Semantic Consistency Proofs for Systems of Illative Combinatory Logic

(extended abstract)

Łukasz Czajka

January 14, 2015

1 Introduction

Illative systems of combinatory logic or lambda-calculus consist of type-free combinatory logic or lambda-calculus extended with additional constants intended to represent logical notions. In fact, early systems of combinatory logic and lambda calculus (by Schönfinkel [15], Curry [4] and Church [2, 3]) were meant as very simple foundations for logic and mathematics. However, the Kleene-Rosser and Curry paradoxes caused most logicians to abandon this work.

It has proven surprisingly difficult to formulate and show consistent illative systems strong enough to interpret traditional logic. This was accomplished in [1], [13] and [14], where several systems were shown complete for the universal-implicational fragment of first-order intuitionistic predicate logic. In [9] an extension of one system from [1, 13, 14] in which full higher-order classical logic may be interpreted was shown consistent by semantic methods.

The difficulty in proving consistency of illative systems in essence stems from the fact that, lacking a type regime, arbitrary recursive definitions involving logical operators may be formulated, including negative ones. In early systems containing an unrestricted implication introduction rule this was the reason for the Curry's paradox. Formulating appropriate and not too cumbersome restrictions is not easy if the fundamental property of allowing unrestricted recursion is to be retained.

1.1 Contribution

In this work we develop semantics for various systems of illative combinatory logic and lambda-calculus which are extensions of some systems from [1, 13, 14, 9]. The systems are then shown consistent by constructing models. We also consider natural embeddings of traditional logical systems into corresponding illative systems. Using semantic methods, we investigate soundness and completeness of these translations.

Some of the systems shown consistent in the present work are much stronger than the systems of [1, 13, 14]. In particular, the strongest of our systems essentially incorporates full extensional classical higher-order logic extended with dependent function types, dependent sums, subtypes and W-types.

In most previous work the approach is syntactic – consistency is shown by cut-elimination or by analysis of possible forms of derivable terms using grammars. Establishing cut-elimination is more informative than only constructing a model, but for illative systems it also seems much harder. Our methods are semantic. The consistency proofs are not constructive and need much of the power of set theory. In fact, the model construction for the strongest of our systems assumes the existence of a strongly inaccessible cardinal, so it is not formalisable in ZFC.

1.2 Motivation

From the point of view of computer science, an interesting feature of illative systems is that by basing on the untyped lambda-calculus (combinatory logic) they incorporate general recursion into the logic. Therefore, unrestricted recursive definitions may be formulated directly, including definitions of possibly non-terminating partial functions. In [10, 11] it has been suggested that this feature of illative combinatory logic makes it potentially interesting as a logic for an interactive theorem prover intended to be used for program verification.

Most popular proof assistants allow only total functions, and totality must be ensured by the user, either by very precise specifications of function domains, restricting recursion in a way that guarantees termination, explicit well-foundedness proofs, or other means.

An advantage of illative systems is that no justifications are needed for formulating unrestricted recursive definitions. One may just introduce a possibly non-well-founded recursive function definition and start reasoning about it within the logic. There is obviously a trade-off – some inference rules need to be restricted by adding premises which essentially state that some terms are "propositions". To be able to derive that some terms are propositions, illative systems include certain "typing rules", i.e., rules for reasoning about which types (categories) a term belongs to. In contrast to traditional systems, however, these rules are internal to the system. The functions do not need to be "typed" a priori, but reasoning about "types" may be interleaved with other reasoning. For instance, one may show typability by induction. This may possibly be an interesting way of reasoning about potentially non-well-founded function definitions in an interactive theorem prover.

2 Illative combinatory logic

The illative systems we consider come in three variants differing in the underlying reduction system, which is either combinatory logic with weak reduction, (untyped) lambda-calculus with β-reduction or lambda-calculus with $\beta\eta$-reduction, with constants from a fixed signature Σ. Since most of the proofs and definitions are the same or very similar for each of the variants, we usually give only a single generic proof or definition, and possibly note the differences for each variant. We use $\mathbb{T}$ to generically denote the set of terms of an illative system, which is either the set of terms of combinatory logic with extra constants from Σ ($\mathbb{T}_{\mathrm{CL}}(\Sigma)$) or the set of terms of lambda-calculus with constants from Σ ($\mathbb{T}_{\lambda}(\Sigma)$). Analogously, we use $=$ to generically denote $=_w$, $=_\beta$ or $=_{\beta\eta}$, as appropriate. By $\equiv$ we denote syntactic identity of terms (up to α-conversion in lambda-calculus). We use S and K to generically denote either the constants of combinatory logic, or the terms $\lambda xyz.xz(yz)$ and $\lambda xy.x$ in lambda-calculus. We define $\mathsf{I} \equiv \lambda x.x$ in lambda-calculus, or $\mathsf{I} \equiv \mathsf{SKK}$ in combinatory logic. The notation $\lambda x.M$ is used to denote either combinatory abstraction in CL, or abstraction in lambda-calculus. We set $\pi \equiv \lambda xyz.zxy$, $\pi_1 \equiv \lambda x.x\mathsf{K}$ and $\pi_2 \equiv \lambda x.x(\mathsf{KI})$.

Illative systems extend combinatory logic (or lambda-calculus) with *illative primitives* representing logical notions. Unlike in most traditional systems of logic, there is no a priori distinction between various categories: propositions (formulas), individual terms, functions, relations, etc. Instead, there are inference rules which allow some categorisations to be performed *inside* the system. Certain illative primitives represent primitive types[1] (categories), and there are combinators which allow the formation of new types. If a term T represents a type, then TX is an assertion that X has type T. In fact, any term may be potentially asserted as a proposition (which does not mean that all terms represent well-formed propositions), and equal terms (in the sense of weak, β-, or $\beta\eta$-equality, as appropriate) are always interchangeable. Intuitively, types represent permissible quantifier ranges – quantification is allowed only over elements of a fixed type. Predicates on a type T, or subsets of T, are represented by functions from T to the type of propositions H.

The illative primitives need not be constants – they may be composite terms. An illative primitive which is a constant is called an *illative constant*. Below we list some common illative primitives together with an informal explanation of their meaning (cf. [6, §12B2]). Any given illative system may contain any number of these primitives, and possibly some more. In what follows, by $X, Y, Z, \ldots$ we denote arbitrary terms from $\mathbb{T}$.

P Implication. Instead of $\mathsf{P}XY$ we often write $X \supset Y$. Implication is sometimes defined by $\mathsf{P} \equiv \lambda xy.\Xi(\mathsf{K}x)(\mathsf{K}y)$ (see below for an explanation of Ξ).

$\mathsf{\Lambda}$ Conjunction. Instead of $\mathsf{\Lambda}XY$ we often write $X \wedge Y$.

V Disjunction. Instead of $\mathsf{V}XY$ we often write $X \vee Y$.

$\neg$ Negation.

$\bot$ False proposition.

$\top$ True proposition. Often defined by $\top \equiv \mathsf{P}\bot\bot$.

Ξ Restricted generality – a restricted universal quantifier. The term ΞXY is intuitively interpreted as "$X \subseteq Y$", or "for every object Z such that XZ we have YZ", or "for every object Z of type X we have YZ". The notation $\forall x : X . Y$ is often used to denote $\Xi X(\lambda x.Y)$. Note that x is not bound in X.

X Restricted existential quantifier. The term $\mathsf{X}YZ$ is intuitively interpreted as "there is an object X such that YX and ZX", or "there exists an object X of type Y such

[1] The notion of "type" is used informally in this section, interchangeably with "category".

that ZX". The notation $\exists x : Y.\, Z$ is often used to denote $\mathsf{X}Y\lambda x.Z$. Note that x is not bound in Y.

F Functionality (cf. [5, §8C]). The term $\mathsf{F}XYF$ is intuitively interpreted as "F is a function from X to Y", or "for every object Z of type X we have $Y(FZ)$". Functionality is often defined by $\mathsf{F} \equiv \lambda xyf.\Xi x(\lambda z.y(fz))$.

G Dependent functionality. The term $\mathsf{G}XYF$ is intuitively interpreted as "F is a dependent function which for each Z of type X gives an object of type YZ", or "for every object Z of type X we have $YZ(FZ)$". Dependent functionality is often defined by $\mathsf{G} \equiv \lambda xyf.\Xi x(\lambda z.yz(fz))$.

F_n Functionality of n arguments. The term $\mathsf{F}_n X_1 \ldots X_n Y F$ is intuitively interpreted as "F is an n-argument function from $X_1, \ldots, X_n$ to Y". Usually F_n is defined inductively as follows:

$$\begin{aligned} \mathsf{F}_0 &\equiv \mathsf{I} \\ \mathsf{F}_{n+1} &\equiv \lambda x_1 \ldots x_{n+1} y.\mathsf{F} x_1(\mathsf{F}_n x_2 \ldots x_{n+1} y) \end{aligned}$$

H Type of propositions. The term $\mathsf{H}X$ is intuitively interpreted as "X is a proposition". The type of propositions is sometimes defined by $\mathsf{H} \equiv \lambda x.\mathsf{P}xx$ or by $\mathsf{H} \equiv \lambda x.\mathsf{L}(\mathsf{K}x)$.

L Category of types. The term $\mathsf{L}X$ is intuitively interpreted as "X is a type" or "X represents a permissible range of quantification". The category of types is sometimes defined by $\mathsf{L} \equiv \lambda x.\Xi xx$.

A Type of individuals.

O The empty type. Often defined by $\mathsf{O} \equiv \mathsf{K}\bot$.

ϵ Choice operator. A term ϵAX is interpreted as some unspecified object of type A satisfying the predicate X, if such an object exists.

Υ Subtype constructor. A term ΥAX is interpreted as the subtype of A consisting of all objects Y of type A such that XY. Subtype constructor is often defined by $\Upsilon \equiv \lambda xyz.xz \wedge yz$.

Σ Dependent sum constructor. A term ΣABX is intuitively interpreted as: "$\pi_1 X$ has type A, and $\pi_2 X$ has type $B(\pi_1 X)$". Dependent sum constructor is often defined by $\Sigma \equiv \lambda xyz.x(\pi_1 z) \wedge y(\pi_1 z)(\pi_2 z)$.

W W-type constructor. A term $\mathsf{W}AB$ is interpreted as a W-type: the type of all well-founded trees with nodes labelled with objects of the type A and branching specified by Ba for a node labelled with a, i.e., a node labelled with a has a distinct child for each object of type Ba.

Using illative primitives, it is possible to interpret ordinary logic in illative combinatory logic. For instance, a first-order sentence

$$\forall x(r(x) \to s(f(x), g(x)) \wedge r(f(x)))$$

is translated as the statement

$$\forall x : \mathsf{A} \,.\, rx \supset s(fx)(gx) \wedge r(fx)$$

which is

$$\Xi\mathsf{A}(\lambda x.\mathsf{P}(rx)(\mathsf{\Lambda}(s(fx)(gx))(r(fx))))$$

where r, s, f, g are constants corresponding to the relation and function symbols from the first-order language, and A represents the first-order universe.

In an illative system judgements have the form $\Gamma \vdash X$ where Γ is a finite set of terms and X is a term. If X is a term and Γ a set of terms, then by Γ, X we denote $\Gamma \cup \{X\}$. For an infinite set of terms Γ we write $\Gamma \vdash X$ if there exists a finite subset $\Gamma' \subseteq \Gamma$ with $\Gamma' \vdash X$.

All illative systems are required to include the following axiom (Ax) and the rule (Eq) (cf. the definition of $\mathcal{F}_0$ in [5, §8E]). The rule (Eq) essentially incorporates unrestricted recursion into the system.

$$\frac{}{\Gamma, X \vdash X}\ (\mathrm{Ax}) \qquad \frac{\Gamma \vdash X \quad X = Y}{\Gamma \vdash Y}\ (\mathrm{Eq})$$

Here $X = Y$ is a meta-level side condition. Recall that $=$ denotes either weak, β-, or $\beta\eta$-equality, as appropriate.

If an illative system includes one of the illative primitives P, Ξ, F, G, then we require that it incorporates the corresponding elimination rules (either directly or as derived rules).

$$\frac{\Gamma \vdash X \supset Y \quad \Gamma \vdash X}{\Gamma \vdash Y}\ (\mathsf{P}\mathrm{E}) \qquad \frac{\Gamma \vdash \Xi XY \quad \Gamma \vdash XZ}{\Gamma \vdash YZ}\ (\Xi\mathrm{E})$$

$$\frac{\Gamma \vdash \mathsf{F}XYF \quad \Gamma \vdash XZ}{\Gamma \vdash Y(FZ)}\ (\mathsf{F}\mathrm{E}) \qquad \frac{\Gamma \vdash \mathsf{G}XYF \quad \Gamma \vdash XZ}{\Gamma \vdash YZ(FZ)}\ (\mathsf{G}\mathrm{E})$$

It is less clear how introduction rules should look like. Curry's paradox implies that adding the following natural candidate for an introduction rule for P yields an inconsistent system.

$$\frac{\Gamma, X \vdash Y}{\Gamma \vdash X \supset Y}\ (\mathrm{DED})$$

Intuitively, the problem is that, a priori, we do not know whether X is a proposition, so $X \supset Y$ may not make any sense. If $X = (X \supset Y)$ then using the above rule we can derive a contradiction.

A way out of the paradox is to add the illative primitive H, appropriately restrict introduction rules, and add rules to reason about which terms represent propositions. Of course, we would like the restrictions in introduction rules to be as unobtrusive as possible. It would not be difficult to formulate and show consistent an "illative" system in which the restrictions would be so strong as to make it indistiguishable in practice from a system in which terms are a priori assigned to definite syntactic categories (or typed statically), but the point of introducing such a system is dubious.

3 Overview of the results

3.1 Illative systems

We shall now give an overview of the illative systems studied and shown consistent in the thesis. For the sake of brevity, we do not present all rules of the systems, and some of the rules given here actually differ slightly from the ones used in the thesis. The purpose of this section is to provide the reader with a general idea of how these systems look like and what their essential features are.

We study four main systems of illative combinatory logic: the propositional system $\mathcal{I}\mathrm{Kp}$, the first-order system $\mathcal{I}\mathrm{K}$, the higher-order system $e\mathcal{I}\mathrm{K}\omega$, and the extended higher-order system $\mathcal{I}^+$. All these systems are classical. We also study the intuitionistic variant $\mathcal{I}\mathrm{Jp}$ (resp. $\mathcal{I}\mathrm{J}$) of $\mathcal{I}\mathrm{Kp}$ (resp. $\mathcal{I}\mathrm{K}$), and an intensional variant $\mathcal{I}\mathrm{K}\omega$ of $e\mathcal{I}\mathrm{K}\omega$. The system $\mathcal{I}\mathrm{Jp}$ contains the illative primitives $\mathsf{P}, \mathsf{V}, \mathsf{\Lambda}, \neg, \top, \bot, \mathsf{H}$. Most of the rules of $\mathcal{I}\mathrm{Jp}$ are shown in Figure 1. The system $\mathcal{I}\mathrm{J}$ extends $\mathcal{I}\mathrm{Jp}$ by the illative primitives $\Xi, \mathsf{X}, \mathsf{A}$ and, among others, the rules from Figure 2. The system $\mathcal{I}\mathrm{Kp}$ (resp. $\mathcal{I}\mathrm{K}$) extends $\mathcal{I}\mathrm{Jp}$ (resp. $\mathcal{I}\mathrm{J}$) by the rule of excluded middle:

$$\frac{\Gamma \vdash \mathsf{H}X}{\Gamma \vdash X \vee \neg X} \ (\mathrm{EM})$$

The system $\mathcal{I}\mathrm{K}\omega$ extends the system $\mathcal{I}\mathrm{K}$ by the rules (HL) and (FL) from Figure 3. The system $e\mathcal{I}\mathrm{K}\omega$ extends $\mathcal{I}\mathrm{K}\omega$ by (Ext_f) and (Ext_b). The system $\mathcal{I}^+$ extends $\mathcal{I}\mathrm{K}$ by all the rules from Figure 3, and a few other rules which we omit.

In Figure 3 we use the abbreviations for O, F, G, Υ and Σ presented in the previous section. We also use the notation $X =_A Y$ for $\forall p : \mathsf{F}A\mathsf{H} \,.\, pX \supset pY$, which represents Leibniz equality in type A. In the rule (WInd) we assume $x, y, z \notin \mathrm{FV}(\Gamma, A, B, Z)$.

The rules omitted from this summary are mostly additional rules concerning H, which make the logics complete w.r.t. appropriate semantics. Also, some rules omitted from the summary of $\mathcal{I}^+$ make it possible to derive suitable unrestricted induction rules for inductive types defined using W-types.

3.2 Semantics

In this section we outline the semantics for our illative systems. The models for intuitionistic systems are essentially a combination of a combinatory algebra with a Kripke frame. For classical systems, the models are combinatory algebras with two sets $\mathcal{T}$ and $\mathcal{F}$ of true and false elements of the algebra. Some natural conditions are imposed on $\mathcal{T}$ and $\mathcal{F}$. In the thesis, we prove that all systems are sound w.r.t. the corresponding semantics. The systems $\mathcal{I}\mathrm{Jp}$, $\mathcal{I}\mathrm{Kp}$ and $\mathcal{I}\mathrm{J}$ are also shown to be complete. The system $\mathcal{I}\mathrm{K}$ is shown complete w.r.t. a sligthly less natural class of models, which essentially combine combinatory algebras with some special Kripke frames.

Here we shall give detailed definitions of the models only for the simplest cases of the semantics for $\mathcal{I}\mathrm{Jp}$ and $\mathcal{I}\mathrm{Kp}$ based on combinatory logic with weak equality.

$$\frac{}{\Gamma, X \vdash X}\ (\mathrm{Ax}) \qquad \frac{}{\Gamma \vdash \mathsf{H}\bot}\ (\bot\mathsf{HI})$$

$$\frac{\Gamma, X \vdash Y \quad \Gamma \vdash \mathsf{H}X}{\Gamma \vdash X \supset Y}\ (\mathsf{PI}) \qquad \frac{\Gamma \vdash X \quad \Gamma \vdash X \supset Y}{\Gamma \vdash Y}\ (\mathsf{P}\mathrm{E})$$

$$\frac{\Gamma, X \vdash \mathsf{H}Y \quad \Gamma \vdash \mathsf{H}X}{\Gamma \vdash \mathsf{H}(X \supset Y)}\ (\mathsf{PHI})$$

$$\frac{\Gamma \vdash X \quad \Gamma \vdash Y}{\Gamma \vdash X \wedge Y}\ (\mathsf{\Lambda}\mathrm{I}) \qquad \frac{\Gamma \vdash X \wedge Y}{\Gamma \vdash X}\ (\mathsf{\Lambda}\mathrm{E}_l) \quad \frac{\Gamma \vdash X \wedge Y}{\Gamma \vdash Y}\ (\mathsf{\Lambda}\mathrm{E}_r)$$

$$\frac{\Gamma \vdash \mathsf{H}X \quad \Gamma, X \vdash \mathsf{H}Y}{\Gamma \vdash \mathsf{H}(X \wedge Y)}\ (\mathsf{\Lambda HI}_l) \qquad \frac{\Gamma \vdash \mathsf{H}Y \quad \Gamma, Y \vdash \mathsf{H}X}{\Gamma \vdash \mathsf{H}(X \wedge Y)}\ (\mathsf{\Lambda HI}_r)$$

$$\frac{\Gamma \vdash X}{\Gamma \vdash X \vee Y}\ (\mathsf{VI}_l) \quad \frac{\Gamma \vdash Y}{\Gamma \vdash X \vee Y}\ (\mathsf{VI}_r) \qquad \frac{\Gamma \vdash X \vee Y \quad \Gamma, X \vdash Z \quad \Gamma, Y \vdash Z}{\Gamma \vdash Z}\ (\mathsf{V}\mathrm{E})$$

$$\frac{\Gamma \vdash \mathsf{H}X \quad \Gamma \vdash \mathsf{H}Y}{\Gamma \vdash \mathsf{H}(X \vee Y)}\ (\mathsf{VHI})$$

$$\frac{\Gamma \vdash X}{\Gamma \vdash \mathsf{H}X}\ (\mathsf{HI}) \qquad \frac{\Gamma \vdash \bot}{\Gamma \vdash X}\ (\bot\mathrm{E})$$

$$\frac{\Gamma \vdash X \quad X = Y}{\Gamma \vdash Y}\ (\mathrm{Eq})$$

Figure 1: Basic rules

$$\frac{\Gamma, Xx \vdash Yx \quad \Gamma \vdash \mathsf{L}X \quad x \notin \mathrm{FV}(\Gamma, X, Y)}{\Gamma \vdash \Xi XY}\ (\Xi\mathrm{I}) \qquad \frac{\Gamma \vdash \Xi XY \quad \Gamma \vdash XZ}{\Gamma \vdash YZ}\ (\Xi\mathrm{E})$$

$$\frac{\Gamma, Xx \vdash \mathsf{H}(Yx) \quad \Gamma \vdash \mathsf{L}X \quad x \notin \mathrm{FV}(\Gamma, X, Y)}{\Gamma \vdash \mathsf{H}(\Xi XY)}\ (\Xi\mathsf{HI}) \qquad \frac{}{\Gamma \vdash \mathsf{LA}}\ (\mathsf{AL})$$

$$\frac{\Gamma \vdash MZ \quad \Gamma \vdash NZ \quad \Gamma \vdash \mathsf{L}M}{\Gamma \vdash \mathsf{X}MN}\ (\mathsf{XI})$$

$$\frac{\Gamma \vdash \mathsf{X}MN \quad \Gamma, Mx, Nx \vdash Z \quad x \notin \mathrm{FV}(\Gamma, M, N, Z)}{\Gamma \vdash Z}\ (\mathsf{XE})$$

$$\frac{\Gamma, Mx \vdash \mathsf{H}(Nx) \quad \Gamma \vdash \mathsf{L}M \quad x \notin \mathrm{FV}(\Gamma, M, N)}{\Gamma \vdash \mathsf{H}(\mathsf{X}MN)}\ (\mathsf{XHI})$$

Figure 2: Rules for quantifiers (Ξ and X)

A *propositional illative combinatory algebra* (PICA) is a tuple

$$\mathcal{C} = \langle C, \cdot, \mathsf{k}, \mathsf{s}, \mathsf{h}, \mathsf{p}, \wedge, \vee, \neg, \bot \rangle$$

where $\langle C, \cdot, \mathsf{k}, \mathsf{s} \rangle$ is a combinatory algebra and $\mathsf{h}, \mathsf{p}, \wedge, \vee, \neg, \bot \in C$, i.e., it is simply a combinatory algebra with distinguished elements $\mathsf{h}, \mathsf{p}, \wedge, \vee, \neg, \bot$. Given a PICA $\mathcal{C}$ we often confuse $\mathcal{C}$ with C.

An $\mathcal{I}$Jp-*model* is a tuple $\mathcal{S} = \langle \mathcal{C}, I, S, \leq, \sigma_0, \sigma_1 \rangle$ where:

- $\mathcal{C}$ is a propositional illative combinatory algebra satisfying $\mathsf{h} \cdot a = \mathsf{p} \cdot a \cdot a$ and $\neg \cdot a = \mathsf{p} \cdot a \cdot \bot$ for any $a \in \mathcal{C}$,
- I is a function from Σ to $\mathcal{C}$ providing an interpretation for constants,
- S is a non-empty set of states,
- $\leq$ is a partial order on S,
- σ_0 and σ_1 are functions from $\mathcal{C}$ to $\mathbb{P}(S)$, satisfying the following for any $a, b \in \mathcal{C}$, where $\sigma_h(a) = \sigma_0(a) \cup \sigma_1(a)$:
    1. $\sigma_h(a)$ and $\sigma_1(a)$ are upward-closed[2] wrt. $\leq$,
    2. $\sigma_0(\bot) = S$,
    3. $\sigma_0(a) \cap \sigma_1(a) = \emptyset$,
    4. $\sigma_1(\vee \cdot a \cdot b) = \sigma_1(a) \cup \sigma_1(b)$,
    5. $\sigma_0(\vee \cdot a \cdot b) = \sigma_0(a) \cap \sigma_0(b)$,

[2] A set $A \subseteq S$ is upward-closed wrt. $\leq$ iff $s \in A$ and $s' \geq s$ imply $s' \in A$.

$$\frac{}{\Gamma \vdash \mathsf{LH}}\ (\mathsf{HL}) \qquad \frac{\Gamma \vdash \mathsf{L}X \quad \Gamma, Xx \vdash \mathsf{L}Y \quad x \notin \mathrm{FV}(\Gamma, X, Y)}{\Gamma \vdash \mathsf{L}(\mathsf{F}XY)}\ (\mathsf{FL})$$

$$\frac{\Gamma \vdash \forall x : A \,.\, Xx =_B Yx \quad x \notin \mathrm{FV}(X, Y, A, B)}{\Gamma \vdash X =_{\mathsf{F}AB} Y}\ (\mathrm{Ext}_f)$$

$$\frac{\Gamma \vdash X \supset Y \quad \Gamma \vdash Y \supset X}{\Gamma \vdash X =_H Y}\ (\mathrm{Ext}_b)$$

$$\frac{\Gamma \vdash \mathsf{L}A \quad \Gamma, Ax \vdash \mathsf{L}(Bx) \quad x \notin \mathrm{FV}(\Gamma, A, B)}{\Gamma \vdash \mathsf{L}(\mathsf{G}AB)}\ (\mathsf{GL})$$

$$\frac{\Gamma \vdash \forall x : A \,.\, Xx =_{Bx} Yx \quad x \notin \mathrm{FV}(X, Y, A, B)}{\Gamma \vdash X =_{\mathsf{G}AB} Y}\ (\mathrm{Ext}_g)$$

$$\frac{}{\Gamma \vdash \mathsf{LO}}\ (\mathsf{OL}) \qquad \frac{\Gamma \vdash \mathsf{L}A \quad \Gamma \vdash \mathsf{F}A\mathsf{L}B}{\Gamma \vdash \mathsf{L}(\Sigma AB)}\ (\Sigma\mathsf{L})$$

$$\frac{\Gamma \vdash \mathsf{L}A \quad \Gamma \vdash \mathsf{F}A\mathsf{L}B}{\Gamma \vdash \mathsf{L}(\mathsf{W}AB)}\ (\mathsf{WL}) \qquad \frac{\Gamma \vdash \mathsf{L}A \quad \Gamma \vdash \mathsf{F}A\mathsf{H}X}{\Gamma \vdash \mathsf{L}(\Upsilon AX)}\ (\Upsilon\mathsf{L})$$

$$\frac{\Gamma \vdash \mathsf{X}AX \quad \Gamma \vdash \mathsf{F}A\mathsf{H}X}{\Gamma \vdash X(\epsilon AX)}\ (\epsilon\mathrm{I}_l) \qquad \frac{\Gamma \vdash \mathsf{X}AA \quad \Gamma \vdash \mathsf{F}A\mathsf{H}X}{\Gamma \vdash A(\epsilon AX)}\ (\epsilon\mathrm{I}_r)$$

$$\frac{\Gamma \vdash A(\pi_1 X) \quad \Gamma \vdash \mathsf{F}(B(\pi_1 X))(\mathsf{W}AB)(\pi_2 X)}{\Gamma \vdash \mathsf{W}ABX}\ (\mathsf{WI})$$

$$\frac{\Gamma \vdash \mathsf{W}ABX}{\Gamma \vdash A(\pi_1 X)}\ (\mathsf{WE}_1) \qquad \frac{\Gamma \vdash \mathsf{W}ABX}{\Gamma \vdash \mathsf{F}(B(\pi_1 X))(\mathsf{W}AB)(\pi_2 X)}\ (\mathsf{WE}_2)$$

$$\frac{\Gamma \vdash \mathsf{L}(\mathsf{W}AB) \quad \Gamma, Ax, \mathsf{F}(Bx)(\mathsf{W}AB)y, \forall z{:}Bx.X(yz) \vdash X(\mathrm{sup}(\mathsf{W}AB)xy)}{\Gamma \vdash \Xi(\mathsf{W}AB)X}\ (\mathsf{WInd})$$

(in (WInd) we assume $x, y, z \notin \mathrm{FV}(\Gamma, A, B, X)$)

$$\frac{\Gamma \vdash \pi_1 X =_A \pi_1 Y \quad \Gamma \vdash \pi_2 X =_{B(\pi_1 X)} \pi_2 Y \quad \Gamma \vdash \mathsf{L}(\Sigma AB)}{\Gamma \vdash X =_{\Sigma AB} Y}\ (\mathrm{Ext}_s)$$

Figure 3: Additional rules

6. $\sigma_1(\mathsf{\wedge} \cdot a \cdot b) = \sigma_1(a) \cap \sigma_1(b)$,
7. $s \in \sigma_0(\mathsf{\wedge} \cdot a \cdot b)$ iff
   - $s \in \sigma_0(a)$ and for every $s' \geq s$ such that $s' \in \sigma_1(a)$ we have $s' \in \sigma_h(b)$, or
   - $s \in \sigma_0(b)$ and for every $s' \geq s$ such that $s' \in \sigma_1(b)$ we have $s' \in \sigma_h(a)$,
8. $s \in \sigma_1(\mathsf{p} \cdot a \cdot b)$ iff
   - $s \in \sigma_h(a)$ and for every $s' \geq s$ such that $s' \in \sigma_1(a)$ we have $s' \in \sigma_1(b)$, or
   - $s \in \sigma_1(b)$,
9. $s \in \sigma_0(\mathsf{p} \cdot a \cdot b)$ iff
   - $s \in \sigma_h(a)$, and
   - for every $s' \geq s$ such that $s' \in \sigma_1(a)$ we have $s' \in \sigma_h(b)$, and
   - there exists $s' \geq s$ such that $s' \in \sigma_1(a)$ and $s' \in \sigma_0(b)$.

Intuitively, $s \in \sigma_1(a)$ means that a is known to be a true proposition in state s, and $s \in \sigma_0(a)$ means that in state s, the element a is known to be a proposition which is not (known/forced to be) true. So $s \in \sigma_h(a) = \sigma_0(a) \cup \sigma_1(a)$ means that a is known to be a proposition in state s. Thus, if $s \in \sigma_0(a)$ then we may have $s' \in \sigma_1(a)$ for some $s' \geq s$. A proposition which is not true may become true with expanding our knowledge. However, if $s \in \sigma_0(a)$ then $s' \in \sigma_0(a) \cup \sigma_1(a)$ for all $s' \geq s$, because knowledge is monotonous – once we know a is a proposition it will be a proposition in any future state of knowledge. If a is a proposition which is not true, then in any future state, it may either remain so, or become true. That a is false in state s is expressed by $s \in \sigma_1(\mathsf{p} \cdot a \cdot \bot)$, i.e., that its negation is true, not by $s \in \sigma_0(a)$. A proposition is false in state s if it is a proposition which is not true in all states $s' \geq s$. If $s \in \sigma_h(a)$, i.e., a is a proposition in state s, then a is "always ultimately knowable", i.e., however we expand our knowlege, it is always possible to expand it further so that a becomes either true or false.

Note that the conditions on σ_1 and σ_0 above are not a definition of σ_1 or σ_0, but just some properties we wish σ_1 and σ_0 to satisfy. Because of the combinatory completeness of $\mathcal{C}$, it is not obvious that there exists a structure satisfying the above requirements.

An $\mathcal{I}$Kp-*model* is an $\mathcal{I}$Jp-model with exactly one state s_0. For an $\mathcal{I}$Kp-model we use the abbreviations $\mathcal{T} = \{a \mid s_0 \in \sigma_1(a)\}$ and $\mathcal{F} = \{a \mid s_0 \in \sigma_0(a)\}$. Note that a PICA $\mathcal{C}$ and the sets $\mathcal{T}$ and $\mathcal{F}$ uniquely determine an $\mathcal{I}$Kp-model. We reformulate in terms of $\mathcal{T}$ and $\mathcal{F}$ the conditions on σ_0 and σ_1:

1. $\bot \in \mathcal{F}$,
2. $\mathcal{T} \cap \mathcal{F} = \emptyset$.
3. $\mathsf{v} \cdot a \cdot b \in \mathcal{T}$ iff $a \in \mathcal{T}$ or $b \in \mathcal{T}$,
4. $\mathsf{v} \cdot a \cdot b \in \mathcal{F}$ iff $a \in \mathcal{F}$ and $b \in \mathcal{F}$,
5. $\mathsf{\wedge} \cdot a \cdot b \in \mathcal{T}$ iff $a \in \mathcal{T}$ and $b \in \mathcal{T}$,
6. $\mathsf{\wedge} \cdot a \cdot b \in \mathcal{F}$ iff $a \in \mathcal{F}$ or $b \in \mathcal{F}$,
7. $\mathsf{p} \cdot a \cdot b \in \mathcal{T}$ iff $a \in \mathcal{F}$ or $b \in \mathcal{T}$,
8. $\mathsf{p} \cdot a \cdot b \in \mathcal{F}$ iff $a \in \mathcal{T}$ and $b \in \mathcal{F}$.

The notions of $\mathcal{I}$J-models and $\mathcal{I}$K-models are defined in a similar way to $\mathcal{I}$Jp-models and $\mathcal{I}$Kp-models, respectively. In an analogous way one also defines $\mathcal{I}$Kω-, $e\mathcal{I}$Kω- and $\mathcal{I}^+$-models. We define Kripke $\mathcal{I}$K-models as $\mathcal{I}$J-models satisfying the law of excluded middle for any state s and element a: if $s \in \sigma_h(a)$ then $s \in \sigma_1(a)$ or $s \in \sigma_1(\mathsf{p} \cdot a \cdot \bot)$.

We define the notations $\Gamma \models_{\mathcal{I}\mathrm{Jp}} X$, $\Gamma \models_{\mathcal{I}\mathrm{Kp}} X$, $\Gamma \models_{\mathcal{I}\mathrm{J}} X$, etc., in the standard way. We use $\models_{k\mathcal{I}\mathrm{K}}$ to denote the semantic consequence relation with respect to Kripke $\mathcal{I}$K-models. Theorems 4.1.8, 4.1.11, 4.1.14, 4.1.16, 5.1.7, 5.1.11, 5.1.13, 5.1.15, 5.1.16, 6.1.6 and 7.1.9 in the thesis may be combined into the following result.

Theorem 1 (Soundness and completeness with respect to the semantics)**.**

1. *If $\mathcal{I}$ is one of $\mathcal{I}$*Jp*, $\mathcal{I}$*Kp *or $\mathcal{I}$*J*, then the condition $\Gamma \vdash_{\mathcal{I}} X$ is equivalent to $\Gamma \models_{\mathcal{I}} X$, i.e., the system $\mathcal{I}$ is sound and complete with respect to the corresponding semantics.*
2. *The condition $\Gamma \vdash_{\mathcal{I}\mathrm{K}} X$ is equivalent to $\Gamma \models_{k\mathcal{I}\mathrm{K}} X$, i.e., the system $\mathcal{I}$*K *is sound and complete with respect to the semantics based on Kripke $\mathcal{I}$*K*-models.*
3. *If $\mathcal{I}$ is one of $\mathcal{I}$*K*, $\mathcal{I}$*K*ω, $e\mathcal{I}$*K*ω or $\mathcal{I}^+$, then the condition $\Gamma \vdash_{\mathcal{I}} X$ implies $\Gamma \models_{\mathcal{I}} X$, i.e., the system $\mathcal{I}$ is sound with respect to the corresponding semantics.*

For classical illative systems with quantifiers, the standard Henkin-style completeness proof cannot be easily adapted, essentially because of the fact that we have excluded middle only for terms X for which $\mathsf{H}X$ is provable. This is why we prove only soundness for the systems $\mathcal{I}$K, $\mathcal{I}$Kω, $e\mathcal{I}$Kω and $\mathcal{I}^+$. The system $\mathcal{I}$K is complete with respect to a modified classical semantics (Kripke $\mathcal{I}$K-models) which allows more than one state.

3.3 Translations

In the thesis we show translations of traditional systems of logic into corresponding illative systems. We prove all those translations to be sound, i.e., if a judgement of a traditional system is provable, then so is its translation. For $\mathcal{I}$Jp, $\mathcal{I}$Kp, $\mathcal{I}$J and $\mathcal{I}$K we also show the translations complete, i.e., if the translation of a judgement is provable, then so is the original judgement. For $\mathcal{I}$Kω and $e\mathcal{I}$Kω we derive a limited completeness result: if a translated judgement of higher-order logic is provable in $e\mathcal{I}$Kω then it is valid in all standard models for higher-order logic. The proofs of these results are done semantically, by showing a truth-preserving transformation of models of illative systems into models of corresponding traditional systems, and vice versa.

To give the reader a general idea of how the translations look like, we provide a definition of the translation from classical first-order logic into $\mathcal{I}$K. The sole definitions of the translations are similar to those in [1, 13, 14]. We assume that all function and relation symbols of first-order logic occur as constants in $\mathcal{I}$K, and all variables of first-order logic occur as variables in $\mathcal{I}$K. We define a mapping $\lceil - \rceil$ from first-order terms and formulas to the set of terms $\mathbb{T}$ of the illative system $\mathcal{I}$K, and a context-providing mapping $\Gamma(-)$ from sets of first-order terms and formulas to sets of terms from $\mathbb{T}$. The definition of $\lceil - \rceil$ is by induction of the structure of its argument:

- $\lceil x \rceil \equiv x$, for x a variable,
- $\lceil f(t_1, \ldots, t_n) \rceil \equiv f \lceil t_1 \rceil \ldots \lceil t_n \rceil$, for f an n-ary function symbol,
- $\lceil r(t_1, \ldots, t_n) \rceil \equiv r \lceil t_1 \rceil \ldots \lceil t_n \rceil$, for r an n-ary relation symbol,
- $\lceil \bot \rceil \equiv \bot$,
- $\lceil \varphi \vee \psi \rceil \equiv \lceil \varphi \rceil \vee \lceil \psi \rceil$,
- $\lceil \varphi \wedge \psi \rceil \equiv \lceil \varphi \rceil \wedge \lceil \psi \rceil$,
- $\lceil \varphi \rightarrow \psi \rceil \equiv \lceil \varphi \rceil \supset \lceil \psi \rceil$,
- $\lceil \forall x.\varphi \rceil \equiv \Xi \mathsf{A} \lambda x. \lceil \varphi \rceil$,
- $\lceil \exists x.\varphi \rceil \equiv \mathsf{X} \mathsf{A} \lambda x. \lceil \varphi \rceil$.

We extend the mapping $\lceil - \rceil$ to sets of first-order formulas thus: $\lceil \Delta \rceil = \{ \lceil \varphi \rceil \mid \varphi \in \Delta \}$.

For a set of first-order terms and formulas Δ, the set $\Gamma(\Delta)$ is defined to contain:

- $\mathsf{F}_n \mathsf{A} \ldots \mathsf{A} \mathsf{H} r$ for each relation symbol r of arity n, where A occurs n times,
- $\mathsf{F}_n \mathsf{A} \ldots \mathsf{A} \mathsf{A} f$ for each function symbol f of arity n, where A occurs $n+1$ times,
- $\mathsf{A} x$ for each $x \in \mathrm{FV}(\Delta)$,
- $\mathsf{A} y$ for a fresh variable y, i.e., we assume y not to occur free in any first-order formula.

The last point is necessary, because in ordinary logic the universe is implicitly assumed to be non-empty. The term F_n is defined like in Section 2.

Similar translations are provided for the other systems. Now soundness and completeness of the translations may be formulated in the following theorem, where $\models_{\mathrm{std}}$ denotes the semantic consequence relation with respect to standard models for higher-order logic, and NJp, NKp, NJ, NK, NKω, eNKω denote respective traditional systems: intuitionistic propositional logic, classical propositional logic, intuitionistic first-order logic, classical first-order logic, intensional classical higher-order logic, extensional classical higher-order logic.

Theorem 2 (Soundness and completeness of the translations)**.**

1. *The conditions* $\Delta \vdash_{\mathcal{N}} \varphi$ *and* $\Gamma(\Delta, \varphi), \lceil \Delta \rceil \vdash_{\mathcal{I}} \lceil \varphi \rceil$ *are equivalent, where*
   - $\mathcal{N} = \mathrm{NJp}$ *and* $\mathcal{I} = \mathcal{I}\mathrm{Jp}$*, or*
   - $\mathcal{N} = \mathrm{NKp}$ *and* $\mathcal{I} = \mathcal{I}\mathrm{Kp}$*, or*
   - $\mathcal{N} = \mathrm{NJ}$ *and* $\mathcal{I} = \mathcal{I}\mathrm{J}$*, or*
   - $\mathcal{N} = \mathrm{NK}$ *and* $\mathcal{I} = \mathcal{I}\mathrm{K}$*.*

   In other words, for the illative systems $\mathcal{I}\mathrm{Jp}$*,* $\mathcal{I}\mathrm{Kp}$*,* $\mathcal{I}\mathrm{J}$ *and* $\mathcal{I}\mathrm{K}$ *the translation from corresponding traditional systems is both sound and complete.*

2. *If* $\Delta \vdash_{\mathcal{N}} \varphi$ *then* $\Gamma(\Delta, \varphi), \lceil \Delta \rceil \vdash_{\mathcal{I}} \lceil \varphi \rceil$*, where* $\mathcal{N} = \mathrm{NK}\omega$ *and* $\mathcal{I} = \mathcal{I}\mathrm{K}\omega$*, or* $\mathcal{N} = e\mathrm{NK}\omega$ *and* $\mathcal{I} = e\mathcal{I}\mathrm{K}\omega$*. In other words, for the higher-order illative systems, the translation from corresponding traditional systems is sound.*

3. *If* $\Gamma(\Delta, \varphi), \lceil \Delta \rceil \vdash_{e\mathcal{I}\mathrm{K}\omega} \lceil \varphi \rceil$ *then* $\Delta \models_{\mathrm{std}} \varphi$. *In other words, if a translation of a judgement is provable in* $e\mathcal{I}\mathrm{K}\omega$ *then this judgement is valid in standard semantics.*

The above theorem actually combines and reformulates theorems 4.3.3, 4.3.5, 5.3.4, 5.3.6, 5.3.7, 6.3.6 and 6.3.7 from the thesis. The proof of these theorems uses the model constructions outlined in the next section. We do not have completeness of the translations for higher-order systems, because our model construction relies on the fact that the model of traditional higher-order logic by which it is parameterised is a standard model, and traditional higher-order logic is not complete with respect to standard semantics. However, the model construction suffices to show that if a translation of a judgement is provable in $e\mathcal{I}\mathrm{K}\omega$ then this judgement is valid in standard semantics.

3.4 Model constructions

The main results of the thesis are consistency proofs for the introduced illative systems, in particular for the strongest system $\mathcal{I}^+$. The proofs are carried out by constructing models for each of the systems. In fact, since $\mathcal{I}^+$ essentially extends the other illative systems, to establish consistency for all the systems we could just construct a model for $\mathcal{I}^+$. However, the model constructions are parameterised by models for corresponding tranditional systems of logic, and later used in completeness-of-translation proofs, and for this we need separate constructions for each system.

All constructions are based on the same general idea of defining the model by a fixpoint construction. The details of the constructions significantly increase in complexity with the increase in the strength of the systems. The most significant increase in the complexity of the model construction is with the transition from first-order to higher-order systems. We shall briefly outline the main ideas of the model construction for $\mathcal{I}\mathrm{K}\omega$, and indicate where the greatest difficulty lies.

The model construction for $\mathcal{I}\mathrm{K}\omega$ is parameterised by a standard model

$$\mathcal{N} = \langle \{\mathcal{D}_\tau \mid \tau \in \mathscr{T}\}, I \rangle$$

for higher-order logic. Here $\mathscr{T}$ is the set of types of traditional higher-order logic, defined by the grammar

$$\mathscr{T} \quad ::= \quad o \mid \iota \mid \mathscr{T} \to \mathscr{T}$$

where o is the type of propositions, and ι the type of individuals. The set $\mathcal{D}_\tau$ is a domain of objects of type $\tau \in \mathscr{T}$. If $\tau = \tau_1 \to \tau_2$ then $\mathcal{D}_\tau$ consists of all functions from $\mathcal{D}_{\tau_1}$ to $\mathcal{D}_{\tau_2}$. The mapping I provides an interpretation of constants. We assume that all constants of NKω are present in the syntax of $\mathcal{I}\mathrm{K}\omega$. For the model construction, we also assume that each element $d \in \mathcal{D}_\tau$ for any $\tau \in \mathscr{T}$ occurs as a distinct constant in the set of terms $\mathbb{T}$. If $I(c) = d \in \mathcal{D}_\tau$ then without loss of generality we assume that $c \equiv d$. If $f \in \mathcal{D}_{\tau \to \rho}$ and $a \in \mathcal{D}_\tau$, then to avoid confusion with the term fa we write $f^{\mathcal{N}}(a)$ instead of $f(a)$ to denote the value of the function f at argument a. Without loss of generality, we identify the term $\bot$ (resp. $\top$) with the element $\bot$ (resp. $\top$) of $\mathcal{D}_o$.

For $\tau \in \mathscr{T}$ and an ordinal α we define the representation relations $\succ_\tau^\alpha \in \mathbb{T} \times \mathbb{T}$, the contraction relation $\to^\alpha \in \mathbb{T} \times \mathbb{T}$, and the relation $\succ_\mathscr{T}^\alpha \in \mathbb{T} \times \mathscr{T}$ inductively. By way of an example, we shall give some of the (slightly modified) clauses of the definition. Below, the notation $X \rightsquigarrow_\tau^\alpha Y$ stands for $X \overset{*}{\to}^{\alpha} \cdot \succ_\tau^\alpha Y$, and we define $\succ_\tau^{<\alpha} = \bigcup_{\beta<\alpha} \succ_\tau^\beta$ and $\rightsquigarrow_\tau^{<\alpha} = \bigcup_{\beta<\alpha} \rightsquigarrow_\tau^\beta$.

(β) $(\lambda x.X)Y \to^\alpha X[x/Y]$,

(γ) $fX \to^\alpha b$ if $f \in \mathcal{D}_{\tau_1 \to \tau_2}$, $a \in \mathcal{D}_{\tau_1}$, $b \in \mathcal{D}_{\tau_2}$, $f^\mathcal{N}(a) = b$ and $X \succ_{\tau_1}^{<\alpha} a$,

($\mathcal{F}_\tau$) $X \succ_\tau^\alpha d$ if $\tau = \tau_1 \to \tau_2$, $d \in \mathcal{D}_{\tau_1 \to \tau_2}$ and for every $a \in \mathcal{D}_{\tau_1}$ we have $Xa \rightsquigarrow_{\tau_2}^{<\alpha} d^\mathcal{N}(a)$,

($\mathsf{V}_\top$) $X \vee Y \succ_o^\alpha \top$ if $X \succ_o^{<\alpha} \top$ or $Y \succ_o^{<\alpha} \top$,

($\mathsf{V}_\bot$) $X \vee Y \succ_o^\alpha \bot$ if $X \succ_o^{<\alpha} \bot$ and $Y \succ_o^{<\alpha} \bot$,

($\Xi_\top$) $\Xi XY \succ_o^\alpha \top$ if $X \succ_\mathscr{T}^{<\alpha} \tau$ and for every $d \in \mathcal{D}_\tau$ we have $Yd \rightsquigarrow_o^{<\alpha} \top$,

($\Xi_\bot$) $\Xi XY \succ_o^\alpha \bot$ if $X \succ_\mathscr{T}^{<\alpha} \tau$ and there exists $d \in \mathcal{D}_\tau$ with $Yd \rightsquigarrow_o^{<\alpha} \bot$,

($\mathsf{L}_\top$) $\mathsf{L}X \succ_o^\alpha \top$ if $X \succ_\mathscr{T}^{<\alpha} \tau$ for some $\tau \in \mathscr{T}$,

($\mathsf{H}_\mathscr{T}$) $\mathsf{H} \succ_\mathscr{T}^\alpha o$,

($\mathsf{A}_\mathscr{T}$) $\mathsf{A} \succ_\mathscr{T}^\alpha \iota$,

($\mathsf{F}_\mathscr{T}$) $\mathsf{F}XY \succ_\mathscr{T}^\alpha \tau_1 \to \tau_2$ if $X \succ_\mathscr{T}^{<\alpha} \tau_1$ and $Y \succ_\mathscr{T}^{<\alpha} \tau_2$.

It is to be understood that the relation $\to^\alpha$ is the compatible closure of the rules (β), (η) and (γ), while the relations $\succ_\tau^\alpha$ for $\tau \in \mathscr{T}$ and $\succ_\mathscr{T}^\alpha$ are defined directly by the corresponding rules, i.e., without taking compatible closure – these are not contraction relations.

For $\alpha \le \kappa$ we have $\to^\alpha \subseteq \to^\kappa$, $\succ_\tau^\alpha \subseteq \succ_\tau^\kappa$ for $\tau \in \mathscr{T}$, and $\succ_\mathscr{T}^\alpha \subseteq \succ_\mathscr{T}^\kappa$. Hence there is the closure ordinal ζ with $\to^\zeta = \to^{<\zeta}$, $\succ_\tau^\zeta = \succ_\tau^{<\zeta}$ for $\tau \in \mathscr{T}$, and $\succ_\mathscr{T}^\zeta = \succ_\mathscr{T}^{<\zeta}$. We use the notations $\to$, $\succ_\tau$ ($\tau \in \mathscr{T}$), $\succ_\mathscr{T}$ for $\to^\zeta$, $\succ_\tau^\zeta$ ($\tau \in \mathscr{T}$), $\succ_\mathscr{T}^\zeta$, respectively.

The idea is to define the $\mathcal{I}\mathrm{K}\omega$-model $\mathcal{M}_\mathcal{N}$ as $\mathcal{M}_\mathcal{N} = \langle \mathcal{C}, I, \mathcal{T}, \mathcal{F} \rangle$ where:

- $\mathcal{C}$ is a higher-order illative combinatory algebra constructed from the $\beta\gamma$-equivalence classes of terms, with $\mathsf{k} = [\mathsf{K}]$, $\mathsf{s} = [\mathsf{S}]$, $\mathsf{\Xi} = [\Xi]$, etc., where by $[X]$ we denote the equivalence class of X,
- the interpretation of constants I is defined by $I(c) = [c]$ for $c \in \Sigma$,
- $\mathcal{T} = \{[X] \mid X \rightsquigarrow_o \top\}$,
- $\mathcal{F} = \{[X] \mid X \rightsquigarrow_o \bot\}$.

The intuition behind $\succ_\tau$ for $\tau \in \mathscr{T}$ is that $X \succ_\tau d$ means "X is represented by d in type τ", i.e., "X behaves exactly like d in every context where a value of type τ is expected". The closure under arbitrary contexts where a value of type τ is "expected" is essentially implemented by γ-reduction. The relation $X \succ_\mathscr{T} \tau$ is interpreted as "X intepreted as a type is represented by τ".

The rules for $\succ_o$ correspond to the conditions on $\mathcal{T}$ and $\mathcal{F}$ in the definition of an $\mathcal{I}\mathrm{K}\omega$-model. They are as one would expect them to be, except perhaps the rules ($\Xi_\top$) and ($\Xi_\bot$). Instead of the rule ($\Xi_\top$) one might expect

($\Xi'_{\top}$) $\Xi XY \succ_o^{\alpha} \top$ if $\mathsf{L}X \succ_o^{<\alpha} \top$ and for all Z such that $XZ \rightsquigarrow_o^{<\alpha} \top$ we have $YZ \rightsquigarrow_o^{<\alpha} \top$.

However, in this rule there is a negative reference to $\rightsquigarrow_o^{<\alpha}$ in $XZ \rightsquigarrow_o^{<\alpha} \top$, so it may no longer be the case that $\succ_o^{\alpha} \subseteq \succ_o^{\kappa}$ for $\alpha \leq \kappa$, and we would not necessarily reach a fixpoint. The way we solve this major problem is to restrict quantification to constants from appropriate $\mathcal{D}_\tau$. We show that if $X \succ_{\mathscr{T}} \tau$ then quantifying over only elements of $\mathcal{D}_\tau$ is equivalent to quantifying over all Z such that $XZ \rightsquigarrow_o \top$. Showing this property presents a major challenge, and the proof that $\mathcal{M}_{\mathcal{N}}$ actually is an $\mathcal{I}\mathrm{K}\omega$-model becomes complicated.

The model constructions for illative systems imply the following theorem, which is a combination of corollaries 4.2.9, 4.2.14, 5.2.10, 5.2.15, 6.2.22 and 7.2.33 from the thesis.

Theorem 3 (Main result)**.** *All the systems $\mathcal{I}\mathrm{Jp}$, $\mathcal{I}\mathrm{Kp}$, $\mathcal{I}\mathrm{J}$, $\mathcal{I}\mathrm{K}$, $\mathcal{I}\mathrm{K}\omega$, $e\mathcal{I}\mathrm{K}\omega$ and $\mathcal{I}^+$ are consistent, i.e., $\bot$ is not derivable in the empty context.*

References

[1] Henk Barendregt, Martin W. Bunder, and Wil Dekkers. Systems of illative combinatory logic complete for first-order propositional and predicate calculus. *Journal of Symbolic Logic*, 58(3):769–788, 1993.

[2] Alonzo Church. A set of postulates for the foundation of logic I. *Annals of Mathematics, ser. 2*, 33(2):346–366, 1932.

[3] Alonzo Church. A set of postulates for the foundation of logic II. *Annals of Mathematics, ser. 2*, 34(4):839–864, 1933.

[4] Haskell B. Curry. Grundlagen der kombinatorischen Logik. *American Journal of Mathematics*, 52:509–536, 789–834, 1930.

[5] Haskell B. Curry, Robert Feys, and William Craig. *Combinatory Logic*, volume 1. North-Holland, 1958.

[6] Haskell B. Curry, J. Roger Hindley, and Jonathan P. Seldin. *Combinatory Logic*, volume 2. North-Holland, 1972.

[7] Łukasz Czajka. A semantic approach to illative combinatory logic. In *Computer Science Logic, 25th International Workshop / 20th Annual Conference of the EACSL, CSL 2011, September 12-15, 2011, Bergen, Norway, Proceedings*, volume 12 of *LIPIcs*, pages 174–188. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2011.

[8] Łukasz Czajka. Confluence of an extension of Combinatory Logic by Boolean constants. 2013. Submitted.

[9] Łukasz Czajka. Higher-order illative combinatory logic. *Journal of Symbolic Logic*, 73(3):837–872, 2013. See also `http://arxiv.org/abs/1202.3672` for a revised and corrected version.

[10] Łukasz Czajka. Partiality and recursion in higher-order logic. In Frank Pfenning, editor, *FoSSaCS*, volume 7794 of *Lecture Notes in Computer Science*, pages 177–192. Springer, 2013.

[11] Łukasz Czajka. Partiality and recursion in higher-order logic. Technical report, University of Warsaw, 2013. Available at `http://arxiv.org/abs/1210.2039`.

[12] Łukasz Czajka. A coinductive confluence proof for infinitary lambda-calculus. In Gilles Dowek, editor, *Rewriting and Typed Lambda Calculi - Joint International Conference, RTA-TLCA 2014, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 14-17, 2014. Proceedings*, volume 8560 of *Lecture Notes in Computer Science*, pages 164–178. Springer, 2014.

[13] Wil Dekkers, Martin W. Bunder, and Henk Barendregt. Completeness of the propositions-as-types interpretation of intuitionistic logic into illative combinatory logic. *Journal of Symbolic Logic*, 63(3):869–890, 1998.

[14] Wil Dekkers, Martin W. Bunder, and Henk Barendregt. Completeness of two systems of illative combinatory logic for first-order propositional and predicate calculus. *Archive for Mathematical Logic*, 37(5-6):327–341, 1998.

[15] Moses Schönfinkel. Über die Bausteine der mathematischen Logik. *Mathematische Annalen*, 92:305–316, 1924.