SJÄLVSTÄNDIGA ARBETEN I MATEMATIK

MATEMATISKA INSTITUTIONEN, STOCKHOLMS UNIVERSITET

Investigations into a model of type theory based on the concept of basic pair

av

Simone Tonelli

2013 - No 8

MATEMATISKA INSTITUTIONEN, STOCKHOLMS UNIVERSITET, 106 91 STOCKHOLM

Investigations into a model of type theory based on the concept of basic pair

Simone Tonelli

Självständigt arbete i matematik 30 högskolepoäng, Avancerad nivå

Handledare: Erik Palmgren och Giovanni Sambin

2013

Master's Thesis

Investigations into a Model of Type Theory Based on the Concept of Basic Pair

Simone Tonelli

Supervisor:
Giovanni Sambin

Supervisor:
Erik Palmgren

Outside Examiner:
Per Martin-Löf

Università degli studi di Padova

Stockholm University

May, 2013

Abstract

The principal aim of this thesis is to explain a possible model of Per Martin-Löf's type theory based on the concept of Giovanni Sambin's basic pair. This means to extend the concept of "set" in the easiest and most natural way: transforming it in a couple of sets and an arbitrary relations set between them, i.e. a basic pair. This reasoning will be applied to all the judgment forms and will give us an interpretation of Martin-Löf's type theory. Our purpose will be to find a model which satisfies this interpretation, and we will look for it following two different approaches. The first one is meant to remain inside the standard type theory constructing an internal model; the second one, arisen from some impasses reached in the development of the first attempt, is aimed at adding new type constructors at the standard theory, extending it and allowing us to create an external model. These new types, that we have denoted here with a star, have to be seen like an arbitrary relations set between two set of the same type without star. This extended theory will give us all the results needed in a natural way, and might be useful in different interpretations for further research.

Contents

Chapter 1

Introduction

The principal aim of this thesis is to investigate into possible interpretation, based on the concept of the basic pair [Sambin, 201], of Martin-Löf's Type Theory ([Martin-Löf, 1984], and [Nordström et al., 1990]).

We will face two different approaches: the first one is meant to remain inside the usual theory of types giving us an internal interpretation; the second one aims at adding new types to the theory, leading us to a more natural but external interpretation.

In this introductory chapter we first discuss the notion of basic pair, without going in depth, and then we explain the motivation of our work, and finally give a survey of the contents of the following chapters.

1.1 Basic Picture

"Basic Picture" is a new framework that gives a structural basis to constructive topology and at the same time it generalizes both *pointfree* and *pointwise* topology, it even deals even with the foundation of mathematics. This theory has been introduced by Giovanni Sambin in [Sambin, 201].

We now report the following part of the foreward section of the book mentioned above, which fits perfectly to our purpose, i.e. sketching a few of the central concepts of the book, and we refer the reader to it for further in-depth analysis. To define a topological space in the usual sense, it suffices in the first place to be given a set X of points and a family $\mathcal{B}$ of subsets called *basic open* sets or *neighbourhoods*. For the resulting structure to be a topological space, it is further required that certain well-known conditions be satisfied by X and $\mathcal{B}$: these are presented in terms of the membership relation $\in_X$ between elements of X and elements of $\mathcal{B}$. The key concept of the Basic Picture, that of *basic pair*, now emerges with compelling simplicity:

to wit, by replacing $\mathcal{B}$ by an *arbitrary* set S and $\in_X$ by an *arbitrary* relation $\Vdash_X$ between X and S. Thus a basic pair is just a binary relation: more precisely, a structure $(X, \Vdash_X, S)$ with $\Vdash_X$ a relation between X and S. If we think of a basic pair $(X, \Vdash_X, S)$ as a "space", the left-hand component X is conceived as representing its domain of points-its "concrete", or "pointed" side–and its right-hand component S its domain of neighbourhoods–its "formal" or "pointfree" side. The symmetry and consequent duality between the concrete and formal sides of basic pairs arising from the Galois connection induced by the relation linking the two sides underlies the whole development of the Basic Picture. In particular the duality is used to reintroduce the topological concepts of open and closed sets. By symmetry these notions are interpretable on both "sides", so giving rise to concrete open and concrete closed sets, as well as formal open and closed sets. Each of these forms a (not necessarily distributive) complete lattice: the lattices of concrete open (closed) and formal open (closed) sets are isomorphic and the lattices of concrete (formal) open and concrete (formal) closed sets are in a natural sense dual to one another. Moreover, the definitions of concrete open (closed), and formal closed (open), involve a quantifier alternation of the form $\exists\forall$ ($\forall\exists$), resulting in a logical symmetry between these notions. In fact, one can see that the topological notions of interior and closure are the result of the dynamics between the two sets, furthermore the definition of interior of a subset can be seen to have the logical form of $\exists\forall$ while closure has the dual form $\forall\exists$. This allows one to discover a clear structure underlying topology: logical duality between open and closed, symmetry between the traditional approach with points (or pointwise) and the pointfree one. Only when adopting classical logic this is "simplified" to the fact that closed is complement of open. So considering only the notion of open means reducing to a half of the picture, that which is enough from a classical perspective. That is why a truly constructive approach must begin everything anew, including the task of finding correct definitions.

Given the intrinsic symmetry of the concept of basic pair, how should a *continuous map* between basic pairs be defined? Again, the definition is suggested by the classical case. It is easily shown that, given two topological spaces $(X, \mathcal{B})$ and $(Y, \mathcal{C})$, a map $f : X \to Y$ is continuous precisely when the diagram of relations

$$\begin{tikzcd} X \arrow[r, "\in_X"] \arrow[d, "f"'] \arrow[dr, phantom, "\circlearrowleft"] & \mathcal{B} \arrow[d, "F"] \\ Y \arrow[r, "\in_Y"'] & \mathcal{C} \end{tikzcd}$$

commutes, where F is the relation defined by $UFV \leftrightarrow f[U] \subseteq V$. In keeping with the spirit of symmetry it is natural then to define a continuous map between basic pairs $(X, \Vdash_X, S)$ and $(Y, \Vdash_Y, T)$ to be a pair of *relations* r, s between X and Y, and S and T, respectively, for which the diagram of relations

$$\begin{tikzcd} X \arrow[r, "\Vdash_X"] \arrow[d, "r"'] \arrow[dr, phantom, "\circlearrowleft"] & S \arrow[d, "F"] \\ Y \arrow[r, "\Vdash_Y"'] & T \end{tikzcd}$$

commutes. Such a pair of relations is called simply a relation-pair. By passing from functions to relations, one can see that also the notion of continuity has a structural characterization, since it is commutativity of a square diagram.

These definitions give rise to a category. Clearly, if **Rel** denotes the category of (sets with) binary relations, then the functor category $\mathbf{Rel}^2$ is the category with basic pairs as objects and relation-pairs as arrows. The category **BP** of basic pairs is defined to be the quotient category obtained from $\mathbf{Rel}^2$ by identifying two arrows $(r1, s1), (r2, s2) : (X, \Vdash_X, S) \to (Y, \Vdash_Y, T)$ when $\Vdash_Y \circ r_1 = \Vdash_Y \circ r_2$ and $s_1 \circ \Vdash_X = s_2 \circ \Vdash_X$. These conditions ensure, in the author's words, that the two arrows "behave equally with respect to the topological structure a relation-pair is meant to preserve".

These, then, are the fundamental ideas underlying the Basic Picture. The author sees the concept of a basic pair as "the simplest extension of the notion of a set": in my view this represents a radical advance on the classical conception of a topological space, which is normally viewed as the simplest extension of the concept of a discrete set to a structure supporting continuity. As has already been mentioned, an essential feature of the definition of a basic pair is its symmetry: the definition of basic pair allows "movement" between the concrete and the formal "sides", and topological concepts are shown to arise from this "movement". While classical topology emphasizes points, and pointless topology neighbourhoods, in the Basic Picture a symmetry is fashioned between point and neighbourhood. It could be said that if point-set topology is taken as Thesis, and pointless topology as Antithesis, then the Basic Picture represents Synthesis.

It must be emphasized that the full richness of the conceptual framework offered by the Basic Picture can *only* be seen by adopting a constructive viewpoint under which both the law of excluded middle and the power set axiom have been discarded. The version of constructivism adopted in the book (and called "minimalist foundation", see [Sambin, 2011]) not only is intuitionistic and predicative, but also has no choice principles. This makes it compatible with an intuitionistic impredicative foundation, as topos theory.

From a philosophical and methodological point of view, Sambin has elsewhere introduced the term *dynamic constructivism* (see [Sambin, 2002]) to identify the approach to the problem of the nature of mathematics on which his work rests.

Dynamic constructivism shares with Brouwer's *intuitionism* the insight that mathematical truth and mathematical objects are not simply given, but must be *constructed*. Indeed, as Sambin has observed, dynamic constructivism may be regarded as essentially intuitionism purged of its mystical and solipsistic elements, in which Brouwer's identification of mathematics with the immediate intuitions of the creative subject is replaced by the acknowledgment of the existence of other individuals. In this way the hermetic world of the subject is opened out into the *umwelt*, the lived, changing world which includes other persons, their perceptions and the interactions both between themselves and the natural environment. The whole mathematical universe exists in this *umwelt*, "in the minds of human beings". In this way mathematics becomes a *cooperative* enterprise. The objectivity of mathematics then amounts essentially to a shared process of reification, through a "democratic, though occasionally turbulent, dynamic process of achieving consensus." Such a process must include "the interaction with other individuals, and with the outcome of *their* mental processes." If intuitionism allows just one subject (and formalism does away with the subject altogether) then dynamic constructivism requires the interaction of many subjects. It is, in a word, *pluralistic*.

1.2 Motivations of our work and further research

Starting from the concept of a basic pair our aim is to build a new model for Martin-Löf's Type Theory. As in a basic pair, our interpretation is based on the simplest extension of the concept of set: a couple of sets and an arbitrary relation between them. So, what we are going to do is to transform each judgment in the usual Type Theory, into a triple of judgments which will follow the idea of the basic pair. For example a judgment of the form:

$$A : set$$

will be transformed into a arbitrary triple of judgments:

$$\begin{cases} A_0 : set \\ A_1 : set \\ A_r : rel(A_0, A_1) \quad \text{or equivalent} \quad A_r(x_0, x_1) : set \quad [x_0 : A_0, x_1 : A_1] \end{cases}$$

After giving an interpretation to all possible judgments, substitution and equality rules, we begin to construct an internal model transforming each type into a triple of preexisting types, for example the type of natural numbers:

$$\mathbb{N} : set$$

when passing to the internal model will become:

$$\begin{cases} \mathbb{N} : set \\ \mathbb{N} : set \\ \mathtt{Id}(\mathbb{N}, x_0, x_1) : set \quad [x_0 : \mathbb{N}, x_1 : \mathbb{N}] \end{cases}$$

where `Id` denotes the usual Identity type.

As we will show later, the internal model leads us to an impasse when we have to deal with the interpretation of natural numbers, identity and disjoint union type. Due to this reason we will abandon this approach in order to work on an external model. Hence, the idea is to extend the usual type theory introducing new types, that we will denote with a star. These new "star" types represent the relation set between two sets of the same type "without the star". For example, in the external model the set of natural numbers is interpreted as the following triple:

$$\begin{cases} \mathbb{N} : set \\ \mathbb{N} : set \\ \mathbb{N}^* : rel(\mathbb{N}, \mathbb{N}) \end{cases}$$

An important observation is that we are not introducing any new judgments but we are just expanding the standard type theory, in a way that we can consider as the following inclusion:

$$\textit{standard type theory} \subset \textit{star type theory}$$

Furthermore this extended theory will gain some new properties, for example a kind of extensionality between functions, as we will show later.

Further researches about this work could be, for example to investigate in depth the relation between the Sambin's Basic picture and the Martin-Löf's type theory since what we have done is just to build a model inspired on a basic pair's concept. What I mean is that now maybe one could investigate the relation which subsists between this extended type theory and the whole category of basic pairs, **BP**.

Another important application of this model can be found in the field of category theory. Indeed the structure of our interpretation could be further seen as a model for a generic Set-valued presheaf. A *Set*-valued presheaf on a category C is a functor $F : C^{op} \to Set$. If we consider for example a category C with just three objects, that we can call $0, 1, r$ and with just unique morphisms $f : 0 \to r$ and $g : 1 \to r$, such that they satisfy the following scheme:

$$0 \xrightarrow[f]{} r \xleftarrow[g]{} 1$$

Then a Set-valued presheaf, F, over this category will be described by this diagram:

$$F(0) \xleftarrow[F(f)]{} F(r) \xrightarrow[F(g)]{} F(1)$$

Immediately this presheaf can be related with our interpretation, indeed it is enough to identify the set $F(r)$ with the set of relations between $F(0)$ and $F(1)$, and the morphisms $F(f)$ and $F(g)$ as its two projections into them. So this shall be seen as an easy and first example of how our model could be even related to category theory. Although this was not the main motivation of our work, some further research into this direction might give some good results.

Moreover, this model has arisen from the necessity to formalize the path topology on types, introduced by Martin-Löf in his talks during the third and the fourth workshops on formal topology ([4WO, 2012]). Indeed starting from infinitely many types $A_0, A_1, \ldots, A_n, \ldots$ and relations between them $P_0 : rel(A_0, A_1), P_1 : rel(A_1, A_2), \ldots, P_n : rel(A_n, A_{n+1}), \ldots$ one can define a formal topology where the open sets consist of all the infinite branches starting from the endpoint of a finite path; where the paths are the infinite sequences of elements $a_0 : A_0, a_1 : A_1, a_2 : A_2, \ldots$, such that $P_n(a_n, a_n + 1)$ for all $n = 0, 1, 2, \ldots$. This model has been named by Martin-Löf, the *spread model* or *interference spread model*, but no papers about it have been published after he introduced them during the third and fourth workshops on formal topology. Obviously, our model is just the simplest example of this spread model since we have just a pair of sets and one relation sets between them. Anyway this could be taken like a starting point in order to construct a complete model for this formal topology.

So, as we can see, a lot of different further research can arise from this model, even if our first intention was just to find a possible relation between the type theory and the basic pairs.

1.3 Notation and survey of contents

About what concerns the notation for the type theory we mainly follow the one used in the book [Nordström et al., 1990] with possibly some slight differences; however, in order to avoid misunderstandings, when we will go to interpret types we often recall their rules.

The totally of our proof trees has been created using the package *bussproof.sty* for LaTeX. When the premises don't fit on a line we will stack them like the following picture:

$$\dfrac{\begin{array}{c} P_1 \\ \vdots \\ P_n \end{array}}{C}$$

Furthermore, in this case, it will often happen that one or more premises don't fit neither on a single line, due to its length or its context, so to solve this problem we will gather all its lines in a bracket, as follows:

$$\dfrac{\begin{array}{c} \begin{cases} P_1 \\ \dots \\ \dots \end{cases} \\ \vdots \\ P_n \end{array}}{\begin{cases} C \\ \dots \end{cases}}$$

About the contents, in the second chapter we will explain all the judgments and rules' interpretations. In the following chapter we will show our attempt to construct the internal model and we will point out the problems encountered. Finally, in the last chapter, we will explain the external model with its new type constructors and their rules, and we will use them to interpret the standard type theory as wanted.

Chapter 2

The interpretation into basic pairs

As we said in the introduction our aim is to build a model for type theory based on the concept of the basic pair.

Basically what we will do is extend in the same way as the basic pair, the concept of set to a triple of elements formed by a couple of sets and a set of relationships between them. This extension of the concept of set will allow us to give an interpretation of the judgment "to be a set", namely a generic judgment of the form:

$$A : set$$

is interpreted into:

$$\begin{cases} A_0 : set \\ A_1 : set \\ P : rel(A_0, A_1) \end{cases}$$

Consequently, all other interpretations of the different forms of judgment of type theory will be constructed in analogous way, as we will show in the next sections.

Furthermore observe that we are not introducing any new form of judgment, but simply we are, in some way, "tripling" any judgment already exists. So we shall not give any new explanation, philosophical or other beliefs, since in interpreting a judgment will keep the same value it had before. For this reason even the logical framework of [Nordström et al., 1990], i.e. the dependent function space and the universe Set, is interpreted following the same pattern.

This idea has been inspired by Martin-Löf mostly during his latest seminary at University of Stockholm [Sem, 2013], where I had the fortune to participate. During these talks he extended the type theory with new types in order to be able to interpret a set as two identical sets and an arbitrary relation between them. Our interpretation is for this reason more general since it is possible for us to have two different sets.

Anyway we are going to explain in the following sections all the different interpretation of the judgment forms, and the validation of the substitution rules.

2.1 Interpretation of type theory's judgment forms

We now show how to give the interpretation of the four different judgments:

1. the judgment A : set, in the empty context is interpreted as three different judgments in the empty context

$$A_0 : set \qquad A_1 : set \qquad P : rel(A_0, A_1)$$

   Where $rel(A_0, A_1)$ denotes the binary relations between A_0 and A_1 in the type theoretic sense. In the higher order notation we may write

$$rel(A_0, A_1) =_{def} (A_0, A_1)set$$

2. The elementhood judgment $a : A$ in the empty context is interpreted in three different judgments in the empty context

$$a_0 : A_0 \qquad a_1 : A_1 \qquad p : P(a_0, a_1)$$

   Here p is regarded as the proof object for a_0 and a_1 being related via P.

3. The judgment B(x): set in context $[x : A]$ is interpreted as the triple judgment:

$$\begin{cases} B_0(x_0) : set \quad [x_0 : A_0] \\ B_1(x_1) : set \quad [x_1 : A_1] \\ Q(x_0, x_1, p)(y_0, y_1) : set \\ \quad [x_0 : A_0, x_1 : A_1, p : P(x_0, x_1), y_0 : B_0(x_0), y_1 : B_1(x_1)] \end{cases}$$

   the last judgment may be expressed in higher order notation,

$$Q(x_0, x_1, p) : rel(B_0(x_0), B_1(x_1)) \quad [x_0 : A_0, x_1 : A_1, p : P(x_0, x_1)]$$

4. The elementhood judgment $b(x) : B(x)$ in context $[x : A]$ is interpreted as three judgments

$$\begin{cases} b_0(x_0) : B_0(x_0) \quad [x_0 : A_0] \\ b_1(x_1) : B_1(x_1) \quad [x_1 : A_1] \\ q(x_0, x_1, p) : Q(x_0, x_1, p)(b_0(x_0), b_1(x_1)) \quad [x_0 : A_0, x_1 : A_1, p : P(x_0, x_1)] \end{cases}$$

5. the equality judgment are interpreted in the expected way: one definitional equality judgment becomes three definitional equality judgments determined by the relevant judgment above.

   For example the judgment of equality between sets in the empty context: Generally speaking considering equality between sets:

$$A = B : set$$

   It is interpreted in the following triple of standard judgments:

$$\begin{cases} A_0 = B_0 : set \\ A_1 = B_1 : set \\ P(x_0, x_1) = Q(x_0, x_1) : set \quad [x_0 : A_0 = B_0, x_1 : A_1 = B_1] \end{cases}$$

   where, obviously, $P(x_0, x_1)$ (and so even $Q(x_0, x_1)$) must be consider, like in the usual type theory, extensional in the sense the whenever $a_0 = b_0 : A_0$ and $a_1 = b_1 : A_1$ then $P(a_0, a_1) = P(b_0, b_1)$.

   In an analogous way we can interpret the equivalence between elements in a set:

$$a = b : A$$

   using the following triple:

$$\begin{cases} a_0 = b_0 : A_0 \\ a_1 = b_1 : A_1 \\ a_r = b_r : P(a_0, a_1) = P(b_0, b_1) \end{cases}$$

   where $P(a_0, a_1) = P(b_0, b_1)$ follows by its extensionality property.

   The interpretations of the remaining equivalence judgments follow the same pattern. So we leave them to the reader.

2.2 Interpretation of judgments with more than one assumption

We may now further generalize judgments to include hypothetical judgments with an arbitrary number n of assumptions. We explain their meaning by induction, that is, assuming we understand the meaning of judgment with $n-1$ assumptions. So assume we know:

- A_1 is a set
- $A_2(x_1)$ is a family of sets over A_1
- $A_3(x_1, x_2)$ is a family of sets with two indexes $x_1 : A_1$ and $x_2 : A_2(x_1)$
- $\ldots$
- $A_n(x_1, \ldots, x_{n-1})$ is a family of sets with $n-1$ indexes $x_1 : A_1, x_2 : A_2(x_1), \ldots, x_{n-1} : A_{n-1}(x_1, \ldots, x_{n-2})$

Then a judgment of the form:

$$A(x_1, \ldots, x_n)\ set \quad [x_1 : A_1, x_2 : A_2(x_1), \ldots, x_n : A_n(x_1, \ldots, x_{n-1})] \tag{2.1}$$

means that $A(a_1, \ldots, a_n)$ is a set whenever $[a_1 : A_1, a_2 : A_2(a_1), \ldots, a_n : A_n(a_1, \ldots, a_{n-1})]$ and that $A(a_1, \ldots, a_n) = A(b_1, \ldots, b_n)$ whenever $[a_1 = b_1 : A_1, a_2 = b_2 : A_2(a_1), \ldots, a_n = b_n : A_n(a_1, \ldots, a_{n-1})]$. We say that $A(x_1, \ldots, x_n)$ is a family of set with n indexes. The n assumptions in a judgment of the form (2.1) constitute what we call the context, which plays a role analogous to the sets of formulae Δ, Γ (extra formulae) appearing in Gentzen sequents. Note also that any initial segment of a context is always a context. Because of the meaning of a hypothetical judgment of the form (2.1), we see that the first two rules of substitution may be extended to the case of n assumptions, and we understand these extensions to be given.

2.2.1 Interpretation

In our interpretation a generic judgment becomes a triple of judgments as shown before, and each of these it will have a different context, so we will show how to obtain an interpretation of a judgment with a context of arbitrary length. In the following section I will use superscript instead of subscript, where necessary, in order to avoid confusion.

- First of all we show how to interpret the the judgment with two assumptions (a context of length 2):

$$A(x_1, x_2) : set \quad [x_1 : A_1, x_2 : A_2(x_1)] \tag{2.2}$$

We can start interpreting the assumptions:

$$x_1 : A_1 \rightsquigarrow \begin{cases} x_1^0 : A_1^0 \\ x_1^1 : A_1^1 \\ x_1^r : A_1^r(x_1^0, x_1^1) \quad [x_1^0 : A_1^0, x_1^1 : A_1^1] \end{cases}$$

and

$$x_2 : A_2(x_1) \quad [x_1 : A_1] \rightsquigarrow \begin{cases} x_2^0 : A_2^0(x_1^0) \quad [x_1^0 : A_1^0] \\ x_2^1 : A_2^1(x_1^1) \quad [x_1^1 : A_1^1] \\ x_2^r : A_2^r(x_1^0, x_1^1, x_1^r)(x_2^0, x_2^1) \\ \quad [x_1^0 : A_1^0, x_1^1 : A_1^1, x_1^r : A_1^r(x_1^0, x_1^1), \\ \quad x_2^0 : A_2^0(x_1^0), x_2^1 : A_2^1(x_1^1)] \end{cases}$$

Then from this assumptions we can create three well-formed contexts:

$$\begin{cases} \Gamma^0 \equiv [x_1^0 : A_1^0, x_2^0 : A_2^0(x_1^0)] \\ \Gamma^1 \equiv [x_1^1 : A_1^1, x_2^1 : A_2^1(x_1^1)] \\ \Gamma^r \equiv [x_1^0 : A_1^0, x_1^1 : A_1^1, x_1^r : A_1^r(x_1^0, x_1^1), \\ \qquad x_2^0 : A_2^0(x_1^0), x_2^1 : A_2^1(x_1^1), x_2^r : A_2^r(x_1^0, x_1^1, x_1^r)(x_2^0, x_2^1)] \end{cases}$$

Finally, with these, we can give an interpretation of (2.2):

$$\begin{cases} A^0(x_1^0, x_2^0) : set \quad [\Gamma^0] \\ A^1(x_1^1, x_2^1) : set \quad [\Gamma^1] \\ A^r(x_1^0, x_1^1, x_1^r, x_2^0, x_2^1, x_2^r)(z^0, z^1) : set \\ \quad [\Gamma^r, z^0 : A^0(x_1^0, x_2^0), z^1 : A^1(x_1^1, x_2^1)] \end{cases}$$

- Now we have to argue in a similar way in order to find an interpretation for a judgment with an arbitrary n number of assumptions:

$$A(x_1, ...x_n) : set \; [x_1 : A_1, x_2 : A_2(x_1), ..., x_n : A_n(x_1, ..., x_{n-1})] \tag{2.3}$$

In the same way, as we have done before, we will get three well-formed contexts:

$$\begin{cases} \Gamma_0 \equiv [x_1^0 : A_1^0, x_2^0 : A_2^0(x_1^0), ..., x_n^0 : A_n^0(x_1^0, x_2^0, ..., x_{n-1}^0)] \\ \Gamma_1 \equiv [x_1^1 : A_1^1, x_2^1 : A_2^1(x_1^1), ..., x_n^1 : A_n^1(x_1^1, x_2^1, ..., x_{n-1}^1))] \\ \Gamma_r \equiv [x_1^0 : A_1^0, x_1^1 : A_1^1, x_1^r : A_1^r(x_1^0, x_1^1), x_2^0 : A_2^0(x_1^0), \\ \quad x_2^1 : A_2^1(x_1^1), x_2^r : A_2^r(x_1^0, x_1^1, x_1^r)(x_2^0, x_2^1), ..., \\ \quad x_n^0 : A_n^0(x_1^0, x_2^0, ..., x_{n-1}^0), x_n^1 : A_n^1(x_1^1, x_2^1, ..., x_{n-1}^1)) \\ \quad x_n^r : A_n^r(x_1^0, x_1^1, x_1^r, x_2^0, x_2^1, x_2^r, ..., x_{n-1}^0, x_{n-1}^1, x_{n-1}^r)(x_n^0, x_n^1)] \end{cases}$$

Hence we can give an interpretation of (2.3):

$$\begin{cases} A_0(x_1^0, ..., x_n^0) : set \quad [\Gamma_0] \\ A_1(x_1^1, ..., x_n^1) : set \quad [\Gamma_1] \\ A_r(x_1^0, x_1^1, x_1^r, ..., x_n^0, x_n^1, x_n^r)(z_0, z_1) : set \\ \quad [\Gamma_r, z_0 : A_0(x_1^0, ..., x_n^0), z_1 : A_1(x_1^1, ..., x_n^1)] \end{cases}$$

2.2.2 Elementhood judgment with more assumptions

Since we have defined what is the interpretation to "be a set" judgment with an arbitrary number of assumptions, we can now try to get the interpretation of a generic elementhood judgment:

$$a(x_1, ..., x_n) : A(x_1, ..., x_n) \quad [\Gamma] \tag{2.4}$$

where Γ is a generic context of length n:

$$[x_1 : A_1, x_2 : A_2(x_1), ..., x_n : A_n(x_1, ..., x_{n-1})]$$

First of all we recall the interpretation of judgment with a context of length one and two, and then we will argue with a generic context of length n:

length 1 : the interpretation is the following:

$$a : A \rightsquigarrow \begin{cases} a_0 : A_0 \\ a_1 : A_1 \\ a_r : A_r(a_0, a_1) \end{cases}$$

length 2 : the interpretation is the following:

$$a(x_1) : A(x_1)\ [x_1 : A_1] \rightsquigarrow \begin{cases} a_0(x_1^0) : A_0(x_1^0) \quad [x_1^0 : A_1^0] \\ a_1(x_1^1) : A_1(x_1^1) \quad [x_1^1 : A_1^1] \\ a_r(x_1^1, x_1^1, x_1^r) : A_r(x_1^0, x_1^1, x_1^r)(a_0(x_1^0), a_1(x_1^1)) \\ \quad [x_1^0 : A_1^0, x_1^1 : A_1^1, x_1^r : A_1^r(x_1^0, x_1^1)] \end{cases}$$

length n : in this case we should create an interpretation on (2.4), and we can do it using the same pattern as in the previous section. Then our intended interpretation will be:

$$\begin{cases} a_0(x_1^0,...,x_n^0) : A_0(x_1^0,...,x_n^0) \quad [\Gamma_0] \\ a_1(x_1^1,...,x_n^1) : A_1(x_1^1,...,x_n^1) \quad [\Gamma_1] \\ a_r(x_1^0,x_1^1,x_1^r,...,x_n^0,x_n^1,x_n^r) : \\ \quad A_r(x_1^0,x_1^1,x_1^r,....,x_n^0,x_n^1,x_n^r)(a_0(x_1^0,...,x_n^0),a_1(x_1^1,...,x_n^1)) \quad [\Gamma_r] \end{cases}$$

where $\Gamma_0, \Gamma_1, \Gamma_r$ denote the same contexts as in the previous section.

2.3 The interpretation of substitution rules

In the usual type theory the meanings of the four judgment forms when they depend on a nonempty context yield four sets of substitution rules. The judgment

$$C(x) : set \quad [x : A]$$

means that $C(a)$ is a set, provided $a : A$, and that $C(a) = C(b)$ whenever $a = b : A$. This explanation immediately gives us the rules:

$$\frac{C(x) : set \quad [x : A] \qquad a : A}{C(a) : set} \qquad \frac{C(x) : set \quad [x : A] \qquad a = b : A}{C(a) = C(b)}$$

The same reasoning yields for the explanation of the other form of judgments.

Anyway these rules for substitution are not sufficient because if we have a judgment

$$C(x, y) : set \quad [x : A, y : B(x)]$$

and want to substitute $a : A$ for x and $b : B(a)$ for y we cannot use the rules given above since they cannot handle the case with simultaneous substitution of several variables. We therefore extend the substitution rules to n simultaneous substitutions. We present only the rule for substitution in equal sets. Substitution in equal sets of n variables:

$$\frac{\begin{array}{c} B(x_1, \ldots, x_n) = C(x_1, \ldots, x_n) \quad [x_1 : A_1, \ldots, x_n : A_n(x_1, \ldots, x_{n-1})] \\ a_1 : A_1 \\ \vdots \\ a_n : A_n(a_1, \ldots, a_{n-1}) \end{array}}{B(a_1, \ldots, a_n) = C(a_1, \ldots, a_n)}$$

The rule is justified from the meaning of a hypothetical judgment with several assumptions.

Another way to achieve the same effect is to allow substitution in the middle of a context. For example if we have a judgment

$$C(x, y) : set \quad [x : A, y : B(x)]$$

we could first substitute $a : A$ for x obtaining the judgment

$$C(a, y) : set \quad [y : B(a)]$$

then substitute $b : B(a)$ for y. When using type theory to do formal proofs, it is convenient to have substitution rules of this form.

2.3.1 Interpretation

Now in our intended interpretation each judgment becomes a triple of judgments, and so each substitution rule will become a triple of substitution rules. For example the first rule that we showed in the previous section will become the following three:

$$\frac{C_0(x_0) : set \quad [x_0 : A_0] \qquad a_0 : A_0}{C_0(a_0) : set} \qquad \frac{C_1(x_1) : set \quad [x_1 : A_1] \qquad a_1 : A_1}{C_1(a_1) : set}$$

and the last one on the set of relations will be:

$$\frac{\begin{array}{c} C_r(x_0, x_1, x_r) : rel(C_0(x_0), C_1(x_1)) \quad [x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, x_1)] \\ a_0 : A_0 \\ a_1 : A_1 \\ a_r : A_r(a_0, a_1) \end{array}}{C_r(a_0, a_1, a_r) : rel(C_0(a_0), C_1(a_1))}$$

which is validated by the rule of multiple substitution explained before.

Furthermore the same reasoning is valid for each substitution rule, included the one of multiple substitution, so we are not going to show all the interpretations, since they follow the same pattern.

Chapter 3

The internal model

We present now our first attempt to create a model based on the interpretation explained in the previous chapter. This approach consists of a creation of an internal model of the standard type theory, presented in [Martin-Löf, 1984] with intentional equality type. Generally speaking what we do, is for each type and each of its constants (canonical or not), we will try to interpret it as a triple of judgments explained before. At the same time this interpretation (triple of judgments) must, however, validate all the rules that characterize that particular type: formation, introduction, elimination and equality. The validation of the rules follows a common pattern: starting from the interpretations of all the premises we should be able to construct an interpretation for the conclusion. Generally speaking this reasoning leads to a tripling, somehow of, the single rule since we need to find three judgments in which to interpret its conclusion. The tripling happens often in a specific manner: substantially the first two rules (and therefore the constants that are introduced with it, or the equivalences that are asserted with it) remain similar to the original one; on the other hand, for the third judgment needed to derive the conclusion's interpretation, we will often need to apply multiple rules. This depends mainly on the interpretation that has been assigned to the formation-rule, and the complexity of the set of relations that has been chosen in it influences, as we shall see, all the complexity of the subsequent validation rules and their constant.

However, as we shall see, we are not going to succeed in the interpretation of all the types: the first obstacle will come out during the validation of the natural number type, and furthermore some bigger ones in the interpretation of identity and disjoint union type. These problems, as mentioned above, will depend mainly on the choice that has been made during the interpretation of the respective formation-rule, and so a choice more accurate and targeted to solve the unexpected obstacles may perhaps succeed in the interpretation.

Anyway we are not going to show different choices and/or to solve the problems, but we will give just some advice on how, if it should be possible, try to avoid them.

The matter is that during the developing of this internal model, when we faced the first obstacles, we also started, under the advice of Martin-Löf, the construction of an external model, which was aimed to show that an interpretation like ours was naturally obtained into an extended theory of types with new types constructors. As we shall see the external model, which will be explained in the next chapter, is going to result in a more natural and easier construction than the internal one.

So this is the main reason why we did not waste energy on developing the internal model, trying to solve the problems encountered at any cost. Hence we will just limit ourselves to the observation of them and we will leave to the discretion of the reader the opportunity to solve them.

Furthermore the lack of resolution of these problems and at the same time the good behavior of the external model led us to omit the interpretation of the first universe, U. This, however, could be carried out, without much difficulty, starting from a definition *a là Tarski* like the one presented in [Martin-Löf, 1984]. An idea could be for example interpreting the judgment (that is at the same time the U-formation rule) :

$$\mathsf{U} : set$$

into the following triple:

$$\begin{cases} \mathsf{U} : set \\ \mathsf{U} : set \\ (x_0 : \mathsf{U})(x_1 : \mathsf{U})\mathsf{U} : rel(\mathsf{U}, \mathsf{U}) \quad [x_0, x_1 : \mathsf{U}] \end{cases}$$

and the family of set T over U, such that:

$$\frac{a : \mathsf{U}}{T(a) : set}$$

would became in the internal model the following triple of rules:

$$\frac{a_0 : \mathsf{U}}{T(a_0) : set} \quad \frac{a_1 : \mathsf{U}}{T(a_1) : set} \quad \frac{a_r : \mathsf{U}}{T(a_r) : set}$$

where

$$\begin{cases} a_0 : \mathsf{U} \\ a_1 : \mathsf{U} \\ a_r : \mathsf{U} \end{cases}$$

is the standard interpretation of the premise $a : \mathtt{U}$.

About what concerning the notation we principally refer to the book of Nordström at al. [Nordström et al., 1990], even if sometimes we could use some different notations, although in order to avoid misunderstandings we will introduce each rule every time that we need it, taken for granted that the reader as already confidence with types and their generic rules' constructions, otherwise we refer him to [Nordström et al., 1990].

Finally we show what we did, presenting the interpretation of the various types in the following order: `Bool`, $\mathbb{N}_n$, Π, Σ, $\mathbb{N}$, `Id` and $+$. I suggest the reader to follow the order in which they are presented as it is growing in complexity.

3.1 Interpretation of $\mathbb{N}_2$

From the interpretation of the judgment forms explained in the previous chapter, we are now able to give an explicit interpretation of all the rules of the $\mathbb{N}_2$ type (or the so called `Bool` type). Here I prefer to use the generic notation of enumeration type since it will result more convenient when we will pass from 2 elements to an arbitrary number n of elements.

- The formation rule
$$\mathbb{N}_2 : set$$
when passing to the model, becomes the following triple of judgments:
$$\mathbb{N}_2 : set \qquad \mathbb{N}_2 : set \qquad \mathtt{Id}(\mathbb{N}_2, x, y) : set\ [x, y : \mathbb{N}_2]$$
Where here `Id` denotes the identity type construction.

- For each introduction rule we will have three judgments. For example:
$$i_0 : \mathbb{N}_2$$
becomes:
$$i_0 : \mathbb{N}_2 \qquad i_0 : \mathbb{N}_2 \qquad r(i_0) : \mathtt{Id}(\mathbb{N}_2, i_0, i_0)$$
and analogous for $i_1 : \mathbb{N}_2$:
$$i_1 : \mathbb{N}_2 \qquad i_1 : \mathbb{N}_2 \qquad r(i_1) : \mathtt{Id}(\mathbb{N}_2, i_1, i_1)$$

Note that each canonical element, when passing to the model, is essentially transformed into the reflexivity relation of itself inside the set $\mathbb{N}_2$.

- We have now to interpret the elimination rule, we recall it:

$\mathbb{N}_2$ – Elimination:

$$\frac{\begin{array}{c} c : \mathbb{N}_2 \\ C(x) : \ set \ [x : \mathbb{N}_2] \\ d : C(i_0) \\ e : C(i_1) \end{array}}{case(c, d, e) : C(c)}$$

First of all we need to interpret the premises, and from these interpretations, we should be able to construct an interpretation for the conclusion. Well, each one of the premises will become a triple, as follow:

$$c : \mathbb{N}_2 \rightsquigarrow \begin{cases} c_0 : \mathbb{N}_2 \\ c_1 : \mathbb{N}_2 \\ c_r : \mathtt{Id}(\mathbb{N}_2, c_0, c_1) \end{cases}$$

$C(x) : \ set \ (x : \mathbb{N}_2)$ becomes:

$$\begin{cases} C_0(x_0) : \ set \quad [x_0 : \mathbb{N}_2] \\ C_1(x_1) : \ set \quad [x_1 : \mathbb{N}_2] \\ Q(x_0, x_1, p)(y_0, y_1) : set \quad [x_0 : \mathbb{N}_2, x_1 : \mathbb{N}_2, \\ \quad p : \mathtt{Id}(\mathbb{N}_2, x_0, x_1), y_0 : C_0(x_0), y_1 : C_1(x_1)] \end{cases}$$

And the same yields for the other two premises:

$$d : C(i_0) \rightsquigarrow \begin{cases} d_0 : C_0(i_0) \\ d_1 : C_1(i_0) \\ d_r : Q(i_0, i_0, r(i_0))(d_0, d_1) \end{cases}$$

$$e : C(i_1) \rightsquigarrow \begin{cases} e_0 : C_0(i_1) \\ e_1 : C_1(i_1) \\ e_r : Q(i_1, i_1, r(i_1))(e_0, e_1) \end{cases}$$

Now using these ones we should be able, as said above, to build an interpretation for the conclusion:

$$case(c, d, e) : C(c)$$

that will be of the following form:

$$\begin{cases} k_0 : C_0(c_0) \\ k_1 : C_1(c_1) \\ k_r : Q(c_0, c_1, c_r)(k_0, k_1) \end{cases}$$

Let's begin to construct k_0 and k_1. If we pick respectively the first and the second judgment from all the triples of the interpretations of the premises we get all what is needed to apply the normal $\mathbb{N}_2$-Elimination rule, i.e:

$$\frac{\begin{array}{c} c_0 : \mathbb{N}_2 \\ C_0(x_0) : \ set\ (x_0 : \mathbb{N}_2) \\ d_0 : C_0(i_0) \\ e_0 : C_0(i_1) \end{array}}{case(c_0, d_0, e_0) : C_0(c_0)} :\mathbb{N}_2\text{-El} \qquad \frac{\begin{array}{c} c_1 : \mathbb{N}_2 \\ C_1(x_1) : \ set\ (x_1 : \mathbb{N}_2) \\ d_1 : C_1(i_0) \\ e_1 : C_1(i_1) \end{array}}{case(c_1, d_1, e_1) : C_1(c_1)} :\mathbb{N}_2\text{-El}$$

Hence we can just put:

$$k_0 \equiv case(c_0, d_0, e_0)$$

$$k_1 \equiv case(c_1, d_1, e_1)$$

and we have exactly the first two judgments that we were looking for.

It remains to make the last judgment:

$$k_r : Q(c_0, c_1, c_r)(k_0, k_1)$$

where $c_0 : \mathbb{N}_2, c_1 : \mathbb{N}_2, c_r : \mathtt{Id}(\mathbb{N}_2, c_0, c_1), k_0 : C_0(c_0), k_1 : C_1(c_1)$. In order to find an element of this set, we will use the `Id`-Elimination rule, that we recall here:

`Id` – elimination:

$$\frac{\begin{array}{c} a : A \\ b : A \\ c : \mathtt{Id}(A, a, b) \\ C(x, y, z) : set \quad [x : A, y : A, z : \mathtt{Id}(A, x, y)] \\ f(x) : C(x, x, r(x)) \quad [x : A] \end{array}}{idpeel(c, f) : C(a, b, c)}$$

At this point, to be able to use this rule we can take all the judgments of the interpretation of the premise $c : \mathbb{N}_2$ and we get:

$$c_0 : \mathbb{N}_2 \qquad c_1 : \mathbb{N}_2 \qquad c_r : \mathtt{Id}(\mathbb{N}_2, c_0, c_1)$$

that are similar to the first three premises in the `Id`-elimination, except for the fact that instead of A, we have $\mathbb{N}_2$. Now to get something similar to the fourth premise we can work on

$$Q(x_0, x_1, p)(y_0, y_1) : set$$

under the assumptions

$$[x_0 : \mathbb{N}_2, x_1 : \mathbb{N}_2, p : \mathtt{Id}(\mathbb{N}_2, x_0, x_1), y_0 : C_0(x_0), y_1 : C_1(x_1)]$$

obtained from the interpretation of $C(x) : set\ [x : \mathbb{N}_2]$. To do this, we have to avoid the dependence by y_0 and y_1, and we can do it by instantiating them. Hence we have to select some appropriate elements of, respectively, $C_0(x_0)$ and $C_1(x_1)$, when x_0 and x_1 are generic element of $\mathbb{N}_2$. From what we said above for k_0 and k_1, we can readily put:

$$y_0 = case(x_0, d_0, e_0) : C_0(x_0)$$

$$y_1 = case(x_1, d_1, e_1) : C_1(x_1)$$

In this way we get a similar fourth assumption of the `Id`-elimination, since we don't have anymore the dependence by y_0 and y_1, i.e:

$$Q(x_0, x_1, p)(case(x_0, d_0, e_0), case(x_1, d_1, e_1)) : set$$

in the context:

$$[x_0 : \mathbb{N}_2, x_1 : \mathbb{N}_2, p : \mathtt{Id}(\mathbb{N}_2, x_0, x_1)]$$

Now it remains just to find something similar to the last premise:

$$f(x) : C(x, x, r(x)) \quad [x : A]$$

where instead of $C(x, x, r(x))$ $[x : A]$ we have:

$$Q(x, x, r(x))(case(x, d_0, e_0), case(x, d_1, e_1)) \quad [x : \mathbb{N}_2].$$

We can do this using the $\mathbb{N}_2$-Elimination rule with the following premises:

$$\frac{\begin{array}{c} c : \mathbb{N}_2 \\ Q(x, x, r(x))(case(x, d_0, e_0), case(x, d_1, e_1)) \quad [x : \mathbb{N}_2] \\ d_r : Q(i_0, i_0, r(i_0))(case(i_0, d_0, e_0), case(i_0, d_1, e_1)) \\ e_r : Q(i_1, i_1, r(i_1))(case(i_1, d_0, e_0), case(i_1, d_1, e_1)) \end{array}}{case(c, d_r, e_r) : Q(c, c, r(c))(case(c, d_0, e_0), case(c, d_1, e_1))} :\mathbb{N}_2\text{-El}$$

Observe that all the premises of this rules are correct since when $(case(i_0, d_0, e_0), case(i_0, d_1, e_1))$ is evaluated, by $\mathbb{N}_2$-equality, yields the couple of element (d_0, d_1), and so correctly

$$d_r : Q(i_0, i_0, r(i_0))(d_0, d_1)$$

And for the same reason when we compute $(case(i_1, d_0, e_0), case(i_1, d_1, e_1))$ we obtain (e_0, e_1) and so:

$$e_r : Q(i_1, i_1, r(i_1))(e_0, e_1)$$

as we already had from the initial interpretation.

So we are finally able to use it to obtain the conclusion:

$$case(c, d_r, e_r) : Q(c, c, r(c))(case(c, d_0, e_0), case(c, d_1, e_1))$$

from which we immediately get, abstracting:

$$case(x, d_2, e_2) : Q(x, x, r(x))(case(x, d_0, e_0), case(x, d_1, e_1)) \quad [x : \mathbb{N}_2]$$

that is, exactly, the last premise that we needed to apply the `Id`-elimination (the $f(x)$, in the `Id`-elimination rule, that we were looking for).

Hence, we shall just put all what we said together and use `Id`-elimination:

$$\cfrac{\begin{array}{c} c_0 : \mathbb{N}_2 \\ c_1 : \mathbb{N}_2 \\ c_r : \mathtt{Id}(\mathbb{N}_2, c_0, c_1) \\ \begin{cases} Q(x_0, x_1, p)(\mathit{case}(x_0, d_0, e_0), \mathit{case}(x_1, d_1, e_1)) : \mathit{set} \\ \quad [x_0, x_1 : \mathbb{N}_2, p : \mathtt{Id}(\mathbb{N}_2, x_0, x_1)] \end{cases} \\ \mathit{case}(x, d_r, e_r) : Q(x, x, r(x))(\mathit{case}(x, d_0, e_0), \mathit{case}(x, d_1, e_1)) \quad [x : \mathbb{N}_2] \end{array}}{\mathit{idpeel}(c_r, \mathit{case}(x, d_r, e_r)) : Q(c_0, c_1, c_r)(\mathit{case}(c_0, d_0, e_0), \mathit{case}(c_1, d_1, e_1))}$$

Now notice that above we decided to define:

$$k_0 \equiv \mathit{case}(c_0, d_0, e_0)$$

$$k_1 \equiv \mathit{case}(c_1, d_1, e_1)$$

and so we get that:

$$\mathit{idpeel}(c_r, \mathit{case}(x, d_r, e_r)) : Q(c_0, c_1, c_r)(k_0, k_1),$$

Hence, we can finally define k_r in the following way:

$$k_r \equiv \mathit{idpeel}(c_r, \mathit{case}(x, d_r, e_r)) : Q(c_0, c_1, c_r)(k_0, k_1).$$

To conclude we have the complete interpretation of $\mathit{case}(c, d, e) : C(c)$, i.e.

$$\begin{cases} \mathit{case}(c_0, d_0, e_0) : C_0(c_0) \\ \mathit{case}(c_1, d_1, e_1) : C_1(c_1) \\ \mathit{idpeel}(c_r, \mathit{case}(x, d_r, e_r)) : Q(c_0, c_1, c_r)(\mathit{case}(c_0, d_0, e_0), \mathit{case}(c_1, d_1, e_1)) \end{cases}$$

- At the end we have to check that the equality rules are validated. For the $\mathbb{N}_2$ type we have just the following two equality rules:

  $\mathbb{N}_2$ –Eqaulity:

$$\frac{C(x) : \mathit{set} \quad [x : \mathbb{N}_2] \qquad d : C(i_0) \qquad e : C(i_1)}{\mathit{case}(i_0, d, e) = d : C(i_0)}$$

$$\frac{C(x) : \ \mathit{set}\ [x : \mathbb{N}_2] \qquad d : C(i_0) \qquad e : C(i_1)}{\mathit{case}(i_1, d, e) = e : C(i_1)}$$

We give an explanation for the first one since the second will follow in the same way. As usual we have to construct, starting from the interpretations of the premises, three equality judgments that will form the interpretation of the conclusion.

As we did before, $C(x) : set\ [x : \mathbb{N}_2]$ becomes:

$$\begin{cases} C_0(x_0) : set \quad [x_0 : \mathbb{N}_2] \\ C_1(x_1) : set \quad [x_1 : \mathbb{N}_2] \\ Q(x_0, x_1, p)(y_0, y_1) : set \\ \quad [x_0, x_1 : \mathbb{N}_2, p : \mathtt{Id}(\mathbb{N}_2, x_0, x_1), y_0 : C_0(x_0), y_1 : C_1(x_1)] \end{cases}$$

And the same yields for the other two premises:

$$d : C(i_0) \rightsquigarrow \begin{cases} d_0 : C_0(i_0) \\ d_1 : C_1(i_0) \\ d_r : Q(i_0, i_0, r(i_0))(d_0, d_1) \end{cases}$$

$$e : C(i_1) \rightsquigarrow \begin{cases} e_0 : C_0(i_1) \\ e_1 : C_1(i_1) \\ e_r : Q(i_1, i_1, r(i_1))(e_0, e_1) \end{cases}$$

During the interpretation of the elimination rule we found the interpretation of $case(c, d, e) : C(c)$, that is:

$$\begin{cases} case(c_0, d_0, e_0) : C_0(c_0) \\ case(c_1, d_1, e_1) : C_1(c_1) \\ idpeel(c_r, case(x, d_r, e_r) : Q(c_0, c_1, p)(case(c_0, d_0, e_0), case(c_1, d_1, e_1)) \end{cases}$$

so we have just to substitute c with i_0 and switch their interpretations, i.e. the triple (c_0, c_1, c_r) will become $(i_0, i_0, r(i_0))$. Thus the interpretation of $case(i_0, d, e) : C(i_0)$ is:

$$\begin{cases} case(i_0, d_0, e_0) : C_0(i_0) \\ case(i_0, d_1, e_1) : C_1(i_0) \\ idpeel(r(i_0), case(x, d_r, e_r)) : \\ \quad Q(i_0, i_0, r(i_0))(case(i_0, d_0, e_0), case(i_0, d_1, e_1)) \end{cases}$$

that, by the equality rules of $\mathtt{Id}$ and $\mathbb{N}_2$ types, is definitionally equal to:

$$\begin{cases} d_0 : C_0(i_0) \\ d_1 : C_1(i_0) \\ d_r : Q(i_0, i_0, r(i_0))(d_0, d_1) \end{cases}$$

And this is exactly the interpretation of $d : C(i_0)$. The validation of the other elimination rule is analogous. Ergo even the equality rule is satisfied.

So the validation of the finite type $\mathbb{N}_2$ in the internal model is complete, and as we will see in the next section, even the generic finite type's validation will work in a similar way.

3.2 Interpretation of $\mathbb{N}_n$

We will now argue in the same way we did before, to extend the interpretation of the set with two elements to the set with n elements: $\mathbb{N}_n$, giving an interpretation of all its rules.

- The formation rule,
$$\mathbb{N}_n : set$$
, becomes the following triple of judgments:
$$\mathbb{N}_n : set \qquad \mathbb{N}_n : set \qquad \mathtt{Id}(\mathbb{N}_n, x, y) : set\ [x, y : \mathbb{N}_n]$$
where Id denotes the identity type construction.

- For each introduction rule we have three judgments, as for the set with two elements. The generic introduction rule:
$$i_j : \mathbb{N}_n$$
becomes:
$$i_j : \mathbb{N}_n \qquad i_j : \mathbb{N}_n \qquad r(i_j) : \mathtt{Id}(\mathbb{N}_n, i_j, i_j)$$
And this yields for each $0 \leq j < n$.
So even in this interpretation each canonical element of $\mathbb{N}_n$ is transformed in the reflexivity relation between itself.

- We have now to interpret the elimination rule, we recall it:

$\mathbb{N}_n$ – Elimination:

$$\frac{\begin{array}{c} c : \mathbb{N}_n \\ C(x) : set \quad [x : \mathbb{N}_n] \\ b_0 : C(i_0) \\ \vdots \\ b_{n-1} : C(i_{n-1}) \end{array}}{case(c, b_0, ..., b_{n-1}) : C(c)}$$

The interpretation of the premises are exactly the same we had in the two elements type, where in place of $\mathbb{N}_2$, we simply have $\mathbb{N}_n$:

$$c : \mathbb{N}_n \rightsquigarrow \begin{cases} c_0 : \mathbb{N}_n \\ c_1 : \mathbb{N}_n \\ c_r : \mathtt{Id}(\mathbb{N}_n, c_0, c_1) \end{cases}$$

$C(x) : set\ [x : \mathbb{N}_n]$ becomes:

$$\begin{cases} C_0(x_0) : set\ [x_0 : \mathbb{N}_n] \\ C_1(x_1) : set\ [x_1 : \mathbb{N}_n] \\ Q(x_0, x_1, p)(y_0, y_1) : set \\ \quad [x_0, x_1 : \mathbb{N}_n, p : \mathtt{Id}(\mathbb{N}_n, x_0, x_1), y_0 : C_0(x_0), y_1 : C_1(x_1)] \end{cases}$$

In the interpretation of the generic premises $b_j : C(i_j)$, to avoid confusion with the subscript numbers, we will use superscripts in the following way:

$$b_j : C(i_j) \rightsquigarrow \begin{cases} b_j^0 : C_0(i_j) \\ b_j^1 : C_1(i_j) \\ b_j^r : Q(i_j, i_j, r(i_j))(b_j^0, b_j^1) \end{cases}$$

for each $0 \leq j < n$.

Exactly as we did before, using these ones we shall now be able to construct an interpretation for the conclusion:

$$case(c, b_0, ..., b_{n-1}) : C(c)$$

that will be of the following form:

$$\begin{cases} k_0 : C_0(c_0) \\ k_1 : C_1(c_1) \\ k_r : Q(c_0, c_1, c_r)(k_0, k_1) \end{cases}$$

The way to construct k_0 and k_1 is exactly the same: if we pick respectively the first and the second judgment from all the triples of the interpretations of the premises, we get all what is needed to apply the normal $\mathbb{N}_n$-Elimination rule, hence as before, we can just put:

$$k_0 \equiv case(c_0, b_0^0, \ldots, b_{n-1}^0) : C_0(c_0)$$

$$k_1 \equiv case(c_0, b_0^1, \ldots, b_{n-1}^1) : C_1(c_1)$$

and we have exactly the first two judgments that we were looking for.

Furthermore to built k_r we can move like we did in $\mathbb{N}_2$, adjusting the premises to be able to apply the Id-Elimination rule. Reasoning in a similar way to the previous section we get the following:

$$\frac{\begin{array}{c} c_0 : \mathbb{N}_n \\ c_1 : \mathbb{N}_n \\ c_r : \mathtt{Id}(\mathbb{N}_n, c_0, c_1) \\ \begin{cases} Q(x_0, x_1, p)(\mathit{case}(x_0, b^0_0, ..., b^0_{n-1}), \mathit{case}(x_1, b^1_0, ..., b^1_{n-1})) : \mathit{set} \\ \quad [x_0, x_1 : \mathbb{N}_n, p : \mathtt{Id}(\mathbb{N}_n, x_0, x_1)] \end{cases} \\ \begin{cases} \mathit{case}(x, b^r_0, ..., b^r_{n-1}) : \\ Q(x, x, r(x))(\mathit{case}(x, b^0_0, ..., b^0_{n-1}), \mathit{case}(x, b^1_0, ..., b^1_{n-1})) \\ \quad [x : \mathbb{N}_n] \end{cases} \end{array}}{\begin{cases} \mathit{idpeel}(c_r, \mathit{case}(x, b^r_0, ..., b^r_{n-1})) : \\ \quad Q(c_0, c_1, c_r)(\mathit{case}(c_0, b^0_0, ..., b^0_{n-1}), \mathit{case}(c_1, b^1_0, ..., b^1_{n-1})) \end{cases}} \text{ Id-El}$$

Now notice that by how we defined k_0 and k_1 so we have that:

$$\mathit{idpeel}(c_r, \mathit{case}(x, b^r_0, ..., b^r_{n-1})) : Q(c_0, c_1, c_r)(k_0, k_1),$$

hence, we can finally define k_r in the following way:

$$k_r \equiv \mathit{idpeel}(c_r, \mathit{case}(x, b^r_0, ..., b^r_{n-1})) : Q(c_0, c_1, c_r)(k_0, k_1).$$

and we have finally the complete interpretation of $\mathit{case}(c, b_0, ..., b_{n-1}) : C(c)$.

- In the end we have to check that the equality rules hold. For the $\mathbb{N}_n$ type we have n equality rules,for each $0 \leq j < n$, with the following generic form:

  $\mathbb{N}_n$ – Equality:

$$\frac{C(x) : \mathit{set}\ [x : \mathbb{N}_2] \qquad b_0 : C(i_0) \qquad ... \qquad b_{n-1} : C(i_{n-1})}{\mathit{case}(i_j, b_0, ..., b_{n-1}) = b_j : C(i_j)}$$

As usual we have to give the interpretation of the premises and from these try to build three equality judgments that will form the interpretation of the conclusion.

As we did before, $C(x) : set\ [x : \mathbb{N}_n]$ becomes:

$$\begin{cases} C_0(x_0) : set \quad [x_0 : \mathbb{N}_n] \\ C_1(x_1) : set \quad [x_1 : \mathbb{N}_n] \\ Q(x_0, x_1, p)(y_0, y_1) : set \\ \quad [x_0, x_1 : \mathbb{N}_n, p : \mathtt{Id}(\mathbb{N}_n, x_0, x_1), y_0 : C_0(x_0), y_1 : C_1(x_1)] \end{cases}$$

and for each $0 \le j < n$:

$$b_j \in C(i_j) \rightsquigarrow \begin{cases} b_j^0 : C_0(i_j) \\ b_j^1 : C_1(i_j) \\ b_j^r : Q(i_j, i_j, r(i_j))(b_j^0, b_j^1) \end{cases}$$

Now we have just to substitute c with i_j and switch their interpretation, i.e. the triple (c_0, c_1, c_r) will become $(i_j, i_j, r(i_j))$, in the interpretation, explained above, of $case(c, b_0, ..., b_{n-1}) : C(c)$.

So the interpretation of $case(i_j, b_0, .., b_{n-1}) : C(i_j)$ will be:

$$\begin{cases} case(i_j, b_0^0, .., b_{n-1}^0) : C_0(i_j) \\ case(i_j, b_0^1, .., b_{n-1}^1) : C_1(i_j) \\ idpeel(r(i_j), case(x, b_0^r, ..., b_{n-1}^r)) : \\ \quad Q(i_j, i_j, r(i_j))(case(i_j, b_0^0, .., b_{n-1}^0)), case(i_j, b_0^1, .., b_{n-1}^1)) \end{cases}$$

that from the equality rules of $\mathtt{Id}$ and $\mathbb{N}_n$ types it is definitionally equal to:

$$\begin{cases} b_j^0 : C_0(i_j) \\ b_j^1 : C_1(i_j) \\ b_j^r : Q(i_j, i_j, r(i_j))(b_j^0, b_j^1) \end{cases}$$

which is exactly the interpretation of $b_j : C(i_j)$.

Ergo even the equality rule is satisfied.

Finally we have concluded the interpretation of a generic finite type in the internal model.

3.3 Interpretation of Π type

We show in this section how to interpret the Π type. We will use some slightly different notations from the book [Nordström et al., 1990], for example we call the non-canonical constant of the elimination rule *app* instead of *apply*. Moreover we will just show the elimination for the non-canonical constant *app*, omitting the one for *funsplit*, with the understanding that the same reasoning would lead to the desired result.

- As usual we need to give an interpretation into a triple of judgments. Let start from the Π-formation rule.

  Π – formation:

$$\frac{A : set \qquad B(x) : set \ [x : A]}{(\Pi x : A)B(x) : set}$$

  In higher order notation we can define $(\Pi x : A)B(x) \equiv \Pi(A, B)$, and we are going to use the latter.

  The interpretations of the premises are:

$$A : set \rightsquigarrow \begin{cases} A_0 : set \\ A_1 : set \\ P(x_0, x_1) : set \quad [x_0 : A_0, x_1 : A_1] \end{cases}$$

  and the other is the normal interpretation of the judgment $B(x) : set \ [x : A]$, i.e.

$$\begin{cases} B_0(x_0) : set \quad [x_0 : A_0] \\ B_1(x_1) : set \quad [x_1 : A_1] \\ Q(x_0, x_1, p)(y_0, y_1) : set \\ \quad [x_0 : A_0, x_1 : A_1, p : P(x_0, x_1), y_0 : B_0(x_0), y_1 : B_1(x_1)] \end{cases} \tag{3.1a}$$

  From these two triples we can get, using the Π-formation rule, the following:

$$\frac{A_0 : set \qquad B_0(x_0) : set \ [x_0 : A_0]}{\Pi(A_0, B_0) : set} \ \Pi\text{-formation}$$

$$\frac{A_1 : set \qquad B_1(x_1) : set\ [x_1 : A_1]}{\Pi(A_1, B_1) : set}\ \Pi\text{-formation}$$

and hence we can define:

$$\Pi(A_0, B_0) \equiv \Pi_0(A_0, B_0)$$

$$\Pi(A_1, B_1) \equiv \Pi_1(A_1, B_1)$$

which are the first two judgments of the interpretation of $\Pi(A, B) : set$.

We should now define a set R of relation between $\Pi(A_0, B_0)$ and $\Pi(A_1, B_1)$. In order to do this we recall that in high order notation Π is of the following type:

$$\Pi : (X set, (El(X))set)set$$

By abstraction we can translate between hypothetical judgments and functions as showed in the chapter 19 of [Nordström et al., 1990]. So applying this reasoning on the judgment (3.1a), we obtain by abstracting on p:

$$(p)Q(x_0, x_1, p)(y_0, y_1) : (p : P(x_0, x_1))set$$

in the context

$$[x_0 : A_0, x_1 : A_1, y_0 : B_0(x_0), y_1 : B_1(x_1)]$$

Hence this allows us to create the following set (where the context will be the same as above):

$$\Pi(P(x_0, x_1), (p)Q(x_0, x_1, p)(y_0, y_1)).$$

So iterating the same process by abstracting on x_0 and x_1 we will get:

$$\Pi(A_0, (x_0)\Pi(A_1, (x_1)\Pi(P(x_0, x_1), (p)Q(x_0, x_1, p)(y_0, y_1))))) \tag{3.2}$$

in the context $[y_0 : B_0(x_0), y_1 : B_1(x_1)]$.

Finally we can define the set of relations between $f_0 : \Pi(A_0, B_0)$ and $f_1 : \Pi(A_1, B_1)$, just observing that $app(A_0, B_0, f_0, x_0) : B_0(x_0)$ and $app(A_1, B_1, f_1, x_1) : B_1(x_1)$ (I will omit the first two arguments of *app* when they are understandable from the context). So we can substitute them inside (3.2), obtaining

$$\Pi(A_0, (x_0)\Pi(A_1, (x_1)\Pi(P(x_0, x_1), (p)Q(x_0, x_1, p)(app(f_0, x_0), app(f_1, x_1))))) \tag{3.3}$$

and defining, in order to lighten the notation,

$$R(f_0, f_1) \equiv (3.3)$$

under the assumptions $[f_0 : \Pi(A_0, B_0), f_1 : \Pi(A_1, B_1)]$. Ergo this completes the interpretation of the formation rule:

$$\Pi(A, B) : set \rightsquigarrow \begin{cases} \Pi(A_0, B_0) : set \\ \Pi(A_1, B_1) : set \\ R(f_0, f_1) : set \quad [f_0 : \Pi(A_0, B_0), f_1 : \Pi(A_1, B_1)] \end{cases}$$

- We have now to interpret the introduction rule:

  Π − introduction:

$$\frac{A : set \qquad B(x) : set\ [x : A] \qquad b(x) : B(x)\ [x : A]}{\lambda(A, B, b) : \Pi(A, B)}$$

So, as usual, from the interpretations of the premises we have to find an interpretation of $\lambda(A, B, b) : \Pi(A, B)$ that shall have the following form:

$$\begin{cases} \lambda_0(A_0, B_0, b_0) : \Pi_0(A, B) \\ \lambda_1(A_1, B_1, b_1) : \Pi_1(A, B) \\ \lambda_r(A_0, A_1, P, B_0, B_1, Q, b_0, b_1, q) : R(\lambda_0(A_0, B_0, b_0), \lambda_1(A_0, B_0, b_0)) \end{cases}$$

The interpretations of $A : set$ and $B(x) : set\ [x : A]$ are the same that we used during the formation rule, so we are not going to write again. On the other hand the interpretation of $b(x) : B(x)\ [x : A]$ is:

$$\begin{cases} b_0(x_0) : B_0(x_0) \quad [x_0 : A_0] \\ b_1(x_1) : B_1(x_1) \quad [x_1 : A_1] \\ q(x_0, x_1, p) : Q(x_0, x_1, p)(b_0(x_0), b_1(x_1)) \\ \quad [x_0 : A_0, x_1 : A_1, p : P(x_0, x_1)] \end{cases} \tag{3.4a}$$

Hence taking the first judgment of each triple and using Π-introduction we get:

$$\frac{\begin{array}{c} A_0 : set \\ B_0(x_0) : set \quad [x_0 : A_0] \\ b_0(x_0) : B_0(x_0) \quad [x_0 : A_0] \end{array}}{\lambda(A_0, B_0, b_0) : \Pi(A_0, B_0)} \,\Pi\text{-intro}$$

and so we can define:

$$\lambda_0(A_0, B_0, b_0) \equiv \lambda(A_0, B_0, b_0) : \Pi_0(A, B) = \Pi(A_0, B_0)$$

and, obviously, the same reasoning holds for

$$\lambda_1(A_1, B_1, b_1) \equiv \lambda(A_1, B_1, b_1) : \Pi_1(A, B) = \Pi(A_1, B_1)$$

So we get the first two judgments, and to get the third one we just work as in the formation rule, i.e. we need to abstract on the element in (3.4a) above to be able to use the Π-introduction rule, and we have to repeat this procedure for three times starting, from p:

$$(p)q(x_0, x_1, p) : (p)Q(x_0, x_1, p)(b_0(x_0), b_1(x_1)) \quad [x_0 : A_0, x_1 : A_1]$$

and then using the introduction rules we get:

$$\lambda(P(x_0, x_1), (p)Q(x_0, x_1, p)(b_0(x_0), b_1(x_1)), (p)q(x_0, x_1, p))$$

is an element of:

$$\Pi(P(x_0, x_1), (p)Q(x_0, x_1, p)(b_0(x_0), b_1(x_1)).$$

Now starting from this element and iterating the same reasoning by abstracting on x_0 and x_1 we will get the following (I will omit the sets in the lambda structure since they have a really long notation and they are the same used during the Π formation):

$$\lambda((x_0)\lambda((x_1)\lambda((p)q(x_0, x_1, p)))) \tag{3.5}$$

that is an element of

$$\Pi(A_0, (x_0)\Pi(A_1, (x_1)\Pi(P(x_0, x_1), (p)Q(x_0, x_1, p)(b_0(x_0), b_1(x_1))))))$$

in which, from the equality (computational) rule of *app*, we can exchange the couple $(b_0(x_0), b_1(x_1))$ with:

$$(app(A_0, B_0, \lambda(A_0, B_0, b_0), x_0), app(A_1, B_1, \lambda(A_1, B_1, b_1), x_1))$$

finally from all the definitions given before, we obtain that the element (3.5) belongs to the set:

$$\lambda((x_0)\lambda((x_1)\lambda((p)q(x_0, x_1, p)))) : R(\lambda(A_0, B_0, b_0), \lambda_(A_1, B_1, b_1)).$$

that is the exactly the last judgment that we were seeking. Then to complete the interpretation it suffices to define:

$$\lambda_r(A_0, A_1, P, B_0, B_1, Q, b_0, b_1, q) \equiv \lambda((x_0)\lambda((x_1)\lambda((p)q(x_0, x_1, p))))$$

- At this point we try to validate the elimination rule:

  Π – elimination:

$$\frac{A : set \qquad B(x) : set\ [x : A] \qquad f : \Pi(A, B) \qquad a : A}{app(A, B, f, a) : B(a)}$$

Again starting from the interpretation of the premises we have to find an interpretation of $app(A, B, f, a) : B(a)$ that shall be of this form:

$$\begin{cases} app_0(A_0, B_0, f_0, a_0) : B_0(a_0) \\ app_1(A_1, B_1, f_1, a_1) : B_1(a_1) \\ app_r(A_0, A_1, P, B_0, B_1, Q, f_0, f_1, f_r, a_0, a_1, a_r) : \\ \quad Q(a_0, a_1, a_r)(app_0(A, B, f, a), app_1(A, B, f, a)) \end{cases}$$

where

$$\begin{cases} a_0 : A_0 \\ a_1 : A_1 \\ a_r : P(a_0, a_1) \end{cases}$$

is the interpretation of $a : A$.

The first two premises of the rule are interpreted in the usual way and $f \in \Pi(A, B)$ becomes:

$$\begin{cases} f_0 : \Pi(A_0, B_0) \\ f_1 : \Pi(A_1, B_1) \\ f_r : R(f_0, f_1) \end{cases}$$

Hence, as usual, taking the first judgment of the interpretation of each premise and applying Π-elimination we get:

$$\frac{A_0 : set \qquad B_0(x_0) : set\ [x_0 : A_0] \qquad f_0 : \Pi_0(A, B) \qquad a_0 : A_0}{app(A_0, B_0, f_0, a_0) : B_0(a_0)}$$

and since the same holds even if we take the second judgment from each triple, we can define:

$$app_0(A_0, B_0, f_0, a_0) \equiv app(A_0, B_0, f_0, a_0)$$

$$app_1(A_1, B_1, f_1, a_1) \equiv app(A_1, B_1, f_1, a_1)$$

It remains to find what should be

$$app_r(A_0, A_1, P, B_0, B_1, Q, f_0, f_1, f_r, a_0, a_1, a_r) \tag{3.6}$$

We just claim that since $f_r : R(f_0, f_1)$, that is to be explicit:

$$\Pi(A_0, (x_0)\Pi(A_1, (x_1)\Pi(P(x_0, x_1), (p)Q(x_0, x_1, p)(app(f_0, x_0), app(f_1, x_1)))))$$

it's enough to do the following application $app(app(app(f_r, a_0), a_1), a_r)$ that will be an element of

$$Q(a_0, a_1, a_r)(app(A_0, B_0, f_0, a_0), app(A_1, B_1, f_1, a_1))$$

as required. Hence if we put the element in (3.6) equivalent to the following element

$$app(app(app(f_r, a_0), a_1), a_r)$$

we get the validation of the elimination rule.

- Finally we need just to show that even the equality rule is correct.

Π – equality:

$$\frac{\begin{array}{c} A : set \\ a : A \\ B(x) : set \quad [x : A] \\ b(x) : B(x) \quad [x : A] \end{array}}{app(A, B, \lambda(A, B, b), a) = b(a) : B(a)}$$

So we need to show that the following triple of equality is satisfied starting from the interpretation of the premises:

$$\begin{cases} app_0(A_0, B_0, \lambda_0(A_0, B_0, b_0), a_0) = b_0(a_0) : B_0(a_0) \\ app_1(A_1, B_1, \lambda_1(A_1, B_1, b_1), a_1) = b_1(a_1) : B_1(a_1) \\ app_r(A_0, A_1, P, B_0, B_1, Q, \lambda_0(A_0, B_0, b_0), \lambda_1(A_1, B_1, b_1), \\ \quad \lambda_r(A_0, A_1, P, B_0, B_1, Q, b_0, b_1, q), a_0, a_1, a_r) \\ = q(a_0, a_1, a_r) : \quad Q(a_0, a_1, a_r)(b_0(a_0), b_1(a_1)) \end{cases}$$

But this follows immediately from how we have defined the interpretation of *app* and λ, as required.

Ergo this concludes the interpretation of the product type inside the internal model.

3.4 Interpretation of Σ type

We will explain what the Σ type becomes through the interpretation and validation of its rules.

- Let start from the Σ-formation rule.

  Σ – formation:

  $$\frac{A : set \qquad B(x) : set\ [x : A]}{(\Sigma x : A)B(x) : set}$$

  In higher order notation we can define $(\Sigma x : A)B(x) \equiv \Sigma(A, B)$, and we are going to use the latter.

  As usual we start from the interpretation of the premises:

  $$A : set \rightsquigarrow \begin{cases} A_0 : set \\ A_1 : set \\ P(x_0, x_1) : set \quad [x_0 : A_0, x_1 : A_1] \end{cases}$$

  and the other is the normal interpretation of the judgment $B(x) : set\ [x : A]$:

  $$\begin{cases} B_0(x_0) : set \quad [x_0 : A_0] \\ B_1(x_1) : set \quad [x_1 : A_1] \\ Q(x_0, x_1, p)(y_0, y_1) : set \\ \quad [x_0 : A_0, x_1 : A_1, p : P(x_0, x_1), y_0 : B_0(x_0), y_1 : B_1(x_1)] \end{cases}$$

  From these two triples we can get, by using the Σ-formation rule, the following:

  $$\frac{A_0 : set \qquad B_0(x_0) : set\ [x_0 : A_0]}{\Sigma(A_0, B_0) : set}\ \Sigma\text{-formation}$$

  $$\frac{A_1 : set \qquad B_1(x_1) : set\ [x_1 : A_1]}{\Sigma(A_1, B_1) : set}\ \Sigma\text{-formation}$$

and hence we can define:

$$\Sigma(A_0, B_0) \equiv \Sigma_0(A_0, B_0)$$

$$\Sigma(A_1, B_1) \equiv \Sigma_1(A_1, B_1)$$

where, as usual, $\Sigma_0(A_0, B_0)$ and $\Sigma_1(A_1, B_1)$ are the first two judgments of the interpretation of $\Sigma(A, B)$.

We should now define a set S of relations between $\Sigma_0(A_0, B_0)$ and $\Sigma_1(A_1, B_1)$. Hence, given two elements $c_0 : \Sigma(A_0, B_0)$ and $c_1 : \Sigma(A_1, B_1)$, we have to define what is a relation between them. We can use the projection *fst* and *snd* of the Σ type, defined in the book [Nordström et al., 1990], to obtain the following elements:

$$\mathit{fst}(c_0) : A_0$$

$$\mathit{snd}(c_0) : B_0(\mathit{fst}(c_0))$$

and the same for c_1:

$$\mathit{fst}(c_1) : A_1$$

$$\mathit{snd}(c_1) : B_1(\mathit{fst}(c_1))$$

Using these facts, from the premises

$$\begin{array}{rl} P(x_0, x_1) : \mathit{set} & [x_0 : A_0, x_1 : A_1] \\ Q(x_0, x_1, p)(y_0, y_1) : \mathit{set} & [x_0 : A_0, x_1 : A_1, \\ & \quad p : P(x_0, x_1), y_0 : B_0(x_0), y_1 : B_1(x_1)] \end{array}$$

we get, by substitution:

$$\begin{array}{c} P(\mathit{fst}(c_0), \mathit{fst}(c_1)) : \mathit{set} \\ Q(\mathit{fst}(c_0), \mathit{fst}(c_1), p)(\mathit{snd}(c_0), \mathit{snd}(c_1)) : \mathit{set}\ [p : P(\mathit{fst}(c_0), \mathit{fst}(c_1))] \end{array}$$

So we can use the Σ-formation to create a new set that will be the set of relations $S(c_0, c_1)$:

$$\frac{\begin{array}{c} P(\mathit{fst}(c_0), \mathit{fst}(c_1)) : \mathit{set} \\ \begin{cases} Q(\mathit{fst}(c_0), \mathit{fst}(c_1), p)(\mathit{snd}(c_0), \mathit{snd}(c_1)) : \mathit{set} \\ \quad [p : P(\mathit{fst}(c_0), \mathit{fst}(c_1))] \end{cases} \end{array}}{\begin{cases} \Sigma(P(\mathit{fst}(c_0), \mathit{fst}(c_1)), \\ (p : P(\mathit{fst}(c_0), \mathit{fst}(c_1)))Q(\mathit{fst}(c_0), \mathit{fst}(c_1), p)(\mathit{snd}(c_0), \mathit{snd}(c_1))) : \mathit{set} \end{cases}}$$

So we can define the set of relations between $c_0 : \Sigma(A_0, B_0)$ and $c_1 : \Sigma(A_1, B_1)$:

$$S(c_0, c_1)$$

putting it equivalent to:

$$\Sigma(P(\mathit{fst}(c_0), \mathit{fst}(c_1)), (p)Q(\mathit{fst}(c_0), \mathit{fst}(c_1), p)(\mathit{snd}(c_0), \mathit{snd}(c_1)))$$

Hence this completes the interpretation of the formation rule:

$$\Sigma(A, B) : set \rightsquigarrow \begin{cases} \Sigma(A_0, B_0) : set \\ \Sigma(A_1, B_1) : set \\ S(c_0, c_1) : set \quad [c_0 : \Sigma(A_0, B_0), c_1 : \Sigma(A_1, B_1)] \end{cases}$$

- Now we have to interpret the introduction rule:

  Σ – introduction:

$$\frac{A : set \qquad B(x) : set\ [x : A] \qquad a : A \qquad b : B(a)}{\langle a, b\rangle : \Sigma(A, B)}$$

  So as usual from the interpretation of the premises we have to find an interpretation of $\langle a, b\rangle : \Sigma(A, B)$ that shall have the following form:

$$\begin{cases} \langle a_0, b_0\rangle_0 : \Sigma(A_0, B_0) \\ \langle a_1, b_1\rangle_1 : \Sigma(A_1, B_1) \\ \langle a_0, a_1, a_r, b_0, b_1, b_r\rangle_r : S(\langle a_0, b_0\rangle_0, \langle a_1, b_1\rangle_1) \end{cases}$$

  The interpretation of $A : set$ and $B(x) : set\ [x : A]$ will be the same that we used in the formation rule, so we are not going to write them again. Instead the interpretation of $b : B(a)$ and $a : A$ are:

$$a : A \rightsquigarrow \begin{cases} a_0 : A_0 \\ a_1 : A_1 \\ a_r : P(a_0, a_1) \end{cases}$$

$$b : B(a) \rightsquigarrow \begin{cases} b_0 : B_0(a_0) \\ b_1 : B_1(a_1) \\ b_r : Q(a_0, a_1, a_r)(b_0, b_1) \end{cases}$$

  At this point, taking the first judgment of each triple and using Σ-introduction we get:

$$\frac{\begin{array}{c} A_0 : set \\ B_0(x_0) : set \quad [x_0 : A_0] \\ a_0 : A_0 \\ b_0 : B_0(a_0) \end{array}}{\langle a_0, b_0 \rangle : \Sigma(A_0, B_0)} \ \Sigma\text{-intro}$$

and so we can define:

$$\langle a_0, b_0 \rangle_0 \equiv \langle a_0, b_0 \rangle$$

and the same reasoning holds for

$$\langle a_1, b_1 \rangle_1 \equiv \langle a_1, b_1 \rangle$$

Hence we get the first two judgments, and in order to obtain the third one we just need to observe that we can use the following rule:

$$\frac{\begin{array}{c} P(a_0, a_1) : set \\ Q(a_0, a_1, p)(b_0, b_1) : set \quad [p : P(a_0, a_1)] \\ a_r : P(a_0, a_1) \\ b_r : Q(a_0, a_1, a_r)(b_0, b_1) \end{array}}{\langle a_r, b_r \rangle : \Sigma(P(a_0, a_1), (p)Q(a_0, a_1, p)(b_0, b_1))} \ \Sigma\text{-intro}$$

and we just need to claim, by the definition we gave before for the set S, that:

$$\Sigma(P(a_0, a_1), (p)Q(a_0, a_1, p)(b_0, b_1)) \equiv S(\langle a_0, b_0 \rangle, \langle a_1, b_1 \rangle)$$

So we have found an element belonging to the set required, and we can complete the interpretation by putting:

$$\langle a_0, a_1, a_r, b_0, b_1, b_r \rangle_r \equiv \langle a_r, b_r \rangle$$

which belong to the set:

$$S(\langle a, b \rangle_0, \langle a, b \rangle_1)$$

as wanted.

- We try now to validate the elimination rule:

  Σ – elimination:

$$\frac{\begin{array}{c} C(v) : set \quad [v : \Sigma(A, B)] \\ c : \Sigma(A, B) \\ d(x, y) : C(\langle x, y \rangle) \quad [x : A, y : B(x)] \end{array}}{split(c, d) : C(c)}$$

Again starting from the interpretation of the premises we have to find an interpretation of $split(c, d) : C(c)$ that shall be of this form:

$$\begin{cases} split_0(c_0, d_0) : C_0(c_0) \\ split_1(c_1, d_1) : C_1(c_1) \\ split_r(c_0, c_1, c_r, d_0, d_1, d_r) : R(c_0, c_1, c_r)(split_0(c_0, d_0), split_1(c_1, d_1)) \end{cases}$$

where

$$\begin{cases} c_0 : \Sigma(A_0, B_0) \\ c_1 : \Sigma(A_1, B_1) \\ c_r : S(c_0, c_1) \end{cases}$$

$$\begin{cases} C_0(v_0) : set \quad [v_0 : \Sigma(A_0, B_0)] \\ C_1(v_1) : set \quad [v_1 : \Sigma(A_1, B_1)] \\ R(v_0, v_1, v_r)(z_0, z_1) : set \quad [v_0 : \Sigma(A_0, B_0), \\ \quad v_1 : \Sigma(A_1, B_1), v_r : S(v_0, v_1), z_0 : C_0(v_0), z_1 : C_1(v_1)] \end{cases}$$

are the usual interpretation of $c : \Sigma(A, B)$ and $C(v) : set\ [v : \Sigma(A, B)]$. Instead, $d(x, y) : C(\langle x, y \rangle)\ [x : A, y : B(x)]$ becomes:

$$\begin{cases} d_0(x_0, y_0) : C_0(\langle x_0, y_0 \rangle) \quad [x_0 : A_0, y_0 : B_0(x_0)] \\ d_1(x_1, y_1) : C_1(\langle x_1, y_1 \rangle) \quad [x_1 : A_1, y_1 : B_1(x_1)] \\ d_r(p, q) : R(\langle x_0, y_0 \rangle, \langle x_1, y_1 \rangle, \langle p, q \rangle)(d_0(x_0, y_0), d_1(x_1, y_1)) \\ \quad [x_0 : A_0, y_0 : B_0(x_0), x_1 : A_1, y_1 : B_1(x_1), p : P(x_0, x_1), \\ \quad q : Q(x_0, x_1, p)(y_0, y_1), d_0(x_0, y_0) : C_0(\langle x_0, y_0 \rangle), \\ \quad d_1(x_1, y_1) : C_1(\langle x_1, y_1 \rangle)] \end{cases}$$

Now, as usual, taking the first judgment of the interpretations of each premise and applying Σ-elimination we get:

$$\frac{\begin{array}{c} C_0(v_0)\ set \quad [v_0 : \Sigma_0(A, B)] \\ c_0 : \Sigma_0(A, B) \\ d_0(x_0, y_0) : C_0(\langle x_0, y_0 \rangle) \quad [x_0 : A_0, y_0 : B_0(x_0)] \end{array}}{split(c_0, d_0) : C_0(c_0)}$$

and since the same holds even if we take the second judgment from each triple, we can define:

$$split_0(c_0, d_0) \equiv split(c_0, d_0) : C_0(c_0)$$

$$split_1(c_1, d_1) \equiv split(c_1, d_1) : C_1(c_1)$$

It remains to find what should be $split_r(c_0, c_1, c_r, d_0, d_1, d_r)$. First of all we can substitute $c_0 : \Sigma(A_0, B_0)$ and $c_1 : \Sigma(A_1, B_1)$ in place of v_0 and v_1 in the following judgment:

$$R(v_0, v_1, v_r)(z_0, z_1) : set$$

where

$$[v_0 : \Sigma(A_0, B_0), v_1 : \Sigma(A_1, B_1), v_r : S(v_0, v_1), z_0 : C_0(v_0), z_1 : C_1(v_1)]$$

obtaining:

$$R(c_0, c_1, v_r)(z_0, z_1) : set \quad [v_r : S(c_0, c_1), z_0 : C_0(c_0), z_1 : C_1(c_1)] \quad (3.7)$$

Moreover we can observe that from $c_0 : \Sigma(A_0, B_0)$ we obtain $fst(c_0) : A_0$ and $snd(c_0) : B_0(fst(c_0))$, so if we substitute them inside the interpretation of $d(x, y)$ in place of x_0 and y_0 we get:

$$d_0(fst(c_0), snd(c_0)) : C_0(\langle fst(c_0), snd(c_0)\rangle) = C_0(c_0)$$

and the same reasoning leads us to the following:

$$d_1(fst(c_1), snd(c_1)) : C_1(\langle fst(c_1), snd(c_1)\rangle) = C_1(c_1)$$

Applying the same substitutions in d_r we get:

$$d_r(p, q) : R(c_0, c_1, \langle p, q\rangle)(d_0(fst(c_0), snd(c_0)), d_1(fst(c_1), snd(c_1))$$

with the context

$$[p : P(fst(c_0), fst(c_1)), q : Q(fst(c_0), fst(c_1), p)(snd(c_0), snd(c_1))]$$

Then we can even substitute the two elements z_0 and z_1 in the set (3.7), obtaining:

$$R(c_0, c_1, v_r)(d_0(fst(c_0), snd(c_0)), d_1(fst(c_1), snd(c_1))) : set$$

when $[v_r : S(c_0, c_1)]$

Hence we need just to remember that S is a Σ-type, and we should be able to use the Σ-elimination (I will omit the context but is exactly the same as we stated before):

$$\frac{\begin{cases} R(c_0, c_1, v_r)(d_0(fst(c_0), snd(c_0)), d_1(fst(c_1), snd(c_1))) : set \\ \quad [v_r : S(c_0, c_1)] \end{cases} \quad c_r : S(c_0, c_1) \quad \begin{cases} d_r(p,q) : R(c_0, c_1, \langle p, q\rangle)(d_0(fst(c_0), snd(c_0)), d_1(fst(c_1), snd(c_1)) \\ \quad [p : P(fst(c_0), fst(c_1)), q : Q(fst(c_0), fst(c_1), p)(snd(c_0), snd(c_1))] \end{cases}}{split(c_r, d_r) : R(c_0, c_1, c_r)(d_0(fst(c_0), snd(c_0)), d_1(fst(c_1), snd(c_1)))}$$

Finally it is enough to recall the computation rule of $split(c, d)$, i.e. we have to evaluate first of all c that it will yields a canonical element of the form $\langle fst(c), snd(c)\rangle$; then the value of $split(c, d)$ will be the value of the computation of $d(fst(c), snd(c))$.

So since before we defined:

$$split_0(c_0, d_0) \equiv split(c_0, d_0) : C_0(c_0)$$

$$split_1(c_1, d_1) \equiv split(c_1, d_1) : C_1(c_1)$$

we get that the set:

$$R(c_0, c_1, c_r)(d_0(fst(c_0), snd(c_0)), d_1(fst(c_1), snd(c_1)))$$

is exactly:

$$R(c_0, c_1, c_r)(split(c_0, d_0), split(c_1, d_1))$$

ergo $split(c_r, d_r)$ is exactly an element in the set that we required and so we can conclude the interpretation just putting:

$$split_r(c_0, c_1, c_r, d_0, d_1, d_r) \equiv split(c_r, d_r)$$

- Finally we need just to show that also the equality rule is valid. Σ

  – equality:

$$\frac{\begin{array}{c} a : A \\ b : B(a) \\ C(v)\ set\ [v : \Sigma(A, B)] \\ d(x, y) : C(\langle x, y\rangle)\ [x : A, y : B(x)] \end{array}}{split(\langle a, b\rangle, d) = d(a, b) : C(\langle a, b\rangle)}$$

So we need to show that the following triple of equality is satisfied starting from the interpretation of the premises:

$$\begin{cases} split_0(\langle a_0, b_0\rangle, d_0) = d_0(a_0, b_0) : C_0(\langle a, b\rangle_0) \\ split_1(\langle a_1, b_1\rangle, d_1) = d_1(a_1, b_1) : C_1(\langle a, b\rangle_1) \\ split_r(\langle a_0, b_0\rangle, \langle a_1, b_1\rangle, \langle a_r, b_r\rangle, d_0, d_1, d_r) = d_r(a_r, b_r) : \\ \quad R(\langle a_0, b_0\rangle, \langle a_1, b_1\rangle, \langle a_r, b_r\rangle)(d_0(a_0, b_0), d_1(a_1, b_1)) \end{cases}$$

But this follows immediately from how we have defined the interpretation of *split*, and from its standard computation. Hence we leave this easy check to the reader.

Finally even the Σ-type is interpreted in our internal model.

3.5 Interpretation of $\mathbb{N}$

Following the same pattern of the finite types, we show now the interpretation of the $\mathbb{N}$ type.

- The formation rule:

$$\mathbb{N} : set$$

becomes the following triple:

$$\begin{cases} \mathbb{N} : set \\ \mathbb{N} : set \\ \mathtt{Id}(\mathbb{N}, x_0, x_1) : set \quad [x_0, x_1 : \mathbb{N}] \end{cases}$$

- The introduction rules:

$$0 : \mathbb{N} \qquad\qquad \frac{a : \mathbb{N}}{succ(a) : \mathbb{N}}$$

we can immediately define the interpretation of the canonical constant 0 just putting:

$$\begin{cases} 0 : \mathbb{N} \\ 0 : \mathbb{N} \\ r(0) : \mathtt{Id}(\mathbb{N}, 0, 0) \end{cases}$$

The second rule's premise give us the following triple:

$$\begin{cases} a_0 : \mathbb{N} \\ a_1 : \mathbb{N} \\ a_r : \mathtt{Id}(\mathbb{N}, a_0, a_1) \end{cases}$$

from which we can get immediately $succ(a_0) : \mathbb{N}$ and $succ(a_1) : \mathbb{N}$. We need now to construct an element in the set $\mathtt{Id}(\mathbb{N}, succ(a_0), succ(a_1))$ and we can do it using the $\mathtt{Id}$-elimination:

$\mathtt{Id}$ – elimination:

$$\frac{\begin{array}{c} a_0 : \mathbb{N} \\ a_1 : \mathbb{N} \\ a_r : \mathtt{Id}(\mathbb{N}, a_0, a_1) \\ C(x, y, z) : set \quad [x, y : \mathbb{N}, z : \mathtt{Id}(\mathbb{N}, x, y)] \\ f(x) : C(x, x, r(x)) \quad [x : \mathbb{N}] \end{array}}{idpeel(a_r, f) : C(a_0, a_1, a_r)}$$

where in place of C we have $(x, y, z)\mathtt{Id}(\mathbb{N}, succ(x), succ(y))$ and in place of $f(x)$ we have $r(succ(x))$. So we get the following conclusion:

$$idpeel(a_r, (x)r(succ(x))) : \mathtt{Id}(\mathbb{N}, succ(a_0), succ(a_1))$$

as required.

Hence the interpretation of $succ(a) : \mathbb{N}$ becomes:

$$\begin{cases} succ_0(a_0) \equiv succ(a_0) : \mathbb{N} \\ succ_1(a_1) \equiv succ(a_1) : \mathbb{N} \\ succ_r(a_0, a_1, a_r) \equiv idpeel(a_r, (x)r(succ(x))) : \mathtt{Id}(\mathbb{N}, succ(a_0), succ(a_1)) \end{cases}$$

- Now we have to validate the elimination rules:

  $\mathbb{N}$ – Elimination:

$$\frac{\begin{array}{c} c : \mathbb{N} \\ C(v) : set \quad [v : \mathbb{N}] \\ d : C(0) \\ e(x, y) : C(succ(x)) \quad [x : \mathbb{N}, y : C(x)] \end{array}}{natrec(c, d, e) : C(c)}$$

  First of all we need to interpret the premises, and from these interpretations, we should be able to get an interpretation for the conclusion. Well, each one of the premises will become a triple, as follow:

$$c : \mathbb{N} \rightsquigarrow \begin{cases} c_0 : \mathbb{N} \\ c_1 : \mathbb{N} \\ c_r : \mathtt{Id}(\mathbb{N}, c_0, c_1) \end{cases}$$

  $C(v) : set\ [v : \mathbb{N}]$ becomes:

$$\begin{cases} C_0(v_0) : set \quad [v_0 : \mathbb{N}] \\ C_1(v_1) : set \quad [v_1 : \mathbb{N}] \\ Q(v_0, v_1, p)(w_0, w_1) : set \\ \quad [v_0, v_1 : \mathbb{N}, p : \mathtt{Id}(\mathbb{N}, v_0, v_1), w_0 : C_0(v_0), w_1 : C_1(v_1)] \end{cases}$$

$$d : C(0) \rightsquigarrow \begin{cases} d_0 : C_0(0) \\ d_1 : C_1(0) \\ d_r : Q(0, 0, r(0))(d_0, d_1) \end{cases}$$

  and the last one, $e(x, y) : C(succ(x))\ [x : \mathbb{N}, y : C(x)]$, becomes:

$$\begin{cases} e_0(x_0, y_0) : C_0(succ(x_0)) \quad [x_0 : \mathbb{N}, y_0 : C_0(x_0)] \\ e_1(x_1, y_1) : C_1(succ(x_1)) \quad [x_1 : \mathbb{N}, y_1 : C_1(x_1)] \\ e_r(x_0, x_1, x_r, y_0, y_1, y_r) : \\ Q(succ(x_0), succ(x_1), idpeel(x_r, (x)r(succ(x))))(e_0(x_0, y_0), e_1(x_1, y_1)) \\ \quad [x_0 : \mathbb{N}, y_0 : C_0(x_0), x_1 : \mathbb{N}, y_1 : C_1(x_1), \\ \quad x_r : \mathtt{Id}(\mathbb{N}, x_0, x_1), y_r : Q(x_0, x_1, x_r)(y_0, y_1)] \end{cases}$$

Now, using these ones, we should be able to construct an interpretation for the conclusion:

$$natrec(c, d, e) : C(c)$$

that it will be of the following form:

$$\begin{cases} k_0 : C_0(c_0) \\ k_1 : C_1(c_1) \\ k_r : Q(c_0, c_1, c_r)(k_0, k_1) \end{cases}$$

where (c_0, c_1, c_r) is the triple of elements getting by interpretation of $c : \mathbb{N}$.

Let's begin to construct k_0 and k_1. If we pick respectively the first and the second judgment from all the triples of the interpretations of the premises we get all what is needed to apply the normal ℕ-Elimination rule, i.e.

$$\frac{\begin{array}{c} c_0 : \mathbb{N} \\ C_0(v_0) : set\ [v_0 : \mathbb{N}] \\ d_0 : C_0(0) \\ e_0(x_0, y_0) : C_0(succ(x_0)) \quad [x_0 : \mathbb{N}, y_0 : C_0(x_0)] \end{array}}{natrec(c_0, d_0, e_0) : C_0(c_0)} \text{:ℕ-El}$$

$$\frac{\begin{array}{c} c_1 : \mathbb{N} \\ C_1(v_1) : set \quad [v_1 : \mathbb{N}] \\ d_1 : C_1(0) \\ e_1(x_1, y_1) : C_1(succ(x_1)) \quad [x_1 : \mathbb{N}, y_1 : C_1(x_1)] \end{array}}{natrec(c_1, d_1, e_1) : C_1(c_1)} \text{:ℕ-El}$$

Hence we can just put:

$$k_0 \equiv natrec(c_0, d_0, e_0)$$

$$k_1 \equiv natrec(c_1, d_1, e_1)$$

and we have exactly the first two judgments that we were looking for.

It remains to make the last judgment:

$$k_r : Q(c_0, c_1, c_r)(k_0, k_1)$$

where

$$[c_0 : \mathbb{N}, c_1 : \mathbb{N}, c_r : \mathtt{Id}(\mathbb{N}, c_0, c_1), k_0 : C_0(c_0), k_1 : C_1(c_1)]$$

In order to find an element of this set, we will use the Id-Elimination rule, that we recalled before.

Now to be able to use this rule we must take all the judgments of the interpretation of the premise $c : \mathbb{N}$ and we get:

$$c_0 : \mathbb{N} \qquad c_1 : \mathbb{N} \qquad c_r : \mathtt{Id}(\mathbb{N}, c_0, c_1)$$

that are similar to the first three premises in the Id-elimination, and to get something similar to the fourth one we have to work on

$$Q(v_0, v_1, p)(w_0, w_1) : set$$

in the context:

$$[v_0 : \mathbb{N}, v_1 : \mathbb{N}, p : Id(\mathbb{N}, v_0, v_1), w_0 : C_0(v_0), w_1 : C_1(v_1)]$$

obtained from the interpretation of $C(v)$: *set* $[v : \mathbb{N}]$. In order to obtain what we wanted, we have to avoid the dependence by w_0 and w_1, and we can do it by instantiating them. Hence we have to choose some appropriate elements of, respectively, $C_0(v_0)$ and $C_1(v_1)$, when v_0 and v_1 are generic elements of $\mathbb{N}$. From what we said above for k_0 and k_1 we can immediately put:

$$w_0 = natrec(v_0, d_0, e_0) : C_0(v_0)$$

$$w_1 = natrec(v_1, d_1, e_1) : C_1(v_1)$$

In this way we get a similar fourth premise of the Id-elimination, since we do not have anymore the dependence of w_0 and w_1:

$$Q(v_0, v_1, p)(natrec(v_0, d_0, e_0), natrec(v_1, d_1, e_1)) : set$$

under the assumptions

$$[v_0 : \mathbb{N}, v_1 : \mathbb{N}, p : \mathtt{Id}(\mathbb{N}, v_0, v_1)]$$

Now it remains just to find something similar to the last premise:

$$f(x) : C(x, x, r(x)) \quad [x : A]$$

but where instead of $C(x, x, r(x))$ $[x : A]$ we have:

$$Q(v, v, r(v))(natrec(v, d_0, e_0), natrec(v, d_1, e_1)) \quad [v : \mathbb{N}].$$

We may do this, exactly in the same way as we built k_0 and k_1, using the $\mathbb{N}$-Elimination rule, but in order to use it we have to adjust the following judgment:

$$\begin{array}{l} e_r(x_0, x_1, x_r, y_0, y_1, y_r) : \\ Q(succ(x_0), succ(x_1), idpeel(x_r, (x)r(succ(x))))(e_0(x_0, y_0), e_1(x_1, y_1)) \\ \quad [x_0 : \mathbb{N}, y_0 : C_0(x_0), x_1 : \mathbb{N}, y_1 : C_1(x_1), \\ \quad x_r : \mathtt{Id}(\mathbb{N}, x_0, x_1), y_r : Q(x_0, x_1, x_r)(y_0, y_1)] \end{array}$$

where we have $x_0 = x_1 = v$.

We shall do this in a way that it will depend only on two variables: $v : \mathbb{N}$ and $y_r : Q(v, v, r(v))(y_0, y_1)$. So we need just to choose $x_r = r(v) : \mathtt{Id}(\mathbb{N}, v, v)$ and to take two elements in the sets $C_0(v)$ and $C_1(v)$ in such way that we can substitute y_0 and y_1. From what we said above for k_0 and k_1 a natural choice it will be:

$$y_0 = natrec(v, d_0, e_0) : C_0(v)$$

$$y_1 = natrec(v, d_1, e_1) : C_1(v)$$

Moreover, by using the following equivalence:

$$idpeel(r(v), (x)r(succ(x))) = r(succ(v))$$

obtained readily applying $\mathtt{Id}$-elimination, we get:

$$\begin{array}{l} e_r(v, v, r(v), natrec(v, d_0, e_0), natrec(v, d_1, e_1), y_r) : \\ Q(succ(v), succ(v), r(succ(v))) \\ (e_0(v, natrec(v, d_0, e_0)), e_1(v, natrec(v, d_1, e_1))) \\ \quad [v : \mathbb{N}, y_r : Q(v, v, r(v))(natrec(v, d_0, e_0), natrec(v, d_1, e_1))] \end{array}$$

It suffices now to note that, from the *natrec* computation (ℕ-equality):

$$e_0(v, natrec(v, d_0, e_0)) = natrec(succ(v), d_0, e_0)$$

$$e_1(v, natrec(v, d_1, e_1)) = natrec(succ(v), d_1, e_1)$$

and so we found an element in the set, when $[v : \mathbb{N}]$,:

$$Q(succ(v), succ(v), r(succ(v)))(natrec(succ(v), d_0, e_0), natrec(succ(v), d_1, e_1))$$

as required.

In order to lighten the syntax we define $g(x, y_r)$ to be equivalent to:

$$e_r(x, x, r(x), natrec(x, d_0, e_0), natrec(x, d_1, e_1), y_r)$$

Finally we are able to apply ℕ-elimination rule with the following premises:

$$\frac{\begin{array}{c} c : \mathbb{N} \\ Q(v, v, r(v))(natrec(v, d_0, e_0), natrec(v, d_1, e_1)) : set \quad [v : \mathbb{N}] \\ d_r : Q(0, 0, r(0))(natrec(0, d_0, e_0), natrec(0, d_1, e_1)) \\ \begin{cases} g(x, y_r) : Q(succ(x), succ(x), r(succ(x))) \\ (natrec(succ(x), d_0, e_0), natrec(succ(x), d_1, e_1)) \\ \quad [x : \mathbb{N}, y_r : Q(x, x, r(x))(natrec(x, d_0, e_0), natrec(x, d_1, e_1))] \end{cases} \end{array}}{natrec(c, d_r, g) : Q(c, c, r(c))(natrec(c, d_0, e_0), natrec(c, d_1, e_1))} \text{:ℕ-El}$$

The premises of this rules are correct since when

$$(natrec(0, d_0, e_0), natrec(0, d_1, e_1))$$

are evaluated yield the couple of element,

$$(d_0, d_1)$$

and correctly

$$d_r : Q(0, 0, r(0))(d_0, d_1)$$

and analogous for g_r, for the explanation given above.

So we can use the rule to obtain the conclusion:

$$natrec(c, d_r, g) : Q(c, c, r(c))(natrec(c, d_0, e_0), natrec(c, d_1, e_1))$$

from which we immediately get by abstraction, when $x : \mathbb{N}$:

$$natrec(x, d_r, g) : Q(x, x, r(x))(naterc(x, d_0, e_0), natrec(x, d_1, e_1))$$

that is, exactly, the last premise that we needed to apply the $\mathtt{Id}$-elimination (is the $f(x)$, in the $\mathtt{Id}$-elimination rule, that we were looking for).

Hence now, we have just to put all together:

$$\frac{\begin{array}{c} c_0 : \mathbb{N} \\ c_1 : \mathbb{N} \\ c_r : \mathtt{Id}(\mathbb{N}, c_0, c_1) \\ \begin{cases} Q(x_0, x_1, x_r)(natrec(x_0, d_0, e_0), natrec(x_1, d_1, e_1)) : set \\ \quad [x_0, x_1 : \mathbb{N}, x_r : \mathtt{Id}(\mathbb{N}, x_0, x_1)] \end{cases} \\ \begin{cases} natrec(x, d_r, g) : \\ Q(x, x, r(x))(natrec(x, d_0, e_0), natrec(x, d_1, e_1)) \\ \quad [x : \mathbb{N}] \end{cases} \end{array}}{\begin{cases} idpeel(c_r, (x)natrec(x, d_r, g)) : \\ Q(c_0, c_1, c_r)(natrec(c_0, d_0, e_0), natrec(c_1, d_1, e_1)) \end{cases}}$$

Note that before we decided to put:

$$k_0 \equiv natrec(c_0, d_0, e_0)$$

$$k_1 \equiv natrec(c_1, d_1, e_1)$$

and so we have that:

$$idpeel(c_r, (x)natrec(x, d_r, g)) : Q(c_0, c_1, c_r)(k_0, k_1),$$

hence, we can finally define k_r in the following way:

$$k_r \equiv idpeel(c_r, (x)natrec(x, d_r, g)) : Q(c_0, c_1, c_r)(k_0, k_1).$$

In conclusion we have the complete interpretation of $natrec(c, d, e) : C(c)$:

$$\begin{cases} natrec(c_0, d_0, e_0) : C_0(c_0) \\ natrec(c_1, d_1, e_1) : C_1(c_1) \\ idpeel(c_r, (x)natrec(x, d_r, g)) : \\ \quad Q(c_0, c_1, c_r)(natrec(c_0, d_0, e_0), natrec(c_1, d_1, e_1)) \end{cases}$$

- In the end we have to check that the equality rule is valid. For the $\mathbb{N}$ type we have the following two equality rules:

  $\mathbb{N}$ – Eqaulity:

$$\frac{\begin{array}{c} C(v) : set \quad [v : \mathbb{N}] \\ d : C(0) \\ e(x,y) : C(succ(x)) \quad [x : \mathbb{N}, y : C(x)] \end{array}}{natrec(0,d,e) = d : C(0)}$$

$$\frac{\begin{array}{c} C(v) : set \quad [v : \mathbb{N}] \\ a : \mathbb{N} \\ d : C(0) \\ e(x,y) : C(succ(x)) \quad [x : \mathbb{N}, y : C(x)] \end{array}}{natrec(succ(a),d,e) = e(a, natrec(a,d,e)) : C(succ(a))}$$

  We start giving an explanation for the first one. As usual we have to give the interpretation of the premises and from these try to build three equality judgments that will form the interpretation for the conclusion.

  The interpretations of the premises are exactly the same we used before, so we will not write them again.

  In the last part we found the interpretation of $natrec(c,d,e) : C(c)$, that is:

$$\begin{cases} natrec(c_0, d_0, e_0) : C_0(c_0) \\ natrec(c_1, d_1, e_1) : C_1(c_1) \\ idpeel(c_r, (x)natrec(x, d_r, g)) : \\ \quad Q(c_0, c_1, c_r)(natrec(c_0, d_0, e_0), natrec(c_1, d_1, e_1)) \end{cases}$$

  so we have just to substitute c with 0 and switch their interpretations, i.e. the triple (c_0, c_1, c_r) will become $(0, 0, r(0))$. So the interpretation of $natrec(0,d,e) : C(0)$ is:

$$\begin{cases} natrec(0, d_0, e_0) : C_0(0) \\ natrec(0, d_1, e_1) : C_1(0) \\ idpeel(r(0), (x)natrec(x, d_r, g)) : \\ \quad Q(0, 0, r(0))(natrec(0, d_0, e_0), natrec(0, d_1, e_1)) \end{cases}$$

that from the equality rules of `Id` and $\mathbb{N}$ types it is equivalent to:

$$\begin{cases} d_0 : C_0(0) \\ d_1 : C_1(0) \\ d_r : Q(0,0,r(0))(d_0,d_1) \end{cases}$$

which is exactly the interpretation of $d : C(0)$, as required.

We will show now that we will not be able to validate the second equality rule.

What we should validate is the triple of equalities coming from the following judgment:

$$natrec(succ(a), d, e) = e(a, natrec(a, d, e)) : C(succ(a))$$

i.e. for all what we said during the previous points:

$$\begin{cases} natrec(succ(a_0), d_0, e_0) = e_0(a_0, natrec(a_0, d_0, e_0)) : C_0(c_0) & (3.8a) \\ natrec(succ(a_1), d_1, e_1) = e_1(a_1, natrec(a_1, d_1, e_1)) : C_1(c_1) & (3.8b) \\ idpeel(idpeel(a_r, (x)r(succ(x))), (x)natrec(x, d_r, g)) = & (3.8c) \\ e_r(a_0, a_1, a_r, natrec(a_0, d_0, e_0), natrec(a_1, d_1, e_1), idpeel(a_r, (x)natrec(x, d_r, g))) : \\ \quad Q(a_0, a_1, a_r)(natrec(a_0, d_0, e_0), natrec(a_1, d_1, e_1)) \end{cases}$$

We derive readily (3.8a) and (3.8b) just by applying the $\mathbb{N}$-equality, but on the other side we are not able to derive (3.8c). The reason is that $a_r : \mathtt{Id}(\mathbb{N}, a_0, a_1)$, which we need to compute *idpeel*, is a generic element of that set, and we do not know which canonical element it will yield in its computation. Hence we cannot state the last equality in our model.

Ergo this show the failure to complete an interpretation of $\mathbb{N}$ inside this internal model.

We have now headed the first concrete problem of this model, which comes from the generality of the interpretation of the judgment $a : \mathbb{N}$, and the impossibility to argue with it inside the intentional equality.

Probably there are different ways to avoid this obstacle, but we will show in the next chapter a possible interpretation in an extended type theory that will make everything working in a natural and easier way; although I'm not asserting the impossibility to find a validation of the natural numbers type.

3.6 Interpretation of Equality type

In this section we are going to explain what could be an interpretation of the equality type (intensional), and at the same time we will try to validate it, through the interpretation of its introduction, elimination, and equality rules.

During the validation we are going to find different obstacles and we will show how to avoid them, anyway we will arrive at a dead-end problem which implies the failure of this interpretation in an internal model, like for the natural number type.

- We begin with the formation rule:

  $\texttt{Id}$ – Formation:

  $$\frac{A : set \qquad a : A \qquad b : A}{\texttt{Id}(A, a, b) : set}$$

  we try to built an interpretation of

  $$\texttt{Id}(A, a, b) : set \tag{3.9}$$

  The premises are interpreted, respectively, in the following triple:

  $$\begin{cases} A_0 : set \\ A_1 : set \\ A_r : rel(A_0, A_1) \end{cases} \qquad \begin{cases} a_0 : A_0 \\ a_1 : A_1 \\ a_r : A_r(a_0, a_1) \end{cases} \qquad \begin{cases} b_0 : A_0 \\ b_1 : A_1 \\ b_r : A_r(b_0, b_1) \end{cases}$$

  Now taking from each triple the first (respectively the second) judgment and then applying the $\texttt{Id}$-formation rule, we get:

  $$\texttt{Id}(A_0, a_0, b_0) : set$$
  $$\texttt{Id}(A_1, a_1, b_1) : set$$

  which really seem natural candidates to be the first two judgments for the interpretation we are looking for.

  At this point we shall define, what is a set of relations between them. What we would like to do is transport $a_r : A(a_0, a_1)$ and $b_r : A_r(b_0, b_1)$ into the same set, in order to be able to make a comparison between them. The matter is, at the same time, trying to keep some kind of symmetric property: for example the transportation of $a_r : A_r(a_0, a_1)$ into $A_r(b_0, b_1)$, or viceversa, doesn't sound correct, since all the construction would lose its symmetry. The symmetric property might be possibly important in a further interpretation of the concept of basic pair, when for example we would deal with all the category of the Basic Picture.

So what I suggest to do is the following idea:

(a) trying to define a set of the type $A_r(x_0, y_0) \quad [x_0 : A_0, x_1 : A_1]$. So what we need are two elements respectively inside A_0 and A_1

(b) then trying to transport a_r and b_r into this set, in a symmetric way.

In order to solve the first problem the best thing to do is to use the other information we have: $c_0 : \mathtt{Id}(A_0, a_0, b_0)$ and $c_1 : \mathtt{Id}(A_1, a_1, b_1)$. A way to do this is using the Id-elimination rule, so we can to introduce the following two elements:

$$\frac{\begin{array}{c} a_0, b_0 : A_0 \\ c_0 : \mathtt{Id}(A_0, a_0, b_0) \\ A_0 : set \quad [x_0, y_0 : A_0, z_0 : \mathtt{Id}(A_0, x_0, y_0)] \\ x_0 : A_0 \quad [x_0 : A_0, r(x_0) : \mathtt{Id}(A_0, x_0, x_0)] \end{array}}{idpeel(c_0, (x_0)x_0) : A_0} \ \mathtt{Id\text{-}el}$$

$$\frac{\begin{array}{c} a_1, b_1 : A_1 \\ c_1 : \mathtt{Id}(A_1, a_1, b_1) \\ A_1 : set \quad [x_1, y_1 : A_1, z_1 : \mathtt{Id}(A_1, x_1, y_1)] \\ x_1 : A_1 \quad [x_1 : A_1, r(x_1) : \mathtt{Id}(A_1, x_1, x_1)] \end{array}}{idpeel(c_1, (x_1)x_1) : A_1} \ \mathtt{Id\text{-}el}$$

So we can just select the set:

$$A_r(idpeel(c_0, (x_0)x_0), idpeel(c_1, (x_1)x_1)) \qquad (*)$$

as the one in which we want to transport a_r and b_r, in order to compare them.

But, how to do this?

If we want to redirect the set $A_r(a_0, a_1)$ into the set (*) we just need to do two transportation: the first one:

$$A_r(a_0, a_1) \longrightarrow A_r(\mathit{idpeel}(c_0, (x_0)x_0), a_1) \tag{3.10}$$

and the second one:

$$A_r(\mathit{idpeel}(c_0, (x_0)x_0), a_1) \longrightarrow A_r(\mathit{idpeel}(c_0, (x_0)x_0), \mathit{idpeel}(c_1, (x_1)x_1)) \tag{3.11}$$

We can make (3.10) by choosing in the `Id`-elimination rule, regarding the set A_0, the following:

$$\begin{aligned} C(x, y, z) &\equiv A_r(x, a_1) \longrightarrow A_r(\mathit{idpeel}(z, (x_0)x_0), a_1) \\ d(x) &\equiv \lambda w.w : A_r(x, a_1) \longrightarrow A_r(\mathit{idpeel}(r(x), (x_0)x_0), a_1) \end{aligned} \tag{3.12}$$

In this way we get: $\mathit{idpeel}(c_0, d) : C(a_0, b_0, c_0)$ that by the previous equivalence is:

$$\mathit{idpeel}(c_0, (x)\lambda w.w) : A_r(a_0, a_1) \longrightarrow A_r(\mathit{idpeel}(c_0, (x_0)x_0), a_1)$$

So we can define a new operator:

$$J_{c_0}(q) = \mathit{apply}(\mathit{idpeel}(c_0, (x)\lambda w.w), q) \quad [q : A_r(a_0, a_1)]$$

that it will be an element in the set $A_r(\mathit{idpeel}(c_0, (x_0)x_0), a_1)$.

A really important observation is that, the value of the operator J_{c_0} does not depend on c_0, since independently by the canonical element obtained in the computation of c_0, the operator $J_{c_0}(q)$ will keep the same value as q. So for this reason it is correct to call it a transportation operator, since what is doing is just keep an element from a set and put it inside a different one, without changing its value.

Now in order to do the (3.11) transportation we can always using `Id`-elimination, but this time regarding the set A_1, with the following equivalence:

$$\begin{cases} C(x, y, z) \equiv \\ A_r(\mathit{idpeel}(c_0, (x_0)x_0), x) \longrightarrow A_r(\mathit{idpeel}(c_0, (x_0)x_0), \mathit{idpeel}(z, (x_1)x_1) \end{cases} \tag{3.13}$$

$$\begin{cases} d(x) \equiv \lambda v.v : \\ A_r(\mathit{idpeel}(c_0, (x_0)x_0), x) \longrightarrow A_r(\mathit{idpeel}(c_0, (x_0)x_0), \mathit{idpeel}(r(x), (x_1)x_1)) \end{cases} \tag{3.14}$$

obtaining:

$$\begin{cases} \textit{idpeel}(c_1, (x)\lambda v.v) : \\ A_r(\textit{idpeel}(c_0, (x_0)x_0), a_1) \longrightarrow A_r(\textit{idpeel}(c_0, (x_0)x_0), \textit{idpeel}(c_1, (x_1)x_1)) \end{cases}$$

and defining a new operator:

$$T_{c_1}(p) = \textit{apply}(\textit{idpeel}(c_1, (x)\lambda v.v), p) \quad [p : A_r(\textit{idpeel}(c_0, (x_0)x_0), a_1)]$$

The same observation we did before for the previous operator holds also for this one. Indeed $T_{c_1}(p)$ just changes the set of the element p, but without modifying its value.

Hence collecting all these observations we can make the following assumption:

$$T_{c_1}(J_{c_0}(q)) = q \quad \forall q : A_r(a_0, a_1), c_0 : \mathtt{Id}(A_0, a_0, b_0), c_1 : \mathtt{Id}(A_1, a_1, b_1) \tag{3.15}$$

Here we have first problem since (3.15) could be true (and probably it is) in extensional type theory, but probably not in the intensional theory. The fact is that we will need it later when we are going to interpret the elimination rule. Hence in order to succeed in the interpretation of the `Id` type we have to, for this reason, consider an extensional theory of types, going against our initial intentions.

Now applying this two operators to a_r we get exactly what we were looking for:

$$T_{c_1}(J_{c_0}(a_r)) : A_r(\textit{idpeel}(c_0, (x_0)x_0), \textit{idpeel}(c_1, (x_1)x_1)) \tag{3.16}$$

With an analogous reasoning we can work with the transportation of b_r: it is sufficient just change to in the right-hand side of the definitions (3.12) and (3.13) the variable x with the variable y. In this way we will get two new similar operators $I_{c_0}(\circ)$ and $S_{c_1}(\circ)$ such that:

$$S_{c_1}(I_{c_0}(b_r)) : A_r(\textit{idpeel}(c_0, (x_0)x_0), \textit{idpeel}(c_1, (x_1)x_1)) \tag{3.17}$$

as required.

Now using (3.16) and (3.17) we could define as the set of relation between $\mathtt{Id}(A_0, a_0, b_0)$ and $\mathtt{Id}(A_1, a_1, b_1)$ the following:

$$\mathtt{Id}(A_r(\textit{idpeel}(c_0, (x_0)x_0), \textit{idpeel}(c_1, (x_1)x_1)), T_{c_1}(J_{c_0}(a_r)), S_{c_1}(I_{c_0}(b_r)))$$

under the assumption $c_0 : \mathtt{Id}(A_0, a_0, b_0)$ and $c_1 : \mathtt{Id}(A_1, a_1, b_1)$, furthermore we recall that $a_r : A_r(a_0, a_1)$ and $b_r : A_r(b_0, b_1)$. Finally this concludes our interpretation of (3.9).

- Now we check the validation of the other rules. First of all we consider the introduction rule:

  $\texttt{Id}$ – Introduction:

$$\frac{a : A}{r(a) : \texttt{Id}(A, a, a)}$$

We shall now find an interpretation, starting from the one of $a : A$ shown before, of $r(a) : \texttt{Id}(A, a, a)$ that should be of the following form:

$$\begin{cases} r_0(a_0) : \texttt{Id}(A_0, a_0, a_0) \\ r_1(a_1) : \texttt{Id}(A_1, a_1, a_1) \\ r_r(a_0, a_1, a_r) : \texttt{Id}(A_r(\mathit{idpeel}(r_0(a_0), (x_0)x_0), \mathit{idpeel}(r_1(a_1), (x_1)x_1)), \\ \quad T_{r_1(a_1)}(J_{r_0(a_0)}(a_r)), S_{r_1(a_1)}(I_{r_0(a_0)}(a_r))) \end{cases} \tag{**}$$

Easily we can put, by $\texttt{Id}$-introduction:

$$r_0(a_0) \equiv r(a_0) : \texttt{Id}(A_0, a_0, a_0)$$
$$r_1(a_1) \equiv r(a_1) : \texttt{Id}(A_1, a_1, a_1)$$

From these definitions we get immediately, by $\texttt{Id}$-equality, and by the definitions of the operators explained in the previous stage:

$$\mathit{idpeel}(r_0(a_0), (x_0)x_0) \equiv \mathit{idpeel}(r(a_0), x_0) = a_0 : A_0$$
$$\mathit{idpeel}(r_1(a_1), (x_1)x_1) \equiv \mathit{idpeel}(r(a_1), x_1) = a_1 : A_1$$
$$T_{r_1(a)}(J_{r_0(a)}(a_r)) \equiv T_{r(a_1)}(J_{r(a_0)}(a_r)) = a_r : A_r(a_0, a_1)$$
$$S_{r_1(a)}(I_{r_0(a)}(a_r)) \equiv S_{r(a_1)}(I_{r(a_0)}(a_r)) = a_r : A_r(a_0, a_1)$$

and so the last element that we are searching is in the set:

$$\texttt{Id}(A_r(\mathit{idpeel}(r_0(a_0), (x_0)x_0), \mathit{idpeel}(r_1(a_1), (x_1)x_1), a_r, a_r)$$

which for all what we said is:

$$\texttt{Id}(A_r(a_0, a_1), a_r, a_r)$$

hence we can just take:

$$r_r(a_0, a_1, a_r) \equiv r(a_r) : \texttt{Id}(A_r(a_0, a_1), a_r, a_r)$$

and finally we have concluded the introduction rule's validation.

- Now we have to validate the elimination rule, which we recall:

  $\texttt{Id}$ – Elimination:

$$\frac{\begin{array}{c} a, b : A \\ c : \texttt{Id}(A, a, b) \\ C(x, y, z) : set \quad [x, y : A, z : \texttt{Id}(A, x, y)] \\ d(x) : C(x, x, r(x)) \quad [x : A] \end{array}}{idpeel(c, d) : C(a, b, c)}$$

First of all we need to interpret the premises, and from these interpretations, we shall be able to get an interpretation for the conclusion. Well, each one of the premises will become a triple, as follows:

$$c : \texttt{Id}(A, a, b) \rightsquigarrow \begin{cases} c_0 : \texttt{Id}(A_0, a_0, b_0) \\ c_1 : \texttt{Id}(A_1, a_1, b_1) \\ c_r : \texttt{Id}(A_r(idpeel(c_0, (x_0)x_0), idpeel(c_1, (x_1)x_1)), \\ \quad T_{c_1}(J_{c_0}(a_r)), S_{c_1}(I_{c_0}(b_r))) \end{cases}$$

$C(x, y, z) : set\ [x, y : A, z : \texttt{Id}(A, x, y)]$ becomes:

$$\begin{cases} C_0(x_0, y_0, z_0) : set \quad [x_0, y_0 : A_0, z_0 : \texttt{Id}(A_0, x_0, y_0)] \\ C_1(x_1, y_1, z_1) : set \quad [x_1, y_1 : A_1, z_1 : \texttt{Id}(A_1, x_1, y_1)] \\ C_r(x_0, x_1, x_r, y_0, y_1, y_r, z_0, z_1, z_r)(v_0, v_1) : \ set \\ \quad [x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, x_1), y_0 : A_0, y_1 : A_1, y_r : A_r(y_0, y_1), \\ \quad z_0 : \texttt{Id}(A_0, x_0, y_0), z_1 : \texttt{Id}(A_1, x_1, y_1), \\ \quad z_r : \texttt{Id}(A_r(idpeel(z_0, (x_0)x_0), idpeel(z_1, (x_1)x_1)), T_{z_1}(J_{z_0}(x_r)), S_{z_1}(I_{z_0}(y_r))), \\ \quad v_0 : C_0(x_0, y_0, z_0), v_1 : C_1(x_1, y_1, z_1)] \end{cases}$$

And the last one, $d(x) : C(x, x, r(x))\ [x : A]$, gives the following triple:

$$\begin{cases} d_0(x_0) : C_0(x_0, x_0, r(x_0)) \quad [x_0 : A_0] \\ d_1(x_1) : C_1(x_1, x_1, r(x_1)) \quad [x_1 : A_1] \\ d_r(x_0, x_1, x_r) : C_r(x_0, x_1, x_r, x_0, x_1, x_r, r(x_0), r(x_1), r(x_r))(d_0(x_0), d_1(x_1)) \\ \quad [x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, x_1)] \end{cases}$$

Hence using these ones we have to construct an interpretation for the conclusion:

$$idpeel(c, d) : C(a, b, c)$$

that will be of the following form:

$$\begin{cases} k_0 : C_0(a_0, b_0, c_0) \\ k_1 : C_1(a_1, b_1, c_1) \\ k_r : C_r(a_0, a_1, a_r, b_0, b_1, b_r, c_0, c_1, c_r)(k_0, k_1) \end{cases}$$

where the triple (a_0, a_1, a_r) and (b_0, b_1, b_r) are the usual interpretations of $a : A$ and $b : A$.

Let's begin to construct k_0 and k_1. If we pick respectively the first and the second judgment from all the triples of the interpretations of the premises we get what is needed to apply the `Id`-Elimination rule, i.e.

$$\frac{\begin{array}{c} a_0, b_0 : A_0 \\ c_0 : \mathtt{Id}(A_0, a_0, b_0) \\ C_0(x_0, y_0, z_0) : set \quad [x_0, y_0 : A_0, z_0 : \mathtt{Id}(A_0, x_0, y_0)] \\ d_0(x_0) : C_0(x_0, x_0, r(x_0)) \quad [x_0 : A_0] \end{array}}{idpeel(c_0, d_0) : C_0(a_0, b_0, c_0)} \text{:Id-El}$$

$$\frac{\begin{array}{c} a_1, b_1 : A_1 \\ c_1 : \mathtt{Id}(A_1, a_1, b_1) \\ C_1(x_1, y_1, z_1) : set \quad [x_1, y_1 : A_1, z_1 : \mathtt{Id}(A_1, x_1, y_1)] \\ d_1(x_1) : C_1(x_1, x_1, r(x_1)) \quad [x_1 : A_1] \end{array}}{idpeel(c_1, d_1) : C_1(a_1, b_1, c_1)} \text{:Id-El}$$

Hence we can just put:

$$k_0 \equiv idpeel(c_0, d_0) \tag{3.18}$$
$$k_1 \equiv idpeel(c_1, d_1) \tag{3.19}$$

and we have exactly the first two judgments we were looking for.

Now it remains to construct the last judgment:

$$k_r : C_r(a_0, a_1, a_r, b_0, b_1, b_r, c_0, c_1, c_r)(k_0, k_1)$$

The idea is to work on the following:

$$d_r(x_0, x_1, x_r) : C_r(x_0, x_1, x_r, x_0, x_1, x_r, r(x_0), r(x_1), r(x_r))(d_0(x_0), d_1(x_1)) \tag{3.20}$$

in the context:

$$[x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, x_1)]$$

in such a way to be able to transport the element in (3.20) into the set

$$C_r(a_0, a_1, a_r, b_0, b_1, b_r, c_0, c_1, c_r)(k_0, k_1)$$

But, how can we do this? What we need is to find a set from which starting and then make three transportation.

- First of all we can act like we did during the formation rule: i.e. substituting the elements x_0, x_1, x_r inside (3.20) with respectively:

$$\begin{aligned} idpeel(c_0, (x_0)x_0) &: A_0 \\ idpeel(c_1, (x_1)x_1) &: A_1 \\ idpeel(c_r, (x_r)x_r) &: A_r(idpeel(c_0, (x_1)x_1), idpeel(c_1, (x_1)x_1)) \end{aligned}$$

  obtaining the following judgment

$$\begin{aligned} & d_r(idpeel(c_0, (x_0)x_0), idpeel(c_1, (x_1)x_1), idpeel(c_r, (x_r)x_r)) : \\ & \quad C_r(idpeel(c_0, (x_0)x_0), idpeel(c_1, (x_1)x_1), idpeel(c_r, (x_r)x_r), \\ & \quad\quad idpeel(c_0, (x_0)x_0), idpeel(c_1, (x_1)x_1), idpeel(c_r, (x_r)x_r), \\ & \quad\quad r(idpeel(c_0, (x_0)x_0)), r(idpeel(c_1, (x_1)x_1)), r(idpeel(c_r, (x_r)x_r))) \\ & \quad\quad (d_0(idpeel(c_0, (x_0)x_0)), d_1(idpeel(c_1, (x_1)x_1))) \end{aligned} \tag{3.21}$$

- Now starting from (3.21) we can do the first transportation: just choosing in the `Id`-elimination rule, with respect to the set A_0, in place of $C(x, y, z)$ and $d(x) : C(x, x, r(x))$ the following:

$$\begin{aligned} & C(x, y, z) \equiv \\ & \quad C_r(idpeel(z, (x_0)x_0), idpeel(c_1, (x_1)x_1), idpeel(c_r, (x_r)x_r), \\ & \quad\quad idpeel(z, (x_0)x_0), idpeel(c_1, (x_1)x_1), idpeel(c_r, (x_r)x_r), \\ & \quad\quad r(idpeel(z, (x_0)x_0)), r(idpeel(c_1, (x_1)x_1)), r(idpeel(c_r, (x_r)x_r))) \\ & \quad\quad (d_0(idpeel(z, (x_0)x_0)), d_1(idpeel(c_1, (x_1)x_1))) \\ & \quad \longrightarrow \\ & \quad C_r(x, idpeel(c_1, (x_1)x_1), idpeel(c_r, (x_r)x_r), \\ & \quad\quad y, idpeel(c_1, (x_1)x_1), idpeel(c_r, (x_r)x_r), \\ & \quad\quad z, r(idpeel(c_1, (x_1)x_1)), r(idpeel(c_r, (x_r)x_r))) \\ & \quad\quad (idpeel(z, d_0), d_1(idpeel(c_1, (x_1)x_1))) \end{aligned} \tag{3.22}$$

Note that when we consider the set $C(x, x, r(x))$ by the computation rule of *idpeel* we get exactly the same set in the right-hand side and left-hand of the arrow. So the natural choice of the element $d(x) : C(x, y, z)$ will be the identity element of this product type, and we can call it $\lambda w_1.w_1$.

So we can apply the `Id`-elimination rule with $c_0 : \mathtt{Id}(A_0, a_0, b_0)$ obtaining:

$$
\begin{aligned}
&\mathit{idpeel}(c_0, (x)\lambda w_1.w_1) : \\
&\quad C_r(\mathit{idpeel}(c_0, (x_0)x_0), \mathit{idpeel}(c_1, (x_1)x_1), \mathit{idpeel}(c_r, (x_r)x_r), \\
&\qquad \mathit{idpeel}(c_0, (x_0)x_0), \mathit{idpeel}(c_1, (x_1)x_1), \mathit{idpeel}(c_r, (x_r)x_r), \\
&\qquad r(\mathit{idpeel}(c_0, (x_0)x_0)), r(\mathit{idpeel}(c_1, (x_1)x_1)), r(\mathit{idpeel}(c_r, (x_r)x_r))) \\
&\qquad (d_0(\mathit{idpeel}(c_0, (x_0)x_0)), d_1(\mathit{idpeel}(c_1, (x_1)x_1))) \\
&\quad \longrightarrow \\
&\quad C_r(a_0, \mathit{idpeel}(c_1, (x_1)x_1), \mathit{idpeel}(c_r, (x_r)x_r), \\
&\qquad b_0, \mathit{idpeel}(c_1, (x_1)x_1), \mathit{idpeel}(c_r, (x_r)x_r), \\
&\qquad c_0, r(\mathit{idpeel}(c_1, (x_1)x_1)), r(\mathit{idpeel}(c_r, (x_r)x_r))) \\
&\qquad (\mathit{idpeel}(c_0, d_0), d_1(\mathit{idpeel}(c_1, (x_1)x_1)))
\end{aligned}
\tag{3.23}
$$

Now using the Π-elimination rules on the element of (3.23) we are able to introduce this new operator:

$$
\begin{aligned}
\mathcal{T}^1_{c_0}(q_1) \equiv &\mathit{apply}(\mathit{idpeel}(c_0, (x)\lambda w_1.w_1), q_1) : \\
&C_r(a_0, \mathit{idpeel}(c_1, (x_1)x_1), \mathit{idpeel}(c_r, (x_r)x_r), \\
&\quad b_0, \mathit{idpeel}(c_1, (x_1)x_1), \mathit{idpeel}(c_r, (x_r)x_r), \\
&\quad c_0, r(\mathit{idpeel}(c_1, (x_1)x_1)), r(\mathit{idpeel}(c_r, (x_r)x_r))) \\
&\quad (\mathit{idpeel}(c_0, d_0), d_1(\mathit{idpeel}(c_1, (x_1)x_1)))
\end{aligned}
\tag{3.24}
$$

when q_1 is an element of the left-hand arrow's side of (3.23).

This new operator is exactly the first operator that we are looking for, where instead of q_1 we can use the element in (3.21).

- Now we need in an analogous way to do another transportation, but where the new $C(x, y, z)$, in the `Id`-elimination rule, this time concerning the set A_1, it will be the following:

$$
\begin{aligned}
C(x, y, z) \equiv& \\
&C_r(a_0, idpeel(z, (x_1)x_1), idpeel(c_r, (x_r)x_r), \\
&\quad b_0, idpeel(z, (x_1)x_1), idpeel(c_r, (x_r)x_r), \\
&\quad c_0, r(idpeel(z, (x_1)x_1)), r(idpeel(c_r, (x_r)x_r))) \\
&\quad (idpeel(c_0, d_0), d_1(idpeel(z, (x_1)x_1))) \\
&\longrightarrow \\
&C_r(a_0, x, idpeel(c_r, (x_r)x_r), b_0, y, idpeel(c_r, (x_r)x_r), \\
&\quad c_0, z, r(idpeel(c_r, (x_r)x_r)))(idpeel(c_0, d_0), idpeel(z, d_1))
\end{aligned} \tag{3.25}
$$

Reasoning in the same way as before we can take as $d(x) : C(x, x, r(x))$ the identity element that we will call this time $\lambda w_2.w_2$. Hence, as before, we can introduce a new operator:

$$
\begin{aligned}
\mathcal{T}^2_{c_1}(q_2) \equiv& apply(idpeel(c_1, (x)\lambda w_2.w_2), q_2) : \\
&C_r(a_0, a_1, idpeel(c_r, (x_r)x_r), b_0, b_1, idpeel(c_r, (x_r)x_r), \\
&\quad c_0, c_1, r(idpeel(c_r, (x_r)x_r)))(idpeel(c_0, d_0), idpeel(c_1, d_1))
\end{aligned} \tag{3.26}
$$

when q_2 is an element of the left-hand arrow's side of (3.25),where instead of z we have c_1, which is the same set where $\mathcal{T}^1_{c_0}(q_1)$ belongs.

- Again we can do the last transportation, using the `Id`- elimination rule, this time with respect to the set $A_r(idpeel(c_0, (x_0)x_0), idepeel(c_1, (x_1)x_1))$ where:

$$
\begin{aligned}
C(x, y, z) \equiv& \\
&C_r(a_0, a_1, idpeel(z, (x_r)x_r), b_0, b_1, idpeel(z, (x_r)x_r), \\
&\quad c_0, c_1, r(idpeel(z, (x_r)x_r)))(idpeel(c_0, d_0), idpeel(c_1, d_1)) \\
&\longrightarrow \\
&C_r(a_0, a_1, x, b_0, b_1, y, c_0, c_1, z) \\
&\quad ((idpeel(c_0, d_0), idpeel(c_1, d_1))
\end{aligned} \tag{3.27}
$$

Also this time, reasoning in the same way as in the first point we can take as $d(x) : C(x, x, r(x))$ the identity element that we will call this time $\lambda w_3.w_3$, since the left and right-hand sides of the arrow in the set $C(x, x, r(x))$ are exactly the same. Hence as before we can introduce a new operator:

$$\begin{aligned}\mathcal{T}^3_{c_r}(q_3) \equiv & \mathit{apply}(\mathit{idpeel}(c_r, (x)\lambda w_3.w_3), q_3) : \\ & C_r(a_0, a_1, T_{c_1}(J_{c_0}(a_r)), b_0, b_1, S_{c_1}(I_{c_0}(b_r)), \\ & \quad c_0, c_1, c_r)(\mathit{idpeel}(c_0, d_0), \mathit{idpeel}(c_1, d_1))\end{aligned} \tag{3.28}$$

when q_3 is an element of the left-hand arrow's side of (3.27),where instead of z we have c_r, that is the same set where $\mathcal{T}^2_{c_1}(q_2)$ belongs. Now we can just observe that applying this operator:

$$\mathcal{T}^3_{c_r}(\mathcal{T}^2_{c_1}(\mathcal{T}^1_{c_0}(-)))$$

to the element in (3.21), i.e.

$$d_r(\mathit{idpeel}(c_0, (x_0)x_0), \mathit{idpeel}(c_1, (x_1)x_1), \mathit{idpeel}(c_r, (x_r)x_r))$$

we get an element in the set:

$$C_r(a_0, a_1, T_{c_1}(J_{c_0}(a_r)), b_0, b_1, S_{c_1}(I_{c_0}(b_r)), c_0, c_1, c_r)(\mathit{idpeel}(c_0, d_0), \mathit{idpeel}(c_1, d_1)) \tag{3.29}$$

At this point arises the necessity to use the assumption (3.15) stated before, since we would assert the following equalities:

$$\begin{aligned}T_{c_1}(J_{c_0}(a_r)) &= a_r \\ S_{c_1}(I_{c_0}(b_r)) &= b_r\end{aligned}$$

but, as explained above, (3.15) is true probably only in an extensional type theory. Anyway assuming it true, it is enough to note that (3.29) is the same as:

$$C_r(a_0, a_1, a_r, b_0, b_1, b_r, c_0, c_1, c_r)(\mathit{idpeel}(c_0, d_0), \mathit{idpeel}(c_1, d_1))$$

that is exactly what we wanted.

Finally we have complete the validation of the `Id`-elimination rule, using as interpretation of $\mathit{idpeel}(c, d) : C(a, b, c)$, the following triple:

$$\begin{cases}\mathit{idpeel}(c_0, d_0) : C_0(a_0, b_0, c_0) \\ \mathit{idpeel}(c_1, d_1) : C_1(a_1, b_1, c_1) \\ \mathcal{T}^3_{c_r}(\mathcal{T}^2_{c_1}(\mathcal{T}^1_{c_0}(d_r(\mathit{idpeel}(c_0, (x_0)x_0), \mathit{idpeel}(c_1, (x_1)x_1), \mathit{idpeel}(c_r, (x_r)x_r))))) : \\ \quad C_r(a_0, a_1, T_{c_1}(J_{c_0}(a_r)), b_0, b_1, S_{c_1}(I_{c_0}(b_r)), c_0, c_1, c_r) \\ \quad (\mathit{idpeel}(c_0, d_0), \mathit{idpeel}(c_1, d_1))\end{cases}$$

- The `Id`-equality rule is automatically satisfied just observing that, from the validation of the previous rules, we got that the equality $idpeel(r(a), d) : C(a, a, r(a))$ is interpreted in the following triple:

$$\begin{cases} idpeel(r(a_0), d_0) : C_0(a_0, a_0, r(a_0)) \\ idpeel(r(a_1), d_1) : C_1(a_1, a_1, r(a_1)) \\ \mathcal{T}^3_{r(a_r)}(\mathcal{T}^2_{r(a_1)}(\mathcal{T}^1_{r(a_0)}(d_r(idpeel(r(a_0), (x_0)x_0), \\ idpeel(r(a_1), (x_1)x_1), idpeel(r(a_r), (x_r)x_r))))) : \\ \quad C_r(a_0, a_1, T_{r(a_1)}(J_{r(a_0)}(a_r)), a_0, a_1, S_{r(a_1)}(I_{r(a_0)}(a_r)), r(a_0), r(a_1), r(a_r)) \\ \quad (idpeel(r(a_0), d_0), idpeel(r(a_1), d_1)) \end{cases}$$

which is equivalent by `Id`-equality to:

$$\begin{cases} d_0(a_0) : C_0(a_0, a_0, r(a_0)) \\ d_1(a_1) : C_1(a_1, a_1, r(a_1)) \\ \mathcal{T}^3_{r(a_r)}(\mathcal{T}^2_{r(a_1)}(\mathcal{T}^1_{r(a_0)}(d_r(a_0, a_r, a_r)))) : \\ \quad C_r(a_0, a_1, T_{r(a_1)}(J_{r(a_0)}(a_r)), a_0, a_1, S_{r(a_1)}(I_{r(a_0)}(a_r)), r(a_0), r(a_1), r(a_r)) \\ \quad (d_0(a_0), d_1(a_1)) \end{cases}$$

and by the evaluation of all the operators we get:

$$\begin{cases} d_0(a_0) : C_0(a_0, a_0, r(a_0)) \\ d_1(a_1) :: C_1(a_1, a_1, r(a_1)) \\ d_r(a_0, a_r, a_r) : \\ \quad C_r(a_0, a_1, a_r, a_0, a_1, a_r, r(a_0), r(a_1), r(a_r))(d_0(a_0), d_1(a_1)) \end{cases}$$

that is exactly the interpretation of $d(a) : C(a, a, r(a))$, as required.

Ergo this conclude the validation of `Id`-type.

We want just claim, another time, that this interpretation is valid only assuming the assumption (3.15), and this is probably true only in an extensional type theory. So since our initial purpose was to stay inside an intensional theory this can be consider like a failure, and a problem to solve in another way.

A possible different solution, in order to stay inside the intensional type theory, might be obtained using the result contained in [Streicher, 1993], like for example the K-axiom, which is a computational axiomatization of uniqueness of identity proof-objects (UIP in [Palmgren, 2012]).

Anyway as stated in the introduction, we have just pointed out the problem giving a possible solution, even if it is not completely satisfactory.

3.7 Interpretation of the disjoint union

Concerning the disjoint union type, +, we will immediately face the main problem of defining the relations set between two generic + types. This problem regards the lack of symmetry of many possible constructions or, more generally, the necessity to give a definition by cases, making more difficult all the other interpretations of its canonical and non-canonical constants. Generally speaking, the symmetry properties for this interpretation is a very important fact since it is the base, and one of its strengths, in the development of topology in the Basic Picture [Sambin, 201]. So in this first confrontation with the basic pair is important, if not essential, try to keep as much as possible these properties, since this will facilitate, maybe afterwards, further rapprochement between the two theories. Hence, as we said in the introduction of this chapter, we just present here the problem trying to give some possible ideas on how to solve it, but without developing them.

Finally we can just point out some of these ideas arguing on the validation of the formation rule.

- $+ -$ formation:

$$\frac{A : set \qquad B : set}{A + B : set}$$

where in place of $A + B$ we can use sometimes the high order notation $+(A, B)$. From the premises we obtain these two triples:

$$A : set \rightsquigarrow \begin{cases} A_0 : set \\ A_1 : set \\ A_r(x_0, x_1) : set \quad [x_0 : A_0, x_1 : A_1] \end{cases}$$

$$B : set \rightsquigarrow \begin{cases} B_0 : set \\ B_1 : set \\ B_r(y_0, y_1) : set \quad [y_0 : B_0, y_1 : B_1] \end{cases}$$

As usual, taking the first two judgments of each triple, and applying +-formation we can readily put

$$A_0 +_0 B_0 \equiv A_0 + B_0 \tag{3.30}$$

$$A_1 +_1 B_1 \equiv A_1 + B_1 \tag{3.31}$$

Now what we have to do is define a set of relations between them, which depends on the following sets:

$$+_r (A_0, A_1, A_r, B_0, B_1, B_r) : rel(+_0(A_0, B_0), +_1(A_1, B_1)) \tag{3.32}$$

At the beginning I thought that the best choice should be something like

$$A_r(x_0, x_1) + B_r(y_0, y_1)$$

However here the problem is that if we take two elements $c_0 : A_0 +_0 B_0$ and $c_1 : A_1 +_1 B_1$ it could be that they had different canonical forms: one of the kind $inl(a_0)$ $[a_0 : A_0]$ or $inl(a_1)$ $[a_1 : A_1]$ and the other of the kind $inr(b_0)$ $[b_0 : B_0]$ or $inr(b_1)$ $[b_1 : B_1]$; in this case there shouldn't be any reason that a relation between them exists.

Hence here arises the necessity to create a definition by cases for the set (3.32) in order to distinguish the provenience of the elements c_0 and c_1, and in this way keeping the information about that. Thus some definition for (3.32), that could satisfy this necessity, could be the following:

$$\begin{cases} A_r(x_0, x_1) + B_r(y_0, y_1) & if \quad c_0, c_1 \quad \text{come from the same set} \\ \emptyset & otherwise \end{cases}$$

where, with "they come from the same set" I mean they both are of the form $inl(\circ)$ or $inr(\circ)$. Another observation is that we need to avoid the dependence by x_0, x_1, y_0, y_1, but this can be done just using the +-elimination rule, and afterwards in place of x_0 and y_0 substituting for example the element:

$$when(c_0, (x_0)x_0, (y_0)y_0)$$

and analogous yields for x_1 and y_1.

Eventually another definition for (3.32) could be:

$$\begin{cases} A_r(x_0, x_1) & if\ (\exists a_0 : A_0)(\exists a_1 : A_1) \\ & \quad (\mathtt{Id}(A_0 + B_0, c_0, inl(a_0)) \wedge (\mathtt{Id}(A_1 + B_1, c_1, inl(a_1)))) \\ B_r(y_0, y_1) & if\ (\exists b_0 : B_0)(\exists b_1 : B_1) \\ & \quad (\mathtt{Id}(A_0 + B_0, c_0, inr(b_0)) \wedge (\mathtt{Id}(A_1 + B_1, c_1, inr(b_1)))) \\ \emptyset & otherwise \end{cases}$$

Anyway definitions by cases, like these, imply the use of the first universe, which probably will make more complicated the validation of the next rules. Although the last one seems a good way in order to hold on in the validation of the disjoint union, as we said in the introduction of the chapter, we will leave to the reader's interest to complete it, since in the external model even this type will be interpreted in a natural way.

Chapter 4

The external model

The failure to build an internal model leads us to the necessity to construct new types that will solve all the problems. For this reason we will introduce in this chapter some new "star"-types which are nothing more than relation sets between two same types "without star". What we mean is that, for example, a Π^*-type it will be simply a relation set between two Π-types. As we will see, adding these new types to the standard type theory, will create a really natural model for our interpretation, and the validation of each rule will turn out in an easy way.

First of all, we will introduce $\texttt{Bool}^*, \Pi^*, \Sigma^*, \texttt{Id}^*, +^*, \mathbb{N}^*$ (we will explain only Bool since it will be analogous to a generic finite star type), by giving their formal rules, and after that we will validate the respective type "without star" in this external model.

After the validation of all the standard types we will then validate also the first universe U, inside what we can call a "first universe of relations", U^*; and even this last validation will turn out in a natural way.

Hence, the external model, obtained by these extending the usual type theory with these new star types, will solve all the problems encountered before when we tried to stay inside an internal model. Generally speaking not only the problems we encountered will be solved, but also the types that we have successfully interpreted in the internal model, in the external one their interpretations will seem more natural.

4.1 $\texttt{Bool}^*$ type

- $\texttt{Bool}^*$ – formation:

$$\texttt{Bool}^* : rel(\texttt{Bool}, \texttt{Bool})$$

- $\texttt{Bool}^*$ – introduction:

$$0^* : \texttt{Bool}^*(0,0) \qquad\qquad 1^* : \texttt{Bool}^*(1,1)$$

- $\texttt{Bool}^*$ – elimination:

$$\frac{\begin{array}{c} D(z,z',r)\ set\ [z,z' : \texttt{Bool}, r : \texttt{Bool}^*(z,z')] \\ d : D(0,0,0^*) \\ e : D(1,1,1^*) \end{array}}{case^*(z,z',r,d,e) : D(z,z',r)\ [z,z' : \texttt{Bool}, r : \texttt{Bool}^*(z,z')]}$$

- $\texttt{Bool}^*$ – equality:

$$\frac{\begin{array}{c} 0 : \texttt{Bool} \\ D(z,z',r) : set \quad [z,z' : \texttt{Bool}, r : \texttt{Bool}^*(z,z')] \\ d : D(0,0,0^*) \\ e : D(1,1,1^*) \end{array}}{case^*(0,0,0^*,d,e) = d : D(0,0,0^*)}$$

$$\frac{\begin{array}{c} 1 : \texttt{Bool} \\ D(z,z',r) : set \quad [z,z' : \texttt{Bool}, r : \texttt{Bool}^*(z,z')] \\ d : D(0,0,0^*) \\ e : D(1,1,1^*) \end{array}}{case^*(1,1,1^*,d,e) = e : D(1,1,1^*)}$$

4.1.1 The interpretation of $\texttt{Bool}$ in the model

- $\texttt{Bool}$ – formation:

$$\texttt{Bool} : set$$

in the model becomes the rule:

$$\texttt{Bool}_0 : set \qquad \texttt{Bool}_1 : set \qquad \texttt{Bool}_r : rel(\texttt{Bool}_0, \texttt{Bool}_1)$$

which is automatically validated by the rule of $\texttt{Bool}^*$ formation just putting:

$$\begin{aligned} \texttt{Bool}_0 &= \texttt{Bool} \\ \texttt{Bool}_1 &= \texttt{Bool} \\ \texttt{Bool}_r = \texttt{Bool}^* : rel(\texttt{Bool}, \texttt{Bool}) &= rel(\texttt{Bool}_0, \texttt{Bool}_1) \end{aligned}$$

- $\texttt{Bool}$ – introduction:

$$0 : \texttt{Bool} \qquad\qquad\qquad 1 : \texttt{Bool}$$

when passing to the model, these are transformed into, using the interpretation of the $\texttt{Bool}$-formation:

$$\begin{array}{lll} 0_0 : \texttt{Bool} & 0_1 : \texttt{Bool} & 0_r : \texttt{Bool}^*(0_0, 0_1) \\ 1_0 : \texttt{Bool} & 1_1 : \texttt{Bool} & 1_r : \texttt{Bool}^*(1_0, 1_1) \end{array}$$

which are automatically validated by the rule of $\texttt{Bool}^*$-introduction just using the following definitions:

$$\begin{aligned} 0_0 = 0_1 = 0 : \texttt{Bool} \\ 1_0 = 1_1 = 1 : \texttt{Bool} \end{aligned}$$

and finally put:

$$\begin{aligned} 0_r = 0^* : \texttt{Bool}^*(0, 0) \\ 1_r = 1^* : \texttt{Bool}^*(1, 1) \end{aligned}$$

- $\texttt{Bool}$ – elimination:

$$\frac{C(z) : set \quad [z : \texttt{Bool}] \qquad d : C(0) \qquad e : C(1)}{case(z, d, e) : C(z)\ [z : \texttt{Bool}]}$$

When passing to the model all the judgments are transformed into a triple of judgments; our aim is to validate the interpretation of the conclusion starting from all the judgments gotten from the interpretation of the assumptions, which are respectively:

$$\begin{cases} C_0(z_0) : set \quad [z_0 : \texttt{Bool}] \\ C_1(z_1) : set \quad [z_1 : \texttt{Bool}] \\ C_r(z_0, z_1, z_r) : rel(C_0(z_0), C_1(z_1)) \\ \quad [z_0 : \texttt{Bool}, z_1 : \texttt{Bool}, z_r : \texttt{Bool}^*(z_0, z_1)] \end{cases}$$

$$\begin{cases} d_0 : C_0(0) \\ d_1 : C_1(0) \\ d_r : C_r(0, 0, 0^*)(d_0, d_1) \end{cases}$$

$$\begin{cases} e_0 : C_0(1) \\ e_1 : C_1(1) \\ e_r : C_r(1, 1, 1^*)(e_0, e_1) \end{cases}$$

And what we need to find is a triple of elements of this kind:

$$\begin{cases} case_0(z_0, d_0, e_0) : C_0(z_0) \quad [z_0 : \texttt{Bool}] & (4.1a) \\ case_1(z_1, d_1, e_1) : C_1(z_1) \quad [z_1 : \texttt{Bool}] & (4.1b) \\ case_r(z_0, z_1, z_r, d_0, d_1, d_r, e_0, e_1, e_r) : & (4.1c) \\ \quad C_r(z_0, z_1, z_r)(case_0(z_0, d_0, e_0), case_1(z_1, d_1, e_1)) \\ \quad [z_0 : \texttt{Bool}, z_1 : \texttt{Bool}, z_r : \texttt{Bool}^*(z_0, z_1)] \end{cases}$$

We can immediately define (4.1a) and (4.1b) using the standard $\texttt{Bool}$-elimination, taking the premises we need from the interpretations above, obtaining:

$$case_0(z_0, d_0, e_0) = case(z_0, d_0, e_0) : C_0(z_0) \quad [z_0 : \texttt{Bool}]$$
$$case_1(z_1, d_1, e_1) = case(z_1, d_0, e_0) : C_1(z_1) \quad [z_1 : \texttt{Bool}]$$

In order to define (4.1c) it is sufficient to apply the $\texttt{Bool}^*$-elimination with the following premises (where the equivalence between the sets in the second and third assumption, comes immediately from the $\texttt{Bool}$-equality applied to the sets $C_0(z)$ and $C_1(z) \quad [z : \texttt{Bool}]$)

$$\frac{\begin{array}{c}\begin{cases} C_r(z_0,z_1,z_r)(case_0(z_0,d_0,e_0),\,case_1(z_1,d_1,e_1)) \\ \quad [z_0,z_1:\texttt{Bool},\,z_r:\texttt{Bool}^*(z_0,z_1)]\end{cases} \\ d_r : C_r(0,0,0^*)(case_0(0,d_0,e_0),\,case_1(0,d_1,e_1)) = C_r(0,0,0^*)(d_0,d_1) \\ e_r : C_r(1,1,1^*)(case_0(1,d_0,e_0),\,case_1(1,d_1,e_1)) = C_r(1,1,1^*)(e_0,e_1)\end{array}}{\begin{cases} case^*(z_0,z_1,z_r,d_r,e_r) : C_r(z_0,z_1,z_r)(case_0(z_0,d_0,e_0),\,case_1(z_1,d_1,e_1)) \\ \quad [z_0,z_1:\texttt{Bool},\,z_r:\texttt{Bool}^*(z_0,z_1)]\end{cases}}$$

which is validated by the rules of `Bool`*-elimination, by choosing

$$\begin{array}{c} D(z_0,z_1,z_r) = C_r(z_0,z_1,z_r)(case_0(z_0,d_0,e_0),\,case_1(z_1,d_1,e_1)) \\ [z_0,z_1:\texttt{Bool},\,r:\texttt{Bool}^*(z_0,z_1)] \\ d = d_r : D(0,0,0^*) = C_r(0,0,0^*)(h_0(0),h_1(0)) \\ e = e_r : D(1,1,1^*) = C_r(1,1,1^*)(h_0(1),h_1(1)) \end{array}$$

So to conclude we can just define:

$$\begin{array}{c} case_r(z_0,z_1,z_r,d_0,d_1,d_r,e_0,e_1,e_r) = case^*(z_0,z_1,z_r,d_r,e_r) : \\ C_r(z_0,z_1,z_r)(case_0(z_0,d_0,e_0),\,case_1(z_1,d_1,e_1)) \end{array}$$

in the usual context.

- `Bool`-equality

$$\frac{\begin{array}{c}0:\texttt{Bool}\\ C(z):set\quad [z:\texttt{Bool}]\\ d:C(0)\\ e:C(1)\end{array}}{case(0,d,e)=d:C(0)} \qquad \frac{\begin{array}{c}1:\texttt{Bool}\\ C(z):set\quad [z:\texttt{Bool}]\\ d:C(0)\\ e:C(1)\end{array}}{case(1,d,e)=e:C(1)}$$

The validation of the equality rule is an easy check which comes immediately from the above interpretation and applying, `Bool`-equality in the first two equivalences and `Bool`*-equality in the last one.

This finishes the validation of the `Bool`-rules in the model.

The construction of a generic finite type, N_n^*, and the respective validation will follow exactly the same pattern so we leave it to the reader.

4.2 Π^* type

- Π^* – formation:

$$\frac{\begin{array}{c} A\ set \\ A'\ set \\ P : rel(A, A') \\ B(x) :\ set\ [x : A] \\ B'(x') :\ set\ [x' : A'] \\ Q(x, x', p) : rel(B(x'), B'(x'))\ [x : A, x' : A', p : P(x, x')] \end{array}}{\Pi^*(A, A', P, B, B', Q) : rel(\Pi(A, B), \Pi(A', B'))}$$

- Π^* – introduction:

$$\frac{\begin{array}{c} b(x) : B(x)\ [x : A] \\ b'(x') : B'(x'); [x' : A'] \\ q(x, x', p) : Q(x, x', p)(b(x), b'(x'))\ [x : A, x' : A', p : P(x, x')] \end{array}}{\lambda^*(b, b', q) : \Pi^*(A, A', P, B, B', Q)(\lambda(b), \lambda(b'))}$$

where we are writing $\lambda^*(A, A', P, B, B', Q, b, b, q)$ for $\lambda^*(b, b', q)$, $\lambda(A, B, b)$ for $\lambda(b)$, and $\lambda(A', B', b')$ for $\lambda(b')$.

- Π^* – elimination:

$$\frac{\begin{array}{c} \text{the premises of } \Pi^*\text{-formation} \\ \begin{cases} D(z, z', r) : set \quad [z : \Pi(A, B), z' : \Pi(A', B'), \\ \quad r : \Pi^*(A, A', P, B, B', Q)(z, z')] \end{cases} \\ \begin{cases} d(f, f', q) : D(\lambda(f), \lambda(f'), \lambda^*(f, f', q)) \\ \quad [f : (x : A)B(x), f' : (x : A')B'(x), \\ \quad q : (x : A, x' : A', p : P(x, x'))Q(x, x', p)(f(x), f'(x'))] \end{cases} \end{array}}{\begin{cases} \mathit{funsplit}^*(z, z', r, d) : D(z, z', r) \\ \quad [z : \Pi(A, B), z' : \Pi(A', B'), r : \Pi^*(A, A', P, B, B', Q)(z, z')] \end{cases}}$$

- Π^* – equality:

the premises of Π^*-formation and -introduction

$$\begin{cases} D(z, z', r) : set \quad [z : \Pi(A, B), z' : \Pi(A', B'), \\ \quad r : \Pi^*(A, A', P, B, B', Q)(z, z')] \end{cases}$$

$$\frac{\begin{cases} d(f, f', q) : D(\lambda(f), \lambda(f'), \lambda^*(f, f', q)) \\ \quad [f : (x : A)B(x), f' : (x' : A')B'(x'), \\ \quad q : (x : A, x' : A', p : P(x, x'))Q(x, x', p)(f(x), f'(x'))] \end{cases}}{\begin{cases} \mathit{funsplit}^*(\lambda(b), \lambda(b'), \lambda^*(b, b', q), d) = d(b, b', q) \\ \quad : D(\lambda(b), \lambda(b'), \lambda^*(b, b', q)) \end{cases}}$$

4.2.1 The interpretation of Π in the model

- Π – formation:

$$\frac{A : set \qquad B(x) : set\ [x : A]}{\Pi(A, B) : set}$$

in the model the assumptions become:

$$A : set \rightsquigarrow \begin{cases} A_0 : set \\ A_1 : set \\ A_r : rel(A_0, A_1) \end{cases}$$

$$B(x) : set \quad [x : A] \rightsquigarrow \begin{cases} B_0(x_0) : set \quad [x_0 : A_0] \\ B_1(x_1) : set \quad [x_1 : A_1] \\ B_r(x_0, x_1, x_r) : rel(B_0(x_0), B_1(x_1)) \\ \quad [x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, x_1)] \end{cases}$$

Starting from these we need to find an interpretation of $\Pi(A, B)$, i.e. a triple of this kind:

$$\begin{cases} \Pi_0(A, B) : set & (4.2a) \\ \Pi_1(A, B) : set & (4.2b) \\ \Pi_r(A, B) : rel(\Pi_0(A, B), \Pi_0(A, B)) & (4.2c) \end{cases}$$

Taking the first judgment (respectively the second one) from interpretations of the assumptions, we can define immediately (4.2a) and (4.2b), by applying Π-formation:

$$\Pi_0(A, B) = \Pi(A_0, B_0) \tag{4.3}$$
$$\Pi_1(A, B) = \Pi(A_1, B_1) \tag{4.4}$$

Now to obtain (4.2c) it is enough to apply Π^*-formation with the following premises:

$$\frac{\begin{array}{c} A_0 : set \\ A_1 : set \\ A_r : rel(A_0, A_1) \\ B_0(x_0) : set\ [x_0 : A_0] \\ B_1(x_1) : set\ [x_1 : A_1] \\ \begin{cases} B_r(x_0, x_1, x_r) : rel(B_0(x_0), B_1(x_1)) \\ \quad [x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, x_1)] \end{cases} \end{array}}{\Pi^*(A_0, A_1, A_r, B_0, B_1, B_r) : rel(\Pi(A_0, B_0), \Pi(A_1, B_1))}$$

in which substituting (4.3) and (4.4) we get exactly the set of relations that we were looking for. So it suffices to put:

$$\Pi_r(A, B) = \Pi^*(A_0, A_1, A_r, B_0, B_1, B_r)$$

- Π – introduction:

$$\frac{b(x) : B(x)\ [x : A]}{\lambda(b) : \Pi(A, B)}$$

when passing to the model the premises is transformed into:

$$b(x) : B(x) \quad [x : A] \rightsquigarrow \begin{cases} b_0 : B_0(x_0) \quad [x_0 : A_0] \\ b_1 : B_1(x_1) \quad [x_1 : A_1] \\ b_r(x_0, x_1, x_r) : B_r(x_0, x_1, x_r)(b_0(x_0), b_1(x_1)) \\ \quad [x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, x_1)] \end{cases}$$

And the interpretation of the conclusion will be:

$$\lambda(b) : \Pi(A, B) \rightsquigarrow \begin{cases} \lambda(b_0) : \Pi(A_0, B_0) \\ \lambda(b_1) : \Pi(A_1, B_1) \\ \lambda^*(b_0, b_1, b_r) : \Pi^*(A_0, A_1, A_r, B_0, B_1, B_r)(\lambda(b_0), \lambda(b_1)) \end{cases}$$

where the first two are readily obtained from the standard Π-introduction, starting from the first two judgments in the interpretation of the premise. Furthermore the last one is immediately validated by applying the Π^*-introduction with the following assumptions:

$$\frac{\begin{array}{c} b_0(x_0) : B_0(x_0)\ [x_0 : A_0] \\ b_1(x_1) : B_1(x_1)\ [x_1 : A_1] \\ \begin{cases} b_r(x_0, x_1, x_r) : B_r(x_0, x_1, x_r)(b_0(x_0), b_1(x_1)) \\ \quad [x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, x_1)] \end{cases} \end{array}}{\lambda^*(b_0, b_1, b_r) : \Pi^*(A_0, A_1, A_r, B_0, B_1, B_r)(\lambda(b_0), \lambda(b_1))}\ \Pi^*\text{-intro}$$

- Π – elimination:

$$\frac{C(z)\ set\ [z : \Pi(A, B)] \qquad c(f) : C(\lambda(f))\ [f : (x : A)B(x)]}{funsplit(z, c) : C(z)\ [z : \Pi(A, B)]}$$

When passing to the model the premises are transformed respectively into

$$\begin{cases} C_0(z_0) : set \quad [z_0 : \Pi(A_0, B_0)] \\ C_1(z_1) : set \quad [z_1 : \Pi(A_1, B_1)] \\ C_r(z_0, z_1, z_r) : rel(C_0(z_0), C_1(z_1)) \\ \quad [z_0 : \Pi(A_0, B_0), z_1 : \Pi(A_1, B_1), z_r : \Pi^*(A_0, A_1, A_r, B_0, B_1, B_r)] \end{cases}$$

$$\begin{cases} c_0(f_0) : C_0(\lambda(f_0)) \quad [f_0 : (x_0 : A_0)B_0(x_0)] \\ c_1(f_1) : C_1(\lambda(f_1)) \quad [f_0 : (x_0 : A_0)B_0(x_0)] \\ c_r(f_0, f_1, f_r) : C_r(\lambda(f_0), \lambda(f_1), \lambda^*(f_0, f_1, f_r))(c_0(f_0), c_1(f_1)) \\ \quad [f_0 : (x_0 : A_0)B_0(x_0), f_1 : (x_1 : A_1)B_1(x_1), \\ \quad f_r : (x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, x_r))B_r(x_0, x_1, x_r)(f_0(x_0), f_1(x_1))] \end{cases}$$

Starting from these we need to find an interpretation for the conclusion:

$$\begin{cases} funsplit_0(z_0, c_0) : C_0(z_0) \quad [z_0 : \Pi(A_0, B_0)] & (4.5a) \\ funsplit_1(z_1, c_1) : C_1(z_1) \quad [z_1 : \Pi(A_1, B_1)] & (4.5b) \\ funsplit_r(z_0, z_1, z_r, c_0, c_1, c_r) : & (4.5c) \\ \quad C_r(z_0, z_1, z_r)(funsplit_0(z_0, c_0), funsplit_1(z_1, c_1)) & \\ \quad [z_0 : \Pi(A_0, B_0), z_1 : \Pi(A_1, B_1), z_r : \Pi^*(A_0, A_1, A_r, B_0, B_1, B_r)] & \end{cases}$$

We can use readily the following definition, just applying the Π-elimination:

$$\begin{array}{ll} \mathit{funsplit}_0(z_0, c_0) \equiv \mathit{funsplit}(z_0, c_0) : C_0(z_0) & [z_0 : \Pi(A_0, B_0)] \\ \mathit{funsplit}_1(z_1, c_1) \equiv \mathit{funsplit}(z_1, c_1) : C_1(z_1) & [z_1 : \Pi(A_1, B_1)] \end{array}$$

In order to get (4.5c) we need to apply the Π^*-elimination by choosing in the assumptions:

$$\begin{aligned} D(z_0, z_1, z_r) =& C_r(z_0, z_1, z_r)(\mathit{funsplit}(z_0, c_0), \mathit{funsplit}(z_1, c_1)) \\ & [z_0 : \Pi(A_0, B_0), z_1 : \Pi(A_1, B_1), z_r : \Pi^*(A_0, A_1, A_r, B_0, B_1, B_r)] \\ d(f_0, f_1, f_r) =& c_r(f_0, f_1, f_r) : \\ & C_r(\lambda(f_0), \lambda(f_1), \lambda^*(f_0, f_1, f_r))(c_0(f_0), c_1(f_1)) \\ =& C_r(\lambda(f_0), \lambda(f_1), \lambda^*(f_0, f_1, f_r))(\mathit{funsplit}(\lambda(f_0), c_0), \mathit{funsplit}(\lambda(f_0), c_1)) \\ =& D(\lambda(f_0), \lambda(f_1), \lambda^*(f_0, f_1, f_r)) \\ & [f_0 : (x_0 : A_0)B_0(x_0), f_1 : (x_1 : A_1)B_1(x_1), \\ & f_r : (x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, x_r))B_r(x_0, x_1, x_r)(f_0(x_0), f_1(x_1))] \end{aligned}$$

Hence we can use the following rule:

$$\dfrac{\begin{array}{l} \left\{\begin{array}{l} C_r(z_0, z_1, z_r) : rel(C_0(z_0), C_1(z_1)) \quad [z_0 : \Pi(A_0, B_0), \\ \quad z_1 : \Pi(A_1, B_1), z_r : \Pi^*(A_0, A_1, A_r, B_0, B_1, B_r)] \end{array}\right. \\ \left\{\begin{array}{l} c_r(f_0, f_1, f_r) : C_r(\lambda(f_0), \lambda(f_1), \lambda^*(f_0, f_1, f_r))(c_0(f_0), c_1(f_1)) \\ [f_0 : (x_0 : A_0)B_0(x_0), f_1 : (x_1 : A_1)B_1(x_1), \\ f_r : (x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, x_r))B_r(x_0, x_1, x_r)(f_0(x_0), f_1(x_1))] \end{array}\right. \end{array}}{\left\{\begin{array}{l} \mathit{funsplit}^*(z_0, z_1, z_r, c_r) : C_r(z_0, z_1, z_r)(\mathit{funsplit}(z_0, c_0), \mathit{funsplit}(z_1, c_1)) \\ \quad [z_0 : \Pi(A_0, B_0), z_1 : \Pi(A_1, B_1), \\ \quad z_r : \Pi^*(A_0, A_1, A_r, B_0, B_1, B_r)] \end{array}\right.} \ \Pi^*\text{-eli}$$

To conclude the validation of this rule it suffices to define:

$$\mathit{funsplit}_r(z_0, z_1, z_r, c_0, c_1, c_r) = \mathit{funsplit}^*(z_0, z_1, z_r, c_r)$$

since they belong to equivalent sets.

- Π – equality:

$$\frac{\begin{array}{c} b(x) : B(x)\ [x : A] \\ C(z)\ set\ [z : \Pi(A, B)] \\ c(f) : C(\lambda(f))\ [f : (x : A)B(x)] \end{array}}{\mathit{funsplit}(\lambda(b), c) = c(b) : C(\lambda(b))}$$

The validation of the equality rules follows exactly the same pattern as the elimination rule, so we leave it to the reader.

This finishes the validation of the Π-rules in the model.

4.3 Σ^* type

- Σ^* – formation:

$$\frac{\begin{array}{c} A\ set \\ A'\ set \\ P : rel(A, A') \\ B(x) : set \quad [x : A] \\ B'(x') : set \quad [x' : A'] \\ Q(x, x', p) : rel(B(x'), B'(x')) \quad [x : A, x' : A', p : P(x, x')] \end{array}}{\Sigma^*(A, A', P, B, B', Q) : rel(\Sigma(A, B), \Sigma(A', B'))}$$

- Σ^* – introduction:

$$\frac{\begin{array}{c} a : A \\ a' : A' \\ a_r : P(a, a') \\ b : B(a) \\ b' : B'(a') \\ b_r : Q(a, a', a_r)(b, b') \end{array}}{\langle a, a', a_r, b, b', b_r\rangle^* : \Sigma^*(A, A', P, B, B', Q)(\langle a, b\rangle, \langle a', b'\rangle)}$$

- Σ^* – elimination:

$$\frac{\begin{array}{c} \left\{\begin{array}{l} D(z, z', r) : set \quad [z : \Sigma(A, B), z' : \Sigma(A', B'), \\ \quad r : \Sigma^*(A, A', P, B, B', Q)(z, z')] \end{array}\right. \\ \left\{\begin{array}{l} d(x, x', p, y, y', q) : D(\langle x, y\rangle, \langle x', y'\rangle, \langle x, x', p, b, b', q\rangle^*) \\ \quad [x : A, x' : A', p : P(x, x'), y : B(x), \\ \quad y' : B'(x'), q : Q(x, x', p)(y, y')] \end{array}\right. \end{array}}{\left\{\begin{array}{l} split^*(z, z', r, d) : D(z, z', r) \quad [z : \Sigma(A, B), z' : \Sigma(A', B'), \\ \quad r : \Sigma^*(A, A', P, B, B', Q)(z, z')] \end{array}\right.}$$

- Σ^* – equality:

Introduction and formation rule's premises

$$\begin{cases} D(z, z', r) : set \quad [z : \Sigma(A, B), z' : \Sigma(A', B'), \\ \quad r : \Sigma^*(A, A', P, B, B', Q)(z, z')] \end{cases}$$

$$\frac{\begin{cases} d(x, x', p, y, y', q) : D(\langle x, y\rangle, \langle x', y'\rangle, \langle x, x', p, b, b', q\rangle^*) \\ \quad [x : A, x' : A', p : P(x, x'), y : B(x), \\ \quad y' : B'(x'), q : Q(x, x', p)(y, y')] \end{cases}}{\begin{cases} split^*(\langle a, b\rangle, \langle a', b'\rangle, \langle a, a', a_r, b, b', b_r\rangle^*) = d(a, a', a_r, b, b', b_r) : \\ \quad D(\langle a, b\rangle, \langle a', b'\rangle, \langle a, a', a_r, b, b', b_r\rangle^*) \end{cases}}$$

4.3.1 The interpretation of Σ in the model

- Σ – formation:

$$\frac{A : set \qquad B(x) : set\ [x : A]}{\Sigma(A, B) : set}$$

in the model the two premises become:

$$A : set \rightsquigarrow \begin{cases} A_0 : set \\ A_1 : set \\ A_r : rel(A_0, A_1) \end{cases}$$

$$B(x) : set \quad [x : A] \rightsquigarrow \begin{cases} B_0(x_0) : set \quad [x_0 : A_0] \\ B_1(x_1) : set \quad [x_1 : A_1] \\ B_r(x_0, x_1, x_r) : rel(B_0(x_0), B_1(x_1)) \\ \quad [x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, x_1)] \end{cases}$$

From which we can use the following rules:

$$\frac{A_0 : set \qquad B_0(x_0) : set\ [x_0 : A_0]}{\Sigma(A_0, B_0) : set}\ \Sigma\text{-form}$$

$$\frac{A_1 : set \qquad B_1(x_1) : set\ [x_1 : A_1]}{\Sigma(A_1, B_1) : set}\ \Sigma\text{-form}$$

$$\frac{\begin{array}{c} A_0\ set \\ A_1\ set \\ A_r : rel(A_0, A_1) \\ B_0(x_0) : set \quad [x_0 : A_0] \\ B_1(x_1) : set \quad [x_1 : A_1] \\ B_r(x_0, x_1, x_r) : rel(B_0(x_0), B_1(x_1)) \quad [x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, x_1)] \end{array}}{\Sigma^*(A_0, A_1, A_r, B_0, B_1, B_r) : rel(\Sigma(A_0, B_0), \Sigma(A_1, B_1))}\ \Sigma^*\text{-form}$$

Now using these rules we can give valid interpretation of the judgment

$$\Sigma(A, B) : set$$

by defining it as follow:

$$\begin{cases} \Sigma(A_0, B_0) : set \\ \Sigma(A_1, B_1) : set \\ \Sigma^*(A_0, A_1, A_r, B_0, B_1, B_r) : rel(\Sigma(A_0, B_0), \Sigma(A_1, B_1)) \end{cases}$$

- Σ – introduction:

$$\frac{a : A \qquad b : B(a)}{\langle a, b\rangle : \Sigma(A, B)}$$

when passing to the model, the assumptions are transformed into:

$$a : A \rightsquigarrow \begin{cases} a_0 : A_0 \\ a_1 : A_1 \\ a_r : A_r(a_0, a_1) \end{cases}$$

$$b : B(a) \rightsquigarrow \begin{cases} b_0 : B_0(a_0) \\ b_1 : B_1(a_1) \\ b_r : B_r(a_0, a_1, a_r)(b_0, b_1) \end{cases}$$

which immediately give us the following rules:

$$\frac{a_0 : A_0 \qquad b_0 : B_0(a_0)}{\langle a_0, b_0\rangle : \Sigma(A_0, B_0)}\ \Sigma\text{-intro}$$

$$\frac{a_1 : A_1 \qquad b_1 : B_1(a_1)}{\langle a_1, b_1\rangle : \Sigma(A_1, B_1)}\ \Sigma\text{-intro}$$

$$\frac{\begin{array}{c} a_0 : A_0 \\ a_1 : A_1 \\ a_r : A_r(a_0, a_1) \\ b_0 : B_0(a_0) \\ b_1 : B_1(a_1) \\ b_r : B_r(a_0, a_1, a_r)(b_0(a_0), b_1(a_1)) \end{array}}{\langle a_0, a_1, a_r, b_0, b_1, b_r\rangle^* : \Sigma^*(A_0, A_1, A_r, B_0, B_1, B_r)(\langle a_0, b_1\rangle, \langle a_1, b_1\rangle)}\ \Sigma^*\text{-intro}$$

Like before, using these ones, we are now able to build a valid interpretation of the judgment:

$$\langle a, b\rangle : \Sigma(A, B)$$

that is:

$$\begin{cases} \langle a_0, b_0\rangle : \Sigma(A_0, B_0) \\ \langle a_1, b_1\rangle : \Sigma(A_1, B_1) \\ \langle a_0, a_1, a_r, b_0, b_1, b_r\rangle^* : \Sigma^*(A_0, A_1, A_r, B_0, B_1, B_r)(\langle a_0, b_1\rangle, \langle a_1, b_1\rangle) \end{cases}$$

- Σ – elimination:

$$\frac{C(z) : set\ [z : \Sigma(A, B)] \qquad c(x, y) : C(\langle x, y\rangle)\ [x : A, y : B(x)]}{split(z) : C(z) \quad [z : \Sigma(A, B)]}$$

When passing to the model the premises are respectively transformed into

$$\begin{cases} C_0(z_0) : set \quad [z_0 : \Sigma(A_0, B_0)] \\ C_1(z_1) : set \quad [z_1 : \Sigma(A_1, B_1)] \\ C_r(z_0, z_1, z_r) : rel(C_0(z_0), C_1(z_1)) \\ \quad [z_0 : \Sigma(A_0, B_0), z_1 : \Sigma(A_1, B_1), z_r : \Sigma^*(A_0, A_1, A_r, B_0, B_1, B_r)] \end{cases}$$

$$\begin{cases} c_0(x_0, y_0) : C_0(\langle x_0, y_0 \rangle)\ [x_0 : A_0, y_0 : B_0(x_0)] \\ c_1(x_1, y_1) : C_1(\langle x_1, y_1 \rangle)\ [x_1 : A_1, y_1 : B_1(x_1)] \\ c_r(x_0, x_1, x_r, y_0, y_1, y_r) : \\ \quad C_r(\langle x_0, y_0 \rangle, \langle x_1, y_1 \rangle, \langle a_0, a_1, a_r, b_0, b_1, b_r \rangle^*)(c_0(x_0, y_0), c_1(x_1, y_1)) \\ \quad x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, x_r), y_0 : B_0(x_0), y_1 : B_1(x_1), \\ \quad y_r : B_r(x_0, x_1, x_r)(y_0, y_1)] \end{cases}$$

These interpretations give us the premises to apply the following rules:

$$\frac{\begin{array}{c} C_0(z_0) : set\ [z_0 : \Sigma(A_0, B_0)] \\ c_0(x_0, y_0) : C_0(\langle x_0, y_0 \rangle)\ [x_0 : A_0, y_0 : B_0(x_0)] \end{array}}{split(z_0, c_0) : C_0(z_0) \quad [z_0 : \Sigma(A_0, B_0)]}\ \Sigma\text{-eli}$$

$$\frac{\begin{array}{c} C_1(z_1) : set\ [z_1 : \Sigma(A_1, B_1)] \\ c_1(x_1, y_1) : C_1(\langle x_1, y_1 \rangle)\ [x_1 : A_1, y_1 : B_1(x_1)] \end{array}}{split(z_1, c_1) : C_1(z_1) \quad [z_1 : \Sigma(A_1, B_1)]}\ \Sigma\text{-eli}$$

$$\frac{\begin{array}{l} \begin{cases} C_r(z_0, z_1, z_r)(split(z_0, c_0), split(z_1, c_1))) \\ \quad [z_0 : \Sigma(A_0, B_0), z_1 : \Sigma(A_1, B_1), z_r : \Sigma^*(A_0, A_1, A_r, B_0, B_1, B_r)] \end{cases} \\ \begin{cases} c_r(x_0, x_1, x_r, y_0, y_1, y_r) : \\ \quad C_r(\langle x_0, y_0 \rangle, \langle x_1, y_1 \rangle, \langle a_0, a_1, a_r, b_0, b_1, b_r \rangle^*)(c_0(x_0, y_0), c_1(x_1, y_1)) \\ \quad x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, x_r), y_0 : B_0(x_0), y_1 : B_1(x_1), \\ \quad y_r : B_r(x_0, x_1, x_r)(y_0, y_1)] \end{cases} \end{array}}{\begin{cases} split^*(z_0, z_1, z_r, c_r) : C_r(z_0, z_1, z_r)(split(z_0, c_0), split(z_1, c_1))) \\ \quad [z_0 : \Sigma(A_0, B_0), z_1 : \Sigma(A_1, B_1), z_r : \Sigma^*(A_0, A_1, A_r, B_0, B_1, B_r)] \end{cases}}\ \Sigma^*\text{-eli}$$

It is sufficient to note that the last rule is validated since its second premise, using the Σ-equality in the set $\Sigma(A_0, B_0)$ and the set $\Sigma(A_1, B_1)$, is equivalent to:

$$\begin{cases} c_r(x_0, x_1, x_r, y_0, y_1, y_r) : \\ \quad C_r(\langle x_0, y_0 \rangle, \langle x_1, y_1 \rangle, \langle a_0, a_1, a_r, b_0, b_1, b_r \rangle^*)(split(\langle x_0, y_0 \rangle, c_0), split(\langle x_1, y_1 \rangle, c_1)) \\ \quad [x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, x_r), y_0 : B_0(x_0), y_1 : B_1(x_1), \\ \quad y_r : B_r(x_0, x_1, x_r)(y_0, y_1)] \end{cases}$$

So we have exactly the rule of Σ^*-elimination, by choosing

$$
\begin{aligned}
&D(z_0, z_1, z_r) = C_r(z_0, z_1, z_r)(\mathit{split}(z_0, c_0), \mathit{split}(z_1, c_1)))\\
&\quad [z_0 : \Sigma(A_0, B_0), z_1 : \Sigma(A_1, B_1), z_r : \Sigma^*(A_0, A_1, A_r, B_0, B_1, B_r)]\\
&d(x_0, x_1, x_r, y_0, y_1, y_r) = c_r(x_0, x_1, x_r, y_0, y_1, y_r) :\\
&\quad C_r(\langle x_0, y_0\rangle, \langle x_1, y_1\rangle, \langle a_0, a_1, a_r, b_0, b_1, b_r\rangle^*)(c_0(x_0, y_0), c_1(x_1, y_1))\\
&\quad = C_r(\langle x_0, y_0\rangle, \langle x_1, y_1\rangle, \langle a_0, a_1, a_r, b_0, b_1, b_r\rangle^*)(\mathit{split}(\langle x_0, y_0\rangle, c_0), \mathit{split}(\langle x_1, y_1\rangle, c_1))\\
&\quad = D(\langle x, y\rangle, \langle x', y'\rangle, \langle x, x', p, y, y', q\rangle^*)
\end{aligned}
$$

Hence we can complete the validation of the elimination rule, by putting as interpretation of the judgment

$$\mathit{split}(z, c) : C(z) \quad [z : \Sigma(A, B)]$$

the following triple:

$$
\begin{cases}
\mathit{split}(z_0, c_0) : C_0(z_0) \quad [z_0 : \Sigma(A_0, B_0)]\\
\mathit{split}(z_1, c_1) : C_1(z_1) \quad [z_1 : \Sigma(A_1, B_1)]\\
\mathit{split}^*(z_0, z_1, z_r, c_r) : C_r(z_0, z_1, z_r)(\mathit{split}(z_0, c_0), \mathit{split}(z_1, c_1)))\\
\quad [z_0 : \Sigma(A_0, B_0), z_1 : \Sigma(A_1, B_1), z_r : \Sigma^*(A_0, A_1, A_r, B_0, B_1, B_r)]
\end{cases}
$$

- Σ – equality:

$$\frac{C(z) : \mathit{set}\ [z : \Sigma(A, B)] \qquad c(x, y) : C(\langle x, y\rangle)\ [x : A, y : B(x)]}{\mathit{split}(\langle a, b\rangle, c) = c(a, b) : C(\langle a, b\rangle)}$$

for $a : A$ and $b : B(a)$.

The validation of the equality rule follows immediately from the interpretation of the non-canonical constant *split* defined during the previous point, and by the application of the Σ- and Σ^*- equality. So we leave it to the reader.

This concludes the validation of the Σ-type in the external model.

4.4 $\mathtt{Id}^*$ type

- $\mathtt{Id}^*$ – formation:

$$\frac{\begin{array}{ccc} A : set & A' : set & \begin{array}{c} P : rel(A, A') \\ p : P(a, a') \end{array} \\ a : A & a' : A' & \\ b : A & b' : A' & q : P(b, b') \end{array}}{\mathtt{Id}^*_{A,A',P}(a, a', p, b, b', q) : rel(\mathtt{Id}_A(a, b), \mathtt{Id}_{A'}(a', b'))}$$

- $\mathtt{Id}^*$ – introduction:

$$\frac{a : A \qquad a' : A' \qquad p : P(a, a')}{r^*(a, a', p) : \mathtt{Id}^*_{A,A',P}(a, a', p, a, a', p)(r(a), r(a'))}$$

- $\mathtt{Id}^*$ – elimination:

$$\frac{\begin{array}{l} \left\{\begin{array}{l} D(x, x', p, y, y', q, z, z', s) : set \\ \quad [x : A, x' : A', p : P(x, x'), y : A, y' : A', q : P(y, y'), \\ \quad z : \mathtt{Id}_A(x, y), z' : \mathtt{Id}_{A'}(x', y'), s : \mathtt{Id}^*_{A,A',P}(x, x', p, y, y', q)(z, z')] \end{array}\right. \\ \qquad \left\{\begin{array}{l} d(x, x', p) : D(x, x', p, x, x', p, r(x), r(x'), r^*(x, x', p)) \\ \quad [x : A, x' : A', p : P(x, x')] \end{array}\right. \end{array}}{\left\{\begin{array}{l} idpeel^*(z, z', s, d) : D(x, x', p, y, y', q, z, z', s) : set \\ \quad [x : A, x' : A', p : P(x, x'), y : A, y' : A', q : P(y, y'), \\ \quad z : \mathtt{Id}_A(x, y), z' : \mathtt{Id}_{A'}(x', y'), s : \mathtt{Id}^*_{A,A',P}(x, x', p, y, y', q)(z, z')] \end{array}\right.}$$

- $\mathtt{Id}^*$ – equality:

$$\text{same premises as in formation and introduction rules}$$
$$\begin{cases} D(x,x',p,y,y',q,z,z',s) : set \\ \quad [x : A, x' : A', p : P(x,x'), y : A, y' : A', q : P(y,y'), \\ \quad z : \mathtt{Id}_A(a,b), z' : \mathtt{Id}_{A'}(a',b'), s : \mathtt{Id}^*_{A,A',P}(a,a',p,b,b',q)(z,z')] \end{cases}$$
$$\frac{\begin{cases} d(x,x',p) : D(x,x',p,,x,x',p,r(x),r(x'),r^*(x,x',p)) \\ \quad [x : A, x' : A', p : P(x,x')] \end{cases}}{\begin{cases} idpeel^*(r(a),r(a'),r^*(a,a',p),d) = d(a,a',p) : \\ \quad D(a,a',p,a,a',p,r(a),r(a'),r^*(a,a',p)) \end{cases}}$$

4.4.1 The interpretation of Id in the model

- Id– formation:

$$\frac{A : set \qquad a : A \qquad b : A}{\mathtt{Id}(A,a,b) : set}$$

in the model this becomes the rules:

$$\frac{\begin{array}{ccc} A_0 : set & A_1 : set & A_r : rel(A_0,A_1) \\ a_0 : A_0 & a_1 : A_1 & a_r : A_r(a_0,a_1) \\ b_0 : A_0 & b_1 : A_1 & b_r : A_r(b_0,b_1) \end{array}}{\mathtt{Id}_r(A_0,A_1,A_r,a_0,a_1,a_r,b_0,b_1,b_r) : rel(\mathtt{Id}_0(A_0,a_0,b_0),\mathtt{Id}_1(A_1,a_1,b_1))}$$

which is validated by the rule of Id-formation, putting first of all:

$$\mathtt{Id}_0(A_0,a_0,b_0) = \mathtt{Id}(A_0,a_0,b_0)$$
$$\mathtt{Id}_1(A_1,a_1,b_1) = \mathtt{Id}(A_1,a_1,b_1)$$

and then, by Id*-formation:

$$\mathtt{Id}_r(A_0,A_1,A_r,a_0,a_1,a_r,b_0,b_1,b_r) = \mathtt{Id}^*_{A_0,A_1,A_r}(a_0,a_1,a_r,b_0,b_1,b_r)$$

which, from the previous equivalences, is exactly the following set of relations:

$$rel(\mathtt{Id}(A_0,a_0,a_1),\mathtt{Id}(A_1,a_1,b_1)) = rel(\mathtt{Id}_0(A_0,a_0,b_0),\mathtt{Id}_1(A_1,a_1,b_1))$$

as required.

- $\mathtt{Id}$ – introduction:

$$\frac{a : A}{r(a) : \mathtt{Id}(A, a, a)}$$

when passing to the model is transformed into:

$$\frac{a_0 : A_0 \qquad a_1 : A_1 \qquad a_r : A_r(a_0, a_1)}{r_r(a_0, a_1, a_r) : \mathtt{Id}_r(A_0, A_1, A_r, a_0, a_1, a_r, b_0, b_1, b_r)(r_0(a_0), r_1(a_1))}$$

So we need to define three elements which satisfy the following

$$\begin{cases} r_0(a_0) : \mathtt{Id}_0(A_0, a_0, b_0) \\ r_1(a_1) : \mathtt{Id}_1(A_1, a_1, b_1) \\ r_r(a_0, a_1, a_r) : \mathtt{Id}_r(A_0, A_1, A_r, a_0, a_1, a_r, b_0, b_1, b_r)(r_0(a_0), r_1(a_1)) \end{cases}$$

The first two ones are readily validated, using the equivalences showed before and the $\mathtt{Id}$-introduction rule, defining:

$$r_0(a_0) = r(a_0) : \mathtt{Id}_0(A_0, a_0, b_0) = \mathtt{Id}(A_0, a_0, b_0)$$
$$r_1(a_1) = r(a_1) : \mathtt{Id}_1(A_1, a_1, b_1) = \mathtt{Id}(A_1, a_1, b_1)$$

Starting from these, the last one is validated by the rule of Id^*-introduction, putting:

$$r_r(a_0, a_1, a_r) = r^*(a_0, a_1, a_r)$$

which belongs to the set:

$$\mathtt{Id}^*_{A_0, A_1, A_r}(a_0, a_1, a_r, a_0, a_1, a_r)(r(a_0), r(a_1))$$

that is equivalent to:

$$\mathtt{Id}_r(A_0, A_1, A_r, a_0, a_1, a_r, b_0, b_1, b_r)(r_0(a_0), r_1(a_1))$$

as required.

- $\mathtt{Id}$ – elimination:

$$\frac{\begin{array}{c} C(x, y, z) : set \quad [x, y : A, z : \mathtt{Id}(A, x, y)] \\ d(x) : C(x, x, r(x)) \quad [x : A] \end{array}}{idpeel(z, d) : C(x, y, z) \quad [x, y : A, z : \mathtt{Id}(A, x, y)]}$$

When passing to the model each premise is respectively transformed, using the equivalence showed before, into:

$$\begin{cases} C_0(x_0, y_0, z_0) : set \quad [x_0, y_0 : A_0, z_0 : \mathtt{Id}(A_0, x_0, y_0)] \\ C_1(x_1, y_1, z_1) : set \quad [x_1, y_1 : A_1, z_1 : \mathtt{Id}(A_1, x_1, y_1)] \\ C_r(x_0, x_1, x_r, y_0, y_1, y_r, z_0, z_1, z_r) : rel(C_0(x_0, y_0, z_0), C_1(x_1, y_1, z_1)) \\ \quad [x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, y_0), y_0 : A_0, y_1 : A_1, y_r : A_r(y_0, y_1), \\ \quad z_0 : \mathtt{Id}(A_0, x_0, y_0), z_1 : \mathtt{Id}(A_1, x_1, y_1), \\ \quad z_r : \mathtt{Id}^*_{A_0,A_1,A_r}(x_0, x_1, x_r, y_0, y_1, y_r)(z_0, z_1)] \end{cases}$$

$$\begin{cases} d_0(x_0) : C_0(r(x_0)) \quad [x_0 : A_0] \\ d_1(x_1) : C_1(r(x_1)) \quad [x_1 : A_1] \\ d_r(x_0, x_1, x_r) : \\ \quad C_r(x_0, x_1, x_r, x_0, x_1, x_r, r(x_0), r(x_1), r^*(x_0, x_1, x_r))(d_0(x_0), d_1(x_1)) \\ \quad [x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, x_1)] \end{cases}$$

The aim is now to find an interpretation for the conclusion: that should be of the following form:

$$\begin{cases} idpeel_0(z_0, d_0) : C_0(x_0, y_0, z_0) \quad [x_0, y_0 : A_0, z_0 : \mathtt{Id}(A_0, x_0, y_0)] & (4.6a) \\ idpeel_1(z_1, d_1) : C_1(x_1, y_1, z_1) \quad [x_1, y_1 : A_1, z_1 : \mathtt{Id}(A_1, x_1, y_1)] & (4.6b) \\ idpeel_r(z_0, z_1, z_r, d_0, d_1, d_r) : & (4.6c) \\ \quad C_r(x_0, x_1, x_r, y_0, y_1, y_r, z_0, z_1, z_r)(idpeel_0(z_0, d_0), idpeel_1(z_1, d_1)) \\ \quad [x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, y_0), y_0 : A_0, y_1 : A_1, y_r : A_r(y_0, y_1), \\ \quad z_0 : \mathtt{Id}(A_0, x_0, y_0), z_1 : \mathtt{Id}(A_1, x_1, y_1), \\ \quad z_r : \mathtt{Id}^*_{A_0,A_1,A_r}(x_0, x_1, x_r, y_0, y_1, y_r)(z_0, z_1)] \end{cases}$$

Just taking the first (respectively the second) judgment from each triple of the interpretation of the premises, and applying the standard $\mathtt{Id}$-elimination we get immediately the first two judgments of the interpretation of the conclusion, (4.6a),(4.6b):

$$idpeel_0(z_0, d_0) = idpeel(z_0, d_0) : C_0(x_0, y_0, z_0) \quad [x_0, y_0 : A_0, z_0 : \mathtt{Id}(A_0, x_0, y_0)]$$
$$idpeel_1(z_1, d_1) = idpeel(z_1, d_1) : C_1(x_1, y_1, z_1) \quad [x_1, y_1 : A_1, z_1 : \mathtt{Id}(A_1, x_1, y_1)]$$

To be able to get (4.6c) we need to apply the $\mathtt{Id}^*$-elimination rule, with the following premise, taken from the interpretation of the assumptions of $\mathtt{Id}$-elimination:

$$
\dfrac{\begin{cases} d_r(x_0, x_1, x_r) : \\ \quad C_r(x_0, x_1, x_r, x_0, x_1, x_r, r(x_0), r(x_1), r^*(x_0, x_1, x_r))(d_0(x_0), d_1(x_1)) \\ \quad [x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, x_1)] \end{cases}}{\begin{cases} idpeel^*(z_0, z_1, z_r, d_r) : \\ C_r(x_0, x_1, x_r, y_0, y_1, y_r, z_0, z_1, z_r)(idpeel(z_0, d_0), idpeel(z_1, d_1)) \\ \quad [x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, y_0), y_0 : A_0, y_1 : A_1, y_r : A_r(y_0, y_1), \\ \quad z_0 : \mathtt{Id}(A_0, x_0, y_0), z_1 : \mathtt{Id}(A_1, x_1, y_1), \\ \quad z_r : \mathtt{Id}^*_{A_0,A_1,A_r}(x_0, x_1, x_r, y_0, y_1, y_r)(z_0, z_1)] \end{cases}} \; Id^*\text{-eli}
$$

which is validated by the rules of $\mathtt{Id}^*$-elimination, by choosing

$$
\begin{aligned}
D(z_0, z_1, z_r) = {} & C_r(x_0, x_1, x_r, y_0, y_1, y_r, z_0, z_1, z_r)(idpeel(z_0, d_0), idpeel(z_1, d_1)) \\
& [x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, y_0), y_0 : A_0, y_1 : A_1, y_r : A_r(y_0, y_1), \\
& z_0 : \mathtt{Id}(A_0, x_0, y_0), z_1 : \mathtt{Id}(A_1, x_1, y_1), \\
& z_r : \mathtt{Id}^*_{A_0,A_1,A_r}(x_0, x_1, x_r, y_0, y_1, y_r)(z_0, z_1)] \\
d(x_0, x_1, x_r) = {} & d_r(x_0, x_1, x_r) : \\
& C_r(x_0, x_1, x_r, x_0, x_1, x_r, r(x_0), r(x_1), r^*(x_0, x_1, x_r))(d_0(x_0), d_1(x_1)) \\
& [x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, x_1)]
\end{aligned}
$$

Hence we can just complete the validation of the $\mathtt{Id}$-elimination, defining the element in (4.6c) as follows:

$$idpeel^*(z_0, z_1, z_r, d_r) = idpeel_r(z_0, z_1, z_r, d_0, d_1, d_r)$$

inside the set

$$C_r(x_0, x_1, x_r, y_0, y_1, y_r, z_0, z_1, z_r)(idpeel(z_0, d_0), idpeel(z_1, d_1))$$

in the usual context.

- $\mathtt{Id}$ – equality:

$$\frac{\begin{array}{c} a : A \\ C(x,y,z) : set \quad [x, y : A, z : \mathtt{Id}(A,x,y)] \\ d(x) : C(x,x,r(x)) \quad [x : A] \end{array}}{idpeel(r(a), d) = d(a) : C(a,a,r(a))}$$

What we need to check is that the interpretation of

$$idpeel(r(a), d) = d(a) : C(a,a,r(a))$$

yields three correct equivalences, i.e. the correctness of the following:

$$\begin{cases} idpeel_0(r_0(a), d_0) = d_0(a_0) : C_0(a_0, a_0, r_0(a)) \\ idpeel_1(r_1(a), d_1) = d_1(a_1) : C_1(a_1, a_1, r_1(a)) \\ idpeel_r(r_0(a), r_1(a), r_r(a), d_0, d_1, d_r) = d_r(a_0, a_1, a_r) : \\ \quad C_r(a_0, a_1, a_r, a_0, a_1, a_r, r_0(a), r_1(a), r_r(a)) \\ \quad (idpeel_0(r_0(a), d_0), idpeel_1(r_1(a), d_1)) \end{cases}$$

That is from all the definition given before:

$$\begin{cases} idpeel(r(a_0), d_0) = d_0(a_0) : C_0(a_0, a_0, r(a_0)) & (4.7a) \\ idpeel(r(a_1), d_1) = d_1(a_1) : C_1(a_1, a_1, r(a_1)) & (4.7b) \\ idpeel^*(r(a_0), r(a_1), r^*(a_0, a_1, a_r), d_0, d_1, d_r) = & (4.7c) \\ \quad d_r(a_0, a_1, a_r) : & \\ \quad C_r(a_0, a_1, a_r, a_0, a_1, a_r, r(a_0), r(a_1), r^*(a_0, a_1, a_r)) & \\ \quad (idpeel(r(a_0), d_0), idpeel(r(a_1), d_1)) & \end{cases}$$

We had already interpreted all the premises during the previous validations, so we can just observe that (4.7a) and (4.7b) follow immediately from the standard `Id`-equality rule, where in the premises we took respectively the first and the second judgment of the interpretations of the assumptions of `Id`-equality.

Note that now we have that the set

$$C_r(a_0, a_1, a_r, a_0, a_1, a_r, r(a_0), r(a_1), r^*(a_0, a_1, a_r))(idpeel(r(a_0), d_0), idpeel(r(a_1), d_1))$$

is equivalent to

$$C_r(a_0, a_1, a_r, a_0, a_1, a_r, r(a_0), r(a_1), r^*(a_0, a_1, a_r))(d_0(a_0)), d_1(a_1)))$$

Then to get the (4.7c) we can take the third judgment of each interpretation mentioned above (to be more precise all the ones from the interpretation of $a : A$), and then this time apply the `Id`*-equality rule:

$$\dfrac{\begin{array}{c} r(a_0) : \mathtt{Id}(A_0, a_0, a_0) \\ r(a_1) : \mathtt{Id}(A_1, a_1, a_1) \\ r^*(a_0, a_1, a_r) : \mathtt{Id}^*_{A_0,A_1,A_r}(a_0, a_1, a_r, a_0, a_1, a_r)(r(a_0), r(a_1)) \\ \begin{cases} C_r(x_0, x_1, x_r, y_0, y_1, y_r, z_0, z_1, z_r)(\mathit{idpeel}(z_0, d_0), \mathit{idpeel}(z_1, d_1)) \\ \quad [x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, y_0), y_0 : A_0, y_1 : A_1, y_r : A_r(y_0, y_1), \\ \quad z_0 : \mathtt{Id}(A_0, x_0, y_0), z_1 : \mathtt{Id}(A_1, x_1, y_1), \\ \quad z_r : \mathtt{Id}^*_{A_0,A_1,A_r}(x_0, x_1, x_r, y_0, y_1, y_r)(z_0, z_1)] \end{cases} \\ \begin{cases} d_r(x_0, x_1, x_r) : \\ \quad C_r(x_0, x_1, x_r, x_0, x_1, x_r, r(x_0), r(x_1), r^*(x_0, x_1, x_r))(d_0(x_0), d_1(x_1)) \\ \quad [x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, x_1)] \end{cases} \end{array}}{\begin{cases} \mathit{idpeel}^*(r(a_0), r(a_1), r^*(a_0, a_1, a_r), d_r) = d_r(a_0, a_1, a_r) \\ C_r(a_0, a_1, a_r, a_0, a_1, a_r, r(a_0), r(a_1), r^*(a_0, a_1, a_r)) \\ \quad (\mathit{idpeel}(r(a_0), d_0), \mathit{idpeel}(r(a_1), d_1)) = \\ C_r(a_0, a_1, a_r, a_0, a_1, a_r, r(a_0), r(a_1), r^*(a_0, a_1, a_r))(d_0(a_0), d_1(a_1)) \end{cases}} \; Id^*\text{-equ}$$

that is exactly the last equivalence we were seeking.

Ergo this concludes the validation of the $\mathtt{Id}$-type in the external model.

4.5 $+^*$ type

- $+^*$ − formation:

$$\frac{\begin{array}{ccc} A : set & A' : set & P : rel(A, A') \\ B : set & B' : set & Q : rel(B, B') \end{array}}{+^*(A, A', P, B, B', Q) : rel(+(A, B), +(A', B'))}$$

- $+^*$ − introduction:

$$\frac{a : A \qquad a' : A' \qquad a_r : P(a, a')}{i^*(a, a', a_r) : +^*(A, A', P, B, B', Q)(i(a), i(a'))}$$

$$\frac{b : B \qquad b' : B' \qquad b_r : Q(b, b')}{j^*(b, b', b_r) : +^*(A, A', P, B, B', Q)(j(b), j(b'))}$$

- $+^*$ − elimination:

$$\frac{\begin{array}{c} D(z, z', r)\ set \quad [z : +(A, B), z' : +(A', B'), r : +^*(A, A', P, B, B', Q)(z, z')] \\ d(x, x', p) : D(i(x), i(x'), i^*(x, x', p)) \quad [x : A, x' : A', p : P(x, x')] \\ e(y, y', q) : D(j(y), j(y'), j^*(y, y', q)) \quad [y : B, y' : B', q : Q(y, y')] \end{array}}{\left\{ \begin{array}{l} when^*(z, z', r, d, e) : D(z, z', r) \\ \quad [z : +(A, B), z' : +(A', B'), r : +^*(A, A', P, B, B', Q)(z, z')] \end{array} \right.}$$

- $+^*$ − equality:

$$\frac{\begin{array}{c} \text{same premises as in formation and introduction rules} \\ D(z, z', r)\ set\ [z : +(A, B), z' : +(A', B'), r : +^*(A, A', P, B, B', Q)(z, z')] \\ d(x, x', p) : D(i(x), i(x'), i^*(x, x', p)) \quad [x : A, x' : A', p : P(x, x')] \\ e(y, y', q) : D(j(y), j(y'), j^*(y, y', q)) \quad [y : B, y' : B', q : Q(y, y')] \end{array}}{\left\{ \begin{array}{l} when^*(i(a), i(a'), i^*(a, a', a_r), d, e) = d(a, a', a_r) : D(i(a), i(a'), i^*(a, a', a_r)) \\ when^*(j(b), j(b'), j^*(b, b', b_r), d, e) = e(b, b', b_r) : D(j(b), j(b'), j^*(b, b', b_r)) \end{array} \right.}$$

4.5.1 The interpretation of + in the model

- $+-$ formation:

$$\frac{A : set \qquad B : set}{+(A, B) : set}$$

in the model this becomes the rules:

$$\frac{\begin{array}{ccc} A_0 : set & A_1 : set & A_r : rel(A_0, A_1) \\ B_0 : set & B_1 : set & B_r : rel(B_0, B_1) \end{array}}{+_r(A_0, A_1, A_r, B_0, B_1, B_r) : rel(+_0(A_0, B_0), +_1(A_1, B_1))}$$

which is validated by the rule of +-formation, putting first of all:

$$+_0(A_0, B_0) = +(A_0, B_0)$$
$$+_1(A_1, B_1) = +(A_1, B_1)$$

and then, by $+^*$-formation:

$$+_r(A_0, A_1, A_r, B_0, B_1, B_r) = +^*(A_0, A_1, A_r, B_0, B_1, B_r) :$$

that is the following relation set:

$$rel(+(A_0, B_0), +(A_1, B_1))$$

which, from the previous equivalences, is exactly:

$$rel(+(A_0, B_0), +(A_1, B_1)) = rel(+_0(A_0, B_0), +_1(A_1, B_1))$$

as wanted.

- $+-$ introduction:

$$\frac{a : A}{i(a) : +(A, B)} \qquad\qquad \frac{b : B}{j(b) : +(A, B)}$$

when passing to the model, the first one is transformed into:

$$\frac{a_0 : A_0 \qquad a_1 : A_1 \qquad a_r : A_r(a_0, a_1)}{i_r(a_0, a_1, a_r) : +_r(A_0, A_1, A_r, B_0, B_1, B_r)(i_0(a_0), i_1(a_1))}$$

So we need to define three elements which satisfy the following

$$\begin{cases} i_0(a_0) : +_0(A_0, B_0) \\ i_1(a_1) : +_1(A_1, B_1) \\ i_r(a_0, a_1, a_r) : +_r(A_0, A_1, A_r, B_0, B_1, B_r)(i_0(a_0), i_1(a_1)) \end{cases}$$

The first two ones are readily validated, using the equivalences showed before and the +-introduction rule, defining:

$$i_0(a_0) = i(a_0) : +_0(A_0, B_0) = +(A_0, B_0)$$
$$i_1(a_1) = i(a_1) : +_1(A_1, B_1) = +(A_1, B_1)$$

Starting from these, the last one is validated by the rule of $+^*$-introduction, putting:

$$i_r(a_0, a_1, a_r) = i^*(a_0, a_1, a_r)$$

which belongs to the set:

$$+^*(A_0, A_1, A_r, B_0, B_1, B_r)(i(a_0), i(a_1))$$

i.e.

$$+_r(A_0, A_1, A_r, B_0, B_1, B_r)(i_0(a_0), i_1(a_1))$$

as wanted.

The validation of the second introduction rule is analogous and so we will omit it.

- $+ -$ elimination:

$$\frac{\begin{array}{c} C(z) : set \quad [z : +(A, B)] \\ d(x) : C(i(x)) \quad [x : A] \\ e(y) : C(j(y)) \quad [y : B] \end{array}}{when(z, d, e) : C(z) \; [z : +(A, B)]}$$

When passing to the model each premise is respectively transformed, using the equivalence showed before, into:

$$\begin{cases} C_0(z_0) : set \quad [z_0 : +(A_0, B_0)] \\ C_1(z_1) : set \quad [z_1 : +(A_1, B_1)] \\ C_r(z_0, z_1, z_r) : rel(C_0(z_0), C_1(z_1)) \quad [z_0 : +(A_0, B_0), z_1 : +(A_1, B_1), \\ \quad z_r : +^*(A_0, A_1, A_r, B_0, B_1, B_r)(z_0, z_1)] \end{cases}$$

$$\begin{cases} d_0(x_0) : C_0(i(x_0)) \quad [x_0 : A_0] \\ d_1(x_1) : C_1(i(x_1)) \quad [x_1 : A_1] \\ d_r(x_0, x_1, x_r) : C_r(i(x_0), i(x_1), i^*(x_0, x_1, x_r))(d_0(x_0), d_1(x_1)) \\ \quad [x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, x_1)] \end{cases}$$

$$\begin{cases} e_0(y_0) : C_0(j(y_0)) \quad [y_0 : B_0] \\ e_1(y_1) : C_1(j(y_1)) \quad [y_1 : B_1] \\ e_r(y_0, y_1, y_r) : C_r(j(y_0), j(y_1), j^*(y_0, y_1, y_r))(e_0(y_0), e_1(y_1)) \\ \quad [y_0 : B_0, y_1 : B_1, y_r : B_r(y_0, y_1)] \end{cases}$$

The aim is now to find an interpretation for the conclusion, which should be of the following form:

$$\begin{cases} when_0(z_0, d_0, e_0) : C_0(z_0) \quad [z_0 : +(A_0, B_0)] & (4.8a) \\ when_1(z_1, d_1, e_1) : C_1(z_1) \quad [z_1 : +(A_1, B_1)] & (4.8b) \\ when_r(z_0, z_1, z_r, d_0, d_1, d_r, e_0, e_1, e_r) : & (4.8c) \\ \quad C_r(z_0, z_1, z_r)(when(z_0, d_0, e_0), when(z_1, d_1, e_1)) \\ \quad [z_0 : +(A_0, B_0), z_1 : +(A_1, B_1), z_r : +^*(A_0, A_1, A_r, B_0, B_1, B_r)(z_0, z_1)] \end{cases}$$

Just taking the first (respectively the second) judgment from each triple of the interpretation of the premises, and applying the $+$-elimination we get immediately the first two judgments of the conclusion, (4.8a),(4.8b):

$$when_0(z_0, d_0, e_0) = when(z_0, d_0, e_0) : C_0(z_0) \quad [z_0 : +(A_0, B_0)]$$
$$when_1(z_1, d_1, e_1) = when(z_1, d_1, e_1) : C_1(z_1) \quad [z_1 : +(A_1, B_1)]$$

To be able to get (4.8c) we need to apply the $+^*$-elimination rule, with the following premise, taken from the interpretation of the assumptions of $+$-elimination:

$$
\frac{\begin{array}{l}
\left\{\begin{array}{l} C_r(z_0, z_1, z_r)(\mathit{when}(z_0, d_0, e_0), \mathit{when}(z_1, d_1, e_1)) \\ \quad [z_0 : +(A_0, B_0), z_1 : +(A_1, B_1), \\ \quad z_r : +^*(A_0, A_1, A_r, B_0, B_1, B_r)(z_0, z_1)] \end{array}\right. \\
\left\{\begin{array}{l} d_r(x_0, x_1, x_r) : C_r(i(x_0), i(x_1), i^*(x_0, x_1, x_r))(d_0(x_0), d_1(x_1)) \\ \quad [x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, x_1)] \end{array}\right. \\
\left\{\begin{array}{l} e_r(y_0, y_1, y_r) : C_r(j(y_0), j(y_1), j^*(y_0, y_1, y_r))(e_0(y_0), e_1(y_1)) \\ \quad [y_0 : B_0, y_1 : B_1, y_r : B_r(y_0, y_1)] \end{array}\right.
\end{array}}{\left\{\begin{array}{l} \mathit{when}^*(z_0, z_1, z_r, d_r, e_r) : C_r(z_0, z_1, z_r)(\mathit{when}(z_0, d_0, e_0), \mathit{when}(z_1, d_1, e_1)) \\ \quad [z_0 : +(A_0, B_0), z_1 : +(A_1, B_1), \\ \quad z_r : +^*(A_0, A_1, A_r, B_0, B_1, B_r)(z_0, z_1)] \end{array}\right.}
$$

which is validated by the rules of $+^*$-elimination and -equality, by choosing

$$
\begin{array}{rl}
D(z_0, z_1, z_r) = & C_r(z_0, z_1, z_r)(\mathit{when}(z_0, d_0, e_0), \mathit{when}(z_1, d_1, e_1)) \\
& [z_0 : +(A_0, B_0), z_1 : +(A_1, B_1), \\
& z_r : +^*(A_0, A_1, A_r, B_0, B_1, B_r)(z_0, z_1)] \\
d(x_0, x_1, x_r) = & d_r(x_0, x_1, x_r) : C_r(i(x_0), i(x_1), i^*(x_0, x_1, x_r))(d_0(x_0), d_1(x_1)) \\
& [x_0 : A_0, x_1 : A_1, x_r : A_r(x_0, x_1)] \\
e(y_0, y_1, y_r) = & e_r(y_0, y_1, y_r) : C_r(j(y_0), j(y_1), j^*(y_0, y_1, y_r))(e_0(y_0), e_1(y_1)) \\
& [y_0 : B_0, y_1 : B_1, y_r : B_r(y_0, y_1)]
\end{array}
$$

Hence we can just complete the validation of the $+$-elimination, defining the element in (4.8c) as follows:

$$
\begin{array}{c}
\mathit{when}^*(z_0, z_1, z_r, d_r, e_r) = \mathit{when}_r(z_0, z_1, z_r, d_0, d_1, d_r, e_0, e_1, e_r) : \\
C_r(z_0, z_1, z_r)(\mathit{when}(z_0, d_0, e_0), \mathit{when}(z_1, d_1, e_1))
\end{array}
$$

in the usual context.

- $+-$ equality:

$$\frac{\begin{array}{c} a : A \\ C(z) : set \quad [z : +(A, B)] \\ d(x) : C(i(x)) \quad [x : A] \\ e(y) : C(j(y)) \quad [y : B] \end{array}}{when(i(a), d, e) = d(a) : C(i(a))}$$

$$\frac{\begin{array}{c} b : B \\ C(z) : set \quad [z : +(A, B)] \\ d(x) : C(i(x)) \quad [x : A] \\ e(y) : C(j(y)) \quad [y : B] \end{array}}{when(j(b), d, e) = e(b) : C(j(b))}$$

We are going to validate just the first one, since the second one is similar.

What we need to check is that the interpretation of

$$when(i(a)) = d(a) : C(i(a))$$

yields three correct equivalences, i.e. correctness of the following,

$$\begin{cases} when_0(i_0(a), d_0, e_0) = d_0(a_0) : C_0(i_0(a)) \\ when_1(i_1(a), d_1, e_1) = d_1(a_1) : C_1(i_1(a)) \\ when_r(i_0(a), i_1(a), i_r(a), d_0, d_1, d_r, e_0, e_1, e_r) = d_r(a_0, a_1, a_r) : \\ \quad C_r(i_0(a), i_1(a), i_r(a))(when_0(i(a_0), d_0, e_0), when_1(i(a_1), d_1, e_1)) \end{cases}$$

That is, by all the definitions given before:

$$\begin{cases} when(i(a_0), d_0, e_0) = d_0(a_0) : C_0(i(a_0)) & (4.9a) \\ when(i(a_1), d_1, e_1) = d_1(a_1) : C_1(i(a_1)) & (4.9b) \\ when^*(i(a_0), i(a_1), i^*(a_0, a_1, a_r), d_r, e_r) = d_r(a_0, a_1, a_r) : & (4.9c) \\ \quad C_r(i(a_0), i(a_1), i^*(a_0, a_1, a_r))(when(i(a_0), d_0, e_0), when(i(a_1), d_1, e_1)) \end{cases}$$

We have already interpreted all the premises during the previous validation, so we can just observe that (4.9a) and (4.9b) come from easy applications of the standard +-elimination rule, where in the premises we take respectively the first and the second judgment of the interpretation of the assumptions of +-elimination.

Note that now we have that the set

$$C_r(i(a_0), i(a_1), i^*(a_0, a_1, a_r))(when(i(a_0), d_0, e_0), when(i(a_1), d_1, e_1))$$

is equivalent to

$$C_r(i(a_0), i(a_1), i^*(a_0, a_1, a_r))(d_0(a_0)), d_1(a_1)))$$

Then to get (4.9c) we can take the third judgment of each interpretation mentioned above (to be more precise all the ones from the interpretation of $a : A$), and then apply this time the $+^*$-elimination rule.

Ergo, this concludes the validation of the +-type in the external model.

4.6 $\mathbb{N}^*$ type

- $\mathbb{N}^*$ – formation:

$$\mathbb{N}^* : rel(\mathbb{N}, \mathbb{N})$$

- $\mathbb{N}^*$ – introduction:

$$0^* : \mathbb{N}^*(0,0) \qquad \frac{a : \mathbb{N} \qquad a' : \mathbb{N} \qquad p : \mathbb{N}^*(a, a')}{succ^*(a, a', p) : \mathbb{N}^*(succ(a), succ(a'))}$$

- $\mathbb{N}^*$ – elimination:

$$\frac{\begin{array}{c} D(z, z', r)\ set \quad [z, z' : \mathbb{N}, r : \mathbb{N}^*(z, z')] \\ d : D(0, 0, 0^*) \\ \begin{cases} e(x, x', p, y) : D(succ(x), succ(x'), succ^*(x, x', p)) \\ \quad [x, x' : \mathbb{N}, p : \mathbb{N}^*(x, x'), y : D(x, x', p)] \end{cases} \end{array}}{natrec^*(z, z', r, d, e) : D(z, z', r) \quad [z, z' : \mathbb{N}, r : \mathbb{N}^*(z, z')]}$$

- $\mathbb{N}^*$ – equality:

$$\frac{\begin{array}{c} 0^* : \mathbb{N}^*(0,0) \\ D(z, z', r) : set \quad [z, z' : \mathbb{N}, r : \mathbb{N}^*(z, z')] \\ d : D(0, 0, 0^*) \\ \begin{cases} e(x, x', p, y) : D(succ(x), succ(x'), succ^*(x, x', p)) \\ \quad [x, x' : \mathbb{N}, p : \mathbb{N}^*(x, x'), y : D(x, x', p)] \end{cases} \end{array}}{natrec^*(0, 0, 0^* d, e) = d : D(0, 0, 0^*)}$$

$$
\frac{\begin{array}{c}
a : \mathbb{N} \\
a' : \mathbb{N} \\
p : \mathbb{N}^*(a, a') \\
D(z, z', r) : set \quad [z, z' : \mathbb{N}, r : \mathbb{N}^*(z, z')] \\
d : D(0, 0, 0^*) \\
\begin{cases} e(x, x', p, y) : D(succ(x), succ(x'), succ^*(x, x', p)) \\ \quad [x, x' : \mathbb{N}, p : \mathbb{N}^*(x, x'), y : D(x, x', p)] \end{cases}
\end{array}}{\begin{cases} natrec^*(succ(a), succ(a'), succ^*(a, a', p), d, e) = \\ e(a, a', p, natrec^*(a, a', p, d, e)) : \\ \quad D(succ(a), succ(a'), succ^*(a, a', p)) \end{cases}}
$$

4.6.1 The interpretation of $\mathbb{N}$ in the model

- $\mathbb{N}$ – formation:

$$\mathbb{N} : set$$

in the model this becomes the rule:

$$\mathbb{N}_0 : set \qquad \mathbb{N}_1 : set \qquad \mathbb{N}_r : rel(\mathbb{N}_0, \mathbb{N}_1)$$

which is automatically validated by the rule of $\mathbb{N}^*$ formation just putting:

$$
\begin{gathered}
\mathbb{N}_0 = \mathbb{N} \\
\mathbb{N}_1 = \mathbb{N} \\
\mathbb{N}_r = \mathbb{N}^* : rel(\mathbb{N}, \mathbb{N}) = rel(\mathbb{N}_0, \mathbb{N}_1)
\end{gathered}
$$

- $\mathbb{N}$ – introduction:

$$0 : \mathbb{N} \qquad\qquad \frac{a : \mathbb{N}}{succ(a) : \mathbb{N}}$$

when passing to the model, the first one is transformed, using the interpretation of the $\mathbb{N}$-formation, into:

$$0_0 : \mathbb{N} \qquad 0_1 : \mathbb{N} \qquad 0_r : \mathbb{N}^*(0_0, 0_1)$$

which is readily validated by the rule of $\mathbb{N}^*$-introduction just using the following definitions:

$$0_0 = 0_1 = 0 : \mathbb{N}$$
$$0_r = 0^* : \mathbb{N}^*(0,0)$$

In order to validate the second one, we need to find an interpretation for the judgment $succ(a) : \mathbb{N}$, starting from the interpretation of $a : \mathbb{N}$:

$$a : \mathbb{N} \rightsquigarrow \begin{cases} a_0 : \mathbb{N} \\ a_1 : \mathbb{N} \\ a_r : \mathbb{N}^*(a_0, a_1) \end{cases}$$

So we can apply the $\mathbb{N}$-introduction and $\mathbb{N}^*$-introduction to obtain readily the following triple:

$$\begin{cases} succ(a_0) : \mathbb{N} \\ succ(a_1) : \mathbb{N} \\ succ^*(a_0, a_1, a_r) : \mathbb{N}^*(succ(a_0), succ(a_1)) \end{cases}$$

which, as required, is a valid interpretation for the judgment:

$$succ(a) : \mathbb{N}$$

- $\mathbb{N}$ – elimination:

$$\frac{C(z) : set\ [z : \mathbb{N}] \qquad d : C(0) \qquad e(x,y) : C(succ(x))\ [x : \mathbb{N}, y : C(x)]}{natrec(z,d,e) : C(z) \quad [z : \mathbb{N}]}$$

When passing to the model our aim is to validate the interpretation of the conclusion starting from all the judgments that we got from the interpretations of the assumptions, which are respectively:

$$\begin{cases} C_0(z_0) : set \quad [z_0 : \mathbb{N}] \\ C_1(z_1) : set \quad [z_1 : \mathbb{N}] \\ C_r(z_0, z_1, z_r) : rel(C_0(z_0), C_1(z_1)) \quad [z_0, z_1 : \mathbb{N}, z_r : \mathbb{N}^*(z_0, z_1)] \end{cases}$$

$$\begin{cases} d_0 : C_0(0) \\ d_1 : C_1(0) \\ d_r : C(0, 0, 0^*)(d_0, d_1) \end{cases}$$

$$\begin{cases} e_0(x_0, y_0) : C_0(succ(x_0)) \quad [x_0 : \mathbb{N}, y_0 : C_0(x_0)] \\ e_1(x_1, y_1) : C_1(succ(x_1)) \quad [x_1 : \mathbb{N}, y_1 : C_1(x_1)] \\ e_r(x_0, x_1, x_r, y_0, y_1, y_r) : \\ \quad C_r(succ(x_0), succ(x_1), succ^*(x_0, x_1, x_r))(e_0(x_0, y_0), e_1(x_1, y_1)) \\ \quad [x_0, x_1 : \mathbb{N}, x_r : \mathbb{N}^*(x_0, x_1), y_0 : C_0(x_0), y_1 : C_1(x_1), \\ \quad y_r : C_r(x_0, x_1, x_r)(y_0, y_1)] \end{cases}$$

And what we are looking for is a triple of elements of this form:

$$\begin{cases} natrec_0(z_0, d_0, e_0) : C_0(z_0) \quad [z_0 : \mathbb{N}] & (4.10a) \\ natrec_1(z_1, d_1, e_1) : C_1(z_1) \quad [z_1 : \mathbb{N}] & (4.10b) \\ natrec_r(z_0, z_1, z_r, d_0, d_1, d_r, e_0, e_1, e_r) : & (4.10c) \\ \quad C_r(z_0, z_1, z_r)(natrec_0(z_0, d_0, e_0), natrec_1(z_1, d_1, e_1)) \\ \quad [z_0, z_1 : \mathbb{N}, z_r : \mathbb{N}^*(z_0, z_1)] \end{cases}$$

Immediately we can define (4.10a) and (4.10b) using the standard $\mathbb{N}$-elimination, taking the premises we need from the interpretations above, obtaining:

$$natrec_0(z_0, d_0, e_0) = natrec(z_0, d_0, e_0) : C_0(z_0) \quad [z_0 : \mathbb{N}]$$
$$natrec_1(z_1, d_1, e_1) = natrec(z_1, d_1, e_1) : C_1(z_1) \quad [z_1 : \mathbb{N}]$$

Instead in order to define (4.10c) it is enough to apply the $\mathbb{N}^*$-elimination asserting the following equivalences:

$$\begin{aligned}
D(z_0, z_1, z_r) &= C_r(z_0, z_1, z_r)(natrec(z_0, d_0, e_0), natrec(z_1, d_1, e_1)) \\
&\quad [z_0, z_1 : \mathbb{N}, r : \mathbb{N}^*(z_0, z_1)] \\
d = d_r &: C_r(0, 0, 0^*)(d_0, d_1) \\
&= C_r(0, 0, 0^*)(natrec(0, d_0, e_0), natrec(0, d_1, e_1)) \\
&= D(0, 0, 0^*) \\
e(x_0, x_1, x_r, y) &= e_r(x_0, x_1, x_r, natrec(x_0, d_0, e_0), natrec(x_1, d_1, e_1), y_r) : \\
&C_r(succ(x_0), succ(x_1), succ^*(x_0, x_1, x_r)) \\
&\quad (e_0(x_0, natrec(x_0, d_0, e_0)), e_1(x_1, natrec(x_1, d_1, e_1))) \\
&= C_r(succ(x_0), succ(x_1), succ^*(x_0, x_1, x_r)) \\
&\quad (natrec(succ(x_0), d_0, e_0), natrec(succ(x_1), d_1, e_1)) \\
&= D(succ(x_0), succ(x_1), succ^*(x_0, x_1, x_r)) \\
&\quad [x_0, x_1 : \mathbb{N}, x_r : \mathbb{N}^*(x_0, x_1), \\
&\quad y_r : D(x_0, x_1, x_r) \\
&\quad\ = C_r(x_0, x_1, x_r)(natrec(x_0, d_0, e_0), natrec(x_1, d_1, e_1))]
\end{aligned}$$

The equality in these definition is easily obtained by applying the $\mathbb{N}$-equality on the sets $C_0(z)$ and $C_1(z) \quad [z : \mathbb{N}]$.

With the previous premises we are now able to validate the following $\mathbb{N}^*$-elimination rule:

$$\frac{\begin{array}{c}
\left\{\begin{array}{l} C_r(z_0, z_1, z_r)(natrec_0(z_0, d_0, e_0), natrec_1(z_1, d_1, e_1)) \\ \quad [z_0, z_1 : \mathbb{N}, z_r : \mathbb{N}^*(z_0, z_1)] \end{array}\right. \\
d_r : C_r(0, 0, 0^*)(natrec(0, d_0, e_0), natrec(0, d_1, e_1)) = C_r(0, 0, 0^*)(d_0, d_1) \\
\left\{\begin{array}{l} e_r(x_0, x_1, x_r, natrec(x_0, d_0, e_0), natrec(x_1, d_1, e_1), y_r) : \\ \quad C_r(succ(x_0), succ(x_1), succ^*(x_0, x_1, x_r)) \\ \quad (e_0(x_0, natrec(x_0, d_0, e_0)), e_1(x_1, natrec(x_1, d_1, e_1))) \\ \quad [x_0, x_1 : \mathbb{N}, x_r : \mathbb{N}^*(x_0, x_1), \\ \quad y_r : C_r(x_0, x_1, x_r)(natrec(x_0, d_0, e_0), natrec(x_1, d_1, e_1))] \end{array}\right.
\end{array}}{\left\{\begin{array}{l} natrec^*(z_0, z_1, z_r, d_r, e_r) : \\ \quad C_r(z_0, z_1, z_r)(natrec(z_0, d_0, e_0), natrec(z_1, d_1, e_1)) \\ \quad [z_0, z_1 : \mathbb{N}, z_r : \mathbb{N}^*(z_0, z_1)] \end{array}\right.}$$

Hence we can conclude this interpretation putting:

$$natrec_r(z_0, z_1, z_r, d_0, d_1, d_r, e_0, e_1, e_r) = natrec^*(z_0, z_1, z_r, d_r, e_r) :$$
$$C_r(z_0, z_1, z_r)(natrec_0(z_0, d_0, e_0), h_1(z_1, d_1, e_1))$$

in the same context as above.

- $\mathbb{N}$-equality

$$\frac{\begin{array}{c} 0 : \mathbb{N} \\ C(z) : set \quad [z : \mathbb{N}] \\ d : C(0) \\ e(x, y) : C(succ(x)) \quad [x : \mathbb{N}, y : C(x)] \end{array}}{natrec(0, d, e) = d : C(0)}$$

$$\frac{\begin{array}{c} x : \mathbb{N} \\ C(z) : set \quad [z : \mathbb{N}] \\ d : C(0) \\ e(x, y) : C(succ(x)) \quad [x : \mathbb{N}, y : C(x)] \end{array}}{natrec(succ(x), d, e) = e(x, natrec(x, d, e)) : C(succ(x))}$$

The validation of the equality rule is an easy check which comes immediately from the above interpretation and the application of $\mathbb{N}$-equality in the first two equivalences, while $\mathbb{N}^*$-equality in the last equivalence of the triples which we get transporting the conclusion into our model.

Finally this concludes the validation of the $\mathbb{N}$-rules in the model.

4.7 U^* type

We shall first introduce a set of relation U^* between two generic elements of the first universe U, which has constructors corresponding to the relation set forming operations that we defined before, i.e. $\texttt{Bool}^*, \Pi^*, \Sigma^*, \texttt{Id}^*, +^*, \mathbb{N}^*$ (we are using `Bool` since we have explained it, but it will be the same with a generic finite type).

The relation set U^* is defined, as usual, by giving its canonical elements and their equality relation. The idea is to let each canonical element represent (code) a relation set ("star"-types) formed by using the forming operations mentioned earlier. We will denote this encoding function with an hat over the standard relation set that it represents.

Simultaneously with the definition of the canonical elements, we will define a decoding function, T^*, which decodes the elements of this universe to the relation set they represent.

A problem with the set U^* is that, because of the finite star types, the number of constructors is not fixed; this makes it impossible to formulate an induction principle for U^*. Probably, following an analogous reasoning as in the book [**??**rogramming] for the first universe U, we can change the set structure avoiding the dependence of a finite number of constructors in order to justify an elimination rule for the universe U^*. One motivation for this is to introduce a selector *urec**, which is necessary for doing computations with the elements in the set of small relations sets. This it should be done by representing a generic $\mathbb{N}_n^*$ set starting from a finitely many star enumeration sets, although we are not going to develop this idea in our work.

Hence, improperly, we are going to use the decoding function as the elimination and equality rules; even if in our formulation *à la Tarski*, T^* should be seen simply as a family of set over U^*, where what we have called U^*-elimination and U^*-equality, are just its formation and introduction rules, like in [**??**TT].

- U^* – formation:

$$\mathsf{U}^* : rel(\mathsf{U}, \mathsf{U})$$

- U^* – introduction:
  - Introduction-1

$$\widehat{\mathtt{Bool}^*} : \mathsf{U}^*(\widehat{\mathtt{Bool}}, \widehat{\mathtt{Bool}})$$

– Introduction-2

$$\frac{\begin{array}{c} a : \mathsf{U} \\ a' : \mathsf{U} \\ p : \mathsf{U}^*(a, a') \\ b(x) : \mathsf{U}\ [x : T(a)] \\ b'(x') : \mathsf{U}\ [x' : T(a')] \\ q(x, x', t) : \mathsf{U}^*(b(x), b'(x'))\ [x : T(a), x' : T(a'), t : T^*(a, a', p)(x, x')] \end{array}}{\widehat{\Pi^*}(a, a', p, b, b', q) : \mathsf{U}^*(\widehat{\Pi}(a, b), \widehat{\Pi}(a', b'))}$$

– Introduction-3

$$\frac{\begin{array}{c} a : \mathsf{U} \\ a' : \mathsf{U} \\ p : \mathsf{U}^*(a, a') \\ b(x) : \mathsf{U}\ [x : T(a)] \\ b'(x') : \mathsf{U}\ [x' : T(a')] \\ q(x, x', t) : \mathsf{U}^*(b(x), b'(x'))\ [x : T(a), x' : T(a'), t : T^*(a, a', p)(x, x')] \end{array}}{\widehat{\Sigma^*}(a, a', p, b, b', q) : \mathsf{U}^*(\widehat{\Sigma}(a, b), \widehat{\Sigma}(a', b'))}$$

– Introduction-4

$$\frac{\begin{array}{ccc} a : \mathsf{U} & a' : \mathsf{U} & p : \mathsf{U}^*(a, a') \\ u : T(a) & u' : T(a') & s : T^*(a, a', p)(u, u') \\ v : T(a) & v' : T(a') & t : T^*(a, a', p)(v, v') \end{array}}{\widehat{Id^*}_{a,a',p}(u, u', s, v, v', t) : \mathsf{U}^*(\widehat{Id}_a(u, v), \widehat{Id}_{a'}(u', v'))}$$

– Introduction-5

$$\frac{\begin{array}{ccc} a : \mathsf{U} & a' : \mathsf{U} & p : \mathsf{U}^*(a, a') \\ b : \mathsf{U} & b' : \mathsf{U} & q : \mathsf{U}^*(b, b') \end{array}}{\widehat{+^*}(a, a', p, b, b', q) : \mathsf{U}^*(\widehat{+}(a, b), \widehat{+}(a', b'))}$$

– Introduction-6

$$\widehat{\mathbb{N}^*} : \mathsf{U}^*(\widehat{\mathbb{N}}, \widehat{\mathbb{N}})$$

- U^* – elimination:

$$\frac{a : \mathsf{U} \qquad a' : \mathsf{U} \qquad p : \mathsf{U}^*(a, a')}{T^*(a, a', p) : rel(T(a), T(a'))}$$

- U^* – equality:

  – Equality 1:

$$\begin{cases} T^*(\widehat{\mathtt{Bool}}, \widehat{\mathtt{Bool}}, \widehat{\mathtt{Bool}^*}) = \mathtt{Bool}^* : \\ \quad rel(\mathtt{Bool}, \mathtt{Bool}) = rel(T(\widehat{\mathtt{Bool}}), T(\widehat{\mathtt{Bool}})) \end{cases}$$

  – Equality 2:

$$\frac{\text{same premises as Intro-2}}{\begin{cases} T^*(\widehat{\Pi}(a, b), \widehat{\Pi}(a', b'), \widehat{\Pi^*}(a, a', p, b, b', q)) = \\ \Pi^*(T(a), T(a'), T^*(a, a', p), (x)T(b(x)), (x')T(b'(x')), \\ (x, x', t)T(b(x), b'(x'), q(x, x', t))) : \\ \quad rel(\Pi(T(a), (x)T(b(x))), \Pi(T(a'), (x')T(b'(x')))) = \\ \quad rel(T(\widehat{\Pi}(a, b)), T(\widehat{\Pi}(a', b'))) \end{cases}}$$

  – Equality 3:

$$\frac{\text{same premises as Intro-3}}{\begin{cases} T^*(\widehat{\Sigma}(a, b), \widehat{\Sigma}(a', b'), \widehat{\Sigma^*}(a, a', p, b, b', q)) = \\ \Sigma^*(T(a), T(a'), T^*(a, a', p), (x)T(b(x)), (x')T(b'(x')), \\ (x, x', t)T(b(x), b'(x'), q(x, x', t))) : \\ \quad rel(\Sigma(T(a), (x)T(b(x))), \Sigma(T(a'), (x')T(b'(x'))))) = \\ \quad rel(T(\widehat{\Sigma}(a, b)), T(\widehat{\Sigma}(a', b'))) \end{cases}}$$

  – Equality 4:

$$\frac{\text{same premises as Intro-4}}{\begin{cases} T^*(\widehat{Id}_a(u, v), \widehat{Id}_{a'}(u', v'), \widehat{Id^*}_{a,a',p}(u, u', s, v, v', t)) = \\ Id^*_{T(a),T(a'),T^*(a,a'p)}(u, u', s, v, v', t) : \\ \quad rel(Id_{T(a)}(u, v), Id_{T(a')}(u', v')) = \\ \quad rel(T(\widehat{Id}_a(u, v)), T(\widehat{Id}_{a'}(u', v'))) \end{cases}}$$

– Equality 5:

$$\frac{\text{same premises as Intro-5}}{\begin{cases} T^*(\widehat{+}(a,b), \widehat{+}(a',b'), \widehat{+^*}(a,a',p,b,b',q)) = \\ +^*(T(a), T(a'), T(a,a',p), T(b), T(b'), T(b,b',q)) : \\ \quad rel(+(T(a),T(b)), +(T(a'),T(b'))) = \\ \quad rel(T(\widehat{+}(a,b)), T(\widehat{+}(a',b'))) \end{cases}}$$

– Equality 6:

$$\begin{cases} T^*(\widehat{\mathbb{N}}, \widehat{\mathbb{N}}, \widehat{\mathbb{N}^*}) = \mathbb{N}^* : \\ \quad rel(\mathbb{N},\mathbb{N}) = rel(T(\widehat{\mathbb{N}}), T(\widehat{\mathbb{N}})) \end{cases}$$

4.7.1 The interpretation of U in the model

- U – formation:

$$\mathsf{U} : set$$

in the model this becomes the rule:

$$\mathsf{U}_0 : set \qquad \mathsf{U}_1 : set \qquad \mathsf{U}_r : rel(\mathsf{U}_0, \mathsf{U}_1)$$

which is immediately validated by the U^*-formation putting:

$$\mathsf{U}_0 = \mathsf{U}_1 = \mathsf{U}$$
$$\mathsf{U}_r = \mathsf{U}^*$$

- The six introduction rules of U:
  – Introduction 1:

$$\widehat{\texttt{Bool}} : \mathsf{U}$$

  – Introduction 2:

$$\frac{a : \mathsf{U} \qquad b(x) : \mathsf{U}\ [x : T(a)]}{\widehat{\Pi}(a,b) : \mathsf{U}}$$

  – Introduction 3:

$$\frac{a : \mathsf{U} \qquad b(x) : \mathsf{U}\ [x : T(a)]}{\widehat{\Sigma}(a,b) : \mathsf{U}}$$

– Introduction 4:

$$\frac{a : \mathsf{U} \qquad u : T(a) \qquad v : T(a)}{\widehat{Id}_a(u,v) : \mathsf{U}}$$

– Introduction 5:

$$\frac{a : \mathsf{U} \qquad b : \mathsf{U}}{\widehat{+}(a,b) : \mathsf{U}}$$

– Introduction 6:

$$\widehat{\mathbb{N}} : \mathsf{U}$$

The validation of these rules comes almost as an immediate consequence of the respective U^*-introduction rules. We will show the first two validations and we will leave the rest to the reader.

When passing to the model, U-introduction 1 becomes:

$$\widehat{\mathtt{Bool}}_0 : \mathsf{U} \qquad \widehat{\mathtt{Bool}}_1 : \mathsf{U} \qquad \widehat{\mathtt{Bool}}_r : \mathsf{U}^*(\widehat{\mathtt{Bool}}_0, \widehat{\mathtt{Bool}}_1)$$

and it is automatically validated by U^*-introduction 1, putting:

$$\widehat{\mathtt{Bool}}_0 = \widehat{\mathtt{Bool}}_1 = \widehat{\mathtt{Bool}} : \mathsf{U}$$
$$\widehat{\mathtt{Bool}}_r = \widehat{\mathtt{Bool}^*} : \mathsf{U}^*(\widehat{\mathtt{Bool}}_0, \widehat{\mathtt{Bool}}_1) = \mathsf{U}^*(\widehat{\mathtt{Bool}}, \widehat{\mathtt{Bool}})$$

On the other side the premises of U-introduction 2, when passing to the model, are interpreted in the following triples:

$$\begin{cases} a_0 : \mathsf{U} \\ a_1 : \mathsf{U} \\ a_r : \mathsf{U}^*(a_0, a_1) \end{cases}$$

$$\begin{cases} b_0(x_0) : \mathsf{U} \quad [x_0 : T_0(a_0)] \\ b_1(x_1) : \mathsf{U} \quad [x_1 : T_1(a_1)] \\ b_r(x_0, x_1, x_r) : \mathsf{U}^*(b_0(x_0), b_1(x_1)) \\ \quad [x_0 : T_0(a_0), x_1 : T_1(a_1), x_r : T_r(a_0, a_1, a_r)(x_0, x_1)] \end{cases}$$

Starting from these we have to find three judgments that shall be a valid interpretation for

$$\widehat{\Pi}(a, b) : \mathsf{U}$$

We will show, during the next point, that a correct interpretation of the non-canonical constant T, i.e. the triple (T_0, T_1, T_r), is obtained by defining:

$$\begin{gathered} T_0 = T_1 = T \\ T_r = T^* \end{gathered}$$

From this fact, we can apply U-introduction 2 and get:

$$\frac{a_0 : \mathsf{U} \qquad b_0(x_0) : \mathsf{U}\ [x_0 : T(a_0)]}{\widehat{\Pi}(a_0, b_0) : \mathsf{U}}$$

$$\frac{a_1 : \mathsf{U} \qquad b_1(x_1) : \mathsf{U}\ [x_1 : T(a_1)]}{\widehat{\Pi}(a_1, b_1) : \mathsf{U}}$$

which conclusions are exactly the first two judgments that we were looking for. Now, in order to get the third one, it suffices to apply the U^*-introduction 2 with the following premises:

$$\frac{\begin{gathered} a_0 : \mathsf{U} \\ a_1 : \mathsf{U} \\ a_r : \mathsf{U}^*(a, a') \\ b_0(x_0) : \mathsf{U} \quad [x_0 : T(a_0)] \\ b_1(x_1) : \mathsf{U} \quad [x_1 : T(a_1)] \\ \begin{cases} b_r(x_0, x_1, x_r) : \mathsf{U}^*(b_0(x_0), b_1(x_1)) \\ \quad [x_0 : T(a_0), x_1 : T(a_1), t : T^*(a_0, a_1, a_r)(x_0, x_1)] \end{cases} \end{gathered}}{\widehat{\Pi^*}(a_0, a_1, a_r, b_0, b_1, b_r) : \mathsf{U}^*(\widehat{\Pi}(a_0, b_0), \widehat{\Pi}(a_1, b_1))}$$

and finally this complete the validation of this rule:

$$\begin{cases} \widehat{\Pi}(a_0, b_0) : \mathsf{U} \\ \widehat{\Pi}(a_1, b_1) : \mathsf{U} \\ \widehat{\Pi^*}(a_0, a_1, a_r, b_0, b_1, b_r) : \mathsf{U}^*(\widehat{\Pi}(a_0, b_0), \widehat{\Pi}(a_1, b_1)) \end{cases}$$

- U – equality:

$$\frac{a : \mathsf{U}}{T(a) : set}$$

In order to validate this rule we have to find a triple of the following form:

$$\begin{cases} T_0(a_0) : set \\ T_1(a_1) : set \\ T_r(a_0, a_1, a_r) : rel(T_0(a_0), T_1(a_1)) \end{cases}$$

starting from:

$$\begin{cases} a_0 : \mathsf{U} \\ a_1 : \mathsf{U} \\ a_r : \mathsf{U}^*(a_0, a_1) \end{cases}$$

But this follow immediately by U^*-elimination, putting:

$$T_0 = T_1 = T$$
$$T_r = T^*$$

- The validation of the six equality rules is just an easy exercise that comes immediately by the interpretation of the respective introduction rule and the application of the respective U-equality; so we leave all of them to the reader.

Finally this concludes our interpretation.

Bibliography

[4WO, 2012] (2012). *Fourth workshop on formal topology*, Ljubljana.

[Sem, 2013] (Febraury – March 2013). *Series of lectures by Per Martin-Löf at the Stockholm University logic seminary.*

[Martin-Löf, 1984] Martin-Löf, P. (Naples, 1984). *Intuitionistic Type Theory. Notes by Giovanni Sambin of a series of lectures given in Padua, June 1980. Studies in Proof Theory 1*, 17.

[Nordström et al., 1990] Nordström, B., Petersson, K., and Smith, J. M. (1990). *Programming in Martin-Löf's type theory*, volume 200. Oxford University Press, this volume is now available at URL www.cs.chalmers.se/Cs/Research/Logic.

[Palmgren, 2012] Palmgren, E. (2012). Proof-relevance of families of setoids and identity in type theory. *Archive for mathematical logic*, 51(1-2):35–47.

[Sambin, 2002] Sambin, G. (2002). Steps towards a dynamic constructivism. *In the scope of logic, methodology and philosophy of science*, 1:261–284.

[Sambin, 2011] Sambin, G. (2011). A minimalist foundation at work. *Logic, Mathematics, Philosophy, Vintage Enthusiasms. Essays in Honour of John L. Bell*, 75:69–96.

[Sambin, 201] Sambin, G. (to appear 201-). *The Basic Picture and Positive Topology.* Clarendon Press, Oxford.

[Streicher, 1993] Streicher, T. (November 1993). *Invastigations Into Intensional Type Theory.* PhD thesis, Habilitation Thesis, Ludwig-Maximilians Universität, Munich, this is available at URL http://www.mathematik.tu-darmstadt.de/ streicher/.