Advances in the Theory of Computable Functionals TCF$^+$ due to its Implementation

Iosif Petrakis

Mathematisches Institut der Universität München
petrakis@mathematik.uni-muenchen.de

Abstract

The constructive formal theory of computable functionals TCF$^+$ was first sketched in [4]. Its main feature is the incorporation of both ideal objects (functionals) and their approximations (formal neighborhoods) in its language. Here we present some advances in the development of TCF$^+$ due to the implementation in the proof assistant Minlog of some of its basic concepts and results. Namely, we give new definitions of the notions related to the information-system structure $\mathbf{SC}_\iota$ connected to a base type given by a free finitary algebra ι and extend them to the arrow types. Working with the algebra of derivations $\mathbf{D}$ as a case study we describe the fully implemented proofs of $\mathbf{SC}_\mathbf{D}$ and $\mathbf{SC}_{\mathbf{D}\to\mathbf{D}}$ satisfying (an even stronger form of) the axioms of an information system. These proofs can be extended to any given finitary algebra ι. We also present an implemented proof of a point-free version of Kreisel's density theorem for the base type $\mathbf{D}$ which suggests its generalization to the arrow case. All notions involved and the line of results presented here are due to the implementation procedure.

1998 ACM Subject Classification F.4.1 Mathematical Logic

Keywords and phrases higher type computability, density theorem, information systems, interactive theorem proving

1 Introduction

The theory TCF$^+$ is a formal constructive theory of computable functionals which extends the theory TCF. The latter, guided by some fundamental ideas of Kreisel [6] and developed by the Munich logic group[1], is a variant of HA$^\omega$ and its terms extend both Gödel's $\mathbf{T}$ and Plotkin's PCF. The idea of a formal theory of computable functionals based on the partial continuous functionals as its intended domain goes back to Scott's LCF [15] and Platek's Thesis [11]. The terms of LCF denote the partial continuous functionals, which are seen as elements of the so-called Scott-Ershov domains. It is important to note though, that Scott used classical logic. Later Plotkin in [12] treated LCF as a programming language, a typed λ-calculus with constants for the fixed-point operators, and proved that the Scott model is fully abstract for PCF + por, and universal for PCF + por +$\exists$.

In contrast to LCF, TCF has the following features:

1. It uses non-flat free algebras as semantical domains for the base types. These algebras are given by their constructors, and they are injective with disjoint ranges (see [14] p.263).
2. Its underlying logic is intuitionistic (as it is explained in [4], minimal logic suffices).
3. It uses information systems instead of abstract domains for the description of the Scott model[2].

[1] Especially by H. Schwichtenberg, U. Berger and W. Bucholz. An elaborated presentation of TCF can be found in [14].
[2] Information systems were introduced by Scott in [16].

The passage from TCF to TCF$^+$ is motivated by:

1. Our need to have a formal theory better adjusted to the Scott model. The object terms of TCF$^+$ are the terms of TCF, representing the functionals, while its approximation terms represent their finite approximations (tokens and formal neighborhoods). In that way TCF$^+$ is better adjusted to the common (non-flat) Scott model than TCF.
2. The paradigm of the point-free topology (see e.g., [2]). Within higher type computability this directs to an as much as possible reconstruction of the study of the "ideal", abstract functionals (points) through the study of their concrete and finite approximations. Instrumental to this will be the use of information systems instead of abstract domains for the description of the Scott model, and the study of decidable ideals (see section 5).
3. Our need to give completely formal proofs to important theorems of higher type computability. Consider, for example, a generalized version of Kreisel's density theorem[3], according to which a finitely generated functional $\overline{U}$ (or equivalently U) can be extended to a total ideal x i.e.,
$$\forall_{U \in \mathrm{Con}_\rho} \exists_{x \in \mathbb{T}_\rho} (U \subseteq x).$$
It is clear by its formulation that in order to reveal the computational content of the density theorem we need to express in our formal language both formal neighborhoods and functionals.
4. Our need to provide fully implemented proofs of our formal proofs in the proof assistant Minlog[4]. Formalization and implementation in a proof assistant are, in our view, complementary and interdependent processes. As we show here, the implementation procedure sheds new light on the mathematical structure of the "syntactic" information systems defined in sections 3 and 4, and offers new and enriched information on the constructive content of the density theorem.

We present here some advances in the development of TCF$^+$ and its implementation with respect to [4], in which TCF$^+$ was first introduced. The main idea behind the formation of TCF$^+$ is the incorporation within its syntax of its intended Scott model. Thus to each set-theoretic information system $\mathbf{C}_\rho$, the ideals $|\mathbf{C}_\rho|$ of which form the Scott model, corresponds a "syntactic" information system $\mathbf{SC}_\rho$, an object belonging to the syntax of TCF$^+$. In [4] this program was sketched for the case of $\mathbf{SC_D}$, where $\mathbf{D}$, the algebra of derivations, was introduced as a generic case study. But neither a proof of $\mathbf{SC_D}$ being an information system nor any kind of implementation were included[5].

We report on the following advances in TCF$^+$:

1. Based on the coherence property of the information systems $\mathbf{C}_\rho$ we define in section 3 consistency in $\mathbf{SC_D}$ resting on the consistency of pairs of tokens. The implementation of this idea facilitates crucially all proofs concerning consistency. We also give a fully implemented proof of $\mathbf{SC_D}$ satisfying the axioms of an information system.
2. Extending our concepts and methods to the arrow case we give in section 4 a fully implemented proof of $\mathbf{SC_{D\to D}}$ being an information system. Although the initial non-implemented proofs used general induction (see [14], p.322), the simplicity of concepts an

[3] For the importance of this theorem see [9] and [1].

[4] Minlog [7] is developed by H. Schwichtenberg and the Munich logic group during at least the last 20 years.

[5] In [4] one can find though, informal proofs of Kreisel's general density theorem and Plotkin's definability theorem, and also directions for their proof within TCF$^+$. These proofs are the "non-flat" versions of the corresponding proofs in [13] within the flat Scott model. An elaborated version of the informal proof of Plotkin's non-flat definability theorem can be found in [5].

implementation often demands generated simpler proofs, where no general induction is used. Moreover, the implementation revealed that $\mathbf{SC_D}$ and $\mathbf{SC_{D\to D}}$ satisfy a stronger version of the axioms of an information system, a not anticipated fact which generalizes also to the information systems C_ρ of the Scott model.

3. Formalizing in section 5 the notion of a decidable ideal of $\mathbf{SC_D}$ in a point-free way, we give a fully implemented proof of the density theorem in the algebra $\mathbf{D}$, giving a completely constructive description of a total decidable ideal including a formal neighborhood of $\mathbf{D}$. In [4] a central inductive definition of $[\![\lambda_{\vec{x}}M]\!]$ is given, where $\vec{x}$ contains all free variables of a TCF-term M, such that computation rules can define an ideal in a function space. This definition is meant to connect the approximation terms with the object terms. Since one can prove informally the density theorem by proving the existence of a decidable ideal extending a formal neighborhood, this definition can be avoided. Here we define a decidable ideal as an appropriate boolean-valued function on approximation terms (tokens) avoiding object terms (points) and proving the corresponding point-free version of density theorem in D. In the last section we explain why we expect this approach to be extended to general types too. Hence, regarding density theorem, the approximation level of TCF$^+$ seems sufficient for its formalization.

2 The non-flat Scott model

An *information system*[6] (i.s.) is a structure $\mathcal{A} = (A, \mathrm{Con}, \vdash)$, where A is a set of tokens (denoted by a, b, c), $\mathrm{Con} \subseteq \mathcal{P}^{\mathrm{fin}}(A)$ is the set of formal neighborhoods (denoted by U, V, W), and $\vdash \subseteq \mathrm{Con} \times A$ is the entailment relation[7] satisfying the following axioms:

1. $\mathrm{Con}(U) \to V \subseteq U \to \mathrm{Con}(V)$,
2. $\mathrm{Con}(\{a\})$,
3. $\mathrm{Con}(U) \to U \vdash a \to \mathrm{Con}(U \cup \{a\})$,
4. $\mathrm{Con}(U) \to a \in U \to U \vdash a$,
5. $\mathrm{Con}(U) \to \mathrm{Con}(V) \to U \vDash V \to V \vdash a \to U \vdash a$,

where $U \vDash V := \forall_{a\in V}(U \vdash a)$. An *ideal* of $\mathcal{A}$ is an $x \subseteq A$ which is consistent i.e., $U \subseteq^{\mathrm{fin}} x \to \mathrm{Con}(U)$, and deductively closed i.e., $x \supseteq U \to U \vdash a \to a \in x$. If $U \in \mathrm{Con}$, then $\overline{U} = \{a \in A : U \vdash a\}$ is a *compact* ideal. The structure $(|\mathcal{A}|, \subseteq, \overline{\emptyset}, |\mathcal{A}|_0)$, where $|\mathcal{A}|$ is the set of all ideals of $\mathcal{A}$ and $|\mathcal{A}|_0$ is the set of all compact ideals of $\mathcal{A}$, is an algebraic domain[8]. The basic open sets of the Scott topology on $|\mathcal{A}|$ are the cones $\mathcal{O}_U = \{J \in |\mathcal{A}| : U \subseteq J\}$, for each $U \in |\mathcal{A}|_0$. If $\mathcal{A} = (A, \mathrm{Con}_{\mathcal{A}}, \vdash_{\mathcal{A}})$ and $\mathcal{B} = (B, \mathrm{Con}_{\mathcal{B}}, \vdash_{\mathcal{B}})$ are i.s., then one defines an arrow i.s. $\mathcal{A} \to \mathcal{B}$ such that $|\mathcal{A} \to \mathcal{B}| \cong |\mathcal{A}| \to |\mathcal{B}|$, the set of continuous functions with respect to the Scott topologies on $|\mathcal{A}|$ and $|\mathcal{B}|$ (see [14], pp. 253-259). An i.s. is called *coherent*, if U is consistent whenever each pair of its elements is consistent.

Following [14] the algebras ι considered here are of the form $\mu_\xi(K_0, \ldots, K_{l-1})$, where each constructor K_j is of type of the form

$$\vec{\rho} \to (\vec{\sigma_\nu} \to \xi)_{\nu<n} \to \xi.$$

[6] In this section we review briefly some mathematical notions and facts necessary to subsequent sections. We denote the finite subsets of a set A by $\mathcal{P}^{\mathrm{fin}}(A)$.
[7] $U \vdash a$ is intuitively understood as "the information in U is sufficient to compute the bit of data a".
[8] An *algebraic domain* $(D, \leq, \perp, D_0)$ is a consistently complete, algebraic cpo and it is the result of investigating the structure of the domains arising in the Scott model of PCF. Also, each algebraic domain is isomorphic to the ideals of an i.s. (see e.g., [17]).

Some examples of such algebras are the following:

$$\begin{aligned}
&\mathbf{B} := \mu_\xi(\xi, \xi)(\text{booleans}).\\
&\mathbf{N} := \mu_\xi(\xi, \xi \to \xi)(\text{natural numbers}).\\
&\mathbf{D} := \mu_\xi(\xi, \xi \to \xi \to \xi)(\text{derivations}).\\
&\mathbf{O} := \mu_\xi(\xi, \xi \to \xi, (\mathbf{N} \to \xi) \to \xi)(\text{ordinals}).
\end{aligned}$$

Our type system Typ is generated by $\rho, \sigma ::= \iota \mid \rho \to \sigma$, and by induction on the length of ρ one shows that a type $\rho \in$ Typ takes the form $\rho = \tau_1 \to \ldots \to \tau_n \to \iota$, where $n \geq 0$ and ι is one of the base types. We define next the set-theoretic i.s. $(\mathbf{C}_\rho)_\rho$. Since we allow (infinitary) algebras with constructors having function arguments like the algebra of ordinals $\mathbf{O}$ and its constructor Sup, we do not define $\mathbf{C}_\rho$ by recursion on the type ρ. Instead, following [14], we define it by recursion on the height of the syntactic expressions involved. We simultaneously define $C_\iota, C_{\rho\to\sigma}, \mathrm{Con}_\iota, \mathrm{Con}_{\rho\to\sigma}, \vdash_\iota$ and $\vdash_{\rho\to\sigma}$ as follows:

1. A *base type token*, $a \in C_\iota$, is a type correct constructor expression $Ca_1^* \ldots a_n^*$, where each a_i^* is an *extended token* i.e., a *proper* token or the special symbol $*_\iota$ which carries no information.
2. An *arrow type token*, $a \in C_{\rho\to\sigma}$, is a pair (U, b), where $U \in \mathrm{Con}_\rho$ and $b \in C_\sigma$.
3. A *base type formal neighborhood*, $U \in \mathrm{Con}_\iota$, is a finite set of tokens in C_ι starting with the same constructor $C^{\tau_1\to\ldots\to\tau_n\to\iota}$, i.e.,

$$U = \{Ca_{(1)1}^* \ldots a_{(1)n}^*, \ldots, Ca_{(k)1}^* \ldots a_{(k)n}^*\},$$

for some $k \in \mathbb{N}$, such that, for each $1 \leq l \leq n$,

$$U_l = \{a_{(i)l}^* : a_{(i)l}^* \text{ is a proper token in } C_{\tau_l} \wedge 1 \leq i \leq k\} \in \mathrm{Con}_{\tau_l}.$$

4. An *arrow type formal neighborhood*, $W \in \mathrm{Con}_{\rho\to\sigma}$, is a finite set of tokens in $C_{\rho\to\sigma}$ i.e., $W = \{(U_i, b_i) : i \in I\}$, for some finite set I, such that

$$\forall_{J\subseteq I}\Big(\bigcup_{j\in J} U_j \in \mathrm{Con}_\rho \to \{b_j : j \in J\} \in \mathrm{Con}_\sigma\Big).$$

5. If $U = \{Ca_{(1)1}^* \ldots a_{(1)n}^*, \ldots, Ca_{(k)1}^* \ldots a_{(k)n}^*\}$ is a base type formal neighborhood such that $k \geq 1$ and $C^{\tau_1\to\ldots\to\tau_n\to\iota}$ is a constructor, then

$$\{Ca_{(1)1}^* \ldots a_{(1)n}^*, \ldots, Ca_{(k)1}^* \ldots a_{(k)n}^*\} \vdash_\iota C'\vec{a^*} :\leftrightarrow C = C' \wedge \forall_l (U_l \vdash_{\tau_l} a_l^*),$$

where U_l is defined as above, and $U \vdash *$ is always true.
6. If $W = \{(U_i, b_i) : i \in I\}$ is an arrow type formal neighborhood, then

$$W \vdash_{\rho\to\sigma} (V, b) :\leftrightarrow WV := \{b_i : V \vDash_\rho U_i\} \vdash_\sigma b.$$

▶ **Theorem 1.** *The structure* $\mathbf{C}_\rho = (C_\rho, \mathrm{Con}_\rho, \vdash_\rho)$ *is a coherent i.s., for each* $\rho \in$ Typ.

Since the definition of $\mathbf{C}_\rho$ is given by recursion on the height of the syntactic expressions involved, the proof[9] is also done w.r.t. this height. It is simple for the base types, but for the arrow types it uses simultaneous general induction in a non trivial way; note e.g., that $WV \in \mathrm{Con}_\sigma$ is not obvious and has to be proved.

[9] All missing proofs of lemmas and theorems are found in the Appendix. Also, in the next sections a lemma ending to some axiom of i.s. contains exactly the necessary facts for the proof of that axiom.

The *non-flat Scott model* $\mathcal{S} = (S_\rho)_\rho$, or the *partial continuous functionals of type* ρ, for each $\rho \in \mathrm{Typ}$, is defined by:

$$S_\iota = |\mathbf{C}_\iota|,$$
$$S_{\rho\to\sigma} = |\mathbf{C}_{\rho\to\sigma}| \cong |\mathbf{C}_\rho| \to |\mathbf{C}_\sigma| = S_\rho \to S_\sigma.$$

3 The syntactic information system $\mathbf{SC_D}$

The starting point of TCF$^+$ is the syntactic reproduction of the systems $\mathbf{C}_\rho$ avoiding their set-theoretic character. As in [4], we work with the generic algebra of derivations $\mathbf{D}$ defining the *syntactic information systems* $\mathbf{SC_D}$ and $\mathbf{SC_{D\to D}}$. The related notions and results are easy generalizable though, to any finitary[10] algebra ι. It is important to stress that the development of all subsequent concepts and results is determined by their corresponding implementation in the proof assistant Minlog.

The algebra of derivations $\mathbf{D}$ is given by its constructors $0^{\mathbf{D}}$ (axiom) and $C^{\mathbf{D}\to\mathbf{D}\to\mathbf{D}}$ (double premise rule). The extended tokens of $\mathbf{D}$ are defined inductively by a predicate $\mathrm{Tok}_{\mathbf{D}}$ (in our metalanguage this is a predicate on the words of the alphabet $\{*, 0, C\}$) with clauses

$$\mathrm{Tok}_{\mathbf{D}}(*_{\mathbf{D}}), \quad \mathrm{Tok}_{\mathbf{D}}(0), \quad \mathrm{Tok}_{\mathbf{D}}(a_1) \to \mathrm{Tok}_{\mathbf{D}}(a_2) \to \mathrm{Tok}_{\mathbf{D}}(Ca_1a_2).$$

For simplicity we write $*$ instead of $*_{\mathbf{D}}$ and we use a, b, c, d for extended tokens of $\mathbf{D}$. To the inductively defined predicate $\mathrm{Tok}_{\mathbf{D}}$ corresponds the following elimination axiom[11]:

$$(\mathrm{TokElim}) \qquad \varphi(*) \to \varphi(0) \to \forall_{a_1,a_2}(\varphi(a_1) \to \varphi(a_2) \to \varphi(Ca_1a_2)) \to \varphi(a),$$

where φ is an arbitrary formula[12]. In order to avoid the set-theoretic notion of a finite set we use lists of extended tokens of $\mathbf{D}$ defined inductively through the predicate $\mathrm{LTok}_{\mathbf{D}}$ by the clauses

$$\mathrm{LTok}_{\mathbf{D}}(\mathrm{nil}_{\mathbf{D}}), \quad \mathrm{Tok}_{\mathbf{D}}(a) \to \mathrm{LTok}_{\mathbf{D}}(U) \to \mathrm{LTok}_{\mathbf{D}}(a ::_{\mathbf{D}} U),$$

where $\mathrm{nil}_{\mathbf{D}}$ denotes the empty list in $\mathbf{D}$ and $a ::_{\mathbf{D}} U$ denotes the list constructed by appending a to U. We write U, V for lists of extended tokens. The corresponding elimination axiom is

$$(\mathrm{LTokElim}) \qquad \Phi(\mathrm{nil}_{\mathbf{D}}) \to \forall_a \forall_U (\Phi(U) \to \Phi(a :: U)) \to \Phi(U),$$

where Φ is an arbitrary formula. Similarly we define the predicate $\mathrm{Tok}_{\mathbf{B}}$, corresponding to the proper[13] tokens of the algebra $\mathbf{B}$ given by the nullary constructors $\mathtt{tt}^{\mathbf{B}}, \mathtt{ff}^{\mathbf{B}}$. Equality in $\mathbf{B}$ is the Leibniz equality, while the total token-valued functions $\wedge_{\mathbf{B}}, \vee_{\mathbf{B}} : \mathrm{Tok}_{\mathbf{B}} \to \mathrm{Tok}_{\mathbf{B}} \to \mathrm{Tok}_{\mathbf{B}}$ are defined as expected. The predicate $\mathrm{LTok}_{\mathbf{B}}$ of lists of booleans is defined similarly to $\mathrm{LTok}_{\mathbf{D}}$. Its corresponding elimination axiom is defined as above.

Token-equality, $=_{\mathbf{D}}$: $\mathrm{Tok}_{\mathbf{D}} \to \mathrm{Tok}_{\mathbf{D}} \to \mathrm{Tok}_{\mathbf{B}}$, is defined by the clauses:

$$\begin{aligned}
&(* =_{\mathbf{D}} *) := (0 =_{\mathbf{D}} 0) := \mathtt{tt},\\
&(* =_{\mathbf{D}} 0) := (* =_{\mathbf{D}} Ca_1a_2) := (0 =_{\mathbf{D}} *) := (0 =_{\mathbf{D}} Ca_1a_2) := \mathtt{ff},\\
&(Ca_1a_2 =_{\mathbf{D}} *) := (Ca_1a_2 =_{\mathbf{D}} 0) := \mathtt{ff},\\
&(Ca_1a_2 =_{\mathbf{D}} Cb_1b_2) := (a_1 =_{\mathbf{D}} b_1) \wedge_{\mathbf{B}} (a_2 =_{\mathbf{D}} b_2),
\end{aligned}$$

[10] An algebra is called *finitary*, if for each K_j (i) only finitary algebras appear in $\vec{\rho}$ and (ii) $\vec{\sigma_\nu}$'s are empty.
[11] This is a special case of the general elimination axiom $\varphi(*_\iota) \to \forall_{\vec{a},C}(\forall_j(\varphi(\vec{a}_j)) \to \varphi(C\vec{a})) \to \varphi(a)$.
[12] It is not our intention to describe here explicitly our formal language. A sketch of it can be found in [4].
[13] For simplicity in notation we use the same symbol $\mathrm{Tok}_{\mathbf{B}}$, while its corresponding elimination axiom is $\varphi(\mathtt{tt}) \to \varphi(\mathtt{ff}) \to \varphi(\mathtt{bb})$, where $\mathtt{bb}$ denotes an arbitrary proper boolean token. Also, $\mathtt{BB}$ denotes a list of proper boolean tokens.

and appears in the obviously defined equality between lists of tokens. Since a list of tokens is the syntactic counterpart of a finite set of tokens, a syntactic notion of 'belongs to', $\dot{\in}_{\mathbf{D}} : \mathrm{Tok}_{\mathbf{D}} \to \mathrm{LTok}_{\mathbf{D}} \to \mathrm{Tok}_{\mathbf{B}}$, is defined by the clauses:

$$a \dot{\in}_{\mathbf{D}} \mathrm{nil}_{\mathbf{D}} := \mathtt{ff}, \quad \text{and} \quad a \dot{\in}_{\mathbf{D}} (b :: U) := (a = b) \vee_{\mathbf{B}} (a \dot{\in}_{\mathbf{D}} U).$$

To the subset relation corresponds the sublist relation, $\dot{\subseteq}_{\mathbf{D}} : \mathrm{LTok}_{\mathbf{D}} \to \mathrm{LTok}_{\mathbf{D}} \to \mathrm{Tok}_{\mathbf{B}}$, defined by the clauses:

$$\mathrm{nil} \dot{\subseteq}_{\mathbf{D}} V := \mathtt{tt}, \quad \text{and} \quad a ::_{\mathbf{D}} U \dot{\subseteq}_{\mathbf{D}} V := (a \dot{\in}_{\mathbf{D}} V) \wedge_{\mathbf{B}} (U \dot{\subseteq}_{\mathbf{D}} V).$$

▶ **Lemma 2.** *$\forall_U \forall_V (\forall_a (a \dot{\in}_{\mathbf{D}} U \to a \dot{\in}_{\mathbf{D}} V) \leftrightarrow U \dot{\subseteq}_{\mathbf{D}} V)$.*

Incorporating coherence in the definition of consistency we first define consistency between two tokens, $\mathrm{con} : \mathrm{Tok}_{\mathbf{D}} \to \mathrm{Tok}_{\mathbf{D}} \to \mathrm{Tok}_{\mathbf{B}}$, by the clauses:

$$\begin{aligned}
&\mathrm{con}(*, a) := \mathrm{con}(a, *) := \mathtt{tt},\\
&\mathrm{con}(0, 0) := \mathtt{tt},\\
&\mathrm{con}(0, Cab) := \mathrm{con}(Cab, 0) := \mathtt{ff},\\
&\mathrm{con}(Cab, Ccd) := \mathrm{con}(a, c) \wedge_{\mathbf{B}} \mathrm{con}(b, d).
\end{aligned}$$

Of course, the above definition is compatible to the definition of consistency in the set-theoretic $\mathbf{C_D}$. The function $\mathrm{Altcon} : \mathrm{Tok}_{\mathbf{D}} \to \mathrm{LTok}_{\mathbf{D}} \to \mathrm{Tok}_{\mathbf{B}}$ expresses the consistency of a token with all the elements of a list of tokens, and it is defined by the clauses:

$$\mathrm{Altcon}(a, \mathrm{nil}_D) := \mathtt{tt}, \quad \text{and} \quad \mathrm{Altcon}(a, b :: U) := \mathrm{con}(a, b) \wedge_{\mathbf{B}} \mathrm{Altcon}(a, U).$$

The consistency of a list of tokens, $\mathrm{Con} : \mathrm{LTok}_{\mathbf{D}} \to \mathrm{Tok}_{\mathbf{B}}$, is defined by the clauses:

$$\mathrm{Con}(\mathrm{nil}_{\mathbf{D}}) := \mathtt{tt}, \quad \text{and} \quad \mathrm{Con}(a :: U) := \mathrm{Altcon}(a, U) \wedge_{\mathbf{B}} \mathrm{Con}(U).$$

▶ **Lemma 3.** *(i) $\forall_U (\forall_a (\mathrm{Altcon}(a, U) \leftrightarrow \forall_b (b \dot{\in} U \to \mathrm{con}(a, b))))$.*
(ii) $\forall_a (\mathrm{con}(a, a))$.
(iii) (axiom 2) $\forall_a (\mathrm{Con}(a :: \mathrm{nil}))$.
(iv) $\forall_U (\forall_{a_1, a_2} (a_1 \dot{\in} U \to a_2 \dot{\in} U \to \mathrm{con}(a_1, a_2)) \leftrightarrow \mathrm{Con}(U))$.
(v) (axiom 1) $\forall_{U,V} (\mathrm{Con}(U) \to V \dot{\subseteq} U \to \mathrm{Con}(V))$.

The function $\mathrm{arg}_i : \mathrm{LTok}_{\mathbf{D}} \to \mathrm{LTok}_{\mathbf{D}}$, where $i = 1, 2$, acts on a list of tokens and outputs the list of tokens which are the i-argument of the constructor C, namely:

$$\begin{aligned}
&\mathrm{arg}_i(\mathrm{nil}_{\mathbf{D}}) := \mathrm{nil}_{\mathbf{D}},\\
&\mathrm{arg}_i(* :: U) := \mathrm{arg}_i(0 :: U) := \mathrm{arg}_i(U),\\
&\mathrm{arg}_i(Ca_1a_2 :: U) := a_i :: \mathrm{arg}_i(U).
\end{aligned}$$

The function $\mathrm{comp} : \mathrm{Tok}_{\mathbf{D}} \to \mathrm{Tok}_{\mathbf{B}}$ expresses that a token is composite i.e., it is of the form Cab, namely:

$$\mathrm{comp}(*) := \mathrm{comp}(0) := \mathtt{ff}, \text{ and } \mathrm{comp}(Ca_1a_2) := \mathtt{tt},$$

while $\mathrm{Comp} : \mathrm{LTok}_{\mathbf{D}} \to \mathrm{Tok}_{\mathbf{B}}$ expresses that a list of tokens contains a composite token, namely

$$\begin{aligned}
&\mathrm{Comp}(\mathrm{nil}) := \mathtt{ff},\\
&\mathrm{Comp}(* :: U) := \mathrm{Comp}(0 :: U) := \mathrm{Comp}(U),\\
&\mathrm{Comp}(Ca_1a_2 :: U) := \mathtt{tt},
\end{aligned}$$

Entailment in $\mathbf{D}$, $\vdash_{\mathbf{D}} : \mathrm{LTok}_{\mathbf{D}} \to \mathrm{Tok}_{\mathbf{D}} \to \mathrm{Tok}_{\mathbf{B}}$, is defined by the clauses:

$$\begin{aligned}
&(U \vdash *) := \mathtt{tt},\\
&(U \vdash 0) := (0 \dot{\in} U),\\
&(U \vdash Cab) := \mathrm{Comp}(U) \wedge_{\mathbf{B}} \mathrm{arg}_1(U) \vdash a \wedge_{\mathbf{B}} \mathrm{arg}_2(U) \vdash b.
\end{aligned}$$

▶ **Lemma 4.** *(i)* $\forall_a(\forall_U(\mathrm{Comp}(U) \to \mathrm{Comp}(a :: U)))$.
(ii) $\forall_U(\forall_{a_1,a_2}(Ca_1a_2 \dot{\in} U \to \mathrm{Comp}(U)))$.
(iii) $\forall_b(\forall_{a,U}(a \dot{\in} \mathrm{arg}_i(U) \to a \dot{\in} \mathrm{arg}_i(b :: U)))$*, for each* $i = 1, 2$.
(iv) $\forall_U(\forall_{a_1,a_2}(Ca_1a_2 \dot{\in} U \to a_i \dot{\in} \mathrm{arg}_i(U)))$*, for each* $i = 1, 2$.
(v) (axiom $4'$*)* $\forall_a(\forall_U(a \dot{\in} U \to U \vdash a))$.

▶ **Lemma 5.** *(i)* $\forall_U(\forall_a(\mathrm{Con}(U) \to a \dot{\in} U \to U \vdash 0 \to \mathrm{con}(0, a)))$.
(ii) $\forall_U(\forall_a(a \dot{\in} \mathrm{arg}_1(U) \to \exists_b(Cab \dot{\in} U)))$*, and* $\forall_U(\forall_a(a \dot{\in} \mathrm{arg}_2(U) \to \exists_b(Cba \dot{\in} U)))$.
(iii) $\forall_U(\mathrm{Con}(U) \to \mathrm{Con}(\mathrm{arg}_i(U)))$*, for each* $i = 1, 2$.
(iv) $\forall_a(\forall_U(\mathrm{Comp}(a :: U) \to \mathrm{comp}(a) \vee_{\mathbf{B}} \mathrm{Comp}(U)))$.
(v) $\forall_U(\mathrm{Comp}(U) \to \exists_a(a \dot{\in} U \wedge_{\mathbf{B}} \mathrm{comp}(a)))$.
(vi) $\forall_b(\forall_{U,a_1,a_2}(\mathrm{Con}(U) \to b \dot{\in} U \to U \vdash Ca_1a_2 \to b = * \vee_{\mathbf{B}} \mathrm{comp}(b)))$.
(vii) $\forall_a(\forall_{U,b_1,b_2}(\mathrm{Con}(U) \to a \dot{\in} U \to U \vdash Cb_1b_2 \to \mathrm{con}(Cb_1b_2, a)))$.
(viii) $\forall_a(\forall_{U,b}(\mathrm{Con}(U) \to b \dot{\in} U \to U \vdash a \to \mathrm{con}(a, b)))$.
(ix) $\forall_{U,a}(\mathrm{Con}(U) \to U \vdash a \to \mathrm{Altcon}(a, U))$.
(x) (axiom 3) $\forall_{U,a}(\mathrm{Con}(U) \to U \vdash a \to \mathrm{Con}(a :: U))$.

The function $\vDash : \mathrm{LTok}_{\mathbf{D}} \to \mathrm{LTok}_{\mathbf{D}} \to \mathrm{Tok}_{\mathbf{B}}$ expresses that a list entails all the tokens of another list, and it is defined by the clauses:

$$U \vDash \mathrm{nil} := \mathtt{tt}, \text{ and } U \vDash (a :: V) := U \vdash a \wedge_{\mathbf{B}} U \vDash V.$$

▶ **Lemma 6.** *(i)* $\forall_a(\forall_U(\mathrm{comp}(a) \to U \vdash a \to \mathrm{Comp}(U)))$.
(ii) $\forall_U(\forall_V(V \vDash U \to \mathrm{Comp}(U) \to \mathrm{Comp}(V)))$.
(iii) $\forall_U(\forall_V(V \vDash U \to \mathrm{arg}_i(V) \vDash \mathrm{arg}_i(U)))$*, for each* $i = 1, 2$.
(iv) $\forall_U(\forall_V(V \vDash U \leftrightarrow \forall_a(a \dot{\in} U \to V \vdash a)))$.
(v) (axiom $5'$*)* $\forall_a(\forall_{U,V}(U \vDash V \to V \vdash a \to U \vdash a))$.

Proof. (i) We apply TokElim on the obvious $\varphi(a)$.
(ii) We apply LTokElim on the obvious $\Phi(U)$. The case $\Phi(\mathrm{nil})$ is proved through the ex-falso-quodlibet (Efq). For the inductive step we suppose $\Phi(U)$, $V \vDash b :: U$ and $\mathrm{Comp}(b :: U)$, which by Lemma 5(iv) translates into $\mathrm{comp}(b)$ or $\mathrm{Comp}(U)$. In the first case we get by (i) $\mathrm{Comp}(V)$, since $V \vdash b$, while in the second we use $\Phi(U)$ and the fact that $V \vDash U$.
(iii) We apply LTokElim on the obvious $\Phi_i(U)$. The case $\Phi_i(\mathrm{nil})$ is proved trivially by the definitions of arg_i and $\vDash$. For the inductive step we apply TokElim on

$$\varphi_i(b) := V \vDash (b :: U) \to \mathrm{arg}_i(V) \vDash \mathrm{arg}_i(b :: U).$$

The cases $\varphi_i(*), \varphi_i(0)$ are proved directly from the hypothesis $\Phi_i(U)$ and the fact that $V \vDash (b :: U) \to V \vDash U$. For the case $\varphi_i(Ca_1a_2)$ we need to show that

$$\mathrm{arg}_i(V) \vDash \mathrm{arg}_i(Ca_1a_2 :: U) = a_i :: \mathrm{arg}_i(U) := \mathrm{arg}_i(V) \vdash a_i \wedge \mathrm{arg}_i(V) \vDash \mathrm{arg}_i(U).$$

The first conjunct is proved by the definition of $V \vdash Ca_1a_2$ $(V \vDash (b :: U) \to V \vdash b)$, while the second by the hypothesis $\Phi(U)$.
(iv) We show the direction $(\to)$ and the converse is proved similarly. We apply LTokElim on

the obvious $\Phi(U)$. The case $\Phi(\text{nil})$ is proved trivially through Efq. For the inductive step we suppose that $V \vDash (b :: U) := V \vdash b \wedge V \vDash U$ and that $a\dot{\in}(b :: U) := a = b \vee a\dot{\in}U$. If $a = b$, we get $V \vdash a$ by $V \vdash b$. If $a\dot{\in}U$, we use the hypothesis $\Phi(U)$ and the fact that $V \vDash U$.
(v) We apply TokElim on the obvious $\varphi(a)$.The case $\varphi(*)$ is proved automatically, while the case $\varphi(0)$ is proved by the definition of $V \vdash 0$ and the use of (iv). For the case $\varphi(Ca_1a_2)$ we suppose that $U \vDash V$, $V \vdash Ca_1a_2$ and we show that $U \vdash Ca_1a_2$ i.e., $\text{Comp}(U)$ and $\text{arg}_i(U) \vdash a_i$, for each $i = 1, 2$. We get $\text{Comp}(U)$ by (ii), since by (i) $V \vdash Ca_1a_2 \to \text{Comp}(V)$. If we apply $\varphi(a_i)$ on $\text{arg}_i(U), \text{arg}_i(V)$ i.e.,

$$\text{arg}_i(U) \vDash \text{arg}_i(V) \to \text{arg}_i(V) \vdash a_i \to \text{arg}_i(U) \vdash a_i,$$

we get the required $\text{arg}_i(U) \vdash a_i$, since by (iii) we have $\text{arg}_i(U) \vDash \text{arg}_i(V)$.
◀

▶ **Theorem 7.** *The syntactic system* $\mathbf{SC_D} = (\text{Tok}_\mathbf{D}, \text{Con}_\mathbf{D}, \vdash_\mathbf{D})$ *satisfies the axioms of a coherent information system. Moreover, the entailment relation satisfies the stronger axioms* $4'$ *and* $5'$ *of an information system*[14]*, namely*

$4'$. $\forall_a(\forall_U(a\dot{\in}U \to U \vdash a))$,
$5'$. $\forall_a(\forall_{U,V}(U \vDash V \to V \vdash a \to U \vdash a))$.

Proof. The axioms 1 and 2 are Lemma 3(iii) and Lemma 3(v), respectively. The axiom 3 is Lemma 5(x), while the axiom $4'$ is Lemma 4(v) and the axiom $5'$ is Lemma 6(v). Finally, coherence is Lemma 3(iv). ◀

The function $+\!\!+_\mathbf{D} : \text{LTok}_\mathbf{D} \to \text{LTok}_\mathbf{D} \to \text{LTok}_\mathbf{D}$ appends U to V, corresponding in that way to the union of two finite sets, and it is defined by the clauses:

$$\text{nil}{+\!\!+}U := U, \quad \text{and} \quad (b :: V){+\!\!+}U := b :: (V{+\!\!+}U).$$

▶ **Corollary 8.** *(i)* $\forall_U(U \vDash U)$.
(ii) $\forall_{U_1,U_2,U_3}(U_1 \vDash U_2 \to U_2 \vDash U_3 \to U_1 \vDash U_3)$.
(iii) $\forall_U(\forall_V(\text{Con}(V) \to V \vDash U \to \text{Con}(V{+\!\!+}U)))$.
(iv) $\forall_U(\forall_V(\text{Con}(V) \to V \vDash U \to \text{Con}(U)))$.

4 The syntactic information system $\mathbf{SC_{D\to D}}$

While the object-terms M^ρ of TCF$^+$ represent ideals of type $\rho \in \text{Typ}$, the approximation-terms of TCF$^+$ are tokens Tok_ρ of type ρ, lists of tokens LTok_ρ of type ρ, or even lists of lists of tokens LLTok_ρ of type ρ. Thus, in order to incorporate to our syntax the definition of $\mathcal{A} \to \mathcal{B}$ we define

$$\text{Tok}_{\mathbf{D}\to\mathbf{D}} := \text{LTok}_\mathbf{D} \times \text{Tok}_\mathbf{D},$$

while we denote the tokens of type $\text{Tok}_{\mathbf{D}\to\mathbf{D}}$ by u, w and the lists of type $\text{LTok}_{\mathbf{D}\to\mathbf{D}}$ by W. We use the same symbols for the similarly to the $\mathbf{D}$-case defined functions $w\dot{\in}W$ and $W_1\dot{\subseteq}W_2$. The obvious projections $\text{lft} : \text{Tok}_{\mathbf{D}\to\mathbf{D}} \to \text{LTok}_\mathbf{D}$ and $\text{rht} : \text{Tok}_{\mathbf{D}\to\mathbf{D}} \to \text{Tok}_D$, defined by the clauses

$$\text{lft}(U, a) := U, \quad \text{and} \quad \text{rht}(U, a) := a,$$

[14] That axioms 4 and 5 of an information system are satisfied by $\mathbf{SC_D}$ without the corresponding consistency hypotheses is due to the fact that entailment is defined on all lists of tokens and not only on consistent ones.

are extended to the functions One : $\mathrm{LTok}_{\mathbf{D}\to\mathbf{D}} \to \mathrm{LLTok}_{\mathbf{D}}$, and Two : $\mathrm{LTok}_{\mathbf{D}\to\mathbf{D}} \to \mathrm{LTok}_{\mathbf{D}}$:

$$\mathrm{One}(\mathrm{Nil}_{\mathbf{D}\to\mathbf{D}}) := \mathrm{nnil}_{\mathbf{D}}, \quad \text{and} \quad \mathrm{One}(w :: W) := \mathrm{lft}(w) :: \mathrm{One}(W),$$

$$\mathrm{Two}(\mathrm{Nil}_{\mathbf{D}\to\mathbf{D}}) := \mathrm{nil}_{\mathbf{D}}, \quad \text{and} \quad \mathrm{Two}(w :: W) := \mathrm{rht}(w) :: \mathrm{Two}(W),$$

respectively, where $\mathrm{Nil}_{\mathbf{D}\to\mathbf{D}}$, or Nil for simplicity, is the empty list of type $\mathrm{LTok}_{\mathbf{D}\to\mathbf{D}}$, and $\mathrm{nnil}_{\mathbf{D}}$, or nnil, is the empty list of type $\mathrm{LLTok}_{\mathbf{D}}$ (also, we use $\mathbb{U}$ or $\mathbb{V}$ to denote the lists of type $\mathrm{LLTok}_{\mathbf{D}}$). In order to define consistency in $\mathbf{D} \to \mathbf{D}$ we follow the pattern of definition of consistency in $\mathbf{D}$. If $\to_{\mathbf{B}}$: $\mathrm{Tok}_{\mathbf{B}} \to \mathrm{Tok}_{\mathbf{B}} \to \mathrm{Tok}_{\mathbf{B}}$ is the obviously defined total boolean implication, we define arrcon : $\mathrm{Tok}_{\mathbf{D}\to\mathbf{D}} \to \mathrm{Tok}_{\mathbf{D}\to\mathbf{D}} \to \mathrm{Tok}_{\mathbf{B}}$, arrAltcon : $\mathrm{Tok}_{\mathbf{D}\to\mathbf{D}} \to \mathrm{LTok}_{\mathbf{D}\to\mathbf{D}} \to \mathrm{Tok}_{\mathbf{B}}$ and arrCon : $\mathrm{LTok}_{\mathbf{D}\to\mathbf{D}} \to \mathrm{Tok}_{\mathbf{B}}$, respectively by the clauses:

$$\mathrm{arrcon}(u, w) := \mathrm{Con}(\mathrm{lft}(u) +\!\!+ \mathrm{lft}(w)) \to_{\mathbf{B}} \mathrm{con}(\mathrm{rht}(u), \mathrm{rht}(w)),$$

$$\mathrm{arrAltcon}(w, \mathrm{Nil}) := \mathtt{tt}, \quad \text{and} \quad \mathrm{arrAltcon}(u, w :: W) := \mathrm{arrcon}(u, w) \wedge_{\mathbf{B}} \mathrm{arrAltcon}(u, W),$$

$$\mathrm{arrCon}(\mathrm{Nil}) := \mathtt{tt}, \quad \text{and} \quad \mathrm{arrCon}(w :: W) := \mathrm{arrAltcon}(w, W) \wedge_{\mathbf{B}} \mathrm{arrCon}(W).$$

▶ **Lemma 9.** *(i)* $\forall_W(\forall_u(\mathrm{arrAltcon}(u, W) \leftrightarrow \forall_w(w \dot{\in} W \to \mathrm{arrcon}(u, w))))$.
(ii) $\forall_w(\mathrm{arrcon}(w, w))$.
(iii) (axiom 2) $\forall_w(\mathrm{arrCon}(w :: \mathrm{Nil}))$.
(iv) $\forall_W(\forall_{w_1,w_2}(w_1 \dot{\in} W \to w_2 \dot{\in} W \to \mathrm{arrcon}(w_1, w_2)) \leftrightarrow \mathrm{arrCon}(W))$.
(v) (axiom 1) $\forall_{W_1,W_2}(\mathrm{arrCon}(W_1) \to W_2 \dot{\subseteq} W_1 \to \mathrm{arrCon}(W_2))$.

In order to reproduce syntactically the entailment $\vdash_{\mathcal{A}\to\mathcal{B}}$ we need to formalize the *application* WV of W on V, where $W = \{(U_1, b_1), \ldots, (U_t, b_t)\}$, $V \in \mathrm{Con}_{\mathcal{A}}$ and $WV = \{b_i : V \vDash_{\mathcal{A}} U_i\}$. As in the $\mathbf{D}$-case we define arrow-entailment independently from consistency. First we express how from a list of tokens $(a_1 :: \ldots :: a_m)$ we take the list of all a_j's which correspond to the $\mathtt{tt}$-values of a given list of booleans $(\mathbb{b}_1 :: \ldots :: \mathbb{b}_m)$. For that we define the function PTS : $\mathrm{LTok}_{\mathbf{B}} \to \mathrm{LTok}_{\mathbf{D}} \to \mathrm{LTok}_{\mathbf{D}}$ by the clauses

$$\begin{aligned}
&\mathrm{PTS}(\mathrm{nil}_{\mathbf{B}}, U) := \mathrm{nil}_{\mathbf{D}},\\
&\mathrm{PTS}(\mathbb{BB}, \mathrm{nil}_{\mathbf{D}}) := \mathrm{nil}_{\mathbf{D}},\\
&\mathrm{PTS}(\mathtt{tt} :: \mathbb{BB}, a :: U) := a :: \mathrm{PTS}(\mathbb{BB}, U),\\
&\mathrm{PTS}(\mathtt{ff} :: \mathbb{BB}, a :: U) := \mathrm{PTS}(\mathbb{BB}, U).
\end{aligned}$$

The function AltEntList : $\mathrm{LTok}_{\mathbf{D}} \to \mathrm{LLTok}_{\mathbf{D}} \to \mathrm{LTok}_{\mathbf{B}}$ outputs the list of booleans $U \vDash V$, where $V \dot{\in} \mathbb{U}$, and it is defined by the clauses

$$\begin{aligned}
&\mathrm{AltEntList}(U, \mathrm{nnil}) := \mathrm{nil}_{\mathbf{B}},\\
&\mathrm{AltEntList}(U, V :: \mathbb{U}) := (U \vDash V) :: \mathrm{AltEntList}(U, \mathbb{U}).
\end{aligned}$$

The application WV is defined as the function App : $\mathrm{LTok}_{\mathbf{D}\to\mathbf{D}} \to \mathrm{LTok}_{\mathbf{D}} \to \mathrm{LTok}_{\mathbf{D}}$, where

$$\mathrm{App}(W, V) := \mathrm{PTS}(\mathrm{AltEntList}(V, \mathrm{One}(W)), \mathrm{Two}(W)).$$

Entailment in $\mathbf{D} \to \mathbf{D}$ is defined by the function $\vdash_{\mathbf{D}\to\mathbf{D}}$: $\mathrm{LTok}_{\mathbf{D}\to\mathbf{D}} \to \mathrm{Tok}_{\mathbf{D}\to\mathbf{D}} \to \mathrm{Tok}_{\mathbf{B}}$, where[15]

$$W \vdash u := \mathrm{App}(W, \mathrm{lft}(u)) \vdash \mathrm{rht}(u).$$

[15] For simplicity we omit the subscript from the entailment symbol.

▶ **Lemma 10.** *(i)* $\forall_{W,u}(u :: W \vdash u)$.
(ii) $\forall_{W,u,w}(W \vdash w \to u :: W \vdash w)$.
(iii) (axiom $4'$*)* $\forall_W(\forall_u((u\dot{\in}W \to W \vdash u))$.

▶ **Lemma 11.** *(i)* $\forall_U(\forall_{V_1,V_2,\mathbb{U}}(V_1 \vDash V_2 \to$

$$\mathrm{PTS}(\mathrm{AltEntList}(V_2, \mathbb{U}), U) \dot{\subseteq} \mathrm{PTS}(\mathrm{AltEntList}(V_1, \mathbb{U}), U))).$$

(ii) $\forall_{U_1,U_2,W,a}(U_1 \vDash U_2 \to \mathrm{App}(W, U_2) \vdash a \to \mathrm{App}(W, U_1) \vdash a)$.
(iii) $\forall_{W_1}(\forall_{W_2,V}(W_2 \vDash W_1 \to \mathrm{App}(W_2, V) \vDash \mathrm{App}(W_1, V)))$.
(iv) (axiom $5'$*)* $\forall_{W_1,W_2,u}(W_1 \vDash W_2 \to W_2 \vdash u \to W_1 \vdash u)$.

We use n, m to denote tokens of the algebra of naturals $\mathbf{N}$, while their inequality $<: \mathrm{Tok}_\mathbf{N} \to \mathrm{Tok}_\mathbf{N} \to \mathrm{Tok}_\mathbf{B}$ is defined by the clauses (i) $n < 0_\mathbf{N} := \mathtt{ff}$, (ii) $0_\mathbf{N} < S(n) := \mathtt{tt}$, (iii) $S(n_1) < S(n_2) := n_1 < n_2$. The length $|.|$ of a list of an arbitrary type is defined in the obvious way. Additionally, we can define the projection functions on lists of tokens of arbitrary type[16].

▶ **Lemma 12.** *(i)* $\forall_W(\forall_n(n < |W| \to \mathrm{Proj}(n, W)\dot{\in}W))$.
(iia) $\forall_W(\forall_n(n < |W| \to \mathrm{Proj}(n, \mathrm{One}(W)) = \mathrm{lft}(\mathrm{Proj}(n, W))))$.
(iib) $\forall_W(\forall_n(n < |W| \to \mathrm{Proj}(n, \mathrm{Two}(W)) = \mathrm{rht}(\mathrm{Proj}(n, W))))$.
(iii) $\forall_\mathbb{U}(\forall_{n,U}(n < |\mathbb{U}| \to \mathrm{Proj}(n, \mathrm{AltEntList}(U, \mathbb{U})) = (U \vDash \mathrm{Proj}(n, \mathbb{U}))))$.
(iv) $\forall_U(\forall_{\mathbb{B},b}(b\dot{\in}\mathrm{PTS}(\mathbb{B}, U) \to \exists_n(n < |\mathbb{B}| \wedge \mathrm{Proj}(n, \mathbb{B}) = \mathtt{tt} \wedge \mathrm{Proj}(n, U) = b)))$.
(v) $\forall_{b,W,U}(b\dot{\in}\mathrm{App}(W, U) \to \exists_n(n < |W| \wedge U \vDash \mathrm{lft}(\mathrm{Proj}(n, W)) \wedge \mathrm{rht}(\mathrm{Proj}(n, W)) = b))$.
(vi) $\forall_{W,U}(\mathrm{arrCon}(W) \to \mathrm{Con}(U) \to \mathrm{Con}(\mathrm{App}(W, U)))$.
(vii) $\forall_W(\forall_{u,U}(u\dot{\in}W \to \mathrm{rht}(u)\dot{\in}\mathrm{App}(W, U{+\!\!+}\mathrm{lft}(u))))$.
(viii) $\forall_{W,u,w}(\mathrm{arrCon}(W) \to u\dot{\in}W \to W \vdash w \to \mathrm{arrcon}(w, u))$.
(ix) $\forall_{W,u}(\mathrm{arrCon}(W) \to W \vdash u \to \mathrm{arrAltcon}(u, W))$.
(x) (axiom 3*)* $\forall_{W,u}(\mathrm{arrCon}(W) \to W \vdash u \to \mathrm{arrCon}(u :: W))$.

Proof. (vi) If $(W = \mathrm{Nil}) = \mathtt{tt}$, then $\mathrm{App}(\mathrm{Nil}, U) = \mathrm{nil}$, and we have $\mathrm{Con}(\mathrm{nil})$ by definition. If $(W = \mathrm{Nil}) = \mathtt{ff}$, then, by Lemma 3(iv), we show $b_1, b_2\dot{\in}\mathrm{App}(W, U) \to \mathrm{con}(b_1, b_2)$. By (v) we get

$$\exists_{n_i}(n_i < |W| \wedge U \vDash \mathrm{lft}(\mathrm{Proj}(n_i, W)) \wedge \mathrm{rht}(\mathrm{Proj}(n_i, W)) = b_i),$$

for each $i = 1, 2$. Next we show $\mathrm{Con}(\mathrm{lft}(\mathrm{Proj}(n_1, W)){+\!\!+}\mathrm{lft}(\mathrm{Proj}(n_2, W)))$ as follows:

$$U \vDash \mathrm{lft}(\mathrm{Proj}(n_1, W)) \overset{\mathrm{Cor.8(iii)}}{\longrightarrow} \mathrm{Con}(U{+\!\!+}\mathrm{lft}(\mathrm{Proj}(n_1, W))),$$

$$\begin{aligned} U \vDash \mathrm{lft}(\mathrm{Proj}(n_2, W)) &\overset{\mathrm{Lm.6(v)}}{\longrightarrow} U{+\!\!+}\mathrm{lft}(\mathrm{Proj}(n_1, W)) \vDash \mathrm{lft}(\mathrm{Proj}(n_2, W)) \\ &\overset{\mathrm{Cor.8(iii)}}{\longrightarrow} \mathrm{Con}(U{+\!\!+}\mathrm{lft}(\mathrm{Proj}(n_1, W)){+\!\!+}\mathrm{lft}(\mathrm{Proj}(n_2, W))) \\ &\overset{\mathrm{Lm.3(v)}}{\longrightarrow} \mathrm{Con}(\mathrm{lft}(\mathrm{Proj}(n_1, W)){+\!\!+}\mathrm{lft}(\mathrm{Proj}(n_2, W))). \end{aligned}$$

By (i) $\mathrm{Proj}(n_i, W)\dot{\in}W$, therefore, by Lemma 9(iv) and the definition of arrcon we get

$$\mathrm{con}(\mathrm{rht}(\mathrm{Proj}(n_1, W)), \mathrm{rht}(\mathrm{Proj}(n_2, W))) \leftrightarrow \mathrm{con}(b_1, b_2).$$

[16] E.g., $\mathrm{Proj} : \mathbf{N} \to \mathrm{LTok}_\mathbf{D} \to \mathrm{Tok}_\mathbf{D}$ is defined by the clauses (i) $\mathrm{Proj}(0_\mathbf{N}, \mathrm{nil}) := *$, (ii) $\mathrm{Proj}(0_\mathbf{N}, a :: U) := a$, (iii) $\mathrm{Proj}(n + 1, a :: U) := \mathrm{Proj}(n, U)$. The first clause plays no role to what follows.

(vii) We apply $\mathrm{LTokElim}_{\mathbf{D}\to\mathbf{D}}$ on the appropriate $\Phi(W)$. The case $\Phi(\mathrm{Nil})$ is proved by Efq. Next we suppose that $\Phi(W)$ and we show $\Phi(w :: W)$ i.e.,

$$u \dot\in (w :: W) \to \mathrm{rht}(u) \dot\in \mathrm{App}(w :: W, U +\!\!+ \mathrm{lft}(u)).$$

By the definition of the notions involved we have that

$$\begin{aligned}
&\mathrm{App}(w :: W, U +\!\!+ \mathrm{lft}(u)) = \\
&\mathrm{PTS}(\mathrm{AltEntList}(U +\!\!+ \mathrm{lft}(u), \mathrm{One}(w :: W)), \mathrm{Two}(w :: W)) = \\
&\mathrm{PTS}(\mathrm{AltEntList}(U +\!\!+ \mathrm{lft}(u), \mathrm{lft}(w) :: \mathrm{One}(W)), \mathrm{rht}(w) :: \mathrm{Two}(W)) = \\
&\mathrm{PTS}((U +\!\!+ \mathrm{lft}(u) \vDash \mathrm{lft}(w)) :: \mathrm{AltEntList}(U +\!\!+ \mathrm{lft}(u), \mathrm{One}(W)), \mathrm{rht}(w) :: \mathrm{Two}(W)) \overset{u=w}{=} \\
&\mathrm{rht}(w) :: \mathrm{PTS}(\mathrm{AltEntList}(U +\!\!+ \mathrm{lft}(u), \mathrm{One}(W)), \mathrm{Two}(W)),
\end{aligned}$$

since $\mathrm{lft}(u) \dot\subseteq U +\!\!+ \mathrm{lft}(u) \to U +\!\!+ \mathrm{lft}(u) \vDash \mathrm{lft}(u)$. In case $u \dot\in W$, then by the inductive hypothesis $\Phi(W)$ we get

$$\begin{aligned}
&\mathrm{rht}(u) \dot\in \mathrm{PTS}(\mathrm{AltEntList}(U +\!\!+ \mathrm{lft}(u), \mathrm{One}(W)), \mathrm{Two}(W)) \to \\
&\mathrm{rht}(u) \dot\in \mathrm{PTS}((U +\!\!+ \mathrm{lft}(u) \vDash \mathrm{lft}(w)) :: \mathrm{AltEntList}(U +\!\!+ \mathrm{lft}(u), \mathrm{One}(W)), \mathrm{rht}(w) :: \mathrm{Two}(W)),
\end{aligned}$$

as a special case of the direct from the definitions of the related notions fact

$$b \dot\in \mathrm{PTS}(\mathbb{B}, U) \to b \dot\in \mathrm{PTS}(\mathbb{b} :: \mathbb{B}, a :: U).$$

◀

▶ **Theorem 13.** *The syntactic system* $\mathbf{SC}_{\mathbf{D}\to\mathbf{D}} = (\mathrm{Tok}_{\mathbf{D}\to\mathbf{D}}, \mathrm{Con}_{\mathbf{D}\to\mathbf{D}}, \vdash_{\mathbf{D}\to\mathbf{D}})$ *satisfies the axioms of a coherent information system. Moreover, the entailment relation satisfies the stronger axioms* $4'$ *and* $5'$ *of an information system*[17]*.*

Proof. The axioms 1 and 2 are Lemma 9(iii) and Lemma 9(v), respectively. The axiom 3 is Lemma 12(x), while the axiom $4'$ is Lemma 10(iii) and the axiom $5'$ is Lemma 11(iv). Finally, coherence is Lemma 9(iv). ◀

The fact that $\mathbf{SC}_{\mathbf{D}}$ and $\mathbf{SC}_{\mathbf{D}\to\mathbf{D}}$ satisfy the axioms $4', 5'$ motivated the next result on $\mathbf{C}_\rho$.

▶ **Theorem 14.** *The information systems* $\mathbf{C}_\rho$ *satisfy the axioms* $4', 5'$*, for each type* ρ*.*

5 A point-free density theorem in D

Kreisel's density theorem asserts the existence of a total functional x^ρ (point) extending a given compact functional $\overline{U}$ (or equivalently U) of type ρ. A formal proof of it within TCF^+ should involve the construction of an object term M^ρ such that M^ρ "includes" (this should be a formally defined relation between lists of type ρ and terms M^ρ) a given list U of that type. But since one proves that x^ρ can also be decidable (see in [4] the based on [1] informal proof of density theorem), one needs for the formalization of density theorem only a formal representation of decidable ideals. This representation can be given without referring to points (object terms), but only through boolean-valued functions on approximation terms. Here we show how such a point-free notion of a decidable ideal in $\mathbf{D}$ leads to a proof of the

[17] See Theorem 7.

corresponding point-free density theorem. As we indicate in the last section, these notions and proofs are expected to be extended to general types too. This is an example of the “as much as possible reconstruction of the study of points through their finite approximations” that we mentioned in the introduction as a motivation for TCF$^+$.

A decidable ideal in $\mathbf{D}$ is an “appropriate” function $I : \mathrm{Tok}_{\mathbf{D}} \to \mathrm{Tok}_{\mathbf{B}}$. In order to explain what “appropriate” here means we introduce, keeping for simplicity the same notation, elementhood $\dot{\in} : \mathrm{Tok}_{\mathbf{D}} \to (\mathrm{Tok}_{\mathbf{D}} \to \mathrm{Tok}_{\mathbf{B}}) \to \mathrm{Tok}_{\mathbf{B}}$ defined by

$$a \dot{\in} I := I(a),$$

and inclusion $\dot{\subseteq} : \mathrm{LTok}_D \to (\mathrm{Tok}_{\mathbf{D}} \to \mathrm{Tok}_{\mathbf{B}}) \to \mathrm{Tok}_{\mathbf{B}}$ defined by

$$\mathrm{nil}_{\mathbf{D}} \dot{\subseteq} I := \mathtt{tt} \ \text{ and } \ (a :: U) \dot{\subseteq} I := a \dot{\in} I \wedge_{\mathbf{B}} U \dot{\subseteq} I.$$

Next we describe the notion of a decidable ideal at a “local” level i.e., we explain how an ideal I behaves on a list U included in I and on a token a entailed by such a list U. Namely, we define $\mathrm{Ideal} : (\mathrm{Tok}_{\mathbf{D}} \to \mathrm{Tok}_{\mathbf{B}}) \to \mathrm{LTok}_{\mathbf{D}} \to \mathrm{Tok}_{\mathbf{D}} \to \mathrm{Tok}_{\mathbf{B}}$ by the clause

$$\mathrm{Ideal}(I, U, a) := [U \dot{\subseteq} I \to_{\mathbf{B}} \mathrm{Con}(U)] \ \wedge_{\mathbf{B}} \ [(U \dot{\subseteq} I \wedge_{\mathbf{B}} U \vdash a) \to_{\mathbf{B}} a \dot{\in} I].$$

The following lemma expresses the expected fact, that if V is consistent, then $\overline{V}$, which is another notation for the function $\vdash_{\mathbf{D}} (V) : \mathrm{Tok}_{\mathbf{D}} \to \mathrm{Tok}_{\mathbf{B}}$, is an ideal at a “global” level i.e., $\overline{V}$ satisfies the local ideal-property for each U and a.

▶ **Lemma 15.** *(i)* $\forall_U(\forall_V(U \dot{\subseteq} \overline{V} \leftrightarrow V \vDash U))$.
(ii) $\forall_V(\mathrm{Con}(V) \to \forall_{U,a}(\mathrm{Ideal}(\overline{V}, U, a)))$.

The function $\mathrm{total} : \mathrm{Tok}_{\mathbf{D}} \to \mathrm{Tok}_{\mathbf{B}}$ expresses that the tree corresponding to a total token contains no $*$, and it is defined by the clauses:

$$\mathrm{total}(*) := \mathtt{ff}, \quad \mathrm{total}(0) := \mathtt{tt}, \text{ and } \mathrm{total}(Cab) := \mathrm{total}(a) \wedge_{\mathbf{B}} \mathrm{total}(b).$$

The totalization function, $\mathrm{t} : \mathrm{Tok}_{\mathbf{D}} \to \mathrm{Tok}_{\mathbf{D}}$, mapping a token to a total one, is defined by:

$$\mathrm{t}(*) := \mathrm{t}(0) := 0, \text{ and } \mathrm{t}(Cab) := C\mathrm{t}(a)\mathrm{t}(b).$$

▶ **Lemma 16.** *(i)* $\forall_a(\mathrm{total}(\mathrm{t}(a)))$.
(ii) $\forall_a(\mathrm{t}(a) :: \mathrm{nil} \vdash a)$.

The main idea behind the proof of density theorem in $\mathbf{D}$ is to find a total token a_U which entails a given consistent list U. We determine a_U in two steps; first we find a token b entailing U and then a_U is defined as the totalization of b. The initial step in the construction of b is the definition of a token $\sup(a, b)$ having the maximum “common information” between a and b. Namely, $\sup : \mathrm{Tok}_{\mathbf{D}} \to \mathrm{Tok}_{\mathbf{D}} \to \mathrm{Tok}_{\mathbf{D}}$ is defined by the clauses:

$$\begin{aligned}
&\sup(*, a) := \sup(a, *) := a,\\
&\sup(0, 0) := 0,\\
&\sup(0, Cab) := \sup(Cab, 0) := *,\\
&\sup(Cab, Ccd) := C \sup(a, c) \sup(b, d).
\end{aligned}$$

▶ **Lemma 17.** *(i)* $\forall_a(\sup(a, a) = a)$.
(ii) $\forall_a(\forall_b(\sup(a, b) = \sup(b, a)))$.
(iiia) $\forall_a(\forall_b(\mathrm{con}(a, b) \to \sup(a, b) :: \mathrm{nil} \vdash a))$.
(iiib) $\forall_a(\forall_b(\mathrm{con}(a, b) \to \sup(a, b) :: \mathrm{nil} \vdash b))$.
(iv) $\forall_a(\forall_b(\forall_c(\mathrm{con}(a, b) \to \mathrm{con}(b, c) \to \mathrm{con}(a, c) \to \mathrm{con}(\sup(a, b), \sup(b, c)))))$.

The function $\mathrm{Sup} : \mathrm{Tok}_{\mathbf{D}} \to \mathrm{LTok}_{\mathbf{D}} \to \mathrm{Tok}_{\mathbf{D}}$ outputs a token having the maximum "common information" between a token and a list, and it is defined by the clauses:

$$\mathrm{Sup}(a, \mathrm{nil}_{\mathbf{D}}) := a, \text{ and } \mathrm{Sup}(a, b :: U) := \mathrm{sup}(\mathrm{sup}(a, b), \mathrm{Sup}(a, U)).$$

▶ **Lemma 18.** *(i)* $\forall_U (\forall_{a,b}(\mathrm{Con}(a :: b :: U) \to \mathrm{con}(\mathrm{sup}(a, b), \mathrm{Sup}(a, U))))$.
(ii) $\forall_U (\forall_a (\mathrm{Con}(a :: U) \to \mathrm{Sup}(a, U) :: \mathrm{nil} \vDash a :: U))$.

Proof. (i) We apply $\mathrm{LTokElim}_{\mathbf{D}}$ on the obvious $\Phi(U)$. The case $\Phi(\mathrm{nil})$ takes the form

$$\forall_{a,b}(\mathrm{Con}(a :: b :: \mathrm{nil}) \to \mathrm{con}(\mathrm{sup}(a, b), a)).$$

Since it is direct by Lemma 3(iv) that $\mathrm{Con}(a :: b :: \mathrm{nil}) \leftrightarrow \mathrm{con}(a, b)$, by Lemma 17(iiia) we get $\mathrm{sup}(a, b) :: \mathrm{nil} \vdash a$, and consequently, by Lemma 5(x) and the above equivalence, we conclude $\mathrm{con}(\mathrm{sup}(a, b), a)$. Next we suppose that $\Phi(U)$ and we prove $\Phi(c :: U)$ i.e.,

$$\forall_{a,b}(\mathrm{Con}(a :: b :: (c :: U)) \to \mathrm{con}(a', \mathrm{sup}(b', c'))),$$

where

$$(a', b', c') := (\mathrm{sup}(a, b), \mathrm{sup}(a, c), \mathrm{Sup}(a, U)).$$

For that we suppose $\mathrm{Con}(a :: b :: (c :: U))$, where a, b are fixed tokens, and we apply Lemma 17(iv) on a', b', c'. By Lemma 17(ii) $\mathrm{con}(a', b') \leftrightarrow \mathrm{con}(\mathrm{sup}(b, a), \mathrm{sup}(a, c))$, which we get by applying Lemma 17(iv) on tokens b, a, c (the consistencies $\mathrm{con}(b, a), \mathrm{con}(a, c), \mathrm{con}(b, c)$ are implied by $\mathrm{Con}(a :: b :: (c :: U))$ and Lemma 3(iv)). Also, $\mathrm{con}(b', c') \leftrightarrow \mathrm{con}(\mathrm{sup}(a, c), \mathrm{Sup}(a, U))$ is derived by applying the inductive hypothesis $\Phi(U)$ on tokens a, c (clearly, $\mathrm{Con}(a :: b :: (c :: U)) \to \mathrm{Con}(a :: c :: U)$ by Lemma 3(v)). Finally, $\mathrm{con}(a', c') \leftrightarrow \mathrm{con}(\mathrm{sup}(a, b), \mathrm{Sup}(a, U))$ is derived by the application of $\Phi(U)$ on tokens a, b (again, $\mathrm{Con}(a :: b :: (c :: U)) \to \mathrm{Con}(a :: b :: U)$). The conclusion of Lemma 17(iv) on a', b', c' is

$$\mathrm{con}(\mathrm{sup}(a', b'), \mathrm{sup}(b', c')) \leftrightarrow \mathrm{Con}(a'' :: b'' :: \mathrm{nil}),$$

where

$$(a'', b'') := (\mathrm{sup}(a', b'), \mathrm{sup}(b', c')).$$

Since $\mathrm{con}(a', b')$ has already been proved, by Lemma 17(iiia) we get $a'' :: \mathrm{nil} \vdash a'$, which implies, by Lemma 6(v), that $a'' :: b'' :: \mathrm{nil} \vdash a'$. By Lemma 5(x) and the already proved $\mathrm{Con}(a'' :: b'' :: \mathrm{nil})$ we conclude $\mathrm{Con}(a' :: a'' :: b'' :: \mathrm{nil})$. By Lemma 3(iv) we get the required consistency $\mathrm{con}(a', b'')$.
(ii) We apply $\mathrm{LTokElim}_{\mathbf{D}}$ on the obvious $\Phi(U)$. The case $\Phi(\mathrm{nil})$ takes the form $\mathrm{Con}(a :: \mathrm{nil}) \to a :: \mathrm{nil} \vDash a :: \mathrm{nil}$, which is proved by Corollary 8(i). Next we suppose that $\Phi(U)$ and we prove $\Phi(b :: U)$ i.e., fixing a token a we show that

$$\mathrm{Con}(a :: (b :: U)) \to \mathrm{sup}(a', b') :: \mathrm{nil} \vDash a :: (b :: U),$$

where

$$(a', b') := (\mathrm{sup}(a, b), \mathrm{Sup}(a, U)).$$

By (i) we know that $\mathrm{con}(a', b')$. Hence by Lemma 17(iiia) we get $\mathrm{sup}(a', b') :: \mathrm{nil} \vdash a'$, or equivalently, $\mathrm{sup}(a', b') :: \mathrm{nil} \vDash a' :: \mathrm{nil}$. Since $\mathrm{Con}(a :: (b :: U)) \to \mathrm{con}(a, b)$, by Lemma 17(iiia) and (iiib) we have

$$a' :: \mathrm{nil} \vdash a \ \wedge \ a' :: \mathrm{nil} \vdash b.$$

Thus, by Lemma 6(v) we conclude that $\mathrm{sup}(a', b') :: \mathrm{nil} \vdash a$ and $\mathrm{sup}(a', b') :: \mathrm{nil} \vdash b$. By Lemma 17(iiib) we also get that $\mathrm{sup}(a', b') :: \mathrm{nil} \vdash b'$, which is equivalent to $\mathrm{sup}(a', b') :: \mathrm{nil} \vDash$

b' :: nil. Since by the inductive hypothesis $\Phi(U)$ we have $b' :: \text{nil} \vDash U$, we get, by Corollary 8(ii), that $\sup(a', b') :: \text{nil} \vDash U$ too.

◀

▶ **Theorem 19** (point-free density in $\mathbf{D}$). *(i)* $\forall_U(\text{Con}(U) \to \exists_a(\text{total}(a) \wedge_{\mathbf{B}} a :: \text{nil} \vDash U))$.
(ii) $\forall_U(\text{Con}(U) \to \exists_a(\text{total}(a) \wedge_{\mathbf{B}} U \dot{\subseteq} \overline{a :: \text{nil}}))$.

Proof. (i) We apply apply LTokElim$_{\mathbf{D}}$ on the obvious $\Phi(U)$. For the case $\Phi(\text{nil})$ it suffices to take $a = 0$. Next we suppose that $\Phi(U)$ and we prove $\Phi(b :: U)$ i.e.,

$$\text{Con}(b :: U) \to \exists_a(\text{total}(a) \wedge_{\mathbf{B}} a :: \text{nil} \vDash b :: U).$$

By Lemma 18(ii) $\text{Sup}(b, U) :: \text{nil} \vDash b :: U$, therefore we define

$$a := \text{t}(\text{Sup}(b, U)).$$

By Lemma 16(i) we have that total(a), while by Lemma 16(ii) we get that $a :: \text{nil} \vdash \text{Sup}(b, U)$, which is equivalent to $a :: \text{nil} \vDash \text{Sup}(b, U) :: \text{nil}$. Hence by Corollary 8(ii) we conclude that $a :: \text{nil} \vDash b :: U$.
(ii) Consider a consistent list U. By (i) there exists a token a such that $\text{t}(a)$ and $a :: \text{nil} \vDash U$. By Lemma 15(i) though, $a :: \text{nil} \vDash U \to U \dot{\subseteq} \overline{a :: \text{nil}}$.

◀

If we define[18] a decidable ideal in $\mathbf{D}$ to be total by

$$\text{Total}(I) := \exists_a(\text{total}(a) \wedge a \dot{\in} I \wedge \forall_b(b \dot{\in} I \to a :: \text{nil} \vdash b)),$$

then $\text{total}(a) \to \text{Total}(\overline{a :: \text{nil}})$, and Theorem 19(ii) fully recovers the content of density theorem in $\mathbf{D}$.

6 Conclusion and future work

We have presented here some advances in the constructive formal theory of computable functions TCF$^+$ that have taken place after its first sketch in [4]. All definitions and proofs are due to the corresponding implementation procedure, which can be found in [10].

The project of "mirroring" the non-flat Scott model into a formal theory like TCF$^+$ and the elaboration of important case studies[19] within it is still at its beginning. The current development of TCF$^+$ though, promises that this project is far from a mechanical procedure without mathematical benefits. The implementation enterprise revealed, apart from unexpected analogies between the basic notions of $\mathbf{SC}_{\mathbf{D}}$ and $\mathbf{SC}_{\mathbf{D}\to\mathbf{D}}$, that these syntactic systems satisfy an even stronger form of the axioms of an information system, a fact that was extended to the Scott model itself.

Even more surprisingly, the implementation showed that in the case of the density theorem a point-free approach is possible. In the future we want to prove a formal point-free density theorem in $\mathbf{D} \to \mathbf{D}$. Then we will have all the necessary tools for a formal proof of a general

[18] In subsequent work we explain how this definition conforms to Normann's [8] notion of totality as complete information by connecting totality of an ideal with normalization with respect to a certain reduction relation. We also note here that the total token a which entails all the other tokens of I is proved to be unique, and moreover it is the unique total token in I.

[19] Such case studies are Plotkin's definability theorem, the totality of the fan functional, or Normann's solution to the Cook-Berger conjecture.

point-free density theorem (e.g., in $\mathbf{D}_{n+1} = \mathbf{D}_n \to \mathbf{D}$). Such a generalization seems possible because we can define a decidable ideal R in $\mathbf{D} \to \mathbf{D}$ in a way similar to that of a decidable ideal I in $\mathbf{D}$. A decidable ideal R can be represented as a decidable approximable mapping, i.e., an appropriate boolean-valued function on approximation terms which behaves as an approximable mapping at a "local" and a "global" level, exactly like I. In that way the motivation behind the introduction of an approximable mapping, as the point-free version of a function between domains, is realized.

Acknowledgements I want to thank Helmut Schwichtenberg for his ongoing motivation and support.

References

1 U. Berger: Total sets and objects in domain theory, Annals of Pure and Applied Logic, 60:91-117, 1993.

2 T. Coquand, G. Sambin, J. Smith and S. Valentini: Inductively generated formal topologies, Annals of Pure and Applied Logic, 124, pp. 71-106, 2003.

3 A. Heyting (editor): *Constructivity in Mathematics*, 1959. North-Holland, Amsterdam.

4 S. Huber, B. Karadais and H. Schwichtenberg: Towards a formal theory of computability, in R. Schindler (ed.) *Ways of Proof Theory: Festschrift for W. Pohlers*, 251-276, Ontos Verlag, 2010.

5 B. Karadais: *Towards an Arithmetic for Partial Computable Functionals*, PhD Thesis, LMU, 2013.

6 G. Kreisel: Interpretation of analysis by means of constructive functionals of finite types, in [3], pp. 101-128.

7 The Minlog system. `http://minlog-system.de/`

8 D. Normann: Formalizing the notion of total information, in Petkov (ed.) *Mathematical Logic, Proceedings of the 1988 Heyting Conference*, Plenum Press, New York, 1990, pp. 67-94.

9 D. Normann: Applications of the Kleene-Kreisel Density Theorem to Theoretical Computer Science, e-print, 2006.

10 `www.mathematik.uni-muenchen.de/~petrakis/tcf+.scm`

11 R. A. Platek: *Foundations of recursion theory*, PhD Thesis, Stanford University, 1966.

12 G. D. Plotkin. LCF considered as a programming language, Theoretical Computer Science, 5:223-255, 1977.

13 H. Schwichtenberg: Recursion on the partial continuous functionals, in C. Dimitracopoulos, L. Newelski, D. Normann, and J. Steel, editors, Logic Colloquium '05, volume 28 of Lecture Notes in Logic, pages 173-201, Association for Symbolic Logic, 2006.

14 H. Schwichtenberg and S. Wainer: *Proofs and Computations*, Perspectives in Logic. Assoc. Symb. Logic and Cambridge University Press, 2012.

15 D. Scott: A type-theoretic alternative to ISWIM, CHUCH, OWHY, Manuscript, Oxford University, 1969; later published in Theoretical Computer Science, 121:411-440, 1993.

16 D. Scott: Domains for denotational semantics, in E. Nielsen and E. Schmidt, editors, Automata, Languages and Programming, volume 140 of LNCS, pages 577-613. Springer Verlag, Berlin, Heidelberg, New York, 1982.

17 V. Stoltenberg-Hansen, I. Lindström and E. R. Griffor: *Mathematical theory of domains*, Cambridge Tracts in Theoretical Computer Science, Vol.22, Cambridge University Press, 1994.

A Appendix

A.1 The non-flat Scott model

The definition of the non-flat Scott model is actually by recursion on the *height* $||.||$ of the set-theoretical syntactic expressions E involved, where $||.|| : E \to \mathbf{N}$ defined by the following clauses:

$$\begin{aligned}
&|| *_\iota || := 0,\\
&||Ca_1^*...a_n^*|| := 1 + \max\{||a_i^*|| : 1 \leq i \leq n\}\\
&||(U,b)|| := 1 + \max\{||U||, ||b||\},\\
&||U \vdash_\rho b|| := 1 + \max\{||U||, ||b||\},\\
&||\{a_i : i \in I\}|| := 1 + \max\{||a_i|| : i \in I\},
\end{aligned}$$

where I is a finite set, and $C^{\tau_1 \to \dots \to \tau_n \to \iota}$ is a constructor.

In [14], p.262, the proof of $\mathbf{C}_\rho$ being an information system is not given, since it is considered easy. Actually the proof for the ground types is easy, but the proof for the arrow types is less trivial. Since the definition of $\mathbf{C}_\rho$ is given by recursion on the height of the syntactic expressions involved, the proof must also be given w.r.t. this height.

Theorem 1. The structure $\mathbf{C}_\rho = (C_\rho, \mathrm{Con}_\rho, \vdash_\rho)$ is a coherent i.s., for each $\rho \in \mathrm{Typ}$.

Proof. First we show that $\mathbf{C}_\rho$ satisfies the axioms of an information system, i.e.,

1. $V \in \mathrm{Con}_\rho \to U \subseteq V \to U \in \mathrm{Con}_\rho$.
2. $a \in C_\rho \to \{a\} \in \mathrm{Con}_\rho$.
3. $U \in \mathrm{Con}_\rho \to U \vdash_\rho a \to U \cup \{a\} \in \mathrm{Con}_\rho$.
4. $a \in U \to U \in \mathrm{Con}_\rho \to U \vdash_\rho a$.
5. $U \in \mathrm{Con}_\rho \to V \in \mathrm{Con}_\rho \to U \vDash_\rho V \to V \vdash_\rho a \to U \vdash_\rho a$.

(1) If $\rho = \iota$, and $V = \{Ca^*_{(1)1}...a^*_{(1)n}, \dots, Ca^*_{(k)1}...a^*_{(k)n}\}$, for some $k \in \mathbb{N}$, then $U \subseteq V \to U_l \subseteq V_l$, for each $1 \leq l \leq n$. Since, by the definition of Con_ι, $U_l \in \mathrm{Con}_{\tau_l}$ and $||V_l|| < ||V||$, we get that $U_l \in \mathrm{Con}_{\tau_l}$, for each $1 \leq l \leq n$. Hence, by the definition of Con_ι again, we get that $U \in \mathrm{Con}_\iota$. If $W = \{(U_i, b_i) : i \in I\} \in \mathrm{Con}_{\rho\to\sigma}$, $W' = \{(U_{i'}, b_{i'}) : i' \in I'\} \subseteq W$, and $\bigcup_{j' \in J' \subseteq I' \subseteq I} U_{j'} \in \mathrm{Con}_\rho$, then automatically we get that $\{b_{j'} : j' \in J'\} \in \mathrm{Con}_\sigma$. Note that this argument is independent from the height of W.

(2) If $\rho = \iota$, then $U = \{Ca_1^*...a_n^*\} \in \mathrm{Con}_\iota \leftrightarrow \forall_{1\leq l \leq n}(U_l = \{a_l^*\} \in \mathrm{Con}_{\tau_l})$. By the definition of U_l the token a_l^* is proper and $||U_l|| < ||U||$ therefore, $U_l \in \mathrm{Con}_{\tau_l}$, for each l. In the arrow type case we need to show that $\{(U,b)\} \in \mathrm{Con}_{\rho\to\sigma} \leftrightarrow (U \in \mathrm{Con}_\rho \to \{b\} \in \mathrm{Con}_\sigma)$. But since $||b|| < ||(U,b)||$ and $b \in \mathrm{Con}_\sigma$, the inductive hypothesis guarantees that $\{b\} \in \mathrm{Con}_\sigma$.

(3) What we actually prove is simultaneously (3) with the following weaker, because of (4), form of (5)

$$(6)\ \ U \in \mathrm{Con}_\rho \to V \subseteq U \to V \vdash_\rho a \to U \vdash_\rho a.$$

If $\rho = \iota$, $U = \{Ca^*_{(1)1}...a^*_{(1)n}, \dots, Ca^*_{(k)1}...a^*_{(k)n}\}$, for some $k \in \mathbb{N}$, and $a = Ca_1^*...a_n^*$, then, since $||U_l|| < ||U||$,

$$U \vdash_\iota a \to \forall_l(U_l \vdash_{\tau_l} a_l^*) \to \forall_l(U_l \cup \{a_l^*\} = (U \cup \{a\})_l \in \mathrm{Con}_{\tau_l}) \to U \cup \{a\} \in \mathrm{Con}_\iota.$$

Also for (6) we know that $V \vdash_\iota Ca_1^*...a_n^* \to V_l \vdash_{\tau_l} a_l^*$, for each l. But since $V_l \subseteq U_l$ and $||U_l|| < ||U||$, we get $U_l \vdash_{\tau_l} a_l^*$, for each l. In the arrow type case, if $W = \{(U_i, b_i) : i \in I\}$ and $WV = \{b_i : V \vDash_\rho U_i\} = \{b_i : i \in I_0 \subseteq I\}$, we show that

$$W \in \mathrm{Con}_{\rho\to\sigma} \to W \vdash_{\rho\to\sigma} (V,b) \to W \cup \{(V,b)\} \in \mathrm{Con}_{\rho\to\sigma},$$

and simultaneously we prove (6) i.e.,

$$W \in \mathrm{Con}_{\rho\to\sigma} \to W' \subseteq W \to W' \vdash_{\rho\to\sigma} (V,b) \to W \vdash_{\rho\to\sigma} (V,b).$$

Our simultaneous inductive hypotheses (IH1 and IH2, respectively) are that (3) and (6) hold for all formal neighborhoods U and proper tokens a such that $||U|| + ||a|| < ||W|| + ||(V,b)||$. First we show that IH1 guarantees that $WV \in \mathrm{Con}_\sigma$. Since $V \in \mathrm{Con}_\rho$, if $V \vdash_\rho U_i$, for some i, then $V \vdash_\rho a$, for each $a \in U_i$. Since $||V|| + ||a|| < ||(V,b)|| + ||W||$ we get that $V \cup \{a\} \in \mathrm{Con}_\rho$. Since that happens for arbitrary a and i we conclude that

$$V \cup \bigcup_{i\in I_0} U_i \in \mathrm{Con}_\rho \overset{(1)}{\to} \bigcup_{i\in I_0} U_i \in \mathrm{Con}_\rho \overset{(W\in\mathrm{Con}_{\rho\to\sigma})}{\longrightarrow} WV \in \mathrm{Con}_\sigma.$$

Since we know now that $WV \in \mathrm{Con}_\sigma$, and $WV \vdash_\sigma b$, we can apply again IH1 to conclude that

$$\{b_i : V \vDash_\rho U_i\} \cup \{b\} \in \mathrm{Con}_\sigma,$$

by the fact that $||WV|| < ||W||$ and $||V|| < ||(V,b)||$. By the definition of $\mathrm{Con}_{\rho\to\sigma}$ the consistency of $W \cup \{(U,b)\}$ amounts to

$$\forall_{J\subseteq I}(\bigcup_{j\in J} U_j \cup V \in \mathrm{Con}_\rho \to \{b_j : j \in J\} \cup \{b\} \in \mathrm{Con}_\sigma).$$

If we consider such a $J \subseteq I$, then, by our hypothesis, $\bigcup_{j\in J} U_j \cup V \in \mathrm{Con}_\rho$. Suppose that $a \in U_i$, where $i \in I_0$. Clearly, $V \vDash_\rho U_i \to V \vdash_\rho a$. Since $||U_i|| < ||W||$ and $||V|| < ||(V,b)||$, we get that

$$||\bigcup_{j\in J} U_j \cup V|| + ||a|| = 1 + \max\{||c|| : c \in \bigcup_{j\in J} U_j \cup V\} + ||a|| < ||W|| + ||((V,b)||.$$

Applying IH2 to $\bigcup_{j\in J} U_j \cup V$ and a we get that

$$\bigcup_{j\in J} U_j \cup V \vdash_\rho a.$$

Since that is established, we can apply now IH1 to $\bigcup_{j\in J} U_j \cup V$ and a in order to get

$$\bigcup_{j\in J} U_j \cup V \cup \{a\} \in \mathrm{Con}_\rho.$$

Since that is proved for arbitrary $a \in U_i$ and $i \in I_0$, we get that

$$\bigcup_{j\in J} U_j \cup V \cup \bigcup_{i\in I_0} U_i \in \mathrm{Con}_\rho.$$

By (1) we get that

$$\bigcup_{j\in J} U_j \cup \bigcup_{i\in I_0} U_i \in \mathrm{Con}_\rho$$

therefore, by the definition of consistency for W we have that

$$\{b_j : j \in J\} \cup \{b_i : i \in I_0\} \in \mathrm{Con}_\sigma.$$

Since $||\{b_j : j \in J\} \cup \{b_i : i \in I_0\}|| + ||b|| < ||W|| + ||(V,b)||$, and $\{b_i : i \in I_0\} \vdash_\sigma b$ we get by IH2 on $\{b_j : j \in J\} \cup \{b_i : i \in I_0\}$ and b that

$$\{b_j : j \in J\} \cup \{b_i : i \in I_0\} \vdash_\sigma b.$$

Because of the last fact though, we can apply IH1 to them again and take

$$\{b_j : j \in J\} \cup \{b_i : i \in I_0\} \cup \{b\} \in \mathrm{Con}_\sigma \overset{(1)}{\to} \{b_j : j \in J\} \cup \{b\} \in \mathrm{Con}_\sigma,$$

which is what required for the proof of (3). Next we prove (6) for the arrow type case. By the definition of $\vdash_{\rho\to\sigma}$ it suffices to prove $W'V \vdash_\sigma b \to WV \vdash_\sigma b$. Since $WV \in \mathrm{Con}_\sigma$ is already proved and $||WV|| + ||b|| < ||W|| + ||(V, b)||$, and since trivially (independently from the height of W) we have that

$$W' \subseteq W \to W'V \subseteq WV,$$

applying IH2 to WV and b we get the required $WV \vdash_\sigma b$.
(4) If $\rho = \iota$ and $a = Ca^*_{(i)1} \dots a^*_{(i)n} \in U$, where $1 \le i \le k$, then $a^*_{(i)l} \in U_l$, for each $1 \le l \le n$. Since $||U_l|| < ||U||$ and $U_l \in \mathrm{Con}_{\tau_l}$, we get that $U_l \vdash_{\tau_l} a^*_{(i)l}$, for each l, i.e., $U \vdash_\iota a$. In the arrow type case we need to show that

$$(U_i, b_i) \in W \to W \in \mathrm{Con}_{\rho\to\sigma} \to W \vdash_{\rho\to\sigma} (U_i, b_i).$$

Thus, we need to show that $WU_i = \{b_j : U_i \vDash_\rho U_j\} \vdash_\sigma b_i$. We do it by induction on the measure $||W|| + ||(U_i, b_i)||$. If $a \in U_i$, then $U_i \vdash_\rho a$, because $||U_i|| + ||a|| < ||(U_i, b_i)|| + ||W||$. Since this holds for arbitrary $a \in U_i$, we conclude that $U_i \vDash_\rho U_i$ therefore, $b_i \in WU_i$. Since $||WU_i|| + ||b|| < ||W|| + ||(U_i, b_i)||$, we have that $WU_i \vdash_\sigma b_i$ applying our inductive hypothesis. The only thing we need to justify is that $WU_i \in \mathrm{Con}_\sigma$[20]. But we can use the already proved (3) so that $U_i \vdash_\rho a \to U_i \cup \{a\} \in \mathrm{Con}_\rho$ therefore,

$$U_i \cup \bigcup_{U_i \vdash_\rho U_j} U_j \in \mathrm{Con}_\rho \overset{(1)}{\to} \bigcup_{U_i \vdash_\rho U_j} U_j \in \mathrm{Con}_\rho \overset{(W \in \mathrm{Con}_{\rho\to\sigma})}{\longrightarrow} WU_i \in \mathrm{Con}_\sigma.$$

(5) If $\rho = \iota$, $U = \{Ca^*_{(1)1}...a^*_{(1)n}, \dots, Ca^*_{(k)1}...a^*_{(k)n}\}$, for some $k \in \mathbb{N}$,

$$V = \{Cb^*_{(1)1}...b^*_{(1)n}, \dots, Cb^*_{(m)1}...a^*_{(m)n}\},$$

for some $m \in \mathbb{N}$, and $a = Ca^*_1...a^*_n$, then

$$U \vDash_\iota V \to \forall_{1\le l\le n} \forall_{1\le\nu\le m} (U_l \vdash_{\tau_l} b^*_{(\nu)l}).$$

Thus $U_l \vDash_{\tau_l} V_l$, for each l. Also, w.r.t. the measure $||U|| + ||V||$ we have

$$V \vdash_\iota a \to \forall_l (V_l \vdash_{\tau_l} a^*_l) \overset{||V_l||+||U_l||<||U||+||V||}{\longrightarrow} \forall_l (U_l \vdash_{\tau_l} a^*_l) \to U \vdash_\iota a.$$

In the arrow type case we need to show that

$$W, W' \in \mathrm{Con}_{\rho\to\sigma} \to W \vDash_{\rho\to\sigma} W' \to W' \vdash_{\rho\to\sigma} (V, b) \to W \vdash_{\rho\to\sigma} (V, b).$$

We use induction on the measure $||W|| + ||W'|| + ||(V, b)||$. If $W = \{(U_i, b_i) : i \in I\}$ and $W' = \{(V_j, c_j) : j \in J\}$, then $WV, W'V, WV_j \in \mathrm{Con}_\sigma$, for each $j \in J$; for that we work exactly like in the corresponding proof in (4). By our hypotheses we have

$$\forall_{j\in J}(W \vdash_{\rho\to\sigma} (V_j, c_j)) \leftrightarrow \forall_{j\in J}(WV_j = \{b_i : V_j \vDash_\rho U_i\} \vdash_\sigma c_j),$$

[20] Note that we cannot use the result $WV \in \mathrm{Con}_\sigma$ in the proof of (3), because there we used the hypothesis $W \vdash_{\rho\to\sigma} (V, b)$, which is exactly what we want to prove for (U_i, b_i).

$$W' \vdash_{\rho\to\sigma} (V,b)) \leftrightarrow W'V = \{c_j : V \vDash_\rho V_j\} \vdash_\sigma b,$$

and we want to show that

$$W \vdash_{\rho\to\sigma} (V,b)) \leftrightarrow WV = \{b_i : V \vDash_\rho U_i\} \vdash_\sigma b.$$

Since $||WV|| + ||W'V|| + ||b|| < ||W|| + ||W'|| + ||(V,b)||$ we reach our conclusion applying our inductive hypothesis on

$$WV \in \mathrm{Con}_\sigma \to W'V \in \mathrm{Con}_\sigma \to WV \vDash_\sigma W'V \to W'V \vdash_\sigma b \to WV \vdash_\sigma b.$$

The only thing we need to establish in order to complete our proof is the implication

$$(*) \quad W \vdash_{\rho\to\sigma} W' \to WV \vDash_\sigma W'V.$$

If $c_j \in W'V$ therefore, $V \vDash_\rho V_j$, we show that $WV \vdash_\sigma c_j$. If $a \in U_i$ and $b_i \in WV_j$ i.e., $V_j \vDash_\rho U_i \to V_j \vdash_\rho a$ we get

$$V \vDash_\rho V_j \to V_j \vdash_\rho a \to V \vdash_\rho a,$$

since $||V|| + ||V_j|| + ||a|| < ||(V,b)|| + ||W'|| + ||W||$. Since that holds for each $a \in U_i$, we conclude that $V \vDash_\rho U_i$ i.e., we showed that $b_i \in WV$ therefore, we get $WV_j \subseteq WV$. If we apply (6) on

$$WV_j \subseteq WV \to WV_j \vdash_\sigma c_j \to WV \vdash_\sigma c_j,$$

we prove $(*)$, therefore we complete the proof of (5).

To prove the coherence property of the information systems $\mathbf{C}_\rho$ it suffices to show it for the ground information systems $\mathbf{C}_\iota$ and then use the standard fact that if $\mathcal{B}$ is a coherent i.s., then $\mathcal{A} \to \mathcal{B}$ is also coherent (see [14], p.264). For an information system $\mathbf{C}_\iota$ we prove its coherence by induction on the height $||U||$ of a ground formal neighborhood U. If $k \in \mathbb{N}$, $C^{\tau_1\to\ldots\to\tau_n\to\iota}$ is a constructor, and $U = \{Ca^*_{(1)1}...a^*_{(1)n}, \ldots, Ca^*_{(k)1}...a^*_{(k)n}\}$, then

$$\begin{aligned}
U \in \mathrm{Con}_\iota &\leftrightarrow \forall_{1\le l\le n}(U_l = \{a^*_{(i)l} : a^*_{(i)l} \text{ is proper in } C_{\tau_l} \wedge 1 \le i \le k\} \in \mathrm{Con}_{\tau_l}) \\
&\overset{||U_l||<||U||}{\leftrightarrow} \forall_{1\le i,j\le k}(\{a^*_{(i)1}, a^*_{(j)1}\} \in \mathrm{Con}_{\tau_1} \wedge \ldots \wedge \{a^*_{(i)n}, a^*_{(j)n}\} \in \mathrm{Con}_{\tau_n}) \\
&\leftrightarrow \forall_{1\le i,j\le k}(\{Ca^*_{(i)1}...a^*_{(i)n}, Ca^*_{(j)1}...a^*_{(j)n}\} \in \mathrm{Con}_\iota).
\end{aligned}$$

◀

Because of the coherence property of $\mathbf{C}\rho$ we can define $\mathrm{Con}_\iota, \mathrm{Con}_{\rho\to\sigma}$ as follows:

1. A *base type formal neighborhood*, $U \in \mathrm{Con}_\iota$, is a finite set of tokens in C_ι starting with the same constructor $C^{\tau_1\to\ldots\to\tau_n\to\iota}$ i.e., $U = \{Ca^*_{(1)1}...a^*_{(1)n}, \ldots, Ca^*_{(k)1}...a^*_{(k)n}\}$, for some $k \in \mathbb{N}$, such that, for each $1 \le l \le n$ and for each $1 \le i, i' \le k$,

$$U_l^{i,i'} = \{a^*_{(i)l}, a^*_{(i')l} : a^*_{(i)l}, a^*_{(i')l} \text{ proper tokens in } C_{\tau_l}\} \in \mathrm{Con}_{\tau_l}.$$

2. An *arrow type formal neighborhood*, $W \in \mathrm{Con}_{\rho\to\sigma}$, is a finite set of tokens in $C_{\rho\to\sigma}$, i.e., $W = \{(U_i, b_i) : i \in I\}$, for some finite set I, such that

$$\forall_{i,i'\in I}(U_i \cup U_{i'} \in \mathrm{Con}_\rho \to \{b_i, b_{i'}\} \in \mathrm{Con}_\sigma).$$

It is this definition of consistency that we use for the syntactic information systems $\mathbf{SC}_\rho$.

Theorem 14. The information systems $\mathbf{C}_\rho$ satisfy the axioms $4', 5'$, for each type ρ.

Proof. By inspection of the proof of Theorem 1 we see that the axioms $4', 5'$ follow in an even simpler way. One only needs to extend appropriately the definition of entailment on finite sets of tokens which are not necessarily consistent. ◀

A.2 The syntactic information system $\mathrm{SC}_{\mathbf{D}}$

Lemma 2. $\forall_U \forall_V (\forall_a (a \dot{\in}_{\mathbf{D}} U \to a \dot{\in}_{\mathbf{D}} V) \leftrightarrow U \dot{\subseteq}_{\mathbf{D}} V)$.

Proof. The two directions are proved applying LTokElim on

$$\varPhi_1(U) := \forall_V (\forall_a (a \dot{\in}_{\mathbf{D}} U \to a \dot{\in}_{\mathbf{D}} V) \to U \dot{\subseteq}_{\mathbf{D}} V),$$

$$\varPhi_2(U) := \forall_V (U \dot{\subseteq}_{\mathbf{D}} V \to \forall_a (a \dot{\in}_{\mathbf{D}} U \to a \dot{\in}_{\mathbf{D}} V)).$$

◀

Lemma 3. (i) $\forall_U (\forall_a (\mathrm{Altcon}(a, U) \leftrightarrow \forall_b (b \dot{\in} U \to \mathrm{con}(a, b))))$.
(ii) $\forall_a (\mathrm{con}(a, a))$.
(iii) $\forall_a (\mathrm{Con}(a :: \mathrm{nil}))$.
(iv) $\forall_U (\forall_{a_1, a_2} (a_1 \dot{\in} U \to a_2 \dot{\in} U \to \mathrm{con}(a_1, a_2)) \leftrightarrow \mathrm{Con}(U))$.
(v) $\forall_{U,V} (\mathrm{Con}(U) \to V \dot{\subseteq} U \to \mathrm{Con}(V))$.

Proof. (i) We prove the direction ($\to$) and we work similarly for the converse. We apply LTokElim on

$$\varPhi(U) := \forall_a (\mathrm{Altcon}(a, U) \to \forall_b (b \dot{\in} U \to \mathrm{con}(a, b))).$$

Using Efq we prove $\varPhi(\mathrm{nil})$. Suppose that $\mathrm{Altcon}(a, c :: U)$ i.e., $\mathrm{con}(a, c) \wedge \mathrm{Altcon}(a, U)$, and also that $b \dot{\in} (c :: U)$ i.e., $b = c \vee b \dot{\in} U$. If $b = c$, then we get $\mathrm{con}(a, b)$ by hypothesis. If $b \dot{\in} U$ we use the hypothesis $\mathrm{Altcon}(a, U)$ and the inductive hypothesis $\varPhi(U)$.
(ii) We apply TokElim on $\varphi(a) := \mathrm{con}(a, a)$.
(iii) We apply TokElim on $\varphi(a) := \mathrm{Con}(a :: \mathrm{nil})$.
(iv) We prove the direction ($\leftarrow$) and we work similarly for the converse. We apply LTokElim on

$$\varPhi(U) := \mathrm{Con}(U) \to \forall_{a_1, a_2} (a_1 \dot{\in} U \to a_2 \dot{\in} U \to \mathrm{con}(a_1, a_2)).$$

By Efq we get directly $\varPhi(\mathrm{nil})$. Next we suppose that $\varPhi(U)$ and we show $\varPhi(b :: U)$, for some b. Supposing $\mathrm{Con}(b :: U)$ we show that $\forall_{a_1, a_2} (a_1 \dot{\in} (b :: U) \to a_2 \dot{\in} (b :: U) \to \mathrm{con}(a_1, a_2))$. In case $a_1 = b = a_2$ we use (ii). In case $a_1 = b$ and $a_2 \dot{\in} U$ we use the direction ($\to$) of (i), since the hypothesis $\mathrm{Con}(b :: U)$ entails $\mathrm{Altcon}(b, U)$. In case $a_2 = b$ and $a_1 \dot{\in} U$ we work similarly. In case $a_1, a_2 \dot{\in} U$ we use the inductive hypothesis $\varPhi(U)$, since $\mathrm{Con}(b :: U)$ entails $\mathrm{Con}(U)$.
(v) By (iv) it suffices to show that

$$\forall_{a_1, a_2} (a_1 \dot{\in} V \to a_2 \dot{\in} V \to \mathrm{con}(a_1, a_2)).$$

By Lemma 1 the hypothesis $V \dot{\subseteq} U$ implies that $a_1, a_2 \dot{\in} U$, if $a_1, a_2 \dot{\in} V$. By the ($\leftarrow$) direction of (iv) for U we get that $\mathrm{con}(a_1, a_2)$.

◀

Lemma 4. (i) $\forall_a (\forall_U (\mathrm{Comp}(U) \to \mathrm{Comp}(a :: U)))$.
(ii) $\forall_U (\forall_{a_1, a_2} (C a_1 a_2 \dot{\in} U \to \mathrm{Comp}(U)))$.
(iii) $\forall_b (\forall_{a,U} (a \dot{\in} \arg_i(U) \to a \dot{\in} \arg_i(b :: U)))$, for each $i = 1, 2$.
(iv) $\forall_U (\forall_{a_1, a_2} (C a_1 a_2 \dot{\in} U \to a_i \dot{\in} \arg_i(U)))$, for each $i = 1, 2$.
(v) $\forall_a (\forall_U (a \dot{\in} U \to U \vdash a))$.

Proof. All cases are proved through the use of the appropriate elimination axiom. Case (i) is used in the inductive step of (ii) and case (iii) in the inductive step of (iv). We present only the proof of (v). We apply TokElim on

$$\varphi(a) := \forall_U (a \dot{\in} U \to U \vdash a).$$

The cases $\varphi(*), \varphi(0)$ are derived automatically by the definition of entailment. To show $\varphi(Ca_1a_2)$ from $\varphi(a_1), \varphi(a_2)$ we suppose that $Ca_1a_2 \dot{\in} U$, for a fixed U. By (ii) we get $\mathrm{Comp}(U)$. By (iv) we get that $a_i \dot{\in} \mathrm{arg}_i(U)$, for each $i = 1, 2$. Applying the inductive hypotheses $\varphi(a_1), \varphi(a_2)$ on $\mathrm{arg}_1(U)$ and $\mathrm{arg}_2(U)$, respectively, we get that $\mathrm{arg}_1(U) \vdash a_1$ and $\mathrm{arg}_2(U) \vdash a_2$.

◀

Lemma 5. (i) $\forall_U(\forall_a(\mathrm{Con}(U) \to a \dot{\in} U \to U \vdash 0 \to \mathrm{con}(0, a)))$.
(ii) $\forall_U(\forall_a(a \dot{\in} \mathrm{arg}_1(U) \to \exists_b(Cab \dot{\in} U)))$, and $\forall_U(\forall_a(a \dot{\in} \mathrm{arg}_2(U) \to \exists_b(Cba \dot{\in} U)))$.
(iii) $\forall_U(\mathrm{Con}(U) \to \mathrm{Con}(\mathrm{arg}_i(U)))$, for each $i = 1, 2$.
(iv) $\forall_a(\forall_U(\mathrm{Comp}(a :: U) \to \mathrm{comp}(a) \vee_{\mathbf{B}} \mathrm{Comp}(U)))$.
(v) $\forall_U(\mathrm{Comp}(U) \to \exists_a(a \dot{\in} U \wedge_{\mathbf{B}} \mathrm{comp}(a)))$.
(vi) $\forall_b(\forall_{U,a_1,a_2}(\mathrm{Con}(U) \to b \dot{\in} U \to U \vdash Ca_1a_2 \to b = * \vee_{\mathbf{B}} \mathrm{comp}(b)))$.
(vii) $\forall_a(\forall_{U,b_1,b_2}(\mathrm{Con}(U) \to a \dot{\in} U \to U \vdash Cb_1b_2 \to \mathrm{con}(Cb_1b_2, a)))$.
(viii) $\forall_a(\forall_{U,b}(\mathrm{Con}(U) \to b \dot{\in} U \to U \vdash a \to \mathrm{con}(a, b)))$.
(ix) $\forall_{U,a}(\mathrm{Con}(U) \to U \vdash a \to \mathrm{Altcon}(a, U))$.
(x) $\forall_{U,a}(\mathrm{Con}(U) \to U \vdash a \to \mathrm{Con}(a :: U))$.

Proof. (i) Since $U \vdash 0$ means that $0 \dot{\in} U$, then $a, 0 \dot{\in} U$. By Lemma 3(iv) we get $\mathrm{con}(0, a)$.
(ii) We apply LTokElim on

$$\Phi_1(U) := \forall_a(a \dot{\in} \mathrm{arg}_1(U) \to \exists_b(Cab \dot{\in} U)).$$

The case $\Phi_1(\mathrm{nil})$ is proved through Efq. For the inductive step we apply TokElim on

$$\varphi_1(c) := a \dot{\in} \mathrm{arg}_1(c :: U) \to \exists_b(Cab \dot{\in} (c :: U)),$$

for some fixed a, U. The cases $\varphi_1(*)$ and $\varphi_1(0)$ are trivial. If $c = Cd_1d_2$, for some d_1, d_2, then $\mathrm{arg}_1(c :: U) = d_1 :: \mathrm{arg}_1(U)$. If $a = d_1$, then $b = d_2$. If $a \dot{\in} \mathrm{arg}_1(U)$, then we use the inductive hypothesis $\Phi_1(U)$ on a.
(iii) We apply LTokElim on

$$\Phi_i(U) := \mathrm{Con}(U) \to \mathrm{Con}(\mathrm{arg}_i(U)).$$

$\Phi_i(\mathrm{nil})$ is proved automatically by the definitions. For the inductive step we apply TokElim on

$$\varphi_i(b) := \mathrm{Con}(b :: U) \to \mathrm{Con}(\mathrm{arg}_i(b :: U)).$$

The cases $\varphi_i(*)$ and $\varphi_i(0)$ are trivial. If $b = Cd_1d_2$, for some d_1, d_2, then $\mathrm{arg}_i(b :: U) = d_i :: \mathrm{arg}_i(U)$ and we need to show $\mathrm{Con}(d_i :: \mathrm{arg}_i(U))$, or, equivalently, $\mathrm{Altcon}(d_i, \mathrm{arg}_i(U))$ and $\mathrm{Con}(\mathrm{arg}_i(U))$. Since $\mathrm{Con}(b :: U) \to \mathrm{Con}(U)$, we get $\mathrm{Con}(\mathrm{arg}_i(U))$ by the inductive hypothesis $\Phi(U)$. To show $\mathrm{Altcon}(d_i, \mathrm{arg}_i(U))$ it suffices, by Lemma 3(i), that $a \dot{\in} \mathrm{arg}_i(U) \to \mathrm{con}(d_i, a)$, for each a. By (ii) there exist c_1, c_2 such that $Cc_1c_2 \dot{\in} U$ and $c_1 = a$, if $i = 1$, or $c_2 = a$, if $i = 2$. Since $\mathrm{Con}(b :: U) \to \mathrm{Altcon}(b, U)$ and $\mathrm{Altcon}(b, U) \to \mathrm{con}(Cd_1d_2, Cc_1c_2)$, by Lemma 3(i), we get $\mathrm{con}(d_i, c_i)$ i.e., $\mathrm{con}(d_i, a)$.
(iv) We apply TokElim on the obvious $\varphi(a)$.

(v) We apply LTokElim on the obvious $\varPhi(U)$. In the inductive step we use the case (iv).
(vi) We apply TokElim on the obvious $\varphi(b)$. The cases $\varphi(*)$ and $\varphi(Cb_1b_2)$ are proved automatically. If $U \vdash Ca_1a_2$, then by the definition of entailment we have $\mathrm{Comp}(U)$, therefore, by (v), there exists $a\dot{\in}U$ such that $\mathrm{comp}(a)$. By Lemma 3(iv) and $\mathrm{Con}(U)$ we get that $\mathrm{con}(0, a)$ which is false. Then we use the Efq.
(vii) We apply TokElim on the obvious $\varphi(a)$. The case $\varphi(*)$ is trivial. The hypotheses of $\varphi(0)$ lead, because of (vi), to falsity, and then by Efq we get the conclusion of $\varphi(0)$. We prove $\varphi(Cd_1d_2)$ by $\varphi(d_1)$ and $\varphi(d_2)$. Suppose that $\mathrm{Con}(U), Cd_1d_2\dot{\in}U$ and $U \vdash Cb_1b_2$. By (iii) we get that $\mathrm{Con}(\mathrm{arg}_i(U))$, while by the definitions of arg_i and of entailment we have that $d_i\dot{\in}\,\mathrm{arg}_i(U)$ and $\mathrm{arg}_i(U) \vdash b_i$, for each $i = 1, 2$, respectively. To show $\mathrm{con}(Cb_1b_2, Cd_1d_2)$ it suffices to show $\mathrm{con}(b_1, d_1)$ and $\mathrm{con}(b_2, d_2)$. We show $\mathrm{con}(b_1, d_1)$ and we work similarly for $\mathrm{con}(b_2, d_2)$. We apply TokElim on

$$\varphi_1(b_1) := \mathrm{Con}(\mathrm{arg}_1(U)) \to d_1\dot{\in}\,\mathrm{arg}_1(U) \to \mathrm{arg}_1(U) \vdash b_1 \to \mathrm{con}(b_1, d_1).$$

The case $\varphi_1(*)$ is proved automatically, while for the case $\varphi_1(0)$ we apply (i) on $\mathrm{arg}_1(U)$. To show $\varphi_1(Cc_1c_2)$ we use the inductive hypothesis $\varphi(a_1)$ on $\mathrm{arg}_1(U), c_1$ and c_2.
(viii) We apply TokElim on the obvious $\varphi(a)$. The case $\varphi(*)$ is trivial, while the case $\varphi(0)$ is (i) and the case $\varphi(Ca_1a_2)$ is (vii).
(ix) This is proved directly from Lemma 3(i) and the case (viii).
(x) This is proved directly from the definition of $\mathrm{Con}(a :: U)$ and the case (ix). ◀

Corollary 8. (i) $\forall_U(U \vDash U)$.
(ii) $\forall_{U_1,U_2,U_3}(U_1 \vDash U_2 \to U_2 \vDash U_3 \to U_1 \vDash U_3)$.
(iii) $\forall_U(\forall_V(\mathrm{Con}(V) \to V \vDash U \to \mathrm{Con}(V{+\!\!+}U)))$.
(iv) $\forall_U(\forall_V(\mathrm{Con}(V) \to V \vDash U \to \mathrm{Con}(U)$.

Proof. (i) This is a direct consequence of Lemma 6(iv) and Lemma 4(v).
(ii) By Lemma 6(iv) it suffices to show that $\forall_a(a\dot{\in}U_3 \to U_1 \vdash a)$. By the other direction of Lemma 6(iv), if $a\dot{\in}U_3 \to U_2 \vdash a$. Using Lemma 6(v) on U_1, U_2, a we get $U_1 \vdash a$.
(iii) We apply LTokElim on the obvious $\varPhi(U)$. The case $\varPhi(\mathrm{nil})$ is proved by the trivial fact $V{+\!\!+}\mathrm{nil} = V$. We show next that $\varPhi(U) \to \varPhi(b :: U) := \mathrm{Con}(V) \to V \vDash b :: U \to \mathrm{Con}(V{+\!\!+}(b :: U))$ in two steps. First we show that $\mathrm{Con}(b :: (V{+\!\!+}U))$. By Lemma 5(x) it suffices to show that $V{+\!\!+}U \vdash b$ (we have that $\mathrm{Con}(V{+\!\!+}U)$ by the inductive hypothesis $\varPhi(U)$ and the fact that $V \vDash b :: U \to V \vDash U$). That we get by applying Lemma 6(v) on $V{+\!\!+}U, V$ and b. Then we show that $V{+\!\!+}(b :: U)\dot{\subseteq}b :: (V{+\!\!+}U)$; this suffices because then we have $\mathrm{Con}(b :: (V{+\!\!+}U))$ by Lemma 3(v). It is easy to see that $V\dot{\subseteq}b :: (V{+\!\!+}U)$, $b :: U\dot{\subseteq}b :: (V{+\!\!+}U)$, and then we conclude that $V{+\!\!+}(b :: U)\dot{\subseteq}b :: (V{+\!\!+}U)$ by using the following easy to show property of ${+\!\!+}$:

$$\forall_{U_1,U_2,U_3}(U_1\dot{\subseteq}U_3 \to U_2\dot{\subseteq}U_3 \to (U_1{+\!\!+}U_2)\dot{\subseteq}U_3).$$

(iv) By (iii) we know that $\mathrm{Con}(V{+\!\!+}U)$. Since it is easy to see that $U\dot{\subseteq}V{+\!\!+}U$, we get $\mathrm{Con}(U)$ by Lemma 3(v).

◀

A.3 The syntactic information system $\mathrm{SC}_{\mathbf{D}\to\mathbf{D}}$

Lemma 9. (i) $\forall_W(\forall_u(\mathrm{arrAltcon}(u, W) \leftrightarrow \forall_w(w\dot{\in}W \to \mathrm{arrcon}(u, w))))$.
(ii) $\forall_w(\mathrm{arrcon}(w, w))$.
(iii) $\forall_w(\mathrm{arrCon}(w :: \mathrm{Nil}))$.

(iv) $\forall_W(\forall_{w_1,w_2}(w_1\dot{\in}W \to w_2\dot{\in}W \to \mathrm{arrcon}(w_1,w_2)) \leftrightarrow \mathrm{arrCon}(W))$.
(v) $\forall_{W_1,W_2}(\mathrm{arrCon}(W_1) \to W_2\dot{\subseteq}W_1 \to \mathrm{arrCon}(W_2))$.

Proof. Similar to the proof of Lemma 3[21].

◀

Lemma 10. (i) $\forall_{W,u}(u :: W \vdash u)$.
(ii) $\forall_{W,u,w}(W \vdash w \to u :: W \vdash w)$.
(iii) $\forall_W(\forall_u((u\dot{\in}W \to W \vdash u))$.

Proof. (i) By the definition of $\vdash_{\mathbf{D}\to\mathbf{D}}$ and AltEntList we need to show that

$$\mathrm{PTS}((\mathrm{lft}(u) \vDash \mathrm{lft}(u)) :: \mathrm{AltEntList}(\mathrm{lft}(u), \mathrm{One}(W)), \mathrm{rht}(u) :: \mathrm{Two}(W)) \vdash \mathrm{rht}(u).$$

Since by Corollary 8(i) $(\mathrm{lft}(u) \vDash \mathrm{lft}(u)) = \mathtt{tt}$, by the definition of PTS we get

$$\mathrm{rht}(u) :: \mathrm{PTS}(\mathrm{AltEntList}(\mathrm{lft}(u), \mathrm{One}(W)), \mathrm{Two}(W)) \vdash \mathrm{rht}(u),$$

which holds by Lemma 4(v).
(ii) By the definitions of One, Two and AltEntList we need to show

$$\mathrm{PTS}(\mathrm{AltEntList}(\mathrm{lft}(w), \mathrm{lft}(u) :: \mathrm{One}(W)), \mathrm{rht}(u) :: \mathrm{Two}(W)) \vdash \mathrm{rht}(w) \leftrightarrow$$

$$\mathrm{PTS}((\mathrm{lft}(w) \vDash \mathrm{lft}(u)) :: \mathrm{AltEntList}(\mathrm{lft}(w), \mathrm{One}(W)), \mathrm{rht}(u) :: \mathrm{Two}(W)) \vdash \mathrm{rht}(w).$$

If $(\mathrm{lft}(w) \vDash \mathrm{lft}(u)) = \mathtt{tt}$, then by the definition of PTS we get

$$\mathrm{rht}(u) :: \mathrm{PTS}(\mathrm{AltEntList}(\mathrm{lft}(w), \mathrm{One}(W)), \mathrm{Two}(W)) \vdash \mathrm{rht}(w).$$

Since the hypothesis $W \vdash w$ is exactly the entailment

$$\mathrm{PTS}(\mathrm{AltEntList}(\mathrm{lft}(w), \mathrm{One}(W)), \mathrm{Two}(W)) \vdash \mathrm{rht}(w),$$

we get what we want by a trivial use of Lemma 6(v). If $(\mathrm{lft}(w) \vDash \mathrm{lft}(u)) = \mathtt{ff}$, then what we want to show is reduced directly to the above entailment of the hypothesis.
(iii) We apply the, similar to LTokElim defined, $\mathrm{LTokElim}_{\mathbf{D}\to\mathbf{D}}$ on the obvious $\Phi(W)$. The case $\Phi(\mathrm{Nil})$ is proved trivially through Efq. To show $\Phi(u :: W)$ i.e., $w\dot{\in}(u :: W) \to (u :: W) \vdash w$ we use (i), in case $w = u$, while in case $w\dot{\in}W$ we use the hypothesis $\Phi(W)$ and (ii). ◀

Lemma 11. (i) $\forall_U(\forall_{V_1,V_2,\mathsf{UU}}(V_1 \vDash V_2 \to$

$$\mathrm{PTS}(\mathrm{AltEntList}(V_2, \mathsf{UU}), U) \dot{\subseteq} \mathrm{PTS}(\mathrm{AltEntList}(V_1, \mathsf{UU}), U))).$$

(ii) $\forall_{U_1,U_2,W,a}(U_1 \vDash U_2 \to \mathrm{App}(W, U_2) \vdash a \to \mathrm{App}(W, U_1) \vdash a)$.
(iii) $\forall_{W_1}(\forall_{W_2,V}(W_2 \vDash W_1 \to \mathrm{App}(W_2, V) \vDash \mathrm{App}(W_1, V)))$.
(iv) $\forall_{W_1,W_2,u}(W_1 \vDash W_2 \to W_2 \vdash u \to W_1 \vdash u)$.

[21] Actually, both lemmas are special cases of an easy to formulate general proposition. This analogy in the treatment of consistency was revealed again by the implementation procedure.

Proof. (i) We apply LTokElim on the obvious $\Phi(U)$. The case $\Phi(\mathrm{nil})$ is proved by the trivial fact $\mathrm{PTS}(\mathbb{BB}, \mathrm{nil}) = \mathrm{nil}$, for each list of booleans $\mathbb{BB}$. To show $\Phi(U) \to \Phi(b :: U)$ for some fixed token b we assume $V_1 \vDash V_2$ and we prove

$$\Phi'(\mathbb{UU}) := \mathrm{PTS}(\mathrm{AltEntList}(V_2, \mathbb{UU}), b :: U) \dot{\subseteq} \mathrm{PTS}(\mathrm{AltEntList}(V_1, \mathbb{UU}), b :: U)))$$

using the corresponding elimination axiom. The case $\Phi'(\mathrm{nnil})$ holds trivially by the definition of PTS and the fact that $\mathrm{AltEntList}(V_2, \mathrm{nnil}) = \mathrm{nil}_{\mathbf{B}}$. For the inductive step we assume $\Phi'(\mathbb{UU})$ and we show $\Phi'(V :: \mathbb{UU})$, that is,

$$\mathrm{PTS}(\mathrm{AltEntList}(V_2, V :: \mathbb{UU}), b :: U) \dot{\subseteq} \mathrm{PTS}(\mathrm{AltEntList}(V_1, V :: \mathbb{UU}), b :: U))) \leftrightarrow$$

$$\mathrm{PTS}((V_2 \vDash V) :: \mathrm{AltEntList}(V_2, \mathbb{UU}), b :: U) \dot{\subseteq} \mathrm{PTS}((V_1 \vDash V) :: \mathrm{AltEntList}(V_1, \mathbb{UU}), b :: U))).$$

If $(V_2 \vDash V) = \mathtt{tt}$, therefore by Corollary 8(ii) $(V_1 \vDash V) = \mathtt{tt}$ too, the last inclusion is reduced to

$$b :: \mathrm{PTS}(\mathrm{AltEntList}(V_2, \mathbb{UU}), U) \dot{\subseteq} b :: \mathrm{PTS}(\mathrm{AltEntList}(V_1, \mathbb{UU}), U))),$$

which is trivially implied by the hypothesis $\Phi(U)$. If $(V_2 \vDash V) = \mathtt{ff}$, then we are reduced directly to the inductive hypothesis $\Phi(U)$.
(ii) Applying (i) on $(\mathrm{Two}(W), U_1, U_2, \mathrm{One}(W))$ we get $\mathrm{App}(W, U_2) \dot{\subseteq} \mathrm{App}(W, U_1)$, which entails, by Lemma 4(v), that $\mathrm{App}(W, U_1) \vDash \mathrm{App}(W, U_2)$. Then we just use Lemma 6(v).
(iii) We apply $\mathrm{LTokElim}_{\mathbf{D}\to\mathbf{D}}$ on the obvious $\Phi(W_1)$. The case $\Phi(\mathrm{Nil})$ follows by

$$\begin{aligned}\mathrm{App}(\mathrm{Nil}, V) &= \mathrm{PTS}(\mathrm{AltEntList}(V, \mathrm{One}(\mathrm{Nil})), \mathrm{Two}(\mathrm{Nil}))\\ &= \mathrm{PTS}(\mathrm{AltEntList}(V, \mathrm{nnil}), \mathrm{nil})\\ &= \mathrm{PTS}(\mathrm{nil}_{\mathbf{B}}, \mathrm{nil})\\ &= \mathrm{nil}.\end{aligned}$$

Assuming $\Phi(W_1)$ we show $\Phi(u :: W_1)$, for some arrow token u. We suppose that $W_2 \vDash (u :: W_1)$ and we prove that $\mathrm{App}(W_2, V) \vDash \mathrm{App}(u :: W_1, V)$. By definition

$$\begin{aligned}\mathrm{App}(u :: W_1, V) &= \mathrm{PTS}(\mathrm{AltEntList}(V, \mathrm{One}(u :: W_1)), \mathrm{Two}(u :: W_1))\\ &= \mathrm{PTS}(\mathrm{AltEntList}(V, \mathrm{lft}(u) :: \mathrm{One}(W_1)), \mathrm{rht}(u) :: \mathrm{Two}(W_1))\\ &= \mathrm{PTS}((V \vDash \mathrm{lft}(u)) :: \mathrm{AltEntList}(V, \mathrm{One}(W_1)), \mathrm{rht}(u) :: \mathrm{Two}(W_1)).\end{aligned}$$

If $(V \vDash \mathrm{lft}(u)) = \mathtt{tt}$, we need to show that

$$\begin{aligned}\mathrm{App}(W_2, V) \vDash\ & \mathrm{rht}(u) :: \mathrm{PTS}(\mathrm{AltEntList}(V, \mathrm{One}(W_1), \mathrm{Two}(W_1))\\ \vDash\ & \mathrm{rht}(u) :: \mathrm{App}(W_1, V).\end{aligned}$$

The fact that $\mathrm{App}(W_2, V) \vDash \mathrm{App}(W_1, V)$ follows from $\Phi(W_1)$ and the trivial implication $W_2 \vDash (u :: W_1) \to W_2 \vDash W_1$. Also, applying (ii) on $(V, \mathrm{lft}(u), W_2, \mathrm{rht}(u))$, and since $W_2 \vdash u$, we have that

$$V \vDash \mathrm{lft}(u) \to \mathrm{App}(W_2, \mathrm{lft}(u)) \vdash \mathrm{rht}(u) \to \mathrm{App}(W_2, V) \vdash \mathrm{rht}(u).$$

If $(V \vDash \mathrm{lft}(u)) = \mathtt{ff}$, then $\mathrm{App}(u :: W_1, V) = \mathrm{App}(W_1, V)$ and we apply the inductive hypothesis $\Phi(W_1)$.
(iv) Suppose that $W_1 \vDash W_2$ and $W_2 \vdash u := \mathrm{App}(W_2, \mathrm{lft}(u)) \vdash \mathrm{rht}(u)$. By (iii) we get that $\mathrm{App}(W_1, \mathrm{lft}(u)) \vDash \mathrm{App}(W_2, \mathrm{lft}(u))$ and by Lemma 6(v) we conclude that $\mathrm{App}(W_1, \mathrm{lft}(u)) \vdash \mathrm{rht}(u)$ i.e., $W_1 \vdash u$. ◀

Lemma 12. (i) $\forall_W(\forall_n(n<|W|\to \mathrm{Proj}(n,W)\dot{\in}W))$.
(iia) $\forall_W(\forall_n(n<|W|\to \mathrm{Proj}(n,\mathrm{One}(W))=\mathrm{lft}(\mathrm{Proj}(n,W))))$.
(iib) $\forall_W(\forall_n(n<|W|\to \mathrm{Proj}(n,\mathrm{Two}(W))=\mathrm{rht}(\mathrm{Proj}(n,W))))$.
(iii) $\forall_{\mathbb{UU}}(\forall_{n,U}(n<|\mathbb{UU}|\to \mathrm{Proj}(n,\mathrm{AltEntList}(U,\mathbb{UU}))=(U\vDash \mathrm{Proj}(n,\mathbb{UU}))))$.
(iv) $\forall_U(\forall_{\mathbb{BB},b}(b\dot{\in}\mathrm{PTS}(\mathbb{BB},U)\to \exists_n(n<|\mathbb{BB}|\wedge \mathrm{Proj}(n,\mathbb{BB})=\mathbb{tt}\wedge \mathrm{Proj}(n,U)=b)))$.
(v) $\forall_{b,W,U}(b\dot{\in}\mathrm{App}(W,U)\to \exists_n(n<|W|\wedge U\vDash \mathrm{lft}(\mathrm{Proj}(n,W))\wedge \mathrm{rht}(\mathrm{Proj}(n,W))=b))$.
(viii) $\forall_{W,u,w}(\mathrm{arrCon}(W)\to u\dot{\in}W\to W\vdash w\to \mathrm{arrcon}(w,u))$.
(ix) $\forall_{W,u}(\mathrm{arrCon}(W)\to W\vdash u\to \mathrm{arrAltcon}(u,W))$.
(x) $\forall_{W,u}(\mathrm{arrCon}(W)\to W\vdash u\to \mathrm{arrCon}(u::W))$.

Proof. (i) We apply $\mathrm{LTokElim}_{\mathbf{D}\to\mathbf{D}}$ on the appropriate $\Phi(W)$ in order to show

$$(*)\quad \forall_W(\forall_{w,n}(n<|w::W|\to \mathrm{Proj}(n,w::W)\dot{\in}(w::W))).$$

The case $\Phi(\mathrm{Nil})$ is derived by the obvious fact $\mathrm{Proj}(0,w::W)=w\dot{\in}(w::W)$. Next we suppose $\Phi(W)$ and we show $\Phi(u::W)$ i.e., $m<|w::(u::W)|\to \mathrm{Proj}(m,w::(u::W))\dot{\in}(w::(u::W))$. If $m=0$, we work as in the case $\Phi(\mathrm{Nil})$. Supposing that the above holds for m we show it for $m+1=S(m)$. If $m<|w::(u::W)|\leftrightarrow m<|u::W|$, then, since by definition $\mathrm{Proj}(m+1,w::(u::W))=\mathrm{Proj}(m,u::W)$, we take by the inductive hypothesis $\Phi(W)$ on u,m that $\mathrm{Proj}(m,u::W)\dot{\in}(u::W)$, therefore $\mathrm{Proj}(m,u::W)\dot{\in}(w::(u::W))$. In order now to show (i) we apply $\mathrm{LTokElim}_{\mathbf{D}\to\mathbf{D}}$ on the appropriate $\Phi'(W)$. The case $\Phi'(\mathrm{Nil})$ is derived trivially by Efq. The inductive step is proved directly by $(*)$.
(iia) By applying $\mathrm{LTokElim}_{\mathbf{D}\to\mathbf{D}}$ on the appropriate $\Phi(W)$ we show that

$$(**)\quad \forall_W(\forall_n(\forall_u(n<|u::W|\to \mathrm{Proj}(n,\mathrm{One}(u::W))=\mathrm{lft}(\mathrm{Proj}(n,u::W))))).$$

The case $\Phi(\mathrm{Nil})$ is derived by the obvious fact

$$\mathrm{Proj}(0,\mathrm{lft}(u)::\mathrm{One}(\mathrm{Nil}))=\mathrm{lft}(u)=\mathrm{lft}(\mathrm{Proj}(0,(u::\mathrm{Nil})),$$

and, if $0<n$, we use Efq. Supposing next $\Phi(W)$ we show $\Phi(w::W)$ i.e.,

$$\forall_n(\forall_u(n<|u::(w::W)|\to \mathrm{Proj}(n,\mathrm{One}(u::(w::W)))=\mathrm{lft}(\mathrm{Proj}(n,u::(w::W))))),$$

by applying $\mathrm{LTokElim}_{\mathbf{N}}$ again on the obvious formula $A(n)$. The case $A(0)$ is proved directly by

$$\mathrm{Proj}(0,\mathrm{lft}(u)::\mathrm{One}(w::W))=\mathrm{lft}(u)=\mathrm{lft}(\mathrm{Proj}(0,u::(w::W)).$$

In order to show $A(n)\to A(n+1)$ we suppose that $n+1<|u::(w::W)|\leftrightarrow n<|w::W|$, and then by the inductive hypothesis $\Phi(W)$ on n,w we get

$$\begin{aligned}\mathrm{Proj}(n+1,\mathrm{One}(u::(w::W)))&=\mathrm{Proj}(n+1,\mathrm{lft}(u)::\mathrm{One}(w::W))\\&=\mathrm{Proj}(n,\mathrm{One}(w::W)\\&=\mathrm{lft}(\mathrm{Proj}(n,w::W))\\&=\mathrm{lft}(\mathrm{Proj}(n+1,u::(w::W))).\end{aligned}$$

The proof of (iia) then follows by a simple application of $\mathrm{LTokElim}_{\mathbf{D}\to\mathbf{D}}$ on the obvious $\Phi'(W)$. The proof of (iib) is exactly the same.
(iii) The proof is similar to the proof of (iia); first we prove

$$\forall_{\mathbb{UU}}(\forall_n(\forall_{U,V}(n<|V::\mathbb{UU}|\to \mathrm{Proj}(n,\mathrm{AltEntList}(U,V::\mathbb{UU}))=(U\vDash \mathrm{Proj}(n,(V::\mathbb{UU}))))),$$

and then we prove easily (iii).
(iv) We apply $\mathrm{LTokElim}_{\mathbf{D}}$ on the obvious $\Phi(U)$. The case $\Phi(\mathrm{nil})$ is proved directly by Efq. To show $\Phi(U) \to \Phi(a :: U)$ we show

$$b \dot{\in} \mathrm{PTS}(\mathbb{BB}, a :: U) \to \exists_n (n < |\mathbb{BB}| \wedge \mathrm{Proj}(n, \mathbb{BB}) = \mathbb{tt} \wedge \mathrm{Proj}(n, a :: U) = b)$$

by applying $\mathrm{LTokElim}_{\mathbf{B}}$ on the obvious $\Phi'(\mathbb{BB})$. The case $\Phi'(\mathrm{nil}_{\mathbf{B}})$ is proved again by Efq. In order to show $\Phi'(\mathbb{BB}) \to \Phi(\mathbb{bb} :: \mathbb{BB})$ we suppose that $b \dot{\in} \mathrm{PTS}(\mathbb{bb} :: \mathbb{BB}, a :: U)$. If $\mathbb{bb} = \mathbb{tt}$, then $\mathrm{PTS}(\mathbb{tt} :: \mathbb{BB}, a :: U) = a :: \mathrm{PTS}(\mathbb{BB}, U)$. If $b = a$, then clearly $n = 0$. If $b \dot{\in} \mathrm{PTS}(\mathbb{BB}, U)$, then the required natural is $n + 1$, where n is the one determined by the hypothesis $\Phi'(\mathbb{BB})$, since by definition $\mathrm{Proj}(n, \mathbb{BB}) = \mathrm{Proj}(n + 1, \mathbb{bb} :: \mathbb{BB})$. If $\mathbb{bb} = \mathbb{ff}$, then $\mathrm{PTS}(\mathbb{ff} :: \mathbb{BB}, a :: U) = \mathrm{PTS}(\mathbb{BB}, U)$, and the required natural is again $n + 1$, where n is the one determined by the hypothesis $\Phi'(\mathbb{BB})$.
(v) By the definition of $\mathrm{App}(W, U)$ it suffices to apply (iv) on the pair

$$(U, \mathbb{BB}) = (\mathrm{Two}(W), \mathrm{AltEntList}(U, \mathrm{One}(W))).$$

Thus we get the existence of an n such that

$$n < |\mathrm{AltEntList}(U, \mathrm{One}(W))| \wedge \mathrm{Proj}(n, \mathrm{AltEntList}(U, \mathrm{One}(W))) = \mathbb{tt} \wedge \mathrm{Proj}(n, \mathrm{Two}(W)) = b.$$

Since it is easy to show that

$$|\mathrm{AltEntList}(U, \mathrm{One}(W))| = |\mathrm{One}(W)| = |W|,$$

we get the first required conjunct. Also,

$$\begin{aligned}
\mathrm{Proj}(n, \mathrm{AltEntList}(U, \mathrm{One}(W))) = \mathbb{tt} &\overset{(iii)}{\leftrightarrow} U \vDash \mathrm{Proj}(n, \mathrm{One}(W)) = \mathbb{tt} \\
&\leftrightarrow U \vDash \mathrm{Proj}(n, \mathrm{One}(W)) \\
&\overset{(iia)}{\leftrightarrow} U \vDash \mathrm{lft}(\mathrm{Proj}(n, W)).
\end{aligned}$$

Finally, by (iib) we have that $\mathrm{Proj}(n, \mathrm{Two}(W)) = b \leftrightarrow \mathrm{rht}(\mathrm{Proj}(n, W)) = b$.
(viii) It suffices to consider only the case $\mathrm{Con}(\mathrm{lft}(w) {+\!\!+} \mathrm{lft}(u)) = \mathbb{tt}$ and then show that $\mathrm{con}(\mathrm{rht}(w), \mathrm{rht}(u)) = \mathbb{tt}$. By Lemma 11(ii) we have that

$$\mathrm{lft}(w) {+\!\!+} \mathrm{lft}(u) \vDash \mathrm{lft}(w) \to \mathrm{App}(W, \mathrm{lft}(w)) \vdash \mathrm{rht}(w) \to \mathrm{App}(W, \mathrm{lft}(w) {+\!\!+} \mathrm{lft}(u)) \vdash \mathrm{rht}(w).$$

Since by Lemma 3(v) we have that $\mathrm{lft}(w) {+\!\!+} \mathrm{lft}(u) \vDash \mathrm{lft}(w)$ and by the definition of $W \vdash w$ we have that $\mathrm{App}(W, \mathrm{lft}(w)) \vdash \mathrm{rht}(w)$, we conclude $\mathrm{App}(W, \mathrm{lft}(w) {+\!\!+} \mathrm{lft}(u)) \vdash \mathrm{rht}(w)$. By Lemma 5(x) we then get

$$\mathrm{Con}(\mathrm{rht}(w) :: \mathrm{App}(W, \mathrm{lft}(w) {+\!\!+} \mathrm{lft}(u))),$$

and by applying (vii) on $W, u, \mathrm{lft}(w)$ we have that

$$\mathrm{rht}(u) \dot{\in} \mathrm{App}(W, \mathrm{lft}(w) {+\!\!+} \mathrm{lft}(u)).$$

Since $\mathrm{rht}(w), \mathrm{rht}(u) \dot{\in} (\mathrm{rht}(w) :: \mathrm{App}(W, \mathrm{lft}(w) {+\!\!+} \mathrm{lft}(u)))$, through Lemma 3(iv) we get the required $\mathrm{con}(\mathrm{rht}(w), \mathrm{rht}(u))$.
(ix) By Lemma 9(i) we only nee to show $w \dot{\in} W \to \mathrm{arrcon}(u, w)$, which is derived directly from (viii).
(x) By definition $\mathrm{arrCon}(u :: W) = \mathrm{arrAltcon}(u, W) \ \wedge\ \mathrm{arrCon}(W)$. By case (ix) we get $\mathrm{arrAltcon}(u, W)$, while by hypothesis we already have $\mathrm{arrCon}(W)$. ◀

A.4 A point-free density theorem in $\mathbb{D}$

Lemma 15. (i) $\forall_U(\forall_V(U \dot{\subseteq} \overline{V} \leftrightarrow V \vDash U))$.
(ii) $\forall_V(\mathrm{Con}(V) \to \forall_{U,a}(\mathrm{Ideal}(\overline{V}, U, a)))$.

Proof. (i) We apply $\mathrm{LTokElim}_{\mathbf{D}}$ on the obvious $\Phi(U)$. The case $\Phi(\mathrm{nil})$, that is, $\mathrm{nil}\dot{\subseteq}\overline{V} \leftrightarrow V \vDash \mathrm{nil}$, is direct. Next we suppose that $\Phi(U)$ and we show $\Phi(b :: U)$ i.e.,

$$b\dot{\in}\overline{V} \wedge U\dot{\subseteq}\overline{V} \leftrightarrow V \vdash a \wedge V \vDash U.$$

It is obvious by the definition of $\dot{\in}$ that $b\dot{\in}\overline{V} \leftrightarrow V \vdash a$, while by the inductive hypothesis $\Phi(U)$ we get $U\dot{\subseteq}\overline{V} \leftrightarrow V \vDash U$.
(ii) We assume $V, \mathrm{Con}(V), U, a$ and we show $\mathrm{Ideal}(\overline{V}, U, a)$ i.e.,

$$[U\dot{\subseteq}\overline{V} \to_{\mathbf{B}} \mathrm{Con}(U)] \ \wedge_{\mathbf{B}} \ [(U\dot{\subseteq}\overline{V} \wedge_{\mathbf{B}} U \vdash a) \to_{\mathbf{B}} V \vdash a].$$

For the first conjunct it suffices to consider the case $(U\dot{\subseteq}\overline{V}) = \mathtt{tt}$. By (i) we get that $V \vDash U$, and by Corollary 8(iv) we conclude $\mathrm{Con}(U)$. For the second conjunct it suffices also to assume $(U\dot{\subseteq}\overline{V} \wedge_{\mathbf{B}} U \vdash a) = \mathtt{tt}$. By (i) again we have that $V \vDash U$, while by Lemma 6(v) we conclude $V \vdash a$. ◀

▶ **Lemma 20.** *(i)* $\forall_a(\mathrm{total}(\mathrm{t}(a)))$.
(ii) $\forall_a(\mathrm{t}(a) :: \mathrm{nil} \vdash a)$.

Proof. Both proofs are simple applications of $\mathrm{TokElim}_{\mathbf{D}}$ on the obvious formulas $\phi(a)$. ◀

Lemma 17. (i) $\forall_a(\sup(a, a) = a)$.
(ii) $\forall_a(\forall_b(\sup(a, b) = \sup(b, a)))$.
(iiia) $\forall_a(\forall_b(\mathrm{con}(a, b) \to \sup(a, b) :: \mathrm{nil} \vdash a))$.
(iiib) $\forall_a(\forall_b(\mathrm{con}(a, b) \to \sup(a, b) :: \mathrm{nil} \vdash b))$.
(iv) $\forall_a(\forall_b(\forall_c(\mathrm{con}(a, b) \to \mathrm{con}(b, c) \to \mathrm{con}(a, c) \to \mathrm{con}(\sup(a, b), \sup(b, c)))))$.

Proof. Cases (i) and (ii) are simple applications of $\mathrm{TokElim}_{\mathbf{D}}$ on the obvious formulas $\phi(a)$.
(iiia) We apply $\mathrm{TokElim}_{\mathbf{D}}$ on the obvious formula $\phi(a)$. The case $\phi(*)$ is trivial by the definition of $\vdash$. Since by the definition of con we have that $\mathrm{con}(0, b) \leftrightarrow b = * \vee b = 0$, then $\sup(0, b) = 0 \to (0 :: \mathrm{nil}) \vdash 0$. Next we suppose that $\phi(a_1), \phi(a_2)$ and we show $\phi(Ca_1a_2)$ i.e.,

$$\forall_b(\mathrm{con}(Ca_1a_2, b) \to \sup(Ca_1a_2, b) :: \mathrm{nil} \vdash Ca_1a_2),$$

by applying $\mathrm{TokElim}_{\mathbf{D}}$ on the obvious formula $\phi'(a)$. The case $\phi'(*)$ is proved by Lemma 4(v), since by definition $\sup(Ca_1a_2, *) = Ca_1a_2$, while the case $\phi'(0)$ is reduced to the formula $\mathtt{ff} \to \mathtt{ff}$. Next we suppose that $\phi'(b_1), \phi'(b_2)$ and we show $\phi'(Cb_1b_2)$. Since $\mathrm{con}(Ca_1a_2, Cb_1b_2) = \mathrm{con}(a_1, b_1) \wedge \mathrm{con}(a_2, b_2)$ and $\sup(Ca_1a_2, Cb_1b_2) = C\sup(a_1, b_1)\sup(a_2, b_2)$ we get

$$C\sup(a_1, b_1)\sup(a_2, b_2) :: \mathrm{nil} \vdash Ca_1a_2 \leftrightarrow \sup(a_1, b_1) :: \mathrm{nil} \vdash a_1 \wedge \sup(a_2, b_2) :: \mathrm{nil} \vdash a_2,$$

which follows by applying the inductive hypotheses $\phi(a_1), \phi(a_2)$ on b_1, b_2, respectively.
(iiib) The proof is similar to the proof of (iiia).
(iv) We apply $\mathrm{TokElim}_{\mathbf{D}}$ on the obvious formula $\phi(a)$. The case $\phi(*)$ takes the form

$$\forall_b(\forall_c(\mathrm{con}(*, b) \to \mathrm{con}(b, c) \to \mathrm{con}(*, c) \to \mathrm{con}(b, \sup(b, c)).$$

By (iiia) we take $\sup(b, c) :: \mathrm{nil} \vdash b$ therefore, by Lemma 5(x), $\mathrm{Con}(b :: \sup(b, c) :: \mathrm{nil})$, which entails, by Lemma 3(iv), $\mathrm{con}(b, \sup(b, c))$. The case $\phi(0)$ takes the form

$$\forall_b(\forall_c(\mathrm{con}(0, b) \to \mathrm{con}(b, c) \to \mathrm{con}(0, c) \to \mathrm{con}(\sup(0, b), \sup(b, c)).$$

By hypotheses $\mathrm{con}(0, b), \mathrm{con}(0, c)$ though, we conclude that b, c are either 0 or $*$. In all four cases the required consistency follows trivially. Next we suppose $\phi(a_1), \phi(a_2)$ and we show $\phi(Ca_1a_2)$ i.e.,

$$\forall_b(\forall_c(\mathrm{con}(Ca_1a_2, b) \to \mathrm{con}(b, c) \to \mathrm{con}(Ca_1a_2, c) \to \mathrm{con}(\sup(Ca_1a_2, b), \sup(b, c)),$$

by applying $\mathrm{TokElim}_{\mathbf{D}}$ on the obvious formula $\phi'(b)$. The conclusion of the case $\phi'(*)$ becomes $\mathrm{con}(Ca_1a_2, c)$, which is already a hypothesis. The case $\phi'(0)$ is proved directly by Efq, since $\mathrm{con}(Ca_1a_2, 0) = \mathsf{ff}$. Next we suppose $\phi'(b_1), \phi'(b_2)$ and we show $\phi'(Cb_1b_2)$ i.e.,

$$\forall_c(\mathrm{con}(Ca_1a_2, Cb_1b_2) \to \mathrm{con}(Cb_1b_2, c) \to \mathrm{con}(Ca_1a_2, c) \to$$

$$\to \mathrm{con}(\sup(Ca_1a_2, Cb_1b_2), \sup(Cb_1b_2, c)),$$

by applying $\mathrm{TokElim}_{\mathbf{D}}$ on the obvious formula $\phi''(c)$. The conclusion of $\phi''(*)$ becomes

$$\mathrm{con}(C \sup(a_1, b_1) \sup(a_2, b_2), Cb_1b_2) \leftrightarrow \mathrm{con}(\sup(a_1, b_1), b_1) \wedge \mathrm{con}(\sup(a_2, b_2), b_2).$$

We derive the first conjunct and similarly we work for the second. By the hypothesis $\mathrm{con}(Ca_1a_2, Cb_1b_2)$ we have that $\mathrm{con}(a_1b_1)$ and by (iiib) we get $\sup(a_1, b_1) :: \mathrm{nil} \vdash b_1$. Then we conclude that $\mathrm{con}(\sup(a_1, b_1), b_1)$ arguing in exactly the same way as in the case of $\phi(*)$. The case $\phi''(0)$ is proved directly from Efq, since $\mathrm{con}(Cb_1b_2, 0) = \mathsf{ff}$. Finally we suppose $\phi''(c_1), \phi''(c_2)$ and we prove $\phi''(Cc_1c_2)$ i.e.,

$$\mathrm{con}(Ca_1a_2, Cb_1b_2) \to \mathrm{con}(Cb_1b_2, Cc_1c_2) \to \mathrm{con}(Ca_1a_2, Cc_1c_2) \to$$

$$\to \mathrm{con}(\sup(Ca_1a_2, Cb_1b_2), \sup(Cb_1b_2, Cc_1c_2).$$

The conclusion of this case takes the form

$$\mathrm{con}(C \sup(a_1, b_1) \sup(a_2, b_2), C \sup(b_1, c_1) \sup(b_2, c_2)) \leftrightarrow$$

$$\mathrm{con}(\sup(a_1, b_1), \sup(b_1, c_1)) \wedge \mathrm{con}(\sup(b_1, c_1), \sup(b_2, c_2)).$$

We derive the first conjunct and we work similarly for the second. The hypotheses $\mathrm{con}(Ca_1a_2, Cb_1b_2)$, $\mathrm{con}(Cb_1b_2, Cc_1c_2)$ and $\mathrm{con}(Ca_1a_2, Cc_1c_2)$ entail $\mathrm{con}(a_1, b_1), \mathrm{con}(b_1, c_1)$, and $\mathrm{con}(a_1, c_1)$, respectively. Applying the inductive hypothesis $\phi(a_1)$ on b_1, c_1, we get $\mathrm{con}(\sup(a_1, b_1), \sup(b_1, c_1)$. ◀