After several years of research activity, the informal cooperation “Computability in Europe” decided to take a more formal status at their meeting in Athens in June 2008: the Association for Computability in Europe was founded to promote the development, particularly in Europe, of computability-related science, ranging over mathematics, computer science, and applications in various natural and engineering sciences such as physics and biology, including the promotion of the study of philosophy and history of computing as it relates to questions of computability. As mentioned, this association builds on the informal network of European scientists working on computability theory that had been supporting the conference series CiE-CS over the years, and now became its new home.

The aims of the conference series remain unchanged: to advance our theoretical understanding of what can and cannot be computed, by any means of computation. Its scientific vision is broad: computations may be performed with discrete or continuous data by all kinds of algorithms, programs, and machines. Computations may be made by experimenting with any sort of physical system obeying the laws of a physical theory such as Newtonian mechanics, quantum theory or relativity. Computations may be very general, depending on the foundations of set theory; or very specific, using the combinatorics of finite structures. CiE also works on subjects intimately related to computation, especially theories of data and information, and methods for formal reasoning about computations. The sources of new ideas and methods include practical developments in areas such as neural networks, quantum computation, natural computation, molecular computation, and computational learning. Applications are everywhere, especially, in algebra, analysis and geometry, or data types and programming. Within CiE there is general recognition of the underlying relevance of computability to physics and a broad range of other sciences, providing as it does a basic analysis of the causal structure of dynamical systems.
This volume, *Mathematical Theory and Computational Practice*, comprises the proceedings of the fifth in a series of conferences of CiE, that was held at the Ruprecht-Karls-Universität Heidelberg, Germany.

The first four meetings of CiE were at the University of Amsterdam in 2005, at the University of Wales Swansea in 2006, at the University of Siena in 2007, and at the University of Athens in 2008. Their proceedings, edited in 2005 by S. Barry Cooper, Benedikt Löwe and Leen Torenvliet, in 2006 by Arnold Beckmann, Ulrich Berger, Benedikt Löwe and John V. Tucker, in 2007 by S. Barry Cooper, Benedikt Löwe and Andrea Sorbi, and in 2008 by Arnold Beckmann, Costas Dimitracopoulos, and Benedikt Löwe were published as *Springer Lecture Notes in Computer Science*, volumes 3526, 3988, 4497 and 5028, respectively.

CiE and its conferences have changed our perceptions of computability and its interface with other areas of knowledge. The large number of mathematicians and computer scientists attending those conferences had their view of computability theory enlarged and transformed: they discovered that its foundations were deeper and more mysterious, its technical development more vigorous, its applications wider and more challenging than they had known. The annual CiE conference has become a major event, and is the largest international meeting focused on computability theoretic issues. Future meetings in Ponta Delgada, Açores (2010, Portugal), Sofia (2011, Bulgaria), and Cambridge (2012, UK) are in planning. The series is coordinated by the CiE Conference Series Steering Committee consisting of Arnold Beckmann (Swansea), Paola Bonizzoni (Milan), S. Barry Cooper (Leeds), Benedikt Löwe (Amsterdam, Chair), Elvira Mayordomo (Zaragoza), Dag Normann (Oslo), and Peter van Emde Boas (Amsterdam).

The conference was based on invited tutorials and lectures, and a set of special sessions on a range of subjects; there were also many contributed papers and informal presentations. This volume contains 17 of the invited lectures and 34% of the submitted contributed papers, all of which have been refereed. There will be a number of post-conference publications, including special issues of *Annals of Pure and Applied Logic*, *Journal of Logic and Computation*, and *Theory of Computing Systems*.

The tutorial speakers were Pavel Pudlák (Prague) and Luca Trevisan (Berkeley).

The following invited speakers gave talks: Manindra Agrawal (Kanpur), Jeremy Avigad (Pittsburgh), Mike Edmunds (Cardiff, Opening Lecture), Peter Koepke (Bonn), Phokion Kolaitis (San Jose), Andrea Sorbi (Siena), Rafael D. Sorkin (Syracuse), Vijay Vazirani (Atlanta).

Six special Sessions were held:

**Algorithmic Randomness.** *Organizers*: Elvira Mayordomo (Zaragoza) and Wolfgang Merkle (Heidelberg).
*Speakers*: Laurent Bienvenu, Bjørn Kjos-Hanssen, Jack Lutz, Nikolay Vereshchagin.

**Computational Model Theory.** *Organizers*: Julia F. Knight (Notre Dame) and Andrei Morozov (Novosibirsk).
Speakers: Ekaterina Fokina, Sergey Goncharov, Russell Miller, Antonio Montalbán.

**Computation in Biological Systems — Theory and Practice.**
*Organizers:* Alessandra Carbone (Paris) and Erzsébet Csuha-Vajja (Budapest).
*Speakers:* Ion Petre, Alberto Policriti, Francisco J. Romero-Campero, David Westhead.

**Optimization and Approximation.** *Organizers:* Magnús M. Halldórsson (Reykjavik) and Gerhard Reinelt (Heidelberg).
*Speakers:* Jean Cardinal, Friedrich Eisenbrand, Harald Räcke, Marc Uetz.

**Philosophical and Mathematical Aspects of Hypercomputation.**
*Organizers:* James Ladyman (Bristol) and Philip Welch (Bristol).
*Speakers:* Tim Button, Samuel Coskey, Mark Hogarth, Oron Shagrir.

**Relative Computability.** *Organizers:* Rod Downey (Wellington) and Alexandra A. Soskova (Sofia)
*Speakers:* George Barmpalias, Hristo Ganchev, Keng Meng Ng, Richard Shore.

The conference CiE 2009 was organized by Klaus Ambos-Spies (Heidelberg), Timur Bakibayev (Heidelberg), Arnold Beckmann (Swansea), Laurent Bienvenu (Heidelberg), Barry Cooper (Leeds), Felicitas Hirsch (Heidelberg), Rupert Hölzl (Heidelberg), Thorsten Kräling (Heidelberg), Benedikt Löwe (Amsterdam), Gunther Mainhardt (Heidelberg), and Wolfgang Merkle (Heidelberg).

The Program Committee was chaired by Klaus Ambos-Spies and Wolfgang Merkle:

- Klaus Ambos-Spies (Heidelberg, Chair)
- Giorgio Ausiello (Rome)
- Andrej Bauer (Ljubljana)
- Arnold Beckmann (Swansea)
- Olivier Bournez (Palaiseau)
- Vasco Brattka (Cape Town)
- Barry Cooper (Leeds)
- Anuj Dawar (Cambridge)
- Jacques Duparc (Lausanne)
- Pascal Hitzler (Karlsruhe)
- Rosalie Iemhoff (Utrecht)
- Margarita Korovina (Novosibirsk)
- Hannes Leitgeb (Bristol)
- Daniel Leivant (Bloomington)
- Benedikt Löwe (Amsterdam)
- Giancarlo Mauri (Milan)
- Elvira Mayordomo (Zaragoza)
- Wolfgang Merkle (Heidelberg, Chair)
- Andrei Morozov (Novosibirsk)
- Dag Normann (Oslo)
- Isabel Oitavem (Lisbon)
- Luke Ong (Oxford)
- Martin Otto (Darmstadt)
- Prakash Panangaden (Montréal)
- Ivan Sokov (Sofia)
- Viggo Stoltenberg-Hansen (Uppsala)
- Peter van Emde Boas (Amsterdam)
- Jan van Leeuwen (Utrecht)
- Philip Welch (Bristol)
- Richard Zach (Calgary)

We are delighted to acknowledge and thank the following for their essential financial support: Deutsche Forschungsgemeinschaft (German Research Foundation), Ruprecht-Karls-Universität Heidelberg (Bioquant and Department of Mathematics and Computer Science), The Elsevier Foundation.
We were proud to offer the program “Women in Computability” funded by
the Elsevier Foundation as part of CiE 2009. The Steering Committee of the con-
ference series CiE-CS is concerned with the representation of female researchers
in the field of computability. The series CiE-CS has actively tried to increase fe-
male participation at all levels in the past years. Starting in 2008, our efforts are
being funded by a grant of the Elsevier Foundation under the title “Increasing
representation of female researchers in the computability community.” As part of
this program, we had another workshop, a grant scheme for female researchers,
a mentorship program, and free childcare.

The high scientific quality of the conference was possible through the con-
scientious work of the Program Committee, the special session organizers, and
the referees. We are grateful to all members of the Program Committee for their
efficient evaluations and extensive debates, which established the final program.
We also thank the following referees:

Pavel Alaev                      Marina De Vos
Eric Allender                  Pietro Di Gianantonio
Luis Antunes                   Kenny Easwaran
Argimiro Arratia              Leah Epstein
Matthias Aschenbrenner        Javier Esvaran
George Barmpalias             Peter Fejer
Verónica Becher                Fernando Ferreira
Ulrich Berger                  Ekaterina B. Fokina
Daniela Besozzi                Lance Fortnow
Laurent Bienvenu               Willem L. Fouché
Stephen Binns                  Pierre Fraigniaud
Achim Blumensath               Peter Gács
Volker Bosserhoff             Nicola Galesi
Patricia Bouyer                Hristo Ganchev
Thomas Brihaye                 Christine Gañer
Dan Browne                    Lucas Gerin
Marius Bujorianu               Guido Gherardi
Andrés Caicedo                Ilaria Giordani
Wesley Calvert                 Rob Goldstone
Riccardo Camerlo              Erich Grädel
Alessandra Carbone            Daniel Graça
Lorenzo Carlucci              Noam Greenberg
Douglas Cenzer                Marek Gutowski
Peter Cholak                  Amit Hagar
Robin Cockett                 Magnus Halldórsson
Alessandro Colombo             Valentina Harizanov
Erzsébet Csuhiaj-Varjú       Monika Heiner
Ali Dashti                Peter Hertling
Jan de Gier                   Stefan Hetzl
Gianluca Della Vedova          Denis Hirschfeldt
John Hitchcock
Mark Hogarth
Mathieu Hoyrup
Rupert Hölzl
Tseren-Onolt Ishdorj
Emil Jeřábek
Herman Jervell
Jan Johannsen
Iskander Kalimullin
Anna Kasprzik
Klaus Keimel
Viv Kendon
Gabriele Kern-Isberner
Bjørn Kjos-Hanssen
Leszek Kołodziejczyk
Thorsten Kräling
Oleg V. Kudinov
Oliver Kullmann
Geoffrey LaForte
Branimir Lambov
Dominique Larchey-Wendling
Dominique Lecomte
Gyesik Lee
Alexander Leitsch
Steffen Lempp
Alberto Leporati
Davorin Lesnik
Andrew Lewis
Angsheng Li
Maria López
Gunther Mainhardt
Pierre McKenzie
Carlo Mereghetti
Joseph S. Miller
Russell Miller
Malika More
Markus Müller
Norbert Müller
Thomas Müller
André Nies
Stela Nikolova
István Németi
Paulo Oliva
Antonio Porreca
Valeria de Paiva
Jun Pang
Arno Pauly
Gheorghe Paun
Thanases Pheidas
Thomas Piecha
Sergei Podzorov
Lucia Pomello
Petrus Potgieter
R. Ramanujam
Ramyaa Ramyaa
Rolf Rannacher
Sidney Redner
Jan Reimann
Gerhard Reinelt
Klaus Reinhardt
Fred Richman
Beatrice Riviere
Vladimir Rybakov
Dov Samet
Matthias Schröder
Peter Schuster
Monika Seisenberger
Pavel Semukhin
Stephen G. Simpson
Alla Sirokofskich
Andrea Sorbi
Alexandra Soksova
Dieter Spreen
Frank Stephan
Thomas Strahm
Alexey Stukachev
Zhuldyz Talasbaeva
Paul Taylor
Sebastiaan Terwijn
Tristan Tomala
Hideki Tsuiki
John Tucker
Nikolas Vaporis
Panayot S. Vassilevski
Giuseppe Vizzari
Paul Voda
Heribert Vollmer
Nicolai Vorobjov
Peter B. M. Vranas
Andreas Weiermann
We thank Andrej Voronkov for his EasyChair system, which facilitated the work of the Program Committee and the editors considerably.

May 2009

Klaus Ambos-Spies
Benedikt Löwe
Wolfgang Merkle
# Table of Contents

First-Order Universality for Real Programs ........................................ 1  
*Thomas Anberrée*

Skolem + Tetration Is Well-Ordered .................................................. 11  
*Mathias Barra and Philipp Gerhardy*

Structures of Some Strong Reducibilities ........................................ 21  
*David R. Bélanger*

Complexity of Existential Positive First-Order Logic .......................... 31  
*Manuel Bodirsky, Miki Hermann, and Florian Richoux*

Stochastic Programs and Hybrid Automata for (Biological) Modeling ...... 37  
*Luca Bortolussi and Alberto Policriti*

Numberings and Randomness ............................................................. 49  
*Paul Brodhead and Bjørn Kjos-Hanssen*

The Strength of the Grätzer-Schmidt Theorem .................................. 59  
*Paul Brodhead and Bjørn Kjos-Hanssen*

Hyperloops Do Not Threaten the Notion of an Effective Procedure ...... 68  
*Tim Button*

Minimum Entropy Combinatorial Optimization Problems ..................... 79  
*Jean Cardinal, Samuel Fiorini, and Guenaël Joret*

Program Self-Reference in Constructive Scott Subdomains ................... 89  
*John Case and Samuel E. Moelius III*

$\Sigma^0_1$ and $\Pi^0_1$ Equivalence Structures .................................. 99  
*Douglas Cenzer, Valentina Harizanov, and Jeffrey B. Remmel*

Immunity for Closed Sets ............................................................... 109  
*Douglas Cenzer, Rebecca Weber, and Guohua Wu*

Lower Bounds for Kernelizations and Other Preprocessing Procedures .................. 118  
*Yijia Chen, Jörg Flum, and Moritz Müller*

Infinite-Time Turing Machines and Borel reducibility ........................ 129  
*Samuel Coskey*

Cutting Planes and the Parameter Cutwidth ..................................... 134  
*Stefan Dantchev and Barnaby Martin*
Members of Random Closed Sets .................................................. 144
   David Diamondstone and Bjørn Kjos-Hanssen

Lowness for Demuth Randomness ............................................. 154
   Rod Downey and Keng Meng Ng

Graph States and the Necessity of Euler Decomposition ............... 167
   Ross Duncan and Simon Perdrix

On Stateless Multicounter Machines ....................................... 178
   ¨Omer E˘gecio˘glu and Oscar H. Ibarra

Computability of Continuous Solutions of Higher-Type Equations ...... 188
   Martín Escardó

Equivalence Relations on Classes of Computable Structures .......... 198
   Ekaterina B. Fokina and Sy-David Friedman

Fractals Generated by Algorithmically Random Brownian Motion ...... 208
   Willem L. Fouché

Computable Exchangeable Sequences Have Computable de Finetti
Measures .................................................................................. 218
   Cameron E. Freer and Daniel M. Roy

Spectra of Algebraic Fields and Subfields .................................. 232
   Andrey Frolov, Iskander Kalimullin, and Russell Miller

Definability in the Local Theory of the ω-Enumeration Degrees ...... 242
   Hristo Ganchev

Computability of Analytic Functions with Analytic Machines .......... 250
   Tobias Görtner and Günter Hotz

An Application of Martin-Löf Randomness to Effective Probability
Theory ..................................................................................... 260
   Mathieu Hoyrup and Cristóbal Rojas

Index Sets and Universal Numberings ....................................... 270
   Sanjay Jain, Frank Stephan, and Jason Teutsch

Ordinal Computability .............................................................. 280
   Peter Koepke

A Gandy Theorem for Abstract Structures and Applications to
First-Order Definability .......................................................... 290
   Oleg V. Kudinov and Victor L. Selivanov

Constructing New Aperiodic Self-simulating Tile Sets .................. 300
   Grégory Lafitte and Michael Weiss
<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationship between Kanamori-McAloon Principle and</td>
<td>310</td>
</tr>
<tr>
<td>Paris-Harrington Theorem</td>
<td></td>
</tr>
<tr>
<td><em>Gyesik Lee</em></td>
<td></td>
</tr>
<tr>
<td>The First Order Theories of the Medvedev and Muchnik Lattices</td>
<td>324</td>
</tr>
<tr>
<td><em>Andrew Lewis, André Nies, and Andrea Sorbi</em></td>
<td></td>
</tr>
<tr>
<td>Infima of <em>d.r.e.</em> Degrees</td>
<td>332</td>
</tr>
<tr>
<td><em>Jiang Liu, Shengling Wang, and Guohua Wu</em></td>
<td></td>
</tr>
<tr>
<td>A Divergence Formula for Randomness and Dimension</td>
<td>342</td>
</tr>
<tr>
<td><em>Jack H. Lutz</em></td>
<td></td>
</tr>
<tr>
<td>On Ladner’s Result for a Class of Real Machines with Restricted Use</td>
<td>352</td>
</tr>
<tr>
<td>of Constants</td>
<td></td>
</tr>
<tr>
<td><em>Klaus Meer</em></td>
<td></td>
</tr>
<tr>
<td>0''-Categorical Completely Decomposable Torsion-Free Abelian Groups</td>
<td>362</td>
</tr>
<tr>
<td><em>Alexander G. Melnikov</em></td>
<td></td>
</tr>
<tr>
<td>Notes on the Jump of a Structure</td>
<td>372</td>
</tr>
<tr>
<td><em>Antonio Montalbán</em></td>
<td></td>
</tr>
<tr>
<td>A General Representation Theorem for Probability Functions Satisfying</td>
<td>379</td>
</tr>
<tr>
<td>Spectrum Exchangeability</td>
<td></td>
</tr>
<tr>
<td><em>J.B. Paris and A. Vencovská</em></td>
<td></td>
</tr>
<tr>
<td>Stability under Strategy Switching</td>
<td>389</td>
</tr>
<tr>
<td><em>Soumya Paul, R. Ramanujam, and Sunil Simon</em></td>
<td></td>
</tr>
<tr>
<td>Computational Heuristics for Simplifying a Biological Model</td>
<td>399</td>
</tr>
<tr>
<td><em>Ion Petre, Andrzej Mizera, and Ralph-Johan Back</em></td>
<td></td>
</tr>
<tr>
<td>Functions Definable by Arithmetic Circuits</td>
<td>409</td>
</tr>
<tr>
<td><em>Ian Pratt-Hartmann and Ivo Düntsch</em></td>
<td></td>
</tr>
<tr>
<td>Survey on Oblivious Routing Strategies</td>
<td>419</td>
</tr>
<tr>
<td><em>Harald Räcke</em></td>
<td></td>
</tr>
<tr>
<td>An Approach to the Engineering of Cellular Models Based on P Systems</td>
<td>430</td>
</tr>
<tr>
<td><em>Francisco J. Romero-Campero and Natalio Krasnogor</em></td>
<td></td>
</tr>
<tr>
<td>Decidability of Sub-theories of Polynomials over a Finite Field</td>
<td>437</td>
</tr>
<tr>
<td><em>Alla Sirokofskich</em></td>
<td></td>
</tr>
<tr>
<td>Chaitin Ω Numbers and Halting Problems</td>
<td>447</td>
</tr>
<tr>
<td><em>Kohtaro Tadaki</em></td>
<td></td>
</tr>
</tbody>
</table>
Bayesian Data Integration and Enrichment Analysis for Predicting Gene Function in Malaria ................................. 457  
  *Philip M.R. Tedder, James R. Bradford, Chris J. Needham, Glenn A. McConkey, Andrew J. Bulpitt, and David R. Westhead*

Dialectica Interpretation with Fine Computational Control .............. 467  
  *Trifon Trifonov*

Algorithmic Minimal Sufficient Statistic Revisited .................... 478  
  *Nikolay Vereshchagin*

A Computation of the Maximal Order Type of the Term Ordering on Finite Multisets .................................................. 488  
  *Andreas Weiermann*

On Generating Independent Random Strings ............................. 499  
  *Marius Zimand*

**Author Index** .................................................. 509
First-Order Universality for Real Programs

Thomas Anberrée

Division of Computer Science, University of Nottingham,
199 Taikang East Road, Ningbo 315100, China

Abstract. J. Raymundo Marcial–Romero and M. H. Escardó described a functional programming language with an abstract data type \texttt{Real} for the real numbers and a non-deterministic operator \texttt{rtest} : \texttt{Real} \rightarrow \texttt{Bool}. We show that this language is universal at first order, as conjectured by these authors: all computable, first-order total functions on the real numbers are definable. To be precise, we show that each computable function \( f : \mathbb{R} \rightarrow \mathbb{R} \) we consider is the extension of the denotation \([M_f]\) of some program \( M_f : \texttt{Real} \rightarrow \texttt{Real} \), in a model based on powerdomains, described in previous work. Whereas this semantics is only an approximate one, in the sense that programs may have a denotation strictly below their true outputs, our result shows that, to compute a given function, it is in fact always possible to find a program with a faithful denotation. We briefly indicate how our proof extends to show that functions taken from a large class of computable, first-order partial functions in several arguments are definable.

Keywords: computability, real number computation, simply typed lambda-calculus, denotational semantics.

1 Introduction

We prove that all computable total functions \( f : \mathbb{R} \rightarrow \mathbb{R} \) are definable in a certain extension of PCF with a type for real numbers. The language possesses a denotational semantics in the category of bounded-complete domains and our proof proceeds by showing that every computable function under consideration has a representative in the semantics that is the denotation of some program. In this sense, \textit{definable} should be understood as \textit{definable relatively to our semantics}, which is a stronger property than the existence of a program computing a function.

The value of this result is at least twofold. Firstly, it confirms a conjecture by J. Raymundo Marcial–Romero and M. H. Escardó, who first described the language \cite{MarcialRomero}, that the language is operationally expressive enough to compute all computable, first-order and total functions. Secondly, it shows that the denotational semantics proposed for this language by the present author \cite{EscardoFPT}, which is the one we consider here, is good enough to expose the expressivity of the language at first-order. This was not obvious, as our semantics does not enjoy full adequacy with the language, in the sense that there are programs whose
denotation only approximates their behaviour. That is, the denotation of a program may be strictly below its operational semantic. Therefore, it could have been the case that some function were computed by a program but that no program computing this function had a denotation representing the function in the model. The present result shows that it is not the case.

**Organization.** For the sake of conciseness, we leave both the language and its semantics largely unspecified. Rather, we start by describing a few properties they satisfy (Section 2), from which the universality result is derived (Section 3). Some basic background in domain theory is assumed [1].

We base our proof on the fact that the supremum operator which takes a function \( f: [0, 1] \to [0, 1] \) and returns its supremum \( \sup(f) \) is definable in pure PCF, using sequences of integers to represent real numbers [10]. We use this supremum operator to provide finer and finer intervals in which lies the desired value of a given computable function (Section 3.2). This involves translating digital representations into the native type for real numbers of the extended language (Section 3.1).

## 2 The Language and Its Semantics

As just mentioned, we do not fully describe the language nor its denotational semantics. Rather, we emphasize in this section the properties from which we will derive the universality result. A complete definition of the language and the semantics are in the preceding publication [3], as well as in the author’s dissertation [2] which also includes the proofs of properties we assume here.

The language is based on PCF, a simply-typed lambda-calculus with a recursion operator and ground types \( \text{Nat} \) and \( \text{Bool} \) for the natural numbers and the boolean values \{true, false\} [9]. In addition, we assume a type \( \text{Real} \) for the real numbers. To avoid inessential technicalities in representing discrete spaces of numbers and finite products, we further assume, in the present paper, that the language has types \( \mathbb{Z} \) and \( \mathbb{Q} \) for the integers and the rational numbers, as well as product types and their associated projections and pairing functions. Hence, the types for the language are given by

\[
\sigma = \text{Real} \mid \text{Bool} \mid \text{Nat} \mid \mathbb{Z} \mid \mathbb{Q} \mid \sigma \times \sigma \mid \sigma \to \sigma.
\]

A program is a closed term of the language. The model lies in the cartesian closed category of bounded-complete, continuous, directed-complete posets, which we call bc-domains, or just domains in the present paper [1]. Each type \( \sigma \) is interpreted by a domain \([\sigma]\) and every program \( M: \sigma \) has a denotation \([M]\) in \([\sigma]\). We write \( \mathbb{R}, \mathbb{B}, \mathbb{N}, \mathbb{Z} \) and \( \mathbb{Q} \) for \([\text{Real}]\), \([\text{Bool}]\), \([\text{Nat}]\), \([\mathbb{Z}]\) and \([\mathbb{Q}]\), respectively. The real line, that is the set \( \mathbb{R} \) endowed with the standard Euclidean topology, embeds in \( \mathbb{R} \) via a continuous injection \( \eta: \mathbb{R} \to \mathbb{R} \). The other sets \{true, false\}, \( \mathbb{N}, \mathbb{Z} \) and \( \mathbb{Q} \) are seen as discrete spaces which embeds in \( \mathbb{B}, \mathbb{N}, \mathbb{Z} \) and \( \mathbb{Q} \) and we also call \( \eta \) any of these embeddings. All functions considered in
the semantics are Scott-continuous and, for example, the notation \( g : \mathbb{R} \to \mathbb{R} \) subsumes the fact that \( g \) is continuous.

A key aspect of the language is that it possesses a non-deterministic operator \( \text{rtest} : \mathbb{R} \to \text{Bool} \), whose purpose is to exhibit which of \( r < 1 \) and \( 0 < r \) hold, for any input representing some real number \( r \). In cases where \( r \in [0,1] \), either \text{true} or \text{false} is returned, non-deterministically. Via the usual \text{if} \ldots \text{then} \ldots \text{else} \ldots \text{conditional}, this non-determinism propagates to all types and leads one to seeing all programs as sets of values, and hence to interpreting types by sets of sets. So as to remain in a tamed framework for models of non-deterministic simply-typed lambda-calculi, we use Smyth powerdomains over bc-domains \([3]\).

In particular, the denotation of a program \( M : \mathbb{R} \to \mathbb{R} \) whose only possible outputs are real numbers \( r_1, \ldots, r_n \) is \( \eta(r_0) \cup \cdots \cup \eta(r_n) \). In general, Smyth powerdomains are closed under finite unions.

**Usefulness of the Smyth semantics.** It is important to bear in mind that, due to the presence of \( \text{rtest} \), the semantics is only an approximate one, because the denotation of even a total program may consist of more values than the program actually outputs. For example, for any program 0 for the number 0, the program \( \text{rtest}(0) \) only outputs \text{true}, but its denotation is \( \eta(\text{true}) \cup \eta(\text{false}) \).

Indeed, it can be proved that \( \text{rtest} \) possesses no faithful denotation in the Smyth semantics, for continuity reasons [8 Lemma 4.3]. However, if a program of ground type has a maximal denotation, then this denotation faithfully provides the unique output of the program. In particular, if \( [M] = \eta(r) \), we know that program \( M \) can only output the real number \( r \). This is why we can still rely on the semantics to show our universality result. For each function \( f : \mathbb{R} \to \mathbb{R} \) under consideration, we will define a function \( g : \mathbb{R} \to \mathbb{R} \) which is the denotation of a program \( F : \mathbb{R} \to \mathbb{R} \) and such that \( g(\eta(r)) = \eta(f(r)) \) for all \( r \in \mathbb{R} \), hence ensuring that \( F \) computes \( f \), in the sense that \( FM \) outputs \( f(r) \) whenever \( M \) outputs \( r \).

In what follows, we are not making direct use of the \( \text{rtest} \) construct, but rather of a derived operator whose denotation is the subject of the next lemma [312].

**Lemma 1.** There exists a continuous function \( \text{case} : \mathbb{R}^5 \to \mathbb{R} \) which is the denotation of some program and satisfies the following property. For all real numbers \( p, q, r \) such that \( p < q \) and all elements \( x, y \) of the domain \( \mathbb{R} \),

\[
\text{case}(\eta(p), \eta(q), \eta(r), x, y) = \begin{cases} 
  x & \text{if } r < p \\
  x \cup y & \text{if } r \in [p, q] \\
  y & \text{if } q < r.
\end{cases}
\]

The language also possesses operators \( \text{bound}_{[a,b]} : \mathbb{R} \to \mathbb{R} \), one for each pair of rational numbers \((a, b)\) with \( a < b \). A program of the form \( \text{bound}_{[a,b]}(M) \) can only reduce to programs of the form \( \text{bound}_{[a',b']}(M) \) with \( [a', b'] \subseteq [a, b] \), thus providing the mechanism through which real numbers are represented and outputted, as sequences of finer and finer nested rational intervals. The denotations \( \text{bound}_{[a,b]} : \mathbb{R} \to \mathbb{R} \) of the \( \text{bound}_{[a,b]} \) operators satisfy the following properties.
1. For every real number $r \in [a, b]$, we have $\text{bound}_{[a, b]}(\eta(r)) = \eta(r)$.

2. For every $a, b, c, d \in \mathbb{Q}$ with $a < b$ and $c < d$ and for every $x \in \mathbb{R}$,

$$\text{bound}_{[a,b]}(\text{bound}_{[c,d]}(x)) = \begin{cases} 
\text{bound}_{[a,b]\cap[c,d]}(x) & \text{if } [c, d] \cap [a, b] \neq \emptyset \\
\eta(a) & \text{if } d < a \\
\eta(b) & \text{if } b < c.
\end{cases}$$

An operator can be defined, which can be seen as a parameterized version of the $\text{bound}_{[a,b]}$ operators, where $a$ and $b$ are taken as parameters. It enjoys a useful convergence property stated in the next lemma \cite{2} Lemma 5.13.1.

**Lemma 2.** There exists a continuous function $\text{bound}'' : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ which is the denotation of some program and which satisfies the following property. For all real numbers $a$ and $b$ such that $a \leq b$ and all $x \in \mathbb{R}$, there exist finitely many intervals $[c_0, d_0], \ldots, [c_K, d_K]$ such that

$$\text{bound}''(\eta(a), \eta(b), x) = \bigcup_{0 \leq k \leq K} \text{bound}_{[c_k, d_k]}(x)$$

and $[a - 3(b - a), b + 3(b - a)] \supseteq [c_k, d_k] \supseteq [a, b]$ for each $k$. In particular, if $r \in [a, b]$, one has that $\text{bound}''(\eta(a), \eta(b), \eta(r)) = \eta(r) = \eta(\text{bound}_{[a,b]}(r))$.

Of course, the four basic operators $\times$, $\div$, $+$ and $-$ are definable, in the sense of the following lemma, which we will use implicitly throughout the next two sections.

**Lemma 3.** Let $f$ be any of the four basic binary operators over $\mathbb{R}$. There exists a function $g : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$, which is the denotation of some program of type $\text{Real} \times \text{Real} \rightarrow \text{Real}$, such that for all real numbers $r$ and $s$ for which $f(r,s)$ is defined, one has $g(\eta(r), \eta(s)) = \eta(f(r,s))$. As a consequence, the functions $\text{min}$, $\text{max}$ and the absolute value operator are definable.

The last ingredient we need to proceed is a limit operator that returns the limit of numerical sequences whose rate of convergence is bounded \cite{2} Theorem 5.17.1.

**Lemma 4.** There exists a continuous function $\text{lim} : (\mathbb{N} \rightarrow \mathbb{R}) \rightarrow \mathbb{R}$ which is the denotation of some program of type $(\text{Nat} \rightarrow \text{Real}) \rightarrow \text{Real}$ and satisfies the following property. For every sequence $(r_n)_{n \in \mathbb{N}}$ of real numbers such that $|r_n - r_{n+1}| \leq \frac{1}{2^n}$ and for every function $f \in (\mathbb{N} \rightarrow \mathbb{R})$ such that $f(\eta(n)) = \eta(r_n)$ for all $n \in \mathbb{N}$, we have that $\text{lim}(f) = \eta(\lim_{n \rightarrow \infty} (r_n))$.

From now on, we will often identify a number $r \in \mathbb{R}$ with its representation $\eta(r)$ in $\mathbb{R}$, and consider $\mathbb{R}$ to be a subset of $\mathbb{R}$. With this convention, we say that a function $g : \mathbb{R} \rightarrow \mathbb{R}$ extends a function $f : \mathbb{R} \rightarrow \mathbb{R}$ if $f(r) = g(r)$ for all $r \in \text{dom}(f)$. We use similar conventions for other types.
3 Universality at First Order

Using the lemmas of the previous section, we prove that any total, computable function \( f : \mathbb{R} \to \mathbb{R} \) is definable in the language, in the sense that there exists a program \( M : \text{Real} \to \text{Real} \) whose denotation \( \llbracket M \rrbracket : \mathbb{R} \to \mathbb{R} \) extends \( f \). To compute \( f(r) \), we work with the signed-digit representation of real numbers \( [6] \), encoded in the PCF fragment of the language. Using this representation, local suprema and infima of \( f \) near \( r \) are computed to provide approximations of \( f(r) \) (Section 3.2). A first step is to provide a translation from signed digit representations into native representations for real numbers in the full language (Section 3.1), making use of the limit operator mentioned in Lemma 4.

3.1 Signed-Digit Representation

Let \( 3 \) be the set \( \{-1, 0, 1\} \). The function \( \rho : 3^N \times \mathbb{N} \to \mathbb{R} \) mapping \( (w, n) \) to \( 2^n \sum_{i=0}^{\infty} w(i)2^{-i-1} \) is a surjection, called the *signed digit representation* of real numbers. Each element \( (w, n) \) of \( 3^N \times \mathbb{N} \) is also called a signed digit representation of \( \rho(w, n) \). In the context of our semantics, a *signed digit representation* is an ordered pair \( (v, m) \) in

\[
\tilde{R} = [([N \to Z] \times \mathbb{N}] 
\]

that agrees with some \( (w, n) \in 3^N \times \mathbb{N} \) in the sense that \( m = \eta(n) \) and \( v(\eta(i)) = \eta(w(i)) \) for all \( i \in \mathbb{N} \). We also define the type \( \tilde{\text{Real}} \) as the type interpreted by \( \tilde{R} \), namely \( \tilde{\text{Real}} = ([\text{Nat} \to \mathbb{Z}] \times \text{Nat} \).

Let us define the *partial* surjection

\[
e : \tilde{R} \to 3^N \times \mathbb{N} 
\]

\[
e(v, n) = \text{the unique} (w, n) \text{that agrees with} (v, n),
\]

defined at each \( (v, n) \) such that \( v(\eta(i)) \in \{\eta(-1), \eta(0), \eta(1)\} \) for all \( i \in \mathbb{N} \) and such that \( n = \eta(m) \) for some \( m \in \mathbb{N} \).

The following lemma provides a function \( \rho^* : \tilde{R} \to \mathbb{R} \) that translates digital representations of decimal numbers into \( \mathbb{R} \). Notice that there is no continuous retract in the reverse direction. This is why proving universality is not just a matter of transposing functions \( \tilde{R} \to \mathbb{R} \) to functions \( \mathbb{R} \to \mathbb{R} \).

**Lemma 5.** There exists a continuous function \( \rho^* : \tilde{R} \to \mathbb{R} \), which is the denotation of some program, such that the following diagram commutes on the domain of \( e \).
Proof. We have to show that there exists a program whose denotation $\rho_* : \tilde{\mathbb{R}} \to \mathbb{R}$ extends the signed digit representation $\rho$ in the following sense: for all $(v,m) \in \tilde{\mathbb{R}}$ such that $m = \eta(n)$ for some $n \in \mathbb{N}$ and $v(\eta(i)) = w(i)$ for some $w \in 3^\mathbb{N}$ and all $i \in \mathbb{N}$, it holds that $\rho_*(v,m) = \eta(\rho(w,n))$. Let $f : \mathbb{Z}^N \times \mathbb{N} \to \mathbb{R}$ be the function defined by $f(w,n) = 2^n \times \lim_{k \to \infty} (u_k)$ where the sequence $(u_k)_{k \in \mathbb{N}}$ is defined by $u_k = \sum_{i=0}^k w'(i) \times 2^{-i-1}$ and $w'(i) = \max(-1, \min(w(i),1))$. Notice that $w'(i) \in \{-1,0,1\}$ and that $w'(i) = w(i)$ if $w(i) \in \{-1,0,1\}$. Hence the sequence $u_k$ is always convergent and $f(w,n) = \rho(w,n)$ whenever $w \in 3^N$. It is easy to see that $|u_k - \lim_{k \to \infty} (u_k)| \leq \frac{1}{2^k}$, for all $k \in \mathbb{N}$ and all $w \in \mathbb{Z}^N$. From the properties of the language described in Section 2, in particular from Lemma 4 it follows that there is a program $M_f$ whose denotation $[M_f]$ extends $f$. We choose $\rho_* = [M_f]$.

\[\blacksquare\]

Computability. The signed digit representation is commonly used to perform real number computation as well as to provide a definition of computable numerical functions. There are many definitions of computability for real numbers and numerical functions in the literature. However, most of these definitions are equivalent to the one we consider here.

Definition 1. A function $\mathbb{R} \to \mathbb{R}$ is computable if it is computable by a Turing machine with alphabet $\Sigma = \{-1,0,1,\ldots\}$, using the signed digit representation on its input and output tapes (for more details, see [11,5,7]).

This definition is equivalent to PCF definability at first order using the signed-digit representation.

3.2 Total Computable Functions from $\mathbb{R}$ to $\mathbb{R}$ Are Definable

Our strategy is to obtain a definition $\mathbb{R} \to \mathbb{R}$ of a total function $f : \mathbb{R} \to \mathbb{R}$ from a PCF definition $\tilde{\mathbb{R}} \to \mathbb{R}$ of $f$. We use the fact that the supremum and infimum of any computable total function from $[-1,1]$ to $[-1,1]$ can be computed in PCF, using the signed-digit representation [10]. We proceed in two steps. Given a computable function $f : \mathbb{R} \to \mathbb{R}$, our first step is to show that the functions $\sup_f, \inf_f : \mathbb{Q} \times \mathbb{Q} \to \mathbb{R}$ that compute the supremum and infimum of $f$ at any rational interval $[p,q]$ are definable (Lemma 6). In a second step, we define a continuous function $\phi : \mathbb{R} \to \mathbb{R}$ which represents $f$ (Theorem 1). The idea is to approximate $\phi(x)$ with intervals

$$[\inf_f (q - \epsilon, q + \epsilon), \sup_f (q - \epsilon, q + \epsilon)]$$

where $q$ and $\epsilon$ are rational numbers such that $|x - q| \leq \epsilon$. The following lemma is a consequence of a result by Simpson [10].

Lemma 6. Let $f : \mathbb{R} \to \mathbb{R}$ be a computable function. Then the functions

$$\sup_f : \mathbb{Q} \times \mathbb{Q} \to \mathbb{R} \quad \text{and} \quad \inf_f : \mathbb{Q} \times \mathbb{Q} \to \mathbb{R}$$

$$\sup_f (p,q) = \sup_{r \in [p,q]} f(r) \quad \text{and} \quad \inf_f (p,q) = \inf_{r \in [p,q]} f(r)$$

defined on $\{(p,q) \mid p \leq q\}$ are definable by PCF programs of type $\mathbb{Q} \times \mathbb{Q} \to \mathbb{R}_{\text{Real}}$. 

Proof. We only consider the case of the supremum operator; the case of the infimum operator is dual. A. Simpson proved that the operator
\[
\text{sup}_{[-1, 1]} : ([-1, 1] \to [-1, 1]) \to [-1, 1]
\]
\[
\text{sup}_{[-1, 1]}(g) = \sup \{ g(r) \mid r \in [-1, 1] \},
\]
defined on total continuous functions, is definable in PCF, using the signed digit representation \([10]\). Given an interval \([p, q]\) and a continuous, total function \(g' : [p, q] \to \mathbb{R}\), the function \(g : [-1, 1] \to [-1, 1], r \mapsto g'(p + \frac{q-p}{2}(r+1))\) is also continuous and total. Furthermore, its supremum over \([-1, 1]\) is the same as the supremum of \(g'\) over \([p, q]\). It is easy to see that the operator
\[
\text{sup}' : (\mathbb{R} \to [-1, 1]) \times \mathbb{Q} \times \mathbb{Q} \to \mathbb{R}
\]
\[
\text{sup}'(g, p, q) = \text{sup}_{[-1, 1]} \left( \lambda r. g \left( p + \frac{q-p}{2}(r+1) \right) \right),
\]
which takes the supremum at \([p, q]\) of computable total functions from \(\mathbb{R}\) to \([-1, 1]\), can be defined by some PCF program, with respect to signed binary representation, using a definition of \(\text{sup}_{[-1, 1]}\). Let us use the same name for such a PCF program \(\text{sup}' : (\tilde{\text{Real}} \to \tilde{\text{Real}}) \times \mathbb{Q} \times \mathbb{Q} \to \tilde{\text{Real}}\).

In order to now compute the supremum of functions whose range is not restricted to \([-1, 1]\), we remark that, for any continuous function \(h : [p, q] \to \mathbb{R}\), one has that

1. \(\text{sup}_{[p, q]}(h) = 2 \text{sup}_{[p, q]} \left( \frac{1}{2} h \right)\) and
2. if \(\text{sup}_{[p, q]}(\lambda r. \max(-1, \min(h(r), 1))) \in (-1, 1)\) then \(\text{sup}_{[p, q]}(h) \in (-1, 1)\).

Based on these remarks, we define the PCF program
\[
\text{sup}'' : (\tilde{\text{Real}} \to \tilde{\text{Real}}) \times \mathbb{Q} \times \mathbb{Q} \to \tilde{\text{Real}}
\]
\[
\text{sup}''(h, p, q) = \text{if } \text{firstdigit}(2 \times s) = 0 \text{ then } s \text{ else } 2 \times \text{sup}'' \left( \frac{1}{2} h, p, q \right)
\]
where \(s = \text{sup}'(g_h, p, q)\) and \(g_h = \lambda x. \max(-1, \min(h(x), 1))\).

The program \(\text{firstdigit} : \tilde{\text{Real}} \to \mathbb{Z}\) is some program that takes a signed-digit representation \(s\) of a real number and must satisfy the following; it evaluates to 0 if \(s\) is of the form \(2^m \times 0.d_0d_1d_3 \ldots\) with \(m < 0\) or \(d_0 = d_1 \cdots = d_m = 0\); otherwise, for inputs \(s\) representing a real number but not of that form, \(\text{firstdigit}(s)\) evaluates to a number different from 0. The point is that, if \(\text{firstdigit}(2 \times s) = 0\), then \(s\) represents a real number belonging to the interval \((-1, 1)\). The program \(\text{sup}''\) works on the fact that, for \(n \in \mathbb{N}\) large enough, the supremum over \([p, q]\) of the function \(\frac{1}{2^n} h\) is small enough to warrant that all of its signed-digit representations \(s\) satisfy \(\text{firstdigit}(s) = 0\). Since the function \(f : \mathbb{R} \to \mathbb{R}\) is computable, it is defined by a PCF program \(\tilde{f} : \tilde{\text{Real}} \to \tilde{\text{Real}}\). The PCF program \(\text{sup}_f : \mathbb{Q} \times \mathbb{Q} \to \tilde{\text{Real}}, \text{sup}_f(p, q) = \text{sup}''(\tilde{f}, p, q)\) defines the function \(\text{sup}_f : \mathbb{Q} \times \mathbb{Q} \to \mathbb{R}\).
Theorem 1. Any total computable function $f : \mathbb{R} \to \mathbb{R}$ is definable by a program of type $\text{Real} \to \text{Real}$.

Proof. We define a continuous function $\phi : \mathbb{R} \to \mathbb{R}$ that extends $f$ and is the denotation of a program of type $\text{Real} \to \text{Real}$. For each real number $x$, the function $\phi$ works by finding some rational number $q$ such that $|x - q| \leq \epsilon$ and by approximating $f(x)$ with some interval close to $[\inf f(q-\epsilon, q+\epsilon), \sup f(q-\epsilon, q+\epsilon)]$, for $\epsilon$ smaller and smaller, using the function bound': $\mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ mentioned in Lemma 2. We fix a computable enumeration $(\epsilon_i)_{i \in \mathbb{N}}$ of $\mathbb{Q}$ and also call $q : \mathbb{N} \to \mathbb{Q}$ some corresponding extension in the model. The function $\phi : \mathbb{R} \to \mathbb{R}$ is defined by $\phi(x) = \psi(x, 1, 0)$ from the auxiliary function $\psi : \mathbb{R} \times \mathbb{Q} \times \mathbb{N} \to \mathbb{R}$,

$$
\psi(x, \epsilon, i) = \text{cases abs}(x - q_i) \leq \epsilon \rightarrow \text{bound}\left(a, b, \psi\left(x, \frac{\epsilon}{2}, 0\right)\right).
$$

abs $(x - q_i) \geq \frac{\epsilon}{2} \rightarrow \psi(x, \epsilon, i + 1)$

where $a = \inf f(q_i - \epsilon, q_i + \epsilon)$, $b = \sup f(q_i - \epsilon, q_i + \epsilon)$ and

$$
\text{cases } r \leq p \rightarrow x
$$

q $\leq r \rightarrow y$

is a notation for case($p, q, r, x, y$) (Lemma 1). Let $r$ be a real number and let us convince ourselves that $\phi(r) = f(r)$. For any real number $\epsilon > 0$, let $a_\epsilon$ and $b_\epsilon$ be the infimum and supremum of $f$ at $[r - 2\epsilon, r + 2\epsilon]$, respectively.

Let $i_\epsilon$ be the smallest natural number such that $|r - q_{i_\epsilon}| < \frac{\epsilon}{2}$. By definition of $\psi(r, \epsilon, 0)$, there exist $a_0, b_0, \ldots, a_J, b_J \in \mathbb{R}$ satisfying

$$
a_\epsilon \leq a_j \leq f(r) \leq b_j \leq b_\epsilon
$$

for each $j$, and such that

$$
\bigcup_{0 \leq j \leq J} \text{bound}\left(a_j, b_j, \psi\left(r, \frac{\epsilon}{2}, 0\right)\right) \subseteq \psi^{i_\epsilon}(r, \epsilon, 0).
$$

By Lemma 2, for each $j$, there exist finitely many intervals $[c_{j,0}, d_{j,0}], \ldots, [c_{j,K_j}, d_{j,K_j}]$, such that

$$
\text{bound}\left(a_j, b_j, \psi\left(r, \frac{\epsilon}{2}, 0\right)\right) = \bigcup_{0 \leq k \leq K_j} \text{bound}_{[c_{j,k}, d_{j,k}]} \left(\psi^{i_\epsilon}(r, \epsilon, 0)\right)
$$

and $[a_j - 3(b_j - a_j), b_j + 3(b_j - a_j)] \subseteq [c_{j,k}, d_{j,k}] \subseteq [a_j, b_j] \subseteq [f(r), f(r)]$. From this and equations 1 and 2 we obtain that there exist some intervals $[c_0, d_0], \ldots, [c_K, d_K]$ such that

$$
\bigcup_{0 \leq k \leq K} \text{bound}_{[c_k, d_k]} \left(\psi\left(r, \frac{\epsilon}{2}, 0\right)\right) \subseteq \psi^{i_\epsilon}(r, \epsilon, 0)
$$

with $[a_\epsilon - 3(b_\epsilon - a_\epsilon), b_\epsilon + 3(b_\epsilon - a_\epsilon)] \subseteq [c_k, d_k] \subseteq [f(r), f(r)]$. It is easy to prove from there that $\psi(r, 1, 0) = f(r)$, that is $\phi(r) = f(r)$. 

3.3 Generalisation to Partial Functions of Several Arguments

Using the same ideas, it is relatively easy to generalise Theorem 1 so as to encompass functions of several arguments and even many common partial functions, such as the inverse and logarithm functions. In order to apply our proof technique to partial functions \( f: \mathbb{R}^n \to \mathbb{R} \), we need to be able to find smaller and smaller compact neighbourhoods of \((r_1, r_2, \ldots, r_n) \in \text{dom}(f)\) (Such compact neighbourhoods were the intervals \([q - \epsilon, q + \epsilon]\) in the case of total functions in one argument). A sufficient condition to proceed in this manner is that \( \text{dom}(f) \) be recursively open in the sense of the following definition, which roughly says that there exists a recursive enumeration of compact neighbourhoods such that each point of \( \text{dom}(f) \) is the intersection of some of these neighbourhoods.

**Definition 2 (Recursively open sets).** Let \( q_1, \ldots, q_n \) and \( \epsilon > 0 \) be rational numbers. The closed rational ball of \( \mathbb{R}^n \) of centre \((q_1, \ldots, q_n)\) and radius \( \epsilon \) is the set \( B(q_1, \ldots, q_n, \epsilon) = \{(r_1, \ldots, r_n) \in \mathbb{R}^n \mid \max(|r_1 - q_1|, \ldots, |r_n - q_n|) \leq \epsilon\} \).

A subset \( S \) of \( \mathbb{R}^n \) is recursively open if there exists a computable function \( \nu: \mathbb{N} \to \mathbb{Q}^n \times \mathbb{Q} \) such that (1) for all \( k \in \mathbb{N} \), we have \( B(\nu(k)) \subseteq S \) and (2) for all real \( s \in S \) and all \( \epsilon > 0 \), there exists \( k \in \mathbb{N} \) such that \( B(\nu(k)) \) contains \( s \) and has a radius smaller than \( \epsilon \).

The following generalises Theorem 1 [2][3] Theorem 6.3.5].

**Theorem 2.** Let \( f \) be a computable partial function from \( \mathbb{R}^n \) to \( \mathbb{R} \) such that \( \text{dom}(f) \) is recursively open. There exists a program whose denotation extends \( f \).

4 Conclusion

To prove first-order universality of the language, we based our approach on one specific characterization of computable functions (Definition 1). There are at least two other characterizations which might have seemed a more natural choice in our context but are in fact not applicable, at least directly, in the context of our approximate semantics. Brattka [1] characterized computable real-valued functions as those belonging to a certain set of recursive relations. Among other things, this set of recursive relations must contain the relation \( \text{Ord}_\mathbb{R} := \{(x, 0) \mid x < 1\} \cup \{(x, 1) \mid x > 0\} \subseteq \mathbb{R} \times \mathbb{N} \). However, the relation \( \text{Ord}_\mathbb{R} \) cannot be represented in our semantic model. The other characterization, by Brattka and Hertling [3], makes use of “Feasible Real Random Access Machine” manipulating registers for real numbers and natural numbers with some basic operations. But again, a family \( \{<_k\}_{k \in \mathbb{N}} \) of multi-valued test operators required among the basic operations, defined by \( x <_k y = \{\text{true} \mid x < y\} \cup \{\text{false} \mid x > y - 1/(k + 1)\} \), makes it impossible to simulate those RAM machines directly in our semantic model. What would certainly be possible, however, is to use either of these two characterizations to show that all computable functions are computed by some program of the language. But the possibility is open, in principle, that such a program has a denotation strictly below its operational behaviour in
our approximate semantics. Thus, without either inspecting or modifying the constructions of the above two papers, it is not possible to apply their results to obtain our definability result, which not only states that each computable function is computed by some program, but moreover that each computable function is computed by some program whose denotation extends the function. In this sense, our denotational semantics is “close enough” to the operational semantics, albeit being only an approximate one.

Notice that we could more directly derive our results without any detour in signed-digit representations, from a supremum operator sup: \((R \to R) \to R\) that would be the denotation of some program. However, we do not know whether such an operator exists.

Acknowledgement. This work was originally part of the author’s PhD dissertation, written under the supervision of Martín Escardó at the University of Birmingham [2]. All my thanks to him.

References

Skolem + Tetration Is Well-Ordered

Mathias Barra and Philipp Gerhardy*

Dept. of Mathematics, University of Oslo, P.B. 1053, Blindern, 0316 Oslo, Norway

gorgba@math.uio.no, philipge@math.uio.no

http://folk.uio.no/georgba
http://folk.uio.no/philipge

Abstract. The problem of whether a certain set of number-theoretic functions – defined via tetration (i.e. iterated exponentiation) – is well-ordered by the majorisation relation, was posed by Skolem in 1956. We prove here that indeed it is a computable well-order, and give a lower bound $\tau_0$ on its ordinal.

1 Introduction

In this note we solve a problem posed by Thoralf Skolem in [Sk56] regarding the majorisation relation on $\mathbb{N}^N$ restricted to a certain subset $S^*$.

Definition 1 (Majorisation). Define the majorisation relation `$\preceq$' on $\mathbb{N}^N$ by:

$$f \preceq g \iff \exists N \in \mathbb{N} \forall x \geq N \ (f(x) \leq g(x)) .$$

We say that $g$ majorises $f$ when $f \preceq g$, and as usual $f \prec g \iff f \preceq g \land g \not\preceq f$. We say that $f$ and $g$ are comparable if $f \prec g$ or $f = g$ or $g \prec f$.

Hence $g$ majorises $f$ when $g$ is almost everywhere (a.e.) greater than $f$. The relation $\preceq$ is transitive and ‘almost’ anti-symmetric on $\mathbb{N}^N$; that is, we cannot have both $f \prec g$ and $g \prec f$, and $f \preceq g \land g \preceq f \Rightarrow f \equiv g$.

Given $A \subseteq \mathbb{N}^N$, one may ask whether $(A, \preceq)$ is a total order? if it is a well-order? – and if so – what is its ordinal?

In his 1956-paper An ordered set of arithmetic functions representing the least $\epsilon$-number [Sk56], Skolem introduced the class of functions $S$, defined by:

$$0, 1 \in S \text{ and } f, g \in S \Rightarrow f + g, x^f \in S .$$

In his words (our italics): ‘we use the two rules of production [which] from [...] functions $f(x)$ and $g(x)$ we build $f(x) + g(x)$’. That is, $S$ is a typical inductively defined class, or an inductive closure.

In [Sk56], the set $S$ is stratified into the hierarchy $\bigcup_{n \in \mathbb{N}} S_n$ in a natural way: For $f \in S$ define the Skolem-rank $\rho_S(f)$ of $f$ inductively by:

$$\rho_S(0) \overset{\text{def}}{=} \rho_S(1) \overset{\text{def}}{=} 0; \rho_S(f + g) \overset{\text{def}}{=} \max(\rho_S(f), \rho_S(g)) \text{ and } \rho_S(x^f) \overset{\text{def}}{=} \rho_S(f) + 1 ,$$

* Both authors are supported by a grant from the Norwegian Research Council.

© Springer-Verlag Berlin Heidelberg 2009
and define $S_n \overset{\text{def}}{=} \{ f \in S \mid \rho_S(f) \leq n \}$. So e.g. $S_0 = \{0, 1, 2, \ldots \}$ and $S_1 = \mathbb{N}[x]$.

Skolem next defined functions $\phi_0 \overset{\text{def}}{=} 1$, $\phi_{n+1} \overset{\text{def}}{=} x^{\phi_{n}}$, and it is immediate that $\rho_S(\phi_n) = n$ and $\rho_S(f) < n \leq \rho_S(g) \Rightarrow f \prec \phi_n \preceq g$.

When $(A, \preceq)$ is a well-order and $f \in A$, we let $O(A, \preceq)$ denote the ordinal/ order-type of the well-order and $O(f)$ denotes the ordinal of $f$ w.r.t. $(A, \preceq)$.

The main results from [Sk056] are summarised in a theorem below:

**Theorem A (Skolem [Sk056])**

1. If $f \in S_{n+1}$, then $f$ can be uniquely written as $\sum_{i=1}^{k} a_i f_i$ where $a_i \in N$, $f_i \in S_n$ and $f_i \succ f_{i+1}$;
2. $f, g \in S \Rightarrow f \cdot g \in S$, i.e. $S$ is closed under multiplication;
3. $\phi_{n+1} \in \overline{S}_{n+1} \overset{\text{def}}{=} S_{n+1} \setminus S_n$;
4. $(S, \preceq)$ is a well-order;
5. $O(S_n, \preceq) = \omega \cdot \omega_{n+1} = O(\phi_{n+1})$;
6. $O(S, \preceq) = \sup_{n<\omega}O(\phi_{n}) = \epsilon_0 \overset{\text{def}}{=} \min \{ \alpha \in \text{ON} \mid \omega^\alpha = \alpha \}$.

Above, ON is the class of all ordinals. \(\square\)

In the final paragraphs of [Sk056] the following problems are suggested:

**Problem 1 (Skolem [Sk056]).** Define $S^*$ by $0, 1 \in S^*$, and, if $f, g \in S^*$ then $f + g, f^g \in S^*$. Is $(S^*, \preceq)$ a well-order? If so, what is $O(S^*, \preceq)$?

For the accurate formulation of the 2. problem, we define the number-theoretic function $t(x, y)$ of *tetration* by: $t(x, y) \overset{\text{def}}{=} x_y = \begin{cases} 1 & y = 0 \\ x_{(x^{y-1})} & y > 0 \end{cases} = x \underbrace{\ldots \underbrace{\cdot x}}_y$.

**Problem 2 (Skolem [Sk056]).** Define $S_*$ by $0, 1 \in S_*$, and, if $f, g \in S_*$ then $f + g, f \cdot g, x^f, x_f \in S_*$. Is $(S_*, \preceq)$ a well-order? If so, what is $O(S_*, \preceq)$?

We remark here that with respect to asymptotic growth, $S^*$ is a ‘horizontal extension’ of $S$, while $S_*$ is a ‘vertical extension’. More precisely, let $\mathcal{E}^n$ be the $n^{\text{th}}$ Grzegorczyk-class, and set $\overline{\mathcal{E}}^{n+1} \overset{\text{def}}{=} \mathcal{E}^{n+1} \setminus \mathcal{E}^n$. Then:

$$S^* \subseteq \mathcal{E}^3 \quad \text{while} \quad S_* \subseteq \mathcal{E}^4 \quad \text{and} \quad \overline{\mathcal{E}}^4 \cap S_* \neq \emptyset.$$ 

Also $2^x \in S^* \setminus S_*$, and $x_s \in S_* \setminus S^*$, so the classes are incomparable.

Problem 1. has been subjected to extensive studies, leading to a ‘Yes!’ on the well-orderedness, and to estimates on the ordinal. We shall briefly review the relevant results below. Problem 2. – to our best knowledge – is solved for the first time here.

On Problem 1. In [Ehr73], A. Ehrenfeucht provides a positive answer to Problem 1. by combining results by J. Kruskal [Kru60] and D. Richardson [Ric69]. Richardson uses analytical properties of certain inductively defined sets of functions $A \subset \mathbb{R}^\mathbb{R}$ to show (as a corollary) that $(S^*, \preceq)$ is a *total order*. Ehrenfeucht
then gives a very basic well-partial-order $\subseteq$ on $S^*$, and invokes a deep combinatorial result by Kruskal which ensures that the total extensions of $\subseteq$ – which include $(S^*, \preceq)$ – are necessarily well-orders.

Later, in a series of papers, H. Levitz has isolated sub-orders of $(S^*, \preceq)$ with ordinal $\epsilon_0$ [Lev75] [Lev77], and he has provided the upper bound $\tau_0$ on $O(S^*, \preceq)$ [Lev78]. Here $\tau_0 \doteq \inf \{ \alpha \in \text{ON} \ | \ \epsilon_0 = \alpha \}$ Hence $\epsilon_0 \leq O(S^*, \preceq) \leq \tau_0$; a rather large gap.

On Problem 2. Since Skolem did not precisely formulate his second problem, below follows the last paragraphs of [Sko56] verbatim:

> It is natural to ask whether the theorems [of [Sko56] could be extended to the set $S^*$ of functions that may be constructed from 0 and $x$ by addition, multiplication and the general power $f(x)^{g(x)}$. However, I have not yet had the opportunity to investigate this.

> It seems probable that we will get a representation of a higher ordinal by taking the set of functions of one variable obtained by use of not only $x + y$, $xy$, and $x^y$ but also $f(x, y)$ defined by the recursion

$$f(0, y) = y, \ f(x + 1, y) = x^{f(x, y)}$$

> The difficulty will be to show the general comparability and that the set is really well ordered by the relation $\preceq$.

- Skolem [Sko56]

Exactly what he meant here is not clear, and at least two courses are suggested: One is to study the class obtained by general tetration – allowing from $f$ and $g$ the formation of $f^g$ – or to consider the class most analogous to $S$ – allowing the formation of $x^f$ only. This paper is concerned with the second interpretation.

Finally, we have included multiplication of functions as a basic rule of production for $S_*$, since we feel this best preserves the analogy with $S$. Whereas in $S$ multiplication is derivable, in $S_*$ it is not: e.g. $x_x \cdot x$ is not definable without multiplication as a primitive.

# 2 Main Results and Proofs

It is obvious that $f \preceq g \iff x^f \preceq x^g \land x_f \preceq x_g$. Secondly, an $S_*$-function $f$ belongs to $S$ iff no honest application of the rule $f \doteq x_g$ has been used, where honest means that $g \not\in \mathbb{N}$ (identically a constant). For, if $g \equiv c$, then $x_g = x_c$ which belongs in $S$, and in the sequel we will tacitly assume that in the expression ‘$x_g$’ the $g$ is not a constant. It is straightforward to show that $x_x$ majorises all functions of $S$, and that $x_x$ is $\preceq$-minimal in $\sum_x \doteq S_* \setminus S$. 
2.1 Pre-Normal- and Normal-Forms

We next prove that all functions are represented by a unique normal-form (NF).

Strictly speaking we need to distinguish between functions in $S_*$ and the terms which represent them, as different terms may define the same function. We will write $f \equiv g$ to denote syntactical identity of terms generated from the definition of $S_*$, write $f = g$ to denote extensional identity (e.g. $x_{x+1} \neq x^x$ but $x_{x+1} = x^{x^x}$), and blur this distinction when convenient. In this spirit, we will also refer to arithmetic manipulations of $S_*$-functions as rewriting.

Definition 2 (Pre-normal-form). For $f \in S$, we call the unique normal-form for $f$ from [Sko56] the Skolem normal-form (SNF) of $f$.

Let $s, t$ range over $S$. We say that a function $f \in S_*$ is in $(\Sigma \Pi)$ pre-normal-form ($(\Sigma \Pi)$-PNF) if $f \in S$ and $f$ is in SNF, or if $f \not\in S$ and $f$ is of the form $f = \sum_{i=1}^{n} \Pi_{j=1}^{n_i} f_{ij}$ where either

- $f_{ij} \equiv x_g$ where $g$ is in $\Sigma \Pi$-PNF;
- $f_{ij} \equiv x^g$ where $g$ is in PNF, $g \not\in S$, $g \equiv (\prod_{i=1}^{n_g} g_i) \neq x_h$, and $g_{n_g} \neq (s + t)$;
- $f_{ij} \equiv s$ where $s$ is in SNF, and $j = n_i$.

An $f_{ij}$ on one of the above three forms is an $S_*$-factor, a product $\Pi f_i$ of $S_*$-factors is an $S_*$-product, also called a $\Pi$-PNF. We say that $x_h$ is a tetration-factor, that $x^{\Pi g_j}$ is an exponent-factor with exponent-product $\Pi g_j$, and that $s \in S$ is a Skolem-factor.

The requirement on exponent-factors can be reformulated as: the exponent-product is not a single tetration-factor, nor is its Skolem-factor a sum.

Proposition 1. All $f \in S_*$ have a $\Sigma \Pi$-PNF.

Proof. By induction on the build-up of $f$. The induction start is obvious, and the cases $f = g + h$, $f = gh$ and $f = x_g$ are straightforward. Let $f = x^g$, and let $g = \Sigma \Pi g_i$. Then $f = x^{\Sigma \Pi g_i} = \Pi_i x^{\Pi_{s=1}^{n_i} g_{ij}}$. By hypothesis, this is a PNF for $f$ except when some exponent-product $P = \Pi_{s=1}^{n_i} g_{ij}$ is either a single factor or when $\Pi_{s=1}^{n_i} g_{ij} = P' \cdot (s + t)$. Such exponent factors can be rewritten as $x_{h+1}$ in the first case, and as $x^{P's} \cdot x^{P't}$ in the second case. \hfill \Box

Definition 3 (Normal-form). Let $f \in S_*$. We say that the PNF $\Sigma \Pi f_{ij}$ is a normal-form (NF) for $f$ if

1. $f_{ij} \geq f_{i(j+1)}$, and $f_{ij} \succ f_{i(j+1)} \Rightarrow \forall \ell \in \mathbb{N} (f_{ij} \succ (f_{i(j+1)})^\ell)$;
2. $\forall s \in S \left( \Pi_j f_{ij} \succ (\Pi_j f_{i(j+1)}j) \cdot s \right)$;
3. If $f_{ij}$ is on the form $x_h$ or $x^h$, then $h$ is in NF.

Informally NF1–NF3 says that NF’s are inherently ordered PNF’s. Proving uniqueness is thus tantamount to showing that two terms in NF are syntactically identical, lest they define different functions.

The property marked FPP below we call the finite power property.
Lemma 1 (and definition of FPP). Let $F \subseteq S_*$ be a set of comparable $S_*$-factors in NF such that

$$\forall f_1, f_2 \in F \forall \ell \in \mathbb{N} \left( f_1 \in S_* \land f_1 \succ f_2 \Rightarrow f_1 \succ (f_2)^\ell \right) .$$

(FPP)

Then all NF’s $f \equiv \Sigma \Pi f_{ij}$, $g \equiv \Sigma \Pi g_{ij}$ composed of factors from $F$ are comparable. In particular $f \succ g \iff f_{i_0j_0} \succ g_{i_0j_0}$ for the least index $(i_0j_0)$ such that $f_{ij} \neq g_{ij}$.

Proof. Let $f = \Sigma_{i=1}^{n_f} \Pi_{j=1}^{m_i} f_{ij} \neq \Sigma_{i=1}^{n_g} \Pi_{j=1}^{k_i} g_{ij} = g$ , i.e. $f$ and $g$ have distinct NF’s. Let $(i_0j_0)$ be as prescribed above, and assume w.l.o.g. that $f_{i_0j_0} \succ g_{i_0j_0}$ (comparable by hypothesis). Since $\Sigma \Pi g_{ij}$ is a NF all summands majorise later summands. Hence, for $\ell = \max \{k_i \mid i \leq n_g \}$ and $c = n_g$ we have $(g_{i_0j_0})^\ell \cdot c \geq g_{i_0j_0} \cdots g_{ik_i} + g_{i(i+1)k_{i+1}} + \cdots + g_{nk_n} \cdots g_{nk_{n_0}} g$. Clearly $f_{i_0j_0} \succ (g_{i_0j_0})^\ell \cdot c$ implies $f \succ g$.

□

Lemma 2. $f \prec g \Rightarrow \forall \ell \in \mathbb{N} \left( (x_f)^\ell \prec x_g \right)$.

I.e. (honest) tetration-factors satisfy the FPP. We skip the proof.

2.2 The Well-Order ($S_*, \preceq$)

In this section we prove our main theorem:

Main Theorem 1. ($S_*, \preceq$) is a well-order.

We establish this through the following lemmata:

Definition 4 (Tetration rank). The tetration rank, denoted $\rho_T(f)$, of a function $f \in S_*$ is defined by induction as follows:

$$\rho_T(0) \overset{\text{def}}{=} \rho_T(1) \overset{\text{def}}{=} 0 \land \rho_T(f + g) \overset{\text{def}}{=} \rho_T(fg) \overset{\text{def}}{=} \rho_T(x_f) \overset{\text{def}}{=} \max(\rho_T(f), \rho_T(g)) ,$$

$$\rho_T(x_f) \overset{\text{def}}{=} \rho_T(f) + 1 \quad (f \text{ not constant}).$$

For all $n \in \mathbb{N}$, define $S_{*,n} \overset{\text{def}}{=} \{ f \in S_* \mid \rho_T(f) \leq n \}$, and $S_{*,n+1} \overset{\text{def}}{=} S_{*,n+1} \setminus S_{*,n}$.

Clearly $S_* = \bigcup_{n \in \mathbb{N}} S_{*,n}$, and $f, g \in S_{*,n}$ implies $f + g, fg, x_f \in S_{*,n}$. Calculating the tetration-rank of any $f \in S_*$ is straightforward, and terms with different tetration-rank cannot define the same function:

Theorem 2. Let $\psi_n \in S_*$ be defined by $\psi_0 \overset{\text{def}}{=} x_x$, and $\psi_{n+1} \overset{\text{def}}{=} x_{\psi_n}$. Then $\psi_n$ is comparable to all functions in $S_*$, and $\rho_T(f) < n \leq \rho_T(g) \Rightarrow f \prec \psi_n \preceq g$.

1 Actually, the assertion remains true when $s \in S$ is substituted for $\ell \in \mathbb{N}$, but we shall not need this here.
We omit the proof for lack of space. The theorem above states that $\psi_{n+1}$ is $\preceq$-minimal in $S_\ast \setminus S_{\ast,n}$, and that $\psi_{n+1}$ majorises all $g \in S_{\ast,n}$.

The next definition is rather technical, but we will depend upon some way of ‘dissecting’ exponent-factors in order to facilitate comparison with tetration-factors.

**Definition 5 (Tower height).** Let $f = \Sigma \Pi f_{ij}$ and $\rho_T(f) = n$. Define the $n$-tower height $\tau_n(\Sigma \Pi f_{ij})$ inductively by $\tau_n(\Sigma \Pi f_{ij}) \overset{\text{def}}{=} \max_{ij}(\tau_n(f_{ij}))$, where $\tau_n$ is defined for factors by:

$$
\tau_n(f_{ij}) \overset{\text{def}}{=} \begin{cases} 
0, & \text{if } \rho_T(f_{ij}) < n \quad \text{or} \quad f_{ij} \equiv x_h, \\
\tau_n(\Pi g_k) + 1, & \text{if } \rho_T(f_{ij}) = n \quad \text{and} \quad f_{ij} \equiv x^\Pi g_k.
\end{cases}
$$

**Lemma 3.** Assume that each $f \in S_\ast,n$ has a unique NF (satisfying NF1–NF3), and that $S_{\ast,n}$ satisfies FPP (so that $(S_{\ast,n}, \preceq)$ is a well-order). Then:

1. Any two $S_{\ast,n+1}$-factors are comparable, and $S_{\ast,n+1}$ has the FPP;
2. All $S_{\ast,n+1}$-products are comparable and have a unique NF;
3. If $\Pi g_j$ is a $S_{\ast,n+1}$-product, then

$$
\exists h \in S_{\ast,n} \exists 0 < c, d \in \mathbb{N} \forall 0 < \ell \in \mathbb{N} \left( x_{h+c} \prec x^\Pi g_j \preceq (x^\Pi g_j)^\ell \preceq x^{(x^d)^{h+c-1}} \right).
$$

**Proof.** The proof is by induction on the maximum of the tower heights of the involved terms. More precisely, since all functions have a PNF, the number

$$
\min_{\tau_{n+1}}(f) \overset{\text{def}}{=} \min \{ m \in \mathbb{N} \mid \exists f = \Sigma \Pi f_{ij} (f = \Sigma \Pi f_{ij} \land \tau_{n+1}(\Sigma \Pi f_{ij}) = m) \}
$$

is well-defined for any $f \in S_{\ast,n+1}$. Given two factors, or a product of factors, when we want to prove one of the items (1)–(3), we proceed by induction on $m = \max_{i}(\min_{\tau_{n+1}}(f_i))$, where $f_1, \ldots, f_k$ are the involved factors.

In light of Theorem 2 which immediately yields the FPP for pairs of factors of different tetration rank – the proof need only deal with $S_{\ast,n+1}$-products. We note first that such products may be written in the $\Pi$-PNF $f_1 \cdots f_m \cdot P$, where $\rho_T(f_i) = n + 1$ and $\rho_T(P) \leq n$. By assumption, the part $P$ of the product has a unique NF, and is comparable to all other $S_{\ast,n}$-products and $S_{\ast,n+1}$-factors.

**Induction start ($\tau_{n+1} = 0$):** If $\tau_{n+1}(f) = 0$, each $f_j$ is a tetration factor $x_{h_j}$ for some $h_j \in S_{\ast,n}$ with $h_j$ in NF. Ordering the $h_j$ in a decreasing sequence yields a unique NF for $f_1 \cdots f_m \cdot P$ satisfying NF1–NF3 (by invoking Lemma 2). Since all factors compare, all products compare and so (1) and (2) hold.

With regard to (3), $\tau_{n+1}(\Pi_{j=1}^k g_j) = 0$ means that $\Pi_{j=1}^k g_j$ may be assumed to be a NF by (1) and (2). Because $\rho_T(g_1) = n + 1$ and $\tau_{n+1}(g_1) = 0$, we must have $g_1 \equiv x^{g_1'}$ for some $g_1' \in S_{\ast,n}$. We have

$$
x_{g_1+1} = x^{x_{g_1'}} \equiv x^{g_1} \prec x^\Pi g_j = f \preceq x^{(x_{g_1'})^k} \preceq \left(x^{(x_{g_1'})^k}\right)^\ell \preceq x^{(x_{g_1'})^{k+1}} \preceq x^{x^{(x_{g_1'})^{k+1}}}.
$$

Above the $'\preceq'$ is justified by a generalisation of the inequalities $(x^y)^k = x^{yk} \prec x^{y^2}$, the proof of which we omit for lack of space. Thus, setting $h = g_1'$, $c = 1$ and $d = 2$ we have the desired inequalities.
**Induction step** \((\tau_{n+1} = m + 1)\): As the induction start contains all the important ideas we only include a sketch. First we prove comparability of factors \((1)\), and only the case \(x_f \text{ vs. } x^{\Pi_2}\) is involved: here we rely on the \(\Pi(3)\) to obtain the FPP. Items \((2)\) and \((3)\) follow more or less as for the induction start. \(\square\)

We can now prove \textbf{THEOREM 1}.

\textbf{Proof (of THEOREM 1).} That \(S_{*,0} = S\) is well-ordered and have unique NF’s is clear, and it vacuously satisfies the FPP.

\textbf{LEMMA 2} furnishes \(S_{*,n+1}\) with the FPP. If we can produce unique \(\Sigma\Pi\)-NF’s for \(f \in S_{*,n+1}\) we are – by \textbf{LEMMA 1} and induction on \(n\) – done, as an increasing union of well orders is itself well ordered.

Now, let \(\Sigma\Pi f_{ij} = f \in S_{*,n+1}\). Since all products have a unique NF, by rewriting each of the PNF-products \(\Pi f_{ij}\) to their respective NF’s, and then rewriting \(P \cdot s + P \cdot t\) to \(P \cdot (s + t)\) where necessary – clearly an easy task given that all products are unique – before reordering the summands in decreasing order produces the desired NF. \(\square\)

It is interesting to note that the above proofs are completely constructive, and an algorithm for rewriting functions to their normal forms for comparison can be extracted easily from the proof of \textbf{LEMMA 1}.

\textbf{Theorem 3.} \((S_*, \preceq)\) is computable. \(\square\)

This stands in stark contrast to the Ehrenfeucht-Richardson-Kruskal proof(s) that \(S^*\) is well-ordered. Indeed, substituting \(S^*\) for \(S_*\) in the above theorem turns it into an open problem.

In the next section we will see that NF’s are a clear advantage when searching for the order type \(O(S_*, \preceq)\) of the well-order.

### 2.3 On the Ordinal of Normal-Forms

In this section the reader is assumed to be familiar with ordinals and their arithmetic. See eg. Sierpiński [Si65]; \(\alpha, \beta\) range over ordinals, \(\gamma\) over limits.

That \(O(x_\varphi) = \epsilon_0\) follows from \textbf{THEOREMS A\&B}. It is also obvious that \(O(f + 1) = O(f) + 1\) for all \(f \in S_\ast\), and that if \(O(f) = \alpha + 1\), then \(f = f' + 1\) for some \(f' \in S_\ast\). It follows that \(\Sigma\Pi f_{ij}\) correspond to a limit – except when \(f_{nn_n} \equiv c \in \mathbb{N}\).

When \(f \preceq g\), we let \([f, g)\) denote the segment \(\{h \in S_\ast \mid f \preceq h < g\}\), and we write \(O([f, g))\) for \(O([f, g), \preceq)\). In particular \(O(f) = O([0, f))\). Also \(f \preceq g\) implies \([0, g) = [0, f) \cup [f, g), \text{ viz. } O(g) = O([0, f)) + O([f, g))\).

**Lemma 4.** Let \(f \equiv \Pi_i^m f_i\). Then \(f \preceq h \preceq f \cdot 2 \Rightarrow \exists f' \leq f (h = f + f')\). \(\square\)

We omit the proof, and remark that this lemma \emph{cannot} be generalised to the case where \(f\) is a general function.

**Lemma 5.** Let \(g \preceq f \equiv \Pi_i f_i\). Then \((1)\) \(O(f + g) = O(f) + O(g)\). Moreover, \((2)\) \(O(\Sigma\Pi f_{ij}) = \Sigma O(\Pi f_{ij})\), and \((3)\) \(O(f \cdot n) = O(f) \cdot n\).