

A Reflection Principle for Potential Infinite Models of Type Theory

Matthias Eberl  

LMU Munich, Germany

Abstract

Denotational models of type theory, such as set-theoretic, domain-theoretic, or category-theoretic models use (actual) infinite sets of objects in one way or another. The potential infinite, seen as an extensible finite, requires a dynamic understanding of the infinite sets of objects. It follows that the type *nat* cannot be interpreted as a set of all natural numbers, $\llbracket nat \rrbracket = \mathbb{N}$, but as an increasing family of finite sets $\mathbb{N}_i = \{0, \dots, i - 1\}$. Any reference to $\llbracket nat \rrbracket$, either by the formal syntax or by meta-level concepts, must be a reference to a (sufficiently large) set \mathbb{N}_i .

We present the basic concepts for interpreting a fragment of the simply typed λ -calculus within such a dynamic model. A type ϱ is thereby interpreted as a process, which is formally a factor system together with a limit of it. A factor system is very similar to a direct or an inverse system, and its limit is also defined by a universal property. It is crucial to recognize that a limit is not necessarily an unreachable end beyond the process. Rather, it can be regarded as an intermediate state within the factor system, which can still be extended.

The logical type *bool* plays an important role, which we interpret classically as the set $\{true, false\}$. We provide an interpretation of simply typed λ -terms in these factor systems and limits. The main result is a reflection principle, which states that an element in the limit has a “full representative” at a sufficiently large stage within the factor system. For propositions, that is, terms of type *bool*, this implies that statements about the limit are true if and only if they are true at that sufficiently large stage.

2012 ACM Subject Classification Theory of computation \rightarrow Lambda calculus; Theory of computation \rightarrow Higher order logic; Theory of computation \rightarrow Type theory; Theory of computation \rightarrow Denotational semantics

Keywords and phrases Indefinite extensibility, Potential infinite, Reflection principle

Digital Object Identifier 10.4230/LIPIcs.TYPES.2023.6

1 Introduction

In set theory, infinite sets are given by the dictum of the axiom of infinity. There is no idea of “construction” or “approximation” involved – it is a *static* concept where only existence is required, without any way to get to these sets. In contrast, consider the constructions of infinite sets as limits of direct and inverse systems. These sets are approximated and so can be understood from the perspective of a potential infinite. Moreover, they possess the structure of their approximating parts. Finally, and most importantly from a finitistic perspective, there is no necessity to “jump over” to an absolute, actual infinite limit set if all states of the system are finite. Instead, if one takes care of all the stages of one’s investigation, then a sufficiently large state within the system is sufficient and is a full substitute for an infinite limit set.

From a consequent potentialist’s point of view, it is actually a misuse of language to call a set infinite. Since the potentialist’s view is a form of finitism, and since sets are given by their extension, every set is finite in this regard. Therefore, what can be considered infinite is the type, say *nat*, whereas the set \mathbb{N} would be more accurately described as *indefinitely extensible*. Thus, the usual terminology, saying that a set is infinite, means that its type is



© Matthias Eberl;

licensed under Creative Commons License CC-BY 4.0

29th International Conference on Types for Proofs and Programs (TYPES 2023).

Editors: Delia Kesner, Eduardo Herme Reyes, and Benno van den Berg; Article No. 6; pp. 6:1–6:20



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

infinite and that the extension of the type, i.e., the set of elements of that type, is given by some *indefinitely large* state (or *sufficiently large* state) in an indefinitely extensible system of finite sets.

A system, introduced in Section 2.1, formalizes a changing totality of objects, an “open” and potentially unending process in which both the collection and its elements expand simultaneously. This process allows a (relative) completion or compactification by constructing a limit, which temporarily ends or “closes” the process.

The concept of partiality becomes relevant in extensible systems. It is a consequence of the fact that objects can be generated, or if you prefer a less constructive language, detected, so that they do not exist from the beginning. Partiality first appears in direct systems, where objects do not exist at the initial stages. In this situation, partiality is not difficult to deal with, but it becomes demanding in the function space construction, especially in the presence of higher types.

A few words about the notation. The term “iff” is an abbreviation for “if and only if”. \mathbb{N} refers to the set of natural numbers $\{0, 1, 2, \dots\}$, $\mathbb{N}^+ := \mathbb{N} \setminus \{0\}$ and $\mathbb{N}_i := \{0, \dots, i - 1\}$. We write $[A \rightarrow B]$ for the function space of all functions with domain A and codomain B , and $\mathcal{P}(\mathcal{M})$ for the power set of \mathcal{M} .

1.1 Extensibility, Coinduction and Domain Theory

A fundamental concept in the context of the potential infinite is *extensibility*. The main modes of extensibility are the creation of new objects and the creation of new knowledge about existing objects. However, we understand the latter as a differentiation of an object or an identification of several objects. In the case of differentiation, there may be multiple versions of an object at a later stage. When considering limit constructions, this is different from the understanding of the accumulation of information. To illustrate this point, let us consider inductive and coinductive definitions. These are related to adding and differentiating objects respectively, but they are not the only way.

1. The dynamic reading of an inductive definition leads to an infinite process of creating objects, related to direct systems.
2. The dynamic reading of a coinductive definition leads to an infinite process of differentiating objects, related to inverse systems.

To give an example, let Seq denote finite 0-1-sequences and Seq^∞ stand for infinite 0-1-sequences. Then the algebra with constructors $nil : \{*\} \rightarrow Seq$ and $append : Seq \times \mathbb{N}_2 \rightarrow Seq$ inductively defines the structure of Seq , whereby $\mathbb{N}_2 = \{0, 1\}$. The coinductive definition of Seq^∞ has one destructor-pair $(head, tail) : Seq^\infty \rightarrow \mathbb{N}_2 \times Seq^\infty$ and defines a coalgebra. To give these structural definitions a dynamic reading requires an index set, which will be \mathbb{N}^+ and \mathbb{N} .

In case of inductive definitions one starts with the element $*$. The states of Seq are thus $Seq_0 = \{*\}$, $Seq_1 = \{()\}$, $Seq_2 = \{(), 0, 1\}$, $Seq_3 = \{(), 0, 1, 00, 01, 10, 11\}$ and so on. The inductive definition of Seq gives rise to a direct system $(Seq_i)_{i \in \mathbb{N}^+}$ with subset inclusion as embedding. A direct system is more general than this construction by an inductive definition, as it allows for the possibility of non-injective embeddings. This corresponds to the addition and identification of objects in a single process and is relevant for a construction of quotients.

For a dynamic understanding of coinductive definitions, start with a “generic” element, say s , so $Seq_0^\infty = \{s\}$. The destructor-pair $(head, tail)$ gives $Seq_1^\infty = \{0s, 1s\}$, $Seq_2^\infty = \{00s, 01s, 10s, 11s\}$, and so on. The coinductive definition of Seq^∞ allows one to increase the knowledge about the sequence, which we understand as a process of differentiation. The

projections $proj_i^{i'} : Seq_i^\infty \rightarrow Seq_{i'}^\infty$ then remove this information gained by differentiation, e.g., $proj_1^2(01s) = 0s$. We read $01s \in Seq_2^\infty$ and $0s \in Seq_1^\infty$ as states of the same (infinite) object, differing only in the amount of information that we have about that object. The inverse system $(Seq^\infty, proj)$ corresponds to subsets $A \subseteq \mathbb{N}$ if we take $A = \{i \in \mathbb{N} \mid s_i = 1\}$ for $(s_i)_{i \in \mathbb{N}} \in Seq^\infty$.

This idea of differentiating objects is also related to Brouwer's concept of (lawlike and lawless) choice sequences within a spread – the standard text on this subject is [13]. However, our approach does not involve the constructive reasoning that is inherent in the definition of a spread.

Differentiating infinite objects differs from domain theory [1], where the ideal, infinite elements are ideals (being infinite sets) of their approximating parts, which are the compact elements. In domain theory, each approximation is seen as a different object, not as states of one object, and the ideal completion uses the idea that sets are *actual* infinite. In other words, the idea of differentiating infinite objects, as formulated in coinductive definitions and inverse systems, is not present in domain theory. Domain theory uses inductive definitions and least fixed point constructions.

Another way to get these ideal elements in domains is to think of an infinite domain as a bilimit of finite domains. A bilimit is a limit construction where direct and inverse limits coincide. One could interpret this coincidence as a reduction of inverse systems to direct systems. In the approach here, the limit construction is, roughly speaking, between the direct and the inverse limit, since the predecessor relation $\overset{p}{\mapsto}$ (which will be introduced soon) is, again roughly speaking, between the embedding emb and the projection $proj$. In Section 2.1 we will introduce such an embedding-projection pair, associated with the predecessor relation $\overset{p}{\mapsto}$. Direct and inverse systems can then be seen as extreme situations of adding and differentiating objects, while the system based on the predecessor relation $\overset{p}{\mapsto}$ is in general a combination of both.

1.2 Formalizing the Potential Infinite

A potential infinite set \mathcal{M} is a dynamic, finite set. To formalize this consequently, any reference to \mathcal{M} can only be made by reference to some finite state \mathcal{M}_i . A completed totality of all elements or all states of this set does not exist. For example, there exist states \mathbb{N}_i of the set of natural numbers, but there is neither a complete set \mathbb{N} nor the complete family of all sets \mathbb{N}_i . The latter has to do with the fact that, for a consequent reading, this finitistic view applies to meta-level concepts as well, in particular to the index set from which the indices i are taken; see Section 1.4 for more on this.

In a first step we introduce indices i , given by a directed index set \mathcal{I} , i.e., a set \mathcal{I} of stages together with a binary, reflexive and transitive relation \leq so that any finite set of indices has an upper bound. A potential infinite set is thus a family $\mathcal{M}_{\mathcal{I}}$ of states \mathcal{M}_i , $i \in \mathcal{I}$, where all sets are finite.

We want to express that two elements $a_i \in \mathcal{M}_i$ and $a_{i'} \in \mathcal{M}_{i'}$ at different stages i and i' are equal. This equality is not given by an equivalence relation, but by a family of reflexive relations $\overset{p}{\mapsto}_{i',i} \subseteq \mathcal{M}_{i'} \times \mathcal{M}_i$ for $i' \geq i$. We write $a_{i'} \overset{p}{\mapsto} a_i$ for $(a_{i'}, a_i) \in \overset{p}{\mapsto}_{i',i}$, saying that a_i is a predecessor of $a_{i'}$, and we use $\overset{p}{\mapsto}$ as an abbreviation for $\overset{p}{\mapsto}_{i',i}$. Reflexivity means that $a_i \overset{p}{\mapsto} a_i$ for all $i \in \mathcal{I}$ and $a_i \in \mathcal{M}_i$. We do not require $\overset{p}{\mapsto}$ to be transitive.¹ A *system* $(\mathcal{M}_{\mathcal{I}}, \overset{p}{\mapsto})$

¹ The reason is this: The index set of the function space is $\mathcal{I} \times \mathcal{J}$ with pairs written as $i \rightarrow j$ (c.f. Section 2.2). Call an extension of the index from $i \rightarrow j$ to $i \rightarrow j'$, $j' \geq j$, *covariant* and from $i \rightarrow j$ to $i' \rightarrow j$, $i' \geq i$, *contravariant*. Then a combination of two different kind of extensions may fail to be a correct extension. For an example, see [5].

consists of a family $\mathcal{M}_{\mathcal{I}}$ together with reflexive relations $\vdash^{\mathcal{I}}$. Later we will introduce further properties in order to allow a function space construction, leading to the notion of a *factor system* [5].

Let types $\varrho \in Typ$ be given. In this paper they consist of some base types ι , including type *bool* for propositions, which are interpreted classically as the Boolean values *true* and *false*, and a type constructor \rightarrow . So types are $\varrho ::= \iota \mid \varrho \rightarrow \varrho$. A (typing) context Γ is a list of types $(\varrho_0, \dots, \varrho_{n-1})$. More explicitly we can write $(x_0 : \varrho_0, \dots, x_{n-1} : \varrho_{n-1})$ for the context Γ , since we use a fixed list of variables x_0, x_1, x_2, \dots in Section 3. The empty context is $()$ and $\Gamma.\varrho$ denotes the context Γ , extended by ϱ . We assume that for each type ϱ there is an index set $(\mathcal{I}_{\varrho}, \leq)$, so each type comes with its own set of stages. For instance, the index set for type *nat* is (\mathbb{N}^+, \leq) with $\mathcal{M}_i = \mathbb{N}_i$ for $i \in \mathbb{N}^+$.² Another example is type *bool* with the singleton set $(\{\text{bool}\}, =)$ of one index *bool* and $\mathcal{M}_{\text{bool}} := \mathbb{B} := \{\text{true}, \text{false}\}$. For a typing context $\Gamma = (\varrho_0, \dots, \varrho_{n-1})$ define $\mathcal{I}_{\Gamma} := \mathcal{I}_{\varrho_0} \times \dots \times \mathcal{I}_{\varrho_{n-1}}$ and endow \mathcal{I}_{Γ} with the product order.

Since we want to consider sets, relations and functions as objects, i.e., as elements of further potential infinite sets, we have to consider elements as dynamic entities as well. A dynamic object a of type ϱ is given by its states a_i , $i \in \mathcal{I}_{\varrho}$. As with sets, any reference to a can only be a reference to one of its states a_i .

A dynamic object needs not be defined on all stages $i \in \mathcal{I}_{\varrho}$, so the index set \mathcal{I}_a , for which a state a_i of a exists, is a subset of \mathcal{I}_{ϱ} , i.e., $\mathcal{I}_a \subseteq \mathcal{I}_{\varrho}$. One of the basic requirements is that every object has “sufficiently many” indices \mathcal{I}_a . For natural numbers, or more generally for base type objects, the situation is simple: If a number n occurs at stage $n + 1$, i.e., $n \in \mathbb{N}_{n+1}$, it will be there for all future stages $m > n + 1$, i.e., $n \in \mathbb{N}_m$, so the index set $\mathcal{I}_n = \{n + 1, n + 2, \dots\}$ is an up-set. By *up-set* we mean a non-empty, upward closed subset, in this case a subset of \mathbb{N}^+ . For higher-order functions, however, the situation is less straightforward and is one of the main challenges of this approach.

The concept of a potential infinite has two aspects, a cardinal aspect \mathfrak{D} and an ordinal aspect \ll . Let $C = (i_0, \dots, i_{n-1}) \in \mathcal{I}_{\Gamma}$ be a list of indices $i_0 \in \mathcal{I}_{\varrho_0}, \dots, i_{n-1} \in \mathcal{I}_{\varrho_{n-1}}$, for $\Gamma = (\varrho_0, \dots, \varrho_{n-1})$, called *state context*. Write $C.i$ for the extension of C by i .

1. The cardinal aspect is given by a set $\mathfrak{D}_{\Gamma} \subseteq \mathcal{P}(\mathcal{I}_{\Gamma})$ for each context Γ . $\mathcal{H} \in \mathfrak{D}_{\Gamma}$ says that there are *indefinitely many*, or *sufficiently many* contexts in \mathcal{H} .
2. The ordinal aspect is given by relations $\ll^{\Gamma.e} \in \mathfrak{D}_{\Gamma.e}$. Let $C \ll i$ stand for $C.i \in \ll^{\Gamma.e}$, meaning that a stage i is *indefinitely large*, or *sufficiently large* relative to the state context C .

The sets \mathfrak{D}_{Γ} satisfy the following properties: Each set $\mathcal{H} \in \mathfrak{D}_{\Gamma}$ is cofinal – recall that cofinality of \mathcal{H} means $\forall C \in \mathcal{I}_{\Gamma} \exists C' \geq C$ with $C' \in \mathcal{H}$. This minimal requirement simply states that we will always find a context in \mathcal{H} beyond any bound. \mathfrak{D}_{Γ} is closed under supersets, so “more than indefinitely many is indefinitely many”. Furthermore, an up-set on \mathcal{I}_{Γ} always has indefinitely many indices. Finally, the main restriction is that \mathfrak{D}_{Γ} is closed under intersections. This is necessary in order to guarantee that a relation between two objects can be established on indefinitely many indices. This amounts to saying:

\mathfrak{D}_{Γ} is a proper filter on $\{\mathcal{H} \subseteq \mathfrak{D}_{\Gamma} \mid \mathcal{H} \text{ is cofinal}\}$ and \mathfrak{D}_{Γ} contains all up-sets. (Filter)

² The use of \mathbb{N}^+ instead of \mathbb{N} as index set is done for technical reasons: A factor system has projections between different states \mathbb{N}_i , and there is no projection from \mathbb{N}_i , $i > 0$ into $\mathbb{N}_0 = \emptyset$.

We will use the locution \mathfrak{D} -many indices, which refers to a set in \mathfrak{D}_Γ , and *cofinal many indices*, which refers to a cofinal index set. The interpretation in a potential infinite structure, which we introduce in Section 3, will be relative to \mathfrak{D} (and later also relative to \ll , when we introduce the universal quantifier in a subsequent paper).

A basic theme of the potential infinite is *dependency*. In particular, there are no fixed sets \mathfrak{D} and \ll , but these are parameters that depend on factors of the concrete mathematical investigation and the state of it. Another way of expressing this is to say that \mathfrak{D} and \ll are intensional notions whose extension depends on the context of investigation. In the same way that \mathfrak{D} and \ll depend on the investigation, some concepts, in turn, depend on \mathfrak{D} and \ll ; the notion of continuity depends on \mathfrak{D} and the interpretation of the universal quantifier depends on \ll . It is possible to define continuity on limit sets without reference to the underlying system. The basic relation is then a family of PERs and we will explore this structure in more detail in a separate paper. Moreover, in this paper we deal only with the cardinal aspect, i.e., \ll is not considered here.

1.3 Relation to Constructive Approaches

We are only investigating the idea of a potential infinite, not that of constructivity, decidability, complexity or knowledge about existence, which are important concepts in intuitionism [14] and in theories about computability [8]. The common models of intuitionistic logic, such as Kripke models, or more generally topos-theoretic models [9], use unbounded universal quantification. For instance, in a Kripke model, the validity of a universal quantified formula uses a reference to all “future” nodes – there could be infinitely many of them and at each such node the carrier set could be an infinite set as well. In our approach, only finite sets are used, and in a consequent finitistic view only finitely many of them.

We use classical logic. This is because it has a simpler model than constructive models and is more widely used. Furthermore, the results presented here rely on the fact that classical logic has a finite number of truth values. This is an important difference from intuitionistic logic, which has, if truth values are used at all, infinitely many of them.

The novel aspect of this potential infinite model is the introduction of *state judgements*, which are refinements of typing judgements. If $\Gamma \vdash r : \varrho$ is such a typing judgement, then a state judgement has the form $C \mid r : i$, where C is a stage of the context Γ and i is a stage of the type ϱ . The idea that we can only refer to infinite objects via a specific state is reflected in the fact that the primary object of interpretations is a state judgement $C \mid r : i$. It is interpreted in a family of factor systems, which have only “local” application functions given as

$$App_{i,j} : [\mathcal{M}_i \rightarrow \mathcal{N}_j] \times \mathcal{M}_i \rightarrow \mathcal{N}_j, (f_{i \rightarrow j}, a_i) \mapsto f_{i \rightarrow j}(a_i).$$

Additionally, the typing judgement $\Gamma \vdash r : \varrho$ has an interpretation in the limit set. However, the notion of a limit of a factor system depends on the notion of “indefinitely many stages”, which we formalize as sets \mathfrak{D}_ϱ . This is in particular relevant for the function space and the possibility to define a “global” application function $App : [\mathcal{M} \rightarrow \mathcal{N}] \times \mathcal{M} \rightarrow \mathcal{N}, (f, a) \mapsto f(a)$ on limit sets, leading to a common *extensional type structure* [2]. The question which higher-order functions exist and whether all local applications together yield a global application depends on the properties of \mathfrak{D}_ϱ . A global application function is available for all first-order functions, however, this may not be the case for higher-order functions. In contrast, in a type structure, also referred to as (*typed*) *applicative structure*, a global application is available for all types. This holds analogously for *Kripke applicative structures* [10].

1.4 The Meta-Theory

The meta-theory in which the concepts are developed is classical higher-order logic, as formalized in Church’s simple type theory [3]. This theory will also serve as the investigated theory. At the object-level, we develop a potential infinite model in order to interpret typed λ -terms. At the meta-level these two views of infinity are relevant:

1. One accepts actual infinity at the meta-level. In this case, a limit can be seen as the usual actual infinite set beyond the system. This view allows a comparison of an actual infinite model, given as limit structure, with the potential infinite part, i.e., the system.
2. One uses the view that infinity is an extensible finite. Then a limit is an intermediate state of the system. This is the consistent realization of the finitistic approach.

The reflection principle, which is the main theorem of this paper, states in the first case that all objects in the limit, including propositions, have a counterpart in the factor system. So it says something like this: Whatever exists and holds under the assumption that actual infinite sets exist, already exists and holds at a sufficiently large stage in the system. For infinite objects, these can be seen as approximations. For type *bool*, which is interpreted in classical logic as a finite set of truth values $\{true, false\}$, the values are the same in the actual infinite limit set and at a finite stage of the system. This is because these values are not approximated.

In the second case, the reflection principle is only a means to show the correctness of the interpretation. However, a consistent realization requires that an infinite set on meta-level, like the index set \mathcal{I} , is only available at a stage j . A consequent realization in type theory moreover uses a type in place of the index set, together with a term \leq , which is then shown to be reflexive, transitive, and directed.

The use of a classical meta-theory instead of a constructive one is not essential here. We could also take an intuitionistic type theory at meta-level and develop most of the model theory in a pure constructive way. So we expect that the model construction can be formalized in common proof assistants such as Coq, Lean or Agda. We need, however, a bit of classical reasoning, at least when introducing the universal quantifier. To prove the reflection principle with universal quantifier as an extension of Theorem 24, one has to do a kind of Löwenheim-Skolem construction. This requires that a universal quantified formula is either true (at all stages), or it is false at some stage and is false at all later stages, too. It is of greater significance that, for a consequent realization, in which only potential infinities are used at meta-level, the proof assistant must implement state judgments.

1.5 Structure of the Paper

We already started in Section 1.2 to formalize the potential infinite as a dynamic concept, which replaces an actual infinite set with a *factor system*, which is, roughly, a generalization of a direct and inverse system. The concept of factor systems was first introduced in [5]. In Section 2 we reiterate the definition of a factor system and add further definitions that are necessary for an interpretation. These are primarily the notions of a direct and inverse factor system, which are required to interpret variables. We show the construction of the function space between two factor systems and elucidate the notions of a target and a limit of a factor system. As with direct and inverse systems, limits are targets that satisfy a universal property and have a concrete construction. In Section 2.5 we demonstrate how to introduce an application that makes limit sets a type structure.

Section 3 introduces a judgement for states, parallel to judgements for types. These are defined on a fragment of the simply typed λ -calculus, which we call *core fragment*. Based on these state judgements we give a first version of an interpretation of λ -terms in the core

fragment, not including any constants. In particular, the present paper does not yet include logic. We give an interpretation of types and terms with two parts, one is within the system, the other is in its limit. Based on this interpretation we show a first version of a reflection principle, which says that an element a of type ρ in the limit set is reflected by an element a_i at some stage $i \in \mathcal{I}_\rho$ in the system. This element a_i is an approximation of a , and at the same time it fully represents a . If logic is included, then the representation includes all propositions about these elements, so anything we can say about a is true if and only if it is true for a_i .

2 Factor Systems and their Limits

Factor systems and their limits have been introduced in [5]. In this paper, we summarize their properties. The main concept is that of a factor system, those of a prefactor system (with embeddings/projections) are afferent notions. In addition to that, we introduce the notion of a *stable* system, which is a natural notion to prove stronger properties, although these are not necessary here. We prove that stability is closed under both the function space construction and the limit construction.

Relevant for the interpretation in Section 3 are the specific forms of a *direct* and *inverse* factor system. A direct factor system is similar to a direct system, it is more specific in the sense that the embedding is part of an embedding-projection pair. On the other hand it is more general in the sense that equations hold only up to an equivalence relation. The same holds for inverse factor systems and inverse systems. Moreover, we introduce the concept of a *homomorphism* between two systems.

The subsequent Lemmata 9 and 10 are extended versions of corresponding lemmata in [5], which use the property (Filter), not only cofinality. These versions are necessary to prove Corollary 15, which states that the function space construction commutes with the limit construction. This is a prerequisite for the definition of a model. Proposition 7 and Corollary 12 describe how direct and inverse factor systems extend to the function space and to the limit. Both are essential to prove the reflection principle for variables.

2.1 Factor Systems

\mathcal{I} will always denote a non-empty directed index set with preorder \leq . A *system* is a pair $(\mathcal{M}_{\mathcal{I}}, \overset{p}{\rightrightarrows})$ consisting of a family $\mathcal{M}_{\mathcal{I}} := (\mathcal{M}_i)_{i \in \mathcal{I}}$ and reflexive (for $i = i'$) relations $\overset{p}{\rightrightarrows}$ on $\mathcal{M}_{i'} \times \mathcal{M}_i$ for $i' \geq i$. Two elements $a_i \in \mathcal{M}_i$ and $b_j \in \mathcal{M}_j$ are *consistent*, written as $a_i \asymp b_j$, iff there is an index $i' \geq i, j$ and some $a_{i'} \in \mathcal{M}_{i'}$ such that $a_{i'} \overset{p}{\rightrightarrows} a_i$ and $a_{i'} \overset{p}{\rightrightarrows} b_j$. As a convention, whenever we use a suffix $i \in \mathcal{I}$ for some element, this refers to the state, e.g. $a_i \in \mathcal{M}_i$. An important special case is that the relations $\overset{p}{\rightrightarrows}$ are partial functions, which is equivalent to:

$$a_i \asymp b_i \iff a_i \overset{p}{\rightrightarrows} b_i \iff a_i = b_i \quad (\text{Fun})$$

for all $a_i, b_i \in \mathcal{M}_i$ and all $i \in \mathcal{I}$. A system that satisfies (Fun) is called *standard*. $(\mathcal{M}_{\mathcal{I}}, \overset{p}{\rightrightarrows})$ is a *prefactor system* iff it is a system satisfying

$$a_{i'} \asymp a_i \iff a_{i'} \overset{p}{\rightrightarrows} a_i \quad (\text{Factor})$$

for all $a_i \in \mathcal{M}_i$ and $a_{i'} \in \mathcal{M}_{i'}$ with $i \leq i'$. The relation \asymp is then an equivalence relation on a single set \mathcal{M}_i with $a_i \asymp b_i \iff a_i \overset{p}{\rightrightarrows} b_i \iff b_i \overset{p}{\rightrightarrows} a_i$ for $a_i, b_i \in \mathcal{M}_i$. In a prefactor system $b_{i'} \asymp a_{i'} \overset{p}{\rightrightarrows} a_i$ implies $b_{i'} \overset{p}{\rightrightarrows} a_i$, but sometimes we want to have the “dual” property

as well, which we call *stability*. Although we can do without stability for most properties, it is an obvious requirement, and all natural examples satisfy this property. Note that a system that satisfies (Fun) is automatically stable.

► **Definition 1.** A system $(\mathcal{M}_{\mathcal{I}}, \overset{P}{\mapsto})$, and in particular the relation $\overset{P}{\mapsto}$, is called *stable* iff for all $i' \geq i$, all $a_{i'} \in \mathcal{M}_{i'}$ and $a_i, b_i \in \mathcal{M}_i$

$$a_{i'} \overset{P}{\mapsto} a_i \asymp b_i \Rightarrow a_{i'} \overset{P}{\mapsto} b_i. \quad (\text{Stab})$$

A family $emb = (emb_i^{i'})_{i \leq i'}$ of \asymp -embeddings consists of \asymp -preserving maps $emb_i^{i'} : \mathcal{M}_i \rightarrow \mathcal{M}_{i'}$ satisfying $emb_i^{i'}(a_i) \asymp a_i$ and $emb_{i'}^{i''}(emb_i^{i'}(a_i)) \asymp emb_i^{i''}(a_i)$. The requirement \asymp -preserving means that $emb_i^{i'}(a_i) \asymp emb_i^{i'}(b_i)$ holds for $a_i \asymp b_i$, for all $a_i, b_i \in \mathcal{M}_i$. Similar to a family of \asymp -embeddings, \asymp -projections $proj = (proj_i^{i'})_{i \leq i'}$ consist of \asymp -preserving maps $proj_i^{i'} : \mathcal{M}_{i'} \rightarrow \mathcal{M}_i$ satisfying $proj_i^{i'}(a_i) \asymp a_i$ and $proj_i^{i'}(proj_{i'}^{i''}(a_{i''})) \asymp proj_i^{i''}(a_{i''})$. Moreover, \asymp -embeddings emb together with \asymp -projections $proj$ form an \asymp -embedding-projection pair iff $proj_i^{i'}(emb_i^{i'}(a_i)) \asymp a_i$ holds for all $a_i \in \mathcal{M}_i$ and all $i \leq i'$.

The \asymp -embeddings emb and \asymp -projections $proj$ are *coherent* if they satisfy for all indices $i \leq i' \leq i''$

$$a_{i'} \overset{P}{\mapsto} a_i \Rightarrow emb_{i'}^{i''}(a_{i'}) \overset{P}{\mapsto} a_i \text{ and} \quad (\text{Emb})$$

$$a_{i''} \overset{P}{\mapsto} a_i \Rightarrow proj_{i'}^{i''}(a_{i''}) \overset{P}{\mapsto} a_i \text{ resp.} \quad (\text{Proj})$$

A \asymp -embedding-projection pair $(emb, proj)$ is coherent if emb and $proj$ are both coherent. Property (Emb) implies that $emb_i^{i'}(a_i) \overset{P}{\mapsto} a_i$ holds for all $a_i \in \mathcal{M}_i$, and in case that Property (Fun) holds, $\overset{P}{\mapsto}$ is a *partial surjection*.

► **Definition 2.** A prefactor system with embeddings is a *prefactor system* $(\mathcal{M}_{\mathcal{I}}, \overset{P}{\mapsto})$ with coherent \asymp -embeddings emb . A prefactor system with projections is a *prefactor system* $(\mathcal{M}_{\mathcal{I}}, \overset{P}{\mapsto})$ with coherent \asymp -projections $proj$. A factor system is a *prefactor system* $(\mathcal{M}_{\mathcal{I}}, \overset{P}{\mapsto})$ with a coherent \asymp -embedding-projection pair $(emb, proj)$. A *prefactor system* is *direct* iff it has coherent \asymp -embeddings which satisfy

$$a_{i'} \overset{P}{\mapsto} a_i \iff a_{i'} \asymp emb_i^{i'}(a_i). \quad (\text{Dir})$$

A *prefactor system* is *inverse* iff it has coherent \asymp -projections which satisfy

$$a_{i'} \overset{P}{\mapsto} a_i \iff proj_i^{i'}(a_{i'}) \asymp a_i. \quad (\text{Inv})$$

If $\overset{P}{\mapsto}$, emb and $proj$ are known, we often call $\mathcal{M}_{\mathcal{I}}$ a *factor system* (and likewise with prefactor systems with embeddings/projections).

► **Example 3.** The embeddings emb in a direct system $(\mathcal{M}_{\mathcal{I}}, emb)$, with $a_{i'} \overset{P}{\mapsto} a_i : \iff emb_i^{i''}(a_i) = emb_{i'}^{i''}(a_{i'})$ for some $i'' \geq i, i'$, are automatically \asymp -embeddings, since they preserve \asymp . If there are \asymp -projections $proj$, such that emb and $proj$ form a \asymp -embedding-projection pair, then $(\mathcal{M}_{\mathcal{I}}, \overset{P}{\mapsto}, emb, proj)$ is a direct factor system.

An inverse system $(\mathcal{M}_{\mathcal{I}}, proj)$ satisfies (Fun), so the projections $proj$ are automatically \asymp -projections. If there are \asymp -embeddings emb , such that emb and $proj$ form a \asymp -embedding-projection pair, then $(\mathcal{M}_{\mathcal{I}}, \overset{P}{\mapsto}, emb, proj)$, with $a_{i'} \overset{P}{\mapsto} a_i : \iff proj_i^{i'}(a_{i'}) = a_i$, is an inverse factor system.

► **Example 4.** Consider $(\mathbb{N}_i)_{i \in \mathbb{N}^+}$ with the embedding-projection pair $emb_i^{i'} : \mathbb{N}_i \rightarrow \mathbb{N}_{i'}$, $n \mapsto n$ and $proj_i^{i'} : \mathbb{N}_{i'} \rightarrow \mathbb{N}_i$ with $n \mapsto \min(n, i - 1)$. In all three cases for $\overset{P}{\mapsto}$ they form an \asymp -embedding-projection pair and $((\mathbb{N}_i)_{i \in \mathbb{N}^+}, \overset{P}{\mapsto}, emb, proj)$ is a factor system:

1. The standard model of the natural numbers has $\mathbb{N}_{i'} \ni n \xrightarrow{p} n \in \mathbb{N}_i$ for all $n < i$, which is a direct factor system.
2. The definition $n' \xrightarrow{p} n : \iff \text{proj}_i^{i'}(n') = n$ makes it an inverse factor system.
3. With $n' \xrightarrow{p} n$ for all $n' \in \mathbb{N}_{i'}$ and $n \in \mathbb{N}_i$, the factor system $(\mathbb{N}_i)_{i \in \mathbb{N}^+}$ is direct and inverse.

The second example basically adds an infinite number to \mathbb{N} , while the last example is artificial, but shows the difference to direct and inverse limits when we pick up this factor system again in Example 8. One can easily check that $(\mathbb{N}_i)_{i \in \mathbb{N}^+}$ is stable in all three cases, and in the first two cases Property (Fun) is also satisfied.

► **Lemma 5.** *Let $\mathcal{M}_{\mathcal{I}}$ be a prefactor system and given indices $i'' \geq i' \geq i$.*

1. *If $\mathcal{M}_{\mathcal{I}}$ is direct, then $a_{i''} \xrightarrow{p} a_i$ implies $a_{i''} \xrightarrow{p} \text{emb}_i^{i'}(a_i)$.*
2. *If $\mathcal{M}_{\mathcal{I}}$ is inverse, then $a_{i''} \xrightarrow{p} a_{i'}$ implies $a_{i''} \xrightarrow{p} \text{proj}_i^{i'}(a_{i'})$.*

Proof. First, $a_{i''} \xrightarrow{p} a_i$ implies $\text{emb}_i^{i''}(\text{emb}_i^{i'}(a_i)) \simeq \text{emb}_i^{i''}(a_i) \simeq a_{i''}$ by (Dir), hence $\text{emb}_i^{i''}(\text{emb}_i^{i'}(a_i)) \xrightarrow{p} a_{i''}$. From (Emb) we deduce $\text{emb}_i^{i''}(\text{emb}_i^{i'}(a_i)) \xrightarrow{p} \text{emb}_i^{i'}(a_i)$ and thus $a_{i''} \xrightarrow{p} \text{emb}_i^{i'}(a_i)$ by (Factor). For the second clause let $a_{i''} \xrightarrow{p} a_{i'}$, then $\text{proj}_i^{i''}(a_{i''}) \simeq a_{i'}$ by (Inv), hence $\text{proj}_i^{i''}(a_{i''}) \simeq \text{proj}_i^{i''}(\text{proj}_i^{i'}(a_{i'})) \simeq \text{proj}_i^{i'}(a_{i'})$. Consequently $a_{i''} \xrightarrow{p} \text{proj}_i^{i'}(a_{i'})$ by (Inv) again, as claimed. ◀

Compare these properties with (Emb) and (Proj), which hold in any prefactor system with embeddings/projections.

► **Definition 6.** *A homomorphism $\Phi = (\Phi_0, (\Phi^i)_{i \in \mathcal{I}})$ between two systems $(\mathcal{M}_{\mathcal{I}}, \xrightarrow{p})$ and $(\mathcal{N}_{\mathcal{J}}, \xrightarrow{p})$ consists of maps $\Phi_0 : \mathcal{I} \rightarrow \mathcal{J}$ and $\Phi^i : \mathcal{M}_i \rightarrow \mathcal{N}_{\Phi_0(i)}$ such that Φ_0 is monotone and for all $i \leq i'$*

$$a_{i'} \xrightarrow{p} a_i \implies \Phi^{i'}(a_{i'}) \xrightarrow{p} \Phi^i(a_i). \quad (1)$$

If (1) is an equivalence, then Φ is said to be strong. A homomorphism between two prefactor systems with embeddings $(\mathcal{M}_{\mathcal{I}}, \xrightarrow{p}, \text{emb})$ additionally satisfies $\Phi^{i'}(\text{emb}_i^{i'}(a_i)) = \text{emb}_j^{j'}(\Phi^i(a_i))$, $j := \Phi_0(i)$, $j' := \Phi_0(i')$, a homomorphism between two prefactor systems with projections $(\mathcal{M}_{\mathcal{I}}, \xrightarrow{p}, \text{proj})$ satisfies $\Phi^i(\text{proj}_i^{i'}(a_{i'})) = \text{proj}_j^{j'}(\Phi^{i'}(a_{i'}))$, and a homomorphism between two factor systems satisfies both equations.

We call Φ injective (surjective) iff every map in Φ is injective (surjective resp.). An isomorphism Φ between two systems (prefactor systems with embeddings/projections, factor systems) is a homomorphism with a further homomorphism Ψ as its inverse, i.e., each part of Ψ is inverse to that of Φ . So an isomorphism is automatically strong. We write $\mathcal{M}_{\mathcal{I}} \simeq \mathcal{N}_{\mathcal{J}}$ if an isomorphism between $\mathcal{M}_{\mathcal{I}}$ and $\mathcal{N}_{\mathcal{J}}$ exists.

In the following we will use homomorphisms Φ only for the special situation that $\mathcal{I} = \mathcal{J}$ and Φ_0 is the identity map. In that case we do not mention Φ_0 .

2.2 The Function Space

The function space of two factor systems $\mathcal{M}_{\mathcal{I}}$ and $\mathcal{N}_{\mathcal{J}}$, denoted as $[\mathcal{M}_{\mathcal{I}} \rightarrow \mathcal{N}_{\mathcal{J}}]$, is a family of (finite) sets $[\mathcal{M}_i \rightarrow \mathcal{N}_j]$ indexed by pairs $(i, j) \in \mathcal{I} \times \mathcal{J}$ with product order, whereby we write $i \rightarrow j$ for such an index in $\mathcal{I} \times \mathcal{J}$. The set $[\mathcal{M}_i \rightarrow \mathcal{N}_j]$ consists of all (total) functions $f : \mathcal{M}_i \rightarrow \mathcal{N}_j$ which preserve \simeq , i.e., which satisfy $f(a_i) \simeq f(b_i)$ for $a_i \simeq b_i$. If the relations \xrightarrow{p} are partial functions on $\mathcal{M}_{\mathcal{I}}$ and $\mathcal{N}_{\mathcal{J}}$, then \simeq is the identity on \mathcal{M}_i and \mathcal{N}_j and $[\mathcal{M}_i \rightarrow \mathcal{N}_j]$ simply consists of all functions from \mathcal{M}_i to \mathcal{N}_j .

6:10 Potential Infinite Models of Type Theory

Let $f \in [\mathcal{M}_i \rightarrow \mathcal{N}_j]$, $f' \in [\mathcal{M}_{i'} \rightarrow \mathcal{N}_{j'}]$ and $i \rightarrow j \leq i' \rightarrow j'$, i.e., $i \leq i'$ and $j \leq j'$. The basic relation $\overset{P}{\mapsto}$ on the function space is a logical relation [12]. It is thus defined as

$$f' \overset{P}{\mapsto} f \iff a_{i'} \overset{P}{\mapsto} a_i \text{ implies } f'(a_{i'}) \overset{P}{\mapsto} f(a_i)$$

for all $a_{i'} \in \mathcal{M}_{i'}$ and $a_i \in \mathcal{M}_i$. The embedding-projection pair for the function space is defined in the usual way:

$$\begin{aligned} \text{emb}_{i \rightarrow j}^{i' \rightarrow j'} : [\mathcal{M}_i \rightarrow \mathcal{N}_j] &\rightarrow [\mathcal{M}_{i'} \rightarrow \mathcal{N}_{j'}] & f &\mapsto \text{emb}_j^{j'} \circ f \circ \text{proj}_i^{i'}, \\ \text{proj}_{i \rightarrow j}^{i' \rightarrow j'} : [\mathcal{M}_{i'} \rightarrow \mathcal{N}_{j'}] &\rightarrow [\mathcal{M}_i \rightarrow \mathcal{N}_j] & f' &\mapsto \text{proj}_j^{j'} \circ f' \circ \text{emb}_i^{i'}. \end{aligned}$$

► **Proposition 7.** *If $\mathcal{M}_{\mathcal{I}}$ and $\mathcal{N}_{\mathcal{J}}$ are both factor systems, so is their function space. Moreover,*

1. *If $\overset{P}{\mapsto}$ satisfies (Fun) or (Stab) on $\mathcal{N}_{\mathcal{J}}$, so does $\overset{P}{\mapsto}$ on $[\mathcal{M}_{\mathcal{I}} \rightarrow \mathcal{N}_{\mathcal{J}}]$.*
2. *If $\mathcal{M}_{\mathcal{I}}$ is inverse and $\mathcal{N}_{\mathcal{J}}$ direct, then $[\mathcal{M}_{\mathcal{I}} \rightarrow \mathcal{N}_{\mathcal{J}}]$ is direct.*
3. *If $\mathcal{M}_{\mathcal{I}}$ is direct and $\mathcal{N}_{\mathcal{J}}$ inverse, then $[\mathcal{M}_{\mathcal{I}} \rightarrow \mathcal{N}_{\mathcal{J}}]$ is inverse.*

Proof. This has been proven in [5], except for the statements about (Stab) and about direct and inverse factor systems. Assume $f' \overset{P}{\mapsto} f \asymp g$ for $f' \in [\mathcal{M}_{i'} \rightarrow \mathcal{N}_{j'}]$ and $f, g \in [\mathcal{M}_i \rightarrow \mathcal{N}_j]$, $i \rightarrow j \leq i' \rightarrow j'$, so that we have to show $f' \overset{P}{\mapsto} g$. Let $a_{i'} \overset{P}{\mapsto} a_i$, and to confirm that $f'(a_{i'}) \overset{P}{\mapsto} g(a_i)$, use the stability condition on $\mathcal{N}_{\mathcal{J}}$ applied to $f'(a_{i'}) \overset{P}{\mapsto} f(a_i) \asymp g(a_i)$.

Next, let $\mathcal{M}_{\mathcal{I}}$ be inverse and $\mathcal{N}_{\mathcal{J}}$ direct. Assume first that $f' \overset{P}{\mapsto} f$ with $f' \in [\mathcal{M}_{i'} \rightarrow \mathcal{N}_{j'}]$, $f \in [\mathcal{M}_i \rightarrow \mathcal{N}_j]$ and $i \rightarrow j \leq i' \rightarrow j'$. We wish to show that $f' \asymp \text{emb}_{i \rightarrow j}^{i' \rightarrow j'}(f)$, which is the same as $f' \asymp \text{emb}_j^{j'} \circ f \circ \text{proj}_i^{i'}$. Let $a_{i'} \overset{P}{\mapsto} b_{i'}$ and define $b_i := \text{proj}_i^{i'}(b_{i'})$. Then $\text{proj}_i^{i'}(a_{i'}) \asymp b_i$ since projections preserve \asymp . It follows that $a_{i'} \overset{P}{\mapsto} b_{i'}$ since $\mathcal{M}_{\mathcal{I}}$ is inverse, and thus $f'(a_{i'}) \overset{P}{\mapsto} f(b_{i'})$. Consequently, $f'(a_{i'}) \asymp \text{emb}_j^{j'}(f(b_{i'})) = \text{emb}_j^{j'} \circ f \circ \text{proj}_i^{i'}(b_{i'})$, since $\mathcal{N}_{\mathcal{J}}$ is direct.

For the other direction assume $f' \asymp \text{emb}_{i \rightarrow j}^{i' \rightarrow j'}(f)$ and let $a_{i'} \overset{P}{\mapsto} a_i$. We shall prove that $f'(a_{i'}) \overset{P}{\mapsto} f(a_i)$. Then $a_{i'} \overset{P}{\mapsto} a_i$ implies $\text{proj}_i^{i'}(a_{i'}) \asymp a_i$ since $\mathcal{M}_{\mathcal{I}}$ is inverse. Now f and the embeddings preserve \asymp , so $\text{emb}_j^{j'}(f(\text{proj}_i^{i'}(a_{i'}))) \asymp \text{emb}_j^{j'}(f(a_i))$. Certainly $f'(a_{i'}) \asymp \text{emb}_j^{j'}(f(a_i))$. This shows $f'(a_{i'}) \overset{P}{\mapsto} f(a_i)$, since $\mathcal{N}_{\mathcal{J}}$ is direct, as claimed. The proof of the last statement is verified in a similar way. ◀

2.3 Targets and Limits

The subsequent concepts require sets $\mathfrak{D}(\mathcal{I})$ from Section 1.2 with Property (Filter), whereby we write \mathfrak{D}_ρ for $\mathfrak{D}(\mathcal{I}_\rho)$ and \mathfrak{D}_Γ for $\mathfrak{D}(\mathcal{I}_\Gamma)$. A *target* $(\mathcal{M}, \overset{P}{\mapsto})$ for a system $\mathcal{M}_{\mathcal{I}}$ extends the system “at the top”, i.e., the extension leads to a system $\mathcal{M}_{\bar{\mathcal{I}}}$, called *compactification* of $\mathcal{M}_{\mathcal{I}}$, with index set $\bar{\mathcal{I}} := \mathcal{I} \cup \{top\}$, *top* as greatest index, and $\mathcal{M}_{top} = \mathcal{M}$. Let $a \overset{P}{\mapsto} a_i$ denote $a \overset{P}{\mapsto} a_i$ provided that $a \in \mathcal{M}$, and we also write Emb_i for emb_i^{top} and $Proj_i$ for proj_i^{top} . Relation $\overset{P}{\mapsto}$ on a target \mathcal{M} , and consequently relation \asymp on \mathcal{M} as well, is by definition the identity. The *extension* of an element $a \in \mathcal{M}$ is $Ext(a) := \{a_i \in \bigcup_{i \in \mathcal{I}} \mathcal{M}_i \mid a \overset{P}{\mapsto} a_i\}$. A target \mathcal{M} for a system $\mathcal{M}_{\mathcal{I}}$ satisfies by definition

$$\mathcal{I}_a := \{i \in \mathcal{I} \mid \exists a_i \in \mathcal{M}_i \ a \overset{P}{\mapsto} a_i\} \in \mathfrak{D}(\mathcal{I})$$

for all objects $a \in \mathcal{M}$. Moreover, if the system is a prefactor system, a prefactor system with embeddings/projections or a factor system, then the compactification $\mathcal{M}_{\bar{\mathcal{I}}}$ must have this additional structure with its properties as well. Whereas the compactification of a system is automatically a system, the compactification of a *prefactor system* requires for all $a \in \mathcal{M}$, $a_i \in \mathcal{M}_i$, $a_{i'} \in \mathcal{M}_{i'}$ and $i \leq i'$

$$a \overset{P}{\mapsto} a_{i'}, a \overset{P}{\mapsto} a_i \Rightarrow a_{i'} \overset{P}{\mapsto} a_i. \quad (2)$$

If \mathcal{M} is a target for a *prefactor system with embeddings* $\mathcal{M}_{\mathcal{I}}$, then this implies the existence of \asymp -embeddings $Emb_i : \mathcal{M}_i \rightarrow \mathcal{M}$, satisfying $Emb_{i'}(emb_i^{i'}(a_i)) = Emb_i(a_i)$ and

$$a_{i'} \xrightarrow{p} a_i \Rightarrow Emb_{i'}(a_{i'}) \xrightarrow{p} a_i \text{ for all } i \leq i', \quad (3)$$

$a_i \in \mathcal{M}_i$, $a_{i'} \in \mathcal{M}_{i'}$ and $a \in \mathcal{M}$. If \mathcal{M} is a target for a *prefactor system with projections* $\mathcal{M}_{\mathcal{I}}$, then there are moreover \asymp -projections $Proj_i : \mathcal{M} \rightarrow \mathcal{M}_i$ such that $proj_i^{i'}(Proj_{i'}(a)) \asymp Proj_i(a)$ and

$$a \xrightarrow{p} a_i \Rightarrow Proj_{i'}(a) \xrightarrow{p} a_i \text{ for all } i \leq i'. \quad (4)$$

If \mathcal{M} is a target for a *factor system*, then Emb and $Proj$ with these properties exist and both form a \asymp -embedding-projection pair. We write $(\mathcal{M}, \xrightarrow{p})$, $(\mathcal{M}, \xrightarrow{p}, Emb)$, $(\mathcal{M}, \xrightarrow{p}, Proj)$ and $(\mathcal{M}, \xrightarrow{p}, Emb, Proj)$, respectively, for these targets. If \mathcal{M} is a target for a prefactor system with projections $\mathcal{M}_{\mathcal{I}}$, then the projections $Proj_i$ can be defined by

$$Proj_i(a) := proj_i^{i'}(a_{i'}) \text{ for some } i' \geq i \text{ with } a \xrightarrow{p} a_{i'} \quad (5)$$

for $a \in \mathcal{M}$. It follows from the properties of a prefactor system that $Proj_i(a)$ is unique modulo \asymp . Since $Emb_{i'}(a_{i'}) \xrightarrow{p} a_{i'}$ for $a_{i'} \in \mathcal{M}_{i'}$ by (3), we have for $i \leq i'$

$$Proj_i(Emb_{i'}(a_{i'})) \asymp proj_i^{i'}(a_{i'}). \quad (6)$$

A target $(\mathcal{M}, \xrightarrow{p})$ for a system $\mathcal{M}_{\mathcal{I}}$ is a *limit* of $\mathcal{M}_{\mathcal{I}}$ iff for every further target $(\mathcal{N}, \xrightarrow{q})$ for $\mathcal{M}_{\mathcal{I}}$ there is a unique map $\Phi : \mathcal{N} \rightarrow \mathcal{M}$ such that $a \xrightarrow{q} a_i$ implies $\Phi(a) \xrightarrow{p} a_i$. If the underlying system is a factor system or a prefactor system, then we call the limit *factor limit* and *prefactor limit*, resp. It turns out, however, that a factor limit is the same as the prefactor limit, and that this limit $\underline{\lim}(\mathcal{M}_{\mathcal{I}})$ is unique modulo isomorphism. Therefore we simply speak of a *limit*, or a *limit set*, if we want to distinguish it from a *limit element* in this limit set.

► **Example 8.** Recall Example 3. If a direct system is also a direct factor system and $a_{i'} \xrightarrow{p} a_i : \iff emb_i^{i''}(a_i) = emb_{i'}^{i''}(a_{i'})$ for some $i'' \geq i, i'$, then the factor limit is the direct limit, i.e., $\underline{\lim}(\mathcal{M}_{\mathcal{I}}) = \underline{\lim}(\mathcal{M}_{\mathcal{I}})$. If an inverse system is additionally an inverse factor system and $a_{i'} \xrightarrow{p} a_i : \iff proj_i^{i'}(a_{i'}) = a_i$, then the factor limit $\underline{\lim}(\mathcal{M}_{\mathcal{I}})$ is the inverse limit $\overleftarrow{\lim}(\mathcal{M}_{\mathcal{I}})$.

In the first case of Example 4 the limit is \mathbb{N} (which is the direct limit as well as the factor limit). In the second case the limit (inverse limit and factor limit) is $\mathbb{N}_{\infty} := \mathbb{N} \cup \{\infty\}$ with $\infty = (0, 1, 2, \dots)$. In the third case the factor limit is a singleton set, which is neither the direct nor the inverse limit.

2.4 Consistent Sets and Dynamic Elements

It is possible to define concrete targets and limits in the form of sets of states: A set $\alpha \subseteq \bigcup_{i \in \mathcal{I}} \mathcal{M}_i$ in a system $\mathcal{M}_{\mathcal{I}}$ is called a *consistent set* iff $a_{i'} \xrightarrow{p} a_i$ holds for all $a_{i'}, a_i \in \alpha$ with $i' \geq i$, and $\mathcal{I}_{\alpha} := \{i \in \mathcal{I} \mid \alpha \cap \mathcal{M}_i \neq \emptyset\} \in \mathfrak{D}(\mathcal{I})$. If we already have a target \mathcal{M} for a prefactor system, then the set $Ext(a)$ is such a consistent set for all $a \in \mathcal{M}$. The set of all consistent sets in a (pre)factor system $\mathcal{M}_{\mathcal{I}}$ is itself a target for $\mathcal{M}_{\mathcal{I}}$ with $\alpha \xrightarrow{p} a_i : \iff a_i \in \alpha$. In other words, if \mathcal{M} denotes the set of all consistent sets, then the target is (\mathcal{M}, \exists) .

A *dynamic element* is a maximal (w.r.t. subset inclusion) consistent set, and for each consistent set α in a prefactor system there is exactly one dynamic element α^m such that $\alpha \subseteq \alpha^m$. Let $\mathcal{EL}(\mathcal{M}_{\mathcal{I}})$ denote the set of all of these dynamic elements, then

$$(\mathcal{EL}(\mathcal{M}_{\mathcal{I}}), \exists) \quad (7)$$

6:12 Potential Infinite Models of Type Theory

is a prefactor limit of $\mathcal{M}_{\mathcal{I}}$. For the next two lemmata, recall that it is assumed that $\mathfrak{D}(\mathcal{I})$ is a filter as defined in (Filter).

► **Lemma 9.** *Let α be a consistent set in a prefactor system $(\mathcal{M}_{\mathcal{I}}, \overset{p}{\mapsto})$ and $b_i \in \mathcal{M}_i$, then the following are equivalent:*

1. $b_i \in \alpha^m$.
2. $\alpha \cup \{b_i\}$ is a consistent set.
3. There are cofinal many $i' \in \mathcal{I}$ with $a_{i'} \overset{p}{\mapsto} b_i$ for some $a_{i'} \in \alpha$.
4. $a_{i'} \overset{p}{\mapsto} b_i$ for all $a_{i'} \in \alpha$ with $i' \geq i$.
5. There are \mathfrak{D} -many $i' \in \mathcal{I}$ with $a_{i'} \overset{p}{\mapsto} b_i$ for some $a_{i'} \in \alpha$.

Proof. This has been shown in [5], except the last clause. For its equivalence to the other statements notice that it follows from Clause 4. since $\mathcal{I}_\alpha \cap \uparrow i \in \mathfrak{D}(\mathcal{I})$, and it implies Clause 3. since each set in $\mathfrak{D}(\mathcal{I})$ is cofinal. ◀

► **Lemma 10.** *Let α and β be consistent sets in a prefactor system $(\mathcal{M}_{\mathcal{I}}, \overset{p}{\mapsto})$, then the following are equivalent:*

1. $\alpha^m = \beta^m$.
2. $a_i \asymp b_i$ for all $i \in \mathcal{I}$ with $a_i \in \alpha$ and $b_i \in \beta$.
3. $a_i \asymp b_j$ for all $i, j \in \mathcal{I}$ with $a_i \in \alpha$ and $b_j \in \beta$.

Proof. The equivalence of 1. and 3. has been shown in [5], so it suffices to prove that 2. implies $b_i \in \alpha^m$ for each $b_i \in \beta$. We wish to find cofinal many $a_{i'} \in \alpha$ with $a_{i'} \overset{p}{\mapsto} b_i$ and apply Lemma 9. There are \mathfrak{D} -many $i' \geq i$ with $i' \in \mathcal{I}_\alpha \cap \mathcal{I}_\beta$. For all $a_{i'} \in \alpha$, $b_{i'} \in \beta$ we have $a_{i'} \asymp b_{i'}$ by assumption, hence $a_{i'} \overset{p}{\mapsto} b_i$, since $b_{i'} \overset{p}{\mapsto} b_i$. ◀

If one, and hence all, of the conditions in Lemma 10 are true, then we write $\alpha \sim \beta$. Call an element $a \in \mathcal{M}$ of a target \mathcal{M} for a prefactor system $\mathcal{M}_{\mathcal{I}}$ a *limit (element)* of a consistent set α iff $\alpha \sim \text{Ext}(a)$. Obviously, a is a limit of its extension $\text{Ext}(a)$. If there is only one limit element, we denote it as $\text{lim}(\alpha)$.

Given a factor system $\mathcal{M}_{\mathcal{I}}$. Then $\mathcal{EL}(\mathcal{M}_{\mathcal{I}})$ is, up to isomorphism, the limit $\underline{\text{lim}}(\mathcal{M}_{\mathcal{I}})$, whereby projections have been defined by (5) and embeddings are

$$\text{Emb}_i(a_i) := \{ \text{emb}_i^{i'}(a_i) \in \bigcup_{j \in \mathcal{I}} \mathcal{M}_j \mid i' \geq i \}^m. \quad (8)$$

Let \mathcal{M} be a target for a system $\mathcal{M}_{\mathcal{I}}$, then \mathcal{M} is *maximal (over $\mathcal{M}_{\mathcal{I}}$)* iff $\text{Ext}(a) \in \mathcal{EL}(\mathcal{M}_{\mathcal{I}})$ for all $a \in \mathcal{M}$. \mathcal{M} is *extensional (over $\mathcal{M}_{\mathcal{I}}$)* iff $\text{Ext}(a) \sim \text{Ext}(b)$ implies $a = b$ for all $a, b \in \mathcal{M}$. \mathcal{M} is *complete (over $\mathcal{M}_{\mathcal{I}}$)* iff for all consistent sets α there is a limit element a , i.e., some $a \in \mathcal{M}$ with $\text{Ext}(a) \sim \alpha$. One can characterize a limit $\underline{\text{lim}}(\mathcal{M}_{\mathcal{I}})$ also as a target that is maximal, extensional and complete over $\mathcal{M}_{\mathcal{I}}$.

► **Proposition 11.** *Assume a prefactor system $\mathcal{M}_{\mathcal{I}}$ has been compactified with a target \mathcal{M} , yielding the extended prefactor system $\mathcal{M}_{\bar{\mathcal{I}}}$.*

1. *If $\mathcal{M}_{\mathcal{I}}$ is stable and \mathcal{M} maximal over $\mathcal{M}_{\mathcal{I}}$, then $\mathcal{M}_{\bar{\mathcal{I}}}$ is stable, too.*
2. *If $\mathcal{M}_{\mathcal{I}}$ is direct and \mathcal{M} extensional over $\mathcal{M}_{\mathcal{I}}$, then $\mathcal{M}_{\bar{\mathcal{I}}}$ is direct and \mathcal{I}_a contains an up-set for all $a \in \mathcal{M}$.*
3. *If $\mathcal{M}_{\mathcal{I}}$ is inverse and \mathcal{M} maximal over $\mathcal{M}_{\mathcal{I}}$, then $\mathcal{M}_{\bar{\mathcal{I}}}$ is inverse and $\mathcal{I}_a = \mathcal{I}$ for all $a \in \mathcal{M}$.*

Proof.

1. In order to show stability of $\mathcal{M}_{\bar{\mathcal{I}}}$ it suffices to prove that $a \dashv^p a_i \asymp b_i$ implies $a \dashv^p b_i$ for all $a \in \mathcal{M}$. Indeed, there are cofinal many indices $i' \geq i$ with $a \dashv^p a_{i'}$, and consequently $a_{i'} \xrightarrow{p} a_i$. Applying (Stab) yields $a_{i'} \xrightarrow{p} b_i$ for cofinal many indices $i' \in \mathcal{I}$. By Lemma 9 $a \dashv^p b_i$ follows, since $Ext(a)$ is a maximal consistent set.
2. We claim that $a \dashv^p a_i \iff a = Emb_i(a_i)$ for $a \in \mathcal{M}$. So assume $a \dashv^p a_i$ and we first establish that $a \dashv^p a_j$ implies $Emb_i(a_i) \dashv^p a_j$ for all $j \in \mathcal{I}$. There are cofinal many $i' \geq i, j$ with $a \dashv^p a_{i'}$, $a_{i'} \xrightarrow{p} a_i$ and $a_{i'} \dashv^p a_j$, hence $a_{i'} \asymp emb_i^{i'}(a_i)$, because $\mathcal{M}_{\mathcal{I}}$ is direct. It follows that $emb_i^{i'}(a_i) \dashv^p a_j$ for cofinal many indices $i' \in \mathcal{I}$ and thus $Emb_i(a_i) \dashv^p a_j$. This proves $Ext(a) \sim Ext(Emb_i(a_i))$, so a and $Emb_i(a_i)$ are equal since \mathcal{M} is extensional over $\mathcal{M}_{\mathcal{I}}$. The inverse implication $a = Emb_i(a_i) \Rightarrow a \dashv^p a_i$ holds trivially. Moreover, $a = Emb_i(a_i)$ for $a \in \mathcal{M}$ implies that \mathcal{I}_a contains the up-set $\uparrow i$.
3. We shall prove that $a \dashv^p a_i \iff Proj_i(a) \asymp a_i$. For the forward implication assume $Proj_i(a) = proj_i^{i'}(a_{i'})$ for some $a_{i'}$ with $a \dashv^p a_{i'}$. We establish that $proj_i^{i'}(a_{i'}) \asymp a_i$ for $a \dashv^p a_i$ by using the fact that $\mathcal{M}_{\mathcal{I}}$ is inverse and that $a \dashv^p a_{i'}$ and $a \dashv^p a_i$ implies $a_{i'} \xrightarrow{p} a_i$.

For the backward implication assume $Proj_i(a) = proj_i^{i'}(a_{i'}) \asymp a_i$. To show $a \dashv^p a_i$ it suffices to find cofinal many indices $i'' \geq i'$ with $a \dashv^p a_{i''}$ and $a_{i''} \xrightarrow{p} a_i$. To this aim, we use maximality of \mathcal{M} . Indeed, there are cofinal many indices $i'' \geq i'$ with $a \dashv^p a_{i''}$, and to confirm that $a_{i''} \xrightarrow{p} a_i$ it suffices to prove $proj_i^{i''}(a_{i''}) \asymp a_i$. Now $a_{i''} \xrightarrow{p} a_{i'}$ is a consequence of $a \dashv^p a_{i''}$ and $a \dashv^p a_{i'}$, hence $proj_i^{i''}(a_{i''}) \asymp a_{i'}$. Because \asymp -projections preserve \asymp , we deduce $proj_i^{i''}(a_{i''}) \asymp proj_i^{i'}(proj_i^{i''}(a_{i''})) \asymp proj_i^{i'}(a_{i'}) \asymp a_i$, as claimed. For all $i \in \mathcal{I}$ there is some a_i with $a \dashv^p a_i \asymp Proj_i(a)$, hence $\mathcal{I}_\alpha = \mathcal{I}$. \blacktriangleleft

► **Corollary 12.** *Assume a prefactor system $\mathcal{M}_{\mathcal{I}}$ has been compactified with a limit \mathcal{M} , yielding the extended prefactor system $\mathcal{M}_{\bar{\mathcal{I}}}$. If $\mathcal{M}_{\mathcal{I}}$ is stable (direct, inverse), then $\mathcal{M}_{\bar{\mathcal{I}}}$ is also stable (direct, inverse resp.).*

2.5 Targets and Limits on the Function Space

Consider two factor systems $\mathcal{M}_{\mathcal{I}}$ and $\mathcal{N}_{\mathcal{J}}$ with limits \mathcal{M} and \mathcal{N} resp. Then

$$[\mathcal{M} \rightarrow_{\mathfrak{D}} \mathcal{N}] := \{f : \mathcal{M} \rightarrow \mathcal{N} \mid \mathcal{I}_f \in \mathfrak{D}(\mathcal{I} \times \mathcal{J})\}$$

is a target for $[\mathcal{M}_{\mathcal{I}} \rightarrow \mathcal{N}_{\mathcal{J}}]$, but not necessarily a limit. We now state a condition which guarantees that $[\mathcal{M} \rightarrow_{\mathfrak{D}} \mathcal{N}]$ is indeed a limit. So let ζ be a consistent set on the factor system $[\mathcal{M}_{\mathcal{I}} \rightarrow \mathcal{N}_{\mathcal{J}}]$, and α a consistent set on the factor system $\mathcal{M}_{\mathcal{I}}$. Define

$$App(\zeta, \alpha) := \zeta(\alpha) := \{f_{i \rightarrow j}(a_i) \mid f_{i \rightarrow j} \in \zeta \text{ and } a_i \in \alpha\}. \quad (\text{Appl})$$

It is easy to see that this set satisfies $b_{j'} \xrightarrow{p} b_j$ for all $j \leq j'$ and $b_j, b_{j'} \in \zeta(\alpha)$. However, $\zeta(\alpha)$ is not necessarily a consistent set. For that, $\mathcal{I}_f[\mathcal{I}_a] \in \mathfrak{D}(\mathcal{J})$ must be the case, whereby $\mathcal{H}[\mathcal{I}'] := \{j \in \mathcal{J} \mid \exists i \in \mathcal{I}' \ i \rightarrow j \in \mathcal{H}\}$ for $\mathcal{H} \subseteq \mathcal{I} \times \mathcal{J}$ and $\mathcal{I}' \subseteq \mathcal{I}$. This is ensured when the following condition for \mathfrak{D} is met for all sets \mathcal{I} and \mathcal{J} :

$$\mathcal{H} \in \mathfrak{D}(\mathcal{I} \times \mathcal{J}) \text{ and } \mathcal{I}' \in \mathfrak{D}(\mathcal{I}) \text{ implies } \mathcal{H}[\mathcal{I}'] \in \mathfrak{D}(\mathcal{J}). \quad (\text{D})$$

We will require Condition (D) for the rest of this paper.

► **Lemma 13.** *The application defined by (Appl) is extensional, i.e., for all $\zeta, \zeta' \in \mathcal{EL}([\mathcal{M}_{\mathcal{I}} \rightarrow \mathcal{N}_{\mathcal{J}}])$ we have:*

$$\zeta(\alpha) = \zeta'(\alpha) \text{ for all } \alpha \in \mathcal{EL}(\mathcal{M}_{\mathcal{I}}) \text{ implies } \zeta = \zeta'.$$

Proof. Assume $\zeta(\alpha) = \zeta'(\alpha)$ for all $\alpha \in \mathcal{EL}(\mathcal{M}_{\mathcal{I}})$ and let $f_{i \rightarrow j} \in \zeta$. We will show that $f_{i \rightarrow j} \in \zeta'$, which proves the lemma. Choose indices $i' \rightarrow j' \in \mathcal{I}_{\zeta} \cap \mathcal{I}_{\zeta'} \cap \uparrow(i \rightarrow j) \in \mathfrak{D}(\mathcal{I} \times \mathcal{J})$ and let $f'_{i' \rightarrow j'} \in \zeta'$. We wish to show $f'_{i' \rightarrow j'} \xrightarrow{P} f_{i \rightarrow j}$. Since this is the case for \mathfrak{D} -many indices $i' \rightarrow j'$, we have shown $f_{i \rightarrow j} \in \zeta'$ by Lemma 9, which suffices.

Now assume $a_{i'} \xrightarrow{P} a_i$ for $a_{i'} \in \mathcal{M}_{i'}$ and $a_i \in \mathcal{M}_i$, so that we have to prove $f'_{i' \rightarrow j'}(a_{i'}) \xrightarrow{P} f_{i \rightarrow j}(a_i)$. For $\alpha := \text{Emb}_{i'}(a_{i'}) \in \mathcal{EL}(\mathcal{M}_{\mathcal{I}})$ we know that $a_i \in \alpha$, by maximality and (3). Further, $f'_{i' \rightarrow j'}(a_{i'}) \in \zeta'(\alpha) = \zeta(\alpha)$ and $f_{i \rightarrow j}(a_i) \in \zeta(\alpha)$ implies $f'_{i' \rightarrow j'}(a_{i'}) \xrightarrow{P} f_{i \rightarrow j}(a_i)$, because $\zeta(\alpha)$ is a consistent set. \blacktriangleleft

► **Proposition 14.** *Let \mathcal{M} and \mathcal{N} be a targets for the factor systems $\mathcal{M}_{\mathcal{I}}$ and $\mathcal{N}_{\mathcal{J}}$ resp. Assume Condition (D) holds. If \mathcal{N} is maximal (extensional, complete) over $\mathcal{N}_{\mathcal{J}}$, so is $[\mathcal{M} \rightarrow_{\mathfrak{D}} \mathcal{N}]$ over $[\mathcal{M}_{\mathcal{I}} \rightarrow \mathcal{N}_{\mathcal{J}}]$.*

Proof. First we shall prove the claim about maximality, so let $g_{i \rightarrow j} \in \mathcal{M}_{i \rightarrow j}$ and we apply Lemma 9. Assume there is a set $\mathcal{H} \in \mathfrak{D}(\mathcal{I} \times \mathcal{J})$ with $f \xrightarrow{P} f_{i' \rightarrow j'} \xrightarrow{P} g_{i \rightarrow j}$ for $i' \rightarrow j' \in \mathcal{H}$, then it suffices to show $f \xrightarrow{P} g_{i \rightarrow j}$. By the definition of \xrightarrow{P} it must be checked that

$$f(a) \xrightarrow{P} g_{i \rightarrow j}(a_i) \text{ for } a \xrightarrow{P} a_i.$$

There are \mathfrak{D} -many indices $i' \geq i$ with $a \xrightarrow{P} a_{i'}$, since $\mathcal{I}_a \cap \uparrow i \in \mathfrak{D}(\mathcal{I})$. They satisfy $a_{i'} \xrightarrow{P} a_i$, because $a \xrightarrow{P} a_{i'}$ and $a \xrightarrow{P} a_i$. Applying Condition (D) give us \mathfrak{D} -many indices $j' \in \mathcal{H}[\mathcal{I}_a \cap \uparrow i]$, which all satisfy $f(a) \xrightarrow{P} f_{i' \rightarrow j'}(a_{i'}) \xrightarrow{P} g_{i \rightarrow j}(a_i)$. So maximality of \mathcal{N} yields $f(a) \xrightarrow{P} g_{i \rightarrow j}(a_i)$, showing the maximality of $[\mathcal{M} \rightarrow_{\mathfrak{D}} \mathcal{N}]$.

For extensionality we prove that $f = g$, provided $\text{Ext}(f) \sim \text{Ext}(g)$. So let $a \in \mathcal{M}$ and we claim that $f(a) = g(a)$. For all $i \rightarrow j \in \mathcal{I}_f \cap \mathcal{I}_g$ and $f_{i \rightarrow j} \in \text{Ext}(f)$, $g_{i \rightarrow j} \in \text{Ext}(g)$ we have $f_{i \rightarrow j} \simeq g_{i \rightarrow j}$ by definition. Condition (D) shows that

$$\mathcal{J}' := (\mathcal{I}_f \cap \mathcal{I}_g)[\mathcal{I}_a] \in \mathfrak{D}(\mathcal{J}),$$

and for all these indices $i \in \mathcal{I}_a$ and $j \in \mathcal{J}'$ we have $a \xrightarrow{P} a_i$ for some $a_i \in \mathcal{M}_i$, $f(a) \xrightarrow{P} f_{i \rightarrow j}(a_i)$, $g(a) \xrightarrow{P} g_{i \rightarrow j}(a_i)$ and $f_{i \rightarrow j}(a_i) \simeq g_{i \rightarrow j}(a_i)$. This yields $\text{Ext}(f(a)) \sim \text{Ext}(g(a))$, since the elements of $\text{Ext}(f(a))$ and $\text{Ext}(g(a))$ are consistent (i.e., in \simeq -relation to each other) for cofinal many indices $j \in \mathcal{J}$. We now appeal to extensionality of \mathcal{N} to get $f(a) = g(a)$.

To show completeness of $[\mathcal{M} \rightarrow_{\mathfrak{D}} \mathcal{N}]$ over $[\mathcal{M}_{\mathcal{I}} \rightarrow \mathcal{N}_{\mathcal{J}}]$ let ζ be a consistent set in $[\mathcal{M}_{\mathcal{I}} \rightarrow \mathcal{N}_{\mathcal{J}}]$. We define a limit element $f : \mathcal{M} \rightarrow \mathcal{N}$ of ζ by

$$f(a) := \text{limit element of } \beta := \{g_{i \rightarrow j}(a_i) \mid g_{i \rightarrow j} \in \zeta \cap [\mathcal{M}_i \rightarrow \mathcal{N}_j] \text{ and } a \xrightarrow{P} a_i\}$$

for all $a \in \mathcal{M}$. This function is well-defined: If $a \xrightarrow{P} a_i$, $a \xrightarrow{P} b_i$ and $f_{i \rightarrow j}, g_{i \rightarrow j} \in \zeta \cap [\mathcal{M}_i \rightarrow \mathcal{N}_j]$, then $a_i \simeq b_i$ follows and consequently $f_{i \rightarrow j}(a_i) \simeq g_{i \rightarrow j}(b_i)$; note that $a_i \simeq b_i \iff a_i \xrightarrow{P} b_i$ and the same for $f_{i \rightarrow j}(a_i) \simeq g_{i \rightarrow j}(b_i)$. This implies that any two elements in β are consistent. Condition (D) guarantees that $\mathcal{I}_f = \mathcal{I}_{\zeta}[\mathcal{I}_a] \in \mathfrak{D}(\mathcal{J})$. The set β is thus a consistent set and completeness of \mathcal{N} ensures that a limit element of β exists.

We claim that f is a limit of ζ . Let $f \xrightarrow{P} f_{i \rightarrow j}$ and $g_{i \rightarrow j} \in \zeta$ with $f_{i \rightarrow j}, g_{i \rightarrow j} \in \mathcal{M}_{i \rightarrow j}$ and we have to show $f_{i \rightarrow j} \xrightarrow{P} g_{i \rightarrow j}$. Given $b_i \xrightarrow{P} a_i$, both in \mathcal{M}_i , so it suffices to show that $f_{i \rightarrow j}(b_i) \xrightarrow{P} g_{i \rightarrow j}(a_i)$. Take $a := \text{Emb}_i(a_i) = \text{Emb}_i(b_i)$, then $a \xrightarrow{P} a_i$ and $a \xrightarrow{P} b_i$, hence $f(a) \xrightarrow{P} f_{i \rightarrow j}(b_i)$ and $g_{i \rightarrow j}(a_i) \in \beta$. Since $f(a)$ is a limit element of β , it follows that $\beta \sim \text{Ext}(f(a))$, and thus $f_{i \rightarrow j}(b_i) \simeq g_{i \rightarrow j}(a_i)$ by Lemma 10, so we are done. \blacktriangleleft

Recall that limits are targets which are maximal, extensional and complete. Recall also that $\mathcal{EL}([\mathcal{M}_{\mathcal{I}} \rightarrow \mathcal{N}_{\mathcal{J}}])$ is a limit of $[\mathcal{M}_{\mathcal{I}} \rightarrow \mathcal{N}_{\mathcal{J}}]$ and that limits are unique modulo isomorphism.

► **Corollary 15.** *Given two factor systems $\mathcal{M}_{\mathcal{I}}$ and $\mathcal{N}_{\mathcal{J}}$ with limits \mathcal{M} and \mathcal{N} resp. Assume Condition (D) holds, then $[\mathcal{M} \rightarrow_{\mathfrak{D}} \mathcal{N}]$ is a limit of $[\mathcal{M}_{\mathcal{I}} \rightarrow \mathcal{N}_{\mathcal{J}}]$. Equivalently stated, there is an isomorphism $[\mathcal{EL}(\mathcal{M}_{\mathcal{I}}) \rightarrow_{\mathfrak{D}} \mathcal{EL}(\mathcal{N}_{\mathcal{J}})] \simeq \mathcal{EL}([\mathcal{M}_{\mathcal{I}} \rightarrow \mathcal{N}_{\mathcal{J}}])$.*

3 Towards Models and Interpretations

This section contains the first steps towards an interpretation of a typed λ -term in a factor system and a limit of it. Let types $\varrho \in Typ$ be given by $\varrho ::= \iota \mid \varrho \rightarrow \varrho$ for base types ι . We assume that type *bool* is one of these base types. For each type ϱ let a non-empty, directed set \mathcal{I}_{ϱ} of indices be given and let $\mathfrak{D}_{\Gamma} := \mathfrak{D}(\mathcal{I}_{\Gamma})$ be a filter which satisfies (Filter). Maps on indices, such as Emb_i , extend in an obvious way to state contexts $C = (i_0, \dots, i_{n-1})$ as products, e.g. $Emb_C = Emb_{i_0} \times \dots \times Emb_{i_{n-1}}$.

Given also a basic version of a simply typed λ -calculus (without constants) and for convenience we use a fixed list of variables x_0, x_1, x_2, \dots within the Church-style λ -terms $r ::= x_k \mid rr \mid \lambda x_k^{\varrho} r$. We apply the usual conventions for λ -terms [11].

3.1 Judgements for Types and States

Typing judgements $\Gamma \vdash r : \varrho$ with $\Gamma = (\varrho_0, \dots, \varrho_{n-1})$ are defined recursively as follows:

$$\text{(VAR)} \frac{k < n}{\Gamma \vdash x_k : \varrho_k} \quad \text{(APP)} \frac{\Gamma \vdash r : \varrho \rightarrow \sigma \quad \Gamma \vdash s : \varrho}{\Gamma \vdash rs : \sigma} \quad \text{(ABS)} \frac{\Gamma.\varrho \vdash r : \sigma}{\Gamma \vdash \lambda x_n^{\varrho} r : \varrho \rightarrow \sigma}$$

We can write more explicitly $(x_0 : \varrho_0, \dots, x_{n-1} : \varrho_{n-1}, x_n : \varrho_n)$ for the context $\Gamma.\varrho_n$, so λ -abstractions binds the last variable of the context.³ When we speak of typed terms this refers to this typing judgement. It is well known that each term r in the simply typed λ -calculus has a unique type for a given context Γ .

► **Definition 16.** *Let ι denote a base type. The positive and negative types are defined as*

$$Typ^+ \ni \varrho^+ ::= \iota \mid \varrho^- \rightarrow \varrho^+ \quad \text{and} \quad Typ^- \ni \varrho^- ::= \text{bool} \mid \varrho^+ \rightarrow \varrho^-.$$

For an index $i \in \mathcal{I}_{\varrho}$ with $\varrho \in Typ^+$ we write $i \in Idx^+$, and similar for $i \in Idx^-$. Moreover, $Typ^c \ni \varrho^c ::= \iota \mid \varrho^+ \rightarrow \varrho^c \mid \varrho^- \rightarrow \varrho^c$.

For example, *nat* is a positive type, $\text{nat} \rightarrow \text{nat} \rightarrow \text{bool}$ is a negative type, $\text{bool} \rightarrow \text{bool}$ is both and $\text{nat} \rightarrow \text{nat}$ is neither. Obviously $Typ^+ \cup Typ^- \subseteq Typ^c$ and $\varrho \in Typ^+ \cap Typ^- \iff \varrho ::= \text{bool} \mid \varrho \rightarrow \varrho$. In order to formulate the rules for states, we use a fragment of the simply typed λ -calculus, which we refer to as the *core fragment*. This fragment is constrained in such a way that contexts $\Gamma = (\varrho_0, \dots, \varrho_{n-1})$ contain only positive or negative types, that is,

$$\varrho_k \in Typ^+ \cup Typ^- \text{ for all } \varrho_k \text{ in } \Gamma. \quad (9)$$

A more general definition would permit the use of further base types in place of the negative type *bool*. This requires that the base type in question has a finite set of objects and an interpretation analogous to that of type *bool* in Definition 18. The primary motivation for this definition is the existence of suitable rules for state judgements of variables and the fact that positive types can be interpreted as direct factor systems and negative types as inverse factor systems.

► **Lemma 17.** *The typed term $\Gamma \vdash r : \varrho$ is in the core fragment iff $\Gamma \rightarrow \varrho \in Typ^c$.*

³ This style is basically the de Bruijn level notation of λ -terms, see e.g. [6].

6:16 Potential Infinite Models of Type Theory

This lemma follows easily by induction on $\Gamma \vdash r : \varrho$. Therein $\Gamma \rightarrow \sigma$ is defined by $() \rightarrow \sigma := \sigma$ and $\Gamma.\varrho \rightarrow \sigma := \Gamma \rightarrow (\varrho \rightarrow \sigma)$; the index $C \rightarrow i$ is defined in the same way as $\Gamma \rightarrow \sigma$. We now introduce judgements for states of the form $C \mid r : i$. They require a typing judgement $\Gamma \vdash r : \varrho$ such that $C \rightarrow i \in \mathcal{I}_{\Gamma \rightarrow \varrho}$. The rules are based on the typing rules⁴, whereby $C = (i_0, \dots, i_{n-1})$.

$$\begin{array}{c} \text{(VAR+)} \frac{j \geq i_k \in Idx^+}{C \mid x_k : j} \\ \text{(APP)} \frac{C \mid r : i \rightarrow j \quad C \mid s : i}{C \mid rs : j} \end{array} \qquad \begin{array}{c} \text{(VAR-)} \frac{j \leq i_k \in Idx^-}{C \mid x_k : j} \\ \text{(ABS)} \frac{C.i \mid r : j}{C \mid \lambda x_n^{\varrho} r : i \rightarrow j} \end{array}$$

The condition $j \geq i_k$ in Rule (VAR+) is a consequence of the fact that objects a_{i_k} of positive type exist at all later stages j as $emb_{i_k}^j(a_{i_k})$. The condition $j \leq i_k$ in Rule (VAR-) is related to the fact that we find unique restrictions of a relation, but not unique extensions. Note that for an index $i_k \in Idx^+ \cap Idx^-$ both of the rules for variables apply. In this case, however, there is only one index, so both rules are the same.

A term usually has several state judgement, not only one. Indeed, we expect that there are \mathfrak{D} -many of them and it seems necessary to prove this stronger property in order to guarantee that at least one judgement exists. However, we have not shown that all terms of the core fragment have a state judgement. It is possible to define sets \mathfrak{D}_Γ , similar as in [4], but with a reorder of the context Γ that takes the positive types first, and afterward the negative ones. In a subsequent paper we will present such a definition of sets \mathfrak{D}_Γ satisfying Condition (D) and a proof that each term in the above mentioned fragment has indeed \mathfrak{D} -many state judgements.

3.2 Interpretation of Types

An interpretation of a type ϱ has a static and a dynamic part. Although the limit is uniquely defined from a structural perspective, the extension, i.e., the set of elements in the limit, depends on the stage of the meta-level investigation, assuming a consequent finitistic view. In this case, the limit is not the absolute end of the extensible system. As the investigation progresses, both the system and the limit increase. The main part of the interpretation is the factor system – the limit is necessary to prove that the definition in the factor system is correct.

Let $(\llbracket \varrho \rrbracket_i)_{i \in \mathcal{I}_\varrho}$ be the factor system that interprets type ϱ , which consists of finite sets $\llbracket \varrho \rrbracket_i$. The limit, up to isomorphism, is then $\llbracket \varrho \rrbracket := \mathcal{EL}((\llbracket \varrho \rrbracket_i)_{i \in \mathcal{I}_\varrho})$, see Section 2.4. These limit sets give rise to an extensional type structure if we use (Appl) as application. Recall Lemma 13 and our assumption of Condition (D).

► **Definition 18.** *An interpretation $\llbracket \cdot \rrbracket$ of types assigns to a type ϱ a factor system $(\llbracket \varrho \rrbracket_i)_{i \in \mathcal{I}_\varrho}$ and a limit $\llbracket \varrho \rrbracket$ of it. So the pair $((\llbracket \varrho \rrbracket_i)_{i \in \mathcal{I}_\varrho}, \llbracket \varrho \rrbracket)$ interprets the type ϱ . This interpretation shall satisfy the following properties:*

1. *For a base type ι , $(\llbracket \iota \rrbracket_i)_{i \in \mathcal{I}_\iota}$ is a direct factor system.*
2. *Type $bool$ is interpreted by the factor system $\mathbb{B}_{\{bool\}}$ with $\mathbb{B}_{bool} = \mathbb{B} = \{true, false\}$ and limit \mathbb{B} .*
3. *The interpretation of $\varrho \rightarrow \sigma$ is the factor system $[(\llbracket \varrho \rrbracket_i)_{i \in \mathcal{I}_\varrho} \rightarrow (\llbracket \sigma \rrbracket_j)_{j \in \mathcal{I}_\sigma}]$ together with the limit $\llbracket \varrho \rrbracket \rightarrow_{\mathfrak{D}} \llbracket \sigma \rrbracket$.*

⁴ One may also combine both rules into one. The rules are however more readable if one keeps them separated.

Since Condition (D) holds, it follows from Corollary 15 that the function space $\llbracket \varrho \rrbracket \rightarrow \mathfrak{D} \llbracket \sigma \rrbracket$ is indeed the limit of the underlying factor system $[(\llbracket \varrho \rrbracket_i)_{i \in \mathcal{I}_\varrho} \rightarrow (\llbracket \sigma \rrbracket_j)_{j \in \mathcal{I}_\sigma}]$. Definition 18 extends to contexts Γ in the usual way by taking products.

► **Example 19.** The standard interpretation of the natural numbers $\llbracket nat \rrbracket = \mathbb{N}$ is the limit of the direct factor system $(\llbracket nat \rrbracket_i)_{i \in \mathbb{N}^+}$ with $\llbracket nat \rrbracket_i = \mathbb{N}_i$.⁵ Of course, there are other non-standard models of nat with non-standard natural numbers as well. These non-standard numbers do not appear at the limit step, but at some stage within the system.

The next lemma is required for the reflection principle stated in Theorem 24. Its proof proceeds by induction on Definition 16 and uses Proposition 7 and Corollary 12.

- **Lemma 20.** *Given an interpretation of types $\llbracket \cdot \rrbracket$ and let $i \leq i'$ with $i, i' \in \mathcal{I}_\varrho$ for a type ϱ .*
1. *If $\varrho \in Typ^+$, then the factor system $(\llbracket \varrho \rrbracket_i)_{i \in \mathcal{I}_\varrho}$ is direct and $a \mapsto^p a_i$ implies $a \mapsto^p emb_i^{i'}(a_i)$.*
 2. *If $\varrho \in Typ^-$, then the factor system is inverse and $a \mapsto^p a_{i'}$ implies $a \mapsto^p proj_i^{i'}(a_{i'})$.*

3.3 Interpretation of Terms and a First Version of the Reflection Principle

The interpretation of terms in the type structure of the limit sets is the common interpretation. What is new is the interpretation in the factor system. This part of the interpretation requires a state judgement, so it is defined only for the core fragment, defined in Section 3.1.

► **Definition 21.** *A Γ -environment is a list of elements $\mathbf{A} \in \llbracket \Gamma \rrbracket := \llbracket \varrho_0 \rrbracket \times \cdots \times \llbracket \varrho_{n-1} \rrbracket$ for $\Gamma = (\varrho_0, \dots, \varrho_{n-1})$. The value $\llbracket r \rrbracket_{\mathbf{A}} \in \llbracket \varrho \rrbracket$ of a typed term $\Gamma \vdash r : \varrho$ is defined recursively on the derivation of the judgement $\Gamma \vdash r : \varrho$ relative to a Γ -environment $\mathbf{A} \in \llbracket \Gamma \rrbracket$:*

$$\begin{aligned} \llbracket x_k \rrbracket_{\mathbf{A}} &:= A_k && \text{for } \Gamma \vdash x_k : \varrho_k, \\ \llbracket rs \rrbracket_{\mathbf{A}} &:= \llbracket r \rrbracket_{\mathbf{A}}(\llbracket s \rrbracket_{\mathbf{A}}) && \text{for } \Gamma \vdash r : \varrho \rightarrow \sigma \text{ and } \Gamma \vdash s : \varrho, \\ \llbracket \lambda x_n^{\varrho} r \rrbracket_{\mathbf{A}}(B) &:= \llbracket r \rrbracket_{\mathbf{A}.B} && \text{for } \Gamma.\varrho \vdash r : \sigma \text{ and } B \in \llbracket \varrho \rrbracket. \end{aligned}$$

► **Definition 22.** *A C -environment is a list of elements $\mathbf{a} \in \llbracket \Gamma \rrbracket_C := \llbracket \varrho_0 \rrbracket_{i_0} \times \cdots \times \llbracket \varrho_{n-1} \rrbracket_{i_{n-1}}$ for $C = (i_0, \dots, i_{n-1})$. The value $\llbracket r \rrbracket_{\mathbf{a}.C}^i \in \llbracket \varrho \rrbracket_i$ of a term with $C \mid r : i$ relative to a C -environment $\mathbf{a} \in \llbracket \Gamma \rrbracket_C$ is defined recursively on the derivation of the state judgement $C \mid r : i$:*

$$\begin{aligned} \llbracket x_k \rrbracket_{\mathbf{a}.C}^j &:= emb_{i_k}^j(a_k) && \text{for } C \mid x_k : i_k \text{ and } \varrho_k \in Typ^+, \\ \llbracket x_k \rrbracket_{\mathbf{a}.C}^j &:= proj_j^{i_k}(a_k) && \text{for } C \mid x_k : i_k \text{ and } \varrho_k \in Typ^-, \\ \llbracket rs \rrbracket_{\mathbf{a}.C}^j &:= \llbracket r \rrbracket_{\mathbf{a}.C}^{i \rightarrow j}(\llbracket s \rrbracket_{\mathbf{a}.C}^i) && \text{for } C \mid r : i \rightarrow j \text{ and } C \mid s : i, \\ \llbracket \lambda x_n^{\varrho} r \rrbracket_{\mathbf{a}.C}^{i \rightarrow j}(b) &:= \llbracket r \rrbracket_{\mathbf{a}.b.C.i}^j && \text{for } C.i \mid r : j \text{ and } b \in \llbracket \varrho \rrbracket_i. \end{aligned}$$

By definition, the value $\llbracket r \rrbracket_{\mathbf{a}.C}^i$ depends on the way the judgement $C \mid r : i$ has been derived. So different derivations could lead to different values, making this definition incorrect. The value $\llbracket rs \rrbracket_{\mathbf{a}.C}^j$ seemingly depend on the chosen index i used in $\llbracket s \rrbracket_{\mathbf{a}.C}^i$. A consequence of the main Theorem 24 is that the value $\llbracket r \rrbracket_{\mathbf{a}.C}^i$ is indeed independent (modulo \simeq) of the derivation of the state judgement $C \mid r : i$.

⁵ Here we assume that the interpretation of type nat is the set \mathbb{N} of all natural numbers, which is an actual infinite set if one accepts actual infinities at meta-level. If not, \mathbb{N} is an arbitrary large finite set, which corresponds to a set in the implicitly given factor system at meta-level, i.e., \mathbb{N} is \mathbb{N}_j for some sufficiently large index j depending on the stages of the meta-level investigation.

► **Definition 23.** Given a pair (\mathbf{A}, \mathbf{a}) with a Γ -environment \mathbf{A} and a C -environment \mathbf{a} such that $\mathbf{A} \mapsto^p \mathbf{a}$. The interpretation of a typed term $\Gamma \vdash r : \varrho$ with state judgement $C \mid r : i$ relative to (\mathbf{A}, \mathbf{a}) is the pair $(\llbracket r \rrbracket_{\mathbf{A}}, \llbracket r \rrbracket_{\mathbf{a}:C}^i) \in \llbracket \varrho \rrbracket \times \llbracket \varrho \rrbracket_i$.

One would expect that $\mathbf{A} \mapsto^p \mathbf{a}$ implies $\llbracket r \rrbracket_{\mathbf{A}} \mapsto^p \llbracket r \rrbracket_{\mathbf{a}:C}^i$. Indeed, this is the content of the next theorem, which is also the main theorem.

► **Theorem 24.** Given a typed term $\Gamma \vdash r : \varrho$ with $C \mid r : i$. If $\mathbf{A} \in \llbracket \Gamma \rrbracket$, $\mathbf{a} \in \llbracket \Gamma \rrbracket_C$ are variable assignments with $\mathbf{A} \mapsto^p \mathbf{a}$, then

$$\llbracket r \rrbracket_{\mathbf{A}} \mapsto^p \llbracket r \rrbracket_{\mathbf{a}:C}^i.$$

Proof. The proof proceeds by induction on the derivation of $C \vdash r : i$. For a variable x_k of a positive type we apply Lemma 20. This yields

$$\llbracket x_k \rrbracket_{\mathbf{A}} = A_k \mapsto^p \text{emb}_{i_k}^j(a_k) = \llbracket x_k \rrbracket_{\mathbf{a}:C}^j,$$

since $A_k \mapsto^p a_k$. A similar consideration holds for negative types. For application and abstraction the claim follows from the fact that \mapsto^p is a logical relation. ◀

Consider the case of a closed term $r : \varrho$. Let us assume that ϱ is some data type, then the theorem states that $\llbracket r \rrbracket \mapsto^p \llbracket r \rrbracket^i$. Recall that the predecessor relation \mapsto^p is seen as a directed equality between two states of the same object. We can thus read \mapsto^p , which is nothing more than \mapsto^p applied limits, as an equality between $\llbracket r \rrbracket$ and $\llbracket r \rrbracket^i$. As indicated in Section 1.1, $\llbracket r \rrbracket^i$ is a strong form of an approximation of $\llbracket r \rrbracket$.

Due to the use of logical relations on higher types, application and λ -abstraction respect this equality. Since the type *bool* of propositions is part of the calculus, the equality holds also for truth values. The relation $\llbracket r \rrbracket \mapsto^p \llbracket r \rrbracket^i$ for type *bool*, with the only values *true* and *false* in classical logic, is the identity. Consequently, truth in the limit and truth in a sufficiently large stage of the factor system coincide. Theorem 24 is the main result, and at the same time this theorem is necessary to show that the basic definition of the interpretation of a term, as introduced in Definition 22, is correct.

► **Corollary 25.** Let $\Gamma \vdash r : \varrho$ be a typed term with $i \leq i'$, $C \mid r : i$ and $C \mid r : i'$. Then $\llbracket r \rrbracket_{\mathbf{a}:C}^{i'} \mapsto^p \llbracket r \rrbracket_{\mathbf{a}:C}^i$ holds for all C -environments $\mathbf{a} \in \llbracket \Gamma \rrbracket_C$.

Proof. Define $\mathbf{A} := \text{Emb}_C(\mathbf{a})$. Then $\mathbf{A} \mapsto^p \mathbf{a}$ follows from Condition (Emb), hence $\llbracket r \rrbracket_{\mathbf{A}} \mapsto^p \llbracket r \rrbracket_{\mathbf{a}:C}^{i'}$ and $\llbracket r \rrbracket_{\mathbf{A}} \mapsto^p \llbracket r \rrbracket_{\mathbf{a}:C}^i$ by Theorem 24, and therefore $\llbracket r \rrbracket_{\mathbf{a}:C}^{i'} \mapsto^p \llbracket r \rrbracket_{\mathbf{a}:C}^i$ by Condition (Factor). ◀

The independence of the value $\llbracket r \rrbracket_{\mathbf{a}:C}^i$ (modulo \asymp) from the state judgement follows from Corollary 25 if we take $i = i'$.

4 Conclusion and Further Work

We presented a model that can be used to interpret a fragment of the simply typed λ -calculus (which we called *core fragment*) based on the assumption that infinite sets are potential infinite. We gave a formalization of the potential infinite based on the filters \mathfrak{D} and sets \ll within these filters. This allows one to avoid any notion of actual infinity. The function space in this model is a family of finite function spaces and the model already has an interpretation of the logical type *bool* with the usual two truth values.

We introduced an interpretation of λ -terms from the core fragment which has a dynamic part, the factor system, and a static part, the limit of the factor system. Nevertheless, both parts are necessary, because from a dynamic point of view, the construction of a limit is not the end of the process, but an intermediate state.

The next step is to extend the λ -calculus to constants, in particular to the logical constants of implication and universal quantifier. The challenge here is that the universal quantifier is not continuous and cannot be interpreted as a higher-order functional. The solution is to introduce an additional rule with a side condition $C \ll i$ on the state judgments. The correctness of the interpretation uses the fact that propositions have “stable truth values” in the sense that for each proposition there is a stage in the model where the truth value of the proposition does not change anymore during further extensions. A corresponding interpretation for first-order logic has been given in [4].

The reflection principle from Theorem 24 then states that all objects and propositions (propositions are specific terms of type *bool*) in the limit are reflected in a sufficiently large state in the system. The interpretation is possible on the core fragment of simple type theory, not for all terms. However, this fragment includes a version of a classical higher-order logic. Unlike other models, such as domain theory [1], it does not require (actual) infinite sets at all, and it includes logic.⁶

We do not expect that the theorem about the reflection principle can be extended from the core fragment to all terms of simply typed λ -calculus without some kind of further restrictions. This is because variables of function type correspond to arbitrary functions, which can also be used to define higher-order functions by λ -abstraction. From the perspective of extensibility, this necessitates the consideration of both covariant (with respect to the codomain) and contravariant (with respect to the domain) extensions simultaneously. For instance, the totality of higher-order functions requires that, for each argument, which could be a function f of type $nat \rightarrow nat$, there must be a value. However, f is not given as a single entity. So one must identify a property that guarantees totality and conditions that ensure invariance, that is, existing properties must be preserved by future extensions of f . This is a significant challenge for higher-order functions and it is related to properties of the filter \mathfrak{D} . It is also the reason that for arbitrary functions, there are no simple state judgements as formulated in the rules (VAR+) and (VAR-). A future task will be to identify conditions under which the core fragment can be extended to include function variables.

References

- 1 Samson Abramsky and Achim Jung. Domain theory. In *Handbook of logic in computer science*. Oxford University Press, 1994.
- 2 Hendrik Pieter Barendregt. *The lambda calculus - its syntax and semantics*, volume 103 of *Studies in logic and the foundations of mathematics*. North-Holland, Amsterdam, 1985.
- 3 Alonzo Church. A formulation of the simple theory of types. *The Journal of Symbolic Logic*, 5(2):56–68, 1940. doi:10.2307/2266170.
- 4 Matthias Eberl. A model theory for the potential infinite. *Reports on Mathematical Logic*, 57:3–30, 2022. doi:10.4467/20842589rm.22.001.16658.
- 5 Matthias Eberl. Higher-order concepts for the potential infinite. *Theoretical Computer Science*, 945:113667, 2023. doi:10.1016/j.tcs.2022.12.017.

⁶ The idea of a logic-enriched type theory [7] separates propositions from the underlying type theoretic framework. In our approach the logic is inside type theory, but the interpretation of the universal quantifier has an extra treatment.

- 6 Marcelo P. Fiore, Gordon D. Plotkin, and Daniele Turi. Abstract syntax and variable binding. In *Proc. 14th LICS Conf.*, pages 193–202. IEEE, Computer Society Press, 1999. doi:10.1109/LICS.1999.782615.
- 7 Nicola Gambino and Peter Aczel. The generalised type-theoretic interpretation of constructive set theory. *The Journal of Symbolic Logic*, 71(1):67–103, 2006. doi:10.2178/js1/1140641163.
- 8 John Longley and Dag Normann. *Higher-order computability*, volume 100. Springer, 2015. doi:10.1007/978-3-662-47992-6.
- 9 Saunders MacLane and Ieke Moerdijk. *Sheaves in geometry and logic: A first introduction to topos theory*. Springer Science & Business Media, 2012. doi:10.1007/978-1-4612-0927-0.
- 10 John C. Mitchell and Eugenio Moggi. Kripke-style models for typed lambda calculus. *Annals of Pure and Applied Logic*, 51(1-2):99–124, 1991. doi:10.1016/0168-0072(91)90067-V.
- 11 Rob Nederpelt and Herman Geuvers. *Type theory and formal proof: an introduction*. Cambridge University Press, 2014.
- 12 Gordon Plotkin. *Lambda-definability and logical relations*. Edinburgh University, 1973.
- 13 Anne S. Troelstra. *Choice sequences. A chapter of intuitionistic mathematics. Oxford logic guides*, volume 44. Clarendon Press, 1979.
- 14 Anne S. Troelstra and Dirk van Dalen. Constructivism in mathematics. vol. i, volume 121 of *Studies in Logic and the Foundations of Mathematics*, 26, 1988.