Lecture Notes
Computability and Complexity
KGRC, Vienna, Summer 2015.

Moritz Müller

moritz.mueller@univie.ac.at

June 30, 2015

1 Introduction

Barwise begins his chapter on α-recursion theory with the words: "There are many equivalent definitions of the class of recursive functions on the natural numbers. [...] As the various definitions are lifted to domains other than the integers (e. g., admissible sets) some of the equivalences break down. This break-down provides us with a laboratory for the study of recursion theory." ([4, p.153]) We focus on the definability theoretic and the recursion theoretic characterization of the computable functions on ω. According to the first a function on ω is computable if it is Σ_1-definable (in the language of arithmetic), according to the second a function is computable if it is obtainable from certain simple initial functions by means of composition, primitive recursion and the μ-operator.

In α-recursion theory one considers functions which are Σ_1-definable (in the language of set theory) in an admissible set, that is, a model of Kripke-Platek set theory KP. Recall, KP consists in the axioms for Extensionality, Union, Pair, Δ_0-Separation, Δ_0-Collection and $\in$-Induction

$$(\forall y(\forall u \in y\ \varphi(u, \vec{x}) \to \varphi(y, \vec{x})) \to \varphi(x, \vec{x})) \tag{1}$$

for all formulas $\varphi(x, \bar{x})$. To some extent this generalization of the computability concept extends to the recursion theoretic view. The Σ-Recursion Theorem states that the Σ_1-definable functions of KP are closed under $\in$-recursion. This implies that the primitive recursive set functions (PRSF) of [?] are all Σ_1-definable in KP. By definition a function is PRSF if it is obtained from certain simple initial functions by means of composition and $\in$-recursion. Hence, PRSF is a generalization of the primitive recursive functions according to the recursion theoretic view.

Rathjen [13] who observed that this generalization primitive recursive functions extends to the definability theoretic view. Rathjen noted that the Σ-Recursion Theorem is provable in KP_1, the fragment of KP where $\in$-induction is adopted for Σ_1-formulas only, and proved the converse: the PRSF functions are precisely those that are Σ_1-definable in KP_1.

Rathjen's theorem parallels Parson's (cf. [10]) stating that on ω, the primitive recursive functions are precisely those that are Σ_1-definable in $I\Sigma_1$, the theory of arithmetic based on the induction scheme for Σ_1-formulas. Maybe one can thus accept PRSF give a reasonable generalization of primitive recursive computation to arbitrary sets.

It is only natural to wonder whether one can find a similarly good analogue of polynomial time computation on arbitrary sets. In the prequel of this paper [6] such an analogue has been proposed following the recursion theoretic view: Cobham [11] characterized the polynomial time computable functions as those obtained from certain simple initial functions, including the smash function #, by means of composition and so-called *limited recursion on notation*. This type of recursion restricts both the depth of the recursion and the size of recursive values. Namely, a recursion on notation on n has depth roughly $\log n$; being limited means that all values are required to be bounded by some smash term $n\#\cdots\#n$. In [6] an analogue of the smash function for sets is introduced. The role of recursion on notation is taken by $\in$-recursion, and being limited is taken to mean to be in a certain sense embeddable into smash term. This way, [6] defined the class of *Cobham recursive set functions (CRSF)*, an analogue of polynomial time computation on arbitrary sets. This paper tries to extend the analogy to the definability theoretic view.

A definability theoretic characterization of polynomial time functions on ω has been given by Buss (cf. [10]). In analogy with Parson's theorem, $I\Sigma_1$ is replaced by S^1_2 and Σ_1 is replaced by Σ^b_1, "bounded" Σ_1-formulas. The theory S^1_2 has a language including the smash # and is based on an induction scheme for Σ^b_1-formulas such that the depth of an induction is as the depth of a recursion on notation.

Both directions in Buss' characterization hold in a strong way. First, S^2_1 *defines* polynomial time functions in that one can conservatively add Σ^b_1-defined function symbols such that S^1_2 proves all defining equations coming from Cobham's recursion theoretic characterization. Conversely, polynomial time functions *witness* certain simple S^2_1-theorems. More precisely, for every S^2_1-theorem of the form $\forall x \exists y \varphi(x, y)$ with φ in Δ^b_0, i.e. a "bounded" Δ_0-formula, the mentioned definitorial extension of S^2_1 proves $\varphi(x, f(x))$ for one of its functions symbols.

We proceed in analogy with Rathjen's theorem and define a theory K_2 and "bounded" Σ_1-formulas, denoted $\Sigma^{\preccurlyeq}_1$, using the set smash from [6]. The theory K_2 has a language containing some basic CRSF functions including the set smash and has $\in$-Induction restricted to $\Sigma^{\preccurlyeq}_1$-formulas.

In Section 1, we show that K_2 *defines* CRSF functions in the sense that it has a conservative extension by ${\Sigma_1}^{\preccurlyeq}$ -defined function symbols such that it proves all defining equations coming from the recursion theoretic definition of CRSF (Theorem **??**). But the analogue of the Witnessing Theorem fails. We show in Section 2 that witnessing can be ressurrected in the presence of an arbitrary Global Choice (Witnessing Theorem **??**). Thus Buss' theorem for S^1_2 and polynomial time on ω has a full analogy for K_2 and CRSF on universes of sets equipped with a global choice function.

Finally, Section 3 addresses the question as to what extent Global Choice is conservative over certain local choice principles.

We mention some related work. The characterization of polynomial time by Turing machines has been generalized in [?] to allow binary input strings of length ω. We refer to [7] for some comparison with CRSF. Yet another characterization of polynomial time comes from the Immerman-Vardi Theorem from descriptive complexity theory (cf. [12]). Following this, Sazonov [14] gives a theory operating with terms allowing for least fixed-point constructs to capture polynomial time computations on (binary encodings of) Mostowski graphs of hereditarily finite sets. Not all of Sazonov's set functions are CRSF [6, p.29]. But, proceeding dually, under a suitable encoding of binary strings by hereditarily finite sets, CRSF does capture polynomial time [6, Theorems 30, 31]. Arai [1] gave a different such class of functions. His *Predicatively Computable Set Functions (PCSF)* form a subclass of the *Safe Recursive Set Functions (SRSF)* from [5]. SRSF is defined by analogy of Bellantoni and Cook's [8] recursion theoretic characterization of polynomial time, different from Cobham's. We refer to [6] for a comparison of PCSF and CRSF.

2 Cobham recursive set functions

We review some definitions and results from [6]. We are going to formalize most of these results in suitable fragments of KP.

The Mostowski graph of a set x has as vertices the elements of the transitive closure $\mathrm{tc}^+(x) := \mathrm{tc}(\{x\})$ and directed edges from u to v if $u \in v$. Every such graph has a unique source and a unique sink. The *set smash* function $x\#y$ replaces each vertex of x by a copy of (the graph of) y with incoming edges now going to the source of y and outgoing edges now leaving the sink of y. It is defined using the *set composition* function $x \odot y$ which places a copy of x above y and identifies the source of x with the sink of y. Formally,

$$\begin{aligned} x \odot y &:= \begin{cases} y & \text{if } x = \emptyset \\ \{u \odot y : u \in x\} & \text{else,} \end{cases} \\ x\#y &:= y \odot \{u\#y : u \in x\}. \end{aligned}$$

The function $\sigma_{x,y}(u,v)$ is a graph isomorphism onto $x\#y$ and from the graph with vertices $\mathrm{tc}^+(x) \times \mathrm{tc}^+(y)$ and an directed edge from $\langle u,v\rangle$ to $\langle u',v'\rangle$ if either $u' = u, v' \in v$ or $u \in u', v = y, v' = 0$ (cf. [6, Section 2]). Formally,

$$\sigma_{x,y}(u,v) := v \odot \{u'\#y : u' \in u\}.$$

The corresponding projections $\pi_{1,x,y}(z), \pi_{2,x,y}(z)$ satisfy for $z \in \mathrm{tc}^+(x\#y)$

$$\sigma_{x,y}(\pi_{1,x,y}(z), \pi_{2,x,y}(z)) = z.$$

A $\#$-*term* is a function obtained by composition from $\#, \odot$ and the constant 1. Such terms serve as analogues of polynomial bounds, the bounding relation $\preccurlyeq$ being defined as follows: $x \preccurlyeq y$ means that there is an *embedding* that maps $u \in \mathrm{tc}(x)$ to pairwise disjoint non-empty $V_u \subseteq \mathrm{tc}(y)$ such that whenever $u \in u'$ and $v' \in V_{u'}$ there exists $v \in V_u \cap \mathrm{tc}(v')$. The notation $\tau(\cdot, \vec{y}) : x \preccurlyeq y$ means that $u \mapsto \tau(u, \vec{y})$ is such an embedding.

Definition 1. The *Cobham recursive set functions (CRSF)* are obtained from the *initial functions*, namely projections $\pi^r_j(x_1, \ldots, x_r) := x_j$, constant $0 := \emptyset$, $\mathrm{pair}(x, y) := \{x, y\}$, set smash $x\#y$, and the conditional

$$\mathrm{cond}_{\in}(x, y, u, v) := \begin{cases} x & \text{if } u \in v \\ y & \text{else,} \end{cases}$$

by composition and *Cobham Recursion*: if $g(x, y, \vec{x},), \tau(u, x, \vec{x})$ are CRSF and $t(x, \vec{x})$ is a #-term, then the function $f(x, \vec{x})$ given by

$$f(x, \vec{x}) = g(x, \{f(u, \vec{x}) : u \in x\}, \vec{x})$$

is also CRSF provided that $\tau(\cdot, x, \vec{x}) : f(x, \vec{x}) \preccurlyeq t(x, \vec{x})$ holds for all $x, \vec{x}$.

The definition given in [6] is more liberal in the sense that the bound t is not demanded to be a #-term but allowed to be an arbitrary CRSF function. That this is equivalent to the more restrictive definition given here s proved in [6, Theorem 17]. It is not hard to show that CRSF functions are "polynomially bounded". More precisely, for every CRSF function $f(\vec{x})$ there is a smash term $t(\vec{x})$ and a CRSF function $\tau(u, \vec{x})$ such that $\tau(\cdot, \vec{x}) \preccurlyeq t(\vec{x})$. This is proved in [6, Theorem 21]. The proof relies on the following, important lemma [6, Lemma 21]. It implies that our analogue of being polynomially bounded has some of the monotonicity properties one would expect from such a concept.

Lemma 2. *Assume $t(x, \vec{x})$ is a #-term and $\tau(u, \vec{y})$ is in CRSF such that $\tau(\cdot, \vec{y}) : x \preccurlyeq y$. Then there exists $\sigma(u, \vec{y})$ in CRSF such that $\sigma(\cdot, \vec{y}) : t(x, \vec{x}) \preccurlyeq t(y, \vec{x})$.*

It is not hard to show that set composition $x \odot y$ and a kind of inverse of it $\odot^{-1}$, a function satisfying $(x \odot y) \odot^{-1} y = x$, are CRSF. Also the functions $\sigma_{x,y}(u, v), \pi_{1,x,y}(z), \pi_{2,x,y}(z)$ are CRSF [6, Theorem 13]. More interestingly, crossproduct $x \times y$ and rank $\mathrm{rk}(x)$ are CRSF [6, Theorems 14, 15]. Indeed, CRSF is closed as follows [6, Theorems 13, 23, 26].

Theorem 3.

1. *If $g(u, \vec{x})$ is CRSF, then so is $f(x, \vec{x}) := \{u \in x : g(u, \vec{x}) \neq 0\}$.*
2. *If $g(u, x, \vec{x})$ is CRSF, then so is $f(x, \vec{x}) := \{g(u, x, \vec{x}) : u \in x\}$.*
3. *If $g(x, y, \vec{x}), \tau(u, x, \vec{x}), h(x, \vec{x})$ are CRSF, then so is*

$$f(x, \vec{x}) := g(x, \{\langle u, f(u, \vec{x})\rangle : u \in x\}, \vec{x})$$

provided $\tau(\cdot, x, \vec{x}) : f(x, \vec{x}) \preccurlyeq h(x, \vec{x})$ holds for all $\vec{x}, x$.

3 A theory for CRSF

In this section we present a weak set theory able to define CRSF functions and prove their recursive equations. For expository purposes we start with a very weak theory K_0 and stepwise strengthen it.

3.1 K_0 and the basic language L_0

As outlined in the introduction our theory K_2 is going to be formulated in a language

$$L_0 := \{\in, 0, 1, \mathrm{union}, \mathrm{pair}, \times, \mathrm{tc}, \odot, \odot^{-1}, \#\}.$$

The meaning of these symbols is given by their *defining axioms*:

symbol	*defining axiom*
0	$(u \in 0 \leftrightarrow u \neq u)$,
1	$(u \in 1 \leftrightarrow u = 0)$,
$\mathrm{union}(x)$	$(u \in \mathrm{union}(x) \leftrightarrow \exists y \in x\ u \in y)$
$\mathrm{pair}(x, z)$	$(u \in \mathrm{pair}(x, y) \leftrightarrow u = x \vee u = y)$
$x \times y$	$(u \in x \times y \leftrightarrow \exists x' \in x \exists y' \in y\ u = \langle x', y' \rangle)$
$\mathrm{tc}(x)$	$(u \in \mathrm{tc}(x) \leftrightarrow u \in x \vee \exists y \in x\ u \in \mathrm{tc}(y))$
$x \odot y$	$(u \in x \odot y \leftrightarrow \exists z \in x\ u = z \odot y \vee (x = 0 \wedge u \in y))$
$x \odot^{-1} y$	$(u \in x \odot^{-1} y \leftrightarrow y \in \mathrm{tc}(x) \wedge \exists x' \in x\ u = x' \odot^{-1} y)$
$x\#y$	$\exists w\ (w = \{v : \exists z \in x\ v = z\#y\} \wedge (u \in x\#y \leftrightarrow u \in y \odot w))$

Here and below we use the following standard abbreviations: $\{x, y\}$ stands for $\mathrm{pair}(x, y)$, $\{x\}$ for $\mathrm{pair}(x, x)$ and $\langle x, y \rangle$ for $\{\{x\}, \{x, y\}\}$. We write $x \subseteq y$ for $\forall u \in x\ u \in y$. We use comprehension terms as in the $\#$-axiom understanding that $z = \{u \in x : \varphi(u, x, \vec{x})\}$ abbreviates $(\forall u \in z(u \in x \wedge \varphi) \wedge \forall u \in x(\varphi \to u \in z))$; similarly, $z = \{u : \varphi(u, \vec{x})\}$ abbreviates $(\forall u \in z\varphi(u, \vec{x}) \wedge \forall u(\varphi(u, \vec{x}) \to u \in z))$.

When denoting formula classes we shall always make explicit the underlying language. For a language L containing $\in$, $\Delta_0(L)$ denotes the class of L-formulas with all quantifiers $\in$-bounded, i.e. of the form $\exists x \in t \ldots$ or $\forall x \in t \ldots$ where t is an L-term not involving x; we refer to t as an $\in$-*bounding term in* φ.

All theories we consider are going to be extensions of the following theory K_0. When saying a formula belongs to a theory we actually mean the universal closure of the formula.

Definition 4. The theory K_0 consists of

the defining axioms for symbols in L_0,

(Extensionality) $(x \subseteq y \wedge y \subseteq x \to x = y)$,

($\Delta_0(L_0)$-Separation) $\exists z\ z = \{u \in x : \varphi(u, \vec{x})\}$ for $\varphi(u, \vec{x}) \in \Delta_0(L_0)$,

($\Delta_0(L_0)$-Induction) (1) for $\varphi(x, \vec{x}) \in \Delta_0(L_0)$,

Remark 5. For a formula class Φ let $\neg\Phi := \{\neg\varphi : \varphi \in \Phi\}$. Then $\neg\Phi$-Induction is logically equivalent to Φ-*Foundation*, i.e. for $\varphi(u, \vec{x}) \in \Phi$

$$(\varphi(x, \vec{x}) \to \exists u(\varphi(u, \vec{x}) \wedge \forall u' \in u\ \neg\varphi(u', \vec{x})))$$

We verify certain basic properties of symbols in L_0 and the embedding relation $\preccurlyeq$. This relation is formally expressed for an arbitrary function symbol $\tau(u,\vec{x})$ by the following formula $\tau(\cdot,\vec{x}) : x \preccurlyeq y$.

$$\begin{aligned}
&\forall u \in x \forall v \in \tau(u,\vec{x})\ v \in y \\
&\wedge\ \forall u \in x\ \tau(u,\vec{x}) \neq 0 \\
&\wedge\ \forall u, u' \in x \forall v \in \tau(u,\vec{x})\ (u \neq u' \to v \notin \tau(u',\vec{x})) \\
&\wedge\ \forall u \in x \forall u' \in u \forall v \in \tau(u,\vec{x}) \exists v' \in \tau(u',\vec{x})\ v' \in \mathrm{tc}(v).
\end{aligned}$$

However, we shall not be able to define function symbols computing relevant embeddings until some substantial bootstrapping of K_0 and its extensions has been carried out. Meanwhile, we will have to work with other version of the embeddability relation. We refer to these informally as *non-uniform versions*.

We let $x \preccurlyeq y$ stand for $\exists z\ z : x \preccurlyeq y$ where $z : x \preccurlyeq y$ stands for

$$\begin{aligned}
&\forall w \in z \exists u \in \mathrm{tc}(x) \exists v \in \mathrm{tc}(y)\ w = \langle u, v\rangle \\
&\wedge\ \forall u \in \mathrm{tc}(x) \exists v \in \mathrm{tc}(y)\ \langle u, v\rangle \in z \\
&\wedge\ \forall u, u' \in \mathrm{tc}(x) \forall v \in \mathrm{tc}(y) (u \neq u' \wedge \langle u, v\rangle \in z \to \langle u', v\rangle \notin z) \\
&\wedge\ \forall u, u' \in \mathrm{tc}(x) \forall v' \in \mathrm{tc}(y) (u \in u' \wedge \langle u', v'\rangle \in z \to \exists v \in \mathrm{tc}(v')\ \langle u, v\rangle \in z).
\end{aligned}$$

In some situations, even when we lack a function symbol $\tau(\cdot,\vec{x})$, we shall at least be able to describe the embedding z by some formula:

Definition 6. Let L be a language containing L_0, T an K-theory and $t_1(\vec{x}), t_2(\vec{x})$ be L_0-terms. That an L-formula $\varepsilon(u,\vec{x})$ *describes an embedding of* $t_1(\vec{x})$ *into* $t_2(\vec{x})$ *in* T means that T proves

$$\exists z (z = \{u \in \mathrm{tc}(t_1(\vec{x})) \times \mathrm{tc}(t_2(\vec{x})) : \varepsilon(u,\vec{x})\} \wedge z : t_1(\vec{x}) \preccurlyeq t_2(\vec{x})).$$

We abbreviate this formula by $\varepsilon(\cdot,\vec{x}) : t_1(\vec{x}) \preccurlyeq t_2(\vec{x})$. If this holds for some $\Delta_0(L)$-formula we say T *describes an embedding of* $t_1(\vec{x})$ *into* $t_2(\vec{x})$.

Lemma 7. *The theory* K_0 *proves*

1. (Set Foundation) $(x \neq 0 \to \exists y \in x \forall u \in y\ u \notin x)$,
2. $(y \in \mathrm{tc}(x) \to y \subseteq \mathrm{tc}(x))$, $(x \subseteq y \to \mathrm{tc}(x) \subseteq \mathrm{tc}(y))$, $\mathrm{tc}(x) = \mathrm{tc}(\mathrm{tc}(x))$,
3. $(e : x \preccurlyeq y \leftrightarrow e : \mathrm{tc}(x) \preccurlyeq y)$, $(x \subseteq y \wedge y \preccurlyeq z \to x \preccurlyeq z)$, $(x \subseteq y \to x \preccurlyeq y)$,
4. $(u \in \mathrm{tc}(x \odot y) \leftrightarrow u \in \mathrm{tc}(y) \vee \exists u' \in \mathrm{tc}(x)\ u = u' \odot y)$,
5. $(x \odot y) \odot^{-1} y = x$, $(x \neq x' \to x \odot y \neq x' \odot y)$.

Proof. Argue in K_0. (1.) $\Delta_0(L_0)$-Induction for the formula $u \notin x$ is logically equivalent to $(\neg u \notin x \to \exists y (\forall u \in y\ u \notin x \wedge \neg y \notin x))$. This implies Set Foundation – note the 0-axiom and Extensionality imply $(x \neq 0 \to \exists u\ u \in x)$.

(2). We show $\varphi(x) := \forall y \in \mathrm{tc}(x)\ y \subseteq \mathrm{tc}(x)$ by $\Delta_0(L_0)$-Induction, so we assume $\forall z \in x\ \varphi(z)$. Let $u \in y \in \mathrm{tc}(x)$. By the tc-axiom we have two cases. If $y \in x$, then $u \in \mathrm{tc}(x)$ follows by the tc-axiom noting $y \subseteq \mathrm{tc}(y)$. Otherwise, there is $z \in x$ with $y \in \mathrm{tc}(z)$. Then $y \subseteq \mathrm{tc}(z)$ by induction, so $u \in \mathrm{tc}(z)$; then $u \in \mathrm{tc}(x)$ by the tc-axiom.

For the second assertion it suffices to show $\psi(x, y) := (x \subseteq \mathrm{tc}(y) \to \mathrm{tc}(x) \subseteq \mathrm{tc}(y))$ since $y \subseteq \mathrm{tc}(y)$. Using $\Delta_0(L_0)$-Induction we can assume $\forall z \in x\ \psi(z, y)$. Suppose $x \subseteq \mathrm{tc}(y)$ and let $u \in \mathrm{tc}(x)$. We claim $u \in \mathrm{tc}(y)$. This is obvious if $u \in x$. Otherwise, $u \in \mathrm{tc}(z)$ for some $z \in x$. By $x \subseteq \mathrm{tc}(y)$ we get $z \in \mathrm{tc}(y)$, and hence $z \subseteq \mathrm{tc}(y)$ as we already proved $\varphi(y)$. By induction, $\mathrm{tc}(z) \subseteq \mathrm{tc}(y)$ and $u \in \mathrm{tc}(y)$ follows.

For the third assertion, note $\mathrm{tc}(x) \subseteq \mathrm{tc}(\mathrm{tc}(x))$ is clear, and $\mathrm{tc}(\mathrm{tc}(x)) \subseteq \mathrm{tc}(x)$ follows from $\psi(\mathrm{tc}(x), x)$.

(3). The first assertion follows from the third of (2). For the second assertion, assume $x \subseteq y$ and $e : y \preccurlyeq z$. Then $e' := \{w \in e : \exists u \in \mathrm{tc}(x) \exists v \in \mathrm{tc}(z)\ w = \langle u, v\rangle\}$ exists by $\Delta_0(L_0)$-Separation and $e' : x \preccurlyeq z$ follows noting $\mathrm{tc}(x) \subseteq \mathrm{tc}(y)$ by (2).

The third assertion follows from the second once we show $x \preccurlyeq x$. This is witnessed by the set of all $\langle u, u\rangle$ for $u \in \mathrm{tc}(x)$. This set exists by $\Delta_0(L_0)$-Separation.

(4). This is clear for $x = 0$, so assume $x \neq 0$. We prove ($\to$) by $\Delta_0(L_0)$-Induction, so assume it to hold for all $x' \in x$. Let $u \in \mathrm{tc}(x \odot y)$. By the tc,$\odot$-axioms, $u \in \mathrm{tc}(x' \odot y)$ for some $x' \in x$ or $u \in x \odot y$. In the first case our claim follows by induction noting $\mathrm{tc}(x') \subseteq \mathrm{tc}(x)$ by (2). In the second case, $u = u' \odot y$ for some $u' \in x$ by the $\odot$-axiom.

Conversely, we first show $(u \in \mathrm{tc}(y) \to u \in \mathrm{tc}(x \odot y))$. By $\Delta_0(L_0)$-Induction we can assume this holds for all $z \in x$. Assume $u \in \mathrm{tc}(y)$ and let $z \in x$ be arbitrary. Then $u \in \mathrm{tc}(z \odot y)$ by induction. But $z \odot y \in x \odot y$, so $\mathrm{tc}(z \odot y) \subseteq \mathrm{tc}(x \odot y)$ by (2).

Finally, we show $(u \in \mathrm{tc}(x) \to u \odot y \in \mathrm{tc}(x \odot y))$ and assume this for all $z \in x$. Let $u \in \mathrm{tc}(x)$. If $u \in x$, then $u \odot y \in x \odot y \subseteq \mathrm{tc}(x \odot y)$. Otherwise $u \in \mathrm{tc}(z)$ for some $z \in x$. Then $u \odot y \in \mathrm{tc}(z \odot y)$ by induction; but $\mathrm{tc}(z \odot y) \subseteq \mathrm{tc}(x \odot y)$ by $z \odot y \in x \odot y$ and (2).

(5). The second assertion follows from the first. To verify the first, first suppose $x = 0$. Then $x \odot y = y$ by the $\odot$-axiom, so we have to show $y \odot^{-1} y = 0$. This follows from $y \notin \mathrm{tc}(y)$, proved by an easy $\Delta_0(L_0)$-Induction.

Now suppose $x \neq 0$. We can assume that (5) holds for all $z \in x$. Then

$$\begin{aligned} u \in (x \odot y) \odot^{-1} y &\leftrightarrow \exists x' \in x \odot y\ \ u = x' \odot^{-1} y \\ &\leftrightarrow \exists z \in x\ \ u = (z \odot y) \odot^{-1} y \\ &\leftrightarrow \exists z \in x\ \ u = z \\ &\leftrightarrow u \in x. \end{aligned}$$

The first equivalence follows from $\odot^{-1}$-axiom using $y \in \mathrm{tc}(x \odot y)$: from $x \neq 0$ one can infer $0 \in \mathrm{tc}(x)$ and then (4) implies $0 \odot y = y \in \mathrm{tc}(x \odot y)$. The second follows from the $\odot$-axiom and $x \neq 0$, and the third by induction. □

Definition 8. A *#-term* is a L_0-term build from $\#, \odot, 1$ and variables.

Notationally, we shall omit right-associative parentheses in #-terms, e.g. we write $1 \odot x \# y \odot z$ instead $1 \odot (x \# (y \odot z))$.

Example 9. Write $2^\odot := 1 \odot 1$, $3^\odot := 1 \odot 1 \odot 1$, etc.. Then K_0 describes an embedding from $\langle x, y\rangle$ into the #-term $t_{\text{pair}}(x,y) := 4^\odot \odot (x \odot y)$.

Proof. The union of the following three sets witnesses $\langle x, y\rangle \preccurlyeq t_{\text{pair}}(x,y)$:

$$\{\langle\{x,y\}, 3^\odot \odot (x \odot y)\rangle, \langle\{x\}, 2^\odot \odot (x \odot y)\rangle, \langle x, 1 \odot (x \odot y)\rangle, \langle y, x \odot y\rangle\},$$
$$\{\langle u, u\rangle : u \in \text{tc}(y)\},$$
$$\{\langle u, u \odot y\rangle : u \in \text{tc}(x) \land u \notin \text{tc}(y)\}.$$

This embedding can be described in K_0 by a $\Delta_0(L_0)$-formula. □

3.2 Adding $\in$-bounded functions

Write $\exists^{\leq 1} x\varphi(x, \vec{x})$ for $\forall x x'(\varphi(x, \vec{x}) \land \varphi(x', \vec{x}) \to x = x')$.

Definition 10. Let L_0^+ be the language obained from L_0 by adding relation symbol $R(\vec{x})$ for every $\Delta_0(L_0)$-formula $\varphi(\vec{x})$, and a function symbol $f(\vec{x})$ for every $\Delta_0(L_0)$-formula $\varphi(y, \vec{x})$ such that K_0 proves $\exists! y \in t(\vec{x})\varphi(y, \vec{x})$ for some L_0-term $t(\vec{x})$; by this we mean

$$\exists y \in t(\bar{x})\varphi(y, \vec{x}) \land \exists^{\leq 1} y \ \varphi(y, \vec{x}).$$

The theory K_0^+ has language L_0^+ and is obtained from K_0 by adding for every relation symbol $R(\vec{x})$ in L_0^+ as above the *defining axiom* $(R(\vec{x}) \leftrightarrow \varphi(\vec{x}))$, and for every function symbol $f(\vec{x})$ in L_0^+ as above the *defining axiom* $\varphi(f(\vec{x}), \vec{x})$.

Proposition 11. *K_0^+ is a conservative extension of K_0. Every $\Delta_0(L_0^+)$-formula is K_0^+-provably equivalent to a $\Delta_0(L_0)$-formula. In particular, K_0^+ proves $\Delta_0(L_0^+)$-Separation and $\Delta_0(L_0^+)$-Induction.*

We omit the proof. The language L_0^+ and the theory K_0^+ are introduced mainly for notational convenience. Interesting functions often do not have $\in$-bounded values.

Examples 12. The relation symbols $\text{ispair}(x)$ and $x \subseteq y$ are in L_0^+ with defining axioms $\exists u, v \in \text{tc}(x)\ x = \langle u, v\rangle$ and $\forall u \in x\ u \in y$. We have in L_0^+ symbols $\pi_i(x)$ for i equal to 1 or 2 such that K_0^+ proves $\pi_i(\langle x_1, x_2\rangle) = x_i$. Further, we have $\text{cond}_S(x, z, u, v)$ for S equal to $\in$ or $=$, and a binary $w\text{'}x$ such that K_0^+ proves

$$\text{cond}_S(x,y,u,v) = \begin{cases} x & \text{if } uSv \\ y & \text{otherwise.} \end{cases}$$
$$w\text{'}x = \begin{cases} y & \text{if } y \text{ is unique with } \langle x, y\rangle \in w, \\ 0 & \text{otherwise.} \end{cases}$$

Write $\text{tc}^+(x)$ for $\text{tc}(\{x\})$. The following lemma formalizes the graph isomorphism for # mentioned in the beginning of Section 2. The function $\#\text{"}(u, y)$ is auxiliary to formulate the defining equation for $\sigma_{x,y}(u, v)$.

Lemma 13. *There are function symbols* $\#"(u,y), \sigma_{x,y}(u,v), \pi_{1,x,y}(w)$ *and* $\pi_{2,x,y}(w)$ *in* L_0^+ *such that* K_0^+ *proves*

$$\begin{aligned} \#"(u,y) &= \{z : \exists u' \in u\ z = u'\#y\}, \\ \sigma_{x,y}(u,v) &= \begin{cases} v \odot \#"(u,y) & \text{if } u \in \text{tc}^+(x), v \in \text{tc}^+(y), \\ 0 & \text{else.} \end{cases} \end{aligned}$$

Moreover, K_0^+ *proves* $\sigma_{x,y}$ *is injective on arguments* $u \in \text{tc}^+(x), v \in \text{tc}^+(y)$*, every* $w \in \text{tc}^+(x\#y)$ *has* $\sigma_{x,y}$*-preimage* $(\pi_{1,x,y}(w), \pi_{2,x,y}(w))$ *and for all* $u, u' \in \text{tc}^+(x), v, v' \in \text{tc}^+(y)$

$$(\sigma_{x,y}(u,v) \in \sigma_{x,y}(u',v') \leftrightarrow (u = u' \wedge v \in v') \vee (u \in u' \wedge v = y \wedge v' = 0)).$$

Proof. The functions $\#"(u,y), \sigma_{x,y}(u,v)$ have obvious defining axioms. Concerning bounding terms, note K_0^+ proves $x\#y = y \odot \#"(x,y)$, and hence, using Lemma 7, $\#"(u,y) \in \text{tc}^+(u\#y)$ and $\sigma_{x,y}(u,v) \in \text{tc}^+(x\#y)$. Next, prove in K_0^+

$$\begin{aligned} v \neq 0 &\rightarrow (z \in \sigma_{x,y}(u,v) \leftrightarrow \exists v' \in v\ z = \sigma_{x,y}(u,v')), \\ v = 0 &\rightarrow (z \in \sigma_{x,y}(u,v) \leftrightarrow \exists u' \in u\ z = \sigma_{x,y}(u',y)). \end{aligned} \tag{2}$$

For the latter note $z \in \sigma_{x,y}(u,0)$ is equivalent to $z \in \#"(u,y)$, hence to $z = u'\#y$ for some $u' \in u$; but $\sigma_{x,y}(u',y) = u'\#y$.

It now suffices to verify in K_0^+ the claimed bijectivity: then $\pi_{1,x,y}(w), \pi_{2,x,y}(w)$ are easily constructed and the last claim follows from (2).

Let $u, \tilde{u}, \ldots$ range over $\text{tc}^+(x)$ and $v, \tilde{v}, \ldots$ range over $\text{tc}^+(y)$. We claim $\sigma_{x,y}(u,v) = \sigma_{x,y}(\tilde{u},\tilde{v})$ implies $u = \tilde{u}, v = \tilde{v}$. By Lemma 7 (5) it suffices to show it implies $u = \tilde{u}$. Assume otherwise. By $\Delta_0(L_0^+)$-Foundation (cf. Remark 5) choose u $\in$-minimal such that there exist $\tilde{u}, v, \tilde{v}$ with $\sigma_{x,y}(u,v) = \sigma_{x,y}(\tilde{u},\tilde{v}), u = \tilde{u}$; then choose $\tilde{u}$ $\in$-minimal such that there are $v, \tilde{v}$ with this property, and so on for $v, \tilde{v}$.

Case $v \neq 0$. Then there is $v' \in v$ such that $\sigma_{x,y}(u,v') \in \sigma_{x,y}(\tilde{u},\tilde{v})$. If $\tilde{v} \neq 0$, then $\sigma_{x,y}(u,v') = \sigma_{x,y}(\tilde{u},v'')$ for some $v'' \in \tilde{v}$, and this contradicts the choice of v. If $\tilde{v} = 0$, then $\sigma_{x,y}(u,v') = \sigma_{x,y}(u',y)$ for some $u' \in \tilde{u}$, and this contradicts the choice of $\tilde{u}$.

Case $v = 0$. If $\tilde{v} = 0$, then $\{\sigma_{x,y}(u',y) : u' \in u\} = \{\sigma_{x,y}(u'',y) : u' \in \tilde{u}\}$, so for each $u' \in u$ there is $u'' \in \tilde{u}$ such that $\sigma_{x,y}(u',y) = \sigma_{x,y}(u'',y)$, so then $u' = u''$ by choice of u; thus $u \subseteq \tilde{u}$; similarly $\tilde{u} \subseteq u$, a contradiction. If $\tilde{v} \neq 0$, then for each $u' \in u$ there is $v' \in \tilde{v}$ such that $\sigma_{x,y}(u',y) = \sigma_{x,y}(\tilde{u},v')$, so $u' = \tilde{u}$ by choice of u. Thus $u = 0$ or $u = \{\tilde{u}\}$; the latter is impossible by choice of u, so $u = 0$; then $\sigma_{x,y}(\tilde{u},\tilde{v}) = \sigma_{x,y}(u,v) = \sigma_{x,y}(0,0) = 0$, so $\tilde{v} = 0$, a contradiction.

To see surjectivity, let $w \in \text{tc}^+(x\#y)$. If $w = x\#y$, put $u := x, v := y$. Otherwise $w \in \text{tc}(x\#y) = \text{tc}(y \odot \#"(x,y))$. By Lemma 7 (4) we have two cases. If $w = v' \odot \#"(x,y)$ for some $v' \in y$, put $u := x, v := v'$. If if $w \in \text{tc}(\#"(x,y))$, then $w \in \text{tc}^+(x'\#y)$ for some $x' \in x$ and, using $\Delta_0(L_0^+)$-Induction on x, we find $u \in \text{tc}^+(x') \subseteq \text{tc}^+(x)$ and $v \in \text{tc}^+(y)$ with $w = \sigma_{x',y}(u,v) = \sigma_{x,y}(u,v)$ (note $\sigma_{x,y}(u,v)$ does not depend on x). □

We can now formalize a non-uniform version of Lemma 2. Note it implies that K_0 proves transitivity:

$$(z \preccurlyeq x \wedge x \preccurlyeq y \rightarrow z \preccurlyeq y). \tag{3}$$

Lemma 14. *For all #-terms $t(x,\vec{x})$ the theory K_0 proves*

$$(z \preccurlyeq t(x,\vec{x}) \wedge x \preccurlyeq y \rightarrow z \preccurlyeq t(y,\vec{x})). \tag{4}$$

Moreover, for all $\Delta_0(L_0)$-formulas $\varepsilon_0, \varepsilon_1$ there is a $\Delta_0(L_0)$-formula ε_2 such that K_0 proves

$$(\varepsilon_0(\cdot,x,z,\vec{x}) : z \preccurlyeq t(x,\vec{x}) \wedge \varepsilon_1(\cdot,x,y,\vec{x}) : x \preccurlyeq y \rightarrow \varepsilon_2(\cdot,x,y,z,\vec{x}) : z \preccurlyeq t(y,\vec{x})).$$

Proof. The second statement follows by inspection of the proof of the first. We only prove the first statement. By Proposition 11 it suffices to show that (4) is provable in K_0^+. We proceed by induction on t.

If $t(x,\vec{x})$ equals 1 or a variable distinct from x, then there is nothing to show. If $t(x,\vec{x})$ equals x then we have to show (3) in K_0^+: assume $e : z \preccurlyeq x$ and $e' : x \preccurlyeq y$. Then

$$f := \{\langle u,w\rangle \in \mathrm{tc}(z) \times \mathrm{tc}(y) : \exists v \in \mathrm{tc}(x)(\langle u,v\rangle \in e \wedge \langle v,w\rangle \in e')\}$$

exists by $\Delta_0(L_0^+)$-Separation. We claim $f : z \preccurlyeq y$. It is easy to see that $\langle u,w\rangle, \langle u',w\rangle \in f$ implies $u = u'$. Assume $u \in u' \in \mathrm{tc}(x)$ and $\langle u',w'\rangle \in f$. Choose v' such that $\langle u',v'\rangle \in e, \langle v',w'\rangle \in e'$. Then there is $v \in \mathrm{tc}(v')$ such that $\langle u,v\rangle \in e$. It now suffices to show that, generally, for all v,v',w' we have

$$(v \in \mathrm{tc}(v') \wedge \langle v',w'\rangle \in e' \rightarrow \exists w \in \mathrm{tc}(w')\ \langle v,w\rangle \in e').$$

This is clear if $v \in v'$. Otherwise, $v \in \mathrm{tc}(v'')$ for some $v'' \in v'$. Then choose $w'' \in \mathrm{tc}(w')$ such that $\langle v'',w''\rangle \in e'$. Using $\Delta_0(L_0^+)$-Induction on v', we get $w \in \mathrm{tc}(w'')$ such that $\langle v,w\rangle \in e'$. Then $w \in \mathrm{tc}(w')$ by Lemma 7 (2), as claimed.

As a preparation for the induction step we show in K_0^+:

$$(x \preccurlyeq x' \wedge y \preccurlyeq y' \rightarrow x \odot y \preccurlyeq x' \odot y' \wedge x\#y \preccurlyeq x'\#y'). \tag{5}$$

Assume $e : x \preccurlyeq x'$ and $e' : y \preccurlyeq y'$. By $\Delta_0(L_0^+)$-Separation the set

$$\{\langle u,v\rangle \in \mathrm{tc}(x\odot y)\times\mathrm{tc}(x'\odot y') : \exists u' \in \mathrm{tc}(x)\exists v' \in \mathrm{tc}(x')(u = u'\odot y\wedge\langle u',v'\rangle \in e\wedge v = v'\odot y')\},$$

exists. We leave it to the reader to check that its union with e' witnesses $x \odot y \preccurlyeq x' \odot y'$.

Observe $e_+ := e \cup \{\langle x,x'\rangle\} : \{x\} \preccurlyeq \{x'\}$ and $e'_+ := e' \cup \{\langle y,y'\rangle\} : \{y\} \preccurlyeq \{y'\}$. Let f be the set containing the pairs $\langle \sigma_{x,y}(a,b), \sigma_{x',y'}(a',b')\rangle$ such that $\langle a,a'\rangle \in e_+$ and $\langle b,b'\rangle \in e'_+$. The set f exists by $\Delta_0(L_0^+)$-Separation. Using the previous lemma it is straightforward to check that $f : x\#y \preccurlyeq x'\#y'$.

Now, the induction step is easy. Assume first that $t(x,\vec{x}) = t_1(x,\vec{x}) \odot t_2(x,\vec{x})$. By Lemma 7 (4) we know $\forall z\ z \preccurlyeq z$ and hence $t_1(x,\vec{x}) \preccurlyeq t_1(x,\vec{x})$. The induction hypothesis yields a proof of $t_1(x,\vec{x}) \preccurlyeq t_1(y,\vec{x})$ from $x \preccurlyeq y$. Analogously, we get a proof of $t_2(x,\vec{x}) \preccurlyeq t_2(y,\vec{x})$ from $x \preccurlyeq y$. Applying (5) we get

$$(x \preccurlyeq y \rightarrow t_1(x,\vec{x}) \odot t_2(x,\vec{x}) \preccurlyeq t_2(y,\vec{x}) \odot t_2(y,\vec{x})),$$

that is, $(x \preccurlyeq y \rightarrow t(x,\vec{x}) \preccurlyeq t(y,\vec{x}))$. This together with (3) implies (4).

The case $t(x,\vec{x}) = t_1(x,\vec{x})\#t_2(x,\vec{x})$ is analogous. □

Lemma 15. *The theory K_0 describes an embedding of $x \times y$ into some #-term $t_\times(x,y)$. In particular, there is is a $\Delta_0(L_0)$-formula $\varepsilon_{\text{emb}}(u,x,y)$ such that K_0 proves*

$$(e : x \preccurlyeq y \rightarrow \varepsilon_{\text{emb}}(\cdot,x,y) : e \preccurlyeq t_\times(x,y)).$$

Proof. It suffices to prove this for K_0^+ and $\Delta_0(L_0^+)$-formulas. We first show that the second statement follows from the first. Let $\varepsilon_\times(u,x,y)$ be a $\Delta_0(L_0^+)$-formula describing in K_0^+ an embedding of $x \times y$ into $t_\times(x,y)$. Then $\varepsilon_\times(\cdot, \mathrm{tc}(x), \mathrm{tc}(y))$ describes (in K_0^+) an embedding of $\mathrm{tc}(x) \times \mathrm{tc}(y)$ into $t_\times(\mathrm{tc}(x), \mathrm{tc}(y))$. The $\Delta_0(L_0^+)$-formula $\pi_1(u) = \pi_2(u)$ describes an embedding of $\mathrm{tc}(x)$ into x (cf. proof of Lemma 7 (3)). Lemma 14 gives a $\Delta_0(L_0^+)$-formula $\varepsilon(u,x,y)$ describing an embedding of $\mathrm{tc}(x) \times \mathrm{tc}(y)$ into $t_\times(x,y)$. But $e : x \preccurlyeq y$ implies $e \subseteq \mathrm{tc}(x) \times \mathrm{tc}(y)$, so this formula also describes an embedding of e into $t_\times(x,y)$.

We now prove the first statement. The required formula $\varepsilon_\times(u,x,y)$ implements the following informal procedure on input z, z', x, y where $z := \pi_1(u), z' := \pi_2(u)$. In the description of this procedure we understand that whenever a "check" is carried out then the computation is aborted and the procedure rejects or accepts depending on whether the check failed or not. For example, line 3 is reached only if $z \notin \mathrm{tc}(x)$.

The #-term $t_\times(x,y)$ is $(x\#y) \odot w$ for the value of w in line 13, i.e.

$$t_\times(x,y) = (x\#y) \odot (x\#y) \odot (x\#x) \odot y \odot x \odot y \odot x.$$

Input: z, z', x, y

1. **if** $z \in \mathrm{tc}(x)$ **then** check $z' = z$
2. $w \leftarrow x$
3. **if** $z \in \mathrm{tc}(y)$ **then** check $z' = z \odot w$
4. $w \leftarrow y \odot w$
5. **guess** $u \in x, v \in y$
6. **if** $z = \{u\}$ **then** check $z' = u \odot w$
7. $w \leftarrow x \odot w$
8. **if** $z = \{u, v\}, v \in \mathrm{tc}(x)$ **then**
check $(z' = \sigma_{x,x}(u,v) \odot w \vee z' = \sigma_{x,x}(v,u) \odot w)$
9. $w \leftarrow (x\#x) \odot w$
10. **if** $z = \{u, v\}, v \notin \mathrm{tc}(x)$ **then** check $z' = \sigma_{x,y}(u,v) \odot w$
11. $w \leftarrow (x\#y) \odot w$
12. **if** $z = \{\{u, v\}, \{v\}\}$ **then** check $z' = \sigma_{x,y}(u,v) \odot w$
13. reject

□

Lemma 16. *For each L_0^+-term $s(\vec{x})$ the theory K_0 describes an embedding of $s(\vec{x})$ into some #-term $s^\#(\vec{x})$.*

Proof. This follows by an induction on $s(\vec{x})$ using Lemma 14 once we verify it for the base case that $s(\vec{x})$ is a function symbol in L_0^+.

In this case there is an L_0-term $r(\vec{x})$ such that K_1 proves $s(x) \in r(\vec{x})$. By Lemma 7 the formula $\pi_1(u) = \pi_2(u)$ describes an embedding of $s(\vec{x})$ into $r(\vec{x})$. By transitivity of $\preccurlyeq$ (Lemma 14, cf. (3)), it suffices to verify the lemma for L_0-terms $r(\vec{x})$.

This follows by an induction on $r(\vec{x})$ using Lemma 14 once we verify it for the base case that $r(\vec{x})$ is a function symbol in L_0. The only non-trivial case is crossproduct $\times$ and this case is handeled by the previous lemma. □

3.3 K_1 and a replacement scheme

Definition 17. The theory K_1 is obtained from K_0^+ by adding $\Delta_0(L_0^+)$-$\preccurlyeq$-*Replacement*, i.e. for $t(\vec{x})$ a #-term and $\varphi(u, v, \vec{x}), \varepsilon(\tilde{u}, u, v, \vec{x}) \in \Delta_0(L_0^+)$

$$\begin{aligned}&(\forall u \in x\ \exists^{\leq 1} v\ \varphi(u, v, \vec{x}) \wedge \forall u \in x\ \exists v\ (\varphi(u, v, \vec{x}) \wedge \varepsilon(\cdot, u, v, \vec{x}) : v \preccurlyeq t(\vec{x}))\\ &\quad \rightarrow \exists V\ V = \{v : \exists u \in x\ \varphi(u, v, \vec{x})\}).\end{aligned}$$

The set V witnessing the conclusion can be embedded into some #-term and, moreover, and K_1 describes such an embedding. Mpre precisely, we have the following.

Lemma 18. *For every #-term $t(\vec{x})$ and all $\Delta_0(L_0^+)$-formulas $\varphi(u, v, \vec{x}), \varepsilon(\tilde{u}, u, v, \vec{x})$ there is a $\Delta_0(L_0^+)$-formula $\varepsilon'(\tilde{u}, V, x, \vec{x})$ such that K_1 proves*

$$\begin{aligned}&(\forall u \in x\ \exists^{\leq 1} v\ \varphi(u, v, \vec{x}) \wedge \forall u \in x\ \exists v\ (\varphi(u, v, \vec{x}) \wedge \varepsilon(\cdot, u, v, \vec{x}) : v \preccurlyeq t(\vec{x}))\\ &\quad \rightarrow \exists V (V = \{v : \exists u \in x\ \varphi(u, v, \vec{x})\} \wedge \varepsilon^*(\cdot, V, x, \vec{x}) : V \preccurlyeq x \# t(\vec{x})).\end{aligned} \tag{6}$$

Proof. For notational simplicity we suppress any mention of the side-variables $\vec{x}$. Assume the antecedens of (6) and let $V = \{v : \exists u \in x\ \varphi(u, v)\}$. The formula $\varepsilon^*(\tilde{u}, V, x)$ implements the following informal procedure on input V, x and $z := \pi_1(\tilde{u}), z' := \pi_2(\tilde{u})$.

Input: V, x, z, z'

1. **guess** $u \in x, v \in V$

2. check $\varphi(u, v) \wedge z \in \mathrm{tc}^+(v)$

3. **if** $z = v$ **then** check $z' = \sigma_{x,t}(u, t)$

4. **guess** $z'' \in \mathrm{tc}(t)$

5. check $z' = \sigma_{x,t}(u, z'') \wedge \varepsilon(\langle z, z'' \rangle, u, v)$

□

It is not hard to generalize the above lemma by allowing the bounding term $t(\vec{x})$ to depend also on u and furthermore allow a tuple $\vec{u}$ instead of a single u.

For tuples of variables $\vec{u} = u_0 \cdots u_n$ and $\vec{y} = y_0 \cdots y_n$ let $\forall \vec{u} \in \vec{y}\ \varphi$ abbreviate the formula $\forall \vec{u} (\bigwedge_{i \leq n} u_i \in y_i \rightarrow \varphi)$; the notation $\exists \vec{u} \in \vec{y}\ \varphi$ is similarly explained.

Lemma 19. *For every #-term $t(\vec{u},\vec{x})$ and all $\Delta_0(L_0^+)$-formulas $\varphi(\vec{u},v,\vec{x}),\varepsilon(\tilde{u},\vec{u},v,\vec{x})$ there are a #-term $t^*(\vec{y},\vec{x})$ and a $\Delta_0(L_0^+)$-formula $\varepsilon^*(\tilde{u},V,\vec{y},\vec{x})$ such that K_1 proves*

$$\begin{aligned}(\forall \vec{u}\in\vec{y}\ \exists^{\leq 1}v\ \varphi(\vec{u},v,\vec{x}) \wedge \forall\vec{u}\in\vec{y}\ \exists v\ (\varphi(\vec{u},v,\vec{x})\wedge\varepsilon(\cdot,\vec{u},v,\vec{x}): v\preccurlyeq t(\vec{u},\vec{x}))\\ \to \exists V(V=\{v:\exists\vec{u}\in\vec{y}\ \varphi(\vec{u},v,\vec{x})\}\wedge\varepsilon^*(\cdot,V,\vec{y},\vec{x}):V\preccurlyeq t^*(\vec{y},\vec{x})).\end{aligned} \tag{7}$$

Proof. Assume the antecedens of (7). Note $u_i\in y_i$ implies that $\pi_1(\tilde{u})=\pi_2(\tilde{u})$ describes in K_0 an embedding of u_i into y_i. By Lemma (14) we can replace $\varepsilon(\cdot,\vec{u},v,\vec{x}):v\preccurlyeq t(\vec{u},\vec{x})$ by $\varepsilon'(\cdot,\vec{u},v,\vec{x}):v\preccurlyeq t'(\vec{y},\vec{x})$ for some suitable #-term t' and $\Delta_0(L_0^+)$-formula ε'. Now replace $\forall\vec{u}\in\vec{y}$ by $\forall u\in s(\vec{y})$ for $s(\vec{y})=y_0\times\cdots\times y_n$ and the formulas φ,ε' by suitable formulas $\varphi'(u,v,\vec{x}),\varepsilon''(\tilde{u},u,v,\vec{x})$. In the succedent of (6) we can replace $\varepsilon^*(\cdot,V,s(\vec{y}),\vec{x}):V\preccurlyeq s(\vec{y})\#t'(\vec{y},\vec{x})$ by $\varepsilon^{**}(\cdot,V,s(\vec{y}),\vec{x}):V\preccurlyeq t^*(\vec{y},\vec{y})$ for suitable #-term t^* and $\Delta_0(L_0^+)$-formula ε^{**}: this follows from Lemmas 16 and 14. □

3.4 The language L_{crsf}

The definition of L_{crsf} is relative to a given theory K. We do not show this dependence on K notationally. We assume that K is a theory in the language L_0^+ extending K_1.

Roughly speaking L_{crsf} is obtained from K by adding Σ_1-defined function symbols $f(\vec{x})$. We consider only special Σ_1-definitions. Of course, we require that the existential quantifier is bounded in the sense that its witness is embeddable into a #-term $t(\vec{x})$ and, moreover, this embedding is uniformly given by a $\Delta_0(L_0^+)$-formula $\varepsilon(\tilde{u},v,\vec{x})$. We require the witness v to be uniquely determined by a $\Delta_0(L_0^+)$-formula $\varphi(v,\vec{x})$. Intuitively, this formula says "v is a computation of the value of f on input $\vec{x}$". The "output" function is a very simple function that *e*xtracts the value $e(v)=f(\vec{x})$ from the computation v.

Definition 20. A *good definition (in K)* is a tuple $(\varphi(v,\vec{x}),\varepsilon(\tilde{u},v,\vec{x}),e(v),t(\vec{x}))$ where φ,ε are $\Delta_0(L_0^+)$-formulas, $e(v)$ is a L_0^+-term and $t(\vec{x})$ is a #-term such that K proves

$$\begin{aligned}&\text{(Witness Existence)} && \exists v\ \varphi(v,\vec{x}),\\ &\text{(Witness Uniqueness)} && \exists^{\leq 1}v\ \varphi(v,\vec{x}),\\ &\text{(Witness Embedding)} && (\varphi(v,\vec{x})\to\varepsilon(\cdot,v,\vec{x}):v\preccurlyeq t(\vec{x})).\end{aligned}$$

The theory $K(L_{\mathrm{crsf}})$ is obtained from K by adding for every such good definition a function symbol $f(\vec{x})$ along with the *defining axiom*

$$\forall\vec{x}\exists v(\varphi(v,\vec{x})\wedge f(\vec{x})=e(v)); \tag{8}$$

we then talk of a good definition *of* f. As indicated, L_{crsf} denotes the language of $K(L_{\mathrm{crsf}})$.

The notation $K(L_{\mathrm{crsf}})$ is somewhat misleading because the schemes are not adopted for the language L_{crsf}, e.g. by definition $K(L_{\mathrm{crsf}})$ has $\Delta_0(L_0^+)$-, not $\Delta_0(L_{\mathrm{crsf}})$-Separation.

Lemma 21. *For every $f(\vec{x})$ in L_0^+ there exists a good definition $(\varphi(v,\vec{x}),\varepsilon(\cdot,v,\vec{x}),e(v),t(\vec{x}))$ such that $K(L_0^+)$ proves (8).*

Proof. For $\varphi(v, \vec{x})$ choose $v = f(\vec{x})$, for $e(v)$ choose v, and for ε, t choose according Lemma 16. $\square$

Notationally, for every $f(\vec{x})$ in L_{crsf} we let $(\varphi_f(v, \vec{x}), \varepsilon_f(\tilde{u}, v, \vec{x}), e_f(v), t_f(\vec{x}))$ denote a good definition of f (in K).

Lemma 22. *For every function symbol $f(\vec{x})$ in L_{crsf}, the theory $K(L_{\mathrm{crsf}})$ describes an embedding of $f(\vec{x})$ into some #-term $t(\vec{x})$.*

Proof. By Lemma 16 $K(L_{\mathrm{crsf}})$ describes an embedding of $e_f(v)$ into $e_f^{\#}(v)$. By Lemma 14 we find a $\Delta_0(L_0^+)$-formula ε such that $K(L_{\mathrm{crsf}})$ proves

$$(\varphi(v, \vec{x}) \to \varepsilon(\cdot, v, \vec{x}) : f(\vec{x}) \preccurlyeq e_f^{\#}(t_f(\vec{x})))$$

Now note that L_{crsf} contains a function symbol $g(\vec{x})$ such that $K(L_{\mathrm{crsf}})$ proves $\varphi_f(g(\vec{x}), \vec{x})$. A good definition of g is obtained from the one of f replacing $e_f(v)$ by v. Then $\varepsilon(\cdot, g(\vec{x}), \vec{x})$ is a $\Delta_0(L_{\mathrm{crsf}})$-formula describing in $K(L_{\mathrm{crsf}})$ an embedding of $f(\vec{x})$ into $e_f^{\#}(t_f(\vec{x}))$. $\square$

Remark 23. The proof shows that the embedding $f(\vec{x}) \preccurlyeq t(\vec{x})$ is indeed very simple: it can be described by a $\Delta_0(L_0^+)$-formula with parameters $\vec{x}$ plus one extra parameter v which is $\Delta_0(L_0^+)$-definable from $\vec{x}$.

Proposition 24. *For all n-ary function symbols $h(x_0, \ldots, x_{n-1})$ in L_{crsf} and m-ary function symbols $g_i(y_0, \ldots, y_{m-1}), i < n$, in L_{crsf} there is an m-ary function symbol $f(\vec{y})$ in L_{crsf} such that $K(L_{\mathrm{crsf}})$ proves*

$$f(\vec{y}) = h(g_0(\vec{y}), \ldots, g_{n-1}(\vec{y})).$$

Proof. For notational simplicity we show this only for the case $n = 1$. So let $h(x)$ and $g(\vec{y})$ be function symbols in L_{crsf}. Set

$$\begin{aligned} \varphi_f(v, \vec{y}) &:= \exists v_h, v_g \in \mathrm{tc}(v)\ \psi(v, v_g, v_h, \vec{y}), \\ \psi(v, v_g, v_h, \vec{y}) &:= (v = \langle v_h, v_g \rangle \wedge \varphi_g(v_g, \vec{y}) \wedge \varphi_h(v_h, e_g(v_g))), \\ e_f(v) &:= e_h(\pi_1(v)). \end{aligned}$$

We claim $(\varphi_f, \varepsilon_f, e_f, t_f)$ is a good definition for suitable $\varepsilon_f(\tilde{u}, v, \vec{y}), t_f(\vec{y})$. More precisley, we need a #-term $t_f(\vec{y})$ such that K proves

$$(\psi(v, v_g, v_h, \vec{y}) \to \varepsilon(\cdot, v, \vec{y}) : \langle v_h, v_g \rangle \preccurlyeq t_f(\vec{y})).$$

Argue in K. Assume $\psi(v, v_g, v_h, \vec{y})$. By Lemmas 16 and 14, we have $e_g(v_g) \preccurlyeq e_g^{\#}(t_g(\vec{y}))$. By Lemma 14, $v_h \preccurlyeq t_h(e_g(v_g))$ implies $v_h \preccurlyeq t_h(e_g^{\#}(t_g(\vec{y})))$. By Example 9,

$$t_f(\vec{y}) := t_{\mathrm{pair}}(t_h(e_g^{\#}(t_g(\vec{y}))), t_g(\vec{y}))$$

is as desired. It is easy to find a formula ε_f as desired. $\square$

3.5 Elimination lemma

Recall that $K(L_{\mathrm{crsf}})$ does not have axioms schemes of K_1 in the language L_{crsf} but only in the language L_{crsf}. In order to prove that L_{crsf} proves equations for functions obtained by Cobham Recursion we shall need axioms schemes like Separation for formulas in in the language L_{crsf}.

In a usual development of KP (e.g. [4, Chapter I]) one proves that one can conservatively add Σ_1-definable function symbols in the sense that the enhanced theory proves the axiom schemes like Δ_0-Separation for formulas mentioning the new symbols. This is done in two steps: one first shows that occurrences of Σ_1-defined symbols can be eliminated in a way that transforms Δ_0-formulas into Δ_1-formulas; second one proves Δ_1-Separation in KP.

The following lemma gives an analogous argument for $K(L_{\mathrm{crsf}})$. The elimination is only partially successful but sufficient for our purpose.

Lemma 25 (Elimination)**.** *Let $\Delta_0^{L_0^+}(L_{\mathrm{crsf}})$ denote the set of $\Delta_0(L_{\mathrm{crsf}})$-formulas all of whose $\in$-bounding terms are L_0^+-terms. For every such formula $\varphi(\vec{x})$ there are $\Delta_0(L_0^+)$-formulas $\varphi^0(\vec{x}, V)$, $\varphi^1(\vec{x}, V)$, $\varphi^2(\tilde{u}, \vec{x}, V)$ and a #-term $t_\varphi(\vec{x})$ such that $K(L_{\mathrm{crsf}})$ proves*

$$\begin{aligned}&\exists^{\leq 1} V\ \varphi^1(\vec{x}, V),\\ &\exists V(\varphi^1(\vec{x}, V) \wedge \varphi^2(\cdot, \vec{x}, V) : V \preccurlyeq t_\varphi(\vec{x})),\\ &(\varphi^1(\vec{x}, V) \to (\varphi(\vec{x}) \leftrightarrow \varphi^0(\vec{x}, V))).\end{aligned} \tag{9}$$

Proof. This is proved by induction on $\varphi(\vec{x})$. The base case for atomic $\varphi(\vec{x})$ is the most involved and proved by induction on the number of occurrences of symbols from $L_{\mathrm{crsf}} \setminus L_0^+$. If this number is 0, there is not much to be shown. Otherwise one can write

$$\varphi(\vec{x}) = \psi(\vec{x}, f(\vec{s}(\vec{x})))$$

for atomic $\psi(\vec{x}, y)$ with y actually occurring freely in ψ, a symbol $f(\vec{z}) \in L_{\mathrm{crsf}} \setminus L_0^+$ and a tuple of L_0^+-terms $\vec{s}(\vec{x})$.

Again, let $(\varphi_f(v, \vec{z}), e_f(v), t_f(\vec{z}))$ be a good definition of $f(\vec{z})$. By Lemmas 16 and 14 we have #-terms $e_f^{\#}(v), \vec{s}^{\#}(\vec{x})$ and $\Delta_0(L_0^+)$-formulas $\varepsilon_0, \varepsilon_1$ such that $K(L_{\mathrm{crsf}})$ proves

$$(\varphi_f(v, \vec{s}(\vec{x})) \to \varepsilon_0(\cdot, v, \vec{x}) : v \preccurlyeq t_f(\vec{s}^{\#}(\vec{x})) \wedge \varepsilon_1(\cdot, v, \vec{x}) : e_f(v) \preccurlyeq e_f^{\#}(t_f(\vec{s}^{\#}(\vec{x}))).$$

By induction, $K(L_{\mathrm{crsf}})$ proves

$$\begin{aligned}&\exists^{\leq 1} W\ \psi^1(\vec{x}, e_f(v), W)\\ &\exists W(\psi^1(\vec{x}, e_f(v), W) \wedge \psi^2(\cdot, e_f(v), W) : W \preccurlyeq t_\psi(\vec{x}, e_f(v))),\\ &(\psi^1(\vec{x}, e_f(v), W) \to (\psi(\vec{x}, e_f(v)) \leftrightarrow \psi^0(\vec{x}, e_f(v), W))).\end{aligned}$$

Define $\varphi^1(\vec{x}, V) := \exists W, v \in \mathrm{tc}(V)\ \chi(\vec{x}, W, v)$ where

$$\chi(\vec{x}, V, W, v) := (V = \langle W, v\rangle \wedge \psi^1(\vec{x}, e_f(v), W) \wedge \varphi_f(v, \vec{s}(\vec{x}))).$$

Lemma 14 gives a $\Delta_0(L_0^+)$-formula ε_2 such that $K(L_{\mathrm{crsf}})$ proves

$$(\chi(\vec{x}, V, W, v) \to \varepsilon_2(\cdot, \vec{x}, V) : W \preccurlyeq t_\psi(\vec{x}, e_f^\#(\vec{x})) \wedge \varepsilon_0(\cdot, \pi_2(V), \vec{x}) : v \preccurlyeq t_f(\vec{s}^\#(\vec{x}))).$$

Using Example 9 we define

$$t_\varphi(\vec{x}) := t_{\mathrm{pair}}(t_\psi(\vec{x}, e_f^\#(\vec{x})), t_f(\vec{s}^\#(\vec{x})))$$

and get a $\Delta_0(L_0^+)$-formula φ^2 that $K(L_{\mathrm{crsf}})$ proves

$$(\chi(\vec{x}, V, W, v) \to \varphi^2(\cdot, \vec{x}, V) : V \preccurlyeq t_\varphi(\vec{x})).$$

Finally, we set

$$\varphi^0(\vec{x}, V) := \psi^0(\vec{x}, e_f(\pi_2(V)), \pi_1(V))$$

It is easy to verify (9). This completes the proof for the case that $\varphi(\vec{x})$ is atomic.

The induction step is easy if $\varphi(\vec{x})$ is a negation, or a conjunction. We treat the case that $\varphi(\vec{x}) = \forall u \in s(\vec{x})\ \psi(u, \vec{x})$ for some L_0^+-*term* $s(\vec{x})$. By induction there are $\psi^0, \psi^1, \psi^2, t_\psi$ as desired for $\psi(y, \vec{x})$. In particular, $K(L_{\mathrm{crsf}})$ proves

$$\forall u \in s(\vec{x}) \exists^{\leq 1} W \psi^1(u, \vec{x}, W) \wedge \forall u \in s(\vec{x}) \exists W (\psi^1(u, \vec{x}, W) \wedge \psi^2(\cdot, u, \vec{x}, W) : W \preccurlyeq t_\psi(u, \vec{x})).$$

By Lemma 19, $K(L_{\mathrm{crsf}})$ proves

$$V = \{W : \exists u \in s(\vec{x}) \psi^1(u, \vec{x}, W)\}. \tag{10}$$

exists and describes an embedding into $s(\vec{x}) \# t_\psi(s(\vec{x}), \vec{x})$. By Lemma 14, $K(L_{\mathrm{crsf}})$ also describes an embedding of V into the $\#$-term

$$t_\varphi(\vec{x}) := s^\#(\vec{x}) \# t_\psi(s^\#(\vec{x}), \vec{x}).$$

where $s^\#(\vec{x})$ is chosen according Lemma 16. For $\varphi^2(\tilde{u}, \vec{x}, V)$ we choose a formula describing such an embedding in $K(L_{\mathrm{crsf}})$. Let $\varphi^1(\vec{x}, V)$ be the $\Delta_0(L_0^+)$-formula (10) and let $\varphi^0(\vec{x}, V)$ be the $\Delta_0(L_0^+)$-formula

$$\forall y \in s(\vec{x}) \exists W \in V\ (\psi^0(y, \vec{x}, W) \wedge \psi^1(y, \vec{x}, W)).$$

It is straightforward to verify that $K(L_{\mathrm{crsf}})$ proves (9). □

Corollary 26 (Separation)**.** *The theory $K(L_{\mathrm{crsf}})$ proves $\Delta_0^{L_0^+}(L_{\mathrm{crsf}})$-Separation.*

Proof. Let $\varphi(u, \vec{x})$ be a $\Delta_0^{L_0^+}(L_{\mathrm{crsf}})$-formula. Choose $\varphi^0, \varphi^1, \varphi^2$ and t_φ according to the previous lemma. Then $K(L_{\mathrm{crsf}})$ proves

$$\forall u \in x \exists^{\leq 1} V \varphi^1(u, \vec{x}, V) \wedge \forall u \in x \exists V (\varphi^1(u, \vec{x}, V) \wedge \varphi^2(\cdot, u, \vec{x}, V) : V \preccurlyeq t_\varphi(u, \vec{x})).$$

By Lemma 19, $K(L_{\mathrm{crsf}})$ proves that the set

$$\tilde{V} := \{V : \exists u \in x\ \varphi^1(u, \vec{x}, V)\}$$

exists. By $\Delta_0(L_0^+)$-Separation, $K(L_{\mathrm{crsf}})$ proves that

$$\{u \in x : \exists V \in \tilde{V} (\varphi^0(u, \vec{x}, V) \wedge \varphi^1(u, \vec{x}, V))\}$$

exists. This set equals $\{u \in x : \varphi(u, \vec{x})\}$, provably in $K(L_{\mathrm{crsf}})$. □

3.6 Adding replacement terms

The following theorem is crucial. It provides a formalized version of the first two statements of Theorem 3 showing, more generally, that $K(L_{\mathrm{crsf}})$ can handle comprehension terms coming from Replacement. Similar terms are basic computation steps in Sazonov's term calculus [14] and in the logic of Blass et al. [9].

For $\vec{x} = x_0 \cdots x_{n-1}$ let $\vec{x} \in x$ stand for $\bigwedge_{i<n} x_i \in x$.

Theorem 27 (Replacement). *Let $\varphi(x, \vec{y}, \vec{x})$ be a $\Delta_0^{L_0^+}(L_{\mathrm{crsf}})$-formula, $t(\vec{y}, \vec{x})$ a #-term and $g(\vec{y}, \vec{x})$ a function symbol in L_{crsf}. Then there exists a function symbol $f(\vec{y}, \vec{x}, x)$ in L_{crsf} such that $K(L_{\mathrm{crsf}})$ proves*

$$f(x, \vec{y}) = \{g(\vec{y}, \vec{x}) : \varphi(x, \vec{y}, \vec{x}) \wedge \vec{x} \in x\}.$$

Proof. For notational simplicity we assume $\vec{y}$ is the empty tuple. It is sufficient to prove the theorem for g such that $K(L_{\mathrm{crsf}})$ proves $g(\vec{x}) \neq 0$. We first show that $K(L_{\mathrm{crsf}})$ proves the existence of

$$z := \{g(\vec{x}) : \varphi(x, \vec{x}) \wedge \vec{x} \in x\}$$

and furthermore describes an embedding of z into $t_1(x)$ for a suitable #-term t_1.

We choose a good definition $(\varphi_g, \varepsilon_g, e_g, t_g)$ of g in $K(L_{\mathrm{crsf}})$ and we choose $\varphi^0, \varphi^1, \varphi^2, t_\varphi$ for φ according to the Elimination Lemma 25. Argue in $K(L_{\mathrm{crsf}})$. For every $\vec{x} \in x$ there is a unique w such that

$$\exists y, v_g, V \in \mathrm{tc}(w)\ \psi(w, y, v_g, V, x, \vec{x}),$$

where $\psi(w, y, v_g, V, x, \vec{x})$ is the following $\Delta_0(L_0^+)$-formula:

$$(w = \langle\langle y, v_g\rangle, V\rangle \wedge \varphi^1(x, \vec{x}, V) \wedge \varphi_g(v_g, \vec{x}) \wedge \left(\begin{array}{c} (y = e_g(v_g) \wedge \varphi^0(x, \vec{x}, V)) \\ \vee\ (y = 0 \wedge \neg\varphi^0(x, \vec{x}, V)) \end{array}\right)).$$

For a suitable $\Delta_0(L_0^+)$-formula ε and #-term $t_2(x, \vec{x})$ we get $\varepsilon(\cdot, w, x, \vec{x}) : w \preccurlyeq t_2(x, \vec{x})$ for this w. Then Lemma 19 gives the set

$$W = \{w : \exists \vec{x} \in x \exists y, v_g, V \in \mathrm{tc}(w)\ \psi(w, y, v_g, V, x, \vec{x})\},$$

and $\varepsilon'(\cdot, W, x) : W \preccurlyeq t_3(x)$ for a suitable $\Delta_0(L_0^+)$-formula ε' and #-term $t_3(x)$. Then

$$z = \{y \in \mathrm{tc}(W) : \exists w \in W (y = \pi_1(\pi_1(w)) \wedge y \neq 0)\}$$

exists by $\Delta_0(L_0^+)$-Separation. Note $\varepsilon'(\cdot, W, x) : z \preccurlyeq t_3(x)$ since z is a subset of $\mathrm{tc}(W)$. Now one gets the following good definition $(\varphi_f, \varepsilon_f, e_f, t_f)$ of $f(x)$.

For $\varphi_f(v, x)$ take the $\Delta_0(L_0^+)$-formula

$$\exists W, z \in \mathrm{tc}(v)\ (v = \langle W, z\rangle \wedge \left(\begin{array}{c} W = \{w : \exists \vec{x} \in x \exists y, v_g, V \in \mathrm{tc}(w)\ \psi(w, y, v_g, V, x, \vec{x})\} \\ \wedge\ z = \{y \in \mathrm{tc}(W) : \exists w \in W\ y = \pi_1(\pi_1(w))\} \end{array}\right)).$$

For $e_f(v)$ take $\pi_2(v)$. Set $t_f(x) := t_{\mathrm{pair}}(t_3(x), t_3(x))$ and choose ε_f such that $\varepsilon_f(\cdot, v, x) : v \preccurlyeq t_f(x)$ for the unique v with $\varphi_f(v, x)$. □

We can now introduce some useful notation:

Examples 28. Let $f(x, \vec{x})$ be a function symbol in L_{crsf}. Then L_{crsf} contains function symbols $f\text{"}(x, \vec{x})$ and $f \upharpoonright (\vec{x}, x)$ such that $K(L_{\mathrm{crsf}})$ proves

$$\begin{aligned} f\text{"}(x, \vec{x}) &= \{f(u, \vec{x}) : u \in x\}, \\ f \upharpoonright (x, \vec{x}) &= \{\langle u, f(u, \vec{x})\rangle : u \in x\}. \end{aligned}$$

Furthermore, L_{crsf} contains $x \cap y, x \setminus y$ and proves the usual defining equations for them.

Corollary 29. *For every $\Delta_0^{L_0^+}(L_{\mathrm{crsf}})$-formula $\varepsilon(\tilde{u}, x, y, \vec{x})$ there is a function symbol $\tau(u, x, y, \vec{x})$ in L_{crsf} such that $K(L_{\mathrm{crsf}})$ proves*

$$(\varepsilon(\cdot, x, y, \vec{x}) : x \preccurlyeq y \rightarrow \tau(\cdot, x, y, \vec{x}) : x \preccurlyeq y).$$

Proof. Choose τ in L_{crsf} such that $K(L_{\mathrm{crsf}})$ proves

$$\tau(u, x, y, \vec{x}) = \{v \in \mathrm{tc}(y) : \varepsilon(\langle u, v\rangle, x, y, \vec{x})\}.$$

The previous theorem implies that such τ exists. $\square$

This implies a uniform version of Lemma 14:

Corollary 30. *For all function symbols $\tau_0(u, x, z, \vec{x}), \tau_1(u, x, y, \vec{x})$ in L_{crsf} and #-terms $t(x, \vec{x})$ there is a function symbol $\sigma(u, x, y, z\vec{x})$ in L_{crsf} such that $K(L_{\mathrm{crsf}})$ proves*

$$(\tau_0(\cdot, x, z, \vec{x}) : z \preccurlyeq t(x, \vec{x}) \wedge \tau_1(\cdot, x, y, \vec{x}) : x \preccurlyeq y \rightarrow \sigma(\cdot, x, y, z, \vec{x}) : z \preccurlyeq t(y, \vec{x})).$$

3.7 K_2 and an induction scheme

We now add a weak form of Induction to our theory K_1. This will suffice to prove the recursive equations coming from definitions by Cobham Recursion.

Definition 31. The theory K_2 is obtained from K_1 by adding *uniformly bounded unique $\Sigma_1(L_0^+)$-Induction*, that is, for $t(\vec{x})$ a #-term and $\varphi(u, v, \vec{x}), \varepsilon(\tilde{u}, u, v, \vec{x}) \in \Delta_0(L_0^+)$

$$\forall u\ \exists^{\leq 1} v\ \varphi(u, v, \vec{x}) \wedge \forall x \Big(\forall u \in x \exists v\ \varphi^{\varepsilon,t}(u, v, \vec{x}) \rightarrow \exists v\ \varphi^{\varepsilon,t}(x, v, \vec{x})\Big) \rightarrow \exists v\ \varphi^{\varepsilon,t}(x, v, \vec{x}),$$

where $\varphi^{\varepsilon,t}(u, v, \vec{x})$ abbreviates the $\Delta_0(L_0^+)$-formula

$$\varphi(u, v, \vec{x}) \wedge \varepsilon(\cdot, u, v, \vec{x}) : v \preccurlyeq t(u, \vec{x}).$$

Proposition 32. *For all function symbols $g(x, z, \vec{x})$ and $\tau(u, x, \vec{x})$ in L_{crsf} and all #-terms $t(x, \vec{x})$ there is a function symbol $f(x, \vec{x})$ in L_{crsf} such that $K_2(L_{\mathrm{crsf}})$ proves*

$$f(x, \vec{x}) = \begin{cases} g(\vec{x}, x, f\text{"}(x, \vec{x})) & \text{if } \tau(\cdot, x, \vec{x}) : g(x, f\text{"}(x, \vec{x}), \vec{x}) \preccurlyeq t(x, \vec{x}), \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

Proof. (Sketch) Let g, τ, t be as stated. For notational simplicity we assume $\vec{x}$ is the empty tuple. We are looking for a good definition $(\varphi_f, \varepsilon_f, e_f, t_f)$ of f such that $K_2(L_{\rm crsf})$ proves (11).

We intend to let $\varphi_f(v, x, \vec{x})$ say that v encodes the course of values of f, namely the set of all pairs $\langle u, f(u)\rangle, u \in {\rm tc}^+(x)$. We use an encoding of this set from [6, Section 3.5] which is tailor-made to easily define an embedding. Roughly said, the idea is to let $\varphi_f(v, x)$ express that $v = f^*(x)$ where

$$f^*(x) = \{0, \langle x, f(x)\rangle\} \odot \{f^*(u) : u \in x\}.$$

More precisely, we shall first define a formula $\psi_f(w_x, x)$ that expresses $w_x = f^*(x)$. This formula is going to use $L_{\rm crsf}$-symbols g, τ, among others. In order to get a $\Delta_(L_0^+)$-formula $\varphi_f(v, x)$ we apply the Elimination Lemma 25, the witness v now comprising w_x plus the parameter needed for the elimination.

We now construct the formula $\psi_f(w_x, x)$. To this end we need some L_0^+-functions for analyzing a code of the form $f^*(x)$. The functions given in [6, Section 3.5] are defined by Cobham recursion which we need to sidestep here.

Intuitively, $f^*(x)$ encodes an *entry* for *arg*ument x and *val*ue $f(x)$ situated above other entries inside the set *bot*tom $\{f^*(u) : u \in x\}$. Formally, first observe that for a set w there exists at most one triple of sets (u, v, z) such that $w = \{0, \langle u, v\rangle\} \odot z$. Then L_0^+ contains function symbols $\mathit{arg}(w)$, $\mathit{val}(w)$, $\mathit{bot}(w)$ and a relation symbol $\mathit{Star}(w)$ such that K_0^+ proves

$$\begin{aligned}
&w = \{0, \langle u, v\rangle\} \odot z \to \mathit{arg}(w) = u \wedge \mathit{val}(w) = v \wedge \mathit{bot}(w) = z,\\
&w \neq \{0, \langle u, v\rangle\} \odot z \to \mathit{arg}(w) = \mathit{val}(w) = \mathit{bot}(w) = 0,\\
&\mathit{Star}(w) \leftrightarrow w = \{0, \langle \mathit{arg}(w), \mathit{val}(w)\rangle\} \odot \mathit{bot}(w).
\end{aligned}$$

This is easy to see because $\odot^{-1} \in L_0$. For example, $\mathit{arg}(w)$ has defining axiom

$$\begin{aligned}
&(\exists z \in {\rm tc}(w) \exists u, v \in {\rm tc}(w \odot^{-1} z)\ (w = \{0, \langle u, v\rangle\} \odot z \wedge y = u)\\
&\vee\ (y = 0 \wedge \neg \exists z \in {\rm tc}(w) \exists u, v \in {\rm tc}(w \odot^{-1} z)\ w = \{0, \langle u, v\rangle\} \odot z)).
\end{aligned}$$

The relation symbol $\mathit{Entry}(u, w)$ is meant to express, given w of the form $f^*(x)$, that $u \in {\rm tc}^+(w)$ codes an entry. This is not the same as having the form $\mathit{Star}(u)$: the argument x and value $f(x)$ in the entry are arbitrary sets whose transitive closure may well contain sets u of the form $\mathit{Star}(u)$. We let $\mathit{Entry}(u, w)$ have defining axiom

$$u \in {\rm tc}^+(w) \wedge \mathit{Star}(u) \wedge \neg \exists u' \in {\rm tc}^+(w)(\mathit{Star}(u') \wedge u \in {\rm tc}(u') \wedge \mathit{bot}(u') \in {\rm tc}^+(u)).$$

The relation symbol $\mathit{CofV}(w_x, x)$ has defining axiom

$$\begin{aligned}
&\mathit{Entry}(w_x, w_x) \wedge \mathit{arg}(w_x) = x \wedge \forall w' \in {\rm tc}(w_x) \exists w \in {\rm tc}^+(w_x)\\
&\quad(\ \mathit{Entry}(w, w_x) \wedge \exists u' \in {\rm tc}^+(w \odot^{-1} \mathit{bot}(w))\ w' = u' \odot \mathit{bot}(w)\\
&\qquad \wedge\ \forall u \in \mathit{bot}(w)(\mathit{Entry}(u, w_x) \wedge \mathit{arg}(u) \in \mathit{arg}(w))\\
&\qquad \wedge\ \forall u \in \mathit{arg}(w) \exists w'' \in \mathit{bot}(w)(\mathit{Entry}(w'', w_x) \wedge \mathit{arg}(w'') = u))).
\end{aligned}$$

The following property of $\mathit{CofV}(w_x, x)$ is straightforwardly verified in K_0^+:

$$(\mathit{CofV}(w_x, x) \wedge \mathit{Entry}(w', w_x) \rightarrow \mathit{CofV}(w', \mathit{arg}(w'))). \tag{12}$$

Intuitively, $\mathit{CofV}(w_x, x)$ says w_x is of the form $f^*(x)$ for *some* function f; the final two lines ensure that $\mathit{bot}(w) = \{f^*(u) : u \in \mathit{arg}(w)\}$.

The next formula $\mathit{Rec}_{g,\tau}(w_x, \vec{x})$ stipulates that this function f satisfies (11). Recall the notation form Example 28. Then $\mathit{Rec}_{g,\tau}(w_x, \vec{x})$ is the formula

$$\begin{aligned}&\forall w \in \mathrm{tc}^+(w_x)\ (\mathit{Entry}(w, w_x) \rightarrow \\ &\quad ((\xi \wedge g(\mathit{arg}(w), \mathit{val}\,''(\mathit{bot}(w))) = \mathit{val}(w)) \vee (\neg\xi \wedge \mathit{val}(w) = 0))),\end{aligned}$$

where $\xi(w)$ expresses case 1 in (11). More precisely, $\xi(w)$ is the formula

$$\tau(\cdot, \mathit{arg}(w)) : g(\mathit{arg}(w), \mathit{val}\,''(\mathit{bot}(w))) \preccurlyeq t(\mathit{arg}(w)).$$

Note $\mathit{Rec}_{g,\tau}(w_x, \vec{x})$ is an L_{crsf}-formula. Along with symbols from L_0^+ it contains the L_{crsf}-symbols τ, g and $\mathit{val}\,''$. We finally define

$$\psi_f(w_x, x) \quad := \quad (\mathit{CofV}(w_x, x) \wedge \mathit{Rec}_{g,\tau}(w_x, \vec{x})).$$

Observe that all $\in$-bounding terms of this formula are in L_0^+, that is, ψ_f is a $\Delta_0^{L_0^+}(L_{\mathrm{crsf}})$-formula. Thus, we can apply the Elimination Lemma 25 to it and get $\Delta_0(L_0^+)$-formulas $\psi_f^0(w_x, x, V)$, $\psi_f^1(w_x, x, V)$, $\psi_f^2(\tilde{u}, w_x, x, V)$ and a #-term $t_{\psi_f}(w_x, x)$. Write $\psi_f^{0,1} := (\psi_f^0 \wedge \psi_f^1)$ and define

$$\begin{aligned}\varphi_f(v, x) \quad &:= \quad \exists w_x, V \in \mathrm{tc}(v)\ (v = \langle w_x, V\rangle \wedge \psi_f^{0,1}(w_x, x, V)), \\ e_f(v) \quad &:= \quad \mathit{val}(\pi_1(v)),\end{aligned}$$

The work still to be done is to show $(\varphi_f, \varepsilon_f, e_f, t_f)$ is a good definition of for suitable ε_f, t_f, and furthermore to prove (11). This is omitted in this lecture. □

References

[1] T. Arai. Predicatively computable functions on sets. arXiv:1204.5582 [math.LO], 2014.

[2] A. Atserias, M. Müller. Partially definable forcing and bounded arithmetic. Archive for Mathematical Logic 54 (1): 1-33, 2015.

[3] J. Avigad. Saturated models of universal theories Annals of Pure and Applied Logic 118:219-234, 2002.

[4] J. Barwise. Admissible Sets and Structures: An Approach to Definability Theory. Perspectives in Logic, Volume 7, 1975.

[5] A. Beckmann, S. R. Buss, Sy Friedman. Safe recursive set functions. Submitted for publication, 2012.

[6] A. Beckmann, S. R. Buss, S.-D. Friedman, M. Müller and N. Thapen. Cobham recursive set functions. Manuscript, 2015.

[7] A. Beckmann, S. R. Buss, S.-D. Friedman, M. Müller and N. Thapen. Cobham recursive set functions and Boolean circuits. Manuscript, 2015.

[8] S. Bellantoni and S. Cook. A new recursion-theoretic characterization of polytime functions. Computational Complexity 2:97-110, 1992.

[9] A. Blass, Y. Gurevich and S. Shelah, Choiceless polynomial time. Annals of Pure and Applied Logic 100:141187, 1999.

[10] S. R. Buss. First-order proof theory of arithmetic. In S. R. Buss (ed.), Handbook of Proof Theory, Studies in Logic 137, Elsevier, Ch. II, pp. 79-148, 1998.

[11] A. Cobham. The intrinsic computational difficulty of functions. In Y. Bar-Hillel (ed.), Logic, Methodology and Philosophy of Science, Proceedings of the Second International Congress, Amsterdam, North-Holland, 1965.

[12] H.-D. Ebbinghaus and J. Flum. Finite Model Theory, Springer, 1999.

[13] M. Rathjen. A proof-theoretic characterization of the primitive recursive set functions. Journal of Symbolic Logic 57: 954-969, 1992.

[14] V. Y. Sazonov. On bounded set theory. Logic and Scientific Methods, M. L. D. C. et al. (eds.), Synthese Library Volume 259, Kluwer Academic, pp. 85-103, 1997.