A semantic approach to illative combinatory logic*

Łukasz Czajka

University of Warsaw
Institute of Informatics
`lukaszczz@gmail.com`

Abstract

This work introduces the theory of illative combinatory algebras, which is closely related to systems of illative combinatory logic. We thus provide a semantic interpretation for a formal framework in which both logic and computation may be expressed in a unified manner. Systems of illative combinatory logic consist of combinatory logic extended with constants and rules of inference intended to capture logical notions. Our theory does not correspond strictly to any traditional system, but draws inspiration from many. It differs from them in that it couples the notion of truth with the notion of equality between terms, which enables the use of logical formulas in conditional expressions. We give a consistency proof for first-order illative combinatory algebras. A complete embedding of classical predicate logic into our theory is also provided. The translation is very direct and natural.

1998 ACM Subject Classification F.4.1 Mathematical Logic

Keywords and phrases Illative combinatory logic, term rewriting, first-order logic

Digital Object Identifier 10.4230/LIPIcs.CSL.2011.174

1 Introduction

When in the early 1930s Curry and Church invented their systems of, respectively, combinatory logic [4] and lambda calculus [3], they intended them to be foundational systems on which logic and mathematics could be based. These systems were soon shown to be inconsistent by Kleene and Rosser [9]. As a result, Church abandoned his program of basing logic on lambda-calculus. Curry, however, persisted in his aims. He and his followers tried to formulate various systems weaker than the original system of Curry in the hope of obtaining ones that would be consistent, but still strong enough to interpret traditional logic. Today, the basic part of Curry's theory is known as combinatory logic, the systems additionally incorporating logical constants as illative combinatory logic. The search for strong and consistent theories proved to be elusive. Only after more than half a century since the first publications of Church and Curry, several systems were proven complete for minimal first-order intuitionistic logic in [2], [5], [6], and for PRED$\lambda\!\to$ in [5]. See [11] for a historical overview of illative combinatory logic.

The tradition of the Curry school has been formalist, with emphasis on constructive proof-theoretic methods (cf. [11]). In this work we propose a semantic interpretation for various illative constants. In contrast to traditional systems, the meaning of these constants is given by appropriately extending the equality relation. We attempt to give a model-theoretic style semantics. As potential models we study illative combinatory algebras. These are combinatory algebras with additional elements corresponding to illative constants. One important constant which is present in our theory, but usually absent from illative systems,

* Partly supported by MNiSW grant N N206 355836.

is the conditional Cond. It acts as a connector between logic and computation, allowing to choose between two branches in a (generalized) program depending on the truth value of a quantified formula. Furthermore, formulas themselves are nothing else than generalized programs, and may contain S and K.

Our formalization is very natural and straightforward. What is non-obvious here is that it is actually correct. Modifying sligthly our axioms in seemingly harmless ways leads to inconsistent theories. In the presence of unrestricted abstraction and fixed points of arbitrary elements it is far from obvious how to formulate a consistent logical system.

Very closely related to our theory are applicative theories of Feferman (see [7]), which form the basis of his systems of explicit mathematics. These systems were intended to provide a foundation for constructive mathematics. Applicative theories are, however, usually based on partial logic. In terms of methods employed perhaps the total applicative theories with non-constructive μ-operator (see [8]) come even closer to our theory than does illative combinatory logic. Indeed, the key idea in the proofs leading to the central Corollary 42 is essentially analogous to that in the proof from the appendix of [8], where similar techniques are used in a much less general context. The author did not know about [8] until after having written down the proof in full.

Our consistency proof for first-order illative combinatory algebras is based on a non-trivial construction of a term model. We show how to extend any left-linear applicative term rewriting system satisfying some mild additional conditions into a term rewriting system whose associated quotient algebra is a first-order illative combinatory algebra. The extension is constructed by transfinitely iterating a process of expanding the term rewriting system with rules implementing quantification, until a fixpoint is reached. This bears some resemblance to transfinite truth definitions as used by Kripke (cf. [10]), which were also the inspiration for the three-valued semantics of logic programming. The details, however, are much more complicated.

The outline of the rest of this paper is as follows. Section 2 contains the definition of first-order illative combinatory algebras. In Section 3 we define a translation from first-order logic to illative language and prove its soundness. Section 4 introduces the class of functional term rewriting systems and recapitulates some known results from the theory of term rewriting. Section 5 contains the details of the term model construction. In Section 6 we use the result of Section 5 to prove completeness of the translation from Section 3.

2 Illative combinatory algebras

In this section we introduce the central concept of this work – illative combinatory algebras. Basic familiarity with ordinary combinatory logic is assumed.

▶ **Definition 1.** An *applicative algebra* $\mathcal{A}$ is a tuple $\langle \omega, \cdot, \upsilon \rangle$ where:
(1) ω is a set of *combinators*
(2) $\cdot : \omega \times \omega \to \omega$ is the *application function*
(3) $\upsilon \subseteq \omega$ is the set of *undefined* combinators
We call $\delta = \omega \setminus \upsilon$ the set of *defined* combinators. By $\omega(\mathcal{A})$, $\upsilon(\mathcal{A})$ we denote respectively the ω and υ components of $\mathcal{A}$, by $\delta(\mathcal{A})$ we denote $\omega(\mathcal{A}) \setminus \upsilon(\mathcal{A})$.

In expressions involving the application function we customarily omit parentheses and adopt the convention of association to the left, i.e. $M \cdot X \cdot Y \cdot Z$ stands for $((M \cdot X) \cdot Y) \cdot Z$. We will also sometimes omit the dots. We adopt the convention of referring to the elements of an algebra as *combinators*.

▶ **Definition 2.** An *illative combinatory algebra* (ICA) is an applicative algebra $\mathcal{A}$ with elements T, F, K, S, P, Q, Cond, A_δ which satisfy the following for any $X, Y, Z \in \omega(\mathcal{A})$:

$$
\begin{array}{ll}
(1) & T \neq F \\
(2) & T,\ F \in \delta \\
(3) & K \cdot X \cdot Y = X \\
(4) & S \cdot X \cdot Y \cdot Z = X \cdot Z \cdot (Y \cdot Z) \\
(5) & \left\{ \begin{array}{lcl} P \cdot F \cdot X & = & T \\ P \cdot X \cdot T & = & T \\ P \cdot T \cdot F & = & F \\ P \cdot X \cdot Y & \in & \upsilon \text{ otherwise} \end{array} \right. \\
(6) & \left\{ \begin{array}{lcl} \mathrm{Cond} \cdot T \cdot X \cdot Y & = & X \\ \mathrm{Cond} \cdot F \cdot X \cdot Y & = & Y \\ \mathrm{Cond} \cdot X \cdot Y \cdot Z & \in & \upsilon \text{ if } X \notin \{T, F\} \end{array} \right. \\
(7) & \left\{ \begin{array}{lcl} Q \cdot X \cdot X & \in & \{T\} \cup \upsilon \\ Q \cdot X \cdot Y & \in & \{F\} \cup \upsilon \text{ for } X \neq Y \\ Q \cdot X \cdot Y & \in & \delta \text{ if } X, Y \in \delta \end{array} \right. \\
(8) & \left\{ \begin{array}{lcl} A_\delta \cdot X & = & T \text{ if } X \in \delta \\ A_\delta \cdot X & \in & \{F\} \cup \upsilon \text{ if } X \in \upsilon \end{array} \right.
\end{array}
$$

We will sometimes write A for A_δ.

▶ Remark. Intuitively, in an illative combinatory algebra undefined combinators are interpreted as meaningless at the object level, but not necessarily completely meaningless. Indeed, there may be undefined combinators which applied to a defined combinator give a defined result. The set δ is intuitively interpreted as the universe of discourse. It is intended to encompass everything we may meaningfully talk about at the object level. In particular, it includes the truth values T and F. The combinator A stands for a partial predicate which is true for elements of δ, and false or undefined for elements of υ. This predicate cannot be defined from the other combinators. The combinator Q is intended to represent a partial equality predicate.

▶ Remark. Any illative combinatory algebra satisfies the principle of combinatory abstraction and has a fixed point combinator. Thus, for every equation of the form

$$M \cdot X_1 \cdot \ldots \cdot X_n = \Phi(M, X_1, \ldots, X_n)$$

where $\Phi(Y, X_1, \ldots, X_n)$ is a combination of Y, X_1, ..., X_n and some of the combinators postulated in the definition of an ICA, there exists a combinator M such that the equation holds for any combinators X_1, ..., X_n. We will often rely on this fact and define combinators by such equations. Sometimes we will also use the lambda-notation $\lambda x\, \Phi(x)$ to denote a combinator M such that $MX = \Phi(X)$ for all $X \in \omega$. If there can be more than one combinator satisfying a given equation, then it is tacitly understood that we choose one such specific combinator and it does not matter which one.

▶ Remark. Our aim is to make as many combinators belong to δ as possible, since these are the combinators on which our additional elements are guaranteed to "work". However, one cannot get rid of υ altogether because the existence of fixed points of arbitrary combinators would lead to a contradiction. In fact, it can be easily shown that if $M \cdot X \in \delta$ for all $X \in \omega$ then $M \cdot X = M \cdot Y$ for all $X, Y \in \omega$.

For brevity, we will mostly omit explicit references to illative combinatory algebras. The following facts and definitions are to be understood that they are relative to some fixed illative combinatory algebra.

We use the notation N for $\lambda x.PxF$, $\wedge$ for $\lambda x.\lambda y.N(Px(Ny))$, and $\vee$ for $\lambda x.\lambda y.P(Nx)y$. We occasionally adopt infix notation for $\wedge$ and $\vee$.

It is easy to see that P, N, $\wedge$ and $\vee$ satisfy the following equations for any $X, Y \in \omega$:

$$\begin{array}{rll} PXY = T & \text{iff} & X = F \text{ or } Y = T \\ PXY = F & \text{iff} & X = T \text{ and } Y = F \\ NX = T & \text{iff} & X = F \\ NX = F & \text{iff} & X = T \\ \wedge XY = T & \text{iff} & X = T \text{ and } Y = T \\ \wedge XY = F & \text{iff} & X = F \text{ or } Y = F \\ \vee XY = T & \text{iff} & X = T \text{ or } Y = T \\ \vee XY = F & \text{iff} & X = F \text{ and } Y = F \end{array}$$

▶ **Definition 3.** A set of combinators $\tau \subseteq \omega$ is a *type* represented by $M \in \omega$ if the following conditions hold:
(1) $M \cdot X = T$ for $X \in \tau$
(2) $M \cdot X \in \{F\} \cup \upsilon$ for $X \in \omega \setminus \tau$

Note that ω and δ are types represented by $K \cdot T$ and A respectively. We use the notation b for the type represented by $A_b = \lambda x.(QxT) \vee (QxF)$.

▶ **Definition 4.** Let $\sigma, \rho \subseteq \omega$. A *function space* $\sigma \Rightarrow \rho$ from σ to ρ is the set of all combinators M such that $M \cdot X \in \rho$ for $X \in \sigma$.

We use small Greek letters τ, σ, ρ, ω, etc. both to denote subsets of ω and as parts of symbols denoting constants or combinators, e.g. in A_τ. In the second case the subscript does not have a meaning of its own, but only highlights a connection of the symbol with some set τ, which may even be defined only after introducing the symbol itself. Analogously, we use subscripts of the form $\sigma \to \rho$ when we intend to highlight a connection to the function space $\sigma \Rightarrow \rho$. In compound expressions $\to$ and $\Rightarrow$ are assumed to be right-associative. We adopt the notation $\sigma^n \Rightarrow \rho$ for $\sigma \Rightarrow \ldots \Rightarrow \sigma \Rightarrow \rho$ where σ occurs n times. Analogously, we use $\sigma^n \to \rho$ in subscripts.

▶ **Definition 5.** A combinator M is τ-total, for $\tau \subseteq \omega$, if $MX \in \delta$ for all $X \in \tau$.

▶ **Definition 6.** Let $\tau \subseteq \omega$. A *τ-quantifier* is any combinator Π_τ such that:

$$\begin{array}{lcl} \Pi_\tau \cdot X & = & T \text{ if for all } Y \in \tau \text{ we have } X \cdot Y = T \\ \Pi_\tau \cdot X & = & F \text{ if there exists } Y \in \tau \text{ such that } X \cdot Y = F \\ \Pi_\tau \cdot X & \in & \upsilon \text{ otherwise} \end{array}$$

We use the notation Σ_τ for $\lambda x.N(\Pi_\tau(S(KN)x))$. A combinator Π_δ satisfying the above equations for $\tau = \delta$ is a *first-order quantifier*. We will sometimes write Π instead of Π_δ.

It is straightforward to verify that Π_τ and Σ_τ satisfy the following for any $X \in \omega$:

$$\begin{array}{lll} \Pi_\tau X = T & \text{iff} & XY = T \text{ for all } Y \in \tau \\ \Pi_\tau X = F & \text{iff} & XY = F \text{ for some } Y \in \tau \\ \Sigma_\tau X = T & \text{iff} & XY = T \text{ for some } Y \in \tau \\ \Sigma_\tau X = F & \text{iff} & XY = F \text{ for all } Y \in \tau \end{array}$$

It is easy to see that if A_τ is a δ-total combinator representing a type $\tau \subseteq \delta$, then the combinator $\lambda x.\Pi_\delta \lambda y.P\,(A_\tau y)\,(xy)$ is a τ-quantifier. Moreover, if Π_{τ_1} is a τ_1-quantifier and A_{τ_2} represents a type τ_2, then $\tau_1 \Rightarrow \tau_2$ is a type represented by $A_{\tau_1 \to \tau_2} = \lambda x.\Pi_{\tau_1}\lambda y.A_{\tau_2}(xy)$.

▶ **Definition 7.** A *first-order illative combinatory algebra* (FO-ICA) is an illative combinatory algebra with signature extended with Π_δ, and with the laws from Definition 6 for Π_δ added as axioms.

▶ Remark. One may wonder why we postulate the existence of Π_δ instead of Π_ω, whose range of quantification is broader. After all, we could use $\Pi'_\delta = \lambda x.\Pi_\omega \lambda y.P(Ay)(xy)$. However, Π'_δ is not a δ-quantifier. The reason is the existence of undefined combinators and the fact that they are included in the range of quantification of Π_ω. For instance, suppose M is such that $M \cdot X = T$ iff $X \in \delta$. One may easily show that there is $Y \in \upsilon$ such that $A \cdot Y \in \upsilon$. Hence, $P(AY)(MY) \in \upsilon$. Moreover, by definitions of A and M there is no Z such that $P(AZ)(MZ) = F$. So the last equation in the definition of Π_ω applies, and we have $\Pi'_\delta M \in \upsilon$.

More generally, if A_τ represents a type $\tau \neq \omega$, then by an analogous argument we could prove that $\Pi_\omega \lambda x.P(A_\tau x)(Mx) \in \upsilon$ for any $M \in \omega$ such that $M \cdot X = T$ iff $X \in \tau$. This shows that Π_ω is not particularly interesting, because its range cannot be restricted in a meaningful way.

▶ Remark. Logic based on the theory of first-order illative combinatory algebras is, in a practical sense, more expressive than traditional predicate logic. For example, denote by $\underline{n}$ the Church numeral representing $n \in \mathbb{N}$. Now we can write a recursive definition of U as follows:

$$U\underline{n} \quad = \quad \mathrm{Cond}\,(Q\,\underline{n}\,0)\,(SKK)\ (\lambda f.\Pi\lambda x.\Sigma\lambda y.U(\mathrm{Pred}\,\underline{n})(fxy))$$

where Pred is the predecessor combinator for Church numerals. By simple induction one can show:

$$U\underline{n} \quad = \quad \lambda f.\Pi\lambda x_1.\Sigma\lambda y_1.\ldots\Pi\lambda x_n.\Sigma\lambda y_n.f x_1 y_1 x_2 y_2 \ldots x_n y_n$$

Now assume that we have a δ-total combinator which represents the type $\mathcal{N}$ consisting of Church numerals, and that all Church numerals are in δ. Theorem 43 implies that the definition of a FO-ICA may be modified to satisfy these assumptions without sacrificing any of the results in this paper. Then there exists an $\mathcal{N}$-quantifier $\Pi_\mathcal{N}$. Now, given a combinator M, the expression

$$\Sigma_\mathcal{N}\lambda x.UxM$$

is true iff there exists an alternation of $2n$ quantifiers such that

$$\Pi\lambda x_1.\Sigma\lambda y_1.\ldots\Pi\lambda x_n.\Sigma\lambda y_n.M x_1 y_1 \ldots x_n y_n$$

is true. To be precise, $\Sigma_\mathcal{N}\lambda x.UxM$ will most often be in υ if such an alternation does not exist.

The power comes from the fact that quantifiers may be freely combined with S and K. This allows for recursive definitions involving logical operators.

Another important feature of our theory is the presence of the combinator Cond and the fact that the truth notion at the meta-level is coupled with the notion of equality between terms. In other words, being true is equivalent to evaluating to a concrete value T, which may be used in the "program" itself. This is significantly different from simply stating that some terms are “true” or “derivable” by means of some meta-level definition, but without providing any possibility of using this information *inside* the system.

For instance, with our approach one can write recursive definitions of the form:

$$M \quad = \quad \lambda x.\Psi\,(\mathrm{Cond}\,(\Pi\Phi_1(x, M))\,\Phi_2(x, M)\,\Phi_3(x, M))$$

and they behave as expected – if $\Pi\Phi_1(X, M)$ is true then the first branch constitutes the value of MX, if false then the second. What is more, it may happen that we know that $\Pi\Phi_1(X, M)$ is true regardless of what X is, and we may conclude that $M = \lambda x.\Psi\,(\Phi_2(x, M))$. The combinator Cond acts as a connector between logic and computation.

3 Translation from first-order to illative theories

In this section we define a natural translation from the language of first-order logic to illative language and prove its soundness with respect to FO-ICAs. We defer the proof of completeness to Section 6. Much of the present section contains some fairly obvious but necessary definitions.

We will be dealing mostly with *applicative* terms, i.e. terms from languages over signatures consisting solely of a single binary function symbol $\cdot$ and constants including all the constants postulated in the definition of a FO-ICA. We denote such a language by $\mathcal{L}(\Sigma, V)$, where Σ is a set of constants, and V is a set of variables. All terms are assumed to be applicative, unless qualified with the phrase *first-order*. We use the symbols t, s, etc. for terms, x, y, etc. for variables, and M, X, etc. for combinators (elements of an algebra), except that we use the same symbols for primitive constants and corresponding combinators defined in Section 2. The intended meaning of a symbol will always be clear from the context.

We use the notation $[\![t]\!]^u_{\mathcal{A}}$ for the value of t under variable valuation u in the structure $\mathcal{A}$. We omit the decorations when obvious from the context or irrelevant. We also adopt the notation $t_1[x/t_2]$ for the term t_1 with all free occurences of x substituted for t_2. Analogously, we use $u[x/M]$ for the valuation u' such that $u'(y) = u(y)$ for $y \neq x$ and $u'(x) = M$.

We define lambda-abstraction at the syntactic level by the standard abstraction algorithm: $\lambda^* x.x = SKK$, $\lambda^* x.t = Kt$ if $x \notin FV(t)$, and $\lambda^* x.t_1 t_2 = S(\lambda^* x.t_1)(\lambda^* x.t_2)$. In what follows the symbols A_b, $\wedge$, etc. will sometimes stand for terms defined completely analogously to the corresponding combinators in Section 2, but at the syntactic level using the abstraction algorithm. We still use these symbols to denote the combinators as well. Again, the intended meaning will always be clear from the context.

Let $\mathcal{A}$ be a FO-ICA, and u a valuation. It is easy to verify that for any terms t_1, t_2 we have $[\![(\lambda^* x.t_1)t_2]\!]^u_{\mathcal{A}} = [\![t_1[x/t_2]]\!]^u_{\mathcal{A}}$. Also for any term t and any $M \in \omega(\mathcal{A})$ we have the identity $[\![\lambda^* x.t_1]\!]^u_{\mathcal{A}} \cdot M = [\![t_1]\!]^{u'}_{\mathcal{A}}$ where $u' = u[x/M]$.

We now redefine some standard notions from elementary first-order logic. Subseqently, we will refer to the original notions by qualifying them with the phrase *first-order*. The redefined notions will be qualified with *illative*, but the qualification will often be dropped. By an *illative theory* we mean a set of applicative terms. We say that a FO-ICA $\mathcal{A}$ *satisfies* a term t under variable valuation u, denoted by $\mathcal{A} \models^u t$, if $[\![t]\!]^u_{\mathcal{A}} = T$. We define the notions of *illative semantic consequence* ($\Gamma \models t$) and *illative model* ($\mathcal{A} \models \Gamma$) completely analogously to standard definitions in first-order logic, but with arbitrary terms in place of formulas and requiring all structures to be FO-ICAs.

We use the symbol Δ for a first-order theory, ϕ, ψ for first-order formulas, $\models_{FO}$ for the first-order semantic consequence relation.

By a *first-order expression* we mean a first-order formula or a first-order term. We extend the notion of first-order valuation to formulas. If $\mathcal{A} \models^u_{FO} \phi$ then $[\![\phi]\!]^u_{\mathcal{A}} = T$, otherwise $[\![\phi]\!]^u_{\mathcal{A}} = F$.

We assume that in a first-order language the only logical connective is $\to$, the only quantifier $\forall$, and there is a constant $\bot$ for false. We also assume that we have a new constant A_ς in the illative signature, and the signature contains as constants all symbols (of any arity) from the corresponding first-order language.

We write A_ι for the term $\lambda^* x.(A_\varsigma x) \wedge (A_\delta x)$ and Π_ι for $\lambda^* y.\Pi_\delta \lambda^* x.P(A_\iota x)(yx)$. We define $A_{\iota^{n+1}\to\iota}$ inductively as $\lambda^* x.\Pi_\iota \lambda^* y.A_{\iota^n\to\iota}(xy)$, where $A_{\iota^0\to\iota} = A_\iota$. Analogously, we define $A_{\iota^{n+1}\to b}$ as $\lambda^* x.\Pi_\iota \lambda^* y.A_{\iota^n \to b}(xy)$.

▶ **Definition 8.** The illative theory Γ_0 constains the terms $\Pi_\delta\,(S\,(KA_b)\,A_\iota)$, $\Sigma_\delta A_\iota$, $A_{\iota^n\to\iota}\,f$ for all function symbols f of arity $n \geq 0$, and $A_{\iota^n\to b}\,r$ for all relation symbols r of arity $n > 0$.

▶ **Definition 9.** We define inductively a translation Ψ as follows.

For first-order terms:
- $\Psi(x) = x$ for a variable x
- $\Psi(f(t_1,\ldots,t_n)) = f \cdot \Psi(t_1) \cdot \ldots \cdot \Psi(t_n)$ for a function symbol f of arity $n \geq 0$

For first-order formulas:
- $\Psi(\bot) = F$
- $\Psi(r(t_1,\ldots,t_n)) = r \cdot \Psi(t_1) \cdot \ldots \cdot \Psi(t_n)$ for a relation symbol r of arity n
- $\Psi(t_1 = t_2) = Q \cdot \Psi(t_1) \cdot \Psi(t_2)$
- $\Psi(\phi_1 \to \phi_2) = P \cdot \Psi(\phi_1) \cdot \Psi(\phi_2)$
- $\Psi(\forall_x \phi) = \Pi_\iota \lambda^* x.\Psi(\phi)$

For a first-order theory Δ we define $\Psi(\Delta)$ to be the sum of Γ_0 and the image $\mathrm{Im}_\Psi(\Delta)$ of Ψ on Δ.

The general idea of the soundness proof is to construct for every illative model $\mathcal{B}$ of $\Psi(\Delta)$ a first-order structure $\mathcal{A}$ which satisfies exactly those sentences whose translations are true in $\mathcal{B}$. Such a structure will obviously be a model of Δ. Hence, any semantic consequence ϕ of Δ will be satisfied by $\mathcal{A}$, so $\Psi(\phi)$ will be true in $\mathcal{B}$.

▶ **Definition 10.** Let $\mathcal{B}$ be an illative model of Γ_0. We define $\iota_{\mathcal{B}}$ to be the set of all combinators $M \in \omega(\mathcal{B})$ such that $[\![A_\iota]\!]_{\mathcal{B}} \cdot M = T$. The subscript will be dropped when obvious from the context.

It is easy to see that if $\mathcal{B}$ is an illative model of Γ_0, then $[\![A_\iota]\!]_{\mathcal{B}}$ is a δ-total combinator representing the non-empty type $\iota_{\mathcal{B}} \subseteq \delta(\mathcal{B})$. It is also true that in every illative model $\mathcal{B}$ of Γ_0 the combinator $[\![\Pi_\iota]\!]_{\mathcal{B}}$ is a $\iota_{\mathcal{B}}$-quantifier.

▶ **Definition 11.** A first-order structure $\mathcal{A}$ and a FO-ICA $\mathcal{B}$ are *correspondent* if the universe U of $\mathcal{A}$ is a subset of $\omega(\mathcal{B})$ and the following conditions hold:
- every function symbol f of arity n is interpreted in $\mathcal{A}$ by the function

$$\{(X_1,\ldots,X_n,Y) \in U^{n+1} \mid [\![f]\!]_{\mathcal{B}} \cdot X_1 \cdot \ldots \cdot X_n = Y\}$$

- every relation symbol r of arity n is interpreted in $\mathcal{A}$ by the relation

$$\{(X_1,\ldots,X_n) \in U^n \mid [\![r]\!]_{\mathcal{B}} \cdot X_1 \cdot \ldots \cdot X_n = T\}$$

▶ **Lemma 12.** *Assume $\mathcal{B}$ is an illative model of Γ_0 and $\mathcal{A}$ is a first-order structure with $\iota_{\mathcal{B}}$ as the universe. If $\mathcal{A}$ and $\mathcal{B}$ are correspondent, then $[\![e]\!]^u_{\mathcal{A}} = [\![\Psi(e)]\!]^u_{\mathcal{B}}$ for any first-order expression e and any valuation u such that $Rg(u) \subseteq \iota_{\mathcal{B}}$.*

Proof. Simple induction on the complexity of e.

For instance, assume $e = \forall_x \psi$ and $[\![\forall_x \psi]\!]^u_{\mathcal{A}} = T$. Then for every $X \in \iota$ we have $T = [\![\psi]\!]^{u'}_{\mathcal{A}} = [\![\Psi(\psi)]\!]^{u'}_{\mathcal{B}}$ by inductive hypothesis, where $u' = u[x/X]$. So for every $X \in \iota$ we have $[\![\lambda^* x.\Psi(\psi)]\!]^u_{\mathcal{B}} \cdot X = [\![\Psi(\psi)]\!]^{u'}_{\mathcal{B}} = T$. Hence, by the fact that $[\![\Pi_\iota]\!]_{\mathcal{B}}$ is a ι-quantifier in $\mathcal{B}$ we have $[\![\Pi_\iota \lambda^* x.\Psi(\psi)]\!]^u_{\mathcal{B}} = T$.

The other cases are similar. We need the assumption of correspondence in the cases $e = f(t_1, \ldots, t_n)$ and $e = r(t_1, \ldots, t_n)$. In the second of these $A_{\iota^n \to b}\, r \in \Gamma_0$ is also needed. ◀

▶ **Theorem 13. Soundness**
Let ϕ and all formulas in Δ be closed. If $\Delta \models_{FO} \phi$ then $\Psi(\Delta) \models \Psi(\phi)$.

Proof. Suppose $\Delta \models_{FO} \phi$. Because all terms in $\Psi(\Delta)$ as well as $\Psi(\phi)$ are closed by our construction, it suffices to show that every illative model of $\Psi(\Delta)$ is an illative model of $\Psi(\phi)$.

So assume $\mathcal{B}$ is an illative model of $\Psi(\Delta)$. Since $\Gamma_0 \subseteq \Psi(\Delta)$ then $\mathcal{B}$ is an illative model of Γ_0. Hence, $\iota_{\mathcal{B}} \subseteq \delta(\mathcal{B})$ is a non-empty type represented by $[\![A_\iota]\!]_{\mathcal{B}}$.

Let $\mathcal{A}$ be a first-order structure with universe ι and functions and relations as in Definition 11. Note that it is not immediately obvious that this is well-defined, because the interpretation of a function symbol f of arity n must be a total function from ι^n to ι. However, this is satisfied because $A_{\iota^n \to \iota}\, f \in \Psi(\Delta)$. Note also that the non-emptiness of ι is necessary because the universe of a first-order structure is always assumed to be non-empty.

Therefore, by Lemma 12 we may conclude that $[\![\psi]\!]_{\mathcal{A}} = [\![\Psi(\psi)]\!]_{\mathcal{B}}$ for every closed first-order formula ψ. We have $\mathcal{A} \models_{FO} \Delta$, because $[\![\psi]\!]_{\mathcal{A}} = [\![\Psi(\psi)]\!]_{\mathcal{B}} = T$ for $\psi \in \Delta$. From the initial assumption $\Delta \models_{FO} \phi$ we may now conclude that $\mathcal{A} \models_{FO} \phi$. This implies $[\![\Psi(\phi)]\!]_{\mathcal{B}} = [\![\phi]\!]_{\mathcal{A}} = T$. Therefore, $\mathcal{B} \models \Psi(\phi)$. ◀

4 Functional term rewriting systems

This section defines the class of functional term rewriting systems and briefly recapitulates some known results from the term rewriting theory for the sake of completeness. Term rewriting notation and terminology conforms to that from [1].

▶ **Definition 14.** The set of *positions* of a term $t \in \mathcal{L}(\Sigma, V)$ is a set $\mathrm{Pos}(t)$ of strings over the alphabet $\{0, 1\}$ defined inductively as follows: $\mathrm{Pos}(t_0 \cdot t_1) = \{\varepsilon\} \cup \{0p \mid p \in \mathrm{Pos}(t_0)\} \cup \{1p \mid p \in \mathrm{Pos}(t_1)\}$, and $\mathrm{Pos}(x) = \varepsilon$, where ε is the empty string and $x \in V$. The *leftmost position* of t is the position 0^i, where 0^n for $n \in \mathbb{N}$ means 0 repeated n times, such that no position of t is of the form 0^j for $j > i$. For $p \in \mathrm{Pos}(t)$, the *subterm of s at position p*, denoted by $t_{|p}$, is defined by induction on the length of p: $t_{|\varepsilon} = t$, $(t_0 \cdot t_1)_{|bq} = t_{b|q}$. A *context* C is a term over $\mathcal{L}(\Sigma \cup \{\Box\}, V)$ with exactly one occurence of $\Box$. By $C[t]$ for $t \in \mathcal{L}(\Sigma, V)$ we denote the term C with $\Box$ replaced by t.

▶ **Definition 15.** A *rewrite rule*, or simply *rule*, over $\mathcal{L}(\Sigma, V)$ is a pair $(l, r) \in \mathcal{L}(\Sigma, V) \times \mathcal{L}(\Sigma, V)$ such that l is not a variable and $\mathrm{Var}(r) \subseteq \mathrm{Var}(l)$. Rewrite rules will be written as $l \to r$. The term l is called the *left side* of the rule, r the *right side*. A rule $l \to r$ is *left-linear* if no variable occurs twice in l. A rule $l \to r$ is *trivial* if $l = r$. A *term rewriting system* is a set of rewrite rules. A term rewriting system is *left-linear* if each of its rules is left-linear.

Let R be a term rewriting system. The *reduction relation* $\to_R \subseteq \mathcal{L}(\Sigma, V) \times \mathcal{L}(\Sigma, V)$ is defined as follows:

$t \to_R s$ iff there exist $l \to r \in R$, a context C and a substitution σ such that $t = C[\sigma l]$ and $s = C[\sigma r]$.

We sometimes write $t \to_R^p s$ to indicate at which position the reduction takes place.

We say that a rule $l \to r \in R$ *applies* to t if there exist $p \in \mathrm{Pos}(t)$ and a substitution σ such that $t_{|p} = \sigma l$. We say that a term is in *R-normal form* if no rule from R applies to it.

▶ **Notation 16.** Let $\to$ be a binary relation on terms. We denote by $\to^{\equiv}$ the reflexive closure of $\to$, by $\xrightarrow{*}$ the reflexive transitive closure, and by $\overset{*}{\leftrightarrow}$ the reflexive transitive symmetric closure. We write $t \to s$ to indicate that $(t, s) \in \to$. Analogously for $\to^{\equiv}$, $\xrightarrow{*}$ and $\overset{*}{\leftrightarrow}$.

▶ **Definition 17.** A position p of a term $t \in \mathcal{L}(\Sigma, V)$ is a *function position* if either $p = q0$ or the size of t is 1 and $p = \varepsilon$. A term $t \in \mathcal{L}(\Sigma, V)$ is *functional* if it does not have any variables at function positions. We use the notation $\Sigma_f(t)$ for the set of constants at function positions in a term t. By $\mathcal{H}(t)$ we denote the constant at the leftmost position in a functional term t. A rule $l \to r$ is *functional* if l is a functional term. A *functional term rewriting system* (FTRS) is a term rewriting system over $\mathcal{L}(\Sigma, V)$, such that all rules are functional. We use the notation $\Sigma_f(R)$ for $\bigcup_{l \to r \in R} \Sigma_f(l)$, and $\mathcal{H}(R)$ for $\{\mathcal{H}(l) \mid l \to r \in R\}$.

▶ **Fact 18.** *If t is a functional term and s is such that $\Sigma_f(t) \nsubseteq \Sigma_f(s)$, then there is no substitution σ and position p such that $s_{|p} = \sigma t$. Moreover, if s is a functional term and $\mathcal{H}(t) \notin \Sigma_f(s)$, then t does not unify with a non-variable subterm of s.*

▶ **Definition 19.** A functional term rewriting system R *generates* an applicative algebra $\mathcal{A}_R = \langle \omega, \cdot, \upsilon \rangle$ where $\omega = \{[t]_R\}$ is the set of equivalence classes of $\overset{*}{\leftrightarrow}_R$ on closed terms, υ is the set of those $[t]_R$ for which there is no t' in R-normal form such that $t \overset{*}{\leftrightarrow}_R t'$, and $\cdot$ is defined by $[t_1]_R \cdot [t_2]_R = [t_1 \cdot t_2]_R$.

▶ **Definition 20.** Let $l_1 \to r_1 \in R_1, l_2 \to r_2 \in R_2$, let p be a position such that $l_{1|p}$ is not a variable. We assume the rules do not share variable names. Let σ be the most general unifier of $l_{1|p}$ and l_2. Then $\langle \sigma r_1, (\sigma l_1)[\sigma r_2]_p \rangle$ is a *critical pair* between R_1 and R_2. A critical pair is a *root critical pair* if $p = \varepsilon$. The set of all critical pairs between R_1 and R_2 is denoted by $\mathrm{Crit}(R_1, R_2)$, the set of all root critical pairs by $\mathrm{Crit}_r(R_1, R_2)$, and $\mathrm{Crit}_i(R_1, R_2)$ is the set of all non-root critical pairs between R_1 and R_2. A critical pair $\langle u_1, u_2 \rangle \in \mathrm{Crit}(R_1, R_2)$ may be *closed* if $u_1 \to_{R_2} u_2$ or $u_2 \to_{R_1} u_1$.

▶ **Definition 21.** R_1 is *compatible* with R_2 if

- for all $\langle u_1, u_2 \rangle \in \mathrm{Crit}(R_1, R_2)$ there is u such that $u_1 \xrightarrow{*}_{R_2} u$ and $u_2 \to_{R_1}^{\equiv} u$,
- for all $\langle u_2, u_1 \rangle \in \mathrm{Crit}_i(R_2, R_1)$ we have $u_1 \to_{R_2}^{\equiv} u_2$.

Two term rewriting systems R_1, R_2 are compatible if R_1 is compatible with R_2 or *vice versa*.

▶ **Fact 22.** *If R_1, R_2 are FTRSes such that $\Sigma_f(R_1) \cap \Sigma_f(R_2) = \emptyset$, then they are compatible.*

▶ **Definition 23.** We say that two relations $\to_1$ and $\to_2$ *commute* whenever $t \xrightarrow{*}_1 t_1$ and $t \xrightarrow{*}_2 t_2$ implies the existence of s such that $t_1 \xrightarrow{*}_2 s$ and $t_2 \xrightarrow{*}_1 s$. Two term rewriting systems R_1 and R_2 commute if $\to_{R_1}$ and $\to_{R_2}$ commute.

▶ **Lemma 24. Commutative Union Lemma**
If R_1 and R_2 are confluent and they commute, then $R_1 \cup R_2$ is confluent.

The following theorem is a special case of the result from [12]. We do not state it in its full generality mostly due to lack of space to introduce the necessary concepts.

▶ **Theorem 25.** *Compatible left-linear term rewriting systems commute.*

It follows from Theorem 25 and the Commutative Union Lemma that if R is left-linear and compatible with itself, then it is confluent.

5 Extensions of standard systems

This section contains the mathematically non-trivial part of this work. We show how to extend any FTRS satisfying some mild additional conditions into an FTRS that generates a FO-ICA.

▶ **Definition 26.** A functional term rewriting system R is *standard* if it is left-linear, confluent, and $\Sigma_f(R) \cap \{\Pi, Q, A\} = \emptyset$.

▶ **Definition 27.** The term rewriting system PROP is defined by the following rules:

$$\begin{aligned} K \cdot x \cdot y &\to x & (1)\\ S \cdot x \cdot y \cdot z &\to x \cdot z \cdot (y \cdot z) & (2)\\ P \cdot F \cdot x &\to T & (3)\\ P \cdot x \cdot T &\to T & (4)\\ P \cdot T \cdot F &\to F & (5)\\ P \cdot x \cdot y &\to P \cdot x \cdot y & (6)\\ \mathrm{Cond} \cdot T \cdot x \cdot y &\to x & (7)\\ \mathrm{Cond} \cdot F \cdot x \cdot y &\to y & (8)\\ \mathrm{Cond} \cdot x \cdot y \cdot z &\to \mathrm{Cond} \cdot x \cdot y \cdot z & (9) \end{aligned}$$

▶ **Lemma 28.** *The term rewriting system PROP is standard.*

Proof. There are only the following root critical pairs, all of which satisfy the requirements of compatibility.

- The pair $\langle T, P \cdot F \cdot x\rangle$ between rules (3) and (6). We have $P \cdot F \cdot x \to T$ by rule (3).
- The root critical pairs between rules (6) and (3), (4) and (6), (6) and (4), (5) and (6), (6) and (5), (7) and (9), (9) and (7), (8) and (9), (9) and (8) are dealt with completely analogously to the above one.
- The trivial critical pair $\langle T, T\rangle$ between rules (3) and (4) or (4) and (3).

◀

Note that it follows directly from Lemma 28, Theorem 25 and the Commutative Union Lemma that $R \cup \mathrm{PROP}$ is standard.

▶ **Definition 29.** A term t is *R-standard* if it is a closed term in $R \cup \mathrm{PROP}$-normal form such that $\Sigma_f(t) \subseteq \Sigma_f(R \cup \mathrm{PROP})$.

For the rest of this section we assume a fixed standard functional term rewriting system R compatible with PROP, and a fixed family $\mathcal{T}_{\mathbb{I}} = \{\mathcal{T}_i \mid i \in \mathbb{I}\}$ of sets of R-standard terms, where $\mathbb{I}$ is some arbitrary index set.

▶ **Definition 30.** The term rewriting system R_I is defined by the following rules:

$$\begin{aligned} \Pi \cdot x &\to \Pi \cdot x\\ Q \cdot x \cdot y &\to Q \cdot x \cdot y\\ A \cdot x &\to A \cdot x\\ A_{\mathcal{T}_i} \cdot t_i &\to T\\ A_{\mathcal{T}_i} \cdot x &\to A_{\mathcal{T}_i} \cdot x \end{aligned}$$

for all $i \in \mathbb{I}$ and all terms $t_i \in \mathcal{T}_i$, where $A_{\mathcal{T}_i}$ are new symbols not present in $\Sigma_f(R) \cup \Sigma_f(\mathrm{PROP}) \cup \{\Pi, Q, A\}$.

It is easy to see that every R-standard term is in R_I-normal form.

▶ **Definition 31.** The term rewriting system R_{II} is defined by the following rules:

$$\begin{array}{rcl} Q \cdot t \cdot t & \to & T \\ Q \cdot t_1 \cdot t_2 & \to & F \\ A \cdot t & \to & T \\ A_{\mathcal{T}_i} \cdot t_i & \to & F \end{array}$$

for all $i \in \mathbb{I}$ and all closed terms t, t_1, t_2, t_i in $R \cup \mathrm{PROP} \cup R_I$-normal form, such that $t_1 \neq t_2$ and $t_i \notin \mathcal{T}_i$.

Below we use the notation R_0 for $R \cup \mathrm{PROP} \cup R_I \cup R_{II}$.

▶ **Lemma 32.** *The term rewriting system R_0 is left-linear and confluent.*

Proof. It is evident that R_0 is left-linear. Notice that $R \cup \mathrm{PROP} \cup R_I$ is confluent because $R \cup \mathrm{PROP}$ and R_I are compatible by Fact 22 and the fact that $R \cup \mathrm{PROP}$ is standard.

We now prove that R_{II} is confluent. It is evident from Definition 31 that there are no root critical pairs. For two rules to form a non-root critical pair the left side of one of the rules has to unify with a proper subterm of the left side of the other rule. Because all left sides are closed terms, this is equivalent to the situation when the left side of one rule is equal to a proper subterm of another. It follows directly from definitions that all proper subterms of left sides of rules are in $R \cup \mathrm{PROP} \cup R_I$-normal form. However, no left side of a rule is in $R \cup \mathrm{PROP} \cup R_I$-normal form because the corresponding trivial rule from R_I applies. This implies that there are no critical pairs.

We show that $R \cup \mathrm{PROP} \cup R_I$ is compatible with R_{II}. Because $R \cup \mathrm{PROP}$ is standard, then by Fact 22 we need to consider only critical pairs between R_I and R_{II}. It is evident that all such pairs are root critical pairs between a trivial rule from R_I and a rule from R_{II}. They may be closed by simply applying the rule from R_{II}. ◀

▶ **Definition 33.** For an ordinal $\alpha > 0$, define R_α as the sum of $\bigcup_{\beta<\alpha} R_\beta$ and the rules:
- $\Pi \cdot t \to T$ for all closed terms t such that for any closed term s in R_0-normal form there is an ordinal $\beta < \alpha$ for which $t \cdot s \xrightarrow{*}_{R_\beta} T$.
- $\Pi \cdot t \to F$ for all closed terms t such that there is a closed term s in R_0-normal form and an ordinal $\beta < \alpha$ for which $t \cdot s \xrightarrow{*}_{R_\beta} F$.

A simple cardinality argument shows that there exists the least ordinal ζ such that $R_\zeta = \bigcup_{\alpha<\zeta} R_\alpha$. We sometimes write $R_\zeta^{\mathcal{T}_\mathbb{I}}$ for R_ζ when $\mathcal{T}_\mathbb{I}$ is not obvious from the context.

▶ **Lemma 34.** *For $\alpha \geq 0$, a term t is in $R \cup PROP \cup R_I$-normal form iff it is in R_α-normal form.*

Proof. If a rule from $R_\alpha \setminus (R \cup \mathrm{PROP} \cup R_I)$, e.g. $Q \cdot s \cdot s \to T$, applies to t, then the corresponding trivial rule, e.g. $Q \cdot x \cdot x \to Q \cdot x \cdot x$, from R_I also applies. The other direction of the equivalence follows from the fact that $R \cup \mathrm{PROP} \cup R_I \subseteq R_\alpha$. ◀

▶ **Lemma 35.** *If $l \to r \in R_I \cup R_{II}$ and $p \neq \varepsilon$ is such that $l_{|p}$ is not a variable, then $\sigma l_{|p}$ is in R_α-normal form for any substitution σ.*

Proof. We show that $\sigma l_{|p}$ is in $R \cup \mathrm{PROP} \cup R_I$-normal form. If $l \to r \in R_I$ then $l = A_{\mathcal{T}_i} \cdot t_i$ where t_i is an R-standard term. Hence t_i is in $R \cup \mathrm{PROP} \cup R_I$-normal form. Because $p \neq \varepsilon$ and t_i is closed, this implies that $\sigma l_{|p} = l_{|p}$ is in $R \cup \mathrm{PROP} \cup R_I$-normal form as well. Analogously, if $l \to r \in R_{II}$ then the fact that $\sigma l_{|p} = l_{|p}$ is a closed term in $R \cup \mathrm{PROP} \cup R_I$-normal form follows directly from Definition 31 and from $p \neq \varepsilon$.

Therefore, an application of Lemma 34 establishes our claim. ◀

▶ **Lemma 36.** *If t is in $R \cup PROP$-normal form and $\Sigma_f(t) \subseteq \Sigma_f(R \cup PROP)$, then t is in R_α-normal form.*

Proof. From $\mathcal{H}(R_I \cup R_{II}) \cap \Sigma_f(R \cup \mathrm{PROP}) = \emptyset$ and Fact 18 it follows that t is in R_0-normal form. Lemma 34 implies that it is also in R_α-normal form. ◀

▶ **Notation 37.** We use S_α for $R_\alpha \setminus \bigcup_{\beta<\alpha} R_\beta$, $\to_{\leq\alpha}$ for $\to_{R_\alpha}$, and $\to_{=\alpha}$ for $\to_{S_\alpha}$.

We will now prove a series of lemmas which together imply that R_α and R_β commute for all $\alpha, \beta \leq \zeta$, and therefore R_ζ is confluent. The key idea in the proofs of these lemmas could be summarized by the following two diagrams.

$$\begin{tikzcd} \Pi \cdot t \arrow[r, "=\alpha"] \arrow[d, "=\beta"'] & \Pi \cdot t' \arrow[dl, dashed, "\leq\beta"] \\ T & \end{tikzcd} \quad \text{because} \quad \begin{tikzcd} t \cdot s \arrow[r, "=\alpha"] \arrow[d, "*", "\leq\gamma"'] & t' \cdot s \arrow[dl, dashed, "*", "\leq\gamma"'] \\ T & \end{tikzcd} \quad \text{by IH, for } \gamma < \beta, \text{ all } s$$

$$\begin{tikzcd} \Pi \cdot t \arrow[r, "=\beta"] \arrow[d, "=\alpha"'] & T \\ F & \end{tikzcd} \quad \text{implies} \quad \begin{tikzcd} t \cdot s \arrow[r, "\leq\gamma", "*"'] \arrow[d, "*", "\leq\delta"'] & T \\ F & \end{tikzcd} \quad \text{for } \delta < \alpha,\ \gamma < \beta, \text{ some } s$$

We adopt the notation $\to_{*\alpha}$ for $\to_{R_\alpha \setminus R_0}$ when $\alpha > 0$, and $\to_{*0}$ for $\to_{=0}$. In the following, whenever we write a reduction sequence of the form $t_0 \xrightarrow{*}_{*\alpha_1} t_1 \xrightarrow{*}_{*\alpha_2} \ldots \xrightarrow{*}_{*\alpha_n} t_n$ we tacitly assume that there exists $\alpha \leq \zeta$ such that each α_i for $i = 1, \ldots, n$ is either 0 or α, and there is no $j \in \{1, \ldots, n-1\}$ such that $\alpha_j = \alpha_{j+1}$. It is easy to see that every reduction $t \xrightarrow{*}_{\leq\alpha} s$ can be represented by a reduction sequence in this form.

▶ **Lemma 38.** *If R_0 and $R_\alpha \setminus R_0$ commute, then so do R_0 and R_α.*

Proof. Let $t \xrightarrow{*}_{=0} t_1$ and $t = s_0 \xrightarrow{*}_{*\alpha_1} s_1 \xrightarrow{*}_{*\alpha_2} \ldots \xrightarrow{*}_{*\alpha_n} s_n = t_2$ for some $n \geq 1$ and $\alpha_1, \ldots, \alpha_n \in \{0, \alpha\}$. The proof proceeds by simple induction on n. ◀

▶ **Lemma 39.** *For all $\alpha \leq \zeta$ the term rewriting systems R_0 and $R_\alpha \setminus R_0$ commute.*

Proof. We use transfinite induction on α to show that R_0 is compatible with $R_\alpha \setminus R_0 = \bigcup_{0<\beta\leq\alpha} S_\beta$.

Let $\langle u_1, u_2 \rangle \in \mathrm{Crit}(R_0, S_\beta)$ for some $0 < \beta \leq \alpha$. Because $R \cup \mathrm{PROP}$ is standard and $\Pi \notin \Sigma_f(R \cup \mathrm{PROP})$, the critical pair must be between a rule from $R_I \cup R_{II}$ and a rule from S_β. Therefore, we have rules $l_1 \to r_1 \in R_I \cup R_{II}$, $l_2 \to r_2 \in S_\beta$, a substitution σ, and a position p such that $u_1 = \sigma r_1$, $u_2 = (\sigma l_1)[\sigma r_2]_p$, $\sigma {l_1}_{|p} = \sigma l_2$, and p is such that ${l_1}_{|p}$ is not a variable. If $p = \varepsilon$ then $l_1 = r_1$ by definition of R_I and R_{II}, and we have $u_1 \to_{\leq\beta} u_2$. The case $p \neq \varepsilon$ is impossible by Lemma 35.

Now let $\langle u_2, u_1 \rangle \in \mathrm{Crit}_i(S_\beta, R_0)$ for some $0 < \beta \leq \alpha$. We have $u_2 \in \{T, F\}$. There are terms t, t' and a context C such that $u_1 = \Pi \cdot C[t']$, $t \to_{=0} t'$ and $\Pi \cdot C[t] \to^{\varepsilon}_{=\beta} u_2$. Assume $u_2 = T$. The proof for $u_2 = F$ is analogous. So let s be a term in R_0-normal form. We have $C[t] \cdot s \xrightarrow{*}_{\leq\gamma} T$ for some $\gamma < \beta$. But we may invoke the inductive hypothesis to conclude that R_0 and $R_\gamma \setminus R_0$ commute. So by Lemma 38 we obtain that R_0 and R_γ commute as well. Hence $C[t'] \cdot s \xrightarrow{*}_{\leq\gamma} T$, because $C[t] \cdot s \xrightarrow{*}_{\leq\gamma} T$, $C[t] \cdot s \to_{=0} C[t'] \cdot s$ and T is in R_0-normal form. But s was an arbitrary term in R_0-normal form, so we obtain $u_1 = \Pi \cdot C[t'] \to_{=\delta} T$ for some $0 < \delta \leq \beta$. ◀

▶ **Lemma 40.** *If $R_\alpha \setminus R_0$ and $R_\beta \setminus R_0$ commute then so do R_α and R_β.*

Proof. The proof may be easily reconstructed from the following diagram. ◀

$$\begin{tikzcd}
t \arrow[r, "*"] \arrow[r, "*\beta_1"'] \arrow[d, "*"', "*\alpha_1"] & u_1 \arrow[r, "*"] \arrow[r, "*\beta_2"'] \arrow[d, dashed, "*", "*\alpha_1"'] \arrow[rr, phantom, bend right=0, "\text{by IH}", shift right=3ex] & \cdots \arrow[r, "*"] \arrow[r, "*\beta_m"'] & t_2 \arrow[d, dashed, "*"', "*\alpha_1"] \\
s_1 \arrow[r, dashed, "*"] \arrow[r, dashed, "*\beta_1"'] \arrow[d, "*"', "*\alpha_2"] & r_0 \arrow[r, dashed, "*"] \arrow[r, dashed, "*\beta_2"'] & \cdots \arrow[r, dashed, "*"] \arrow[r, dashed, "*\beta_m"'] & r_1 \arrow[d, dashed, "*"', "*\alpha_2"] \\
\vdots \arrow[d, "*"', "*\alpha_n"] \arrow[rrr, phantom, "\text{by IH}"] & & & \vdots \arrow[d, dashed, "*"', "*\alpha_n"] \\
t_1 \arrow[r, dashed, "*"] \arrow[r, dashed, "*\beta_1"'] & {} \arrow[r, dashed, "*"] \arrow[r, dashed, "*\beta_2"'] & \cdots \arrow[r, dashed, "*"] \arrow[r, dashed, "*\beta_m"'] & r_2
\end{tikzcd}$$

▶ **Lemma 41.** *For $\alpha', \beta' \leq \zeta$ the term rewriting systems $R_{\alpha'} \setminus R_0$ and $R_{\beta'} \setminus R_0$ commute.*

Proof. We use induction on pairs $\langle \alpha', \beta' \rangle$ of indices of $R_{\alpha'}$, $R_{\beta'}$ ordered lexicographically.

Let $\langle u_2, u_1 \rangle \in \mathrm{Crit}_i(S_\beta, S_\alpha)$ for some $0 < \alpha \leq \alpha'$, $0 < \beta \leq \beta'$. The term u_2 is a constant. There are terms t, t' and a context C such that $t \to^{\varepsilon}_{=\alpha} t'$, $u_1 = \Pi \cdot C[t']$, and $\Pi \cdot C[t] \to^{\varepsilon}_{=\beta} u_2$. Assume $u_2 = F$. There is a term s in R_0-normal form such that $C[t] \cdot s \xrightarrow{*}_{\leq \gamma} F$ for some $\gamma < \beta$. By the inductive hypothesis $R_\alpha \setminus R_0$ and $R_\gamma \setminus R_0$ commute. Therefore, by Lemma 40 we may conclude that $\to_{\leq \alpha}$ and $\to_{\leq \gamma}$ commute. Hence $C[t'] \cdot s \xrightarrow{*}_{\leq \gamma} F$, because $C[t] \cdot s \xrightarrow{*}_{\leq \gamma} F$, $C[t] \cdot s \to_{=\alpha} C[t'] \cdot s$ and F is in R_α-normal form. Therefore, $u_1 = \Pi \cdot C[t'] \to_{\leq \beta} F = u_2$. The argument for $u_2 = T$ is analogous.

Now let $\langle u_1, u_2 \rangle \in \mathrm{Crit}(S_\alpha, S_\beta)$ for some $\alpha \leq \alpha'$, $\beta \leq \beta'$. The case when $\langle u_1, u_2 \rangle$ is a non-root critical pair is analogous to the case we have just considered. If $\langle u_1, u_2 \rangle$ is a root critical pair, then both $u_1, u_2 \in \{T, F\}$, and we need to show that $u_1 = u_2$. It may happen otherwise only when there is a term t such that $\Pi \cdot t \to_{=\alpha} u_1$, $\Pi \cdot t \to_{=\beta} u_2$. Without loss of generality assume $u_1 = T$, $u_2 = F$. So there is a closed term s in R_0-normal form such that $t \cdot s \xrightarrow{*}_{\leq \delta} T$ and $t \cdot s \xrightarrow{*}_{\leq \gamma} F$ for some $\delta < \alpha$, $\gamma < \beta$. The inductive hypothesis and Lemma 40 imply that $\to_{\leq \delta}$ and $\to_{\leq \gamma}$ commute, which gives a contradiction.

We have thus shown that $R_{\alpha'} \setminus R_0$ and $R_{\beta'} \setminus R_0$ are compatible, so they commute by left-linearity and Theorem 25. ◀

▶ **Corollary 42.** *The term rewriting system R_ζ has the Church-Rosser property.*

▶ **Theorem 43.** *Let R be a standard FTRS compatible with PROP, and $\mathcal{T}_{\mathbb{I}} = \{\mathcal{T}_i \mid i \in \mathbb{T}\}$ be a family of sets of R-standard terms. The applicative algebra $\mathcal{A}_{R_\zeta}$ generated by $R_\zeta^{\mathcal{T}_{\mathbb{I}}}$ is a FO-ICA such that for each $i \in \mathbb{I}$ the set $\{[t]_{R_\zeta} \mid t \in \mathcal{T}_i\}$ is a type represented by $[A_{\mathcal{T}_i}]_{R_\zeta}$ which is a δ-total combinator. Furthermore, if $t_1, t_2 \in \mathcal{L}(\Sigma)$ are in $R \cup PROP$-normal form, $t_1 \neq t_2$, and $\Sigma_f(t_1), \Sigma_f(t_2) \subseteq \Sigma_f(R \cup PROP)$, then $[t_1]_{R_\zeta}, [t_2]_{R_\zeta} \in \delta(\mathcal{A}_{R_\zeta})$ and $[t_1]_{R_\zeta} \neq [t_2]_{R_\zeta}$.*

Proof. First, we check that $\mathcal{A}_{R_\zeta}$ is a first-order illative combinatory algebra. To save on notation we use the same symbols for terms and corresponding abstraction classes in $\mathcal{A}_{R_\zeta}$.

- The axioms $T \neq F$ and $T, F \in \delta(A_{R_\zeta})$ follow from the Church-Rosser property of R_ζ and the fact that T and F are in R_ζ-normal form.
- The axioms (3)-(6) in Definition 2 follow directly from the definition of PROP.
- The axioms (7) and (8) follow directly from the definitions of R_I, R_{II} and from Lemma 34, which is needed to prove that $A \cdot X \in \{F\} \cup \upsilon$ for $X \in \upsilon$.

- The axioms for Π follow from the Church-Rosser property of R_ζ, Lemma 34 and the fact that $R_\zeta = \bigcup_{\alpha<\zeta} R_\alpha$.

The fact that each set $\{[t]_{R_\zeta} \mid t \in \mathcal{T}_i\}$ is a type represented by $[A_{\mathcal{T}_i}]_{R_\zeta}$ which is a δ-total combinator follows directly from Lemma 34 and the definitions of R_I and R_{II}. Finally, the last claim follows from Lemma 36 and the Church-Rosser property of R_ζ. ◀

6 Completeness of the first-order translation

In this section we prove completeness of the translation introduced in Section 3. We work under the same assumptions and definitions as in Section 3.

▶ **Theorem 44. Completeness**
Let ϕ and all formulas in Δ be closed. If $\Psi(\Delta) \models \Psi(\phi)$ then $\Delta \models_{FO} \phi$.

Proof. Suppose $\Psi(\Delta) \models \Psi(\phi)$. Let $\mathcal{A}$ be a first-order model of Δ.

We construct a functional term rewriting system R as follows. The signature of R consists of all elements of the universe of $\mathcal{A}$, all relation and function symbols from $\mathcal{L}_{FO}$ and the constants T, F. We assume the relation and function symbols are different from T, F, P, Q, etc. For every n-ary relation $r^{\mathcal{A}}$ on $\mathcal{A}$, which interprets a relation symbol r, the rule $r \cdot a_1 \cdot \ldots \cdot a_n \to T$ belongs to R for exactly those $a_1, \ldots, a_n$ for which $r^{\mathcal{A}}(a_1, \ldots, a_n)$ holds, the rule $r \cdot a_1 \cdot \ldots \cdot a_n \to F$ when $r^{\mathcal{A}}(a_1, \ldots, a_n)$ does not hold. For every n-ary function $f^{\mathcal{A}}$ on $\mathcal{A}$, which interprets a function symbol f, the rule $f \cdot a_1 \cdot \ldots \cdot a_n \to b$ belongs to R if $f^{\mathcal{A}}(a_1, \ldots, a_n) = b$. Nothing else belongs to R.

It is straightforward to verify that R is standard and compatible with PROP. By ς we denote the universe of $\mathcal{A}$. Let $\mathcal{B}$ be the applicative algebra generated by $R_\zeta^{\{\varsigma\}}$. For convenience we use the same symbols for terms and corresponding abstraction classes. Analogously for sets of terms. By Theorem 43 the algebra $\mathcal{B}$ is a FO-ICA with a δ-total combinator A_ς representing ς, and we have $\varsigma \subseteq \delta(\mathcal{B})$. Note that $\varsigma = \iota_{\mathcal{B}}$, where $\iota_{\mathcal{B}}$ is the type represented by $A_\iota = \lambda x.(A_\varsigma x) \wedge (A_\delta x)$, as in Definition 10.

It is easy to check that $\mathcal{B}$ is an illative model of Γ_0, and that $\mathcal{A}$ and $\mathcal{B}$ are correspondent in the sense of Definition 11. Hence by Lemma 12 we may conclude that $[\![\psi]\!]_{\mathcal{A}} = [\![\Psi(\psi)]\!]_{\mathcal{B}}$ for any closed first-order formula ψ. This implies that $\mathcal{B} \models \mathrm{Im}_\Psi(\Delta)$. Therefore $\mathcal{B} \models \Psi(\Delta)$, and consequently $\mathcal{B} \models \Psi(\phi)$, which implies $\mathcal{A} \models_{FO} \phi$, because $[\![\phi]\!]_{\mathcal{A}} = [\![\Psi(\phi)]\!]_{\mathcal{B}}$. ◀

References

1. Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1999.
2. 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.
3. Alonzo Church. A set of postulates for the foundation of logic I. *Annals of Mathematics, ser. 2*, 33:346–366, 1932.
4. Haskell B. Curry. Grundlagen der kombinatorischen Logik. *American Journal of Mathematics*, 52:509–536, 789–834, 1930.
5. 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.

6 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.

7 Gerhard Jäger, Reinhard Kahle, and Thomas Strahm. On applicative theories. In A. Cantini, E. Casari, and P. Minari, editors, *Logic and Foundation of Mathematics*, pages 88–92. Kluwer Academic Publishers, 1999.

8 Gerhard Jäger and Thomas Strahm. Totality in applicative theories. *Annals of Pure and Applied Logic*, 74(2):105–120, 1995.

9 Stephen C. Kleene and J. Barkley Rosser. The inconsistency of certain formal logics. *Annals of Mathematics*, 36:630–636, 1935.

10 Saul A. Kripke. Outline of a theory of truth. *Journal of Philosophy*, 72(19):690–716, 1975.

11 Jonathan P. Seldin. The logic of Church and Curry. In Dov M. Gabbay and John Woods, editors, *Logic from Russell to Church*, volume 5 of *Handbook of the History of Logic*, pages 819–873. North-Holland, 2009.

12 Yoshihito Toyama. Commutativity of term rewriting systems. In K. Fuchi and L. Kott, editors, *Programming of Future Generation Computers II*, pages 393–407. North-Holland, 1988.