JAIST Repository
https://dspace.jaist.ac.jp/

Title	The strength of SCT soundness
Author(s)	Frittaion, Emanuele; Pelupessy, Florian; Steila, Silvia; Yokoyama, Keita
Citation	Journal of Logic and Computation, 28(6): 1217-1242
Issue Date	2018-02-22
Type	Journal Article
Text version	author
URL	http://hdl.handle.net/10119/16083
Rights	(c) The Author, 2018. Published by Oxford University Press. This is a pre-copy-editing, author-produced PDF of an article accepted for publication in JOURNAL OF LOGIC AND COMPUTATION following peer review. The definitive publisher-authenticated version JOURNAL OF LOGIC AND COMPUTATION (2018) 28(6): 1217-1242 is available online at: http://dx.doi.org/10.1093/logcom/exy003
Description	

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

The strength of SCT soundness

Emanuele Frittaion[1], Florian Pelupessy[2], Silvia Steila[3], and Keita Yokoyama[4]

[1] Department of Mathematics, FCUL - Universidade de Lisboa, Portugal
[1] *emanuelefrittaion@gmail.com*
[2]Mathematical Institute, Tohoku University, Japan
[2] *florian.pelupessy@operamail.com*
[3]Institute of Computer Science, University of Bern, Switzerland
[3] *steila@inf.unibe.ch*
[4]School of Information Science, Japan Advanced Institute of Science and Technology, Japan
[4] *y-keita@jaist.ac.jp*

Contents

1 Introduction **2**
1.1 Reverse mathematics . . . 3

2 SCT framework **5**
2.1 Syntax . . . 5
2.2 Semantics . . . 6
2.3 Size-change graphs . . . 8
2.4 Péter-Ackermann . . . 9

3 The standard proof requires ACA_0 **10**

4 Lower bound **12**
4.1 Fast Growing Hierarchy . . . 13
4.2 Generalizing Péter-Ackermann . . . 16
4.3 From soundness to $\mathrm{WO}(\omega_3)$ . . . 18

5 Upper bound **19**
5.1 Aboveness . . . 21
5.2 From $\mathrm{WO}(\omega_3)$ to soundness . . . 22
5.3 Upper bound for tail-recursive programs . . . 25

6 Conclusion **27**

Abstract

In this paper we continue the study, from Frittaion, Steila and Yokoyama (2017), on size-change termination in the context of Reverse Mathematics. We analyze the soundness of the SCT method. In particular, we prove that the statement "any program which satisfies the combinatorial condition provided by the SCT criterion is terminating" is equivalent to $\mathrm{WO}(\omega_3)$ over RCA_0.

Keywords: Ramsey's theorem for pairs, Size-change termination, Reverse Mathematics, Soundness, Well-ordering principles.

1 Introduction

Informally, a recursive definition of a function has the SCT property if, in every infinite sequence of calls, there is some infinite sequence of parameter values which is weakly decreasing and strictly decreasing infinitely many times. If the parameter values are well-ordered, as in the case of natural numbers with the natural ordering, there cannot be such a sequence. Thus, the SCT property is a sufficient condition for termination.

The SCT property concerns the semantics of the program. In [13] Lee, Jones and Ben-Amram provided an alternative property, equivalent to being SCT, but which can be statically verified from the definition of the program. Indeed, they proved the following:

Theorem 1.1 (SCT criterion)**.** *Let $\mathcal{G}$ be a description of a program P. Then $\mathcal{G}$ is SCT iff every idempotent $G \in \mathsf{cl}(\mathcal{G})$ has an arc $x \xrightarrow{\downarrow} x$.*

Here, $\mathsf{cl}(\mathcal{G})$ is a set of graphs, which can be extracted directly from the code of the program P. We refer to Section 2.3 for definitions.

The SCT criterion leads us to consider two distinct, although classically equivalent, properties. For convenience of exposition, we use the following terminology. See Section 2 for definitions.

Definition 1.1.

- $\mathcal{G}$ is **MSCT** (Multipath-Size-Change terminating) if $\mathcal{G}$ is SCT.
- $\mathcal{G}$ is **ISCT** (Idempotent-Size-Change terminating) if every idempotent $G \in \mathsf{cl}(\mathcal{G})$ has an arc $x \xrightarrow{\downarrow} x$.

With this terminology at hand, we outline the following three-step argument from [13] to prove the termination of a first order functional program P:

- Verify that P is ISCT;
- Apply the SCT criterion to prove that P is MSCT;
- Derive the termination of P from the fact that "every MSCT program terminates".

Since the Ackermann function is ISCT provably in RCA_0 (see Section 2.4), a natural question arises: in which theory can we carry out the above argument? Specifically, in

which theory can we prove the SCT criterion? Similarly, in which theory can we prove that every MSCT program terminates?

It is clear that this cannot be done in weak theories, such as RCA_0, which do not prove the termination of the Ackermann function. In [6] we studied the strength of the SCT criterion. We proved that the SCT criterion follows from a special instance of Ramsey's theorem for pairs, which turns out to be equivalent to $\mathsf{I}\Sigma^0_2$ over RCA_0. Note that we are considering a consequence of Ramsey's theorem for pairs for any number of colors, which implies $\mathsf{I}\Sigma^0_2$. If we restrict to colorings in two colors the situation changes (e.g. [19]). Indeed, as proved by Patey and the fourth author, Ramsey's theorem for pairs in two colors has the same proof-theoretic strength as $\mathsf{B}\Sigma^0_2$ (namely they have the same Π^0_3-consequences) [14].

In the present paper we focus our attention on the second question, i.e., the soundness of the MSCT principle. Moreover, we investigate in which theory we can prove directly the termination of programs which are ISCT, without applying the SCT criterion. We thus consider the following two soundness statements.

Theorem 1.2 (ISCT (resp. MSCT) Soundness)**.** *Let $\mathcal{G}$ be a safe description of a program P. If $\mathcal{G}$ is ISCT (resp. MSCT) then P is terminating.*

Following standard notation (e.g., [17]), $\mathrm{WO}(\alpha)$ states that the linear ordering α is well-ordered. In this paper we show that over RCA_0,

- ISCT soundness $= \mathrm{WO}(\omega_3) \geq$ MSCT soundness $> \mathrm{WO}(\omega_2)$,

where $\omega_2 = \omega^\omega$ and $\omega_3 = \omega^{\omega^\omega}$.

One direction of the SCT criterion is provable in RCA_0. That is, within RCA_0, every MSCT description is also ISCT. Therefore, provably in RCA_0, ISCT soundness implies MSCT soundness. Moreover, over $\mathsf{RCA}_0 + \mathsf{I}\Sigma^0_2$, where ISCT and MSCT are equivalent notions, we have

- MSCT soundness $= \mathrm{WO}(\omega_3)$.

It is still an open question whether the inequality is strict over RCA_0:

Question 1.2. Is MSCT soundness equivalent to $\mathrm{WO}(\omega_3)$ over RCA_0?

As a consequence of our analysis, we provide ordinal bounds for the termination of first order functional programs which are ISCT and, in particular, a new proof of primitive recursive bounds for the tail-recursive ISCT programs.

1.1 Reverse mathematics

Reverse mathematics is a research program in mathematical logic and foundations of mathematics. We refer to Simpson [17] and Hirschfeldt [10] for a general overview. The

goal is to assess the relative logical strength of theorems from ordinary (non set-theoretic) mathematics, thus making sense of statements like Theorem A is stronger than Theorem B or Theorem A and Theorem B are equivalent. This program is carried out in second-order arithmetic, a formal system for natural numbers and sets of natural numbers, which is expressive enough to accommodate large parts of ordinary mathematics. Given a theorem A, one looks for the minimal subsystem Ξ needed to prove A, that is, A follows from Ξ and all the axioms from Ξ are provable from A over a base system Ξ_0.

The most important subsystems of second-order arithmetic are obtained by restricting comprehension and induction to some class Γ of formulas.

- Γ-Comprehension: $\exists X \forall n (n \in X \leftrightarrow \varphi(n))$, for $\varphi \in \Gamma$
- Γ-Induction: $\varphi(0) \wedge \forall n(\varphi(n) \rightarrow \varphi(n+1)) \rightarrow \forall n \varphi(n)$, for $\varphi \in \Gamma$

In practice, one uses comprehension to define a set and induction to verify that the set thus defined has the required properties.

Let us briefly recall the definition of Σ^0_n formulas. Bounded quantifiers are of the form $\forall x < t$ and $\exists x < t$, with x being a number variable and t a number term. A formula φ is Σ^0_n if it is of the form $\exists x_1 \forall x_2 \ldots Q x_n \theta$, where the x_i's are number variables and all quantifiers in θ are bounded. Similarly, φ is Π^0_n if it is of the form $\forall x_1 \exists x_2 \ldots Q x_n \theta$, with x_i's and θ as above. A formula is arithmetical if it is Σ^0_n for some n. Note that θ can contain set variables.

In this paper we are mainly concerned with the base system $\mathsf{RCA_0}$ (Recursive Comprehension Axiom), the system $\mathsf{ACA_0}$ (Arithmetical Comprehension Axiom), the induction schemes $\mathsf{I}\Sigma^0_n$ (induction for Σ^0_n formulas), and the principles $\mathrm{WO}(\omega_n)$ (ω_n is well-ordered), where n is a standard natural number.

$\mathsf{RCA_0}$ consists of the usual first-order axioms of Peano arithmetic, without induction, plus comprehension and induction, restricted to Δ^0_1 and Σ^0_1 formulas respectively. Roughly, $\mathsf{RCA_0}$ proves that all computable sets exist. $\mathsf{ACA_0}$ is obtained from $\mathsf{RCA_0}$ by adding arithmetical comprehension (comprehension for all arithmetical formulas). Roughly, $\mathsf{ACA_0}$ proves that the Turing jump of every set exists.

The statements $\mathsf{I}\Sigma^0_n$ and $\mathrm{WO}(\omega_n)$ form two intertwined hierarchies below $\mathsf{ACA_0}$:

$$\begin{array}{ccccccc}
\mathsf{ACA_0} & \longrightarrow & \mathsf{I}\Sigma^0_{n+3} & \longrightarrow & \mathsf{I}\Sigma^0_{n+2} & & \\
 & & \downarrow & & \downarrow & & \\
 & & \mathrm{WO}(\omega_{n+3}) & \longrightarrow & \mathrm{WO}(\omega_{n+2}) & \not\longleftarrow & \mathsf{RCA_0}
\end{array}$$

The implications in the picture are strict. It is known that $\mathrm{WO}(\omega^\omega)$ is equivalent, over $\mathsf{RCA_0}$, to the totality of the (relativized) Péter-Ackermann function A^2_f [12]. In this paper we show that $\mathrm{WO}(\omega_3)$ is equivalent to the totality of all A^n_f, where A^n_f is a natural generalization of A^2_f to n arguments.

As mentioned earlier, the original proof of the SCT criterion makes use of Ramsey's theorem for pairs [15]. This theorem states that for every coloring, on the edges of the complete graph on countably many nodes, in k colors, there exists an infinite homogeneous set. I.e., there exists an infinite subset of the nodes such that any two elements in this subset are connected in the same color. Ramsey's theorem for pairs, in symbols RT^2, is a key principle in reverse mathematics (e.g., see [17, 11, 2, 3, 19, 14]).

2 SCT framework

2.1 Syntax

$$
\begin{array}{rll}
x \in \mathrm{Par} & \text{parameter} \\
f \in \mathrm{Fun} & \text{function identifier} \\
f \in \mathrm{Op} & \text{primitive operator} \\
a \in \mathrm{AExp} & \text{arithmetic expression} \\
& ::= \; x \mid x+1 \mid x-1 \mid f(a,\ldots,a) \\
b \in \mathrm{BExp} & \text{boolean expression} \\
& ::= \; x=0 \mid x=1 \mid x<y \mid x \leq y \mid b \wedge b \mid b \vee b \mid \neg b \\
e \in \mathrm{Exp} & \text{expression} \\
& ::= \; a \mid \textbf{ if } \; b \; \textbf{ then } \; e \; \textbf{ else } \; e \\
d \in \mathrm{Def} & \text{function definition} \\
& ::= \; f(x_0,\ldots,x_{n-1}) = e \\
P \in \mathrm{Prog} & \text{program} \\
& ::= \; d_0,\ldots,d_{m-1}
\end{array}
$$

A program P is a list of finitely many defining equations $f(x_0,\ldots,x_{n-1}) = e^f$, where $f \in \mathrm{Fun}$ and e^f is an expression, called the **body** of f. Let $x_0,\ldots,x_{n-1}$ be the **parameters** of f, denoted $\mathrm{Par}(f)$, and let n be the **arity** of f, denoted $\mathrm{arity}(f)$. Function identifiers on the left-hand side of each equation are assumed to be distinct from one another. By $\mathrm{Fun}(P)$ we denote the set of function identifiers occurring in P and by $\mathrm{Op}(P)$ the set of primitive operators occurring in P. We usually suppress the reference to P whenever it is clear from the context. The **entry** function f_0 is the first in the program list. The idea is that P computes the partial function $f_0 : \mathbb{N}^{\mathrm{arity}(f_0)} \to \mathbb{N}$.

Example 2.1. The following program computes the well-known Péter-Ackermann function:

$$
\begin{aligned}
A(x,y) = \; & \textbf{ if } \; x=0 \; \textbf{ then } \; y+1 \\
& \textbf{ else if } \; y=0 \; \textbf{ then } \; A(x-1,1) \\
& \qquad \textbf{ else } \; A(x-1, A(x,y-1))
\end{aligned}
$$

2.2 Semantics

The standard semantics for first order functional programs is *denotational semantics* (see, e.g., Lee, Jones and Ben-Amram [13]). Another possible choice is *operational semantics* (as defined in, e.g., [7]). In our framework we find it natural and convenient to interpret programs as *term rewriting systems.*

Notation: We use $\mathbf{u}, \mathbf{v}$ for tuples of natural numbers and $\mathbf{s}, \mathbf{t}$ for tuples of terms.

In general a term rewriting system is a set of rules, i.e., objects of the form $s \to t$ for s, t terms. We build up our terms by using natural numbers and function symbols (function identifiers and primitive operators). In particular, terms do not contain if-then.

$$t \in \text{Term} ::= u \in \mathbb{N} \mid f(t, \ldots, t)$$

Boolean expressions are decidable and we can think of a boolean expression b with parameters in $x_0, \ldots, x_{n-1}$ as a primitive operator whose intended interpretation is a boolean function $[\![b]\!] : \mathbb{N}^n \to 2$. For instance, we read $b(x_0, \ldots, x_{n-1})$ as $b(x_0, \ldots, x_{n-1}) = 0$. Symbols such as $0, 1, +, -$ have the intended interpretation. We do not distinguish notationally between these symbols and their interpretation, relying on context to distinguish the two. For instance, $x + 1$ is an expression if x is a parameter, or the successor of x if x is a natural number.

Given an expression e with parameters in $\mathbf{x} = x_0, \ldots x_{n-1}$ and a tuple $\mathbf{u} \in \mathbb{N}^n$, we want to evaluate the expression e on $\mathbf{u}$ and return a term $e(\mathbf{u})$ ([1]). We can easily define $e(\mathbf{u})$ by recursion on the construction of e as follows:

- $x_i(\mathbf{u}) = u_i$, $(x_i + 1)(\mathbf{u}) = u_i + 1$, and $(x_i - 1)(\mathbf{u}) = u_i - 1$ if $u_i > 0$, 0 otherwise.
- If $e = f(e_0, \ldots, e_{k-1})$, then $e(\mathbf{u}) = f(e_0(\mathbf{u}), \ldots, e_{k-1}(\mathbf{u}))$.
- If $e =$ **if** b **then** e_0 **else** e_1, then

$$e(\mathbf{u}) = \begin{cases} e_0(\mathbf{u}) & \text{if } [\![b]\!](\mathbf{u}) = 0 \\ e_1(\mathbf{u}) & \text{otherwise.} \end{cases}$$

Definition 2.2 (subterm)**.** Let s and t be terms. We say that s is a subterm of t if $t = f(t_0, \ldots, t_{k-1})$ and one of the following holds:

- $s = t$,
- s is a subterm of t_i.

[1] Ultimately, this comes down to substitute $\mathbf{x}$ with $\mathbf{u}$ in the unique arithmetic subexpression of e which is determined by the boolean tests for $\mathbf{u}$. That is, if $e_0, \ldots, e_{k-1}$ are the maximal arithmetic subexpressions of e, in the sense that they are not proper subexpressions of any arithmetic expression of e, then we have $e(\mathbf{u}) = e_i[\mathbf{x}/\mathbf{u}]$, where i is uniquely determined.

Every subterm of a given term t has a **position** σ. We can use sequences of natural numbers to determine the position of a subterm. For instance, if $t = f(g(2), 4)$, then t has position $\langle\rangle$ and $g(2)$ has position $\langle 0\rangle$. Formally:

Definition 2.3 (position)**.** Let s be a subterm of t and σ be a sequence of natural numbers. We say that the position of s in t is σ (in symbols $\mathrm{pos}(s,t) = \sigma$) if $t = f(t_0, \ldots, t_{k-1})$ and one of the following holds:

- $s = t$ and $\sigma = \langle\rangle$,
- s is a subterm of t_i and $\sigma = \langle i\rangle^\frown \mathrm{pos}(s, t_i)$.

We write $t|_\sigma$ for the unique subterm of t in position σ (if it exists).

If s is a subterm of t with multiple occurrence, i.e., there exist $\sigma_0, \ldots \sigma_{n-1}$ such that $t|_{\sigma_0} = \cdots = t|_{\sigma_{n-1}} = s$, we say that $t|_{\sigma_i}$ is the **leftmost subterm** s **of** t if $\sigma_i \leq_{\mathrm{lex}} \sigma_j$ for any $j < n$, with respect to the standard lexicographical order of finite sequences of natural numbers.

Similarly, every subexpression of a given expression e has a **position** $\tau \in \mathbb{N}^{<\mathbb{N}}$, and we write $e|_\tau$ for the unique subexpression of e in position τ (if it exists). Formally:

Definition 2.4 (position)**.** Let e, e' be expressions and $\tau \in \mathbb{N}^{<\mathbb{N}}$. We say that e' has position τ in e if one of the following holds:

- $e = e'$ and $\tau = \langle\rangle$;
- $e = f(e_0, \ldots, e_{n-1})$, $\tau = \langle i\rangle^\frown \tau'$, and e' has position τ' in e_i;
- $e =$ **if** b **then** e_0 **else** e_1, $\tau = \langle i\rangle^\frown \tau'$ with $i < 2$, and e' has position τ' in e_i.

Fix an an interpretation $[\![\cdot]\!]$ of all primitive operators, that is, for all $f \in \mathrm{Op}$ let

$$[\![f]\!] \colon \mathbb{N}^{\mathrm{arity}(f)} \to \mathbb{N}.$$

We are now ready to define, given a program P, a term rewriting system T_P.

Definition 2.5 (rules and reduction)**.** A **rule** is of the form $f(\mathbf{u}) \to e^f(\mathbf{u})$ for $f \in \mathrm{Fun}$ or $f(\mathbf{u}) \to [\![f]\!](\mathbf{u})$ for $f \in \mathrm{Op}$. A **one-step reduction** $t \to_P s$ is given by replacing the leftmost subterm $f(\mathbf{u})$ of t according to the rule. We write $s = t[f(\mathbf{u})]_\rho$, where ρ is the position of $f(\mathbf{u})$ in t.

Note that $\to_P$ is decidable.

Example 2.6. Suppose we want to compute $A(2,3)$, the value of the Péter-Ackermann function at $(2,3)$. According to the definition we have:

$$A(2,3) \to A(1, A(2,2)) \to A(1, A(1, A(2,1))) \to \ldots$$

Use $\to_P^*$ to denote the reflexive transitive closure of $\to_P$.

Definition 2.7 (state transition)**.** For $f, g \in \mathrm{Fun}$ and $\tau \in \mathbb{N}^{<\mathbb{N}}$, define a **state transition** $(f, \mathbf{u}) \xrightarrow{\tau} (g, \mathbf{v})$ by $e^f|_\tau(\mathbf{u}) = g(\mathbf{s})$ and $s_i \to_P^* v_i$ for all $i < \mathrm{arity}(g)$.

For every subterm s of $e^f(\mathbf{u})$ there exists a unique position τ in e^f such that $s = e^f|_\tau(\mathbf{u})$. We say that $\langle \tau, f, g \rangle$ is a **call** from f to g and write $\tau : f \to g$. It is worth noticing that there are only finitely many τ's and hence finitely many calls $\tau : f \to g$. This apparently obvious fact is essential for the SCT criterion (from ISCT to MSCT) and for the SCT soundness.

We can extend the state transition relation to $(f, \mathbf{u}) \xrightarrow{\tau} (g, \mathbf{t})$ by the same definition.

The relations $\to_P^*$ and $\xrightarrow{\tau}$ are Σ_1^0. In particular, the latter is Σ_1^0 by $\mathsf{B}\Sigma_1^0$.

Definition 2.8 (reduction sequence)**.** A **reduction sequence** of P is a sequence of terms $t_0 \to_P t_1 \to_P t_2 \to_P \ldots$. Write $t \downarrow s$ if there exists a reduction sequence $t = t_0 \to_P t_1 \to_P t_2 \to_P \ldots \to_P t_l = s$.

Remark 2.1. Our definition of reduction is deterministic (at each step there is at most one possible reduction). It easily follows that for every t there exists a unique reduction sequence starting with t.

Lemma 2.2. *Given terms t_0, t_1, t_2, if $t_0 \downarrow t_1$ and $t_0 \downarrow t_2$ then $t_0 \to_P^* t_1 \to_P^* t_2$ or $t_0 \to_P^* t_2 \to_P^* t_1$ (This includes the case $t_1 = t_2$). Additionally if t_1 and t_2 are natural numbers, then $t_1 = t_2$.*

Proof. By induction on the length of the reduction sequences and exploiting the fact that the reduction is deterministic. □

Definition 2.9 (termination)**.** We say that P **terminates** on $\mathbf{u}$ if $f(\mathbf{u}) \downarrow v$ for some natural number v, where f is the entry function of P. We say that P is **terminating** if P terminates on every input. We also write $f(\mathbf{u}) = v$ for $f(\mathbf{u}) \downarrow v$ and $f(\mathbf{u}) \downarrow$ if there exists a v such that $f(\mathbf{u}) \downarrow v$.

2.3 Size-change graphs

We briefly recall the main definitions from [6].

A **size-change graph** $G : f \to g$ is a finite set of constraints between the parameters of f and the parameters of g. Constraints are of the form $x > y$ and $x \geq y$, where $x \in \mathrm{Par}(f)$ and $y \in \mathrm{Par}(g)$. Formally, we represent size-change graphs $G : f \to g$ as bipartite graphs with edges of the form $x \xrightarrow{\downarrow} y$ (a strict arc denoting the constraint $x > y$) and $x \xrightarrow{\Downarrow} y$ (a non-strict arc denoting the constraint $x \geq y$) with $x \in \mathrm{Par}(f)$ and $y \in \mathrm{Par}(g)$. We write $x \to y \in G$ as a shorthand for $x \xrightarrow{\downarrow} y \in G \vee x \xrightarrow{\Downarrow} y \in G$.

To $G : f \to g$ we can associate a transition relation $\xrightarrow{G}$ consisting of state transitions $(f, \mathbf{u}) \xrightarrow{G} (g, \mathbf{v})$ for all $\mathbf{u}, \mathbf{v}$ satisfying the given constraints. Note that $\xrightarrow{G}$ is decidable.

Moreover, given $G_0 : f \to g$ and $G_1 : g \to h$, we can define the **composition** $G_0; G_1 : f \to h$ such that $\xrightarrow{G_0} \circ \xrightarrow{G_1} \subseteq \xrightarrow{G_0;G_1}$.

The composition of two edges $x \xrightarrow{\Downarrow} y$ and $y \xrightarrow{\Downarrow} z$ is the edge $x \xrightarrow{\Downarrow} z$. In all other cases the composition of an edge, from x to y, with an edge from y to z is the edge $x \xrightarrow{\downarrow} z$. The composition $G_0; G_1$ consists of all compositions of edges $x \to y \in G_0$ with edges $y \to z \in G_1$, with the exception of $x \xrightarrow{\Downarrow} z$ if $G_0; G_1$ contains $x \xrightarrow{\downarrow} z$. Formally:

$$\begin{aligned} E = \{x \xrightarrow{\downarrow} z : \exists y \in \mathrm{Par}(g)\ \exists r \in \{\downarrow, \Downarrow\}\ ((x \xrightarrow{\downarrow} y \in G_0 \wedge y \xrightarrow{r} z \in G_1) \\ \vee (x \xrightarrow{r} y \in G_0 \wedge y \xrightarrow{\downarrow} z \in G_1))\} \\ \cup \{x \xrightarrow{\Downarrow} z : \exists y \in \mathrm{Par}(g)(x \xrightarrow{\Downarrow} y \in G_0 \wedge y \xrightarrow{\Downarrow} z \in G_1) \wedge \forall y \in \mathrm{Par}(g) \\ \forall r, r' \in \{\downarrow, \Downarrow\}\ ((x \xrightarrow{r} y \in G_0 \wedge y \xrightarrow{r'} z \in G_1) \implies r = r' = \Downarrow)\}. \end{aligned}$$

A **description** $\mathcal{G}$ of a program P consists of size-change graphs $G_\tau : f \to g$ for any call $\tau : f \to g$ of P. We say that $\mathcal{G}$ is **safe** if $\xrightarrow{\tau} \subseteq \xrightarrow{G_\tau}$ for all calls τ of P.

Definition 2.10. $\mathcal{G}$ is **MSCT** if for every infinite **multipath** $M = G_0, G_1, \ldots$, i.e., an infinite sequence of size-change graphs with $G_i : f_i \to f_{i+1}$, contains an **infinite descent**, i.e., a sequence of the form $x_t \to x_{t+1} \to x_{t+2} \to \ldots \to x_i \to \ldots$ with $x_i \in \mathrm{Par}(f_i)$ such that for all i we have $x_i \to x_{i+1} \in G_i$ and for infinitely many i we have $x_i \xrightarrow{\downarrow} x_{i+1} \in G_i$.

Let $\mathrm{cl}(\mathcal{G})$ denote the closure of $\mathcal{G}$ under composition.

Definition 2.11. $\mathcal{G}$ is **ISCT** if every **idempotent** $G : f \to f$ in $\mathrm{cl}(\mathcal{G})$, i.e., $G^2 = G$, contains a strict arc of the form $x \xrightarrow{\downarrow} x$ for some $x \in \mathrm{Par}(f)$.

2.4 Péter-Ackermann

As Ben-Amram shows in [1], the Péter-Ackermann function is ISCT.

$$\begin{aligned} A(x, y) = \ & \textbf{if} \ \ x = 0 \ \ \textbf{then} \ \ y + 1 \\ & \textbf{else if} \ \ y = 0 \ \ \textbf{then} \ \ \tau_0 : A(x - 1, 1) \\ & \qquad \textbf{else} \ \ \tau_1 : A(x - 1, \tau_2 : A(x, y - 1)) \end{aligned}$$

Note that we have three calls τ_i $(i < 3)$ which are safely described by the following size-change graphs:

$$\begin{tikzcd} x \arrow[r, "\downarrow"] & x \\ y & y \end{tikzcd} \qquad \begin{tikzcd} x \arrow[r, "\Downarrow"] & x \\ y \arrow[r, "\downarrow"] & y \end{tikzcd}$$

$$G_1 : A \to A \qquad\qquad G_2 : A \to A$$

The size-change graph G_1 safely describes both calls $\tau_0 : A(x - 1, 1)$ and $\tau_1 : A(x - 1, A(x, y - 1))$. In particular, notice that in the call τ_1 the parameter value x decreases no

matter what the value of the expression $A(x, y-1)$ is. Finally, the size-change graph G_2 safely describes the call $\tau_2 : A(x, y-1)$.

Actually, we can prove that Péter-Ackermann is MSCT within RCA_0.

Lemma 2.3 (RCA_0)**.** *A is MSCT.*

Proof. Every multipath contains either infinitely many G_1 or cofinitely many G_2. In the first case we have an infinite descent in the parameter x and in the second case we have an infinite descent in the parameter y. □

These remarks highlight that both ISCT Soundness and MSCT Soundness are not provable over RCA_0, as the Péter-Ackermann function is not provably recursive in RCA_0.

3 The standard proof requires ACA_0

We discuss the standard proof of SCT soundness. This section is self-contained.

Theorem 3.1 (Lee, Jones, Ben-Amram)**.** *If P is MSCT then P is terminating.*

Proof sketch. Let $\mathcal{G}$ be a safe description of a program P and suppose that P does not terminate on $\mathbf{u}$. Then there exists an infinite sequence of state transitions $(f, \mathbf{u}) \xrightarrow{\tau_0} (f_1, \mathbf{u_1}) \xrightarrow{\tau_1} (f_2, \mathbf{u_2}) \ldots$. Consider the corresponding multipath in $\mathcal{G}$. As $\mathcal{G}$ is MSCT, there exists an infinite descent. By safety, we have an infinite descending sequence of natural numbers. A contradiction. □

We will show that this proof requires ACA_0. This is in sharp contrast with our proof (see Section 5) that the ISCT soundness, and in particular the MSCT soundness, is provable in $\mathsf{RCA}_0 + \mathrm{WO}(\omega_3)$, which is much weaker than ACA_0. For instance, the ω-model REC, consisting of all recursive sets, satisfies RCA_0 plus full induction, and so does satisfy $\mathrm{WO}(\omega_3)$ (see the relation between induction and well-foundedness of ω_n in Subsection 1.1), but clearly does not validate ACA_0.

The key step in the proof by Lee, Jones, and Ben-Amram is the existence of an infinite state transition sequence, i.e., the existence of an infinite branch in the so called activation tree.

Definition 3.1 (activation tree)**.** Let P be a program with entry function f. Given $\mathbf{u} \in \mathbb{N}^{\mathrm{arity}(f)}$, the **activation tree** $T_P^{\mathbf{u}}$ consists of all finite state transition sequences starting with $(f, \mathbf{u})$, i.e., sequences of the form:

$$(f, \mathbf{u}) \xrightarrow{\tau_0} (f_1, \mathbf{u}_1) \xrightarrow{\tau_2} \ldots \xrightarrow{\tau_{k-1}} (f_k, \mathbf{u}_k).$$

The activation tree is Σ^0_1 and finitely branching. If P is MSCT, then every branch of the tree is finite (since we are considering values in $\mathbb{N}$, they cannot decrease infinitely many times). Therefore, we have:

Proposition 3.2 (RCA_0)**.** *If P is MSCT, then $T_P^{\mathbf{u}}$ has no infinite branches for all $\mathbf{u}$.*

Proof. From the definitions. The reader is encouraged to fill in the details. □

One can show that P terminates iff $T_P^{\mathbf{u}}$ is finite for all $\mathbf{u} \in \mathbb{N}^{\mathrm{arity}(f)}$. With some effort, this can be done in RCA_0.

Proposition 3.3 (RCA_0)**.** *P terminates on $\mathbf{u}$ iff $T_P^{\mathbf{u}}$ is finite.*

Proof. Let $T = T_P^{\mathbf{u}}$. Suppose first that T is finite. By bounded Σ_1^0-comprehension the tree T exists. By Σ_1^0-induction on T we show that for all $\sigma \in T$, if $\mathrm{last}(\sigma) = (g, \mathbf{v})$, then $g(\mathbf{v}) \downarrow$. Write $t \uparrow$ if there is no natural number v such that $t \downarrow v$. Note that if $t \uparrow$ then there exists a subterm $h(\mathbf{s})$, with $h \in \mathrm{Fun}$, and $\mathbf{w} \in \mathbb{N}^n$, such that $\mathbf{s} \downarrow \mathbf{w}$ and $h(\mathbf{w}) \uparrow$. This can be proved by Π_1^0-induction on a given term t such that $t \uparrow$. In fact, the least subterm s of t (in any linear ordering of subterms of t which respects the subterm relation), such that $s \uparrow$, is as desired. Suppose σ is an end-node. If $g(\mathbf{v}) \uparrow$ then $e^g(\mathbf{v}) \uparrow$, so there exists a subterm $h(\mathbf{s})$ as above. Then $(g, \mathbf{v}) \xrightarrow{\tau} (h, \mathbf{w})$ for some τ, so σ is not an end-node. The case when σ is not an end-node can be proved similarly.

In the other direction, suppose that P terminates on $\mathbf{u}$ with witness $f(\mathbf{u}) = t_0 \to_P t_1 \to_P \ldots \to_P t_l = v \in \mathbb{N}$. Let $\mathcal{T}$ be the set of all subterms appearing in the reduction sequence. We claim that if $g(\mathbf{v}) \in \mathcal{T}$ and $(g, \mathbf{v}) \xrightarrow{\tau} (h, \mathbf{w})$, then also $h(\mathbf{w}) \in \mathcal{T}$. Let $g(\mathbf{v})$, with $e^g(\mathbf{v})|_\tau = h(\mathbf{t})$ and $\mathbf{t}_i \to_P^* \mathbf{w}_i$, be given. Take the maximum $i \leq l$ such that $g(\mathbf{v})$ appears in t_i. Then t_{i+1} is obtained by reducing $g(\mathbf{v})$, so $h(\mathbf{t})$ is a subterm of t_{i+1}. Since the program terminates, there exists a tuple of natural numbers $\mathbf{n}$ such that $\mathbf{t}_i \to_P^* \mathbf{n}_i$. By Lemma 2.2, we have $\mathbf{w} = \mathbf{n}$. Since, moreover, the reduction is deterministic, $h(\mathbf{w})$ must be a subterm of t_j for some $i < j \leq l$, hence $h(\mathbf{w}) \in \mathcal{T}$. This proves the claim. Now, given $\sigma \in T$, one can show by induction that every initial segment of σ consists of pairs $(g, \mathbf{v})$ with $g(\mathbf{v}) \in \mathcal{T}$. It easily follows that T is finite. □

Proposition 3.4 (ACA_0)**.** *If P does not terminate on $\mathbf{u}$, then $T_P^{\mathbf{u}}$ has an infinite branch.*

Proof. Suppose that P does not terminate on $\mathbf{u}$. Then $T = T_P^{\mathbf{u}}$ is infinite by Proposition 3.3. Note that the tree T exists within ACA_0. Since T is finitely branching, it has an infinite branch by König's lemma (which is provable in ACA_0). □

Proposition 3.5 (RCA_0)**.** *The statement "If P does not terminate on $\mathbf{u}$, then $T_P^{\mathbf{u}}$ has an infinite branch" implies ACA_0.*

Proof. We argue in RCA_0. By [17, Lemma III.1.3], it is enough to show that the range of any one-to-one function exists. Let $f \colon \mathbb{N} \to \mathbb{N}$ be given. Say that $x \in \mathbb{N}$ is an f-true stage (or simply true) if $f(x) < f(y)$ for all $x < y$. It is well-known that, provably in RCA_0, we can define the range of f from any infinite set of f-true stages (see, e.g, [5]). We show that the set of true stages exists. We may safely assume that 0 is true.

Define P as follows:

$$\begin{aligned} g(x,t,y,z) = \ & \textbf{if} \ \ f(z) < f(y) \ \ \textbf{then} \ \ z \\ & \textbf{else} \ \ g(x+1,\bot,y,g(x+1,\top,x+1,x+2)) \end{aligned}$$

The idea is that $g(x,t,y,z)$ tests the truth of y by seeking the least $v \geq z$ such that $f(v) < f(y)$.

Claim 3.5.1. *If $y \leq x < z$ and $g(x,t,y,z) = v$ then $z \leq v$ and $f(v) < f(y)$.*

Proof. By Π^0_1-induction on the length of a reduction sequence. □

By the claim, P does not terminate on $\mathbf{u} = (0,\top,0,1)$. Let $(g,\mathbf{u}) \to (g,\mathbf{u}_1) \to \ldots (g,\mathbf{u}_x) \to$ be an infinite branch in $T^{\mathbf{u}}_P$, where $\mathbf{u}_x = (x,t_x,y_x,z_x)$. By Σ^0_0-induction it is easy to show that $y_x \leq y_{x+1}$ and $y_x \leq x < z_x$ for all x.

We now show that for all $x \in \mathbb{N}$,

$$x \text{ is true if and only if } t_x = \top,$$

so we can define the set of true stages by Δ^0_1-comprehension (indeed Δ^0_0).

Suppose x is true and $t_x = \bot$. Then $\mathbf{u}_x = (x,\bot,y,g(x,\top,x,x+1))$ and by the claim above we have that $f(z_x) < f(x)$ with $x < z_x$, so x is not true, a contradiction.

Suppose that $t_x = \top$ and x is false. We have that $y_x = x$. Let $v > x$ be least such that $f(v+1) < f(x)$. Consider $\mathbf{u}_v = (v,t,y,z)$. Now $\mathbf{u}_v = (v,\top,v,v+1)$ or $\mathbf{u}_v = (v,\bot,y,v+1)$ with $x \leq y \leq v$. By the minimality of v we have $f(z) < f(y)$ in both cases. Thus there is no state transition from $\mathbf{u}_v$, a contradiction. □

Remark 3.6. Within our syntax, primitive operators do not appear in boolean expressions. We can modify P as follows:

$$\begin{aligned} g(x,t,y,z) &= h(f(y),f(z),z,g(x+1,\bot,y,g(x+1,\top,x+1,x+2))) \\ h(a,b,c,d) &= \ \textbf{if} \ \ b < a \ \ \textbf{then} \ \ c \ \ \textbf{else} \ \ d \end{aligned}$$

This program computes the same function. Observe that this program does not have a safe SCT description.

4 Lower bound

In this section we show that ISCT soundness implies $\mathrm{WO}(\omega_3)$ over RCA_0. To this end, we consider the (relativized) fast growing hierarchy.

4.1 Fast Growing Hierarchy

We formalise ε_0 in $\mathsf{RCA_0}$ as in Definition 2.3 from [16]:

Definition 4.1. The set $\mathcal{E}$ of notations of ordinals $< \varepsilon_0$ and order $<$ on $\mathcal{E}$ is taken as follows:

1. If $\alpha_0 \geq \cdots \geq \alpha_n \in \mathcal{E}$, then $\omega^{\alpha_0} + \cdots + \omega^{\alpha_n} \in \mathcal{E}$.
2. $\omega^{\alpha_0} + \cdots + \omega^{\alpha_n} < \omega^{\beta_0} + \cdots + \omega^{\beta_m}$ if and only if:
   (a) $n < m$ and $\alpha_i = \beta_i$ for all $i \leq n$, or:
   (b) there is $i \leq \min\{n, m\}$ with $\alpha_j = \beta_j$ for all $j < i$ and $\alpha_i < \beta_i$.

We use 0 to denote the empty sum, $0 < \alpha$ for all $\alpha \neq 0$, $1 = \omega^0$, $n = \overbrace{1 + \cdots + 1}^{n}$, $\omega = \omega^1$, $\omega_0(\alpha) = \alpha$, $\omega_{d+1}(\alpha) = \omega^{\omega_d(\alpha)}$ and $\omega_d = \omega_d(1)$.

Remark 4.1. To show that $\mathcal{E}$ is well defined in $\mathsf{RCA_0}$, one need only observe that the corresponding characteristic function is primitive recursive.

Definition 4.2 (fundamental sequence)**.** For $\alpha = \omega^{\alpha_0} + \cdots + \omega^{\alpha_n} \in \mathcal{E}$ and $x \in \mathbb{N}$, take $0[x] = 0$, $(\alpha + 1)[x] = \alpha$, and for α limit ordinal:

1. If $\alpha_n = \beta + 1$, then $\alpha[x] = \omega^{\alpha_0} + \cdots + \omega^{\alpha_{n-1}} + \omega^{\beta} \cdot x$,
2. If α_n is a limit, then $\alpha[x] = \omega^{\alpha_0} + \cdots + \omega^{\alpha_n[x]}$.

For well-ordered $\alpha < \varepsilon_0$, the fast growing hierarchy relative to $f : \mathbb{N} \to \mathbb{N}$ would be defined as follows:

$$\begin{aligned} F_{0,f}(x) &= f(x), \\ F_{\alpha+1,f}(x) &= F_{\alpha,f}^{(x+1)}(1), \\ F_{\lambda,f}(x) &= F_{\lambda[x],f}(x) \text{ if } \gamma \text{ is limit,} \end{aligned}$$

where $F^{(n)}$ is the n-times iterate of a function F, defined by $F^{(0)}(x) = x$ and $F^{(n+1)}(x) = F(F^{(n)}(x))$.

Remark 4.2. In the usual definition of the fast growing hierarchy we have $F_{\alpha+1}(x) = F_{\alpha}^{(x+1)}(x)$. For our purposes we use this slightly modified version (see Proposition 4.5).

Remark 4.3. The primitive recursive functions are exactly the functions elementary recursive in some F_n with $n < \omega$. Let $\mathcal{F}_\alpha$ be the set of functions elementary recursive in F_α. The multiply recursive functions (functions defined by transfinite recursion on ω^n for some $n \in \mathbb{N}$) are exactly the functions in $\bigcup_{\alpha<\omega^\omega} \mathcal{F}_\alpha$. Ben-Amram proved that SCT programs compute exactly the multiply recursive functions.

We follow standard practice in defining the fast growing hierarchy in terms of its canonical computation. Define the following function $K_f\colon (\varepsilon_0)^* \times \mathbb{N} \to (\varepsilon_0)^* \times \mathbb{N}$. Intuitively, this function represents one step in the computation of $F_{\alpha,f}(x)$. Let

$$K_f(\alpha_0 \dots \alpha_n, x) = \begin{cases} (\alpha_0 \dots \alpha_{n-1}, f(x)) & \text{if } \alpha_n = 0 \\ (\alpha_0 \dots \alpha_{n-1} \overbrace{\beta \dots \beta}^{x+1 \text{ times}}, 1) & \text{if } \alpha_n = \beta + 1 \\ (\alpha_0 \dots \alpha_{n-1} \alpha_n[x], x) & \text{if } \alpha_n \text{ is a limit} \end{cases}$$

and $K_f(\langle\rangle, x) = (\langle\rangle, x)$. Notice that $(\alpha_0 \dots \alpha_n, x)$ simply represents the term

$$F_{\alpha_0}(\dots(F_{\alpha_n}(x)\dots).$$

F_f is the result of repeated applications of the 'computation steps' (when it exists).

Definition 4.3. Let $F_{\alpha,f}(x) = \mu y.\exists l\ K_f^{(l)}(\alpha, x) = (\langle\rangle, y)$. We call the sequence $\{K_f^{(i)}(\alpha, x)\}_{i\in\mathbb{N}}$ the **computation** of $F_{\alpha,f}(x)$. Say that the computation is finite if there exists l such that $K_f^{(l)}(\alpha, x) = (\langle\rangle, y)$.

One can show that this definition is equivalent to usual Δ^0_1 definitions as in, e.g. [8] (adapted to take into account the different initial function and slightly different conditions).

Proposition 4.4 (RCA_0)**.** *For all $\alpha < \varepsilon_0$,*

$$\mathrm{WO}(\omega^\alpha) \iff (\forall f : \mathbb{N} \to \mathbb{N})\mathrm{Tot}(F_{\alpha,f}).$$

Proof. Take $h(\alpha_0 \dots \alpha_n, x) = \omega^{\alpha_0} + \dots + \omega^{\alpha_n}$ and $h(\langle\rangle, x) = 0$. Note that the computation of $F_{\alpha,f}(x)$ is finite iff $\beta_i = 0$ for some i, where $\beta_i = h(K_f^{(i)}(\alpha, x))$. Note also that $\beta_i > \beta_{i+1}$ as long as $\beta_i > 0$.

First suppose that ω^α is well-founded. Then the sequence $\{h(K_f^{(i)}(\alpha, x))\}_{i\in\mathbb{N}}$ reaches zero, so the computation is finite.

For the other direction, we assume that ω^α is not well-founded and prove that $F_{\alpha,f}$ is not total. First, a definition.

Definition 4.4 (Maximal coefficient)**.** By primitive recursion on the construction of $\alpha < \varepsilon_0$, define $\mathrm{mc}(\alpha) \in \mathbb{N}$ as follows. Let $\mathrm{mc}(0) = 0$. If $\alpha = \omega^{\alpha_0} \cdot a_0 + \dots + \omega^{\alpha_n} \cdot a_n$, where $\alpha_0 > \dots > \alpha_n$ and $a_i > 0$, then

$$\mathrm{mc}(\alpha) = \max\{\mathrm{mc}(\alpha_i), a_i\}.$$

Given an infinite descending sequence

$$\omega^\alpha > \alpha_0 > \alpha_1 > \alpha_2 > \dots,$$

take $f(x) > \mathrm{mc}(\alpha_{x+1}) + x + 1$ and strictly increasing. Assume, for a contradiction, that that the computation of $F_{\alpha,f}(f(0))$ is finite.

To ease notation, take

$$(\sigma_i, x_i) = K_f^{(i)}(\alpha, f(0)) \text{ and } \beta_i = h(\sigma_i, x_i).$$

We aim to show that $\beta_i > 0$ for all i, in contradiction with the finiteness of the computation of $F_{\alpha,f}(f(0))$. Note that $\beta_0 = \omega^\alpha$ and $x_0 = f(0)$. One can show that:

(1) if $\gamma > \beta$ is a limit, then $\gamma[\mathrm{mc}(\beta) + 1] > \beta$;

(2) $F_{\beta,f}(y) \geq f(y)$, hence $F_{\beta,f}^{(y)}(1) > y$ for all $\beta \leq \alpha$, y which occur in the computation of $F_{\alpha,f}(f(0))$;

(3) if l is the smallest such that $K_f^{(l)}(\beta, y) = (\langle\rangle, z)$, then $K_f^{(l)}(\sigma\beta, y) = (\sigma, z)$.

By primitive recursion let us define an increasing sequence (a_i) of natural numbers as follows. Set $a_0 = 0$. If σ_{a_i} ends with a zero or $\sigma_{a_i} = \langle\rangle$, let $a_{i+1} = a_i + 1$. Otherwise, let $a \geq a_i$ be the smallest such that σ_a ends with a successor $\beta + 1$, and set $a_{i+1} = a + l + 1$, where l is the least such that

$$K_f^{(l)}(\overbrace{\beta \dots \beta}^{i+1}, 1) = (\langle\rangle, z).$$

Claim 4.4.1. *For every $i > 0$ we have:*

$$\beta_{a_i} > \alpha_i$$

and

$$x_{a_i} \geq f(i).$$

From the claim, it follows that $\beta_i > 0$ since $\beta_i \geq \beta_{a_i} > \alpha_i > 0$, as desired.

Proof of the claim. Induction on i. For $i = 0$, the claim follows directly. For the induction step, assume that the claim is true for i.

Case 1. $a_{i+1} = a_i + 1$. Since $\beta_i > 0$, the inequalities follow directly from the definitions:

$$\beta_{a_i+1} = \beta_{a_i} - 1 \geq \alpha_i > \alpha_{i+1}$$

and

$$x_{a_i+1} = f(x_{a_i}) \geq f(f(i)) \geq f(i+1).$$

Case 2. Let a and β be those from the definition of a_{i+1}. By the definition of a, σ_j ends with a limit for all $j \in [a_i, a)$. Therefore, by the induction hypothesis and notice (1), $\beta_a > \alpha_i$ and $x_a \geq f(i)$. Since σ_a is of the form $\gamma_0 \dots \gamma_l \beta + 1$, σ_{a+1} has the form:

$$\gamma_1 \dots \gamma_l \overbrace{\beta \dots \beta}^{\geq \mathrm{mc}(\alpha_{i+1})+1} \overbrace{\beta \dots \beta}^{i+1},$$

so $\beta_{a_{i+1}} \geq \beta_a[\mathrm{mc}(\alpha_{i+1}) + 1] > \alpha_{i+1}$ by notice (1) and (3). By notice (2) and (3), $x_{a_{i+1}} \geq F_{\beta,f}^{(i+1)}(1) \geq f(i+1)$. This ends the proof of the claim. $\square$

4.2 Generalizing Péter-Ackermann

Recall that for $f\colon \mathbb{N} \to \mathbb{N}$ and $\alpha < \varepsilon_0$ we have:

$$\begin{aligned} F_{0,f}(x) &= f(x) \\ F_{\alpha+1,f}(x) &= F_{\alpha,f}^{(x+1)}(1) \\ F_{\lambda,f}(x) &= F_{\lambda[x],f}(x) \end{aligned}$$

Note that $F_{x,f}(y) = A_f(x,y)$, where A_f is the Péter-Ackermann function relativized to f (see [12]). We now generalize $A_f(x,y)$ as follows. For $n > 0$ and $f\colon \mathbb{N} \to \mathbb{N}$, let

$$\begin{aligned} A_f^n(x_1, x_2, \ldots, x_n, y) = \ & \textbf{if} \ \ x_1 = \ldots = x_n = 0 \ \ \textbf{then} \ \ f(y) \\ & \textbf{else if} \ \ x_1 > 0 \wedge x_2 = \ldots = x_n = 0 \ \ \textbf{then} \\ & \quad \tau_1 : A_f^n(x_1 - 1, y, x_3, \ldots, x_n, y) \\ & \vdots \\ & \textbf{else if} \ \ x_i > 0 \wedge x_{i+1} = \ldots = x_n = 0 \ \ \textbf{then} \\ & \quad \tau_i : A_f^n(x_1 \ldots, x_i - 1, y, x_{i+2}, \ldots, x_n, y) \\ & \vdots \\ & \textbf{else if} \ \ x_{n-1} > 0 \wedge x_n = 0 \ \ \textbf{then} \\ & \quad \tau_{n-1} : A_f^n(x_1, \ldots, x_{n-1} - 1, y, y) \\ & \textbf{else if} \ \ x_n > 0 \wedge y = 0 \ \ \textbf{then} \\ & \quad \tau_n : A_f^n(x_1, x_2, \ldots, x_n - 1, 1) \\ & \textbf{else} \\ & \quad \tau_0 : A_f^n(x_1, \ldots, x_{n-1}, x_n - 1, \tau_{n+1} : A_f^n(x_1, \ldots, x_n, y - 1)) \end{aligned}$$

In the interest of readability, let $\mathbf{x} = x_1, \ldots, x_n$ and $\alpha(\mathbf{x}) = \omega^{n-1}x_1 + \ldots + x_n$. For $x_n > 0$, let $\mathbf{x} - 1 = x_1, \ldots, x_{n-1}, x_n - 1$ and observe that $\alpha(\mathbf{x} - 1) = \alpha(\mathbf{x}) - 1$. Then:

$$A_f^n(\mathbf{x}, y) = \begin{cases} f(y) & \text{if } \alpha(\mathbf{x}) = 0 \\ A_f^n(\mathbf{x}', y) & \text{if } \alpha(\mathbf{x}) \text{ is limit and } \alpha(\mathbf{x}') = \alpha(\mathbf{x})[y] \\ A_f^n(\mathbf{x} - 1, 1) & \text{if } \alpha(\mathbf{x}) \text{ is successor and } y = 0 \\ A_f^n(\mathbf{x} - 1, A_f^n(\mathbf{x}, y - 1)) & \text{if } \alpha(\mathbf{x}) \text{ is successor and } y > 0 \end{cases}$$

We now show the relationship between the fast growing hierarchy and the generalized Péter-Ackermann function.

Proposition 4.5 (RCA_0)**.** *For all $n > 0$ and $f\colon \mathbb{N} \to \mathbb{N}$,*

$$A_f^n(\mathbf{x}, y) = F_{\alpha(\mathbf{x}),f}(y).$$

That is, for all $\mathbf{x}, y, z$, $A_f^n(\mathbf{x}, y) = z$ iff there exists l such that $K_f^{(l)}(\alpha(\mathbf{x}), y) = (\langle\rangle, z)$.

Proof. Write A for A_f^n and α for $\alpha(\mathbf{x})$. Say that $A(\mathbf{x}, y) = z$ in l-many steps if there exists a reduction sequence $A(\mathbf{x}, y) = t_0 \to t_1 \ldots \to t_l = z$. We also write $A(\mathbf{x}, y) \to^{(l)} z$. On the other hand, $K_f^{(l)}(\alpha, y) = (\langle\rangle, z)$ iff there exists a sequence $(\alpha, y) = k_0 \to k_1 \to \ldots \to k_l = (\langle\rangle, z)$ where $k_{i+1} = K_f(k_i)$. We also write $k_0 \to^{(l)} k_l$. We shall use the fact that $(\tau, x) \to^{(l)} (\rho, y)$ iff $(\sigma\tau, x) \to^{(l)} (\sigma\rho, y)$ for all σ.

In one direction, we prove by Π_1^0-induction on l that, for all $\mathbf{x}, y, z$, if $A(\mathbf{x}, y) = z$ in l-many steps then $(\alpha, y) \to^{(l)} (\langle\rangle, z)$. This is a relatively straightforward, if tedious, verification. Let us consider the case $x_n, y > 0$. The other cases are similar and actually simpler. Since $x_n > 0$ we have that $\alpha = \alpha(\mathbf{x})$ is a successor. Note that $\alpha - 1 = \alpha(\mathbf{x} - 1)$, where $\mathbf{x} - 1 = x_1, \ldots, x_{n-1}, x_n - 1$. Let

$$A(\mathbf{x}, y) \to A(\mathbf{x} - 1, A(\mathbf{x}, y - 1)) \to^{(l_0)} A(\mathbf{x} - 1, y') \to^{(l_1)} z$$

with $l = l_0 + l_1 + 1$. Then $A(\mathbf{x}, y - 1) = y'$ in l_0-many steps and $A(\mathbf{x} - 1, y') = z$ in l_1-many steps. By the induction hypothesis, $(\alpha, y - 1) \to^{(l_0)} (\langle\rangle, y')$ and $(\alpha - 1, y') \to^{(l_1)} (\langle\rangle, z)$. Since $(\alpha, y - 1) \to ((\alpha - 1)^{(y)}, 1)$, it follows that $((\alpha - 1)^{(y)}, 1) \to^{(l_0)} (\langle\rangle, y')$. Therefore we have:

$$(\alpha, y) \to (\alpha - 1(\alpha - 1)^{(y)}, 1) \to^{(l_0)} (\alpha - 1, y') \to^{(l_1)} (\langle\rangle, z)$$

with $l_0 + l_1 + 1 = l$.

For the other direction, we show by Π_1^0-induction on l that, for all $\mathbf{x}, y, z$, if $(\alpha, y) \to^{(l)} (\langle\rangle, z)$, then $A(\mathbf{x}, y) = z$ in less than l^2-many steps. The l^2 bound is not optimal but does the job. Once again, consider the case $x_n, y > 0$ so that $\alpha = \alpha(\mathbf{x})$ is a successor and $\alpha - 1 = \alpha(\mathbf{x} - 1)$ with $\mathbf{x} - 1 = x_1, \ldots, x_{n-1}, x_n - 1$. Suppose that

$$(\alpha, y) \to ((\alpha - 1)^{(y+1)}, 1) \to^{(l_0)} (\alpha - 1, y') \to^{(l_1)} (\langle\rangle, z).$$

Then $l = l_0 + l_1 + 1$. As before, note that $(\alpha, y - 1) \to ((\alpha - 1)^{(y)}, 1)$. By induction, $A(\mathbf{x}, y - 1) = y'$ within $\leq (l_0 + 1)^2$-many steps and $A(\mathbf{x} - 1, y') = z$ within $\leq l_1^2$-many steps, where $\mathbf{x} - 1 = x_1, \ldots, x_{n-1}, x_n - 1$. Therefore we have a reduction sequence

$$A(\mathbf{x}, y) \to A(\mathbf{x}', A(\mathbf{x}, y - 1)) \to \ldots \to A(\mathbf{x}', y') \to \ldots \to z$$

of length $\leq (l_0 + 1)^2 + l_1^2 + 1 \leq l^2$. Note in fact that $l_i > 0$. □

Corollary 4.6 (RCA_0)**.** *The following holds:*

- $\mathrm{WO}(\omega^{\omega^\omega}) \iff (\forall n > 0)(\forall f : \mathbb{N} \to \mathbb{N})\mathrm{Tot}(A_f^n)$
- $\mathrm{WO}(\omega^\omega) \iff (\forall f : \mathbb{N} \to \mathbb{N})\mathrm{Tot}(A_f^2)$

Proof. This follows from Proposition 4.4 and Proposition 4.5. □

4.3 From soundness to $\mathrm{WO}(\omega_3)$

We can now give the desired lower bounds.

Definition 4.5 (description $\mathcal{A}_n$ of A_f^n)**.** It is convenient to define $\mathcal{A}_n$ on parameters $x_1, \ldots, x_n, x_{n+1}$. That is, we write x_{n+1} for y. Define $\mathcal{A}_n = \{A_1, \ldots, A_{n+1}\}$ as follows. For every $0 < j \leq n+1$, let A_j be the size-change graph with arcs $x_j \xrightarrow{\downarrow} x_j$ and $x_i \xrightarrow{\Downarrow} x_i$ for all $0 < i < j$.

Note that $\mathcal{A}_n$ does not depend on f.

Proposition 4.7 (RCA_0)**.** *For all $n \in \mathbb{N}$ and $f\colon \mathbb{N} \to \mathbb{N}$, $\mathcal{A}_n$ is a safe ISCT description of A_f^n. More precisely, A_i is a safe description of τ_i for all $0 < i \leq n+1$, and A_n is a safe description of τ_0.*

Proof. It is easy to see that $\mathcal{A}_n$ is a safe description. Let us show that every $G \in \mathsf{cl}(\mathcal{A}_n)$ has an arc $x \xrightarrow{\downarrow} x$. Let $G = G_0; G_1; \ldots; G_{k-1}$ with $G_j \in \mathcal{A}_n$ for every $j < k$. Let $0 < i \leq n+1$ be least such that $A_i \in \{G_0, \ldots, G_{k-1}\}$. Then $x_i \xrightarrow{\downarrow} x_i \in G$. □

Note that the size-change graphs defined in this description could be extended to other size-change graphs, which also safely describe P, by adding to A_j the arcs $y \xrightarrow{\Downarrow} x_{j+1}$, $y \xrightarrow{\Downarrow} y$ and $x_i \xrightarrow{\Downarrow} x_i$ for every $j < i \leq n$. Anyway for our goals the description above is more suitable.

Corollary 4.8 (RCA_0)**.** *ISCT soundness implies* $\mathrm{WO}(\omega_3)$.

Proof. This follows from Corollary 4.6. □

Remark 4.9. Note that for any standard $n > 0$, RCA_0 proves that $\mathcal{A}_n$ is MSCT. In particular, MSCT soundness implies $\mathrm{WO}(\omega^\omega)$ by Corollary 4.6. It turns out that proving MSCT for all $\mathcal{A}_n$ requires $\mathsf{I}\Sigma^0_2$.

Proposition 4.10 (RCA_0)**.** *The following are equivalent:*

- $\mathsf{I}\Sigma^0_2$;
- *For all $n > 0$, $\mathcal{A}_n$ is MSCT.*

Proof. For the forward direction, let $M = G_0, G_1, \ldots$ be a multipath with $G_i \in \mathcal{A}_n$. Let $0 < i \leq n+1$ be least such that A_i appears infinitely often. Then there is an infinite descent starting with x_i.

For the reversal, one can adapt the proof of [6, Theorem 6]. For the sake of completeness, we briefly describe the main idea. Starting point is that $\mathsf{I}\Sigma^0_2$ is equivalent to the Strong Pigeonhole Principle (see e.g., [6]) which states that given a coloring in k many colors of the natural numbers, there exists the set of colors which appear infinitely many times in this coloring. Therefore it is sufficient to show that for every finite coloring

$c : \mathbb{N} \to k$ the set $I^\infty = \{i < k \colon (\exists^\infty x) c(x) = i\}$ exists. As in the proof of [6, Theorem 6] we can define for all $x \in \mathbb{N}$ the set $\mathcal{I}_x$ of guesses at stage x. That is, every $I \in \mathcal{I}_x$ is a non-empty subset of k and $I \subseteq I^\infty$ iff $I \in \mathcal{I}_x$ for infinitely many x. Let $n+1 = 2^k - 1$. Then we have $n+1$-many non-empty subsets of k, say $I_1, \ldots, I_{n+1}$. We can assume that $|I_i| < |I_j|$ implies $i > j$. Now define a multipath $M = G_0, G_1, \ldots$ in $\mathcal{A}_n$ by letting $G_x = A_i$, where i is least such that I_i is a guess at stage x of maximal size. By the assumption there exists an infinite descent starting from some parameter x_i with $0 < i \leq n+1$ at some point t. We claim that $I_i = I^\infty$. Since there are infinitely many arcs of the form $x_i \xrightarrow{\downarrow} x_i$, we have that $I_i \subseteq I^\infty$. Now suppose for a contradiction that $I_i \neq I^\infty$. Then there exists a stage $x > t$ with a guess I of size bigger than I_i. Therefore, by definition, there exists $j < i$ such that $G_x = A_j$ and so in G_x there is no arc from x_i to x_i, a contradiction. $\square$

5 Upper bound

In this section we aim to show the following:

Theorem 5.1 (RCA_0)**.** WO(ω_3) *implies ISCT soundness.*

From the proof of this result we then extract a bound for the length of computations of tail-recursive ISCT programs (see Subsection 5.3).

Definition 5.1. We call **proper** a term of the form $f(\mathbf{u})$ with $f \in \mathrm{Fun}$ and $u \in \mathbb{N}^{\mathrm{arity}(f)}$. We also say that a reduction $t \to_P s = t[f(\mathbf{u})]_\rho$ is **proper** if $f \in \mathrm{Fun}$.

Proof idea: Given an infinite reduction sequence

$$f(\mathbf{u}) = t_0 \to_P t_1 \to_P \ldots \to_P t_n \to_P \ldots$$

we assign ordinals

$$\omega_3 > \alpha_0 > \alpha_1 > \ldots > \alpha_n > \ldots$$

Actually, since we also consider reductions involving primitive operators, we will define a non-increasing sequence of ordinals which decreases infinitely many times. For the details, one can go to the proof. Here we just outline some ideas that hopefully will make the proof easier to follow.

As in Tait [21], we want to assign to each term t_n a finite set of ordinals $\gamma_0, \gamma_1, \ldots$ in ω^ω. Each γ_i corresponds to a subterm of t_n of the form $g(\mathbf{s})$ with $g \in \mathrm{Fun}$. To t_n we assign the ordinal

$$\alpha_n = \bigoplus_i \beta_i < \omega_3,$$

where

$$\beta_i = \begin{cases} \omega^{\gamma_i} \cdot a & \text{if } g(\mathbf{s}) \text{ is proper} \\ \omega^{\gamma_i} & \text{otherwise} \end{cases}$$

Here a is the the maximum **size** of e^f for $f \in \mathrm{Fun}$. The size of an expression is defined as the number of function symbols. The idea is that a one-step reduction gives rise to at most a-many subterms of the form $g(\mathbf{s})$ with $g \in \mathrm{Fun}$. We use the following basic fact from ordinal arithmetic without further notice: if $\gamma > \gamma_0, \gamma_1, \ldots$ then $\omega^\gamma > \bigoplus_i \omega^{\gamma_i}$. Therefore, if a term t with assigned ordinal γ gives rise to a-many subterms $s_0, s_1, \ldots$ with ordinals $\gamma_0, \gamma_1, \ldots$, $\gamma > \gamma_i$ if t_i is proper and $\gamma \geq \gamma_i$ otherwise, then $\omega^\gamma \cdot a > \bigoplus_i \beta_i$, where β_i is defined as above.

Example 5.2. Consider the following illustrative example. To the sequence

$$A(2,3) \to A(1, A(2,2)) \to A(1, A(1, A(2,1))) \to \ldots$$

we assign the ordinals

$$\omega^{\omega 2+3} \cdot 2 > \omega^{\omega 2+3} + \omega^{\omega 2+2} \cdot 2 > \omega^{\omega 2+3} + \omega^{\omega 2+2} + \omega^{\omega 2+1} \cdot 2 > \ldots$$

This is a descending sequence in ω_3. Here, $a = 2$. For a descending sequence in ω^ω use base $b = 3$ instead of ω. We have $b^{\omega^2} = \omega^\omega$. Replace $\omega^{\omega x+y} a$ with $\omega^x \cdot b^y \cdot a$.

Following Ben-Amram [1], there exists a bound $m \in \mathbb{N}$ such that every finite multipath $M = G_0, \ldots, G_n, \ldots$ of length $\geq m$ is **foldable**, where M is foldable if it can be decomposed into three multipaths $M = ABC$ with $H = \overline{B} = \overline{C} = \overline{BC}$, where $\overline{M}$ is the composition of the graphs in M. Note that H is idempotent. In particular, the source and the target functions of H coincide. The idea is to assign to each subterm of t_n of the form $g(\mathbf{s})$ with $g \in \mathrm{Fun}$ an ordinal of the form $\gamma(\mathbf{u}) < \omega^\omega$, where $\mathbf{u} = \mathbf{u}_0, \mathbf{u}_1, \ldots$ is a finite sequence of tuples appearing in a state transition sequence $(f_0, \mathbf{u}_0) \xrightarrow{G_0} (f_1, \mathbf{u}_1) \xrightarrow{G_1} \ldots$ of length $< m$. In a one-step reduction we might either extend or contract a finite state transition sequence $(f_0, \mathbf{u}_0) \xrightarrow{G_0} (f_1, \mathbf{u}_1) \xrightarrow{G_1} \ldots$ into another finite state transition sequence $(g_0, \mathbf{v}_0) \xrightarrow{H_0} (g_1, \mathbf{v}_1) \xrightarrow{H_1} \ldots$. The second case arises when the corresponding multipath becomes foldable. In the first case, the sequence $\mathbf{v} = \mathbf{v}_0, \mathbf{v}_1, \ldots$ properly extends the sequence $\mathbf{u} = \mathbf{u}_0, \mathbf{u}_1, \ldots$. In the second case, the sequence $\mathbf{v}$ is lexicographically smaller than the sequence $\mathbf{u}$. It turns out that we can map sequences $\mathbf{u}$ of bounded length to ordinals $\gamma(\mathbf{u})$ in ω^ω so that in both cases $\gamma(\mathbf{u}) > \gamma(\mathbf{v})$. For the sake of exposition we say that the sequence $\mathbf{u}$ is **above** the sequence $\mathbf{v}$.

The existence of a bound on the length of foldable multipaths is an easy application of finite Ramsey's theorem. We thus have the following:

Lemma 5.2 (RCA_0)**.** *Let $\mathcal{G}$ be a finite set of size-change graphs. Then there exists $m \in \mathbb{N}$ such that every multipath M in $\mathrm{cl}(\mathcal{G})$ of length $\geq m$ is foldable.*

Proof. Finite Ramsey's theorem for pairs. □

Note that the above lemma applies to every $\mathcal{G}$. The ISCT assumption ensures that the idempotent size-change graph H in the definition of foldable multipath contains a strict arc of the form $x \xrightarrow{\downarrow} x$.

In the next subsection we show how to map sequences $\mathbf{u}$ of bounded length to ordinals $\gamma(\mathbf{u}) < \omega^\omega$ so that if $\mathbf{u}$ is above $\mathbf{v}$ then $\gamma(\mathbf{u}) > \gamma(\mathbf{v})$. The reader may skip this part on a first reading.

5.1 Aboveness

Let $p \in \mathbb{N}$ be fixed.

Definition 5.3. Given a sequence $\mathbf{u} \in \mathbb{N}^{<p}$, let $\mathbf{u}_p \in (\mathbb{N} \cup \{\omega\})^p$ be the sequence of length p which is obtained from $\mathbf{u}$ by adding $p - \text{length}(\mathbf{x})$-many ω. That is,

$$\mathbf{u}_p := \mathbf{u}^\frown \omega^{(p-\text{length}(\mathbf{u}))},$$

where for any natural number n, $\omega^{(n)}$ is the sequence of length n with constant value ω.

It is easy to see that $\mathbf{u}$ is above $\mathbf{v}$ if and only if $\mathbf{u}_p >_{\text{lex}_p} \mathbf{v}_p$, where lex_p is the standard lexicographic order of $(\omega + 1)^p$.

Definition 5.4. Given a sequence $\mathbf{u} \in \omega^{<p}$, define

$$\gamma_p(\mathbf{u}) := \bigoplus_{i=0}^{p-1} \omega^{p-1-i}(2 \cdot \mathbf{u}_p(i)).$$

Lemma 5.3 (RCA_0)**.** *Let $p \in \mathbb{N}$ and $\mathbf{u} \in \mathbb{N}^{<p}$. For any $j < p$,*

$$\bigoplus_{i=j}^{p-1} \omega^{p-1-i}(2 \cdot \mathbf{u}_p(i)) < \omega^{p-j} \cdot 2.$$

Proof. We prove it by induction on $p - j + 1$. If $j = p - 1$, the first sum is empty and the thesis follows. Assume that the claim holds for $j + 1$, we prove it for j. Note that $\omega^{p-1-j}(2 \cdot \mathbf{u}_p(j)) \leq \omega^{p-j}$. Moreover by induction hypothesis

$$\bigoplus_{i=j+1}^{p-1} \omega^{p-1-i}(2 \cdot \mathbf{u}_p(i)) < \omega^{p-j-1} \cdot 2.$$

Therefore

$$\bigoplus_{i=j}^{p-1} \omega^{p-1-i}(2 \cdot \mathbf{u}_p(i)) < \omega^{p-j} \oplus \omega^{p-j-1} \cdot 2 < \omega^{p-j} \cdot 2.$$

□

Lemma 5.4 (RCA_0)**.** *Let $p \in \omega$ and $\mathbf{u}, \mathbf{v} \in \mathbb{N}^{<p}$. If $\mathbf{u}$ is above $\mathbf{v}$ then $\gamma_p(\mathbf{u}) > \gamma_p(\mathbf{v})$.*

Proof. If $\mathbf{u}$ is above $\mathbf{v}$ then $\mathbf{u}_p >_{\text{lex}_p} \mathbf{v}_p$. Therefore there exists $j \in p$ such that

$$(\forall i < j)(\mathbf{u}_p(i) = \mathbf{v}_p(i) \wedge \mathbf{u}_p(j) > \mathbf{v}_p(j)).$$

By the lemma above

$$\bigoplus_{i=0}^{p-1} \omega^{p-1-i}(2 \cdot \mathbf{u}_p(i)) \geq \bigoplus_{i=0}^{j-1} \omega^{p-1-i}(2 \cdot \mathbf{v}_p(i)) \oplus \omega^{p-j}(2 \cdot (1 + \mathbf{v}_p(j))) > \bigoplus_{i=0}^{p-1} \omega^{p-1-i}(2 \cdot \mathbf{v}_p(i)).$$

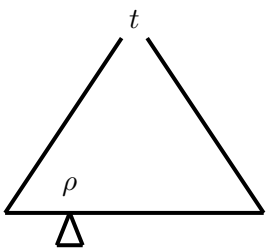
□

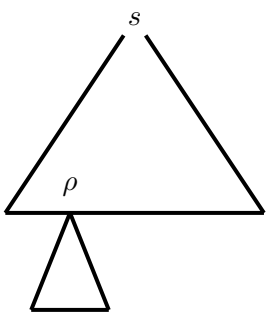
5.2 From $\mathrm{WO}(\omega_3)$ to soundness

We first give the following definition of stem. Note that this is made in RCA_0.

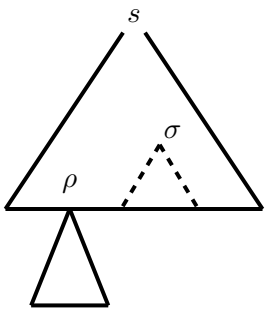
Definition 5.5. Let $t \to_P s$ with $s = t[f(\mathbf{u})]_\rho$. For every subterm $g(\mathbf{s})$ of s with $g \in \mathrm{Fun}$ there exists a unique position σ of t, called the **stem** of $g(\mathbf{s})$, such that $t|_\sigma = h(\mathbf{t})$, $h \in \mathrm{Fun}$, and one of the following holds:

- $\sigma \perp \rho$ and $g(\mathbf{s}) = h(\mathbf{t})$;
- $\sigma \subset \rho$ and $g(\mathbf{s}) = h(\mathbf{s})$. In this case $h(\mathbf{t})$ is not proper and $\mathbf{t} \to_P \mathbf{s}$;
- $\sigma = \rho$, and so $h(\mathbf{t}) = f(\mathbf{u})$, and $g(\mathbf{s}) = e^f|_\tau(\mathbf{u})$ for some $\tau : f \to g$. In this case $(f, \mathbf{u}) \xrightarrow{\tau} (g, \mathbf{s})$.

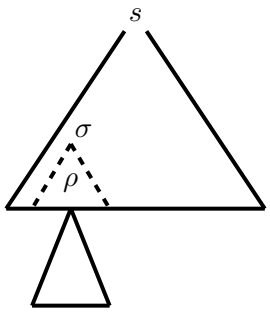




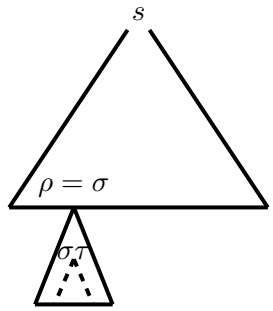
$\sigma \perp \rho$



$\sigma \subset \rho$



$\sigma = \rho$



Proof of Theorem 5.1. Let $\mathcal{G}$ be a safe ISCT description of a program P, i.e., $\mathcal{G}$ is safe for P and every idempotent graph $G \in \mathrm{cl}(\mathcal{G})$ contains a strict arc $x \xrightarrow{\downarrow} x$.

Suppose we are given an infinite reduction sequence

$$f_0(\mathbf{u}_0) = t_0 \to_P t_1 \to_P \ldots \to_P t_n \to_P \ldots$$

We will assign ordinals

$$\omega_3 > \alpha_0 \geq \alpha_1 \geq \ldots \geq \alpha_n > \ldots$$

and prove that $\alpha_n > \alpha_{n+1}$ for infinitely many n. Indeed, we will have $\alpha_n > \alpha_{n+1}$ for every proper reduction $t_n \to_p t_{n+1}$. Note that in any infinite reduction sequence there must be infinitely many proper reductions (exercise).

If m bounds the length of any non-foldable multipath in $\mathrm{cl}(\mathcal{G})$ and r is the maximum arity of f for $f \in \mathrm{Fun}$, set $p = (m+1) \cdot r$. Let a be the the maximum size of e^f for $f \in \mathrm{Fun}$. From now on we identify a sequence $\mathbf{u}$ of length $< p$ with the ordinal $\gamma_p(\mathbf{u}) \in \omega^\omega$.

By primitive recursion we want to define for all n and for every subterm $h(\mathbf{t})$ of t_n with $h \in \mathrm{Fun}$ a finite multipath $M = G_0, G_1 \ldots, G_{l-1}$ in $\mathrm{cl}(\mathcal{G})$ of length $< m$ and a sequence $\mathbf{u} = \mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_l$ of length $< p$ such that:

(a) $(f_0, \mathbf{u}_0) \xrightarrow{G_0} (f_1, \mathbf{u}_1) \xrightarrow{G_1} \ldots \xrightarrow{G_{l-1}} (f_l, \mathbf{u}_l)$, where $G_i : f_i \to f_{i+1}$.

(b) If $h(\mathbf{t})$ is proper, then $(f_l, \mathbf{u}_l) = (h, \mathbf{t})$.

(c) If $h(\mathbf{t})$ is not proper, we also specify $\tau : f_l \to h$ such that $(f_l, \mathbf{u}_l) \xrightarrow{\tau} (h, \mathbf{t})$.

Note that $(f_l, \mathbf{u}_l) \xrightarrow{\tau} (h, \mathbf{t})$ is a Σ^0_1-condition. We assign to every such subterm $h(\mathbf{t})$ the ordinal $\omega^{\mathbf{u}}$ if $h(\mathbf{t})$ is not proper, and the ordinal $\omega^{\mathbf{u}} \cdot a$ otherwise. Finally, we let α_n be the natural sum of all these ordinals.

Construction.

Stage $n = 0$. We have only the term $f_0(\mathbf{u}_0)$. Let M be the multipath of length 0 consisting of $\mathrm{Par}(f_0)$ and $\mathbf{u} = \mathbf{u}_0$. Conditions (a)–(c) are trivially satisfied.

Stage n+1. Let $t_{n+1} = t_n[f(\mathbf{u})]_\rho$. We can assume by Σ^0_1-induction that for every non-proper subterm $h(\mathbf{t})$ of t_n the corresponding $\tau : f_l \to h$ is as above. Let $g(\mathbf{s})$ be a subterm of t_{n+1}. We want to assign a pair $N, \mathbf{v}$. Let σ be the stem of $g(\mathbf{s})$ and $M, \mathbf{u}$ be the pair associated with $h(\mathbf{t}) = t_n|_\sigma$.

Remark 5.5. If $\sigma \subseteq \rho$, we can specify $\tau : f_l \to g$ such that $(f_l, \mathbf{u}_l) \xrightarrow{\tau} (g, \mathbf{s})$. In fact,

- if $\sigma \subset \rho$, i.e., $h(\mathbf{t})$ is not proper, then $(f_l, \mathbf{u}_l) \xrightarrow{\tau} (h, \mathbf{t})$, where τ has been specified earlier in the construction by (c), and therefore $(f_l, \mathbf{u}_l) \xrightarrow{\tau} (g, \mathbf{s})$. Note in fact that $h = g$ and $\mathbf{t} \to_P \mathbf{s}$;

- if $\sigma = \rho$, i.e., $h(\mathbf{t})$ is proper and equal to $f(\mathbf{u})$, then $(f_l, \mathbf{u}_l) = (f, \mathbf{u})$ by (b), and therefore $(f_l, \mathbf{u}_l) \xrightarrow{\tau} (g, \mathbf{s})$, where τ is such that $g(\mathbf{s}) = e^f|_\tau(\mathbf{u})$.

Case 1. $\sigma \perp \rho$ or $g(\mathbf{s})$ is not proper. Do nothing, i.e., $N = M$ and $\mathbf{v} = \mathbf{u}$.

Case 2. $\sigma \subseteq \rho$ and $g(\mathbf{s})$ is proper, say $g(\mathbf{s}) = g(\mathbf{w})$. Consider the multipath MG, where G is the size-change graph for the call $\tau : f_l \to g$, where τ is as in the remark above.

Sub-case 1. MG has length $< m$. Let $N = MG$, and $\mathbf{v} = \mathbf{u}, \mathbf{w}$. Note that $\mathbf{u}$ is above $\mathbf{v}$.

Sub-case 2. MG is foldable, say $MG = ABC$, with $H = \overline{B} = \overline{C} = \overline{BC}$. Then we fold M, i.e., we let $N = AH$. Suppose that $B = G_i, \ldots, G_{j-1}$. Note that $g = f_i = f_j$ and $H : g \to g$. Now, H is idempotent, and so by ISCT contains a strict arc $x_k \xrightarrow{\downarrow} x_k$ for some k, where $\mathrm{Par}(g) = \{x_0, x_1 \ldots, \}$. We define $\mathbf{v}$ according to whether $x_k \xrightarrow{\downarrow} x_k \in H$ or not. Let

$$\mathbf{v} = \mathbf{u}_0, \ldots, \mathbf{u}_{i-1}, \mathbf{v}_i, \mathbf{w},$$

where the k-element of $\mathbf{v}_i$ equals the k-element of $\mathbf{u}_j$ if $x_k \xrightarrow{\downarrow} x_k \in H$, and the k-element of $\mathbf{u}_i$ otherwise. Note that we have $\mathbf{u}_i \geq \mathbf{v}_i$ coordinate-wise. Also, $\mathbf{v}_i$ is lexicographically smaller than $\mathbf{u}_i$ and therefore $\mathbf{u}$ is above $\mathbf{v}$.

Verification.

Let us first check that in all cases $N, \mathbf{v}$ satisfy conditions (a)-(c).

Case 1. Clearly (a) holds. The only interesting case is when $\sigma \subseteq \rho$. In this case when $g(\mathbf{s})$ is not proper we have (c) by the remark above.

Case 2. Clearly (b) holds. We need only to check (a).

Sub-case 1. We just need to show that $(f_l, \mathbf{u}_l) \xrightarrow{G} (g, \mathbf{w})$. This follows from the remark above and the safety of $\mathcal{G}$.

Sub-case 2. We just need to show that $(f_{i-1}, \mathbf{u}_{i-1}) \xrightarrow{G_{i-1}} (g, \mathbf{v}_i) \xrightarrow{H} (g, \mathbf{w})$. The first state transition follows from the fact that $(f_{i-1}, \mathbf{u}_{i-1}) \xrightarrow{G_{i-1}} (g, \mathbf{u}_i)$ and $\mathbf{u}_i \geq \mathbf{v}_i$. The second state transition follows from the safety of $\mathcal{G}$ and the fact that both $(g, \mathbf{u}_i) \xrightarrow{\overline{BC}} (g, \mathbf{w})$ and $(g, \mathbf{u}_j) \xrightarrow{\overline{C}} (g, \mathbf{w})$ hold, and $H = \overline{BC} = \overline{C}$.

Claim 5.5.1. *For all n, $\alpha_n \geq \alpha_{n+1}$, and $\alpha_n > \alpha_{n+1}$ for infinitely many n.*

For every position σ of a subterm $h(\mathbf{t})$ of t_n with $h \in \mathrm{Fun}$, let α_σ be the ordinal corresponding to $h(\mathbf{t})$, S_σ be the set of subterms $g(\mathbf{s})$ of t_{n+1} with stem σ, and β_σ be the natural sum of ordinals corresponding to terms in S_σ. Note the we may have $S_\sigma = \emptyset$.

In such a case let $\beta_\sigma = 0$. On the other hand, every subterm $g(\mathbf{s})$ of t_{n+1} with $g \in \mathrm{Fun}$ belongs to some S_σ. Thus $\alpha_n = \oplus_\sigma \alpha_\sigma$ and $\alpha_{n+1} = \oplus_\sigma \beta_\sigma$.

We claim that $\alpha_\sigma \geq \beta_\sigma$ for every position σ and hence $\alpha_n \geq \alpha_{n+1}$.

Case 1. If $h(\mathbf{t})$ is not proper, then $\alpha_\sigma = \omega^{\mathbf{u}}$ and $|S_\sigma| = 1$. Say $S_\sigma = \{g(\mathbf{s})\}$. We assign ordinal $\omega^{\mathbf{u}}$ if $g(\mathbf{s})$ is not proper, and $\omega^{\mathbf{v}} \cdot a$ otherwise. In the latter case $\mathbf{u}$ is almost above $\mathbf{v}$, and so $\alpha_\sigma > \beta_\sigma$.

Case 2. If $h(\mathbf{t})$ is proper and $\sigma \neq \rho$, then $\alpha_\sigma = \omega^{\mathbf{u}} \cdot a$, $S_\sigma = \{h(\mathbf{t})\}$ and $\alpha_\sigma = \beta_\sigma$.

Case 3. If $\sigma = \rho$, that is $h(\mathbf{t})$ is the $f(\mathbf{u})$ in the function reduction from t_n to t_{n+1}, then $\alpha_\rho = \omega^{\mathbf{u}} \cdot a$ and $|S_\rho| \leq a$. Note that if $e^f(\mathbf{u}) \in \mathbb{N}$ then $S_\rho = \emptyset$, and so $\alpha_\rho > \beta_\rho$. Otherwise, each $g(\mathbf{s})$ in S_ρ is either not proper, in which case we assign the ordinal $\omega^{\mathbf{u}}$, or proper, in which case we assign an ordinal $\omega^{\mathbf{v}} \cdot a$, where $\mathbf{u}$ is almost above $\mathbf{v}$, and so $\omega^{\mathbf{u}} > \omega^{\mathbf{v}} \cdot a$. In both cases we have $\alpha_\rho > \beta_\rho$.

Note that Case 1 might occur in any reduction and hence we can have $\alpha_n > \alpha_{n+1}$ even if the reduction is not proper. However, Case 3 occurs in every proper reduction and $\alpha_n > \alpha_{n+1}$ for every such reduction. The claim follows. □

In particular Theorem 5.1 shows that any descending sequence of ordinals associated to some computation of the generalized Péter-Ackermann function A_f^n (as defined in Section 2.4) is bounded by some ordinal of the form $\omega^{\omega^{b_n}}$ for some natural number b_n. Anyway such b_n depends on the bound on the length of foldable multipaths provided in Lemma 5.2 by an application of the finite Ramsey's theorem for pairs. Since uniform bounds for the finite Ramsey's theorem for pairs are rather large, so are the bounds b_n extracted from our proof. These are definitely larger than ω^{ω^n}, the ordinal which corresponds to A_f^n (see Proposition 4.4 and Proposition 4.5).

5.3 Upper bound for tail-recursive programs

In this subsection we consider tail-recursive programs. A program function definition is tail-recursive is the recursive call occurs only once and it is the most external function. For instance:

$$h(\mathbf{t}) = h(f_0(\mathbf{t}), \dots f_{n-1}(\mathbf{t})).$$

A program is tail-recursive if every function definition is tail-recursive and there is no mutual recursion. There exists a direct transition-based translation into transition-based programs (see, e.g., [9]) and tail-recursive programs are often easy to handle in implementations.

The goal of this section is to show that the functional programs which are tail-recursive and ISCT compute exactly the primitive recursive functions. On the one hand, all primitive recursive functions can be computed by simple tail-recursive programs which are ISCT (e.g. see [20]). On the other hand, Ben-Amram in [1] has already proved that the first order functional programs defined without nested recursion which are ISCT compute primitive recursive functions. Since tail-recursive programs do not allow nested recursion

our result is a corollary of [1]. In [20] there is a different proof which uses an analysis of the intuitionistic proof of the Termination Theorem. As a side result of this analysis, some large bounds are extracted. By following a completely different approach, which follows closely the proof of Theorem 5.1, we provide a new proof and we extract the corresponding bounds.

Proposition 5.6. WO(ω^ω) *implies the termination of every tail-recursive ISCT program.*

Proof. We follow the argument of Theorem 5.1, but we assign a different ordinal to every subterm $g(s)$ of t_n. Given a tail-recursive functional program P let $\{g_0, \ldots, g_{k-1}\}$ be a fixed ordering between the functions of P such that for every $i < j < k$ g_j does not occur in the expression defining the function g_i (note that such an ordering exists since there is no mutual recursion in P).

Given a term $g_j(\mathbf{t})$ and $\mathbf{u}$, where j is the index with respect to our fix ordering, we assign to $g_j(\mathbf{t})$ the ordinal $\omega^{pj}(\omega \cdot \gamma(\mathbf{u}) + 1)$ if $g_j(\mathbf{t})$ is proper and the ordinal $\omega^{pj}(\omega \cdot \gamma(\mathbf{u}))$ if $g_j(s)$ is not proper.

Now following the schema of the verification for the bound as presented in the proof of Theorem 6.1 we have the following cases:

- If $g_j(\mathbf{t})$ is not proper, then $\alpha_\sigma = \omega^{pj}(\omega \cdot \gamma(\mathbf{u}))$ and $S_\sigma = \{g_i(\mathbf{s})\}$ for some $i \leq j$. In both the possible cases for $g_i(\mathbf{s})$ we have $\alpha_\sigma > \beta_\sigma$, since $\omega \cdot \gamma(\mathbf{u})$ is a limit ordinal.
- If $g_j(\mathbf{t})$ is proper and $\sigma \neq \rho$ then $S_\sigma = \{g_j(\mathbf{t})\}$. Therefore $\alpha_\sigma = \beta_\sigma$.
- If $\sigma = \rho$. If $g_j(\mathbf{t})$ is proper, since the program is tail recursive we have in the worst case that $S_\sigma = \{f_0(t), ...f_{n-1}(t), g_{j'}(f_0(t), ..., f_{n-1}(t))\}$ for some functions $f_0, \ldots, f_n$ whose level which respect to our ordering is less than j and $j' \leq j$. Since $f_0, \ldots f_{n-1}$ have index less than j with respect to our fixed ordering, we associate either $\omega^{ph_i}(\omega \cdot \gamma(\mathbf{u}))$ (if not proper) or $\omega^{ph_i}(\omega \cdot \gamma(\mathbf{v_i}) + 1)$ (if proper) to them for some $h_i < j$ and some $\mathbf{v}_i$ such that $\mathbf{u}$ is almost above $\mathbf{v}_i$. If $g_{j'}(f_0(\mathbf{t}), ..., f_{n-1}(\mathbf{t}))$ is not proper we associate $\omega^{pj'}(\omega \cdot \gamma(\mathbf{u}))$ to it, otherwise we associate $\omega^{pj'}(\omega \cdot \gamma(\mathbf{v}) + 1)$ for some $\mathbf{v}$ such that $\mathbf{u}$ almost above $\mathbf{v}$. Since $j \geq j'$ we have $\alpha_\sigma > \beta_\sigma$. □

Now assume that P is a tail-recursive program as above and let g_i be the entry function of P. By the proposition above every computation from $g(x_0, \ldots, x_{n-1})$ corresponds to a descending sequence of ordinals below ω^{pi+1}. We claim that any computation of $g_i(x_0, \ldots, x_{n-1})$ has length less than $F_{pi+2,f}(0)$, for $f(x) = 2x + 2 + \max\{x_0, \ldots, x_{n-1}, p\}$.

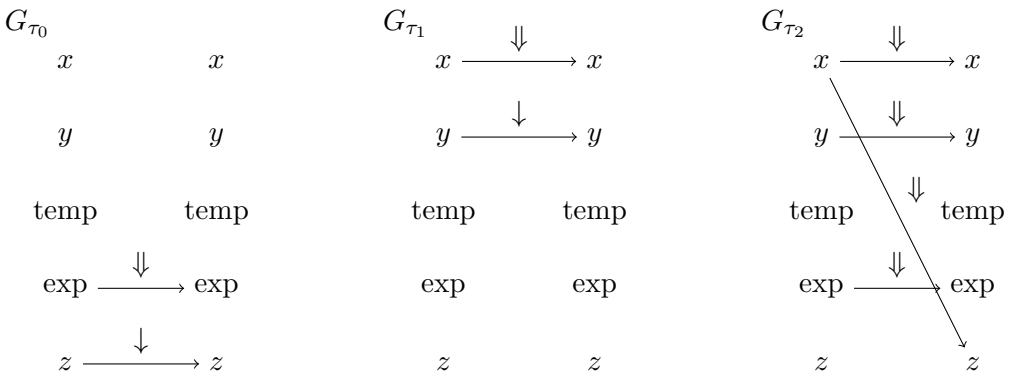
To prove this we directly adapt the proof of Proposition 4.4. Note that for every $\alpha \in \omega^{pi+1}$ we have $\mathrm{mc}(\alpha) \leq \max\{a_i : i \in n\} \cup \{p\}$. Moreover, if $\mathbf{v}$ is obtained after a step from $\mathbf{u}$ in the proof of Proposition 5.6, then $\max\{v_j : j < p+1\} < \max\{u_j : j < p+1\}+1$, therefore $f(x) > \mathrm{mc}(\alpha_{x+1}) + x + 1$, as required in the proof of Proposition 4.4. Observe that the proof of Proposition 4.4 guarantees that for every infinite decreasing sequence α_i we get that every β_i is positive. We can straightforwardly derive from this argument that if the sequence of α_i has length n, then we β_i is positive for every $i < n$.

Assume that we have a decreasing sequence of ordinals below ω^{pi+1} which is derived from a computation of length greater than $F_{pi+2,f}(0)$. Hence we would get that the corresponding β_i are positive for every $i \leq F_{pi+2,f}(0)$. But this provides a contradiction, by definition of β_i.

Observe that, in general, these bounds seem to be huge. For instance let us consider the toy-program analyzed in [20]:

$$
\begin{aligned}
f(x, y, \text{temp}, \exp, z) = \ & \textbf{if} \ \ (y = 0) \ \ \textbf{then} \ \ 1 \\
& \textbf{else if} \ \ (y = 1) \ \ \textbf{then} \ \ \exp \\
& \qquad \textbf{else} \ \ \tau_1 : f(x, y-1, *, \tau_2 : g(x, y, 0, \exp, x), *) \\
g(x, y, \text{temp}, \exp, z) = \ & \textbf{if} \ \ (z = 0) \ \ \textbf{then} \ \ 0 \\
& \textbf{else if} \ \ (z = 1) \ \ \textbf{then} \ \ \text{temp} \\
& \qquad \textbf{else} \ \ \tau_0 : g(*, *, \text{temp} + \exp, \exp, z-1)
\end{aligned}
$$

where $*$ denotes any value. Note that $f(x, y, 0, 1, z)$ computes x^y. Every size-change graph corresponds to some composition of $G_{\tau_0} : g \to g$, $G_{\tau_1} : f \to f$ and $G_{\tau_2} : f \to g$.



The idempotent graphs in $\text{cl}(\mathcal{G})$ are $G_{\tau_0} : g \to g$ and $G_{\tau_1} : f \to f$ (since the source and the target of the other size-change graphs are different). Hence this program is ISCT. Recall that p is defined to be $(m+1)r$. Note that the maximal arity r for this program is 5, so $p > 5$. As we already mentioned, the bound m for the length of the unfoldable multipaths is provided by an application of the finite Ramsey's theorem and it is well-known that the bounds for the finite Ramsey's numbers are pretty large for $n > 4$. Therefore the bounds extracted from this proof involve functions $F_{l,f}$ with l much bigger than 5, which is extremely loose, since $F_{3,i \mapsto i+1}(\max\{x, y\})$ is already a bound for the length of the computations of this program. This can be shown as a direct application of the bound for the Termination Theorem provided in [4].

6 Conclusion

In this paper we proved that, over RCA_0,

- ISCT soundness = $\mathrm{WO}(\omega_3)$ (by Corollary 4.8 and by Theorem 5.1);
- MSCT soundness $\geq \mathrm{WO}(\omega_2)$ (see Remark 4.9).

It is known that $\mathsf{I}\Sigma^0_2$ implies $\mathrm{WO}(\omega_2)$, but does not imply $\mathrm{WO}(\omega_3)$ (see [18, Remark 2.4]). Moreover, $\mathrm{WO}(\omega_3)$ does not imply $\mathsf{I}\Sigma^0_2$ (see [18, Corollary 4.3]). Since $\mathsf{I}\Sigma^0_2+$ MSCT soundness implies $\mathrm{WO}(\omega_3)$, we can conclude that MSCT soundness $> \mathrm{WO}(\omega_2)$.

Finally, since every MSCT program is also ISCT, provably in RCA_0, we have that $\mathrm{WO}(\omega_3) \geq$ MSCT soundness. As discussed in the introduction, we leave open whether the inequality is strict (see Question 1.2). Let A^n be the generalized Péter-Ackermann function for the successor function (i.e. $f(x) = x + 1$). If we restrict MSCT soundness to the statement we call MSCT* soundness:

$$\forall n(A^n \text{ MSCT} \implies A^n \text{ terminates}),$$

then we obtain something strictly weaker than $\mathrm{WO}(\omega_3)$, as one can obtain a model which seperates them, in the following manner:

Starting from a countable, nonstandard model of PA, shorten it (i.e., take an initial segment) to a model M, such that $M \models \mathrm{WO}(\omega^{\omega^n})$ if and only if n is standard. In M, the Strong Pigeonhole Principle SPP_n fails for all nonstandard n, but is true for all standard n. So there exists a coloring in n many colors of the natural numbers for which there does not exist a set of colors which appear infinitely many times in this coloring.

This is a model such that $M \models \forall n(A^n \text{ MSCT} \implies A^n \text{ terminates})$ and $M \models \exists n(A^n \text{ does not terminate})$. Indeed, we have that A^n is MSCT only if n is standard, as A^n MSCT implies SSP_k, for $n = 2^k - 2$, as shown in the proof of Proposition 4.10.

Therefore, there is a separation between ISCT Soundness and MSCT* Soundness. This suggests that a possible direction, to address Question 1.2, could be to solve the following:

Question 6.1. Is MSCT Soundness equivalent to MSCT* Soundness?

Of course the direction from MSCT Soundness to MSCT* Soundness is trivial, as the latter one is a direct corollary of the former one. The vice versa is still open.

Funding

The work of the first author was supported by the Fundação para a Ciência e a Tecnologia [UID/MAT/04561/2013] and Centro de Matemática, Applicações Fundamentais e Investigação Operacional of Universidade de Lisboa.

The work of the fourth author is partially supported by JSPS KAKENHI (grant numbers 16K17640 and 15H03634) and JSPS Core-to-Core Program (A. Advanced Research Networks).

References

[1] Amir M. Ben-Amram. General size-change termination and lexicographic descent. In Torben Mogensen, David Schmidt, and I. Hal Sudborough, editors, *The Essence of Computation: Complexity, Analysis, Transformation. Essays Dedicated to Neil D. Jones*, volume 2566 of *Lecture Notes in Computer Science*, pages 3–17. Springer-Verlag, 2002.

[2] Peter A. Cholak, Carl G. Jockusch, and Theodore A. Slaman. On the strength of Ramsey's theorem for pairs. *Journal of Symbolic Logic*, 66(1):1–55, 2001.

[3] Chi-Tat Chong, Theodore A. Slaman, and Yue Yang. Π^1_1-conservation of combinatorial principles weaker than Ramsey's Theorem for pairs. *Advances in Mathematics*, 230:1060–1077, 2012.

[4] Diego Figueira, Santiago Figueira, Sylvain Schmitz, and Philippe Schnoebelen. Ackermannian and primitive-recursive bounds with Dickson's lemma. In *Proceedings of the 26th Annual IEEE Symposium on Logic in Computer Science, LICS 2011, June 21-24, 2011, Toronto, Ontario, Canada*, pages 269–278, 2011.

[5] Emanuele Frittaion, Matt Hendtlass, Alberto Marcone, Paul Shafer, and Jeroen Van der Meeren. Reverse mathematics, well-quasi-orders, and Noetherian spaces. *Archive for Mathematical Logic*, 55(3):431–459, 2016.

[6] Emanuele Frittaion, Silvia Steila, and Keita Yokoyama. The strength of the SCT criterion. In T. V. Gopal, Gerhard Jäger, and Silvia Steila, editors, *Theory and Applications of Models of Computation - 14th Annual Conference, TAMC 2017, Bern, Switzerland, April 20-22, 2017, Proceedings*, volume 10185 of *Lecture Notes in Computer Science*, pages 260–273, 2017.

[7] Winskel G. *The formal semantics of programming languages: an introduction.* Foundations of computing. MIT Press, 1993.

[8] Petr Hájek and Pavel Pudlák. *Metamathematics of first-order arithmetic.* Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1993.

[9] Matthias Heizmann, Neil D. Jones, and Andreas Podelski. Size-change termination and transition invariants. In *SAS 2010, Perpignan, France, September 14-16, 2010. Proceedings*, pages 22–50, 2010.

[10] Denis R. Hirschfeldt. *Slicing the truth*, volume 28 of *Lecture Notes Series. Institute for Mathematical Sciences. National University of Singapore.* World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2015. On the computable and reverse mathematics of combinatorial principles, Edited and with a foreword by Chitat Chong, Qi Feng, Theodore A. Slaman, W. Hugh Woodin and Yue Yang.

[11] Denis R. Hirschfeldt and Richard A. Shore. Combinatorial principles weaker than Ramsey's theorem for pairs. *Journal of Symbolic Logic*, 72(1):171–206, 2007.

[12] Alexander P. Kreuzer and Keita Yokoyama. On principles between Σ_1- and Σ_2-induction, and monotone enumerations. *J. Math. Log.*, 16(1):1650004, 21, 2016.

[13] Chin Soon Lee, Neil D. Jones, and Amir M. Ben-Amram. The size-change principle for program termination. In *Conference Record of POPL 2001: The 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, London, UK, January 17-19, 2001*, pages 81–92, 2001.

[14] Ludovic Patey and Keita Yokoyama. The proof-theoretic strength of Ramsey's theorem for pairs and two colors. arXiv:1601.00050, 2016.

[15] Frank Plumpton Ramsey. On a problem in formal logic. *Proc. London Math. Soc.*, 30:264–286, 1930.

[16] Stephen G. Simpson. Ordinal numbers and the Hilbert Basis Theorem. *The Journal of Symbolic Logic*, 53:961–974, 1988.

[17] Stephen G. Simpson. *Subsystems of Second Order Arithmetic*. Perspectives in Mathematical Logic. Springer-Verlag, 1999.

[18] Stephen G. Simpson. Comparing $\mathrm{WO}(\omega^\omega)$ with Σ^0_2 induction. arXiv:1508.02655, 2015.

[19] Theodore A. Slaman and Keita Yokoyama. The strength of Ramsey's theorem for pairs and arbitrary many colors. arXiv:1711.08704, 2017.

[20] Silvia Steila. An intuitionistic analysis of size-change termination. In Hugo Herbelin, Pierre Letouzey, and Matthieu Sozeau, editors, *20th International Conference on Types for Proofs and Programs, TYPES 2014, May 12-15, 2014, Paris, France*, volume 39 of *LIPIcs*, pages 288–307. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2014.

[21] William W Tait. Nested recursion. *Mathematische Annalen*, 143(3):236–250, 1961.