vec{f}\}, G_{\Gamma,List,Q,\vec{f},\xi,\rho,\Delta}(pred_{\Gamma,List,\xi,\rho,\Delta}(\Delta \vdash \mathrm{cons}\ A\ \mathrm{nil}))(\Delta \vdash \mathrm{nil}, b)).
\end{aligned}$$

The interpretation of an argument is an appropriate projection of the interpretation $U = \langle m, \vec{U}\rangle$ of the eliminated term. We define

$$g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[M, \langle m, \vec{U}\rangle, C(X), \vec{N}]$$

by simultaneous induction with the definition of the operator $G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}$ as follows.

- If $C(X) = X$ then
$$g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[M, U, C(X), \vec{N}] = \epsilon$$
where ϵ is an empty sequence.

- If $C(X) = \Pi x : T.D(X)$, $\vec{N} = N_0 :: \vec{N'}$, $U = \langle m, \vec{U}\rangle$ and $\vec{U} = U_0 :: \vec{U'}$, $X \notin FV(T)$ and T is a large type then
$$g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[M, U, C(X), \vec{N}] = (\Delta \vdash N_0, U_0) :: g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[M, \langle m, \vec{U'}\rangle, D(X), \vec{N'}].$$

- If $C(X) = \Pi x : T.D(X)$, $\vec{N} = N_0 :: \vec{N'}$ and $X \in FV(T)$ and $T = \Pi \vec{x} : \vec{t}.I$ then
$$g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[M, \vec{U}, C(X), \vec{N}] = (\Delta \vdash N_0, U_0) :: (\Delta \vdash e, R) :: g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[M, \langle m, \vec{U'}\rangle, D(X), \vec{N'}]$$
with
$$e = \lambda \vec{x} : \rho(\vec{t}).\mathrm{Elim}(\rho(I), \rho(Q), N_0\vec{x})\{\rho(\vec{f})\},$$
and R is a function which for an appropriate sequence of arguments $(\Delta_i, a_i, C_i)_{i=1}^k$ for $(\vec{x} : \vec{t})$ at $\langle \xi, \rho\rangle$ in Δ is defined as follows
$$R((\Delta_i, a_i, C_i)_{i=1}^k) = G_{(\Gamma,\vec{x}:\vec{t}),I,Q,\vec{f},\xi_k,\rho_k,\Delta_k}(pred_{(\Gamma,\vec{x}:\vec{t}),I,\xi_k,\rho_k,\Delta_k}(\Delta_k \vdash M)) (\Delta_k \vdash N_0\vec{x},\ U_0(\Delta_i, a_i, C_i)_{i=1}^k).$$

We will define the interpretation using the operator $G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}$. Recall that the operator takes two arguments: an approximation $S \in \mathcal{D}_{\Gamma,I,\xi,\rho,\Delta}$ and a pair $(\Delta' \vdash m, U) \in \overline{T}^{\Delta}_{\rho(I)}$. The value
$$G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta' \vdash m, U)$$
is defined as follows

- If $\pi_1(U) = j$ and there exists J, $\vec{N}$, $\Delta'' \supseteq \Delta'$ such that $m =_{\beta\iota} \mathrm{Constr}(j, J)\vec{N}$ and $(\Delta' \vdash \mathrm{Constr}(j, J)\vec{N}) \in S$ then
$$G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta' \vdash m, U) = \bigsqcup(Base \cup Min)$$
where
  - *Base* consists of all values of the form
$$[\Gamma \vdash f_j]_{\xi|_{\Delta'},\rho,\Delta''} \cdot g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta''}[\mathrm{Constr}(j, J)\vec{M}, U, C_j(I), \vec{M}]$$
such that $\Delta'' \supseteq \Delta'$, $m =_{\beta\iota} \mathrm{Constr}(j, J)\vec{M}$ and $(\Delta'' \vdash \mathrm{Constr}(j, J)\vec{M}) \in S$

– Min consists of all values of the form

$$Min_{\Delta''}(\mathrm{Elim}(\rho(I),\rho(Q),m)\{\rho(\vec{f})\})$$

such that $\Delta'' \supseteq \Delta'$, $m =_{\beta\iota} \mathrm{Constr}(j,J)\vec{M}$ and

$$(\Delta'' \vdash \mathrm{Constr}(j,J)\vec{M}) \in T^{\Delta}_{\rho(I)} - S.$$

- Otherwise

$$G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta' \vdash m, U) = Min_{\Delta'}(\mathrm{Elim}(\rho(I),\rho(Q),m)\{\rho(\vec{f})\}).$$

If I is a large inductive type then we define:

$$[\Gamma \vdash \mathrm{Elim}(I,Q,M)\{\vec{f}\}]_{\xi,\rho,\Delta} = G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}([\Gamma \vdash I]_{\xi,\rho,\Delta})(\Delta \vdash \rho(M), [\Gamma \vdash M]_{\xi,\rho,\Delta}).$$

In the following the abbreviation

$$g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[M,U,C_n(I),(N_i)_{i=0}^{k-1}],$$

will denote the subsequence of $g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[M,U,C_n(I),\vec{N}]$ associated with the first k elements of $\vec{N}$. Observe that $g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[M,U,C_n(I),(N_i)_{i=0}^{k-1}]$ is a sequence of pairs $(\Delta_i \vdash a_i, C_i)_{i=0}^{k-1}$. We will use the notation

$$g^1_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[M,U,C_n(I),(N_i)_{i=0}^{k-1}]$$

for the sequence consisting of terms $(a_i)_{i=0}^{k-1}$.

We want to prove that the definition of interpretation is correct, i.e.

$$[\Gamma \vdash \mathrm{Elim}(I,Q,m)\{\vec{f}\}]_{\xi,\rho,\Delta} \in V_{\Delta}(\rho(\mathrm{Elim}(I,Q,m)\{\vec{f}\})).$$

We will first prove that every element of the set $Base \cup Min$ is in $V_{\Delta'}(\rho(\mathrm{Elim}(I,Q,m)\{\vec{f}\}))$ for an appropriate Δ'. In particular, we will show it for elements of the form

$$[\Gamma \vdash f_n]_{\xi,\rho,\Delta} \cdot g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[\mathrm{Constr}(n,J)\vec{N},U,C_n(I),\vec{N}].$$

For this we need an auxiliary fact: that every application of the form

$$[\Gamma \vdash f_n]_{\xi,\rho,\Delta} \cdot g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[\mathrm{Constr}(n,J)\vec{N},U,C_n(I),(N_i)_{i=0}^{k-1}]$$

is correct.

Lemma 64. *Suppose*

1. $\Gamma \vdash I : *^t$ *and* I *is a large inductive type,* $C_n(I) = \Pi\vec{x}:\vec{T}.I$*;*
2. $\Gamma \vdash f_n : \triangle\{C_n(I),Q,\mathrm{Constr}(n,I)\}$ *and* $[\Gamma \vdash f_n]_{\xi,\rho,\Delta} \in V_{\Delta}(\rho(f_n))$*,*
3. $\mathrm{Constr}(n,J)\vec{N} =_{\beta\iota} m$ *for* m *such that* $\Delta \vdash m : \rho(I)$*;*
4. $U \in V_{\Delta}(\mathrm{Constr}(n,J)\vec{N})$*;*

5. *for every context* $\Gamma' \supseteq \Gamma$*, for every constructor valuation* $\langle \xi', \rho' \rangle$ *which satisfies* Γ' *at* Δ'*, if* $S = pred_{\Gamma', I, \xi', \rho', \Delta'}(\Delta' \vdash \mathrm{Constr}(n, J)\vec{N})$ *then for every term* u *such that* $\Delta'' \vdash u : \rho(I)$ *for every* $C \in V_{\Delta''}(u)$ *we have*

$$G_{\Gamma', I, Q, \vec{f}, \xi', \rho', \Delta'}(S)(\Delta'' \vdash u, C) \in V_{\Delta''}(\mathrm{Elim}(\rho'(I), \rho'(Q), u)\{\rho'(\vec{f})\});$$

6. k *is a natural number at most equal to the length of the sequence* $\vec{N}$*.*

Then it holds that

$$[\Gamma \vdash f_n]_{\xi,\rho,\Delta} \cdot g_{\Gamma, I, Q, \vec{f}, \xi, \rho, \Delta}[\mathrm{Constr}(n, J)\vec{N}, U, C_n(I), (N_i)_{i=0}^{k-1}] \\ \in V_\Delta(\rho(f_n) \cdot g^1_{\Gamma, I, Q, \vec{f}, \xi, \rho, \Delta}[\mathrm{Constr}(n, J)\vec{N}, U, C_n(I), (N_i)_{i=0}^{k-1}]).$$

Proof. We proceed by induction with respect to k.

If $k = 0$ then the sequence $g_{\Gamma, I, Q, \vec{f}, \xi, \rho, \Delta}[\mathrm{Constr}(n, J)\vec{N}, U, C_n(I), (N_i)_{i=0}^{k-1}]$ is empty and the conclusion follows from the assumption.

Suppose $k > 0$ and the conclusion holds for every $k' < k$. We proceed by cases depending on N_{k-1}. We only consider the more complex case: when N_{k-1} is a recursive argument. The case for non-recursive argument is similar but simpler. Then N_{k-1} is of type $\Pi\vec{x} : \vec{\tau}.I$. By the induction hypothesis

$$[\Gamma \vdash f_n]_{\xi,\rho,\Delta} \cdot g_{\Gamma, I, Q, \vec{f}, \xi, \rho, \Delta}[\mathrm{Constr}(n, J)\vec{N}, U, C_n(I), (N_i)_{i=0}^{k-2}] \\ \in V_\Delta(\rho(f_n) \cdot g^1_{\Gamma, I, Q, \vec{f}, \xi, \rho, \Delta}[\mathrm{Constr}(n, J)\vec{N}, U, C_n(I), (N_i)_{i=0}^{k-2}]).$$

By the definition of $\triangle\{C_n(I), Q, \mathrm{Constr}(n, I)\}$ and the definition of $V_\Delta(\rho(f_n))$ we know that

$$[\Gamma \vdash f_n]_{\xi,\rho,\Delta} \cdot g_{\Gamma, I, Q, \vec{f}, \xi, \rho, \Delta}[\mathrm{Constr}(n, J)\vec{N}, U, C_n(I), (N_i)_{i=0}^{k-2}]$$

is a function with the domain

$$\{(\Delta' \vdash M, C) \mid \Delta' \supseteq \Delta, \Delta' \vdash M : (\Pi\vec{x} : \rho(\vec{\tau}).\rho(I)), C \in V_{\Delta'}(M)\}$$

such that

$$[\Gamma \vdash f_n]_{\xi,\rho,\Delta} \cdot g_{\Gamma, I, Q, \vec{f}, \xi, \rho, \Delta}[\mathrm{Constr}(n, J)\vec{N}, U, C_n(I), (N_i)_{i=0}^{k-2}](\Delta' \vdash M, C) \\ \in V_\Delta(\rho(f_n) \cdot g^1_{\Gamma, I, Q, \vec{f}, \xi, \rho, \Delta}[\mathrm{Constr}(n, J)\vec{N}, U, C_n(I), (N_i)_{i=0}^{k-2}]M).$$

Observe that $\Delta \vdash N_{k-1} : \rho(\Pi\vec{x} : \vec{\tau}.I)$, $\Delta \subseteq \Delta$, $U_{k-1} \in V_\Delta(N_{k-1})$. Thus

$$[\Gamma \vdash f_n]_{\xi,\rho,\Delta} \cdot g_{\Gamma, I, Q, \vec{f}, \xi, \rho, \Delta}[\mathrm{Constr}(n, J)\vec{N}, U, C_n(I), (N_i)_{i=0}^{k-2}](\Delta \vdash N_{k-1}, U_{k-1}) \\ \in V_\Delta(\rho(f_n) \cdot g^1_{\Gamma, I, Q, \vec{f}, \xi, \rho, \Delta}[\mathrm{Constr}(n, J)\vec{N}, U, C_n(I), (N_i)_{i=0}^{k-2}]N_{k-1}).$$

By the definition of $V_\Delta(\rho(f_n))$

$$[\Gamma \vdash f_n]_{\xi,\rho,\Delta} \cdot g_{\Gamma, I, Q, \vec{f}, \xi, \rho, \Delta}[\mathrm{Constr}(n, J)\vec{N}, U, C_n(I), (N_i)_{i=0}^{k-2}](\Delta \vdash N_{k-1}, U_{k-1})$$

is a function with the domain

$$\{(\Delta' \vdash M, C) \mid \Delta' \supseteq \Delta, \Delta' \vdash M : (\Pi\vec{x} : \rho(\vec{\tau}).\rho(Q)(N_{k-1}\vec{x})), C \in V_{\Delta'}(M)\}$$

such that

$$[\Gamma \vdash f_n]_{\xi,\rho,\Delta} \cdot g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[\mathrm{Constr}(n,J)\vec{N}, U, C_n(I), (N_i)_{i=0}^{k-2}](\Delta \vdash N_{k-1}, U_{k-1})(\Delta' \vdash M, C)$$
$$\in V_\Delta(\rho(f_n) \cdot g^1_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[\mathrm{Constr}(n,J)\vec{N}, U, C_n(I), (N_i)_{i=0}^{k-2}]N_{k-1}M).$$

By assumption that N_{k-1} is of type $\Pi\vec{x} : \vec{\tau}.I$ we get

$$\Delta \vdash \lambda\vec{x} : \rho(\vec{\tau}).\mathrm{Elim}(\rho(I), \rho(Q), N_{k-1}\vec{x})\{\rho(\vec{f})\} : (\Pi\vec{x} : \rho(\vec{\tau}).\rho(Q)(N_{k-1}\vec{x})).$$

Suppose C is a function which for any sequence of arguments $(\Delta_i, a_i, C_i)_{i=0}^p$ appropriate for $(\vec{x} : \vec{\tau})$ at $\langle \xi, \rho \rangle$ in Δ_p is defined as follows:

$$C((\Delta_i, a_i, C_i)_i) = G_{(\Gamma,\vec{x}:\rho(\vec{\tau})),I,Q,\vec{f},\xi_p,\rho_p,\Delta_p}(pred_{(\Gamma,\vec{x}:\vec{\tau}),I,\xi_p,\rho_p,\Delta_p}(\Delta_p \vdash \mathrm{Constr}(n,J)\vec{N}))$$
$$(\Delta \vdash N_{k-1}\vec{a}, U_{k-1}(\Delta_i, a_i, C_i)_{i=0}^p).$$

Then by the assumption 5

$$C \in V_\Delta(\lambda\vec{x} : \rho(\vec{\tau}).\mathrm{Elim}(\rho(I), \rho(Q), N_{k-1}\vec{x})\{\rho(\vec{f})\}).$$

Thus

$$[\Gamma \vdash f_n]_{\xi,\rho,\Delta} \cdot g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[\mathrm{Constr}(n,J)\vec{N}, U, C_n(I), (N_i)_{i=0}^{k-1}]$$
$$\in V_\Delta(\rho(f_n) \cdot g^1_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[\mathrm{Constr}(n,J)\vec{N}, U, C_n(I), (N_i)_{i=0}^{k-1}]). \quad \square$$

Lemma 65. *The set* $(Base \cup Min)$ *is consistent for* $\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\}$ *at* Δ.

Proof. Recall that the set is consistent for A at Δ if for each $h \in \mathcal{F}$ there exists $\Delta' \supseteq \Delta$ such that $h \in V_{\Delta'}(A)$ and there exists $h' \in \mathcal{F}$ such that $h' \in V_\Delta(A)$.

Let $h \in (Base \cup Min)$. If $h \in Base$ then by Lemma 64

$$h \in V_{\Delta'}(\rho(f_n) \cdot g^1_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[\mathrm{Constr}(n,J)\vec{N}, U, C_n(I), \vec{N}]) = V_{\Delta'}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\}).$$

If $h \in Min$ then by Lemma 49 we have $h \in V_{\Delta'}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\})$. In both cases $h \in V_{\Delta'}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\})$.

Now we prove that there exists $h' \in (Base \cup Min)$ such that

$$h' \in V_{\Delta'}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\}).$$

Case 1: If $m =_{\beta\iota} \mathrm{Constr}(j,J)\vec{N}$ and $(\Delta' \vdash \mathrm{Constr}(j,J)\vec{N}) \in S$, take

$$h' = [\Gamma \vdash f_n]_{\xi,\rho,\Delta'} \cdot g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta'}[\mathrm{Constr}(n,J)\vec{N}, U, C_n(I), \vec{N}].$$

We know that $h' \in Base$ and by Lemma 64 we have

$$h' \in V_{\Delta'}(\rho(f_n) \cdot g^1_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[\mathrm{Constr}(n,J)\vec{N}, U, C_n(I), \vec{N}])$$
$$= V_{\Delta'}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\}).$$

Case 2: Otherwise take

$$h' = Min_{\Delta'}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\}).$$

Then $h' \in Min$ and by Lemma 49 we have $h' \in V_{\Delta'}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\})$. □

Lemma 66. *Let all assumptions of Lemma 64 hold. In addition assume that $S \in \mathcal{D}_{\Gamma,I,\xi,\rho,\Delta}$. Then*

$$G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta' \vdash m, U) \in V_{\Delta'}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\}).$$

Proof. If $S \in \mathcal{D}_{\Gamma,I,\xi,\rho,\Delta}$ then there exists α such that $S = F^{\alpha}_{\Gamma,I,\xi,\rho,\Delta}(\bigcap SAT^{\Delta}_{\rho(I)})$. We proceed by induction with respect to α. There are two cases.
Case 1: If $\pi_1(U) = n$ and there do not exist J, $\vec{N}$ and $\Delta'' \supseteq \Delta'$ such that

$$m =_{\beta\iota} \mathrm{Constr}(n, J)\vec{N} \quad \text{and} \quad (\Delta'' \vdash \mathrm{Constr}(n, J)\vec{N}) \in S$$

then

$$G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta' \vdash m, U) = Min_{\Delta'}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\}).$$

By Lemma 49

$$Min_{\Delta'}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\}) \in V_{\Delta'}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\})$$

Note that if $\alpha = 0$ this is the only case which is possible.

Case 2: If $\pi_1(U) = n$, $m =_{\beta\iota} \mathrm{Constr}(n, J)\vec{N}$ and there exists $\Delta'' \supseteq \Delta'$ such that

$$(\Delta'' \vdash \mathrm{Constr}(n, J)\vec{N}) \in S$$

then

$$G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta' \vdash m, U) = \bigsqcup \mathcal{F},$$

where $\mathcal{F} = Base \cup Min$ as in the definition of $G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}$. We will use Lemma 61 to prove that

$$\bigsqcup \mathcal{F} \in V_{\Delta'}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\}.$$

We have to show two things:

1. for every $f \in \mathcal{F}$ there exists Δ'' such that $f \in V_{\Delta''}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\}$,
2. there exists $f \in \mathcal{F}$ such that $f \in V_{\Delta'}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\}$.

We show the first property. If $f \in \mathcal{F}$ then there are two possibilities. The first one is $f = [\Gamma \vdash f_n]_{\xi,\rho,\Delta''} \cdot g_{\Gamma,I,Q,\vec{f},\xi|_{\Delta''},\rho,\Delta''}[\mathrm{Constr}(n, J)\vec{N}, U, C_n(I), \vec{N}]$. By Lemma 64 and the induction hypothesis we have

$$[\Gamma \vdash f_n]_{\xi,\rho,\Delta''} \cdot g_{\Gamma,I,Q,\vec{f},\xi|_{\Delta''},\rho,\Delta''}[\mathrm{Constr}(n, J)\vec{N}, U, C_n(I), \vec{N}] \\ \in V_{\Delta''}(\rho(f_n) \cdot g^1_{\Gamma,I,Q,\vec{f},\xi|_{\Delta''},\rho,\Delta''}[\mathrm{Constr}(n, J)\vec{N}, U, C_n(I), \vec{N}])).$$

But

$$\rho(f_n) \cdot g^1_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[\mathrm{Constr}(n, J)\vec{N}, U, C_n(I), \vec{N}] =_{\beta\iota} \mathrm{Elim}(\rho(I), \rho(Q), \mathrm{Constr}(n, J)\vec{N})\{\rho(\vec{f})\} \\ =_{\beta\iota} \mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\}$$

and by Lemma 41

$$V_{\Delta''}(\rho(f_n) \cdot g^1_{\Gamma,I,Q,\vec{f},\xi|_{\Delta''},\rho,\Delta''}[U, C_n(I), \vec{N}]) = V_{\Delta''}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\}).$$

Thus

$$f \in V_{\Delta}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\}).$$

The other possibility is that $f = Min_{\Delta''}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\})$. By Lemma 49 indeed

$$Min_{\Delta''}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\}) \in V_{\Delta''}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\}).$$

The second property is easy to observe. Either

$$Min_{\Delta'}(\mathrm{Elim}(\rho(I), \rho(Q), \mathrm{Constr}(n, J)\vec{N})\{\rho(\vec{f})\}) \in \mathcal{F}$$

or

$$[\Gamma \vdash f_n]_{\xi,\rho,\Delta'} \cdot g_{\Gamma,I,Q,\vec{f},\xi|_{\Delta'},\rho,\Delta'}[\mathrm{Constr}(n, J)\vec{N}, U, C_n(I), \vec{N}] \in \mathcal{F}.$$

Therefore by Lemma 61 we have proved

$$\bigsqcup \mathcal{F} \in V_{\Delta'}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\})$$

and thus

$$G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta' \vdash m, U) \in V_{\Delta'}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\}).$$

□

Lemma 67. *Suppose* Γ, *and* Δ *are contexts and* $\Gamma \subseteq \Gamma'$. *Suppose*

1. $\Gamma \vdash I : *^t$ *and* I *is a large inductive type;*
2. $\Gamma \vdash M : I$*;*
3. $\Gamma \vdash \mathrm{Elim}(I, Q, M)\{\vec{f}\} : QM$*;*
4. $\langle \xi, \rho \rangle$ *is a constructor valuation which satisfies* Γ *at* Δ.

In addition suppose that for each sequent $\Gamma' \vdash N' : T$ *in the derivation of*

$$\Gamma \vdash \mathrm{Elim}(I, Q, M)\{\vec{f}\} : QM$$

and for each constructor valuation $\langle \xi', \rho' \rangle$ *which satisfies* Γ' *at* Δ' *we have*

5. $[\Gamma' \vdash N']_{\xi',\rho',\Delta'} \in V_{\Delta'}(\rho'(T))$;
6. *if* $\Delta' \subseteq \Delta''$ *then* $([\Gamma' \vdash N']_{\xi',\rho',\Delta'})|_{\Delta''} = [\Gamma' \vdash N']_{\xi'|_{\Delta''},\rho',\Delta''}$*;*
7. *the value of* $[\Gamma' \vdash N']_{\xi',\rho',\Delta'}$ *depends only on the values of* ξ' *and* ρ' *for variables in* $FV(N')$.

Then

$$([\Gamma \vdash \mathrm{Elim}(I, Q, M)\{\vec{f}\}]_{\xi,\rho,\Delta})|_{\Delta'} = [\Gamma \vdash \mathrm{Elim}(I, Q, M)\{\vec{f}\}]_{\xi|_{\Delta'},\rho,\Delta'}.$$

Proof. Let $m = \rho(M)$. Then

$$([\Gamma \vdash \mathrm{Elim}(I, Q, M)\{\vec{f}\}]_{\xi,\rho,\Delta})|_{\Delta'} = (G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}([\Gamma \vdash I]_{\xi,\rho,\Delta})(\Delta \vdash m, [\Gamma \vdash M]_{\xi,\rho,\Delta}))|_{\Delta'}$$

and

$$[\Gamma \vdash \mathrm{Elim}(I, Q, M)\{\vec{f}\}]_{\xi|_{\Delta'},\rho,\Delta'} = G_{\Gamma,I,Q,\vec{f},\xi|_{\Delta'},\rho,\Delta'}([\Gamma \vdash I]_{\xi|_{\Delta'},\rho,\Delta'})(\Delta' \vdash m, [\Gamma \vdash M]_{\xi|_{\Delta'},\rho,\Delta'}).$$

There are two cases.
Case 1: $\pi_1([\Gamma \vdash M]_{\xi,\rho,\Delta}) = n$ and $m =_{\beta\iota} \mathrm{Constr}(j, J)\vec{N}$ and there exists $\Delta'' \supseteq \Delta'$ such that $(\Delta'' \vdash \mathrm{Constr}(j, J)\vec{N}) \in [\Gamma \vdash I]_{\xi,\rho,\Delta}$. Then

$$(G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}([\Gamma \vdash I]_{\xi,\rho,\Delta})(\Delta \vdash \rho(M), [\Gamma \vdash M]_{\xi,\rho,\Delta}))|_{\Delta'} = (\bigsqcup \mathcal{F}_1)|_{\Delta'}$$

where

$$\mathcal{F}_1 = (Base_1 \cup Min_1)$$

and

$$G_{\Gamma,I,Q,\vec{f},\xi|_{\Delta'},\rho,\Delta'}([\Gamma \vdash I]_{\xi|_{\Delta'},\rho,\Delta'})(\Delta' \vdash \rho(M), [\Gamma \vdash M]_{\xi|_{\Delta'},\rho,\Delta'}) = \bigsqcup \mathcal{F}_2$$

where $\mathcal{F}_2 = Base_2 \cup Min_2$. By the assumption

$$[\Gamma \vdash M]_{\xi|_{\Delta'},\rho,\Delta'} = ([\Gamma \vdash M]_{\xi,\rho,\Delta})|_{\Delta'}$$

and

$$[\Gamma \vdash I]_{\xi|_{\Delta'},\rho,\Delta'} = ([\Gamma \vdash I]_{\xi,\rho,\Delta})|_{\Delta'}.$$

Thus $\pi_1([\Gamma \vdash M]_{\xi|_{\Delta'},\rho,\Delta'}) = \pi_1(([\Gamma \vdash M]_{\xi,\rho,\Delta})|_{\Delta'}) = n$. Moreover $m =_{\beta\iota} \mathrm{Constr}(j, J)\vec{N}$ and there exists $\Delta'' \supseteq \Delta'$ such that

$$(\Delta'' \vdash \mathrm{Constr}(j, J)\vec{N}) \in [\Gamma \vdash I]_{\xi|_{\Delta'},\rho,\Delta'} = ([\Gamma \vdash I]_{\xi,\rho,\Delta})|_{\Delta'}.$$

It is easy to observe that $\mathcal{F}_1$, $\mathcal{F}_2$ satisfy the assumption of Lemma 63. Thus

$$(\bigsqcup \mathcal{F}_1)|_{\Delta'} = \bigsqcup \mathcal{F}_2.$$

Case 2: Otherwise $\pi_1([\Gamma \vdash M]_{\xi,\rho,\Delta}) \neq n$ or $m \neq_{\beta\iota} \mathrm{Constr}(j, J)\vec{N}$ or there does not exist $\Delta'' \supseteq \Delta'$ such that $(\Delta'' \vdash \mathrm{Constr}(j, J)\vec{N}) \in [\Gamma \vdash I]_{\xi,\rho,\Delta}$. Then

$$\begin{aligned}&(G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}([\Gamma \vdash I]_{\xi,\rho,\Delta})(\Delta \vdash m, [\Gamma \vdash M]_{\xi,\rho,\Delta}))|_{\Delta'} \\ &\qquad = (Min_{\Delta}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\}))|_{\Delta'} = Min_{\Delta'}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\}).\end{aligned}$$

But then as well $\pi_1([\Gamma \vdash M]_{\xi|_{\Delta'},\rho,\Delta'}) \neq n$ or $m \neq_{\beta\iota} \mathrm{Constr}(j, J)\vec{N}$ or there does not exist $\Delta'' \supseteq \Delta'$ such that $(\Delta'' \vdash \mathrm{Constr}(j, J)\vec{N}) \in [\Gamma \vdash I]_{\xi|_{\Delta'},\rho,\Delta'}$. Thus

$$\begin{aligned}&G_{\Gamma,I,Q,\vec{f},\xi|_{\Delta'},\rho,\Delta'}([\Gamma \vdash I]_{\xi|_{\Delta'},\rho,\Delta'})(\Delta' \vdash m, [\Gamma \vdash M]_{\xi|_{\Delta'},\rho,\Delta'}) \\ &\qquad = Min_{\Delta'}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\}).\end{aligned}$$

Hence indeed

$$\begin{aligned}&(G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}([\Gamma \vdash I]_{\xi,\rho,\Delta})(\Delta \vdash m, [\Gamma \vdash M]_{\xi,\rho,\Delta}))|_{\Delta'} \\ &\qquad = G_{\Gamma,I,Q,\vec{f},\xi|_{\Delta'},\rho,\Delta'}([\Gamma \vdash I]_{\xi|_{\Delta'},\rho,\Delta'})(\Delta' \vdash m, [\Gamma \vdash M]_{\xi|_{\Delta'},\rho,\Delta'}). \quad \square\end{aligned}$$

Corollary 68. *Under the assumptions of Lemma 67 we have*

- $[\Gamma \vdash \text{Elim}(I, Q, M)\{\vec{f}\}]_{\xi,\rho,\Delta} \in V_\Delta(\rho(\text{Elim}(I, Q, M)\{\vec{f}\}))$,
- $([\Gamma \vdash \text{Elim}(I, Q, M)\{\vec{f}\}]_{\xi,\rho,\Delta})|_{\Delta'} = [\Gamma \vdash \text{Elim}(I, Q, M)\{\vec{f}\}]_{\xi|_{\Delta'},\rho,\Delta'}$.

Proof. It follows from Lemma 66 and Lemma 67.

□

Lemma 69. *If* $(\Delta \vdash \rho(\text{Constr}(n, I)\vec{N})) \in [\Gamma \vdash I]_{\xi,\rho,\Delta}$ *then*

$$[\Gamma \vdash \text{Elim}(I, Q, \text{Constr}(n, I)\vec{N})\{\vec{f}\}]_{\xi,\rho,\Delta} \\ = [\Gamma \vdash f_n]_{\xi,\rho,\Delta} \cdot g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[\rho(\text{Constr}(n, I)\vec{N}), [\Gamma \vdash \text{Constr}(n, I)\vec{N}]_{\xi,\rho,\Delta}, C_n(I), \vec{N}].$$

Proof. By the definition of interpretation we have

$$[\Gamma \vdash \text{Elim}(I, Q, \text{Constr}(n, I)\vec{N})\{\vec{f}\}]_{\xi,\rho,\Delta} = \\ G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}([\Gamma \vdash I]_{\xi,\rho,\Delta})(\Delta \vdash \rho(\text{Constr}(n, I)\vec{N}), [\Gamma \vdash \text{Constr}(n, I)\vec{N}]_{\xi,\rho,\Delta}).$$

Then for $U = [\Gamma \vdash \text{Constr}(n, I)\vec{N}]_{\xi,\rho,\Delta}$ we have

$$G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}([\Gamma \vdash I]_{\xi,\rho,\Delta})(\Delta \vdash \rho(\text{Constr}(n, I)\vec{N}), U) = \bigsqcup(Base \cup Min)$$

where $Base$ and Min are as in the definition of the operator $G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}$. Let $\mathcal{F}$ be the set of all values of the form

$$\Big([\Gamma \vdash f_j]_{\xi,\rho,\Delta} \cdot g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[\rho(\text{Constr}(n, I)\vec{N}), [\Gamma \vdash \text{Constr}(n, I)\vec{N}]_{\xi,\rho,\Delta}, C_n(I), \vec{N}]\Big)\,|_{\Delta'}$$

where $\Delta' \supseteq \Delta$. By the Church-Rosser property and the definition of the set $V_\Delta(\rho(f_j))$ it is easy to observe that $\mathcal{F} = Base$. Moreover, if $f \in Min$ then there exists $h \in \mathcal{F}$ such that $f \leq h$. Lemma 63 implies that

$$\bigsqcup(Base \cup Min) = \bigsqcup \mathcal{F}$$

The conclusion is a consequence of Lemma 62. □

5.3.3. Interpretation of inductive predicates

In this section we give the interpretation of inductive predicates. The interpretation is a function which takes an appropriate sequence of arguments (depending on arity) and returns a set of simple sequents. Only sequents which behave correctly under all possible eliminations are in the set: we consider all possible targets of elimination Q together with their interpretations and all possible branches for this target. We expect that under all such eliminations the term in question is an element of the interpretation of Q. Suppose $A = \Pi\vec{x} : \vec{\tau}.*^p$. We use an abbreviation $I = \text{Ind}(X : A)\{\vec{C}\}$. We define the interpretation for an inductive formula as the least fixpoint of a certain operator $H_{\Gamma,I,\xi,\rho,\Delta} : V_\Delta(\rho(I)) \to V_\Delta(\rho(I))$:

$$[\Gamma \vdash I]_{\xi,\rho,\Delta} = \mathit{lfp}(H_{\Gamma,I,\xi,\rho,\Delta}).$$

The value $H_{\Gamma,I,\xi,\rho,\Delta}(S)$ is a function which takes an appropriate sequence of arguments $(\Sigma_i, u_i, U_i)_{i=1}^n$ for $(\vec{x} : \vec{\tau})$ at $\langle \xi, \rho \rangle$ in Δ; if the vector $\vec{\tau}$ is empty then Σ_0 will denote the context Δ. The value is the union of two sets:

$$H_{\Gamma,I,\xi,\rho,\Delta}(S)((\Sigma_i, u_i, U_i)_{i=1}^n) = (\bigcap SAT^{\Sigma_n}_{\rho(I)\vec{u}}) \cup h_{\Gamma,I,\xi,\rho,\Delta}(S)((\Sigma_i, u_i, U_i)_{i=1}^n)$$

where $h_{\Gamma,I,\xi,\rho,\Delta}(S)((\Sigma_i, u_i, U_i)_{i=1}^n)$ consists of simple sequents $(\Delta' \vdash m) \in SN^{\Sigma_n}_{\rho(I)\vec{u}}$ such that

for every context Δ'' and every term J such that $(\Delta'' \vdash J) \in SN^{\Sigma_n}_{\rho(A)}$ and $J =_{\beta\iota} \rho(I)$,

for every context Δ''' and every term Q such that $(\Delta''' \vdash Q) \in [\Gamma \vdash A]_{\xi|_{\Delta''},\rho,\Delta''}$,

for every $P \in V_{\Delta'''}(Q)$, ,

for every context Δ'''' and for every vector $\vec{f}$ such that

$$(\Delta'''' \vdash f_i) \in [\Gamma, q : A, X : A \vdash \triangle\{C_i(X), q\}]_{(\xi|_{\Delta''''};X:=S|_{\Delta''''};q:=P),(\rho;X:=J;q:=Q),\Delta''''}$$

we have $(\Delta'''' \vdash \mathrm{Elim}(J, Q, \vec{u}, m)\{\vec{f}\}) \in P((\Sigma_i, u_i, U_i)_{i=1}^n)$.

The simple sequents in $h_{\Gamma,I,\xi,\rho,\Delta}(S)((\Sigma_i, u_i, U_i)_{i=1}^n)$ are in the set $SN^{\Sigma_n}_{\rho(I)\vec{u}}$, not $SN^{\Sigma_n}_{\rho(I\vec{u})}$ because the intended use of the operator $H_{\Gamma,I,\xi,\rho,\Delta}$ is for sequents already interpreted by ρ. For example we have

$$[\Gamma \vdash Iu_1]_{\xi,\rho,\Delta} = \mathit{lfp}(H_{\Gamma,I,\xi,\rho,\Delta})(\Delta \vdash \rho(u_1), [\Gamma \vdash u_1]_{\xi,\rho,\Delta}).$$

We prove that the definition of interpretation is correct, that is that the fixpoint actually exists. First we state an auxiliary lemma.

Lemma 70. *Let Δ be a context and M, M', I, Q, $\vec{u}$, $\vec{f}$ be terms such that $(\Delta \vdash M) \in SN^{\Delta}_{I\vec{u}}$ and $(\Delta \vdash \mathrm{Elim}(I, Q, \vec{u}, M)\{\vec{f}\}) \in SN^{\Delta}_{Q\vec{u}}$. If $M' \to_k M$ then*

$$(\Delta \vdash \mathrm{Elim}(I, Q, \vec{u}, M')\{\vec{f}\}) \in SN^{\Delta}_{Q\vec{u}}.$$

Proof. It follows from Corollary 38. □

We prove that the operator $H_{\Gamma,I,\xi,\rho,\Delta}$ is well defined.

Lemma 71. *Suppose Γ and Δ are two contexts, $I = \mathrm{Ind}(X : A)\{\vec{C}\}$ is an inductive predicate in Γ and $\langle \xi, \rho \rangle$ is a constructor valuation which satisfies Γ at Δ. If $S \in V_\Delta(\rho(I))$ then*

$$H_{\Gamma,I,\xi,\rho,\Delta}(S) \in V_\Delta(\rho(I)).$$

Proof. Let $S \in V_\Delta(\rho(I))$. Suppose $A = \Pi\vec{x} : \vec{\tau}.*^p$ and $\vec{\tau}$ is a vector of length n and $(\Sigma_i, u_i, U_i)_{i=1}^n$ is an appropriate sequence of arguments for $(\vec{x} : \vec{\tau})$ at $\langle \xi, \rho \rangle$ in Δ. We will show that

$$H_{\Gamma,I,\xi,\rho,\Delta}(S)((\Sigma_i, u_i, U_i)_{i=1}^n) \in SAT^{\Sigma_n}_{\rho(I)\vec{u}}.$$

We have to prove the four conditions in the definition of a saturated set. The conditions **(SAT1)**, **(SAT2)** and **(SAT3)** are straightforward. We only show the condition **(SAT4)**.

Suppose $(\Delta' \vdash M) \in H_{\Gamma,I,\xi,\rho,\Delta}(S)((\Sigma_i, u_i, U_i)_{i=1}^n)$, $M' \to_k M$ and $(\Delta' \vdash M') \in SN^{\Delta}_{\rho(I)\vec{u}}$. We will prove that

$$(\Delta' \vdash M') \in H_{\Gamma,I,\xi,\rho,\Delta}(S)((\Sigma_i, u_i, U_i)_{i=1}^n).$$

There are two cases.
Case 1: $(\Delta' \vdash M) \in \bigcap SAT^{\Sigma_n}_{\rho(I)\vec{u}}$. Then $(\Delta' \vdash M') \in H_{\Gamma,I,\xi,\rho,\Delta}(S)((\Sigma_i, u_i, U_i)^n_{i=1})$ as $\bigcap SAT^{\Sigma_n}_{\rho(I)\vec{u}}$ is saturated.
Case 2: $(\Delta' \vdash M) \notin \bigcap SAT^{\Sigma_n}_{\rho(I)\vec{u}}$. We have $(\Delta' \vdash M) \in h_{\Gamma,I,\xi,\rho,\Delta}(S)((\Sigma_i, u_i, U_i)^n_{i=1})$. We will prove that $(\Delta' \vdash M') \in h_{\Gamma,I,\xi,\rho,\Delta}(S)((\Sigma_i, u_i, U_i)^n_{i=1})$. Note that $(\Delta' \vdash M) \in SN^{\Sigma_n}_{\rho(I)\vec{u}}$. Take

- a context Δ'' and a term J such that $(\Delta'' \vdash J) \in SN^{\Delta'}_{\rho(A)}$ and $J =_{\beta\iota} \rho(I)$
- a context Δ''' and a term Q such that $(\Delta''' \vdash Q) \in [\Gamma \vdash A]_{\xi|_{\Delta''},\rho,\Delta''}$
- an interpretation $P \in V_{\Delta'''}(Q)$
- a context Δ'''' and for every $i = 1, \ldots n$ take a term f_i such that

$$(\Delta'''' \vdash f_i) \in [\Gamma, q : A, X : A \vdash \triangle\{C_i(X), q\}]_{(\xi|_{\Delta''''};X:=S|_{\Delta''''};q:=P),(\rho;X:=J;q:=Q),\Delta''''}$$

The assumption $(\Delta' \vdash M) \in h_{\Gamma,I,\xi,\rho,\Delta}(S)((\Sigma_i, u_i, U_i)^n_{i=1})$ implies that

$$(\Delta'''' \vdash \text{Elim}(J, Q, \vec{u}, M)\{\vec{f}\}) \in P((\Sigma_i, u_i, U_i)^n_{i=1}).$$

We also have

$$\text{Elim}(J, Q, \vec{u}, M')\{\vec{f}\} \to_k \text{Elim}(J, Q, \vec{u}, M)\{\vec{f}\}.$$

By Lemma 70 we have $(\Delta'''' \vdash \text{Elim}(J, Q, \vec{u}, M')\{\vec{f}\}) \in SN^{\Delta''''}_{\rho(Q)\vec{u}}$. As $P((\Sigma_i, u_i, U_i)^n_{i=1})$ is a saturated set we get

$$(\Delta'''' \vdash \text{Elim}(J, Q, \vec{u}, M')\{\vec{f}\}) \in P((\Sigma_i, u_i, U_i)^n_{i=1}).$$

Hence indeed it holds that $(\Delta' \vdash M') \in H_{\Gamma,I,\xi,\rho,\Delta}(S)((\Sigma_i, u_i, U_i)^n_{i=1})$. By the definition of $H_{\Gamma,I,\xi,\rho,\Delta}$ it is easy to observe that the remaining conditions in the definition of $V_\Delta(\rho(I))$ are satisfied. □

We want to prove that the operator is monotone. First we show that interpretation of branches are antimonotone.

Lemma 72. *Suppose that*

1. $A = \Pi\vec{x} : \vec{\tau}.*^p$,
2. $(\Gamma, q : A, X : A, \Gamma')$ *and* Δ *are contexts,*
3. I *is an inductive predicate in* Γ *and* $J =_{\beta\iota} \rho(I)$,
4. $S, S' \in V_\Delta(J)$,
5. $\Delta \vdash Q : A$ *and* $P \in V_\Delta(Q)$,
6. $\langle(\xi; X := S'; q := P), (\rho; X := J; q := Q)\rangle$ *and* $\langle(\xi; X := S; q := P), (\rho; X := J; q := Q)\rangle$ *are constructor valuations which satisfy* $(\Gamma, q : A, X : A, \Gamma')$ *at* Δ.
7. *the interpretation of any term* M *which occurs in the derivation of* $\Gamma \vdash I : A$ *depends only on the values of a constructor valuation for the free variables of* M *(recall the definition on page 58.);*

8. $C(X)$ is a type of constructor of I.

If $S \leq S'$ (the definition of the relation $\leq$ was given on page 52) then

$$[\Gamma, q : A, X : A, \Gamma' \vdash \triangle\{C(X), q\}]_{(\xi;X:=S';q:=P),(\rho;X:=J;q:=Q),\Delta} \subseteq [\Gamma, q : A, X : A, \Gamma' \vdash \triangle\{C(X), q\}]_{(\xi;X:=S;q:=P),(\rho;X:=J;q:=Q),\Delta}.$$

Proof. We will use the abbreviations

$$\hat{\Gamma} = \Gamma, q : A, X : A, \Gamma' \quad \text{and} \quad \hat{\rho} = \rho; X := J; q := Q,$$
$$\hat{\xi}' = \xi; X := S'; q := P \quad \text{and} \quad \hat{\xi} = \xi; X := S; q := P.$$

We proceed by induction with respect to the definition of $\triangle\{C(X), q\}$.

- $C(X) = X\vec{t'}$. We consider the case when every t'_i is a large object, otherwise the proof is similar. Then $\triangle\{X\vec{t'}, q\} = q\vec{t'}$ and

  $$[\hat{\Gamma} \vdash \triangle\{X\vec{t'}, q\}]_{\hat{\xi}',\hat{\rho},\Delta} = P((\Sigma_i, u_i, U'_i)_{i=1}^n)$$

  and

  $$[\hat{\Gamma} \vdash \triangle\{X\vec{t'}, q\}]_{\hat{\xi},\hat{\rho},\Delta} = P((\Sigma_i, u_i, \hat{U}_i)_{i=1}^n)$$

  where $\vec{\Sigma}$, $\vec{u}$, $\vec{U}$, $\vec{\hat{U}}$ are vectors such that for every i:

  $$\Sigma_i = \Delta, \quad \text{and} \quad u_i = \hat{\rho}(t'_i),$$
  $$U'_i = [\Gamma \vdash t'_i]_{\hat{\xi}',\hat{\rho},\Delta} \quad \text{and} \quad \hat{U}_i = [\Gamma \vdash t'_i]_{\hat{\xi},\hat{\rho},\Delta}.$$

  Because $X, q \notin FV(t_i)$ we have $\vec{U'} = \vec{\hat{U}}$. Thus

  $$[\hat{\Gamma} \vdash \triangle\{X\vec{t'}, q\}]_{\hat{\xi}',\hat{\rho},\Delta} = P((\Sigma_i, u_i, U'{}_i)_{i=1}^n) = P(\Sigma_i, u_i, \hat{U}_i)_{i=1}^n) = [\hat{\Gamma} \vdash \triangle\{X\vec{t'}, q\}]_{\hat{\xi},\hat{\rho},\Delta}.$$

- $C(X) = \Pi x : t.D(X)$ and $X \notin FV(t)$. Then $\triangle\{\Pi x : t.D(X), q\} = \Pi x : t.\triangle\{D(X), q\}$. Suppose

  $$(\Delta' \vdash M) \in [\hat{\Gamma} \vdash \triangle\{\Pi x : t.D(X), q\}]_{\hat{\xi}',\hat{\rho},\Delta}.$$

  By the definition of interpretation for the product it means that $\Delta \subseteq \Delta'$ and

  $$\Delta' \vdash M : \rho(\triangle\{\Pi x : t.D(X), q\}).$$

  It means that if $(\Delta'' \vdash a) \in [\hat{\Gamma} \vdash t]_{\hat{\xi}'|_{\Delta''},\hat{\rho},\Delta''}$ and $P' \in V_{\Delta''}(a)$ then

  $$(\Delta'' \vdash Ma) \in [\hat{\Gamma}, x : t \vdash \triangle\{D(X), q\}]_{(\hat{\xi}'|_{\Delta''};x:=P'),(\hat{\rho};x:=a),\Delta''}.$$

  We will prove that

  $$(\Delta' \vdash M) \in [\hat{\Gamma} \vdash \triangle\{\Pi x : t.D(X), q\}]_{\hat{\xi},\hat{\rho},\Delta}.$$

  Take Δ'', a and P such that

  $$\Delta'' \supseteq \Delta', \ (\Delta'' \vdash a) \in [\hat{\Gamma} \vdash t]_{\hat{\xi}|_{\Delta''},\hat{\rho},\Delta''}, \text{ and } P' \in V_{\Delta''}(a).$$

By the assumption the sets $[\hat{\Gamma} \vdash t]_{\hat{\xi}|_{\Delta''},\hat{\rho},\Delta''}$ and $[\hat{\Gamma} \vdash t]_{\hat{\xi'}|_{\Delta''},\hat{\rho},\Delta''}$ are equal. Thus

$$(\Delta'' \vdash a) \in [\hat{\Gamma} \vdash t]_{\hat{\xi'}|_{\Delta''},\hat{\rho},\Delta''}.$$

Then by assumption

$$(\Delta'' \vdash Ma) \in [\hat{\Gamma}, x : t \vdash \triangle\{D(X), q\}]_{(\hat{\xi'}|_{\Delta''};x:=P),(\hat{\rho};x:=a),\Delta''}.$$

By the induction hypothesis and the fact that $S|_{\Delta''} \leq S'|_{\Delta''}$ we get that

$$[\hat{\Gamma}, x : t \vdash \triangle\{D(X), q\}]_{(\hat{\xi'}|_{\Delta''};x:=P),(\hat{\rho};x:=a),\Delta''} \subseteq [\hat{\Gamma}, x : t \vdash \triangle\{D(X), q\}]_{(\hat{\xi}|_{\Delta''};x:=P),(\hat{\rho};x:=a),\Delta''}$$

and thus

$$(\Delta'' \vdash Ma) \in [\hat{\Gamma}, x : t \vdash \triangle\{D(X), q\}]_{(\hat{\xi}|_{\Delta''};x:=P),(\hat{\rho};x:=a),\Delta''}.$$

Hence indeed

$$(\Delta' \vdash M) \in [\hat{\Gamma} \vdash \triangle\{\Pi x : t.D(X), q\}]_{\hat{\xi},\hat{\rho},\Delta}.$$

- $C(X) = \left(\Pi\vec{x} : \vec{t}.X\vec{t'}\right) \to D(X)$. Then

$$\triangle\{\left(\Pi\vec{x} : \vec{t}.X\vec{t'}\right) \to D(X), q\} = (\Pi\vec{x} : \vec{t}.X\vec{t'}) \to (\Pi\vec{x} : \vec{t}.q\vec{t'}) \to \triangle\{D(X), q\}.$$

Let $(\Delta' \vdash M) \in [\hat{\Gamma} \vdash \triangle\{\left(\Pi\vec{x} : \vec{t}.X\vec{t'}\right) \to D(X), q\}]_{\hat{\xi'},\hat{\rho},\Delta}$. Suppose

$$\Delta'' \supseteq \Delta',\ (\Delta'' \vdash a) \in [\hat{\Gamma} \vdash \Pi\vec{x} : \vec{t}.X\vec{t'}]_{\hat{\xi}|_{\Delta''},\hat{\rho},\Delta''},\ P' \in V_{\Delta''}(a).$$

and

$$\Delta''' \supseteq \Delta'',\ (\Delta'' \vdash b) \in [\hat{\Gamma} \vdash \Pi\vec{x} : \vec{t}.q\vec{t'}]_{\hat{\xi}|_{\Delta''},\hat{\rho},\Delta'''},\ P'' \in V_{\Delta'''}(b).$$

Note that $X \notin FV(\vec{t}) \cup FV(\vec{t'})$ and thus by the assumption

$$[\hat{\Gamma} \vdash \Pi\vec{x} : \vec{t}.q\vec{t'}]_{\hat{\xi}|_{\Delta'''},\hat{\rho},\Delta'''} = [\hat{\Gamma} \vdash \Pi\vec{x} : \vec{t}.q\vec{t'}]_{\hat{\xi'}|_{\Delta'''},\hat{\rho},\Delta'''}.$$

Moreover, it is easy to check that

$$[\hat{\Gamma} \vdash \Pi\vec{x} : \vec{t}.X\vec{t'}]_{\hat{\xi}|_{\Delta''},\hat{\rho},\Delta''} \subseteq [\hat{\Gamma} \vdash \Pi\vec{x} : \vec{t}.X\vec{t'}]_{\hat{\xi'}|_{\Delta''},\hat{\rho},\Delta''}.$$

Thus

$$(\Delta'' \vdash a) \in [\hat{\Gamma} \vdash \Pi\vec{x} : \vec{t}.X\vec{t'}]_{\hat{\xi'}|_{\Delta''},\hat{\rho},\Delta''},$$

and

$$(\Delta'' \vdash b) \in [\hat{\Gamma} \vdash \Pi\vec{x} : \vec{t}.q\vec{t'}]_{\hat{\xi'}|_{\Delta'''},\hat{\rho},\Delta'''}.$$

By the assumption we get that

$$(\Delta''' \vdash Mab) \in [\hat{\Gamma}, x : \Pi\vec{x} : \vec{t}.X, y : \Pi\vec{x} : \vec{t}.q \vdash \triangle\{D(X), q\}]_{(\hat{\xi'}|_{\Delta'''};x:=P|'_{\Delta'''};y:=P''),(\hat{\rho};x:=a;y:=b),\Delta'''}.$$

By the induction hypothesis and the fact that $S|_{\Delta'''} \leq S'|_{\Delta'''}$

$$[\hat{\Gamma}, x : \Pi\vec{x} : \vec{t}.X, y : \Pi\vec{x} : \vec{t}.q \vdash \triangle\{D(X), q\}]_{(\hat{\xi}'|_{\Delta'''};x:=P'|_{\Delta'''};y:=P''),(\hat{\rho};x:=a;y:=b),\Delta'''} \subseteq [\hat{\Gamma}, x : \Pi\vec{x} : \vec{t}.X, y : \Pi\vec{x} : \vec{t}.q \vdash \triangle\{D(X), q\}]_{(\hat{\xi}|_{\Delta'''};x:=P'|_{\Delta'''};y:=P''),(\hat{\rho};x:=a;y:=b),\Delta'''}.$$

Hence it holds that

$$(\Delta''' \vdash Mab) \in [\hat{\Gamma} \vdash \{(\Pi\vec{x} : \vec{t}.X) \to D(X), q\}]_{(\hat{\xi}|_{\Delta'''};x:=P'|_{\Delta'''};y:=P''),(\hat{\rho};x:=a;y:=b),\Delta'''},$$

Thus

$$(\Delta' \vdash M) \in [\hat{\Gamma} \vdash \triangle\{\Pi x : t.D(X), q\}]_{\hat{\xi},\hat{\rho},\Delta}. \qquad \square$$

Now we prove that operator $H_{\Gamma,I,\xi,\rho,\Delta}$ is monotone.

Lemma 73. *Assume that*

1. Γ *and* Δ *are contexts,*
2. I *is an inductive predicate in* Γ,
3. $S, S' \in V_\Delta(I)$,
4. $\langle\xi, \rho\rangle$ *is a constructor valuation which satisfies* Γ *at* Δ,
5. *the interpretation of any subterm* M *of* I *depends only on the values of* ξ *and* ρ *for the free variables of* M.

If $S \leq S'$ *then* $H_{\Gamma,I,\xi,\rho,\Delta}(S) \leq H_{\Gamma,I,\xi,\rho,\Delta}(S')$.

Proof. Let $S \leq S'$. Suppose $(\Sigma_i, u_i, U_i)_{i=1}^n$ is an appropriate sequence of arguments for $(\vec{x} : \vec{\tau})$ at $\langle\xi, \rho\rangle$ in Δ. We will prove that

$$H_{\Gamma,I,\xi,\rho,\Delta}(S)((\Sigma_i, u_i, U_i)_{i=1}^n) \subseteq H_{\Gamma,I,\xi,\rho,\Delta}(S')((\Sigma_i, u_i, U_i)_{i=1}^n).$$

Let $(\Delta' \vdash M) \in H_{\Gamma,I,\xi,\rho,\Delta}(S)((\Sigma_i, u_i, U_i)_{i=1}^n)$. Then there are two cases.
Case 1: $(\Delta' \vdash M) \in \bigcap SAT^{\Sigma_n}_{\rho(I)\vec{u}}$. By the definition of $H_{\Gamma,I,\xi,\rho,\Delta}$ it obviously holds that $(\Delta' \vdash M) \in H_{\Gamma,I,\xi,\rho,\Delta}(S')((\Sigma_i, u_i, U_i)_{i=1}^n)$.
Case 2: $(\Delta' \vdash M) \notin \bigcap SAT^{\Sigma_n}_{\rho(I)\vec{u}}$. Then $(\Delta' \vdash M) \in h_{\Gamma,I,\xi,\rho,\Delta}(S)((\Sigma_i, u_i, U_i)_{i=1}^n)$. We will prove that $(\Delta' \vdash M) \in h_{\Gamma,I,\xi,\rho,\Delta}(S')((\Sigma_i, u_i, U_i)_{i=1}^n)$. By the assumption we know that

$$\Delta' \supseteq \Sigma_n,\ (\Delta' \vdash M) : \rho(I)\vec{u},\ \text{and}\ (\Delta' \vdash M) \in SN^{\Sigma_n}_{\rho(I)\vec{u}}.$$

Let

$$(\Delta'' \vdash J) \in SN^{\Delta'}_{\rho(A)} \text{ and } J =_{\beta\iota} \rho(I)$$
$$(\Delta''' \vdash Q) \in [\Gamma \vdash A]_{\xi|_{\Delta''},\rho,\Delta''},\ P \in V_{\Delta'''}(Q),$$
$$(\Delta'''' \vdash f_i) \in [\Gamma, q : A, X : A \vdash \triangle\{C_i(X), q\}]_{(\xi|_{\Delta'''};X:=S'|_{\Delta'''};q:=P),(\rho;X:=J;q:=Q),\Delta'''}.$$

Note that

$$\langle(\xi|_{\Delta'''}; X := S'|_{\Delta'''}; q := P), (\rho; X := J; q := Q)\rangle$$
$$\text{and } \langle(\xi|_{\Delta'''}; X{:=}S|_{\Delta'''}; q{:=}P), (\rho; X{:=}J; q{:=}Q)\rangle$$

are constructor valuations which satisfy $(\Gamma, q : A, X : A)$ at Δ'''. By Lemma 72

$$[\Gamma, q : A, X : A \vdash \triangle\{C_i(X), q\}]_{(\xi|_{\Delta'''};X:=S'|_{\Delta'''};q:=P),(\rho;X:=J;q:=Q),\Delta'''} \subseteq [\Gamma, q : A, X : A \vdash \triangle\{C_i(X), q\}]_{(\xi|_{\Delta'''};X:=S|_{\Delta'''};q:=P),(\rho;X:=J;q:=Q),\Delta'''}.$$

Thus

$$(\Delta'''' \vdash f_i) \in [\Gamma, q : A, X : A \vdash \triangle\{C_i(X), q\}]_{(\xi|_{\Delta'''};X:=S|_{\Delta'''};q:=P),(\rho;X:=J;q:=Q),\Delta'''}.$$

Applying the assumption $(\Delta' \vdash M) \in h_{\Gamma,I,\xi,\rho,\Delta}(S')((\Sigma_i, u_i, U_i)_{i=1}^n)$ we get

$$(\Delta'''' \vdash \mathrm{Elim}(J, Q, \vec{u}, M)\{\vec{f}\}) \in P((\Sigma_i, u_i, U_i)_{i=1}^n).$$

Thus indeed $(\Delta' \vdash M) \in H_{\Gamma,I,\xi,\rho,\Delta}(S')((\Sigma_i, u_i, U_i)_{i=1}^n)$. □

Lemma 74. *Assume that*

1. Γ, Δ *and* Δ' *are contexts, and* $\Delta' \supseteq \Delta$,
2. I *is an inductive formula in* Γ,
3. $\langle \xi, \rho \rangle$ *is a constructor valuation which satisfies* Γ *at* Δ,
4. *the interpretation of any subterm* M *of* I *depends only on the values of* ξ *and* ρ *for the free variables of* M.

Then

$$([\Gamma \vdash \mathrm{Ind}(X : *^p)\{C_i(X)\}]_{\xi,\rho,\Delta}) \,|_{\Delta'} = [\Gamma \vdash \mathrm{Ind}(X : *^p)\{C_i(X)\}]_{\xi|_{\Delta'},\rho,\Delta'}.$$

Proof. Recall that

$$([\Gamma \vdash \mathrm{Ind}(X : *^p)\{C_i(X)\}]_{\xi,\rho,\Delta})|_{\Delta'} = (\mathit{lfp}(H_{\Gamma,I,\xi,\rho,\Delta}))|_{\Delta'}$$

and $\mathit{lfp}(H_{\Gamma,I,\xi,\rho,\Delta})((\Sigma_i, u_i, U_i)_{i=1}^n) = \bigcup_{\alpha'<\alpha} H^{\alpha'}_{\Gamma,I,\xi,\rho,\Delta}(Min^{\Delta}(\rho(I)))((\Sigma_i, u_i, U_i)_{i=1}^n)$ for a certain α. We will prove that

$$(H_{\Gamma,I,\xi,\rho,\Delta}(S))|_{\Delta'} = H_{\Gamma,I,\xi|_{\Delta'},\rho,\Delta'}(S|_{\Delta'}).$$

Observe that the domains of both functions are the same. Moreover if $((\Sigma_i, u_i, U_i)_{i=1}^n)$ is in the domain then

$$(H_{\Gamma,I,\xi,\rho,\Delta}(S))|_{\Delta'}((\Sigma_i, u_i, U_i)_{i=1}^n) = (\bigcap SAT^{\Sigma_n}_{\rho(I)}) \cup h_{\Gamma,I,\xi,\rho,\Delta'}(S)((\Sigma_i, u_i, U_i)_{i=1}^n)$$

and

$$H_{\Gamma,I,\xi|_{\Delta'},\rho,\Delta'}(S|_{\Delta'})((\Sigma_i, u_i, U_i)_{i=1}^n) = (\bigcap SAT^{\Sigma_n}_{\rho(I)}) \cup h_{\Gamma,I,\xi|_{\Delta'},\rho,\Delta'}(S|_{\Delta'})((\Sigma_i, u_i, U_i)_{i=1}^n)$$

Recall that $h_{\Gamma,I,\xi,\rho,\Delta'}(S)((\Sigma_i, u_i, U_i)_{i=1}^n)$ consists of simple sequents $(\Delta' \vdash m) \in SN^{\Sigma_n}_{\rho(I)\vec{u}}$ such that

for every context Δ'' and every term J such that $(\Delta'' \vdash J) \in SN^{\Sigma_n}_{\rho(A)}$ and $J =_{\beta\iota} \rho(I)$,

for every context Δ''' and every term Q such that $(\Delta''' \vdash Q) \in [\Gamma \vdash A]_{\xi|_{\Delta''},\rho,\Delta''}$, for every $P \in V_{\Delta'''}(Q)$, ,

for every context Δ'''' and for every vector $\vec{f}$ such that

$$(\Delta'''' \vdash f_i) \in [\Gamma, q : A, X : A \vdash \triangle\{C_i(X), q\}]_{(\xi|_{\Delta''''};X:=S|_{\Delta''''};q:=P),(\rho;X:=J;q:=Q),\Delta''''}$$

we have $(\Delta'''' \vdash \mathrm{Elim}(J, Q, \vec{u}, m)\{\vec{f}\}) \in P((\Sigma_i, u_i, U_i)_{i=1}^n)$.

and $h_{\Gamma,I,\xi|_{\Delta'},\rho,\Delta'}(S|_{\Delta'})((\Sigma_i, u_i, U_i)_{i=1}^n)$ consists of simple sequents $(\Delta' \vdash m) \in SN^{\Sigma_n}_{\rho(I)\vec{u}}$ such that

for every context Δ'' and every term J such that $(\Delta'' \vdash J) \in SN^{\Sigma_n}_{\rho(A)}$ and $J =_{\beta\iota} \rho(I)$,

for every context Δ''' and every term Q such that $(\Delta''' \vdash Q) \in [\Gamma \vdash A]_{\xi|_{\Delta'}|_{\Delta''},\rho,\Delta''}$,

for every $P \in V_{\Delta'''}(Q)$, ,

for every context Δ'''' and for every vector $\vec{f}$ such that

$$(\Delta'''' \vdash f_i) \in [\Gamma, q : A, X : A \vdash \triangle\{C_i(X), q\}]_{(\xi|_{\Delta'}|_{\Delta''''};X:=S|_{\Delta'}|_{\Delta''''};q:=P),(\rho;X:=J;q:=Q),\Delta''''}$$

we have $(\Delta'''' \vdash \mathrm{Elim}(J, Q, \vec{u}, m)\{\vec{f}\}) \in P((\Sigma_i, u_i, U_i)_{i=1}^n)$.

Note that $(\xi|_{\Delta'})|_{\Delta''} = \xi|_{\Delta''}$ and $(S|_{\Delta'})|_{\Delta''} = S|_{\Delta''}$ for all $\Delta'' \supseteq \Delta'$. Thus indeed

$$(H_{\Gamma,I,\xi,\rho,\Delta}(S))|_{\Delta'} = H_{\Gamma,I,\xi|_{\Delta'},\rho,\Delta'}(S|_{\Delta'}).$$ □

Lemma 75. *Assume that*

1. Γ *and* Δ *are contexts,*
2. $I = \mathrm{Ind}(X : A)\{\vec{C}\}$ *is an inductive predicate in* Γ*,*
3. $S, S' \in V_\Delta(I)$*,*
4. $\langle \xi, \rho \rangle$ *is a constructor valuation which satisfies* Γ *at* Δ*,*
5. *the interpretation of any subterm* M *of* I *depends only on the values of* ξ *and* ρ *for the free variables of* M*.*

Then

- $[\Gamma \vdash \mathrm{Ind}(X : A)\{\vec{C}\}]_{\xi,\rho,\Delta} \in V_\Delta(\rho(I))$.
- *If* $\Delta' \supseteq \Delta$ *then* $\left([\Gamma \vdash \mathrm{Ind}(X : A)\{\vec{C}\}]_{\xi,\rho,\Delta}\right)|_{\Delta'} = [\Gamma \vdash \mathrm{Ind}(X : A)\{\vec{C}\}]_{\xi|_{\Delta'},\rho,\Delta'}$.

Proof. By Lemma 71 the function $H_{\Gamma,I,\xi,\rho,\Delta}$ used in the definition of the interpretation $[\Gamma \vdash I]_{\xi,\rho,\Delta}$ is well defined and by Lemma 73 it is monotone. We have a monotone function on the complete lattice $V_\Gamma(\rho(I))$ (see Lemma 42). Then $lfp(H_{\Gamma,I,\xi,\rho,\Delta})$ exists and we have

$$[\Gamma \vdash \mathrm{Ind}(X : A)\{\vec{C}\}]_{\xi,\rho,\Delta} = lfp(H_{\Gamma,I,\xi,\rho,\Delta}) \in V_\Delta(\rho(I)).$$

If $\Delta' \supseteq \Delta$ then the second item follows from Lemma 74. □

Remark 76. The definitions for inductive types and inductive predicates are similar, yet their interpretations are different. However, we could not use an operator F' similar to F to define an interpretation for inductive predicates. This is because formula polymorphism is allowed in the type system. If we tried to give a definition for F' we would have to have an interpretation for an arbitrary formula Q, including formulas structurally greater than the inductive predicate we are dealing with. But then the definition of F' would not be well-founded. Similarly, we could not use an operator H' resembling H to define the interpretation for inductive types. We are allowed to use dependent elimination for inductive objects. Thus the interpretation of the result q would be a function taking the interpretation of an eliminated term M. However, this elimination could not be given because M could be an arbitrary term of type I.

5.3.4. Correctness of the interpretation

In this section we prove that the definition of interpretation is correct and that it has the property

$$[\Gamma \vdash M]_{\xi,\rho,\Delta} \in V_\Delta(\rho(M)).$$

We combine the correctness results stated in the previous sections: for inductive types, inductive predicates and elimination terms. We also give correctness proofs for the missing cases: the product, the abstraction, the application, the inductive object.

We begin the section with some technical lemmas. We prove that the interpretation of a term M depends only on the values (up to $\beta\iota$-equality) of the constructor valuations for its free variables.

Lemma 77. *Let Γ, Γ' and Δ be contexts. Let M be a term such that $\Gamma \vdash M : T$. Let $\langle \xi, \rho \rangle$ be a constructor valuation which satisfies Γ at Δ and $\langle \xi', \rho' \rangle$ be a constructor valuation which satisfies Γ' at Δ. Suppose for each variable $x \in FV(M)$ it holds that $\Gamma(x) = \Gamma'(x)$, $\rho(x) =_{\beta\iota} \rho'(x)$ and $\xi(x) = \xi'(x)$. Then*

$$[\Gamma \vdash M]_{\xi,\rho,\Delta} = [\Gamma' \vdash M]_{\xi',\rho',\Delta}.$$

Proof. Induction with respect to the structure of M. □

Lemma 78. *Let Γ, Γ' and Δ be contexts. Suppose $\langle \xi, \rho \rangle$ is a constructor valuation which satisfies Γ at Δ. Suppose that $dom(\Gamma) = dom(\Gamma')$ and for each $x \in dom(\Gamma)$ we have $\Gamma(x) =_{\beta\iota} \Gamma'(x)$. Then if M is a term such that $\Gamma \vdash M : T$ then it holds that*

$$[\Gamma \vdash M]_{\xi,\rho,\Delta} = [\Gamma' \vdash M]_{\xi,\rho,\Delta}.$$

In particular, both interpretations are well defined.

Proof. If $\langle \xi, \rho \rangle$ satisfies Γ at Δ then it also satisfies Γ' at Δ: if $(x : T') \in \Gamma'$ then $T =_{\beta\iota} T'$, and $(x : T) \in \Gamma$. Then by assumption $\Delta \vdash \rho(x) : \rho(T)$ but also $\Delta \vdash \rho(x) : \rho(T')$ by conversion rule.

The proof is by easy induction with respect to the structure of M. □

Lemma 79. *Suppose*

1. *Γ, Δ are two contexts and $\langle \xi, \rho \rangle$ is a constructor valuation which satisfies Γ at Δ,*

2. $\Gamma \vdash (\Pi x : M_1.M_2) : s$.

3. for every $\Delta'' \supseteq \Delta$ *we have* $[\Gamma \vdash M_1]_{\xi|_{\Delta''},\rho,\Delta''} \in SAT^{\Delta''}_{\rho(M_1)}$ *and if* $\Delta''' \supseteq \Delta''$ *then* $\big([\Gamma \vdash M_1]_{\xi|_{\Delta''},\rho,\Delta''}\big)|_{\Delta'''} = [\Gamma \vdash M_1]_{\xi|_{\Delta''}|_{\Delta'''},\rho,\Delta'''}$,

4. for every $\Delta'' \supseteq \Delta$ *for every* $\langle \xi', \rho' \rangle$ *satisfying* $(\Gamma, x : M_1)$ *at* Δ'' *we have*

$$[\Gamma, x : M_1 \vdash M_2]_{\xi',\rho',\Delta''} \in SAT^{\Delta''}_{\rho'(M_2)}$$

and if $\Delta''' \supseteq \Delta''$ *then*

$$\big([\Gamma, x : M_1 \vdash M_2]_{\xi',\rho',\Delta''}\big)|_{\Delta'''} = [\Gamma, x : M_1 \vdash M_2]_{\xi'|_{\Delta'''},\rho',\Delta'''}.$$

Then

- $[\Gamma \vdash \Pi x : M_1.M_2]_{\xi,\rho,\Delta} \in V_\Delta(\rho(\Pi x : M_1.M_2))$,
- *if* $\Delta' \supseteq \Delta$ *then* $([\Gamma \vdash \Pi x : M_1.M_2]_{\xi,\rho,\Delta})|_{\Delta'} = [\Gamma \vdash \Pi x : M_1.M_2]_{\xi|_{\Delta'},\rho,\Delta'}$.

Proof. We have

$$\frac{\Gamma \vdash M_1 : s_1 \qquad \Gamma, x : M_1 \vdash M_2 : s_2}{\Gamma \vdash \Pi x : M_1.M_2 : s}$$

We want to prove that $[\Gamma \vdash \Pi x : M_1.M_2]_{\xi,\rho,\Delta}$ is in $SAT^{\Delta}_{\rho(\Pi x:M_1.M_2)}$. Recall that

$$\begin{aligned}[\Gamma \vdash \Pi x : M_1.M_2]_{\xi,\rho,\Delta} = \{&(\Delta' \vdash M) \mid \Delta \subseteq \Delta' \text{ and } \Delta' \vdash M : \rho(\Pi x : M_1.M_2) \text{ and}\\ &\text{for every } \Delta'' \supseteq \Delta', \text{ for every } a \text{ such that } (\Delta'' \vdash a) \in [\Gamma \vdash M_1]_{\xi|_{\Delta''},\rho,\Delta''}\\ &\text{for every } P \in V_{\Delta''}(a)\\ &\text{we have } (\Delta'' \vdash Ma) \in [\Gamma, x : M_1 \vdash M_2]_{(\xi|_{\Delta''};x:=P),(\rho;x:=a);\Delta''})\}.\end{aligned}$$

By assumption $[\Gamma \vdash M_1]_{\xi|_{\Delta''},\rho,\Delta''} \in SAT^{\Delta''}_{\rho(M_1)}$ and thus if

$$(\Delta'' \vdash a) \in [\Gamma \vdash M_1]_{\xi|_{\Delta''},\rho,\Delta''} \text{ and } P \in V_{\Delta''}(a)$$

then $\langle(\xi|_{\Delta''}; x := P), (\rho; x := a)\rangle$ is a constructor valuation which satisfies $(\Gamma, x : M_1)$ at Δ''. By assumption

$$[\Gamma, x : M_1 \vdash M_2]_{(\xi|_{\Delta''};x:=P),(\rho;x:=a),\Delta''} \in SAT^{\Delta''}_{(\rho;x:=a)(M_2)}.$$

Let $X = [\Gamma \vdash \Pi x : M_1.M_2]_{\xi,\rho,\Delta}$. We prove that X is a saturated set.

(SAT1) Let $(\Delta' \vdash M) \in X$. We want to prove that M is strongly normalizing. By the definition of X, for every $\Delta'' \supseteq \Delta$, for every $(\Delta'' \vdash a) \in [\Gamma \vdash M_1]_{\xi|_{\Delta''},\rho,\Delta''}$, for every $P \in V_{\Delta''}(a)$ it holds that

$$(\Delta'' \vdash Ma) \in [\Gamma, x : M_1 \vdash M_2]_{(\xi|_{\Delta''};x:=P),(\rho;x:=a),\Delta''}.$$

By the assumption the set $[\Gamma, x : M_1 \vdash M_2]_{(\xi|_{\Delta''};x:=P),(\rho;x:=a),\Delta''}$ is saturated so every element of it is strongly normalizing. In particular, the term Ma is strongly normalizing and thus M is strongly normalizing as well.

(SAT2) Let $(\Delta' \vdash M) \in B^{\Delta}_{\rho(\Pi x:M_1.M_2)}$. We will show that $(\Delta' \vdash M) \in X$. By the definition of $B^{\Delta}_{\rho(\Pi x:M_1.M_2)}$ we have

$$\Delta \subseteq \Delta' \text{ and } (\Delta' \vdash M) : \rho(\Pi x : M_1.M_2).$$

Let

$$\Delta'' \supseteq \Delta', \quad (\Delta'' \vdash a) \in [\Gamma \vdash M_1]_{\xi|_{\Delta''},\rho,\Delta''}, \quad P \in V_{\Delta''}(a).$$

By assumption $[\Gamma \vdash M_1]_{\xi|_{\Delta''},\rho,\Delta''} \in SAT^{\Delta''}_{\rho(M_1)}$ and hence $\Delta'' \vdash a : \rho(M_1)$. Thus $\Delta'' \vdash Ma : (\rho; x := a)(M_2)$ and

$$(\Delta'' \vdash Ma) \in B^{\Delta''}_{(\rho;x:=a)(M_2)}$$

But $B^{\Delta''}_{(\rho;x:=a)(M_2)} \subseteq [\Gamma, x : M_1 \vdash M_2]_{(\xi|_{\Delta''};x:=P),(\rho;x:=a),\Delta''}$ and thus indeed

$$(\Delta'' \vdash Ma) \in [\Gamma, x : M_1 \vdash M_2]_{(\xi|_{\Delta''};x:=P),(\rho;x:=a),\Delta''}.$$

(SAT3) Let $(\Delta' \vdash M) \in X$ and let $\Delta_1 \supseteq \Delta'$. We will prove that

$$(\Delta_1 \vdash M) \in X.$$

By the definition of X we know that

$$\Delta' \vdash M : \rho(\Pi x : M_1.M_2), \Delta \subseteq \Delta',$$

and for every $\Delta'' \supseteq \Delta'$ for every a such that $(\Delta'' \vdash a) \in [\Gamma \vdash M_1]_{\xi|_{\Delta''},\rho,\Delta''}$ for every $P \in V_{\Delta''}(a)$ we have

$$(\Delta'' \vdash Ma) \vdash [\Gamma, x : M_1 \vdash M_2]_{(\xi|_{\Delta''};x:=P),(\rho;x:=a),\Delta''}.$$

Then obviously $\Delta_1 \vdash M : \rho(\Pi x : M_1.M_2)$ and $\Delta \subseteq \Delta' \subseteq \Delta_1$. Let

$$\Delta'' \supseteq \Delta_1, \quad (\Delta'' \vdash a) \in [\Gamma \vdash M_1]_{\xi|_{\Delta''},\rho,\Delta''}, \quad P \in V_{\Delta''}(a).$$

Then it also holds that $\Delta'' \supseteq \Delta'$ and thus

$$(\Delta'' \vdash Ma) \in [\Gamma, x : M_1 \vdash M_2]_{(\xi|_{\Delta''};x:=P),(\rho;x:=a),\Delta''}.$$

Hence indeed

$$(\Delta_1 \vdash M) \in X.$$

(SAT4) Let $(\Delta' \vdash M') \in X$ and $M \to_k M'$ and $(\Delta' \vdash M) \in SN^{\Delta}_{\rho(\Pi x:M_1.M_2)}$. Then obviously $\Delta' \vdash M : \rho(\Pi x : M_1.M_2)$ and $\Delta \subseteq \Delta'$. Let

$$\Delta'' \supseteq \Delta', \quad (\Delta'' \vdash a) \in [\Gamma \vdash M_1]_{\xi|_{\Delta''},\rho,\Delta''},\ P \in V_{\Delta''}(a).$$

By assumption

$$(\Delta'' \vdash M'a) \in [\Gamma, x : M_1 \vdash M_2]_{(\xi|_{\Delta''};x:=P),(\rho;x:=a)\Delta''}.$$

Note that $Ma \to_k M'a$. By Corollary 38 the term Ma is strongly normalizing. Thus $(\Delta'' \vdash Ma) \in SN^{\Delta''}_{(\rho;x:=a)(M_2)}$. As $[\Gamma, x : M_1 \vdash M_2]_{(\xi|_{\Delta''};x:=P),(\rho;x:=a),\Delta''}$ is a saturated set then $(\Delta' \vdash Ma) \in [\Gamma, x : M_1 \vdash M_2]_{(\xi|_{\Delta''};x:=P),(\rho;x:=a),\Delta''}$.

We have proved that $X \in SAT^{\Delta}_{\rho(\Pi x:M_1.M_2)}$. If $\Delta' \supseteq \Delta$ then the equality

$$([\Gamma \vdash \Pi x : M_1.M_2]_{\xi,\rho,\Delta})\,|_{\Delta'} = [\Gamma \vdash \Pi x : M_1.M_2]_{\xi|_{\Delta'},\rho,\Delta'}$$

holds because $\xi|_{\Delta'}|_{\Delta''} = \xi|_{\Delta''}$. □

Lemma 80. *Let Γ, Δ be two contexts. If $\Gamma \vdash M : A$, where M is a large term, and $\langle \xi, \rho \rangle$ is a constructor valuation which satisfies Γ at Δ, then*

- $[\Gamma \vdash M]_{\xi,\rho,\Delta} \in V_\Delta(\rho(M))$,
- *if* $\Delta' \supseteq \Delta$ *then* $[\Gamma \vdash M]_{\xi,\rho,\Delta}|_{\Delta'} = [\Gamma \vdash M]_{\xi|_{\Delta'},\rho,\Delta}$.

Proof. Induction with respect to the structure of the derivation of $\Gamma \vdash M : A$.
Case 1: The last rule used was the rule (Var). The conclusion is obvious by the definition of constructor valuation.
Case 2: The last rule used was the rule (Weak). Easy induction with respect to the structure of M.
Case 3: The last rule used was the rule (Conv). We have

$$\frac{\Gamma \vdash M : A' \qquad \Gamma \vdash A : s \qquad A =_{\beta\iota} A'}{\Gamma \vdash M : A}$$

By the induction hypothesis $[\Gamma \vdash M]_{\xi,\rho,\Delta} \in V_\Delta(\rho(A'))$. By Lemma 41 it holds that $V_\Gamma(\rho(A)) = V_\Gamma(\rho(A'))$. Thus $[\Gamma \vdash M]_{\xi,\rho,\Delta} \in V_\Delta(\rho(A))$. Moreover by the induction hypothesis $[\Gamma \vdash M]_{\xi,\rho,\Delta}|_{\Delta'} = [\Gamma \vdash M]_{\xi|_{\Delta'},\rho,\Delta'}$.
Case 4: The last rule used was the rule (App).

$$\frac{\Gamma \vdash M_1 : \Pi x : A_1.A_2 \qquad \Gamma \vdash M_2 : A_1}{\Gamma \vdash M_1 M_2 : A_2[x := M_2]}$$

Suppose A_1 is a large type. The other case is similar. Then

$$[\Gamma \vdash M_1 M_2]_{\xi,\rho,\Delta} = [\Gamma \vdash M_1]_{\xi,\rho,\Delta}(\Delta \vdash \rho(M_2), [\Gamma \vdash M_2]_{\xi,\rho,\Delta}).$$

By the induction hypothesis

$$[\Gamma \vdash M_1]_{\xi,\rho,\Delta} \in V_\Delta(\rho(M_1)) \text{ and } [\Gamma \vdash M_2] \in V_\Delta(\rho(M_2)).$$

Then $[\Gamma \vdash M_1]_{\xi,\rho,\Delta}$ is a function with the domain

$$\{(\Delta' \vdash a, P) \mid \Delta' \supseteq \Delta, \Delta' \vdash a : \rho(A_1), P \in V_{\Delta'}(a)\}$$

such that $[\Gamma \vdash M_1]_{\xi,\rho,\Delta}(\Delta' \vdash a, P) \in V_\Delta(\rho(M_1)a)$. By Lemma 41 we have $\Delta \vdash \rho(M_2) : \rho(A_1)$ and thus

$$[\Gamma \vdash M_1 M_2]_{\xi,\rho,\Delta} \in V_\Delta(\rho(M_2)\rho(M_2)) = V_\Delta(\rho(M_1 M_2)).$$

Now

$$\begin{aligned}
&([\Gamma \vdash M_1 M_2]_{\xi,\rho,\Delta})|_{\Delta'} \\
&\quad = ([\Gamma \vdash M_1]_{\xi,\rho,\Delta}(\Delta \vdash \rho(M_2), [\Gamma \vdash M_2]_{\xi,\rho,\Delta}))|_{\Delta'} \\
&\quad = [\Gamma \vdash M_1]_{\xi,\rho,\Delta}(\Delta' \vdash \rho(M_2), [\Gamma \vdash M_2]_{\xi,\rho,\Delta}|_{\Delta'}) \qquad \text{(by def. of } V_\Delta(\rho(M_1))) \\
&\quad = [\Gamma \vdash M_1]_{\xi,\rho,\Delta}|_{\Delta'}(\Delta' \vdash \rho(M_2), [\Gamma \vdash M_2]_{\xi,\rho,\Delta}|_{\Delta'}) \\
&\quad = [\Gamma \vdash M_1]_{\xi|_{\Delta'},\rho,\Delta'}(\Delta' \vdash \rho(M_2), [\Gamma \vdash M_2]_{\xi|_{\Delta'},\rho,\Delta'}) \\
&\quad = [\Gamma \vdash M_1 M_2]_{\xi|_{\Delta'},\rho,\Delta'}.
\end{aligned}$$

Case 5: The last rule used was the rule (Abs).

$$\frac{\Gamma, x : A_1 \vdash M_1 : A_2}{\Gamma \vdash \lambda x : A_1.M_1 : (\Pi x : A_1.A_2)}$$

If $(\Delta' \vdash a) \in T^{\Delta}_{\rho(A_1)}$, $P \in V_{\Delta'}(a)$, then $\langle(\xi|_{\Delta'}; x := P), (\rho; x := a)\rangle$ is a constructor valuation which satisfies $(\Gamma, x : A_1)$ at Δ'. Thus by the induction hypothesis

$$[\Gamma, x : A_1 \vdash M_1]_{(\xi|_{\Delta'};x:=P),(\rho;x:=a),\Delta'} \in V_{\Delta'}((\rho; x := a)(M_1)).$$

By the definition of $V_{\Delta'}((\rho; x := a)(M_1))$

$$[\Gamma \vdash \lambda x : A_1.M_1]_{\xi,\rho,\Delta} = \lambda\!\!\lambda(\Delta' \vdash a, P) : \overline{T}^{\Delta}_{\rho(A_1)}.[\Gamma, x : A_1 \vdash M_1]_{(\xi|_{\Delta'};x:=P),(\rho;x:=a),\Delta'}$$

is a function with the domain

$$\{(\Delta' \vdash a, P) \mid \Delta' \supseteq \Delta, \Delta' \vdash a : \rho(A_1), P \in V_{\Delta'}(a)\}$$

such that

$$[\Gamma \vdash \lambda x : A_1.M_1]_{\xi,\rho,\Delta}(\Delta' \vdash a, P) \in V_{\Delta'}(\rho(\lambda x : A.M_1)a) = V_{\Delta'}((\rho; x := a)(M_1)).$$

Moreover, by Lemma 77 if $a =_{\beta\iota} a'$ then

$$[\Gamma \vdash \lambda x : A_1.M_1]_{\xi,\rho,\Delta}(\Delta' \vdash a, P) = [\Gamma \vdash \lambda x : A_1.M_1]_{\xi,\rho,\Delta}(\Delta' \vdash a', P).$$

By the induction hypothesis if $\Delta \subseteq \hat{\Delta} \subseteq \Delta'$ then

$$\begin{aligned}
&([\Gamma \vdash \lambda x : A_1.M_1]_{\xi,\rho,\Delta}(\hat{\Delta} \vdash a, P))|_{\Delta'} \\
&\quad = ([\Gamma, x : A_1 \vdash M_1]_{(\xi|_{\hat{\Delta}};x:=P),(\rho;x:=a),\hat{\Delta}})|_{\Delta'} \\
&\quad = [\Gamma, x : A_1 \vdash M_1]_{(\xi|_{\hat{\Delta}}|_{\Delta'};x:=P|_{\Delta'}),(\rho;x:=a),\Delta'} \\
&\quad = [\Gamma, x : A_1 \vdash M_1]_{(\xi|_{\Delta'};x:=P|_{\Delta'}),(\rho;x:=a),\Delta'} \\
&\quad = [\Gamma \vdash \lambda x : A_1.M_1]_{\xi,\rho,\Delta}(\Delta' \vdash a, P|_{\Delta'})
\end{aligned}$$

Thus $[\Gamma \vdash \lambda x : A_1.M_1]_{\xi,\rho,\Delta} \in V_{\Delta}(\rho(\lambda x : A_1.M_1))$.

Now suppose $\Delta' \supseteq \Delta$. Then

$$[\Gamma \vdash \lambda x : A_1.M_1]_{\xi,\rho,\Delta}|_{\Delta'} = \lambda\!\!\lambda(\hat{\Delta} \vdash a, P) : T^{\Delta'}_{\rho(A_1)} \times V_{\hat{\Delta}}(a).[\Gamma, x : A_1 \vdash M_1]_{(\xi|_{\hat{\Delta}};x:=P),(\rho;x:=a),\hat{\Delta}}$$

and

$$[\Gamma \vdash \lambda x : A_1.M_1]_{\xi|_{\Delta'},\rho,\Delta'} = \lambda\!\!\lambda(\hat{\Delta} \vdash a, P) : T^{\Delta'}_{\rho(A_1)} \times V_{\hat{\Delta}}(a).$$
$$[\Gamma, x : A_1 \vdash M_1]_{(\xi|_{\Delta'}|_{\hat{\Delta}};x:=P),(\rho;x:=a),\hat{\Delta}}.$$

The equality follows from the fact that $(\xi|_{\Delta'})|_{\hat{\Delta}} = \xi|_{\hat{\Delta}}$.

Case 6: The last rule used was the rule (Prod). It follows from Lemma 79 and the induction hypothesis.

Case 7: The last rule used was the rule (Ind_{*^t}). It follows from Lemma 60, Lemma 77 and the induction hypothesis.

Case 8: The last rule used was the rule (Ind_{*^p}). It follows from Lemma 75, Lemma 77 and the induction hypothesis.

Case 9: The last rule used was the rule $(Intro_{*^t})$. Easy induction with respect to the structure of $C_n(I)$.

Case 10: The last rule used was the rule (Elim). It follows from Lemma 68, Lemma 77 and the induction hypothesis. □

5.4. Properties of the interpretation

In this section we show that the interpretations for convertible terms are equal. This property will be proved in several steps. First we show that one-step reductions are preserved. We prove the substitution lemma which entails that the interpretation preserves β-equality. The substitution lemma has two variants, depending on whether the variable being substituted is large (type, formula, kind, subset, etc. variable) or small (proof or small object variable). Later we deal with one-step ι-reduction. Then we combine the two results to get the desired conclusion.

5.4.1. Preserving beta equality

We begin the section by a few auxiliary lemmas which will be needed to prove the substitution property. In this section we assume that $(\Gamma, x : T, \Gamma')$ and Δ are two contexts, a term B is such that $\Gamma \vdash B : T$ and $\langle \xi, \rho \rangle$ is a constructor valuation which satisfies $(\Gamma, \Gamma'[B/x])$ at Δ. The constructor valuation $\langle \hat{\xi}, \hat{\rho} \rangle$ is defined as follows.

- If B is a large term then

$$\hat{\xi} = \xi; x := [\Gamma \vdash B]_{\xi,\rho,\Delta},$$
$$\hat{\rho} = \rho; x := \rho(B).$$

- If B is a small object or a proof then

$$\hat{\xi} = \xi, x := \emptyset,$$
$$\hat{\rho} = \rho; x := \rho(B).$$

Lemma 81. *The pair $\langle \hat{\xi}, \hat{\rho} \rangle$ is a constructor valuation which satisfies $(\Gamma, x : T, \Gamma')$ at Δ.*

Proof. Suppose $(y : \tau) \in (\Gamma, x : T, \Gamma')$. We will show that $\Delta \vdash \hat{\rho}(y) : \hat{\rho}(\tau)$ and $\hat{\xi}(y) \in V_\Delta(\hat{\rho}(y))$. There are three cases:

- $(y : \tau) \in \Gamma$. Then for every variable $z \in FV(\tau)$ and for $z = y$ it holds that $\hat{\rho}(z) = \rho(z)$ and (if z is a large variable) we have $\hat{\xi}(z) = \xi(z)$. The conditions $\Delta \vdash \hat{\rho}(y) : \hat{\rho}(\tau)$ and $\hat{\xi}(y) \in V_\Delta(\hat{\rho}(y))$ follow from the assumption that $\langle \xi, \rho \rangle$ is a constructor valuation which satisfies Γ at Δ.

- $y = x$ and $\tau = T$. We only consider the case when B is a large object; the other case is similar. By Lemma 80 we have $[\Gamma \vdash B]_{\xi,\rho,\Delta} \in V_\Delta(\rho(B))$. Thus

$$\hat{\xi}(y) = [\Gamma \vdash B]_{\xi,\rho,\Delta} \in V_\Delta(\rho(B)) = V_\Delta(\hat{\rho}(y)).$$

  Moreover by Lemma 50 and the fact that $\rho(z) = \hat{\rho}(z)$ for $z \in FV(\tau)$ it holds that $\Delta \vdash \rho(B) : \rho(\tau)$. Thus $\Delta \vdash \hat{\rho}(x) : \hat{\rho}(\tau)$.

- $(y : \tau) \in \Gamma'$. We only consider the case when y is a large object; the other case is similar. Recall that $\langle \xi, \rho \rangle$ is a constructor valuation which satisfies $(\Gamma, \Gamma'[B/x])$ at Δ. Thus $\Delta \vdash \rho(y) : \rho(\tau[B/x])$. But $\hat{\rho}(y) = \rho(y)$ and $\hat{\rho}(\tau) = \rho(\tau[B/x])$. Thus $\Delta \vdash \hat{\rho}(y) : \hat{\rho}(\tau)$. Moreover

$$\hat{\xi}(y) = \xi(y) \in V_\Delta(\rho(\tau[B/x])) = V_\Delta(\hat{\rho}(\tau)).$$

□

The next lemma proves a substitution property for the function $F_{\Gamma,I,\xi,\rho,\Delta}$, introduced on page 58, used in the definition of interpretation for an inductive type.

Lemma 82. *Let $\Gamma, x : T, \Gamma' \vdash I : T_2$ and suppose I is an inductive type. Suppose that for each subterm M of I we have*

$$[\Gamma, \Gamma'[B/x] \vdash M[B/x]]_{\xi,\rho,\Delta} = [\Gamma, x : T, \Gamma' \vdash M]_{\hat{\xi},\hat{\rho},\Delta}.$$

Then

$$F_{(\Gamma,\Gamma'[B/x]),I[B/x],\xi,\rho,\Delta} = F_{(\Gamma,x:T,\Gamma'),I,\hat{\xi},\hat{\rho},\Delta}.$$

Proof. By Lemma 81 the pair $\langle\hat{\xi},\hat{\rho}\rangle$ is a constructor valuation which satisfies $(\Gamma, x : T, \Gamma')$ at Δ. We will prove that

$$F_{(\Gamma,\Gamma'[B/x]),I[B/x],\xi,\rho,\Delta} = F_{(\Gamma,x:T,\Gamma'),I,\hat{\xi},\hat{\rho},\Delta}.$$

Note that $\rho(I[B/x]) = \hat{\rho}(I)$ and thus

$$SAT^{\Delta}_{\rho(I[B/x])} = SAT^{\Delta}_{\hat{\rho}(I)}.$$

The domains of functions $F_{(\Gamma,\Gamma'[B/x]),I[B/x],\xi,\rho,\Delta}$ and $F_{(\Gamma,x:T,\Gamma'),I,\hat{\xi},\hat{\rho},\Delta}$ are the same. Recall that

$$\begin{aligned} F_{(\Gamma,\Gamma'[B/x]),I[B/x],\xi,\rho,\Delta}(S) = \Big(\bigcap SAT^{\Delta}_{\rho(I[B/x])}\Big) \cup \{(\Delta' \vdash u) \in SN^{\Delta}_{\rho(I[B/x])} \mid \\ \text{if } \Delta'' \supseteq \Delta' \text{ and } \Delta'' \vdash u \to^{*}_{k} \mathrm{Constr}(n, X)\vec{N}, \text{ and } C_n(X) = \Pi\vec{x} : \vec{T}.X \\ \text{then for every } j \text{ we have } (\Delta'' \vdash N_j) \in \mathit{Interp}((\Gamma, \Gamma'[B/x])^j \vdash T_j)_{\xi^j,\rho^j,\Delta'',X,S|_{\Delta''}}\} \end{aligned}$$

and

$$\begin{aligned} F_{(\Gamma,x:T,\Gamma'),I,\hat{\xi},\hat{\rho},\Delta}(S) = \Big(\bigcap SAT^{\Delta}_{\hat{\rho}(I)}\Big) \cup \{(\Delta' \vdash u) \in SN^{\Delta}_{\hat{\rho}(I)} \mid \\ \text{if } \Delta'' \supseteq \Delta' \text{ and } \Delta'' \vdash u \to^{*}_{k} \mathrm{Constr}(n, X)\vec{N}, \text{ and } C_n(X) = \Pi\vec{x} : \vec{T}.X \\ \text{then for every } j \text{ we have } (\Delta'' \vdash N_j) \in \mathit{Interp}((\Gamma, x : T, \Gamma')^j \vdash T_j)_{\hat{\xi}^j,\hat{\rho}^j,\Delta'',X,S|_{\Delta''}}\} \end{aligned}$$

The conclusion follows from the fact that by Lemma 54 and the assumption we have

$$\begin{aligned} &\mathit{Interp}((\Gamma, \Gamma'[B/x])^j \vdash T_j[B/x])_{\xi^j,\rho^j,\Delta'',X,S|_{\Delta''}} \\ &\quad = [(\Gamma, \Gamma'[B/x])^j \vdash T_j[B/x]]_{(\xi^j;X:=S|_{\Delta''}),(\rho^j;X:=I[B/x]),\Delta''} \\ &\quad = [(\Gamma, x : T, \Gamma')^j \vdash T_j]_{(\hat{\xi}^j;X:=S|_{\Delta''}),(\hat{\rho}^j;X:=I),\Delta''} \\ &\quad = \mathit{Interp}((\Gamma, x : T, \Gamma')^j \vdash T_j])_{\hat{\xi}^j,\hat{\rho}^j,\Delta'',X,S|_{\Delta''}}. \end{aligned}$$

□

Now we prove a substitution property for the function G, introduced on page 73, used in the definition of interpretation for elimination terms.

Lemma 83. *Let $\Gamma, x : T, \Gamma' \vdash I : T_2$ and suppose I is an inductive type. Suppose α is an ordinal number such that*

$$(\Delta \vdash \mathrm{Constr}(j, J)\vec{N}) \in F^{\alpha}_{(\Gamma,\Gamma'[B/x]),I[B/x],\xi,\rho,\Delta}(\bigcap SAT^{\Delta}_{\rho(I[B/x])})$$

and for each $\alpha' < \alpha$ *if* $S = F^{\alpha'}_{(\Gamma,\Gamma'[B/x]),I[B/x],\xi,\rho,\Delta}(\bigcap SAT^{\Delta}_{\rho(I[B/x])})$ *then*

$$G_{(\Gamma,\Gamma'[B/x]),I[B/x],Q[B/x],\vec{f}[B/x],\xi,\rho,\Delta}(S) = G_{(\Gamma,x:T,\Gamma'),I,Q,\vec{f},\hat{\xi},\hat{\rho},\Delta}(S).$$

Then for every $(\Delta \vdash M) \in T^{\Delta}_{\rho(I)}$ *it holds that*

$$g_{\Gamma,I[B/x],Q[B/x],\vec{f}[B/x],\xi,\rho,\Delta}[M,U,C_j(I[B/x]),\vec{N}] = g_{\Gamma,I,Q,\vec{f},\hat{\xi},\hat{\rho},\Delta}[M,U,C_j(I),\vec{N}].$$

Proof. Note that by Lemma 82

$$S = F^{\alpha'}_{(\Gamma,\Gamma'[B/x]),I[B/x],\xi,\rho,\Delta}(SAT^{\Delta}_{\rho(I[B/x])}) = F^{\alpha'}_{(\Gamma,x:T,\Gamma'),I,\hat{\xi},\hat{\rho},\Delta}(SAT^{\Delta}_{\hat{\rho}(I)}).$$

We prove by induction with respect to the structure of $C_j(X)$ that

$$g_{\Gamma,I[B/x],Q[B/x],\vec{f}[B/x],\xi,\rho,\Delta}[M,U,C_j(I[B/x]),\vec{N}] = g_{\Gamma,I,Q,\vec{f},\hat{\xi},\hat{\rho},\Delta}[M,U,C_j(I),\vec{N}].$$

- If $C(X) = X$ then

$$g_{\Gamma,I[B/x],Q[B/x],\vec{f}[B/x],\xi,\rho,\Delta}[M,U,C(I[B/x]),\vec{N}] = \epsilon = g_{\Gamma,I,Q,\vec{f},\hat{\xi},\hat{\rho},\Delta}[M,U,C_j(I),\vec{N}].$$

- If $C(X) = \Pi x : T.D(X)$, $X \notin FV(T)$ then $\vec{N} = N_0 :: \vec{N'}$ and $U = \langle m, \vec{U}\rangle$ and $\vec{U} = U_0 :: \vec{U'}$ and

$$\begin{aligned}&g_{\Gamma,I[B/x],Q[B/x],\vec{f}[B/x],\xi,\rho,\Delta}[M,U,C(I[B/x]),\vec{N}]\\ &\qquad = (\Delta \vdash N_0, U_0) :: g_{\Gamma,I[B/x],Q[B/x],\vec{f}[B/x],\xi,\rho,\Delta]}[M,\langle m,\vec{U}\rangle,D(I[B/x]),\vec{N'}].\end{aligned}$$

  and

$$g_{\Gamma,I,Q,\vec{f},\hat{\xi},\hat{\rho},\Delta}[M,U,C(I),\vec{N}] = (\Delta \vdash N_0, U_0) :: g_{\Gamma,I,Q,\vec{f},\hat{\xi},\hat{\rho},\Delta}[M,\langle m,\vec{U}\rangle,D(I),\vec{N'}].$$

  The conclusion follows from the induction hypothesis.

- If $C(X) = \Pi x : T.D(X)$, $X \in FV(T)$ and $T = \Pi\vec{x} : \vec{t}.X$ then $\vec{N} = N_0 :: \vec{N'}$ and $U = \langle m, \vec{U}\rangle$ and $\vec{U} = U_0 :: \vec{U'}$. Recall that

$$\begin{aligned}g_{\Gamma,I[B/x],Q[B/x],\vec{f}[B/x],\xi,\rho,\Delta}[M,U,C(I[B/x]),\vec{N}] &= (\Delta \vdash N_0, U_0) :: (\Delta \vdash e, C)\\ &:: g_{\Gamma,I[B/x],Q[B/x],\vec{f}[B/x],\xi,\rho,\Delta}[M,\langle m,\vec{U}\rangle,D(I[B/x]),\vec{N'}]\end{aligned}$$

  where

$$e = \lambda\vec{x} : \rho(\vec{t}[B/x]).\text{Elim}(\rho(I[B/x]),\rho(Q[B/x]),N_0\vec{x})\{\rho(\vec{f}[B/x])\}$$

  and P is a function which for an appropriate sequence of arguments $(\Delta_i, x_i, P_i)_{i=0}^{p}$ at $\langle\xi,\rho\rangle$ in Δ returns the value

$$\begin{aligned}P((\Delta_i,x_i,P_i)_{i=0}^{p}) = G&_{(\Gamma,\Gamma'[B/x],\vec{x}:\vec{t}[B/x]),I[B/x],Q[B/x],\vec{f}[B/x],\xi,\rho,\Delta_p}\\ &(pred_{(\Gamma,\Gamma'[B/x]\vec{x}:\vec{t}[B/x]),I,\xi_p,\rho_p,\Delta_p}(\Delta_p \vdash \text{Constr}(j,X)\vec{N}))\\ &(\Delta \vdash N_0\vec{x}, U_0(\Delta_i,x_i,P_i)_{i=0}^{p})\end{aligned}$$

and

$$g_{\Gamma,I,Q,\vec{f},\hat{\xi},\hat{\rho},\Delta}[M,U,C(I),\vec{N}] = (\Delta \vdash N_0,U_0)(\Delta \vdash \hat{e},\hat{P}) :: g_{\Gamma,I,Q,\vec{f},\hat{\xi},\hat{\rho},\Delta}[M,\langle m,\vec{U}\rangle,D(I),\vec{N}']$$

with

$$\hat{e} = \lambda\vec{x} : \hat{\rho}(\vec{t}).\mathrm{Elim}(\hat{\rho}(I),\hat{\rho}(Q),N_0\vec{x})\{\hat{\rho}(\vec{f})\},$$

and P is a function which for an appropriate sequence of arguments $(\Delta_i,x_i,P_i)_{i=0}^{p}$ at $\langle\xi,\rho\rangle$ in Δ returns the value

$$\hat{P}((\Delta_i,x_i,P_i)_{i=0}^{p}) = G_{(\Gamma,x:T,\Gamma',\vec{x}:\vec{t}),I,Q,\vec{f},\hat{\xi},\hat{\rho},\Delta_p} (pred_{(\Gamma,x:T,\Gamma',\vec{x}:\vec{t}),I,\hat{\xi}_p,\hat{\rho}_p,\Delta_p}(\Delta_p \vdash \mathrm{Constr}(j,X)\vec{N}) (\Delta \vdash N_0\vec{x},U_0(\Delta_i,x_i,P_i)_{i=0}^{p})).$$

Observe that by the induction hypothesis

$$g_{\Gamma,I[B/x],Q[B/x],\vec{f}[B/x],\xi,\rho,\Delta]}[M,U,D(I[B/x]),\vec{N}'] = g_{\Gamma,I,Q,\vec{f},\hat{\xi},\hat{\rho},\Delta}[M,U,D(I),\vec{N}'].$$

Moreover it is easy to note that $e = \hat{e}$ and by Lemma 82

$$pred_{(\Gamma,\Gamma'[B/x]\vec{x}:\vec{t}[B/x]),I,\xi_p,\rho_p,\Delta_p}(\Delta_p \vdash \mathrm{Constr}(j,X)\vec{N}) = pred_{(\Gamma,x:T,\Gamma',\vec{x}:\vec{t}),I,\hat{\xi}_p,\hat{\rho}_p,\Delta_p}(\Delta_p \vdash \mathrm{Constr}(j,X)\vec{N}).$$

Then by the assumption

$$G_{(\Gamma,\Gamma'[B/x],\vec{x}:\vec{t}[B/x]),I[B/x],Q[B/x],\vec{f}[B/x],\xi,\rho,\Delta_p} (pred_{(\Gamma,\Gamma'[B/x]\vec{x}:\vec{t}[B/x]),I,\xi_p,\rho_p,\Delta_p}(\Delta_p \vdash \mathrm{Constr}(j,X)\vec{N})) = G_{(\Gamma,x:T,\Gamma',\vec{x}:\vec{t}),I,Q,\vec{f},\hat{\xi},\hat{\rho},\Delta_p}(pred_{(\Gamma,x:T,\Gamma',\vec{x}:\vec{t}),I,\hat{\xi}_p,\hat{\rho}_p,\Delta_p}(\Delta_p \vdash \mathrm{Constr}(j,X)\vec{N})). \quad \square$$

We now prove the Substitution Lemma.

Lemma 84. *Let A be a large term such that*

$$\Gamma, x : T, \Gamma' \vdash A : T_2.$$

Then

$$[\Gamma,\Gamma'[B/x] \vdash A[B/x]]_{\xi,\rho,\Delta} = [\Gamma, x : T, \Gamma' \vdash A]_{\hat{\xi},\hat{\rho},\Delta}.$$

Proof. By Lemma 81 the pair $\langle\hat{\xi},\hat{\rho}\rangle$ is a constructor valuation which satisfies $(\Gamma, x : T, \Gamma')$ at Δ. We proceed by induction with respect to $m(T_2)$.

The cases when A is a sort, a variable, an application, an abstraction, a product, or an inductive object are an easy consequence of the induction hypothesis. We consider the remaining cases.

Case 1: inductive type. If $A = \mathrm{Ind}(X : *^t)\{\vec{C}\}$ then $A[B/x] = \mathrm{Ind}(X : *^t)\{\vec{C}[B/x]\}$. From the induction hypothesis and Lemma 82 we get

$$[\Gamma, \Gamma'[B/x] \vdash \mathrm{Ind}(X : *^t)\{\vec{C}[B/x]\}]_{\xi,\rho,\Delta} = \mathit{lfp}(F_{(\Gamma,\Gamma'[B/x]),A[B/x],\xi,\rho,\Delta})$$
$$= \mathit{lfp}(F_{(\Gamma,x:T,\Gamma'),A,\hat{\xi},\hat{\rho},\Delta}) = [\Gamma, x : T, \Gamma' \vdash \mathrm{Ind}(X : *^t)\{\vec{C}\}]_{\hat{\xi},\hat{\rho},\Delta}.$$

Case 2: elimination. Suppose $A = \mathrm{Elim}(I, Q, M)\{\vec{f}\}$. For simplicity, we only consider the case when I is a large inductive type. The other case is similar. Recall that

$$[\Gamma, \Gamma'[B/x] \vdash A[B/x]]_{\xi,\rho,\Delta} = G_{(\Gamma,\Gamma'[B/x]),I[B/x],Q[B/x],\vec{f}[B/x],\xi,\rho,\Delta}([\Gamma, \Gamma'[B/x] \vdash I[B/x]]_{\xi,\rho,\Delta})$$
$$(\Delta \vdash \rho(M[B/x]), [\Gamma, \Gamma'[B/x] \vdash M[B/x]]_{\xi,\rho,\Delta})$$

and

$$[\Gamma, x : T, \Gamma' \vdash A]_{(\xi;x:=[\Gamma\vdash B]_{\xi,\rho,\Delta}),(\rho;x:=\rho(B)),\Delta}$$
$$= G_{(\Gamma,x:=T,\Gamma'),I,Q,\vec{f},\hat{\xi},\hat{\rho},\Delta}([\Gamma, x : T, \Gamma' \vdash I]_{\hat{\xi},\hat{\rho},\Delta})(\Delta \vdash \hat{\rho}(M), [\Gamma, x : T, \Gamma' \vdash M]_{\hat{\xi},\hat{\rho},\Delta}).$$

Note that by the induction hypothesis

$$[\Gamma, \Gamma'[B/x] \vdash I[B/x]]_{\xi,\rho,\Delta} = [\Gamma, x : T, \Gamma' \vdash I]_{\hat{\xi},\hat{\rho},\Delta}$$
$$[\Gamma, \Gamma'[B/x] \vdash M[B/x]]_{\xi,\rho,\Delta} = [\Gamma, x : T, \Gamma' \vdash M]_{\hat{\xi},\hat{\rho},\Delta},$$

and for any i we have

$$[\Gamma, \Gamma'[B/x] \vdash f_i[B/x]]_{\xi,\rho,\Delta} = [\Gamma, x : T, \Gamma' \vdash f_i]_{\hat{\xi},\hat{\rho},\Delta}.$$

Moreover $\rho(M[B/x]) = \hat{\rho}(M)$. Thus the arguments in the above calls to the functions $G_{(\Gamma,\Gamma'[B/x]),I[B/x],Q[B/x],\vec{f}[B/x],\xi,\rho,\Delta}$ and $G_{(\Gamma,x:=T,\Gamma'),I,Q,\vec{f},\hat{\xi},\hat{\rho},\Delta}$ are the same. To get the conclusion it is enough to show that

$$G_{(\Gamma,\Gamma'[B/x]),I[B/x],Q[B/x],\vec{f}[B/x],\xi,\rho,\Delta} = G_{(\Gamma,x:=T,\Gamma'),I,Q,\vec{f},\hat{\xi},\hat{\rho},\Delta}.$$

By the induction hypothesis (Lemma 82)

$$\mathcal{D}_{(\Gamma,\Gamma'[B/x]),I[B/x],\xi,\rho,\Delta} = \mathcal{D}_{(\Gamma,x:T,\Gamma'),I,\hat{\xi},\hat{\rho},\Delta}$$

so the domains of both functions are equal. Suppose

$$S \in \mathcal{D}_{(\Gamma,\Gamma'[B/x]),I[B/x],\xi,\rho,\Delta},\ \Delta' \vdash m : \rho(I[B/x]),\ \text{and } U \in V_{\Delta'}(m).$$

We will prove

$$G_{(\Gamma,\Gamma'[B/x]),I[B/x],Q[B/x],\vec{f}[B/x],\xi,\rho,\Delta}(S)(\Delta' \vdash m, U) = G_{(\Gamma,x:=T,\Gamma'),I,Q,\vec{f},\hat{\xi},\hat{\rho},\Delta}(S)(\Delta' \vdash m, U).$$

Note that if $S \in \mathcal{D}_{(\Gamma,\Gamma'[B/x]),I[B/x],\xi,\rho,\Delta}$ then there exists α such that

$$S = F^{\alpha}_{(\Gamma,\Gamma'[B/x]),I[B/x],\xi,\rho,\Delta}(\bigcap SAT^{\Delta}_{\rho(I[B/x])}).$$

We proceed by induction with respect to α. If $\alpha = 0$ then $S = \bigcap SAT^{\Delta}_{\rho(I[B/x])}$. By Lemma 35 it is not possible that there exists Δ'' and a term $\mathrm{Constr}(j,X))\vec{N}$ such that $(\Delta'' \vdash \mathrm{Constr}(j,X))\vec{N} \in S$ and $m =_{\beta\iota} \mathrm{Constr}(j,X)\vec{N}$. Thus

$$G_{(\Gamma,\Gamma'[B/x]),I[B/x],Q[B/x],\vec{f}[B/x],\xi,\rho,\Delta}(S)(\Delta' \vdash m, U) \\ = Min_{\Delta}(\mathrm{Elim}(\rho(I[B/x]), \rho(Q[B/x]), m)\{\rho(\vec{f}[B/x])\})$$

and

$$G_{(\Gamma,x:T,\Gamma'),I,Q,\vec{f},\hat{\xi},\hat{\rho},\Delta}(S)(\Delta' \vdash m, U) = Min_{\Delta}(\mathrm{Elim}(\hat{\rho}(I), \hat{\rho}(Q), m)\{\hat{\rho}(\vec{f})\}).$$

The conclusion follows from the fact that for all M we have $\rho(M[B/x]) = \hat{\rho}(M)$.

If $\alpha = \alpha' + 1$ then there are two cases. The first case: $\pi_1(U) = j$ and $m =_{\beta\iota} \mathrm{Constr}(j,X)\vec{N}$, $(\Delta \vdash \mathrm{Constr}(j,X)\vec{N}) \in S$. Then

$$G_{(\Gamma,\Gamma'[B/x]),I[B/x],Q[B/x],\vec{f}[B/x],\xi,\rho,\Delta}(S)(\Delta' \vdash m, U) = \bigsqcup(Base \cup Min)$$

where

- $Base$ consists of all values of the form

$$[(\Gamma,\Gamma'[B/x]) \vdash f_j]_{\xi|_{\Delta'},\rho,\Delta''} \\ \cdot g_{(\Gamma,\Gamma'[B/x]),I[B/x],Q[B/x],\vec{f}[B/x],\xi,\rho,\Delta''}[\mathrm{Constr}(j,J)\vec{M}, U, C_j(I[B/x]), \vec{M}]$$

  such that $\Delta'' \supseteq \Delta'$, $\vec{f}[B/x] =_{\beta\iota} \mathrm{Constr}(j,J)\vec{M}$ and $(\Delta'' \vdash \mathrm{Constr}(j,J)\vec{M}) \in S$

- Min consists of all values of the form

$$Min_{\Delta''}(\mathrm{Elim}(\rho(I[B/x]), \rho(Q[B/x]), \vec{f}[B/x])\{\rho(\vec{f})\})$$

  such that $\Delta'' \supseteq \Delta'$, $\vec{f}[B/x] =_{\beta\iota} \mathrm{Constr}(j,J)\vec{M}$ and

$$(\Delta'' \vdash \mathrm{Constr}(j,J)\vec{M}) \in T^{\Delta}_{\rho(I[B/x])} - S.$$

Moreover

$$G_{(\Gamma,x:T,\Gamma'),I,Q,\vec{f},\hat{\xi},\hat{\rho},\Delta}(S)(\Delta' \vdash m, U) = \bigsqcup(Base_1 \cup Min_1)$$

where

- $Base_1$ consists of all values of the form

$$[(\Gamma,x:T,\Gamma') \vdash f_j]_{\xi|_{\Delta'},\rho,\Delta''} \\ \cdot g_{(\Gamma,x:T,\Gamma'),I[B/x],Q[B/x],\vec{f}[B/x],\xi,\rho,\Delta''}[\mathrm{Constr}(j,J)\vec{M}, U, C_j(I[B/x]), \vec{M}]$$

  such that $\Delta'' \supseteq \Delta'$, $\vec{f}[B/x] =_{\beta\iota} \mathrm{Constr}(j,J)\vec{M}$ and $(\Delta'' \vdash \mathrm{Constr}(j,J)\vec{M}) \in S$

- Min_1 consists of all values of the form

$$Min_{\Delta''}(\text{Elim}(\rho(I[B/x]), \rho(Q[B/x]), \vec{f}[B/x])\{\rho(\vec{f})\})$$

  such that $\Delta'' \supseteq \Delta'$, $\vec{f}[B/x] =_{\beta\iota} \text{Constr}(j, J)\vec{M}$ and

$$(\Delta'' \vdash \text{Constr}(j, J)\vec{M}) \in T^{\Delta}_{\rho(I[B/x])} - S.$$

By the induction hypothesis

$$[\Gamma, \Gamma'[B/x] \vdash f_i[B/x]]_{\xi,\rho,\Delta} = [\Gamma, x : T, \Gamma' \vdash f_i]_{\hat{\xi},\hat{\rho},\Delta}.$$

and by Lemma 83

$$g_{\Gamma,I[B/x],Q[B/x],\vec{f}[B/x],\xi,\rho,\Delta}[\text{Constr}(j, J, \vec{N}), U, C_j(I[B/x]), \vec{N}]$$
$$= g_{\Gamma,I,Q,\vec{f},\hat{\xi},\hat{\rho},\Delta}[\text{Constr}(j, J, \vec{N}), U, C_j(I), \vec{N}].$$

Thus the equality holds.

Otherwise $\pi_1(U) \neq j$ or there does not exist a term $\text{Constr}(j, X)\vec{N}$ such that $m =_{\beta\iota} \text{Constr}(j, X)\vec{N}$ and $(\Delta \vdash \text{Constr}(j, X)\vec{N}) \in S$. Then

$$G_{(\Gamma,\Gamma'[B/x]),I[B/x],Q[B/x],\vec{f}[B/x],\xi,\rho,\Delta}(S)(\Delta \vdash m, U)$$
$$= Min_{\Delta}(\text{Elim}(\rho(I[B/x]), \rho(Q[B/x]), m)\{\rho(\vec{f}[B/x])\})$$

and

$$G_{(\Gamma,x:T,\Gamma'),I,Q,\vec{f},\hat{\xi},\hat{\rho},\Delta}(S)(\Delta \vdash m, U) = Min_{\Delta}(\text{Elim}(\hat{\rho}(I), \hat{\rho}(Q), m)\{\hat{\rho}(\vec{f})\}).$$

The conclusion follows from the fact that $\rho(I[B/x]) = \hat{\rho}(I)$, $\rho(Q[B/x]) = \hat{\rho}(Q)$ and for all i we have $\rho(f_i[B/x]) = \hat{\rho}(f_i)$,

If α is a limit ordinal then

$$S = \bigcup_{\alpha'<\alpha} (F^{\alpha'}_{(\Gamma,\Gamma'[B/x]),I[B/x],\xi,\rho,\Delta}(\bigcap SAT^{\Delta}_{\rho(I[B/x])}))$$

and the conclusion follows from the induction hypothesis.

Case 3: inductive predicate. Suppose $A = \text{Ind}(X : A)\{\vec{C}\}$ where $A = \Pi\vec{z} : \vec{\tau}.*^p$. Then

$$[\Gamma, \Gamma'[B/x] \vdash A[B/x]]_{\xi,\rho,\Delta} = \mathit{lfp}(H_{(\Gamma,\Gamma'[B/x]),A[B/x],\xi,\rho,\Delta})$$

Recall that

$$H_{(\Gamma,\Gamma'[B/x]),A[B/x],\xi,\rho,\Delta}(S)((\Sigma_i, u_i, U_i)^n_{i=1}) =$$
$$(\bigcap SAT^{\Sigma_n}_{\rho(A[B/x])\vec{u}}) \cup h_{(\Gamma,\Gamma'[B/x]),A[B/x],\xi,\rho,\Delta}(S)((\Sigma_i, u_i, U_i)^n_{i=1})$$

and $h_{(\Gamma,\Gamma'[B/x]),A[B/x],\xi,\rho,\Delta}(S)((\Sigma_i, u_i, U_i)^n_{i=1})$ consists of simple sequents $(\Delta' \vdash m) \in SN^{\Sigma_n}_{\rho(A[B/x])\vec{u}}$ such that

for every context Δ'' and every term J such that $(\Delta'' \vdash J) \in SN^{\Sigma_n}_{\rho(A)}$ and $J =_{\beta\iota} \rho(A[B/x])$,

for every context Δ''' and every term Q such that $(\Delta''' \vdash Q) \in [(\Gamma, \Gamma'[B/x]) \vdash A]_{\xi|_{\Delta''},\rho,\Delta''}$,
for every $P \in V_{\Delta'''}(Q)$, ,
for every context Δ'''' and for every vector $\vec{f}$ such that

$(\Delta'''' \vdash f_i) \in [(\Gamma, \Gamma'[B/x]), q : A, X : A \vdash \triangle\{C_i(X), q\}]_{(\xi|_{\Delta''''};X:=S|_{\Delta''''};q:=P),(\rho;X:=J;q:=Q),\Delta''''}$

we have $(\Delta'''' \vdash \mathrm{Elim}(J, Q, \vec{u}, m)\{\vec{f}\}) \in P((\Sigma_i, u_i, U_i)_{i=1}^n)$.

The value

$$[\Gamma, x : T, \Gamma' \vdash A]_{\hat{\xi},\hat{\rho},\Delta} = \mathit{lfp}(H_{(\Gamma,x:T,\Gamma'),A,\hat{\xi},\hat{\rho},\Delta})$$

is defined in a similar way. By the induction hypothesis and the fact that for all M we have $\rho(M[B/x]) = \hat{\rho}(M)$ for every S we get that

$$H_{(\Gamma,\Gamma'[B/x]),A[B/x],\xi,\rho,\Delta}(S) = H_{(\Gamma,x:T,\Gamma'),A,\hat{\xi},\hat{\rho},\Delta}(S).$$

Hence the conclusion. □

5.4.2. Preserving iota equality

Let Γ, Δ be contexts. We say that $\langle \xi, \rho \rangle$ is *an object valuation which satisfies* Γ *at* Δ if $\langle \xi, \rho \rangle$ is a constructor valuation which satisfies Γ at Δ and if $(x : A) \in \Gamma$ then $(\Delta \vdash \rho(x)) \in [\Gamma \vdash A]_{\xi,\rho,\Delta}$.

Lemma 85. *Let* $S = F^{\alpha}_{\Gamma,I,\xi,\rho,\Delta}(\bigcap SAT^{\Delta}_{\rho(I)})$ *for a certain* α. *If*

$$(\Delta \vdash M) \in S \textit{ and } M =_{\beta\iota} \mathrm{Constr}(j, X)\vec{N}$$

then there exist terms X', $\vec{N}'$ *such that*

$$M =_{\beta\iota} \mathrm{Constr}(j, X')\vec{N}' \textit{ and } (\Delta \vdash \mathrm{Constr}(j, X')\vec{N}') \in S.$$

Proof. If $M =_{\beta\iota} \mathrm{Constr}(j, X)\vec{N}$ then by Lemma 35 we have $\alpha > 0$. By the Church-Rosser property there exists a term D such that $M \to^*_{\beta\iota} D$ and $\mathrm{Constr}(j, X)\vec{N} \to^*_{\beta\iota} D$. Note that $D = \mathrm{Constr}(j, X^*)\vec{N}^*$. Then $M \to^*_k \mathrm{Constr}(j, X')\vec{N}'$. By the definition of the operator $F_{\Gamma,I,\xi,\rho,\Delta}$ (see page 58) the set S is closed for the key reduction and thus

$$(\Delta \vdash \mathrm{Constr}(j, X')\vec{N}') \in S.$$ □

Lemma 86. *Suppose* $S, S' \in \mathcal{D}_{\Gamma,I,\xi,\rho,\Delta}$. *If* $S \subseteq S'$ *and* $(\Delta' \vdash m) \in S$ *then*

$$G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta' \vdash m, U) = G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S')(\Delta' \vdash m, U).$$

Proof. If $S \in \mathcal{D}_{\Gamma,I,\xi,\rho,\Delta}$ then $S = F^{\alpha}_{\Gamma,I,\xi,\rho,\Delta}(\bigcap SAT^{\Delta}_{\rho(I)})$. We proceed by induction with respect to α. There are two cases.

Case 1: If there are no J, $\vec{N}'$ such that $m =_{\beta\iota} \mathrm{Constr}(n, J)\vec{N}'$ then

$$G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta' \vdash m, U) = Min_{\Delta'}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\})$$

and

$$G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S')(\Delta' \vdash m, U) = Min_{\Delta'}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\})$$

and the conclusion is true. Note that if $\alpha = 0$ then $S = \bigcap SAT^{\Delta}_{\rho(I)}$ and by Lemma 35 this is the only possible case.

Case 2: By the preceding remark we may assume that $\alpha > 0$. Then $S \neq \bigcap SAT^{\Delta}_{\rho(I)}$ and we assume that the conclusion is true for every $S'' < S$ (that is if $S'' \leq S$ and $S'' \neq S$). If $S'' < S$ then $S'' = F^{\alpha''}_{\Gamma,I,Q,\xi,\rho,\Delta}(\bigcap SAT^{\Delta}_{\rho(I)})$. We may apply the induction hypothesis.

We have $m =_{\beta\iota} \mathrm{Constr}(n,J)\vec{N}$ and $(\Delta' \vdash m) \in S$. Then $m \rightarrow^*_k \mathrm{Constr}(n,J')\vec{N}'$ and $(\Delta' \vdash \mathrm{Constr}(n,J')\vec{N}') \in S$. Recall that in this case

$$G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta' \vdash m, U) = \bigsqcup \mathcal{F}_1$$

where $\mathcal{F}_1 = Base_1 \cup Min_1$ and

- $Base_1$ consists of all values of the form

  $$[\Gamma \vdash f_n]_{\xi|_{\Delta'},\rho,\Delta''} \cdot g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta''}[\mathrm{Constr}(n,J)\vec{M}, U, C_n(I), \vec{M}]$$

  such that $\Delta'' \supseteq \Delta'$, $m =_{\beta\iota} \mathrm{Constr}(n,J)\vec{M}$ and $(\Delta'' \vdash \mathrm{Constr}(n,J)\vec{M}) \in S$
- Min_1 consists of all values of the form

  $$Min_{\Delta''}(\mathrm{Elim}(\rho(I),\rho(Q),m)\{\rho(\vec{f})\})$$

  such that $\Delta'' \supseteq \Delta'$, $m =_{\beta\iota} \mathrm{Constr}(n,J)\vec{M}$ and

  $$(\Delta'' \vdash \mathrm{Constr}(n,J)\vec{M}) \in T^{\Delta}_{\rho(I)} - S.$$

The value $G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S')(\Delta' \vdash m, U) = \bigsqcup \mathcal{F}_2$ is defined similarly as the union of sets $Base_2$ and Min_2. We will use Lemma 63 to show that

$$\mathcal{F}_1 = G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta' \vdash m, U) = G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S')(\Delta' \vdash m, U) = \mathcal{F}_2.$$

First we show that $\mathcal{F}_2 \subseteq \mathcal{F}_1$. Let $C \in \mathcal{F}_2$. Then either $C \in Base_2$ or $C \in Min_2$. If $C \in Min_2$ then obviously $C \in Min_1$. If $C \in Base_2$ then there exist $\Delta'' \supseteq \Delta$ and n, J_1 and $\vec{M}_1$ such that

$$C = [\Gamma \vdash f_n]_{\xi|_{\Delta_1},\rho,\Delta_1} \cdot g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta_1}[\mathrm{Constr}(n,J_1)\vec{M}_1, U, C_n(I), \vec{M}_1]$$

and $m =_{\beta\iota} \mathrm{Constr}(n,J_1)\vec{M}_1$ and $(\Delta'' \vdash \mathrm{Constr}(n,J_1)\vec{M}_1) \in S'$. But then

$$m =_{\beta\iota} \mathrm{Constr}(n,J_1)\vec{M}_1 =_{\beta\iota} \mathrm{Constr}(n,J')\vec{N}' \text{ and } (\Delta'' \vdash \mathrm{Constr}(n,J')\vec{N}') \in S.$$

Observe that

$$\begin{aligned} C = [\Gamma \vdash f_n]_{\xi,\rho,\Delta''} \cdot g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta''}&[\mathrm{Constr}(n,J_1)\vec{M}_1, U, C_n(I), \vec{M}_1] \\ &= [\Gamma \vdash f_n]_{\xi,\rho,\Delta''} \cdot g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta''}[\mathrm{Constr}(n,J')\vec{N}', U, C_n(I), \vec{N}']. \end{aligned}$$

Thus $C \in Base_1$ and consequently $\mathcal{F}_2 \subseteq \mathcal{F}_1$. On the other hand, if $f \in \mathcal{F}_1 - \mathcal{F}_2$ then

$$f = Min_{\Delta''}(\mathrm{Elim}(\rho(I),\rho(Q),m)\{\rho(\vec{f})\})$$

for a certain Δ''. There exist terms J_2, $\vec{M}_2$ such that

$$f' = [\Gamma \vdash f_j]_{\xi,\rho,\Delta''} \cdot g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta''}[\mathrm{Constr}(j,J_2)\vec{M}_2, U, C_j(I), \vec{M}_2] \in \mathcal{F}_2.$$

But then $f \leq f'$. $\square$

The basic property we want to prove is

$$[\Gamma \vdash \mathrm{Elim}(I, Q, \mathrm{Constr}(n, I)\vec{N})\{\vec{f}\}]_{\xi,\rho,\Delta} = [\Gamma \vdash f_n\vec{e}[C_n(X), \vec{N}, I, Q, \vec{f}]_{\xi,\rho,\Delta}. \qquad (5.4)$$

Then under some reasonable assumptions we have

$$[\Gamma \vdash \mathrm{Elim}(I, Q, \mathrm{Constr}(n, I)\vec{N})\{\vec{f}\}]_{\xi,\rho,\Delta}$$
$$= [\Gamma \vdash f_n]_{\xi,\rho,\Delta} \cdot g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[\mathrm{Constr}(n, I)\vec{N}, U, C_n(I), \vec{N}]$$

and for a certain sequence $\vec{g}$ the following equality holds

$$[\Gamma \vdash f_n\vec{e}[C(X), \vec{N}, I, Q, \vec{f}]_{\xi,\rho,\Delta} = [\Gamma \vdash f_n]_{\xi,\rho,\Delta} \cdot \vec{g}.$$

We would like to prove that the sequence $\vec{g}$ is equal to $g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}[\mathrm{Constr}(n, I)\vec{N}, U, C_n(I), \vec{N}]$. However, this equality does not hold in general. Consider the type $Tree$ introduced on page 24. Recall that the type of the second constructor of $Tree$ is $(Nat \to Tree)$. Suppose we want to eliminate a term $M = cons\ A\ B$ and $U = \langle 1, a, b\rangle$ and $\rho(M) = cons\ N_0\ N_1$. Then

$$g_{\Gamma,Tree,Q,\vec{f},\xi,\rho,\Delta}[M, U, C_1(Tree), (N_0, N_1)] = (\Delta \vdash N_0, a), (\Delta \vdash N_1, b),$$
$$(\Delta \vdash \lambda x : Nat.\mathrm{Elim}(Tree, \rho(Q), N_1x)\{\rho(\vec{f})\},$$
$$\lambda(\Delta' \vdash a, C).G_{\Gamma,Tree,Q,\vec{f},\xi,\rho,\Delta}(pred(\Delta' \vdash \rho(M)))(\Delta' \vdash N_1a, C))$$

and

$$\vec{g} = (\Delta \vdash N_0, a), (\Delta \vdash N_1, b), (\Delta \vdash \lambda x : Nat.\mathrm{Elim}(Tree, \rho(Q), N_1x)\{\rho(\vec{f})\},$$
$$\lambda(\Delta' \vdash a, C).[\Gamma, x : Nat.\mathrm{Elim}(Tree, Q, Bx)\{\vec{f}\}]_{(\xi;x:=C),(\rho;x:=a),\Delta}).$$

We should prove that

$$\lambda(\Delta' \vdash a, C).G_{\Gamma,Tree,Q,\vec{f},\xi,\rho,\Delta}(pred(\Delta' \vdash \rho(M)))(\Delta' \vdash N_1a, C))$$
$$= \lambda(\Delta' \vdash a, C).[\Gamma, x : Nat.\mathrm{Elim}(Tree, Q, Bx)\{\vec{f}\}]_{(\xi;x:=C),(\rho;x:=a),\Delta}.$$

Lemma 86 only proves that the values of the functions are equal for arguments $(\Delta' \vdash a, C)$ such that $(\Delta' \vdash a) \in [\Gamma \vdash Nat]_{\xi,\rho,\Delta}$. In the general case we cannot say anything about the values of the functions. It turns out that this property (functions have equal values for certain arguments) is enough to have the equality 5.4 in the cases we really need it. We define a relation which formalizes the property of partial equality.

Let Γ, Δ be two contexts and $\langle\xi, \rho\rangle$ and $\langle\xi', \rho\rangle$ be two constructor valuations which satisfy Γ at Δ. If $\Gamma \vdash A : T$ and A is a large term then we define the relation $\simeq_{\Gamma,\xi,\xi',\rho,\Delta}$ in $V_\Delta(\rho(A))$.

- If A is a type, a formula or a kind then

$$C \simeq_{\Gamma,\xi,\xi',\rho,\Delta} C' \Leftrightarrow C = C'.$$

- If A is a subset or a constructor with an argument of type T_1 then

$$\begin{aligned} C \simeq_{\Gamma,\xi,\xi',\rho,\Delta} C' \Leftrightarrow\ & \text{for every } \Gamma \vdash t : T_1 \\ & \text{such that } (\Delta' \vdash \rho(t)) \in [\Gamma \vdash T_1]_{\xi,\rho,\Delta} \cap [\Gamma \vdash T_1]_{\xi',\rho,\Delta} \\ & \text{for every } \mathcal{A}, \mathcal{A}' \in V_{\Delta'}(\rho(A)\rho(t)) \\ & \text{if } \mathcal{A} \simeq_{\Gamma,\xi,\xi',\rho,\Delta} \mathcal{A}' \\ & \text{then } C(\Delta' \vdash \rho(t), \mathcal{A}) \simeq_{\Gamma,\xi,\xi',\rho,\Delta} C'(\Delta' \vdash \rho(t), \mathcal{A}'). \end{aligned}$$

- If T is a large inductive type and $A =_{\beta\iota} \mathrm{Constr}(n, I)\vec{N}$ then

$$C \simeq_{\Gamma,\xi,\xi',\rho,\Delta} C' \Leftrightarrow \forall i > 0\ (\pi_i(C) \simeq_{\Gamma,\xi,\xi',\rho,\Delta} \pi_i(C')) \wedge \pi_0(C) = \pi_0(C').$$

Let Γ, Δ be two contexts and $\langle \xi_1, \rho\rangle$, $\langle \xi_1', \rho\rangle$, $\langle \xi, \rho\rangle$ and $\langle \xi', \rho\rangle$ be constructor valuations which satisfy Γ at Δ. We write $\xi_1 \simeq_{\Gamma,\xi,\xi'\rho,\Delta} \xi_1'$ when

$$\forall \alpha\ (\xi_1(\alpha) \simeq_{\Gamma,\xi,\xi',\rho,\Delta} \xi_1'(\alpha)).$$

Lemma 87. *Let A be a large term in the context Γ_1. Suppose Δ is a context, ρ is an object substitution, $C, C' \in V_\Delta(\rho(A))$, and $\Gamma \subseteq \Gamma_1$, and $FV(A) \subseteq dom(\Gamma)$. Suppose that*

- *$\langle \xi, \rho\rangle$, $\langle \xi', \rho\rangle$ are constructor valuations which satisfy Γ at Δ such that $\xi \simeq_{\Gamma,\xi,\xi',\rho,\Delta} \xi'$;*
- *$\langle \xi_1, \rho\rangle$, $\langle \xi_1', \rho\rangle$ are constructor valuations which satisfy Γ_1 at Δ such that*

$$\xi_1 \simeq_{\Gamma_1,\xi_1,\xi_1',\rho,\Delta} \xi_1'.$$

Suppose that for every $x \in dom(\Gamma)$ we have $\xi(x) = \xi_1(x)$ and $\xi'(x) = \xi_1'(x)$. Then

$$C \simeq_{\Gamma_1,\xi_1,\xi_1',\rho,\Delta} C' \text{ if and only if } C \simeq_{\Gamma,\xi,\xi',\rho,\Delta} C'.$$

Proof. Easy induction with respect to the definition of $\simeq_{\Gamma,\xi,\xi',\rho,\Delta}$ using Lemmas 77 and 78. We only consider the case when $\Gamma_1 \vdash A : T$ is a subset or a constructor of large inductive object with an argument of type T_1.
($\Rightarrow$) Suppose $C \simeq_{\Gamma_1,\xi_1,\xi_1',\rho,\Delta} C'$. For every $x \in FV(A)$ we have $\Gamma_1(x) = \Gamma(x)$, thus $\Gamma \vdash A : T$. Suppose $\Gamma \vdash t : T_1$ is such that

$$(\Delta' \vdash \rho(t)) \in [\Gamma \vdash T_1]_{\xi,\rho,\Delta} \cap [\Gamma \vdash T_1]_{\xi',\rho,\Delta}$$

and $\mathcal{A}, \mathcal{A}' \in V_{\Delta'}(\rho(A)\rho(t))$ such that $\mathcal{A} \simeq_{\Gamma,\xi,\xi',\rho,\Delta} \mathcal{A}'$. By Lemma 77

$$[\Gamma \vdash T_1]_{\xi,\rho,\Delta} = [\Gamma_1 \vdash T_1]_{\xi_1,\rho,\Delta} \quad \text{and} \quad [\Gamma \vdash T_1]_{\xi',\rho,\Delta} = [\Gamma_1 \vdash T_1]_{\xi_1',\rho,\Delta}.$$

Thus

$$(\Delta' \vdash \rho(t)) \in [\Gamma_1 \vdash T_1]_{\xi_1,\rho,\Delta} \cap [\Gamma_1 \vdash T_1]_{\xi_1',\rho,\Delta}.$$

By the induction hypothesis we have $\mathcal{A} \simeq_{\Gamma_1,\xi_1,\xi_1',\rho,\Delta} \mathcal{A}'$ and thus by the assumption

$$C(\Delta' \vdash \rho(t), \mathcal{A}) \simeq_{\Gamma_1,\xi_1,\xi_1',\rho,\Delta} C'(\Delta' \vdash \rho(t), \mathcal{A}').$$

By the induction hypothesis

$$C(\Delta' \vdash \rho(t), \mathcal{A}) \simeq_{\Gamma,\xi,\xi',\rho,\Delta} C'(\Delta' \vdash \rho(t), \mathcal{A}').$$

($\Leftarrow$) Suppose $C \simeq_{\Gamma,\xi,\xi',\rho,\Delta} C'$. We will prove that

$$C \simeq_{\Gamma_1,\xi_1,\xi_1',\rho,\Delta} C'.$$

Suppose $\Gamma \vdash t : T_1$ is such that

$$(\Delta' \vdash \rho(t)) \in [\Gamma_1 \vdash T_1]_{\xi_1,\rho,\Delta} \cap [\Gamma_1 \vdash T_1]_{\xi_1',\rho,\Delta}$$

and $\mathcal{A}, \mathcal{A}' \in V_{\Delta'}(\rho(A)\rho(t))$ such that $\mathcal{A} \simeq_{\Gamma_1,\xi_1,\xi_1',\rho,\Delta} \mathcal{A}'$. By Lemmas 77 and 78 we have

$$[\Gamma \vdash T_1]_{\xi,\rho,\Delta} = [\Gamma_1 \vdash T_1]_{\xi_1,\rho,\Delta} \quad \text{and} \quad [\Gamma \vdash T_1]_{\xi',\rho,\Delta} = [\Gamma_1 \vdash T_1]_{\xi_1',\rho,\Delta}.$$

Thus

$$(\Delta' \vdash \rho(t)) \in [\Gamma \vdash T_1]_{\xi,\rho,\Delta} \cap [\Gamma \vdash T_1]_{\xi',\rho,\Delta}.$$

By the induction hypothesis we have $\mathcal{A} \simeq_{\Gamma,\xi,\xi',\rho,\Delta} \mathcal{A}'$ and thus by the assumption

$$C(\Delta' \vdash \rho(t), \mathcal{A}) \simeq_{\Gamma,\xi,\xi',\rho,\Delta} C'(\Delta' \vdash \rho(t), \mathcal{A}').$$

By the induction hypothesis

$$C(\Delta' \vdash \rho(t), \mathcal{A}) \simeq_{\Gamma_1,\xi_1,\xi_1',\rho,\Delta} C'(\Delta' \vdash \rho(t), \mathcal{A}'). \qquad \square$$

Lemma 88. *Suppose I is an inductive type such that $\Gamma \vdash I : *^t$. Suppose that*

- *for every constructor valuation $\langle \hat{\xi}, \hat{\rho} \rangle$ which satisfies Γ at Δ, if $\Gamma' \vdash t : C$ is structurally smaller than $\Gamma \vdash I : *^t$ then $(\Delta \vdash \hat{\rho}(t)) \in [\Gamma' \vdash C]_{\xi,\rho,\Delta}$;*
- *for every pair $\langle \hat{\xi}, \hat{\rho} \rangle$, $\langle \hat{\xi}', \hat{\rho} \rangle$ of constructor valuations which satisfy Γ at Δ such that $\hat{\xi} \simeq_{\Gamma,\hat{\xi},\hat{\xi}',\hat{\rho},\Delta} \hat{\xi}'$, and for every sequent $\Gamma' \vdash t : C$ structurally smaller than $\Gamma \vdash I : *^t$, we have*

$$[\Gamma' \vdash t]_{\hat{\xi},\hat{\rho},\Delta} \simeq_{\Gamma,\hat{\xi},\hat{\xi}',\rho,\Delta} [\Gamma' \vdash t]_{\hat{\xi}',\hat{\rho},\Delta}.$$

If Γ and Δ are contexts, $\langle \xi, \rho \rangle$, $\langle \xi', \rho \rangle$ are two constructor valuations which satisfy Γ at Δ such that $\xi \simeq_{\Gamma,\xi,\xi',\rho,\Delta} \xi'$ then for every saturated set $S \in V_\Delta(\rho(I))$ we have

$$F_{\Gamma,I,\xi,\rho,\Delta}(S) = F_{\Gamma,I,\xi',\rho,\Delta}(S).$$

Proof. By the definition

$$\begin{aligned} F_{\Gamma,I,\xi,\rho,\Delta}(S) = \Big(\bigcap SAT^{\Delta}_{\rho(I)}\Big) \cup \{ & (\Delta' \vdash u) \in SN^{\Delta}_{\rho(I)} \mid \\ & \text{if } \Delta'' \supseteq \Delta' \text{ and } \Delta'' \vdash u \to^*_k \mathrm{Constr}(n, X)\vec{N}, \text{ and } C_n(X) = \Pi\vec{x} : \vec{T}.X \\ & \text{then for every } j \text{ we have } (\Delta'' \vdash N_j) \in \mathit{Interp}(\Gamma^j \vdash T_j)_{\xi^j,\rho^j,\Delta'',X,S|_{\Delta''}} \} \end{aligned}$$

where ξ^j, ρ^j are as in the definition of $F_{\Gamma,I,\xi,\rho,\Delta}(S)$ (see page 58). The value $F_{\Gamma,I,\xi',\rho,\Delta}(S)$ is defined similarly. By Lemma 54 we have

$$\mathit{Interp}(\Gamma^j \vdash T_j)_{\xi^j,\rho^j,\Delta'',X,S|_{\Delta''}} = [\Gamma^j \vdash T_j]_{(\xi^j;X:=S|_{\Delta''}),(\rho^j;X:=\rho(I)),\Delta''}$$

and

$$\mathit{Interp}(\Gamma^j \vdash T_j)_{\xi'^j,\rho^j,\Delta'',X,S|_{\Delta''}} = [\Gamma^j \vdash T_j]_{(\xi'^j;X:=S|_{\Delta''}),(\rho^j;X:=\rho(I)),\Delta''}.$$

By the assumption we have

$$[\Gamma^j \vdash T_j]_{(\xi^j;X:=S|_{\Delta''}),(\rho^j;X:=\rho(I)),\Delta''} = [\Gamma^j \vdash T_j]_{(\xi'^j;X:=S|_{\Delta''}),(\rho^j;X:=\rho(I)),\Delta''}.$$

The conclusion follows from the facts above. $\square$

Lemma 89. *Suppose I is an inductive predicate such that $\Gamma \vdash I : A$. Suppose that*

- *for every constructor valuation $\langle\hat{\xi},\hat{\rho}\rangle$ which satisfies Γ at Δ, if $\Gamma'\vdash t:C$ is structurally smaller than $\Gamma\vdash I:A$ then $(\Delta\vdash\hat{\rho}(t))\in[\Gamma'\vdash C]_{\hat{\xi},\hat{\rho},\Delta}$;*

- *for every pair $\langle\hat{\xi},\hat{\rho}\rangle$, $\langle\hat{\xi}',\hat{\rho}'\rangle$ of constructor valuations which satisfy Γ at Δ such that $\hat{\xi}\simeq_{\Gamma,\hat{\xi},\hat{\xi}',\hat{\rho},\Delta}\hat{\xi}'$ for every $\Gamma'\vdash t:C$ structurally smaller than $\Gamma\vdash I:A$ we have*

$$[\Gamma'\vdash t]_{\hat{\xi},\hat{\rho},\Delta}\simeq_{\Gamma,\hat{\xi},\hat{\xi}',\hat{\rho},\Delta}[\Gamma'\vdash t]_{\hat{\xi}',\hat{\rho},\Delta}.$$

If Γ and Δ are contexts, $\langle\xi,\rho\rangle$, $\langle\xi',\rho\rangle$ are two constructor valuations which satisfy Γ at Δ such that $\xi\simeq_{\Gamma,\xi,\xi',\rho,\Delta}\xi'$ then for every $S\in V_\Delta(\rho(I))$ we have

$$H_{\Gamma,I,\xi,\rho,\Delta}(S)\simeq_{\Gamma,\xi,\xi',\rho,\Delta}H_{\Gamma,I,\xi',\rho,\Delta}(S).$$

Proof. Let $(\vec{\Sigma},\vec{u},\vec{U})$ be an appropriate sequence of arguments for $(\vec{x}:\vec{\tau})$. By the definition of the operator H (see page 81) we have

$$H_{\Gamma,I,\xi,\rho,\Delta}(S)((\Sigma_i,u_i,U_i)_{i=1}^n)=(\bigcap SAT^{\Sigma_n}_{\rho(I)\vec{u}})\cup h_{\Gamma,I,\xi,\rho,\Delta}(S)((\Sigma_i,u_i,U_i)_{i=1}^n)$$

where $h_{\Gamma,I,\xi,\rho,\Delta}(S)((\Sigma_i,u_i,U_i)_{i=1}^n)$ consists of simple sequents $(\Delta'\vdash m)\in SN^{\Sigma_n}_{\rho(I)\vec{u}}$ such that

for every context Δ'' and every term J such that $(\Delta''\vdash J)\in SN^{\Sigma_n}_{\rho(A)}$ and $J=_{\beta\iota}\rho(I)$,
for every context Δ''' and every term Q such that $(\Delta'''\vdash Q)\in[\Gamma\vdash A]_{\xi|_{\Delta''},\rho,\Delta''}$,
for every $P\in V_{\Delta'''}(Q)$, ,
for every context Δ'''' and for every vector $\vec{f}$ such that

$$(\Delta''''\vdash f_i)\in[\Gamma,q:A,X:A\vdash\triangle\{C_i(X),q\}]_{(\xi|_{\Delta''''};X:=S|_{\Delta''''};q:=P),(\rho;X:=J;q:=Q),\Delta''''}$$

we have $(\Delta''''\vdash\mathrm{Elim}(J,Q,\vec{u},m)\{\vec{f}\})\in P((\Sigma_i,u_i,U_i)_{i=1}^n)$.

The value $H_{\Gamma,I,\xi',\rho,\Delta}(S)$ is defined similarly. It is easy to show that under the assumption

$$[\Gamma,q:A,X:A\vdash\triangle\{C_i(X),q\}]_{(\xi|_{\Delta'''};X:=S|_{\Delta'''};q:=C),(\rho;X:=J;q:=Q),\Delta'''}\simeq_{\Gamma,\xi,\xi',\rho,\Delta}$$
$$[\Gamma,q:A,X:A\vdash\triangle\{C_i(X),q\}]_{(\xi'|_{\Delta'''};X:=S|_{\Delta'''};q:=C),(\rho;X:=J;q:=Q),\Delta'''}.$$

Thus we get the conclusion. □

Lemma 90. *Let $\mathcal{F}$ and $\mathcal{F}'$ be two sets which satisfy the condition: for every $f\in\mathcal{F}$ there exists $f'\in\mathcal{F}'$ such that $f\simeq_{\Gamma,\xi,\xi',\rho,\Delta}f'$ and for every $f'\in\mathcal{F}'$ there exists $f\in\mathcal{F}$ such that $f'\simeq_{\Gamma,\xi,\xi',\rho,\Delta}f$. Then*

$$\bigsqcup\mathcal{F}\simeq_{\Gamma,\xi,\xi',\rho,\Delta}\bigsqcup\mathcal{F}'.$$

Proof. We proceed by induction with respect to the definition of $\simeq_{\Gamma,\xi,\xi',\rho,\Delta}$. If $\mathcal{F}$ is a set of saturated sets the conclusion is obvious. Suppose that $\mathcal{F}$ is a set of functions. Let $(\Delta'\vdash m)\in[\Gamma\vdash T]_{\xi,\rho,\Delta}\cap[\Gamma\vdash T]_{\xi',\rho,\Delta}$ and $C,C'\in V_{\Delta'}(m)$ and $C\simeq_{\Gamma,\xi,\xi'\rho,\Delta}C'$. Then

$$(\bigsqcup\mathcal{F})(\Delta'\vdash m,C)=\bigsqcup\mathcal{F}_1\quad\text{and}\quad(\bigsqcup\mathcal{F}')(\Delta'\vdash m,C')=\bigsqcup\mathcal{F}_2$$

where

$$\mathcal{F}_1 = \{f(\Delta'' \vdash m, C|_{\Delta''}) \mid \Delta'' \supseteq \Delta', (\Delta'' \vdash m, C|_{\Delta''}) \in dom(f), f \in \mathcal{F}\},$$
$$\mathcal{F}_2 = \{f(\Delta'' \vdash m, C'|_{\Delta''}) \mid \Delta'' \supseteq \Delta', (\Delta'' \vdash m, C'|_{\Delta''}) \in dom(f), f \in \mathcal{F}'\}.$$

We show that the sets $\mathcal{F}_1$, $\mathcal{F}_2$ satisfy the assumption. Let $g \in \mathcal{F}_1$. Then $g = f(\Delta'' \vdash m, C|_{\Delta''})$ for a certain $f \in \mathcal{F}$, $\Delta'' \supseteq \Delta'$, and $(\Delta'' \vdash m, C|_{\Delta''}) \in dom(f)$. By the assumption there exists $f' \in \mathcal{F}_2$ such that $f \simeq_{\Gamma,\xi,\xi',\rho,\Delta} f'$. We know that $(\Delta'' \vdash m) \in [\Gamma \vdash T]_{\xi',\rho,\Delta}$ and $C'|_{\Delta''} \in V_{\Delta''}(m)$. Thus $(\Delta'' \vdash m, C'|_{\Delta''}) \in dom(f')$. Let $g' = f'(\Delta'' \vdash m, C'|_{\Delta''})$. Then $g \simeq_{\Gamma,\xi,\xi',\rho,\Delta} g'$ and $g' \in \mathcal{F}_2$. Similarly one can prove that for every $g' \in \mathcal{F}_2$ there exists $g \in \mathcal{F}_1$ such that $g \simeq_{\Gamma,\xi,\xi',\rho,\Delta} g'$. The sets $\mathcal{F}_1$, $\mathcal{F}_2$ indeed satisfy the assumption thus by the induction hypothesis we have $\mathcal{F}_1 \simeq_{\Gamma,\xi,\xi',\rho,\Delta} \mathcal{F}_2$. □

Lemma 91. *Suppose that*

- *for every constructor valuation $\langle\hat{\xi}, \hat{\rho}\rangle$ which satisfies Γ at Δ if $\Gamma' \vdash t : C$ is structurally smaller than $\Gamma \vdash \mathrm{Elim}(I, Q, m)\{\vec{f}\} : A$ then $(\Delta \vdash \hat{\rho}(t)) \in [\Gamma' \vdash C]_{\hat{\xi},\hat{\rho},\Delta}$;*
- *for every pair $\langle\hat{\xi}, \hat{\rho}\rangle$, $\langle\hat{\xi}', \hat{\rho}'\rangle$ of constructor valuations which satisfy Γ at Δ such that $\hat{\xi} \simeq_{\Gamma,\hat{\xi},\hat{\xi}',\hat{\rho},\Delta} \hat{\xi}'$ for every $\Gamma' \vdash t : C$ structurally smaller than $\Gamma \vdash T : A$ we have*

$$[\Gamma \vdash t]_{\hat{\xi},\hat{\rho},\Delta} \simeq_{\Gamma,\hat{\xi},\hat{\xi}',\hat{\rho},\Delta} [\Gamma \vdash t]_{\hat{\xi}',\hat{\rho},\Delta};$$

- *Γ and Δ are contexts, $\langle\xi, \rho\rangle$, $\langle\xi', \rho\rangle$ are two constructor valuations which satisfy Γ at Δ and $\xi \simeq_{\Gamma,\xi,\xi',\rho,\Delta} \xi'$;*
- *S is a saturated set and $S \in \mathcal{D}_{\Gamma,I,\xi',\rho,\Delta}$.*

If $U \simeq_{\Gamma,\xi,\xi',\rho,\Delta} U'$ then

$$G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta' \vdash m, U) \simeq_{\Gamma,\xi,\xi',\rho,\Delta} G_{\Gamma,I,Q,\vec{f},\xi',\rho,\Delta}(S)(\Delta' \vdash m, U').$$

Proof. We proceed by induction with respect to the ordering in $\mathcal{D}_{\Gamma,I,\xi',\rho,\Delta}$. If $S = \bigcap SAT^{\Delta}_{\rho(I)}$ then

$$G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta' \vdash m, U) = Min_{\Delta'}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\} = G_{\Gamma,I,Q,\vec{f},\xi',\rho,\Delta}(S)(\Delta' \vdash m, U').$$

Suppose the conclusion is true for every $S' < S$. There are two cases.
Case 1: If $m \neq_{\beta\iota} \mathrm{Constr}(n, J)\vec{N}$ then

$$G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta' \vdash m, U) = Min_{\Delta'}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\}) \simeq_{\Gamma,\xi,\xi',\rho,\Delta} Min_{\Delta'}(\mathrm{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\}) = G_{\Gamma,I,Q,\vec{f},\xi',\rho,\Delta}(S)(\Delta' \vdash m, U').$$

Case 2: If $m =_{\beta\iota} \mathrm{Constr}(n, J)\vec{N}$ then

$$G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta' \vdash m, U) = \bigsqcup(Base \cup Min)$$

and

$$G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta' \vdash m, U') = \bigsqcup(Base' \cup Min').$$

Recall that $Base$ consists of all values of the form

$$[\Gamma \vdash f_n]_{\xi|_{\Delta'},\rho,\Delta''} \cdot g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta''}[\text{Constr}(n,J)\vec{M}, U, C_n(I), \vec{M}]$$

such that $\Delta'' \supseteq \Delta'$, $m =_{\beta\iota} \text{Constr}(n,J)\vec{M}$ and $(\Delta'' \vdash \text{Constr}(n,J)\vec{M}) \in S$ and $Base'$ consists of all values of the form

$$[\Gamma \vdash f_n]_{\xi|_{\Delta'},\rho,\Delta''} \cdot g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta''}[\text{Constr}(n,J)\vec{M}, U', C_n(I), \vec{M}]$$

such that $\Delta'' \supseteq \Delta'$, $m =_{\beta\iota} \text{Constr}(n,J)\vec{M}$ and $(\Delta'' \vdash \text{Constr}(n,J)\vec{M}) \in S$. The sequences

$$\vec{g}_1 = g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta''}[\text{Constr}(j,J)\vec{M}, U, C_j(I), \vec{M}],$$
$$\vec{g}_2 = g_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta''}[\text{Constr}(j,J)\vec{M}, U', C_j(I), \vec{M}]$$

are two sequences of pairs of the form $(\Delta \vdash a, C)$. For every index i we have

$$(\vec{g}_1)_i = (\Delta \vdash a, C) \quad \text{and} \quad (\vec{g}_2)_i = (\Delta \vdash a, C').$$

The elements C, C' in the sequence are either appropriate elements of sequences U, U' or applications of the operator G to equivalent arguments. Using the induction hypothesis it is easy to observe that for every $f \in Base$ there exists $f' \in Base'$ such that $f \simeq_{\Gamma,\xi,\xi',\rho,\Delta} f'$ and vice versa.

The set Min consists of all values of the form

$$Min_{\Delta''}(\text{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\})$$

such that $\Delta'' \supseteq \Delta'$, $m =_{\beta\iota} \text{Constr}(n,J)\vec{M}$ and

$$(\Delta'' \vdash \text{Constr}(n,J)\vec{M}) \in T^{\Delta}_{\rho(I)} - S.$$

At the same time Min' consists of all values of the form

$$Min_{\Delta''}(\text{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\})$$

such that $\Delta'' \supseteq \Delta'$, $m =_{\beta\iota} \text{Constr}(n,J)\vec{M}$ and

$$(\Delta'' \vdash \text{Constr}(n,J)\vec{M}) \in T^{\Delta}_{\rho(I)} - S.$$

Then $Min = Min'$. The conclusion follows from Lemma 90. □

Lemma 92. *Suppose* Γ *and* Δ *are contexts,* $\langle \xi, \rho \rangle$ *and* $\langle \xi', \rho \rangle$ *are object valuations which satisfy* Γ *at* Δ*,* $S \in \mathcal{D}_{\Gamma,I,\xi',\rho,\Delta}$ *and*

$$(\Delta \vdash \rho(N)) \in [\Gamma \vdash C(X)]_{(\xi';X:=S),(\rho;X:=\rho(I)),\Delta}$$

where $C(X) = \Pi\vec{x} : \vec{T}.X$ *is a type of constructor in* X*. Let* R *be a function such that if* $(\Delta_i, x_i, C_i)_{i=1}^n$ *is an appropriate sequence of arguments for* $(\vec{x} : \rho(\vec{T}))$ *then*

$$R((\Delta_i, x_i, C_i)_{i=1}^n) = G_{\Gamma,I,Q,\vec{f},\xi',\rho,\Delta}(S)(\Delta \vdash \rho(N)\vec{x}, [\Gamma \vdash N]_{\xi',\rho,\Delta}(\Delta_i, x_i, C_i)_{i=1}^n).$$

Then

$$[\Gamma \vdash \lambda\vec{x} : \vec{T}.\text{Elim}(I, Q, N\vec{x})\{\vec{f}\}]_{\xi,\rho,\Delta} \simeq_{\Gamma,\xi,\xi',\rho,\Delta} R.$$

Proof. Lemma 66 implies that $R \in V_\Delta(\rho(\lambda\vec{x}:\vec{T}.\mathrm{Elim}(I,Q,N\vec{x})\{\vec{f}\}))$. By the definition

$$[\Gamma \vdash \lambda\vec{x}:\vec{T}.\mathrm{Elim}(I,Q,N\vec{x})\{\vec{f}\}]_{\xi,\rho,\Delta} \\ = \lambda\!\!\lambda(\Delta_i,x_i,C_i)_i.G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}([\Gamma \vdash I]_{\xi,\rho,\Delta})(\Delta \vdash \rho(N)\vec{x},[\Gamma \vdash N]_{\xi,\rho,\Delta}(\Delta_i,x_i,C_i)_i).$$

We proceed by induction with respect to the ordering in $\mathcal{D}_{\Gamma,I,\xi',\rho,\Delta}$. Let $(\Delta_i,a_i,C_i)_i$ be a sequence of arguments such that

$$(\Delta_i \vdash a_i) \in [\Gamma,(x_j:T_j)_{j=0}^{i-1} \vdash T_i]_{\xi_{i-1},\rho_{i-1},\Delta_{i-1}} \cap [\Gamma,(x_j:T_j)_{j=0}^{i-1} \vdash T_i]_{\xi'_{i-1},\rho_{i-1},\Delta_{i-1}}$$

and

$$C_i \in V_{\Delta_i}(\rho(N)(a_j)_{j=0}^{i-1}).$$

By the assumption $(\Delta \vdash \rho(N)\vec{a}) \in [\Gamma,\vec{x}:\vec{T} \vdash X]_{(\xi';\vec{x}:=\vec{C};X:=S),(\rho;\vec{x}:=\vec{a},X:=\rho(I)),\Delta}$.

If $S = \bigcap SAT^\Delta_{\rho(I)}$ then we have

$$(\Delta_n \vdash \rho(N)\vec{a}) \in S = \bigcap SAT^\Delta_{\rho(I)}.$$

By Lemma 35 there is no m, I, $\vec{N}'$ such that $\mathrm{Constr}(m,I)\vec{N}' =_{\beta\iota} \rho(N)\vec{a}$. Thus

$$[\Gamma \vdash \lambda\vec{x}:\vec{T}.\mathrm{Elim}(I,Q,N\vec{x})\{\vec{f}\}]_{\xi,\rho,\Delta}((\Delta_i,a_i,C_i)_{i=1}^n) \\ = Min_{\Delta_n}(\mathrm{Elim}(\rho(I),\rho(Q),\rho(N)\vec{a})\{\rho(\vec{f})\}))$$

and

$$R((\Delta_i,a_i,C_i)_{i=1}^n) = Min_{\Delta_n}(\mathrm{Elim}(\rho(I),\rho(Q),\rho(N)\vec{a})\{\rho(\vec{f})\})).$$

Thus the conclusion holds.

Assume that the induction hypothesis holds for every $S' < S$. There are two cases.

Case 1: $\pi_1([\Gamma \vdash N]_{\xi,\rho,\Delta}\vec{a}) = j$ and $\rho(N)\vec{a} =_{\beta\iota} \mathrm{Constr}(j,X)\vec{N}'$. Lemma 88 implies that $\mathcal{D}_{\Gamma,I,\xi',\rho,\Delta} = \mathcal{D}_{\Gamma,I,\xi,\rho,\Delta}$. Thus $S \subseteq [\Gamma \vdash I]_{\xi,\rho,\Delta}$. By Lemma 86 we have

$$G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}([\Gamma \vdash I]_{\xi,\rho,\Delta})(\Delta \vdash \rho(N)\vec{x},[\Gamma \vdash N]_{\xi,\rho,\Delta}(\Delta_i,x_i,C_i)_{i=1}^n) \\ = G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta \vdash \rho(N)\vec{x},[\Gamma \vdash N]_{\xi,\rho,\Delta}(\Delta_i,x_i,C_i)_{i=1}^n).$$

By Lemma 91 and the induction hypothesis we get the conclusion.

Case 2: Otherwise

$$G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}([\Gamma \vdash I]_{\xi,\rho,\Delta})(\Delta \vdash \rho(N)\vec{a},[\Gamma \vdash N]_{\xi,\rho,\Delta}(\Delta_i,a_i,C_i)_{i=1}^n) \\ = Min_\Delta(\mathrm{Elim}(\rho(I),\rho(Q),\rho(N)\vec{a})\{\rho(\vec{f})\}.$$

for every j, X, $\vec{N}'$ such that $\mathrm{Constr}(j,X)\vec{N}' =_{\beta\iota} \rho(N)\vec{a}$ we have

$$(\Delta'' \vdash \mathrm{Constr}(j,X)\vec{N}') \in T^\Delta_{\rho(I)} - [\Gamma \vdash I]_{\xi,\rho,\Delta}$$

then $(\Delta'' \vdash \mathrm{Constr}(j,X)\vec{N}') \in T^\Delta_{\rho(I)} - S$ as $S \subseteq [\Gamma \vdash I]_{\xi,\rho,\Delta}$. Thus

$$G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}(S)(\Delta \vdash \rho(N)\vec{a},[\Gamma \vdash N]_{\xi,\rho,\Delta}(\Delta_i,a_i,C_i)_{i=1}^n \\ = Min_\Delta(\mathrm{Elim}(\rho(I),\rho(Q),\rho(N)\vec{a})\{\rho(\vec{f})\}.$$

Either way

$$[\Gamma \vdash \lambda\vec{x} : \vec{T}.\mathrm{Elim}(I, Q, N\vec{x})\{\vec{f}\}]_{\xi,\rho,\Delta} \simeq_{\Gamma,\xi,\xi',\rho,\Delta} R. \qquad \square$$

Lemma 93. *Suppose that*

1. $\Gamma \vdash T : A$ *and* $\Gamma \vdash T' : A$ *and*

$$T = \mathrm{Elim}(I, Q, \mathrm{Constr}(n, I)\vec{N})\{\vec{f}\} \quad \text{and} \quad T' = f_n\vec{e}[C_n(I), \vec{N}, I, Q, \vec{f}].$$

2. *For every constructor valuation* $\langle\hat{\xi}, \hat{\rho}\rangle$ *which satisfies* Γ *at* Δ*, for every* $\Gamma' \vdash t : C$ *structurally smaller than at least one of sequents* $\Gamma \vdash T : A$*,* $\Gamma \vdash T' : A$ *we have*

$$(\Delta \vdash \hat{\rho}(t)) \in [\Gamma' \vdash C]_{\hat{\xi},\hat{\rho},\Delta}.$$

3. *For every pair* $\langle\hat{\xi}, \hat{\rho}\rangle$*,* $\langle\hat{\xi}', \hat{\rho}'\rangle$ *of constructor valuations which satisfy* Γ *at* Δ *such that* $\hat{\xi} \simeq_{\Gamma,\hat{\xi},\hat{\xi}',\hat{\rho},\Delta} \hat{\xi}'$*, for every* $\Gamma' \vdash t : C$ *structurally smaller than* $\Gamma \vdash T : A$ *or* $\Gamma \vdash T' : A$ *we have*

$$[\Gamma' \vdash t]_{\hat{\xi},\hat{\rho},\Delta} \simeq_{\Gamma,\hat{\xi},\hat{\xi}',\hat{\rho},\Delta} [\Gamma' \vdash t]_{\hat{\xi}',\hat{\rho},\Delta}.$$

If $\langle\xi, \rho\rangle$*,* $\langle\xi', \rho'\rangle$ *is a pair of constructor valuations which satisfy* Γ *at* Δ *such that* $\xi \simeq_{\Gamma,\xi,\xi',\rho,\Delta} \xi'$ *then*

$$[\Gamma \vdash T]_{\xi,\rho,\Delta} \simeq_{\Gamma,\xi,\xi',\rho,\Delta} [\Gamma \vdash T']_{\xi',\rho,\Delta}.$$

Proof. We have

$$T = \mathrm{Elim}(I, Q, \mathrm{Constr}(n, I)\vec{N})\{\vec{f}\}$$

and

$$T' = f_n\vec{e}[C_n(I), \vec{N}, I, Q, \vec{f}].$$

Then

$$[\Gamma \vdash T]_{\xi,\rho,\Delta} = G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}([\Gamma \vdash I]_{\xi,\rho,\Delta})(\Delta \vdash \rho(\mathrm{Constr}(n, I)\vec{N}), [\Gamma \vdash \mathrm{Constr}(n, I)\vec{N}]_{\xi,\rho,\Delta'})$$

and

$$[\Gamma \vdash T']_{\xi',\rho,\Delta} = [\Gamma \vdash f_n\vec{e}[C_n(I), \vec{N}, I, Q, \vec{f}]]_{\xi',\rho,\Delta}.$$

By the assumption $(\Delta \vdash \rho(\mathrm{Constr}(n, I)\vec{N})) \in [\Gamma \vdash I]_{\xi,\rho,\Delta}$ and thus by Lemma 69

$$\begin{aligned} &G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}([\Gamma \vdash I]_{\xi,\rho,\Delta})(\Delta \vdash \rho(\mathrm{Constr}(n, I)\vec{N}), [\Gamma \vdash \mathrm{Constr}(n, I)\vec{N}]_{\xi,\rho,\Delta'}) \\ &\qquad = [\Gamma \vdash f_n]_{\xi,\rho,\Delta} \cdot g_{\Gamma,I,Q,\xi,\rho,\Delta}[\rho(\mathrm{Constr}(n, I)\vec{N}), [\Gamma \vdash \mathrm{Constr}(n, I)\vec{N}]_{\xi,\rho,\Delta}, C_n(I), \vec{N}]. \end{aligned}$$

By the assumption

$$[\Gamma \vdash f_n]_{\xi,\rho,\Delta} \simeq_{\Gamma,\xi,\xi',\rho,\Delta} [\Gamma \vdash f_n]_{\xi',\rho,\Delta}.$$

We thus have to prove that the above interpretations are applied to equivalent arguments. This follows from Lemma 92. $\square$

Lemma 94. *Suppose that*

1. $\Gamma \vdash T : A$ *and* $\Gamma \vdash T' : A$ *and* $T \to_{\beta\iota} T'$*.*

2. *For every constructor valuation $\langle \xi, \rho \rangle$ which satisfies Γ at Δ, for every $\Gamma' \vdash t : C$ structurally smaller than at least one of sequents $\Gamma \vdash T : A$ or $\Gamma \vdash T' : A$, we have*

$$(\Delta \vdash \rho(t)) \in [\Gamma \vdash C]_{\xi,\rho,\Delta}.$$

3. *For every $\Gamma' \vdash t : C$ structurally smaller at least one of sequents $\Gamma \vdash T : A$ or $\Gamma \vdash T' : A$ for every pair $\langle \hat{\xi}, \hat{\rho} \rangle$, $\langle \hat{\xi}', \hat{\rho}' \rangle$ of constructor valuations which satisfy Γ' at Δ such that $\hat{\xi} \simeq_{\Gamma',\hat{\xi},\hat{\xi}',\hat{\rho},\Delta} \hat{\xi}'$, we have*

$$[\Gamma' \vdash t]_{\hat{\xi},\hat{\rho},\Delta} \simeq_{\Gamma',\hat{\xi},\hat{\xi}',\hat{\rho},\Delta} [\Gamma' \vdash t]_{\hat{\xi}',\hat{\rho},\Delta}.$$

Then

$$[\Gamma \vdash T]_{\xi,\rho,\Delta} \simeq_{\Gamma,\xi,\xi',\rho,\Delta} [\Gamma \vdash T']_{\xi',\rho,\Delta}.$$

Proof. Induction with respect to the definition of $T \to_{\beta\iota} T'$.

Case 1: T is a beta-redex. Then $T = (\lambda x : A.B)C$ and $T' = B[x := C]$. We only consider the case when A is a large type. The other case is similar. We have

$$\begin{aligned}&[\Gamma \vdash (\lambda x : A.B)C]_{\xi,\rho,\Delta}\\ &\quad = (\lambda\!\!\lambda(\Delta' \vdash a, U) : \overline{T}^{\Delta}_{\rho(A)}.[\Gamma, x : A \vdash B]_{(\xi;x:=C),(\rho;x:=a),\Delta'})(\Delta \vdash \rho(C), [\Gamma \vdash C]_{\xi,\rho,\Delta})\\ &\quad = [\Gamma, x : A \vdash B]_{(\xi;x:=[\Gamma\vdash C]_{\xi,\rho,\Delta}),(\rho;x:=\rho(C)),\Delta}\end{aligned}$$

and

$$[\Gamma \vdash B[x := C]]_{\xi',\rho,\Delta} = [\Gamma, x : A \vdash B]_{(\xi';x:=[\Gamma\vdash C]_{\xi',\rho,\Delta}),(\rho;x:=\rho(C)),\Delta}.$$

By the assumption

$$[\Gamma \vdash C]_{\xi,\rho,\Delta} \simeq_{\Gamma,\xi,\xi',\rho,\Delta} [\Gamma \vdash C]_{\xi',\rho,\Delta}$$

and for every pair $\langle \hat{\xi}, \hat{\rho} \rangle$, $\langle \hat{\xi}', \hat{\rho}' \rangle$ of constructor valuations which satisfy $(\Gamma, x : A)$ at Δ such that $\hat{\xi} \simeq_{\Gamma,\hat{\xi},\hat{\xi}',\hat{\rho},\Delta} \hat{\xi}'$ we have

$$[\Gamma, x : A \vdash B]_{\hat{\xi},\hat{\rho},\Delta} \simeq_{(\Gamma,x:A),\hat{\xi},\hat{\xi}',\hat{\rho},\Delta} [\Gamma, x : A \vdash B]_{\hat{\xi}',\hat{\rho},\Delta}.$$

Thus

$$\begin{aligned}&[\Gamma, x : A \vdash B]_{(\xi;x:=[\Gamma\vdash C]_{\xi,\rho,\Delta}),(\rho;x:=\rho(C)),\Delta}\\ &\qquad\qquad \simeq_{(\Gamma,x:A),\xi,\xi',\rho,\Delta} [\Gamma, x : A \vdash B]_{(\xi';x:=[\Gamma\vdash C]_{\xi',\rho,\Delta}),(\rho;x:=\rho(C)),\Delta}.\end{aligned}$$

But

$$[\Gamma, x : A \vdash B]_{(\xi;x:=[\Gamma\vdash C]_{\xi,\rho,\Delta}),(\rho;x:=\rho(C)),\Delta} = [\Gamma \vdash B[x := C]]_{\xi,\rho,\Delta}.$$

Thus by Lemma 87

$$[\Gamma \vdash B[x := C]]_{\xi,\rho,\Delta} \simeq_{\Gamma,\xi,\xi',\rho,\Delta} [\Gamma \vdash B[x := C]]_{\xi',\rho,\Delta}.$$

Case 2: T is iota-redex and T' is iota-reduct. This is a consequence of Lemma 93.

Case 3: In the other cases the proof is a routine application of the induction hypothesis. □

5.5. Adequacy lemma and strong normalization proof

In this section we combine the previous results to prove the adequacy lemma, that is if $\Gamma \vdash M : T$ then for an appropriate constructor valuation $\langle \xi, \rho \rangle$ which satisfies Γ at a context Δ we have

$$(\Delta \vdash \rho(M)) \in [\Gamma \vdash T]_{\xi,\rho,\Delta}.$$

Lemma 95. *If $(\Delta' \vdash M) \in [\Gamma \vdash I]_{\xi,\rho,\Delta}$ then either $M \to_k^* M'$ and $(\Delta' \vdash M') \in B^{\Delta}_{\rho(I)}$ for a certain Δ', or $M \to_k^* \mathrm{Constr}(n, X)\vec{N}$.*

Proof. If $(\Delta' \vdash M) \in [\Gamma \vdash I]_{\xi,\rho,\Delta}$ then by the definition of $[\Gamma \vdash I]_{\xi,\rho,\Delta}$ we have

$$(\Delta' \vdash M) \in SN^{\Delta}_{\rho(I)}.$$

Thus there exists M' in key normal form such that $M \to_k^* M'$. By induction with respect to the structure of M' it is easy to observe that if M' is in key normal form then one of the following cases hold

$$(\Delta' \vdash M') \in B^{\Delta}_{\tau},\ \ M' = \mathrm{Constr}(n, X)\vec{N},\ \ M' = \lambda x : A.B,\ \ M' = \Pi x : A.B$$
$$\text{or}\ \ M' = \mathrm{Ind}(X : s)\{\vec{C}\}.$$

If M' is of type $\rho(I)$ and $M' \notin B^{\Delta}_{\rho(I)}$ then $M' = \mathrm{Constr}(n, X)\vec{N}$. By the Generation Lemma 17 the other forms mentioned above are not possible. □

Let $C(X) = \Pi\vec{x} : \vec{t}.X\vec{t'}$ be a type of constructor in X. Let Δ, Γ be two contexts and let $\langle \xi, \rho \rangle$ be a constructor valuation which satisfies Γ at Δ. Suppose the vector $\vec{t}$ has length n. Let $(\Delta_i, N_i, P_i)_{i=1}^n$ be an appropriate sequence of arguments for $(\vec{x} : \vec{t})$ at $\langle \xi, \rho \rangle$ in Δ. We say that it is an *adequate sequence of arguments for $(\vec{x} : \vec{t})$ at $\langle \xi, \rho \rangle$ in Δ* if for every $j = 1, \ldots, n$ we have

$$(\Delta_j \vdash N_j) \in [\Gamma, (x_i : t_i)_{i=1}^{j-1} \vdash t_j]_{\xi_{j-1}|_{\Delta_j}, \rho_{j-1}, \Delta_j}$$

Here we state an auxiliary technical lemma which expresses the fact that elimination for inductive objects behaves in the expected way.

Lemma 96. *Let $I = \mathrm{Ind}(X : *^t)\{\vec{C}\}$ be an inductive type with n constructors. Suppose that*

1. *Γ and Δ are contexts;*
2. *Q is a term such that $\Gamma \vdash Q : I \to s$ where s is $*^t$ or $*^p$;*
3. *for every sequence of types $\vec{\tau}$ in the context Γ', for every ordinal number $\alpha' < \alpha$, and for every object valuation $\langle \xi', \rho' \rangle$ which satisfies $(\Gamma', x : (\Pi\vec{y} : \vec{\tau}.I), \vec{y} : \vec{\tau})$ at Δ', if*

$$(\Delta' \vdash \rho'(x\vec{y})) \in F^{\alpha'}_{(\Gamma', x:(\Pi\vec{y}:\vec{\tau}.I), \vec{y}:\vec{\tau}), I, \xi', \rho', \Delta'}(\bigcap SAT^{\Delta'}_{\rho'(I)})$$

   then

$$(\Delta' \vdash \mathrm{Elim}(\rho'(I), \rho'(Q), \rho'(x\vec{y}))\{\rho'(\vec{f})\}) \in [\Gamma', x : (\Pi\vec{y} : \vec{\tau}.I), \vec{y} : \vec{\tau} \vdash Q(x\vec{y})]_{\xi', \rho', \Delta'};$$

4. *$\vec{\sigma}$ is a sequence of types in the context Γ and $\langle \xi, \rho \rangle$ is an object valuation which satisfies $(\Gamma, x : (\Pi\vec{y} : \vec{\sigma}.I), \vec{y} : \vec{\sigma})$ at Δ;*

5. *for every* $i = 0, \ldots, n-1$ *we have* $(\Delta \vdash \rho(f_i)) \in [\Gamma \vdash \triangle\{C_i(I), f_i, \mathrm{Constr}(i, I)\}]_{\xi,\rho,\Delta}$*;*

6. $(\Delta \vdash \rho(x\vec{y})) \in F^{\alpha}_{\Gamma,I,\xi,\rho,\Delta}(\bigcap SAT^{\Delta}_{\rho(I)})$.

Then

$$(\Delta \vdash \mathrm{Elim}(\rho(I), \rho(Q), \rho(x\vec{y}))\{\rho(\vec{f})\}) \in [\Gamma, x : (\Pi\vec{y} : \vec{\sigma}.I), \vec{y} : \vec{\sigma} \vdash Q(x\vec{y})]_{\xi,\rho,\Delta}.$$

Proof. Let $M = \rho(x\vec{y})$. Recall that $F^{\alpha}_{\Gamma,I,\xi,\rho,\Delta}(\bigcap SAT^{\Delta}_{\rho(I)}) \subseteq [\Gamma \vdash I]_{\xi,\rho,\Delta}$. By Lemma 95 the term M key-reduces either

- to a base term M', or
- to a constructor term $\mathrm{Constr}(m, X)\vec{N}$.

Case 1 (Reduction to a base term): If $M \to^*_k M'$ and $(\Delta \vdash M') \in B^{\Delta}_{\rho(I)}$ then

$$(\Delta \vdash \mathrm{Elim}(\rho(I), \rho(Q), M')\{\rho(\vec{f})\}) \in B^{\Delta}_{\rho(Q)M'} = B^{\Delta}_{\rho(Q)M} \subseteq [\Gamma, x : (\Pi\vec{y} : \vec{\sigma}.I), \vec{y} : \vec{\sigma} \vdash Q(x\vec{y})]_{\xi,\rho,\Delta}$$

and

$$\mathrm{Elim}(\rho(I), \rho(Q), M)\{\rho(\vec{f})\} \to^*_k \mathrm{Elim}(\rho(I), \rho(Q), M')\{\rho(\vec{f})\}.$$

By Lemma 70 we know that $(\Delta \vdash \mathrm{Elim}(\rho(I), \rho(Q), M)\{\rho(\vec{f})\}) \in SN^{\Delta}_{\rho(Q)M}$. As the set $[\Gamma, x : (\Pi\vec{y} : \vec{\sigma}.I), \vec{y} : \vec{\sigma} \vdash Q(x\vec{y})]_{\xi,\rho,\Delta}$ is saturated we indeed get

$$(\Delta \vdash \mathrm{Elim}(\rho(I), \rho(Q), M)\{\rho(\vec{f})\}) \in [\Gamma, x : (\Pi\vec{y} : \vec{\sigma}.I), \vec{y} : \vec{\sigma} \vdash Q(x\vec{y})]_{\xi,\rho,\Delta}.$$

Case 2 (Reduction to a constructor term): Otherwise $M \to^*_k \mathrm{Constr}(m, X)\vec{N}$. Suppose $C_m(X) = \Pi\vec{z} : \vec{\tau}.X$. Let

$$U = [\Gamma, x : (\Pi\vec{y} : \vec{\sigma}.I), \vec{y} : \vec{\sigma} \vdash x\vec{y}]_{\xi,\rho,\Delta}.$$

Then $U = \langle m, \vec{P}\rangle \in V_{\Delta}(M)$ and for each object N_i there is a corresponding set $P_i \in V_{\Delta}(N_i)$. Moreover

$$\mathrm{Elim}(\rho(I), \rho(Q), M)\{\rho(\vec{f})\} \to^*_k \rho(f_m)\vec{e}[C_m(I), \vec{N}, \rho(I), \rho(Q), \rho(\vec{f})]$$

and by Lemma 70 we have $(\Delta \vdash \mathrm{Elim}(\rho(I), \rho(Q), M)\{\rho(\vec{f})\}) \in SN^{\Delta}_{\rho(Q)M}$. By the assumption

$$(\Delta \vdash \rho(f_m)) \in [\Gamma \vdash \triangle\{C_m(I), Q, \mathrm{Constr}(m, I)\}]_{\rho,\xi,\Delta}.$$

We will prove that

$$(\Delta \vdash \rho(f_m)\vec{e}[C_m(I), \vec{N}, \rho(I), \rho(Q), \rho(\vec{f})]) \in [\Gamma, x : (\Pi\vec{y} : \vec{\sigma}.I), \vec{y} : \vec{\sigma} \vdash Q(x\vec{y})]_{\xi,\rho,\Delta}. \quad (5.5)$$

For every j less than the length of the sequence $\vec{\tau}$ we introduce the following abbreviations.

$$\begin{array}{ll}
\Gamma^0 = \Gamma, X : *^t, & \Gamma^j = \Gamma^{j-1}, z_j : \tau_j, \\
\overline{\Gamma}^0 = \Gamma, & \overline{\Gamma}^j = \overline{\Gamma}^{j-1}, z_j : \tau_j[I/X], \\
\xi^0 = \xi, & \xi^j = \xi^{j-1}; z_j := Can_{\Delta}(N_j), \\
\bar{\xi}^0 = \xi, & \bar{\xi}^j = \bar{\xi}^{j-1}; z_j := C_j, \\
\rho^0 = \rho, & \rho^j = \rho^{j-1}; z_j := N_j.
\end{array}$$

In order to show the claim (5.5) we will prove that the arguments applied to $\rho(f_m)$ are good: if $\tau_j = \Pi\vec{w} : \vec{T}.X$ then

$$(\Delta \vdash (\vec{e}[C_m(I), \vec{N}, \rho(I), \rho(Q), \rho(\vec{f})])_j) \in [\overline{\Gamma}^{j-1} \vdash \tau_j[I/X]]_{\bar{\xi}^{j-1},\rho^{j-1},\Delta} \tag{5.6}$$

$$(\Delta \vdash (\vec{e}[C_m(I), \vec{N}, \rho(I), \rho(Q), \rho(\vec{f})])_j^R) \in [\overline{\Gamma}^j \vdash \Pi\vec{y} : \vec{T}.Q(z_j\vec{y})]_{\bar{\xi}^j,\rho^j,\Delta}. \tag{5.7}$$

We show (5.6): By the definition of $F^{\alpha}_{\Gamma,I,\xi,\rho,\Delta}(\bigcap SAT^{\Delta}_{\rho(I)})$ we know that for each j we have

$$(\Delta \vdash N_j) \in \mathit{Interp}(\Gamma^{j-1} \vdash \tau_j)_{\xi^{j-1},\rho^{j-1},\Delta,X,F^{\alpha-1}_{\Gamma,I,\xi,\rho,\Delta}(\bigcap SAT^{\Delta}_{\rho(I)})}.$$

By the monotonicity of Interp we get

$$(\Delta \vdash N_j) \in \mathit{Interp}(\Gamma^{j-1} \vdash \tau_j)_{\xi^{j-1},\rho^{j-1},\Delta,X,[\Gamma\vdash I]_{\xi,\rho,\Delta}}.$$

By Lemma 54

$$\mathit{Interp}(\Gamma^{j-1} \vdash \tau_j)_{\xi^{j-1},\rho^{j-1},\Delta,X,[\Gamma\vdash I]_{\xi,\rho,\Delta}} = [\Gamma^{j-1} \vdash \tau_j]_{(\xi^{j-1};X:=[\Gamma\vdash I]_{\xi,\rho,\Delta}),(\rho^{j-1};X:=\rho(I)),\Delta}.$$

Note that ξ^{j-1} and $\bar{\xi}^{j-1}$ only differ in subset or object variables. Thus by Lemma 55 and Lemma 84

$$[\Gamma^{j-1} \vdash \tau_j]_{(\xi^{j-1};X:=[\Gamma\vdash I]_{\xi,\rho,\Delta}),(\rho^{j-1};X:=\rho(I)),\Delta} = [\overline{\Gamma}^{j-1} \vdash \tau_j[I/X]]_{\bar{\xi}^{j-1},\rho^{j-1},\Delta}.$$

Thus for each j we get (5.6):

$$(\Delta \vdash (\vec{e}[C_m(I), \vec{N}, \rho(I), \rho(Q), \rho(\vec{f})])_j) \in [\overline{\Gamma}^{j-1} \vdash \tau_j[I/X]]_{\bar{\xi}^{j-1},\rho^{j-1},\Delta}.$$

Consider the case when N_j is a recursive argument. To prove (5.7) recall that

$$(\vec{e}[C_m(I), \vec{N}, \rho(I), \rho(Q), \rho(\vec{f})])_j^R = \lambda\vec{x} : \rho(\vec{T}).\mathrm{Elim}(\rho(I), \rho(Q), N_j\vec{x})\{\rho(\vec{f})\}.$$

Suppose that $(\Delta_l, M_l, C_l)_{l=0}^p$ is an adequate sequence of arguments for $(\vec{y} : \vec{T})$ at $\langle\xi^j, \rho^j\rangle$ in Δ and that $\langle\xi^{j,l}, \rho^{j,l}\rangle$ is the sequence of constructor valuations associated with it. Note that then $(\Delta_l, M_l, C_l)_{l=0}^p$ is an adequate sequence of arguments for $(\vec{y} : \vec{T})$ at $\langle\bar{\xi}^j, \rho^j\rangle$ in Δ and let $\langle\bar{\xi}^{j,l}, \rho^{j,l}\rangle$ be the sequence of constructor valuations associated with it. We will show that

$$(\Delta_p \vdash \mathrm{Elim}(\rho(I), \rho(Q), N_j\vec{M})\{\rho(\vec{f})\}) \in [\overline{\Gamma}^j, (\vec{y} : \vec{T}) \vdash Q(z_j\vec{y})]_{\bar{\xi}^{j,p},\rho^{j,p},\Delta_p}.$$

By the assumption we know that there exists $\alpha' < \alpha$ such that

$$(\Delta \vdash N_j) \in \mathit{Interp}(\Gamma^{j-1} \vdash \tau_j)_{\xi^{j-1},\rho^{j-1},\Delta,X,F^{\alpha'}_{\Gamma,I,\xi,\rho,\Delta}(\bigcap SAT^{\Delta}_{\rho(I)})}.$$

By the definition of interpretation for Interp we have

$$(\Delta_p \vdash N_j\vec{M}) \in F^{\alpha'}_{\Gamma,I,\xi,\rho,\Delta}(\bigcap SAT^{\Delta}_{\rho(I)})|_{\Delta_p}.$$

By Lemma 58

$$F^{\alpha'}_{\Gamma,I,\xi,\rho,\Delta}(\bigcap SAT^{\Delta}_{\rho(I)})|_{\Delta_p} = F^{\alpha'}_{\Gamma,I,\xi|_{\Delta_p},\rho,\Delta_p}(\bigcap SAT^{\Delta}_{\rho(I)}|_{\Delta_p}) = F^{\alpha'}_{\Gamma,I,\xi|_{\Delta_p},\rho,\Delta_p}(\bigcap SAT^{\Delta_p}_{\rho(I)}).$$

But by Lemma 77

$$F^{\alpha'}_{\Gamma,I,\xi|_{\Delta_p},\rho,\Delta_p}(\bigcap SAT^{\Delta_p}_{\rho(I)}) = F^{\alpha'}_{(\overline{\Gamma}^{j-1},z_j:(\Pi\vec{y}:\vec{T}.I),\vec{y}:\vec{T}),I,\bar{\xi}^{j,p}|_{\Delta_p},\bar{\rho}^{j,p},\Delta_p}(\bigcap SAT^{\Delta_p}_{\rho(I)}).$$

Thus we have

$$(\Delta_p \vdash N_j\vec{M}) \in F^{\alpha'}_{(\overline{\Gamma}^{j-1},z_j:(\Pi\vec{y}:\vec{T}.I),\vec{y}:\vec{T}),I,\bar{\xi}^{j,p}|_{\Delta_p},\bar{\rho}^{j,p},\Delta_p}(\bigcap SAT^{\Delta_p}_{\rho(I)}).$$

Note that $\langle\bar{\xi}^{j,p},\rho^{j,p}\rangle$ is an object valuation which satisfies $(\overline{\Gamma}^{j-1}, z_j : (\Pi\vec{y} : \vec{T}.I), \vec{y} : \vec{T})$ at Δ. By assumption 3 we have

$$(\Delta_p \vdash \mathrm{Elim}(\rho(I), \rho(Q), N_j\vec{M})\{\rho(\vec{f})\}) \in [\overline{\Gamma}^{j-1}, z_j : (\Pi\vec{y} : \vec{T}.I), \vec{y} : \vec{T} \vdash Q(z_j\vec{y})]_{\bar{\xi}^{j,p},\rho^{j,p},\Delta_p}.$$

Hence

$$(\Delta \vdash \lambda\vec{x} : \rho(\vec{T}).\mathrm{Elim}(\rho(I), \rho(Q), N_j\vec{x})\{\rho(\vec{f})\}) \in [\overline{\Gamma}^{j-1}, z_j : (\Pi\vec{y} : \vec{T}.I) \vdash \Pi\vec{y} : \vec{T}.Q(z_j\vec{y})]_{\bar{\xi}^j,\rho^j,\Delta}.$$

Thus we have shown (5.7):

$$(\Delta \vdash (\vec{e}[C_m(I), \vec{N}, \rho(I), \rho(Q), \rho(\vec{f})])^R_j) \in [\overline{\Gamma}^j \vdash \Pi\vec{y} : \vec{T}.Q(z_j\vec{y})]_{\bar{\xi}^j,\rho^j,\Delta}.$$

We now conclude the proof. Let $D_j \in V_\Delta(\lambda\vec{y} : \rho(\vec{T}).\mathrm{Elim}(\rho(I), \rho(Q), N_i\vec{y})\{\rho(\vec{f})\})$ for j such that $\tau_j = \Pi\vec{y} : \vec{T}.I$ is a recursive argument. Let us define a sequence of contexts $\vec{\Gamma'}$ as

$$\begin{array}{ll} \Gamma'_0 = \Gamma, & \\ \Gamma'_j = \Gamma'_{j-1}, p_j : \tau_j, q_j : \Pi\vec{y} : \vec{T}.(Q(p_j\vec{y})), & \text{if } \tau_i = \Pi\vec{y} : \vec{T^i}.I \text{ is a recursive type,} \\ \Gamma'_j = \Gamma'_{j-1}, p_j : \tau_j, & \text{if } \tau_i \text{ is not a recursive type,} \end{array}$$

a sequence of valuations $\vec{\xi'}$ as

$$\begin{array}{ll} \xi'_0 = \xi, & \\ \xi'_j = \xi_{j-1}; p_j := C_j, & \text{if } N_j \text{ is not a recursive argument,} \\ \xi'_j = \xi'_{j-1}; p_j := C_j; q_j := D_j, & \text{if } N_j \text{ is a recursive argument,} \end{array}$$

and a sequence of substitutions $\vec{\rho'}$ as

$$\begin{array}{ll} \rho'_0 = \rho, & \\ \rho'_j = \rho'_{j-1}; p_j := N_j, & \text{if } N_j \text{ is not a recursive argument,} \\ \multicolumn{2}{l}{\rho'_j = \rho'_{j-1}; p_j := N_j; q_j := \lambda\vec{y} : \rho(\vec{T}).\mathrm{Elim}(\rho(I), \rho(Q), N_j\vec{y})\{\rho(\vec{f})\},} \\ & \text{if } N_j \text{ is a recursive argument.} \end{array}$$

Thus by the definition of $[\Gamma \vdash \triangle\{C_m(I), Q, \mathrm{Constr}(m, I)\}]_{\rho,\xi,\Delta}$ we have

$$(\Delta \vdash \rho(f_m)\vec{e}[C_m(I), \vec{N}, \rho(I), \rho(Q), \rho(\vec{f})]) \in [\Gamma' \vdash Q(\mathrm{Constr}(m, I)\vec{x})]_{\xi',\rho',\Delta}.$$

But

$$
\begin{aligned}
&[\Gamma' \vdash Q(\mathrm{Constr}(m, I)\vec{p})]_{\xi',\rho',\Delta} \\
&= [\Gamma' \vdash Q]_{\xi',\rho',\Delta}(\Delta \vdash \mathrm{Constr}(m, \rho'(I))\vec{N}, \langle m, \vec{C}\rangle) \\
&= [\Gamma \vdash Q]_{\xi,\rho,\Delta}(\Delta \vdash \mathrm{Constr}(m, \rho(I))\vec{N}, \langle m, \vec{C}\rangle) && \text{(by Lemma 77)} \\
&= [\Gamma \vdash Q]_{\xi,\rho,\Delta}(\Delta \vdash M, \langle m, \vec{C}\rangle) \\
&= [\Gamma, x : (\Pi\vec{y} : \vec{\sigma}.I), \vec{y} : \vec{\sigma} \vdash Q]_{\xi,\rho,\Delta}(\Delta \vdash M, \langle m, \vec{C}\rangle) && \text{(by Lemma 77)} \\
&= [\Gamma, x : (\Pi\vec{y} : \vec{\sigma}.I), \vec{y} : \vec{\sigma} \vdash Q]_{\xi,\rho,\Delta}(\Delta \vdash M, [\Gamma, x : (\Pi\vec{y} : \vec{\sigma}.I), \vec{y} : \vec{\sigma} \vdash x\vec{y}]_{\xi,\rho,\Delta}) \\
&= [\Gamma, x : (\Pi\vec{y} : \vec{\sigma}.I), \vec{y} : \vec{\sigma} \vdash Q(x\vec{y})]_{\xi,\rho,\Delta}.
\end{aligned}
$$

Thus we have

- $(\Delta \vdash \mathrm{Elim}(\rho(I), \rho(Q), \rho(x\vec{y}))\{\rho(\vec{f})\}) \in SN^{\Delta}_{\rho(Q)M}$,
- $\mathrm{Elim}(\rho(I), \rho(Q), \rho(x\vec{y}))\{\rho(\vec{f})\} \to^*_k \rho(f_m)\vec{e}[C_m(I), \vec{N}, \rho(I), \rho(Q), \rho(\vec{f})]$,
- $(\Delta \vdash \rho(f_m)\vec{e}[C_m(I), \vec{N}, \rho(I), \rho(Q), \rho(\vec{f})]) \in [\Gamma, x : (\Pi\vec{y} : \vec{\sigma}.I), \vec{y} : \vec{\sigma} \vdash Q(x\vec{y})]_{\xi,\rho,\Delta}$.

As $[\Gamma, x : (\Pi\vec{y} : \vec{\sigma}.I), \vec{y} : \vec{\sigma} \vdash Q(x\vec{y})]_{\xi,\rho,\Delta}$ is a saturated set we get

$$(\Delta \vdash \mathrm{Elim}(\rho(I), \rho(Q), \rho(c)\rho(\vec{y}))\{\rho(\vec{f})\}) \in [\Gamma, x : (\Pi\vec{y} : \vec{\sigma}.I), \vec{y} : \vec{\sigma} \vdash Q(x\vec{y})]_{\xi,\rho,\Delta}. \qquad \square$$

Lemma 97. *Suppose that I is an inductive predicate. Let $\Pi\vec{x} : \vec{\tau}.I\vec{t'}$ be such that I does not occur as a subterm neither in $\vec{\tau}$ nor in $\vec{t'}$. If $(\Delta' \vdash N) \in [\Gamma \vdash \Pi\vec{x} : \vec{\tau}.I\vec{t'}]_{\xi,\rho,\Delta}$ and n is the length of $\vec{\tau}$ then there exists an ordinal α such that for each adequate sequence $(\Delta_i, a_i, C_i)_i$, of arguments for $(\vec{x} : \vec{\tau})$ at $\langle\xi, \rho\rangle$ in Δ, we have*

$$(\Delta_n \vdash N\vec{a}) \in H^{\alpha}_{(\Gamma,\vec{x}:\vec{\tau}),I,\xi_n,\rho_n,\Delta_n}(Min^{\Delta_n}_{\rho_n(I)})(\Delta \vdash \rho(t'_i), [\Gamma \vdash t'_i]_{\xi_n,\rho_n,\Delta_n})^n_{i=1}.$$

Proof. By the definition of $[\Gamma \vdash \Pi\vec{x} : \vec{\tau}.I\vec{t'}]_{\xi,\rho,\Delta}$ we know that for each adequate sequence $(\Delta_i, a_i, C_i)_i$ of arguments for $(\vec{x} : \vec{\tau})$ at $\langle\xi, \rho\rangle$ in Δ, if $\langle\xi_i, \rho_i\rangle$ is the sequence of constructor valuations associated with it then we have

$$(\Delta_n \vdash N\vec{a}) \in [\Gamma, \vec{x} : \vec{\tau} \vdash I\vec{t'}]_{\xi_n,\rho_n,\Delta_n}.$$

Thus for each $(\Delta_i, a_i, C_i)_i$ there exists the least ordinal number $\alpha_{(\Delta_i,a_i,C_i)_i}$ such that

$$(\Delta_n \vdash N\vec{a}) \in H^{\alpha_{(\Delta_i,a_i,C_i)_i}}_{(\Gamma,\vec{x}:\vec{\tau}),I,\xi_n,\rho_n,\Delta_n}(Min^{\Delta_n}_{\rho_n(I)})(\Delta \vdash \rho(t'_i), [\Gamma \vdash t'_i]_{\xi_n,\rho_n,\Delta_n})^n_{i=1}.$$

Take $\alpha = sup\{\alpha_{(\Delta_i,a_i,C_i)_i}\}$. Since $H_{(\Gamma,\vec{x}:\vec{\tau}),I,\xi_n,\rho_n,\Delta_n}$ is a monotone function such α satisfies the condition in the lemma. $\square$

Lemma 98. *Suppose I is an inductive predicate, $C(X) = \Pi\vec{x} : \vec{\tau}.X\vec{t}$ is a type of constructor of I, and $\alpha > 0$ is an ordinal number. Assume that*

1. *Γ, Δ are two contexts;*
2. *$\langle\xi, \rho\rangle$ is a constructor valuation which satisfies Γ at Δ;*

3. $(\Delta_i, N_i, C_i)_{i=1}^m$ *is an adequate sequence of arguments for* $(\vec{x} : \vec{\tau})$ *at* $\langle \xi, \rho \rangle$ *in* Δ;

4. *for each recursive argument* τ_j *there exists an ordinal* $\alpha' < \alpha$ *such that if* $\tau_j = \Pi \vec{y} : \vec{\sigma}.I\vec{t'}$ *and* $(\Delta'_i, a_i, C'_i)_{i=1}^n$ *is an adequate sequence of arguments for* $(\vec{y} : \vec{\sigma})$ *at* $\langle \xi_j, \rho_j \rangle$ *then*
$$(\Delta'_n \vdash N_j \vec{a}) \in H^{\alpha'}_{(\Gamma,(x_i:\tau_i)_{i=0}^{j-1},\vec{y}:\vec{\sigma}),I,\xi_n,\rho_n,\Delta_n}(Min^{\Delta_n}_{\rho_n(I)})(\Delta \vdash \rho_n(t'_i), [\Gamma \vdash t'_i]_{\xi_n,\rho_n,\Delta_n})_{i=1}^n.$$

Then
$$(\Delta_m \vdash \mathrm{Constr}(k, \rho(I))\vec{N}) \in H^{\alpha}_{(\Gamma,\vec{x}:\vec{\tau}),I,\xi_m,\rho_m,\Delta_m}(Min^{\Delta_m}_{\rho_m(I)})(\Delta \vdash \rho_m(t_i), [\Gamma \vdash t_i]_{\xi_m,\rho_m,\Delta_m})_{i=1}^n.$$

Proof. There are two cases.

Case 1: α is a successor ordinal. Let $H^{\alpha-1}$ denote $H^{\alpha-1}_{(\Gamma,\vec{x}:\vec{\tau}),I,\xi_m,\rho_m,\Delta_m}(Min^{\Delta_m}_{\rho_m(I)})$. Recall that
$$H^{\alpha}_{(\Gamma,\vec{x}:\vec{\tau}),I,\xi_m,\rho_m,\Delta_m}(Min^{\Delta_m}_{\rho_m(I)})(\Delta_i, \rho_i(t_i), [\Gamma \vdash t_i]_{\xi_i,\rho_i,\Delta_i})_{i=1}^m = (\bigcap SAT^{\Delta_m}_{\rho_m(I\vec{t})}) \cup$$
$$h_{(\Gamma,\vec{x}:\vec{\tau}),I,\xi_m,\rho_m,\Delta_m}(H^{\alpha-1})((\Delta_i, \rho_i(t_i), [\Gamma \vdash t_i]_{\xi_i,\rho_i,\Delta_i})_{i=1}^m)$$

and $h_{\Gamma,I,\xi_m,\rho_m,\Delta_m}(S)((\Sigma_i, u_i, U_i)_{i=1}^n)$ consists of simple sequents $(\Delta' \vdash m) \in SN^{\Sigma_n}_{\rho_m(I)\vec{u}}$ such that

for every context Δ'' and every term J such that $(\Delta'' \vdash J) \in SN^{\Sigma_n}_{\rho_m(A)}$ and $J =_{\beta\iota} \rho_m(I)$,
for every context Δ''' and every term Q such that $(\Delta''' \vdash Q) \in [\Gamma \vdash A]_{\xi_m|_{\Delta''},\rho_m,\Delta''}$,
for every $P \in V_{\Delta'''}(Q)$, ,
for every context Δ'''' and for every vector $\vec{f}$ such that
$$(\Delta'''' \vdash f_i) \in [\Gamma, q : A, X : A \vdash \triangle\{C_i(X), q\}]_{(\xi_m|_{\Delta''''};X:=S|_{\Delta''''};q:=P),(\rho_m;X:=J;q:=Q),\Delta''''}$$
we have $(\Delta'''' \vdash \mathrm{Elim}(J, Q, \vec{u}, m)\{\vec{f}\}) \in P((\Sigma_i, u_i, U_i)_{i=1}^n)$.

We obviously have
$$(\Delta_m \vdash \mathrm{Constr}(k, \rho_m(I))\vec{N}) \in SN^{\Delta_m}_{\rho_m(I)}.$$

Let J, Q and $\vec{f}$ be such that
$$(\Delta'' \vdash J) \in SN^{\Sigma_n}_{\rho_m(A)}, J =_{\beta\iota} \rho_m(I),$$
$$(\Delta''' \vdash Q) \in [\Gamma \vdash A]_{\xi_m|_{\Delta''},\rho_m,\Delta''}, P \in V_{\Delta'''}(Q)$$
$$(\Delta'''' \vdash f_i) \in [\Gamma, q : A, X : A \vdash \triangle\{C_i(X), q\}]_{(\xi_m;q:=P,X:=H^{\alpha-1}),(\rho_m;q:=Q,X:=J),\Delta'''}.$$

Note that
$$\mathrm{Elim}(J, Q, \mathrm{Constr}(k, \rho(I))\vec{N})\{\vec{f}\} \rightarrow_k f_k \vec{e}[C_k(J), \vec{N}, J, Q, \vec{f}].$$

By assumptions 3 and 4 and the definition of
$$[\Gamma, q : A, X : A \vdash \triangle\{C_i(X), q\}]_{(\xi_m;q:=P,X:=H^{\alpha-1}),(\rho_m;q:=Q,X:=J),\Delta'''}$$

we get
$$(\Delta'''' \vdash f_k \vec{e}[C_k(J), \vec{N}, J, Q, \vec{f}]) \in P(\Delta \vdash \rho(t_i), [\Gamma \vdash t_i]_{\xi_m,\rho_m,\Delta_m})_{i=1}^m.$$

By Corollary 38 the term $\mathrm{Elim}(J, Q, \mathrm{Constr}(k, \rho(I))\vec{N})\{\vec{f}\}$ is strongly normalizing. But P is a saturated set and thus
$$(\Delta'''' \vdash \mathrm{Elim}(\rho_m(I), Q, \mathrm{Constr}(k, \rho_m(I))\vec{N})\{\vec{f}\}) \in P(\Delta \vdash \rho(t_i), [\Gamma \vdash t_i]_{\xi_m,\rho_m,\Delta_m})_{i=1}^m.$$

Hence we get the conclusion.

Case 2: α is a limit ordinal. There is only a finite number of recursive arguments. Each recursive argument has its own α' as in the statement of the lemma. One of those (finitely many) ordinals is the greatest, we denote it by α''. The conclusion follows from case 1 for $\alpha'' + 1$ and the monotonicity of H. □

We now prove the adequacy lemma.

Lemma 99 (Adequacy Lemma). *Let* $\Gamma \vdash M : T$.

1. *If* $\langle \xi, \rho \rangle$ *is an object valuation which satisfies* Γ *at* Δ *then* $(\Delta \vdash \rho(M)) \in [\Gamma \vdash T]_{\xi,\rho,\Delta}$.
2. *If* M *is neither a proof nor a small object then for every pair* $\langle \xi, \rho \rangle$, $\langle \xi', \rho \rangle$ *of object valuations which satisfy* Γ *at* Δ *and* $\xi \simeq_{\Gamma,\xi,\xi',\rho,\Delta} \xi'$ *we have*

$$[\Gamma \vdash M]_{\xi,\rho,\Delta} \simeq_{\Gamma,\xi,\xi',\rho,\Delta} [\Gamma \vdash M]_{\xi',\rho,\Delta}.$$

Proof. We proceed by induction with respect to the length of the longest reduction in any type-like term (i.e. a type, a formula, a kind or a sort) occurring in the derivation of $\Gamma \vdash M : T$. By Lemma 31 every type-like term is strongly normalizing, thus every reduction sequence beginning in a type-like term is finite. Suppose for all non-proofs N in the derivation $\Gamma \vdash M : T$ every reduction sequence beginning in N has length at most n. For a fixed n we proceed by auxiliary induction with respect to the structure of the derivation. The cases depend on the last rule used in the derivation.

(Var) Part 1 follows from the assumption that $\langle \xi, \rho \rangle$ is an object valuation which satisfies Γ at Δ. Part 2 follows from the assumption that

$$[\Gamma \vdash M]_{\xi,\rho,\Delta} = \xi(M) \simeq_{\Gamma,\xi,\xi',\rho,\Delta} \xi'(M) = [\Gamma \vdash M]_{\xi',\rho,\Delta}.$$

(Weak) Part 1 is a consequence of Lemma 77. Part 2 follows from the auxiliary induction hypothesis and Lemma 87.

(Conv) We prove Part 1. We have

$$\frac{\Gamma \vdash M : T' \qquad \Gamma \vdash T : s \qquad T' =_{\beta\iota} T}{\Gamma \vdash M : T}$$

By the auxiliary induction hypothesis $(\Delta \vdash \rho(M)) \in [\Gamma \vdash T']_{\xi,\rho,\Delta}$. By the Church-Rosser property there exists T'' such that

$$T \to^*_{\beta\iota} T'' \text{ and } T' \to^*_{\beta\iota} T'' \text{ and } \Gamma \vdash T'' : s.$$

By the main induction hypothesis and Lemma 94

$$[\Gamma \vdash T']_{\xi,\rho,\Delta} \simeq_{\Gamma,\xi,\xi,\rho,\Delta} [\Gamma \vdash T'']_{\xi,\rho,\Delta},$$
$$[\Gamma \vdash T]_{\xi,\rho,\Delta} \simeq_{\Gamma,\xi,\xi,\rho,\Delta} [\Gamma \vdash T'']_{\xi,\rho,\Delta}.$$

Thus

$$[\Gamma \vdash T']_{\xi,\rho,\Delta} \simeq_{\Gamma,\xi,\xi,\rho,\Delta} [\Gamma \vdash T]_{\xi,\rho,\Delta}.$$

By the definition of the relation $\simeq_{\Gamma,\xi,\xi,\rho,\Delta}$

$$[\Gamma \vdash T']_{\xi,\rho,\Delta} = [\Gamma \vdash T]_{\xi,\rho,\Delta}$$

and thus

$$(\Delta \vdash \rho(M)) \in [\Gamma \vdash T]_{\xi,\rho,\Delta}.$$

Part 2 follows from the auxiliary induction hypothesis.

(Ax) Part 1 follows immediately from‘ the definition of $[\Gamma \vdash T]_{\xi,\rho,\Delta}$. Part 2 is immediate consequence of the definition of $[\Gamma \vdash T]_{\xi,\rho,\Delta}$.

(Abs) We prove Part 1. We have $M = \lambda x : A.P$ and

$$\frac{\Gamma, x : A \vdash P : B}{\Gamma \vdash (\lambda x : A.P) : (\Pi x : A.B)}$$

Let $u = \rho(\lambda x : A.P) = \lambda x : \rho(A).\rho(P)$. (We may assume that the variable x is not free in $\rho(y)$ for every $y \in FV(P) - \{x\}$.) We want to prove that

$$(\Delta \vdash u) \in [\Gamma \vdash \Pi x : A.B]_{\xi,\rho,\Delta}.$$

We know that $\Delta \vdash u : \rho(\Pi x : A.B)$. By the auxiliary induction hypothesis we get that $\rho(P)$ is strongly normalizing. This entails that the term u is strongly normalizing. Let

$$\Delta'' \supseteq \Delta,\ (\Delta'' \vdash a) \in [\Gamma \vdash A]_{\xi|_{\Delta''},\rho,\Delta''},\ C \in V_{\Delta''}(a).$$

Then $\langle \xi', \rho' \rangle$ where

$$\rho' = \rho; x := a, \qquad \xi' = \xi|_{\Delta''}; x := C$$

is an object valuation which satisfies $(\Gamma, x : A)$ at Δ''. Moreover

$$ua = (\lambda x : \rho(A).\rho(P))a \to_k (\rho; x := a)(P) = \rho'(P).$$

By the auxiliary induction hypothesis

$$(\Delta'' \vdash \rho'(P)) \in [\Gamma, x : A \vdash B]_{\xi'|_{\Delta''},\rho',\Delta''}.$$

By Corollary 38 the term ua is strongly normalizing. One can easily see that the set $[\Gamma, x : A \vdash B]_{\xi'|_{\Delta''},\rho',\Delta''}$ is saturated, so we have

$$(\Delta'' \vdash ua) \in [\Gamma, x : A \vdash B]_{\xi'|_{\Delta''},\rho',\Delta''}.$$

Thus

$$(\Delta \vdash u) \in [\Gamma \vdash \Pi x : A.B]_{\xi,\rho,\Delta}.$$

Part 2 follows from the auxiliary induction hypothesis.

(App) We prove Part 1. We have $M = AB$ and

$$\frac{\Gamma \vdash M : (\Pi x : A.B) \qquad \Gamma \vdash N : A}{\Gamma \vdash MN : B[x := N]}$$

We only consider the case when A is a large type or $*^p$. The other case is similar. By the auxiliary induction hypothesis we know that

$$(\Delta \vdash \rho(M)) \in [\Gamma \vdash \Pi x : A.B]_{\xi,\rho,\Delta},$$
$$(\Delta \vdash \rho(N)) \in [\Gamma \vdash A]_{\xi,\rho,\Delta}.$$

Then by the definition of $[\Gamma \vdash \Pi x : A.B]_{\xi,\rho,\Delta}$,

$$(\Delta \vdash \rho(M)\rho(N)) \in [\Gamma, x : A \vdash B]_{(\xi;x:=[\Gamma\vdash N]_{\xi,\rho,\Delta}),(\rho;x:=\rho(N)),\Delta}.$$

By Lemma 84

$$[\Gamma, x : A \vdash B]_{(\xi;x:=[\Gamma\vdash N]_{\xi,\rho,\Delta}),(\rho;x:=\rho(N)),\Delta} = [\Gamma \vdash B[N/x]]_{\xi,\rho,\Delta}.$$

Therefore we get the conclusion.

Part 2 follows from the auxiliary induction hypothesis.

(Prod) We prove Part 1. If $M = \Pi x : A.B$ then the derivation ends with

$$\frac{\Gamma \vdash A : s_1 \qquad \Gamma, x : A \vdash B : s_2}{\Gamma \vdash \Pi x : A.B : s_3}$$

By the auxiliary induction hypothesis

$$(\Delta \vdash \rho(A)) \in [\Gamma \vdash s_1]_{\xi,\rho,\Delta} = SN^{\Delta}_{s_1}.$$

and for all $C \in V_{\Delta,x:\rho(A)}(\rho(x))$ it holds that $\langle(\xi; x := C), (\rho; x := x)\rangle$ is an object valuation which satisfies $(\Gamma, x : A)$ at $(\Delta, x : \rho(A))$. Thus

$$(\Delta, x : \rho(A) \vdash \rho(B)) \in [\Gamma, x : A \vdash s_2]_{(\xi;x:=C),(\rho;x:=x),\Delta} = SN^{\Delta}_{s_2}.$$

Then

$$(\Delta \vdash \rho(\Pi x : A.B)) = (\Delta \vdash \Pi x : \rho(A).\rho(B)) \in SN^{\Delta}_{s_3} = [\Gamma \vdash s_3]_{\xi,\rho,\Delta}.$$

Part 2 follows from the auxiliary induction hypothesis.

(Ind-t) We have

$$\frac{\Gamma, X : *^t \vdash C_n(X) : *^t}{\Gamma \vdash \mathrm{Ind}(X : *^t)\{\vec{C}\} : *^t}$$

Let $\Delta' = (\Delta, X : *^t)$. Note that $\langle(\xi|_{\Delta'}; X := SN^{\Delta'}_X), (\rho; X := X)\rangle$ is an object valuation which satisfies $(\Gamma, X : *^t)$ at Δ'. By the auxiliary induction hypothesis

$$(\Delta' \vdash \rho(C_n(X))) \in [\Gamma, X : *^t \vdash *^t]_{(\xi;X:=SN^{\Delta}_X),(\rho;X:=X),\Delta'} = SN^{\Delta'}_{*^t}.$$

Then

$$(\Delta \vdash \rho(I)) \in [\Gamma \vdash *^t]_{\xi,\rho,\Delta} = SN^{\Delta}_{*^t}$$

as expected.

To prove Part 2 note that

$$[\Gamma \vdash I]_{\xi,\rho,\Delta} = \mathit{lfp}(F_{\Gamma,I,\xi,\rho,\Delta})$$

and

$$[\Gamma \vdash I]_{\xi',\rho,\Delta} = \mathit{lfp}(F_{\Gamma,I,\xi',\rho,\Delta}).$$

The conclusion follows from the auxiliary induction hypothesis and Lemma 88.

(Intro-t) We prove Part 1. We have

$$\Gamma \vdash \mathrm{Constr}(n, I) : C_n(I).$$

We want to prove

$$(\Delta \vdash \rho(\mathrm{Constr}(n, I))) \in [\Gamma \vdash C_n(I)]_{\xi,\rho,\Delta}.$$

Let us assume that $C_n(X) = \Pi \vec{x} : \vec{\tau}.X$. Suppose $(\Delta_i, N_i, C_i)_{i=1}^m$ is an adequate sequence of arguments for $(\vec{x} : \vec{\tau})$ at $\langle \xi, \rho \rangle$ in Δ and $\langle \xi_i, \rho_i \rangle$ is a sequence of constructor valuations associated with it. It suffices to show that

$$(\Delta_m \vdash \mathrm{Constr}(n, \rho(I))\vec{N}) \in [\Gamma, \vec{x} : \vec{\tau} \vdash I]_{\xi_m|_{\Delta_m},\rho_m,\Delta_m}.$$

Note that

$$\mathrm{Constr}(n, \rho(I))\vec{N} \to_k^* \mathrm{Constr}(n, \rho(I))\vec{N}$$

and by the definition of adequate sequence of arguments

$$(\Delta_i \vdash N_i) \in [\Gamma, \vec{x} : \vec{\tau}[I/X] \vdash \tau_i[I/X]]_{\xi_i,\rho_i,\Delta_i}.$$

Moreover

$$\begin{aligned}
&[\Gamma, \vec{x} : \vec{\tau}[I/X] \vdash \tau_i[I/X]]_{\xi_i|_{\Delta_i},\rho_i,\Delta_i} \\
&\quad = [\Gamma, X : *^t, \vec{x} : \vec{\tau} \vdash \tau_i]_{(\xi_i|_{\Delta_i}; X := [\Gamma \vdash I]_{\xi_i|_{\Delta_i},\rho,\Delta_i}),(\rho_i; X := \rho(I)),\Delta_i} && \text{(by Lemma 54)} \\
&\quad = \mathit{Interp}(\Gamma, X : *^t, \vec{x} : \vec{\tau} \vdash \tau_i)_{\xi_i,\rho_i,\Delta_i,X,[\Gamma \vdash I]_{\xi_i|_{\Delta_i},\rho,\Delta_i}} && \text{(by Lemma 84).}
\end{aligned}$$

Thus

$$(\Delta_i \vdash N_i) \in \mathit{Interp}(\Gamma, X : *^t, \vec{x_j} : \vec{\tau_j} \vdash \tau_i)_{\xi_i,\rho_i,\Delta_i,X,[\Gamma \vdash I]_{\xi_i|_{\Delta_i},\rho,\Delta_i}}.$$

The by definition of $[\Gamma \vdash I]_{\xi,\rho,\Delta}$ we get that

$$(\Delta_m \vdash \mathrm{Constr}(n, \rho(I))\vec{N}) \in [\Gamma \vdash I]_{\xi_m|_{\Delta_m},\rho_m,\Delta_m}.$$

Part 2 follows from the auxiliary induction hypothesis.

(Elim-t) We prove Part 1. We have

$$\frac{\Gamma \vdash c : I \qquad \Gamma \vdash Q : I \to s \qquad \Gamma \vdash f_i : \triangle\{C_i(I), Q, \mathrm{Constr}(i, I)\}}{\Gamma \vdash \mathrm{Elim}(I, Q, c)\{\vec{f}\} : Qc}$$

By the auxiliary induction hypothesis

$$\begin{aligned}
&(\Delta \vdash \rho(c)) \in [\Gamma \vdash I]_{\xi,\rho,\Delta}, \\
&(\Delta \vdash \rho(Q)) \in [\Gamma \vdash I \to s]_{\xi,\rho,\Delta}, \\
&(\Delta \vdash \rho(f_i)) \in [\Gamma \vdash \triangle\{C_i(I), Q, \mathrm{Constr}(i, I)\}]_{\xi,\rho,\Delta}.
\end{aligned}$$

Note that $[\Gamma \vdash I]_{\xi,\rho,\Delta} = [\Gamma \vdash I]_{(\xi;x:=[\Gamma\vdash c]_{\xi,\rho,\Delta}),(\rho;x:=\rho(c)),\Delta}$. Since

$$(\Delta \vdash \rho(c)) \in [\Gamma \vdash I]_{\xi,\rho,\Delta}$$

then

$$(\Delta \vdash \rho(c)) \in [\Gamma \vdash I]_{(\xi;x:=[\Gamma\vdash c]_{\xi,\rho,\Delta}),(\rho;x:=\rho(c)),\Delta}.$$

There exists α such that

$$(\Delta \vdash \rho(c)) \in F^{\alpha}_{\Gamma,I,(\xi;x:=[\Gamma\vdash c]_{\xi,\rho,\Delta}),(\rho;x:=\rho(c)),\Delta}(\bigcap SAT^{\Delta}_{(\rho;x:=\rho(c))(I)}).$$

We proceed by induction on α:

- $\alpha = 0$. Then $(\Delta \vdash \rho(c)) \in \bigcap SAT^{\Delta}_{(\rho;x:=\rho(c))(I)}$. But

  $$(\rho; x := \rho(c))(I) = \rho(I) \text{ and thus } SAT^{\Delta}_{(\rho;x:=\rho(c))(I)} = SAT^{\Delta}_{\rho(I)}.$$

  By Lemma 35 there exists m such that $\rho(c) \to^*_k m$ and $(\Delta \vdash m) \in B^{\Delta}_{\rho(I)}$. Then

  $$\text{Elim}(\rho(I), \rho(Q), \rho(c))\{\rho(\vec{f})\} \to^*_k \text{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\}$$

  and

  $$(\Delta \vdash \text{Elim}(\rho(I), \rho(Q), m)\{\rho(\vec{f})\}) \in B^{\Delta}_{\rho(Q)m} = B^{\Delta}_{\rho(Q)\rho(c)} \subseteq [\Gamma \vdash Qc]_{\xi,\rho,\Delta}.$$

  By Corollary 38 the sequent $(\Delta \vdash \text{Elim}(\rho(I), \rho(Q), \rho(c))\{\rho(\vec{f})\})$ is strongly normalizing. Thus

  $$(\Delta \vdash \text{Elim}(\rho(I), \rho(Q), \rho(c))\{\rho(\vec{f})\}) \in [\Gamma \vdash Qc]_{\xi,\rho,\Delta}.$$

- $\alpha = \alpha' + 1$ and the conclusion is true for all $\alpha'' \leq \alpha'$. Assume that

  $$(\Delta \vdash \rho(c)) \in F^{\alpha'}_{\Gamma,I,(\xi;x:=[\Gamma\vdash c]_{\xi,\rho,\Delta}),(\rho;x:=\rho(c)),\Delta}(\bigcap SAT^{\Delta}_{(\rho;x:=\rho(c))(I)}).$$

  By Lemma 96

  $$(\Delta \vdash \text{Elim}(\rho(I), \rho(Q), \rho(c))\{\rho(\vec{f})\}) \in [\Gamma, x : I \vdash Qx]_{(\xi;x:=[\Gamma\vdash c]_{\xi,\rho,\Delta}),(\rho;x:=\rho(c)),\Delta}.$$

  But by Lemma 84

  $$[\Gamma, x : I \vdash Qx]_{(\xi;x:=[\Gamma\vdash c]_{\xi,\rho,\Delta}),(\rho;x:=\rho(c)),\Delta} = [\Gamma \vdash Qc]_{\xi,\rho,\Delta}.$$

  Thus we have the conclusion.

- α is a limit ordinal and the conclusion is true for all $\alpha' < \alpha$. If

  $$(\Delta \vdash \rho(c)) \in F^{\alpha}_{\Gamma,I,(\xi;x:=[\Gamma\vdash c]_{\xi,\rho,\Delta}),(\rho;x:=\rho(c)),\Delta}(\bigcap SAT^{\Delta}_{(\rho;x:=\rho(c))(I)})$$

  then there exists $\alpha' < \alpha$ such that

  $$(\Delta \vdash \rho(c)) \in F^{\alpha'}_{\Gamma,I,(\xi;x:=[\Gamma\vdash c]_{\xi,\rho,\Delta}),(\rho;x:=\rho(c)),\Delta}(\bigcap SAT^{\Delta}_{(\rho;x:=\rho(c))(I)}).$$

  The conclusion follows from the induction hypothesis for α'.

We prove Part 2. Note that

$$[\Gamma \vdash \mathrm{Elim}(I, Q, m)\{\vec{f}\}]_{\xi,\rho,\Delta} = G_{\Gamma,I,Q,\vec{f},\xi,\rho,\Delta}([\Gamma \vdash I]_{\xi,\rho,\Delta})(\Delta \vdash \rho(m), [\Gamma \vdash m]_{\xi,\rho,\Delta})$$

and

$$[\Gamma \vdash \mathrm{Elim}(I, Q, m)\{\vec{f}\}]_{\xi',\rho,\Delta} = G_{\Gamma,I,Q,\vec{f},\xi',\rho,\Delta}([\Gamma \vdash I]_{\xi',\rho,\Delta})(\Delta \vdash \rho(m), [\Gamma \vdash m]_{\xi',\rho,\Delta}).$$

By the auxiliary induction hypothesis

$$[\Gamma \vdash I]_{\xi,\rho,\Delta} \simeq_{\Gamma,\xi,\xi',\rho,\Delta} [\Gamma \vdash I]_{\xi',\rho,\Delta}$$

and thus

$$[\Gamma \vdash I]_{\xi,\rho,\Delta} = [\Gamma \vdash I]_{\xi',\rho,\Delta}.$$

The conclusion follows from Lemma 91 and the auxiliary induction hypothesis.

(Ind-p) Then $M = \mathrm{Ind}(X : A)\{\vec{C}\}$ where $A = \Pi\vec{x} : \vec{\tau}.*^p$. We prove Part 1. We have

$$\frac{\Gamma \vdash A : s \qquad \Gamma, X : A \vdash C_i(X) : *^p}{\Gamma \vdash \mathrm{Ind}(X : A)\{\vec{C}\} : A}$$

Recall that by the definition of interpretation for the product the interpretation $[\Gamma \vdash A]_{\xi,\rho,\Delta}$ is the set of simple sequents $(\Delta' \vdash u)$ of type $\rho(A)$ such that, for every adequate sequence of arguments $(\Sigma_i, a_i, U_i)_{i=1}^n$ for $(\vec{x} : \vec{\tau})$ at $\langle \xi, \rho \rangle$ in Δ, the term $u\vec{a}$ is strongly normalizing. Let $\Delta' = (\Delta, X : A)$. Note that $\langle (\xi|_{\Delta'}; X := SN_X^{\Delta'}), (\rho; X := X) \rangle$ is an object valuation which satisfies $(\Gamma, X : A)$ at Δ'. By the auxiliary induction hypothesis

$$(\Delta' \vdash \rho(C_i(X))) \in [\Gamma, X : A \vdash *^p]_{(\xi; X:=SN_X^{\Delta'}),(\rho;X:=X),\Delta'} = SN_{*^p}^{\Delta'}.$$

Thus $\rho(\mathrm{Ind}(X : A)\{\vec{C}\})$ is strongly normalizing. In an adequate sequence of arguments $(\Sigma_i, a_i, U_i)_{i=1}^n$ the terms $\vec{a}$ are strongly normalizing. Then $\rho(\mathrm{Ind}(X : A)\{\vec{C}\})\vec{a}$ is strongly normalizing as it is not a redex and every subterm of it is strongly normalizing. Hence indeed

$$(\Delta \vdash \rho(\mathrm{Ind}(X : A)\{\vec{C}\})) \in [\Gamma \vdash A]_{\xi,\rho,\Delta}.$$

To prove Part 2 note that

$$[\Gamma \vdash M]_{\xi,\rho,\Delta} = \mathit{lfp}(H_{\Gamma,M,\xi,\rho,\Delta})$$

and

$$[\Gamma \vdash M]_{\xi',\rho,\Delta} = \mathit{lfp}(H_{\Gamma,M,\xi',\rho,\Delta}).$$

The conclusion follows from the auxiliary induction hypothesis and Lemma 89.

(Intro-p) Then $M = \mathrm{Constr}(j, I)$. We prove Part 1. We have

$$\frac{\Gamma \vdash \mathrm{Ind}(X : A)\{\vec{C}\} : A}{\Gamma \vdash \mathrm{Constr}(j, I) : C_j(I)}$$

We want to prove that $(\Delta \vdash \rho(\mathrm{Constr}(j, I))) \in [\Gamma \vdash C_j(I)]_{\xi,\rho,\Delta}$. Suppose $C_j(I) = \Pi\vec{x} : \vec{T}.I\vec{t}$. By the definition of interpretation for the product $[\Gamma \vdash C_j(I)]_{\xi,\rho,\Delta}$ it is enough to prove that for every adequate sequence of arguments $(\Delta_i, N_i, C_i)_{i=0}^{n-1}$ for $(\vec{x} : \vec{T})$ at $\langle \xi, \rho \rangle$ in Δ, if $\langle \xi_i, \rho_i \rangle$ is a constructor valuation associated with it, then we have

$$(\Delta_n \vdash \mathrm{Constr}(j, \rho(I))\vec{N}) \in [\Gamma, \vec{x} : \vec{T} \vdash I\vec{t}]_{\xi_n,\rho_n,\Delta_n}.$$

By Lemma 97 for each recursive argument N_m, if $T_m = \Pi\vec{y} : \vec{T'}.I\vec{t'}$ and there exists an ordinal number α_m such that for each adequate sequence $(\Delta'_l, a_l, C_l)_{l=0}^{n_m}$, of arguments for $(\vec{y} : \vec{T'})$ at $\langle \xi_m, \rho_m \rangle$ in Δ', then we have

$$(\Delta'_{n_m} \vdash N_m\vec{a}) \in H^{\alpha_m}_{\Gamma_{n_m},I,\xi_{n_m},\rho_{n_m},\Delta_{n_m}}([\Gamma \vdash I]_{\xi_{n_m},\rho_{n_m},\Delta_{n_m}})(\Delta \vdash \rho(\vec{t'}), [\Gamma \vdash \vec{t'}]_{\xi_n,\rho_n,\Delta_n}).$$

The conclusion follows from Lemma 98 for $\alpha = \max\{\alpha_m\} + 1$.

We do not have to prove Part 2 as M is a proof.

(Elim-p) We prove Part 1. We have $I = \mathrm{Ind}(X : A)\{\vec{C}\}$ and $A = \Pi\vec{x} : \vec{T}.*^p$

$$(Elim_{*^p})\ \frac{\Gamma \vdash I\vec{t'} : *^p \qquad \Gamma \vdash c : I\vec{t'} \qquad \Gamma \vdash Q : A \qquad \Gamma \vdash f_i : \triangle\{C_i(I), Q\}}{\Gamma \vdash \mathrm{Elim}(I, Q, \vec{t'}, c)\{\vec{f}\} : Q\vec{u}}$$

For simplicity we consider only the case when every term t_i is large, in other cases the proof is similar. By the induction hypothesis

$$\begin{aligned}(\Delta \vdash \rho(I)\rho(\vec{t'})) &\in [\Gamma \vdash *^p]_{\xi,\rho,\Delta},\\ (\Delta \vdash \rho(c)) &\in [\Gamma \vdash I\vec{t'}]_{\xi,\rho,\Delta},\\ (\Delta \vdash \rho(Q)) &\in [\Gamma \vdash A]_{\xi,\rho,\Delta},\\ (\Delta \vdash \rho(f_i)) &\in [\Gamma \vdash \triangle\{C_i(I), Q\}]_{\xi,\rho,\Delta}.\end{aligned}$$

By the definition of $[\Gamma \vdash I]_{\xi,\rho,\Delta}$

$$[\Gamma \vdash I\vec{t'}]_{\xi,\rho,\Delta} = H_{\Gamma,I,\xi,\rho,\Delta}([\Gamma \vdash I]_{\xi,\rho,\Delta})(\Delta \vdash \rho(t'_i), [\Gamma \vdash t'_i]_{\xi,\rho,\Delta})_{i=1}^{p}.$$

Then there are two cases.

Case 1: $(\Delta \vdash \rho(c)) \in \bigcap SAT^{\Delta}_{\rho(I)}$. Then by Lemma 35, $\rho(c) \to_k^* m \in B^{\Delta}_{\rho(I)}$ and thus

$$\rho(\mathrm{Elim}(I, Q, \vec{t'}, c)\{\vec{f}\}) \to_k^* \mathrm{Elim}(\rho(I), \rho(Q), \vec{t'}, m)\{\rho(\vec{f})\}$$

where

$$(\Delta' \vdash \mathrm{Elim}(\rho(I), \rho(Q), \vec{t'}, m)\{\rho(\vec{f})\}) \in B^{\Delta}_{\rho(Q\vec{t'})} \subseteq [\Gamma \vdash Q\vec{t'}]_{\xi,\rho,\Delta}.$$

Moreover, the terms $\rho(I)$, $\rho(Q)$, $\rho(\vec{t'})$ $\rho(c)$, $\rho(\vec{f})$ are strongly normalizing and thus by Corollary 38 the term $\rho(\mathrm{Elim}(I, Q, \vec{t'}, c)\{\vec{f}\})$ is strongly normalizing, too. As $[\Gamma \vdash Q\vec{t'}]_{\xi,\rho,\Delta}$ is a saturated set we get

$$(\Delta \vdash \rho(\mathrm{Elim}(I, Q, \vec{t'}, c)\{\vec{f}\})) \in [\Gamma \vdash Q\vec{t'}]_{\xi,\rho,\Delta}.$$

Case 2: $(\Delta \vdash \rho(c)) \notin \bigcap SAT^{\Delta}_{\rho(I\vec{t'})}$. Then

$$(\Delta \vdash \rho(c)) \in h_{\Delta,I,Q,\xi,\rho}([\Gamma \vdash I]_{\xi,\rho,\Delta})((\Delta_i, \rho(t'_i)_i, [\Gamma \vdash t'_i]_{\xi,\rho,\Delta_i})_{i=1}^{n}).$$

Note that

$$\begin{aligned}(\Delta \vdash \rho(I)) &\in SN^{\Delta}_{\rho(A)},\\ (\Delta \vdash \rho(Q)) &\in [\Gamma \vdash A]_{\xi,\rho,\Delta},\\ [\Gamma \vdash Q\vec{t'}]_{\xi,\rho,\Delta} &\in V_{\Delta}(\rho(Q\vec{t'})),\\ (\Delta \vdash \rho(f_i)) &\in [\Gamma \vdash \triangle\{C_i(I), Q\}]_{\xi,\rho,\Delta}.\end{aligned}$$

By Lemma 84

$$\begin{aligned}&[\Gamma, q : A, X : A \vdash \triangle\{C_i(X), q\}]_{(\xi;X:=[\Gamma\vdash I]_{\xi,\rho,\Delta};q:=[\Gamma\vdash Q]_{\xi,\rho,\Delta}),(\rho;X:=\rho(I);q:=\rho(Q)),\Delta}\\ &\qquad = [\Gamma \vdash \triangle\{C_i(I), Q\}]_{\xi,\rho,\Delta}.\end{aligned}$$

Thus

$$(\Delta \vdash \rho(\mathrm{Elim}(I, Q, c, \vec{t'})\{\vec{f}\})) \in [\Gamma \vdash Q\vec{t'}]_{\xi,\rho,\Delta}.$$

We do not have to prove Part 2 as M is a proof.

□

Theorem 100. *If $\Gamma \vdash M : T$ then M is strongly normalizing.*

Proof. Let $\Gamma \vdash M : T$. Define $\langle \rho, \xi \rangle$ so that $\rho(x) = x$ and $\xi(x) = Can_{\Gamma}(x)$ if $x : A \in \Gamma$. Then $\langle \xi, \rho \rangle$ is an object valuation which satisfies Γ at Γ. By Lemma 99,

$$(\Gamma \vdash \rho(M)) \in [\Gamma \vdash T]_{\xi,\rho,\Gamma}.$$

But

$$[\Gamma \vdash T]_{\xi,\rho,\Gamma} \subseteq SN^{\Gamma}_{\rho(T)} \text{ and } \rho(M) = M, \rho(T) = T.$$

Thus M is strongly normalizing. □

Chapter 6

Conclusions and further work

6.1. Conclusions

We have defined a type system with inductive types. The basis of our theory is a Pure Type System which has a rule of the form (s_1, s_2, s_3) where $s_2 \neq s_3$. There are few examples of systems like that in the literature. As noted by van Benthem Jutting [53] several members of the Automath family, see [43], can be described as PTSs with such rules. One is the system λAUT-68

$$\begin{aligned}
&\mathcal{S} = *, \Box, \triangle\\
&\mathcal{A} = * : \Box\\
&\mathcal{R} = (*, *, *), (*, \Box, \triangle), (\Box, *, \triangle), (\Box, \Box, \triangle), (*, \triangle, \triangle), (\Box, \triangle, \triangle).
\end{aligned}$$

The other one is λAUT-QE

$$\begin{aligned}
&\mathcal{S} = *, \Box, \triangle\\
&\mathcal{A} = * : \Box\\
&\mathcal{R} = (*, *, *), (*, \Box, \Box), (\Box, *, \triangle), (\Box, \Box, \triangle), (*, \triangle, \triangle), (\Box, \triangle, \triangle).
\end{aligned}$$

Constructions allowed in LNTT are function space, implication, universal quantification, dependent types and formula polymorhism. The rule $(*^t, \Box^p, *^t)$ causes that powersets become ordinary types, which is unusual in type theories. We can reason about subsets in the same way we reason about other objects. LNTT defines the logic of the theory. Due to the presence of the formula polymorphism the system is powerful enough to define all logical connectives:

$$\begin{aligned}
\bot &= \forall P : *^p.P,\\
\neg\alpha &= \alpha \to \bot,\\
\alpha \wedge \beta &= \forall P : *^p.((\alpha \to \beta \to P) \to P),\\
\alpha \vee \beta &= \forall P : *^p.((\alpha \to P) \to (\beta \to P) \to P),\\
\alpha \leftrightarrow \beta &= (\alpha \to \beta) \wedge (\beta \to \alpha),\\
\exists x : \tau.\varphi(x) &= \forall P : *^p((\forall x : \tau.\varphi(x) \to P) \to P).
\end{aligned}$$

We extend LNTT with inductive types, which are syntactically similar to inductive types in the Calculus of Inductive Constructions [55]. One difference is that in LNTT with inductive types we cannot define types via strong elimination. However, we can still define many functions for which in other type systems one needs strong elimination. An example is the union of the list of sets. Suppose we have an inductive type For example we may define the union of a list of sets. For a fixed type τ we define a list of subsets of τ as follows:

$$List(\tau \to *^p) = \mathrm{Ind}(X : *^t)\{X \mid (\tau \to *^p) \to X \to X\}.$$

We may define the function $union : List(\tau \to *^p) \to *^p$:

$$union\ l = \mathrm{Elim}(List(\tau \to *^t), \lambda l : List(\tau \to *^t).\tau \to *^t, l)$$
$$\{\lambda x : \tau.\bot \mid \lambda h : \tau \to *^p \lambda r : \tau \to *^p \lambda x : \tau.hx \vee rx\}.$$

The equivalent type in CIC is

$$List(\tau \to *) = \mathrm{Ind}(X : *)\{X \mid (\tau \to *) \to X \to X\}.$$

and it is a large inductive type. One would need the strong elimination over large inductive type to define the union function. But this elimination is forbidden.

There are also types and functions which can be defined in CIC but cannot be defined in LNTT with inductive types. CIC has type polymorphism while LNTT does not. We can define heterogeneous lists in CIC

$$PolyList = \mathrm{Ind}(X : *)\{X \mid \Pi T : *(T \to X \to X)\}$$

and we cannot have similar lists in LNTT with inductive types.

We conclude that LNTT with inductive types is incomparable to CIC. There are things which can be done in LNTT with inductive types and cannot be done in CIC and the other way round. Moreover, even without strong elimination scheme one can define functions which require the use of strong elimination in other type systems.

6.2. Further work

LNTT defines the logic of the system. Inductive types allow to define some basic data types. Our theory is by no means complete. There are many features which can be added to the system.

Equality. Equality is an important notion in type theory. There are many different equalities: conversion, Leibniz equality, equality as an inductive predicate, extensional equality for functions. These notions are not equivalent. It remains to be seen which are appropriate for our naive type theory.

Partial functions. In ordinary mathematics we often use partial functions. However, the basic notion in type theory is a complete function and the basic type constructor is a space of complete functions. Thus partial functions have to encoded. Two natural encoding methods are: partial function as a subtype and partial functions with the help of appropriate predicate in the domain. It is not clear which of these approaches would be more suitable in our system. Apart from partial functions we should define equality and the natural extension ordering.

Quotient types. We would like to extend our system with quotients. For a given equivalence relation, we want the quotient to be a type. There is no satisfactory solution in the literature. Some propositions can be found in Barthe [6], Barthe and Geuvers [7], and

Courtieu [18]. Barthe's approach seems to be the most general and at the same time the most natural. It will be our starting point in further work on this issue.

Subtypes. Subtypes enable to create a type which consists of only some objects of a base type. This is a very useful feature and it is often used in elementary mathematics. For example integers can be seen as a subtype of the type of reals. Objects of the subtype keep their relation with the base type. They can be seen as having two types: the base type and the subtype. Subtype inherits equality from the base type. It is not clear how to add subtypes to the type system.

6.3. Luo and Goguen's UTT

It is conjectured that the strong normalization of Less Naive Type Theory with inductive types can be derived from the similar result for Luo and Goguen's UTT [29, 35, 36]. However, there are some important differences between the two systems which suggest that the translation might be difficult. UTT is predicative while our system is impredicative. As a system formulated in Martin-Löf type theory, UTT is a system with judgemental equality while our system uses conversion. In [29] there is a auxiliary notion of reduction but it is not part of the formal system. Equality in UTT is beta-, eta- and iota-equality, while our system only uses beta- and iota-equality. In UTT there are no inductive predicates so one would have to use an impredicative encoding to express them. In spite of those differences we are going to examine the possibility of the reduction.

Bibliography

[1] The Coq proof assistant. `http://coq.inria.fr`.

[2] P. Aczel and M. Rathjen. Notes on constructive set theory. Technical Report No. 40, Mittag-Leffler, 2000/2001.

[3] R. Adams and Z. Luo. Weyl's predicative classical mathematics as a logic-enriched type theory. In T. Altenkirch and C. McBride, editors, *Types for Proofs and Programs*, volume 4502 of *Lecture Notes in Computer Science*, pages 1–17. Springer-Verlag, 2007.

[4] P.B. Andrews. *An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof*, volume 27 of *Applied Logic Series*. Kluwer, 2002.

[5] H.P. Barendregt. Lambda calculi with types. In S. Abramsky, D.M. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume II, pages 117–309. Oxford University Press, 1992.

[6] G. Barthe. Extensions of pure type systems. In M. Dezani-Ciancaglini and G. Plotkin, editors, *Typed Lambda Calculi and Applications*, volume 902 of *Lecture Notes in Computer Science*, pages 16–31. Springer-Verlag, 1995.

[7] G. Barthe and H. Geuvers. Congruence types. In H. Kleine Buening, editor, *Proceedings of CSL'95*, volume 1092 of *Lecture Notes in Computer Science*, pages 36–51. Springer-Verlag, 1995.

[8] S. Berardi. *Type dependence and constructive mathematics*. PhD thesis, University of Torino, 1990.

[9] A. Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5(2):56–68, 1940.

[10] R.L Constable. Naive computational type theory. In H. Schwichtenberg and R. Steinbruggen, editors, *Proof and System-Reliability*, pages 213–259. Kluwer Academic Press, 2002.

[11] T. Coquand. An analysis of Girard's paradox. In *Symposium on Logic in Computer Science*, pages 227–236. IEEE Computer Society Press, 1986.

[12] T. Coquand. Metamathematical investigation of a calculus of constructions. In P. Odifreddi, editor, *Logic in Computer Science*, pages 91–122. Academic Press, 1990.

[13] T. Coquand and J. Gallier. A proof of strong normalization for the theory of constructions using a Kripke-like interpretation. In *In Workshop on Logical Frameworks–Preliminary Proceedings*, 1990.

[14] T. Coquand and H. Herbelin. A-translation and looping combinators in Pure Type Systems. *Journal of Functional Programming*, 4:77–88, 1994.

[15] T. Coquand and G. Huet. Constructions: A higher order proof system for mechanizing mathematics. In B. Buchberger, editor, *European Conference on Computer Algebra*, volume 203 of *Lecture Notes in Computer Science*, pages 151–184. Springer, 1985.

[16] T. Coquand and G. Huet. The calculus of constructions. *Information and Computation*, 76(2/3):95–120, 1988.

[17] T. Coquand and C. Paulin-Mohring. Inductively defined types. In P. Martin-Löf and G. Mints, editors, *Proceedings of Colog'88*, volume 417 of *Lecture Notes in Computer Science*. Springer-Verlag, 1990.

[18] Pierre Courtieu. Normalized types. In L. Fribourg, editor, *Proceedings of CSL2001*, volume 2142 of *Lecture Notes in Computer Science*, 2001.

[19] W.M. Farmer. A partial functions version of Church's simple theory of types. *Journal of Symbolic Logic*, 55(3):1269–1291, 1990.

[20] W.M. Farmer. A simple type theory with partial functions and subtypes. *Annals of Pure and Applied Logic*, 64:211–240, 1993.

[21] W.M. Farmer. A basic extended simple type theory. Technical Report 14, McMaster University, 2003.

[22] W.M. Farmer. The seven virtues of simple type theory. Technical Report 18, McMaster University, 2003.

[23] H. Geuvers. A short and flexible proof of strong normalization for the Calculus of Constructions. *Journal of Functional Programming*, 1(2):155–189, 1991.

[24] H. Geuvers. Inconsistency of classical logic in type theory . `http://www.cs.ru.nl/~herman/PUBS/newnote.ps.gz`, 2001.

[25] H. Geuvers. Private communication. 2006.

[26] H. Geuvers and M.-J. Nederhof. Modular proof of strong normalization for the Calculus of Constructions. *Journal of Functional Programming*, 1(2):155–189, 1991.

[27] C.E. Giménez. *Un Calcul de Constructions Infinies et son application a la verification de systemes communicants*. PhD thesis, Ecole Normale Supérieure de Lyon, 1996.

[28] J. Y. Girard. Interprétation fonctionelle et élimination des coupures dans l'aritmétique d'ordre supérieur. Thèse d'Etat, Université Paris 7, 1972.

[29] H. Goguen. *A Typed Operational Semantics for Type Theory*. PhD thesis, University of Edinburgh, 1994.

[30] A.J.C. Hurkens. A simplification of Girard's paradox. In M. Dezani-Ciancaglini and G. Plotkin, editors, *Typed Lambda Calculi and Applications*, volume 902 of *Lecture Notes in Computer Science*, pages 266–278. Springer-Verlag, 1995.

[31] R.B. Jensen. On the consistency of a slight(?) modification of Quine's NF. *Synthese*, 19:250–263, 1969.

[32] F. Kamareddine, T. Laan, and R. Nederpelt. Types in logic and mathematics before 1940. *Bulletin of Symbolic Logic*, 8(2):185–245, 2002.

[33] F. Kamareddine, T. Laan, and R. Nederpelt. *A Modern Perspective on Type Theory From its Origins Until Today*, volume 29 of *Applied Logic Series.* Kluwer Academic Publishers, 2004.

[34] A. Kozubek and P. Urzyczyn. In the search of a naive type theory. In M. Miculan, I. Scagnetto, and F. Honsell, editors, *Types for Proofs and Programs 2007*, volume 4941 of *Lecture Notes in Computer Science*, pages 110–124. Springer-Verlag, 2008.

[35] Z. Luo. *An Extended Calculus of Constructions.* PhD thesis, University of Edinburgh, 1990.

[36] Z. Luo. *Computation and reasoning: a type theory for computer science.* Oxford University Press, 1994.

[37] Z. Luo. A type-theoretic framework for formal reasoning with different logical foundations. In *Proc. 11th Annual Asian Computing Science Conference*, volume 4435 of *Lecture Notes in Computer Science*, pages 214–222. Springer-Verlag, 2006.

[38] M.E. Maietti. Quotients over Minimal Type Theory. In B. Cooper, B. Löwe, and A. Sorbi, editors, *Computation and Logic in the Real World - Third Conference of Computability in Europe, CiE 2007*, volume 4497 of *Lecture Notes in Computer Science*, pages 517–531. Springer-Verlag, 2007.

[39] M.E. Maietti. A minimalist two-level foundation for constructive mathematics. *Annals of Pure and Applied Logic*, 160(3):319–354, 2009.

[40] M.E. Maietti and G. Sambin. Toward a minimalist foundation for constructive mathematics. In L. Crosilla and P. Schuster, editors, *From Sets and Types to Topology and Analysis: Practicable Foundations for Constructive Mathematics.* Oxford University Press, 2005.

[41] P. Martin-Löf. A theory of types. Technical Report 71-3, University of Stockholm, 1971.

[42] A. Miquel. Lamda-Z: Zermelo's set theory as a PTS with 4 sorts. In J.-C. Filliâtre, C. Paulin-Mohring, and B. Werner, editors, *Types for Proofs and Programs 2004*, volume 3839 of *Lecture Notes in Computer Science*, pages 232–251. Springer, 2004.

[43] R.P. Nederpelt, H. Geuvers, and R.C. de Vrijer, editors. *Selected papers on Automath.* Studies in logic and the foundations of mathematics. Elsevier, 1994.

[44] B. Nordström, K. Petersson, and J. Smith. *Programming in Martin-Löf's Type Theory.* Clarendon Press, Oxford, 1990.

[45] B. Nordström, K. Petersson, and J. Smith. Martin-Löf's Type Theory. In S. Abramsky, Dov M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of logic in computer science*, volume 5. Oxford University Press, Oxford, UK, 2000.

[46] C. Paulin-Mohring. Inductive definitions in the system Coq - rules and properties. In Bezem M. and Groote J. F., editors, *Typed Lambda Calculi and Applications*, volume 664 of *Lecture Notes in Computer Science*, pages 328–345. Springer-Verlag, 1993.

[47] F. Pfenning and C. Paulin-Mohring. Inductively defined types in the calculus of constructions. In M.G. Main, A. Melton, Mislove M.W., and Schmidt D.A., editors, *Mathematical Foundations of Programming Semantics*, pages 209–226. Springer-Verlag, 1990.

[48] W.V. Quine. New foundations for mathematical logic. *American Mathematical Monthly*, 44:70–80, 1937.

[49] M.H. Sørensen and P. Urzyczyn. *Lectures on the Curry-Howard Isomorphism.* Elsevier, 2006.

[50] M. Takahashi. Parallel reductions in lambda-calculus. *Information and Computation*, 118:120–127, 1995.

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

[52] J. Terlouw. Een nadere bewijstheoretische analyse van GSTT's. Manuscript (in Dutch), 1989.

[53] L. S. van Benthem Jutting. Typing in pure type systems. *Information and Computation*, 105:30–41, 1993.

[54] D. Walukiewicz. *Termination of Rewriting in the Calculus of Constructions.* PhD thesis, University of Warsaw and Université de Paris Sud, 2002.

[55] B. Werner. *Une Théorie des Constructions Inductives.* PhD thesis, Université Paris 7, 1994.

[56] H. Weyl. *The Continuum.* Dover, 1994.