Ingo Lepper

Simplification Orders in Term Rewriting

Derivation Lengths, Order Types, and Computability

2001

Mathematische Logik

Simplification Orders in Term Rewriting

Derivation Lengths, Order Types, and Computability

Inaugural-Dissertation
zur Erlangung des Doktorgrades der Naturwissenschaften
im Fachbereich Mathematik und Informatik
der Mathematisch-Naturwissenschaftlichen Fakultät
der Westfälischen Wilhelms-Universität Münster

vorgelegt von
Ingo Lepper
aus Münster
– 2001 –

Typeset with LaTeX 2_ε using the document class KOMA-Script, the packages babel, AMS-LaTeX, amsthm, hyperref, graphicx, url, color, natbib, backref, booktabs, array, ragged2e, acronym, nomencl, thumbpdf, nicefrac, currvita, titletoc, epigraph, and paralist, and the additional font packages amssymb, aeguill, stmaryrd, and pifont.

The home page of this text is http://wwwmath.uni-muenster.de/logik/publ/diss/9.html. If you want to contact the author, this is possible by electronic mail via either lepper@math.uni-muenster.de (expiration date approx. 12/2004) or ingo.lepper@gmx.de.

Dekan:	Prof. Dr. W. Lange
Erster Gutachter:	Prof. Dr. A. Weiermann
Zweiter Gutachter:	Prof. Dr. W. Pohlers

Tage der mündlichen Prüfungen:

❖ Mathematische Logik:	04.12.2001
❖ Reine Mathematik:	29.11.2001
❖ Angewandte Mathematik:	03.12.2001

Tag der Promotion:	04.12.2001

Für Anke
und
meine Familie

Foreword

Appreciate true friendship,
it is for free!

I would like to mention the persons who have significantly contributed to my work. First of all I wish to express my sincere thanks to my thesis supervisor, Prof. Andreas Weiermann, for invaluable help and encouragement. He arouse my interest in term rewriting, he benevolently answered nearly all of my questions from various fields of mathematics, and in turn he asked questions which led to new or improved results. Without his support, this text would not exist.

A further direct scientific contributor is Dieter Hofbauer. Besides cofounding the research concerning derivation lengths, his constant interest has been a main influence on my work. In various discussions he helped me gain insight into Knuth–Bendix orders, and his comments on preliminary versions of some of my results led to a much improved presentation. Everyone familiar with the work of Hélène Touzet will recognize that I also owe a lot to her. Just in time, Prof. Andrei Voronkov provided a Theorem which spared myself technical trouble and caused significant improvements in both presentation and generality of a part of this thesis. My results concerning computability via Knuth–Bendix orders partially emerged from a discussion with Guillaume Bonfante.

The *Institut für mathematische Logik und Grundlagenforschung* has been of fundamental importance to my work. Its directors Prof. Justus Diller and Prof. Wolfram Pohlers provided one major part of my mathematical education. Their kind guidance and the friendly atmosphere at the institute made my work a pleasure. Considering the amount of time I have spent there, work may have been even more pleasant than it ought to be. Related to this are also many former and present members of the institute. I'd like to explicitely mention my dear friend Benjamin Blankertz, who managed to combine hospitality with a constant cheerful urge concerning the completion of my thesis, Arnold Beckmann, Wolfgang Burr, and Michael Möllerfeld, whose knowledge appears to have grown exponentially during the last years. Our secretaries Hildegard Brunstering and Martina Pfeifer, who often succeeded in confronting us mathematicians with the real life, constitute the secret heart of the institute. Looking at the students which are affiliated to the institute at the moment, I am convinced it will remain as peaceful and open-minded a place as it has been for the past decade.

On the private side, I have to commend my friends who managed, partially against my reluctance, to distract me from my work. This always turned out to be fruitful – apart from the headaches. The squirrels (*sciurus vulgaris*, the shy red kind) from my balcony have been a second constant source of amusement. In death-defying expeditions, which sometimes reached my desk, they discovered vast deposits of nuts and managed to transport and hide away large amounts of them.

I wish to express my wholehearted gratitude to all these people and animals.

Finally, there are the persons whose confidence and loving support have made my work possible. I dedicate the thesis to my love Anke and to my family.

Contents

1 Introduction

A happy surprise
is waiting for you.

Concerning the future of proof theory and complexity in the 21$^{\text{st}}$ century, the following prediction has been made by Harvey Friedman (2000):

> *"...there has recently been considerable work on the connections between proof theory and term rewriting. This will also expand in the coming century."*

This text is intended to be such a contribution. We approach complexity problems of term rewriting theory using results and techniques from proof theory.

The concept of *term rewriting* has been developed at the end of the 1960s. A *term rewriting system* (TRS) is a finite set $\mathcal{R}$ of directed equations of terms over some fixed signature Σ. The emerging *rewrite relation* $\to_{\mathcal{R}}$ is the closure of $\mathcal{R}$ under substitutions and contexts. TRSs provide a valuable tool in automated deduction, and additionally they constitute an interesting paradigm for nondeterministic computation, importing the term notion for free. It should be mentioned that it is rather common to consider also infinite sets of rules, but for our purposes this makes not much sense.

One major topic in term rewriting is *termination*. A TRS is said to terminate if all derivations induced by its rewrite relation are finite, i.e. if the rewrite relation does not admit infinite chains. Just as in computation theory, the problem of establishing termination of a given TRS is highly nontrivial.

A vast variety of techniques for proving termination has evolved. Common to these approaches is the association of some well-founded order to the rewrite relation. Of particular importance in this field are the partial orders on terms whose well-foundedness is a consequence of Kruskal's famous Tree Theorem. These orders are called *simplification orders*, and they are the key object of our studies. Prominent examples of such orders are the *multiset path order* (MPO) of Plaisted (1978), the *lexicographic path order* (LPO) of Kamin and Lévy (1980), and the *Knuth–Bendix order* (KBO) of Knuth and Bendix (1970).

A very convenient way of proving termination is by means of Σ-*algebras*. The basis of a Σ-algebra is a partial order $(P, \prec)$. For any symbol of the signature, there is also a function on P of the appropriate arity. Roughly speaking, this corresponds to a $\Sigma \cup \{<\}$-structure in first order logic. We call a Σ-algebra *monotone* if its interpreting functions are monotone. Termination of a TRS follows from its compatibility with a monotone Σ-algebra whose underlying partial order is well-founded. Any TRS compatible with a monotone Σ-algebra which also satisfies the subterm property is called *simply terminating*. It turned out that a TRS is simply terminating if and only if its rewrite relation is contained in a simplification order. Thus simple termination implies termination.

Zantema (1993, 1994) speaks of *total termination* if the monotone Σ-algebra is based on a well-order (being total by definition). This constitutes an important (proper) subclass of simple termination. As any well-order is order-isomorphic to an ordinal, it is possible to classify totally terminating TRSs according to the ordinals α carrying compatible Σ-algebras. This leads to the notion of α-*termination* and unveils a rich structure of subhierarchies. Inside the collection of ω-terminating TRSs we can further differentiate according to the functions used for interpreting the function symbols of the Σ-algebra. If we manage on linear functions, we speak of *linear termination*, and likewise *polynomial termination* indicates that polynomial functions suffice. We can extend this to faster growing functions, yielding *exponential*, *elementary*, *primitive recursive*, and *multiple recursive termination*. The general approach is of course a classification along various subrecursive hierarchies.

Suiting a question of Kreisel to the field of termination proofs in term rewriting theory, one may ask:

> What more do we know about a term rewriting system, once its simple (total, α-, ...) termination is proved?

We investigate this question in two main directions: *derivation lengths* and *computability*. Because *order types* of simplification orders are intertwined with these topics, we study them as well.

Derivation Lengths

There are mainly two functions related to the lengths of derivations. The *(depth) complexity* of a terminating TRS maps each natural number n to the length of a longest possible derivation starting with a term of *depth* bounded by n. By depth we denote the height of the tree representing the term. Closely related but a little finer is the *size complexity*, which is defined similarly, just replacing depth with *size*. The size emerges from counting the number of symbols of the term. Both depth and size complexity provide natural measures of the worst-case behavior of TRSs.

During the last years, upper bounds on size complexities have been established for TRSs terminating via the standard termination proof methods. Hofbauer (1991) observed that the interpreting functions occuring in a proof of ω-termination can be incorporated to impose an upper size complexity bound. In particular, primitive recursive termination implies primitive recursive bounds, while multiple recursive termination goes with multiple recursive bounds.

Optimal results have also been obtained for a particularly important subclass of TRSs. A *string rewriting system* (SRS) is a TRS over a signature consisting of unary symbols and one constant, which does not occur in the rules. By convention, variables are not displayed.

According to Hofbauer and Lautemann (1989), linear termination implies an exponential bound, while Geupel (1988) and Lautemann (1988) independently proved that polynomial termination yields double exponential bounds. Hofbauer (1991, 1992) observed that if termination is provable via MPO, this implies primitive recursive termination and hence primitive recursive bounds. These are also optimal for SRSs terminating via MPO. For LPO, Weiermann (1995) gave a proof of multiple recursive termination and thus imposed a multiple recursive bound. We do not need to separately consider LPO on strings, as MPO and LPO coincide there. All these results are essentially optimal.

For SRSs, termination via KBO implies, according to Hofbauer and Lautemann (1989), an (optimal) exponential complexity bound. The authors also constructed a TRS terminating via KBO whose complexity behaves like $\mathrm{Ack}(n,0)$, with Ack denoting the (2-recursive) binary *Ackermann function*. For any $a \in \mathbb{N}$, a minor extension of this TRS leads to a TRS terminating via KBO whose complexity grows slightly faster than $\mathrm{Ack}(a^n,0)$. As a uniform upper bound Hofbauer (1991, 2000) established a 4-recursive function, and independently Touzet (1997) proved a weaker result – multiple recursive termination. So far, optimal bounds for termination via KBO have not been found. One reason for this is that, in contrast to the other termination proof methods mentioned above, termination via KBO does not imply ω-termination. In fact, Zantema (1992) demonstrated the SRS $fg \to gff$ is ω^2-terminating but not α-terminating for $\alpha < \omega^2$, and its termination via KBO is quickly shown. The lack of ω-termination makes it impossible to use one of the standard termination proof methods – monotone interpretations into $\mathbb{N}$.

A thorough analysis of KBO enables us to construct compatible nonmonotone interpretations into $\mathbb{N}$. These interpretations, which are based on *fast growing functions*, suffice to show the lower bound is optimal.

Theorem. *The complexities accompanying termination via KBO are members of* $\mathrm{Ack}(2^{O(n)},0)$*, while* $\mathrm{Ack}(O(n),0)$ *contains the corresponding size complexities. Both two bounds are essentially optimal.*

The proofs are extendible to more general definitions of KBO, and a slight modification of the interpretation enables us to precisely locate termination via KBO in the α-termination hierarchy.

Theorem. *Termination via KBO implies ω^2-termination.*

As ω-termination implies ω^2-termination, we see that termination via either MPO, LPO, or KBO yields ω^2-termination. Compare this with a result of Ferreira (1995) stating that, for each $\alpha \leqslant \omega$, there is a SRS which is ω^α-terminating but not β-terminating for $\beta < \omega^\alpha$. The three standard classes of simplification orders live at the very ground of the α-termination hierarchy.

Let us turn to general simple termination. Cichon and Tahhan Bittar (1998) showed that the complexity of any simply terminating SRS is bounded by a multiple recursive function, while on the other hand Touzet (1999) constructed, for any given multiple recursive function f, a totally terminating SRS whose complexity eventually dominates f. So the upper bound is essentially optimal.

By analyzing the proof-theoretic strength of Kruskal's Tree Theorem, Weiermann (1994) (with the forerunner Rathjen and Weiermann (1993)) proved that, for any simply terminating TRS $\mathcal{R}$, there is a $<\vartheta(\Omega^\omega)$-recursive function (i.e. a function being α-recursive for some $\alpha < \vartheta(\Omega^\omega)$) which dominates the complexity of $\mathcal{R}$. The ordinal $\vartheta(\Omega^\omega)$ is called the *small Veblen ordinal*, and it is the first infinite ordinal closed under all Veblen (1908) φ functions having finite arities. To give but one example, $\alpha \mapsto \varphi(1, \alpha)$ enumerates the *epsilon numbers*, i.e. the β with $\beta = \omega^\beta$. The gigantic growth rates occuring within the $<\vartheta(\Omega^\omega)$-recursive functions are hard to grasp. Maybe it is helpful to mention that if an ordinal λ looks like $\omega^\omega \cdot \lambda'$ for some $\lambda' > 0$, then there is a λ-recursive function which eventually dominates all $<\lambda$-recursive functions. We should further mention that the multiple recursive functions correspond to the $<\omega_3$-recursive functions, with $\omega_0 := 1$ and $\omega_{n+1} := \omega^{\omega_n}$. The supremum of the ω_n is $\varepsilon_0 = \varphi(1, 0)$, the first epsilon number. Not only is ε_0 far below $\vartheta(\Omega^\omega)$, but even ordinals like $\varphi(\varepsilon_0, \varepsilon_0)$ and its relatives are.

It had been conjectured for quite a while that the upper bound of Weiermann (1994) is far too large and that multiple recursive functions make up the optimal bound. However, Touzet (1997, 1998b) constructed, for any $n \geqslant 3$, a totally terminating TRS with proper ω_n-recursive complexity, thus exhausting the $<\varepsilon_0$-recursive functions. Termination of these TRSs is still provable in first order number theory, leaving a huge gap between all known examples of simply terminating TRSs and the upper bound of Weiermann. We bridge the gap by showing that the upper bound is essentially optimal. Thus simple termination possesses a tremendous computational complexity.

Theorem. *For any $<\vartheta(\Omega^\omega)$-recursive function f there is a totally terminating TRS whose complexity eventually dominates f.*

This closes the case for problem 81 in the RTA list of open problems described by Dershowitz et al. (1995), whose revision by Dershowitz and Treinen (1998) asks what maximal complexity is attainable by simply terminating TRSs.

According to Weiermann (1994), the bound cannot be reached by a single simply terminating TRS. Thus we have to define a hierarchy of TRSs whose increasing complexities approach the bound. Since all these TRSs are totally terminating, concerning largest possible complexities total termination is as powerful as simple termination.

The basic idea of the construction is taken from Touzet (1998b) and consists in simulating the *battle of Hercules and the Hydra* from Kirby and Paris (1982) with a totally terminating TRS. Since the Hydra battle, the Hardy hierarchy, and the descent recursive functions are closely connected, we have to do so for all ordinals below $\vartheta(\Omega^\omega)$. The ϑ function is very powerful and, due to its definition involving lots of closure processes, not easy to handle by a TRS. For our purposes it seems natural to use k-ary fixed point free Veblen φ functions ψ instead of ϑ, since the absence of fixed points significantly simplifies calculations and since $\vartheta(\Omega^\omega)$ can be approached by ψ. By Δ_k we denote the first infinite ordinal which is closed under ordinal addition and the $k+1$-ary ψ, with $k > 0$. This implies $\Delta_k = \psi(1, 0, \dots, 0)$ for the $k+2$-ary ψ. By the way, Δ_2 is just the ordinal Γ_0 celebrated by Gallier (1991). According to Schmidt (1979),

$$\vartheta(\Omega^\omega) = \sup\{\Delta_k : k > 0\}$$

holds. In view of this, a canonical approach is the definition of TRSs $\mathcal{R}_k$ which simulate Hydra battles for all ordinals below Δ_k. Total termination of the $\mathcal{R}_k$ is then established using a technically smooth characterization of total termination which stems from Touzet (1998b). The $\mathcal{R}_k$ are given in a uniform manner, and for $k > l$ we may regard $\mathcal{R}_k$ as a proper extension of $\mathcal{R}_l$. Thus the $(\mathcal{R}_k)_{k\in\mathbb{N}}$ form a hierarchy of totally terminating TRSs, and the complexity of any simply terminating TRS is eventually dominated by the complexities of almost all $\mathcal{R}_k$.

Order Types and the Hardy Function Principle

One way of measuring the strength of a simplification order is to calculate its *order type*. For a well-founded partial order this is the minimal ordinal α such that the order can be embedded into α. In general, the larger the order type the more rewrite relations are contained in the order, that is, the more TRSs can be shown to terminate via this order. Thus the strength of a termination proof method consisting of a collection of simplification orders can be measured by figuring out the supremum of the occuring order types. Generally speaking, the larger the occuring ordinals are the longer derivations may occur. This measure has to be taken with a grain of salt as, due to its finitary character, a TRS may not be able to exhaust the full power of the termination proof method because most of the branches of the compatible order are too far apart for the locally operating rewrite relation.

The order types of simplification orders get rather large, and maximal values are attained by prominent representative. For MPOs the order types are already cofinal in $\vartheta(\Omega \cdot \omega)$, and the order types of LPOs exhaust $\vartheta(\Omega^\omega)$. According to Schmidt (1979), any simplification order has an order type below $\vartheta(\Omega^\omega)$. This second appearance of the small Veblen ordinal is no coincidence. We will come back to it soon.

If we consider MPO (and hence LPO) on strings, much lower ordinals occur – here ω_3 is the optimal strict upper bound. A result of de Jongh and Parikh (1977) tells us that the order types of all simplification orders on strings live below ω_3. So for strings the order types of LPOs are again maximal.

Not much has been known about the order types of KBOs. For strings, Touzet (1997) showed that the maximal attained ordinal is ω^ω. A combination of results of Hofbauer (1991) and Touzet (1997) can be used to establish $\vartheta(\Omega^3 \cdot \omega)$ as a huge upper bound for KBOs on terms. We present a full classification of the order types of KBOs and show that they have a very low bound.

Theorem. *The maximal order type of a KBO is ω^ω – even if real-valued weight functions are allowed.*

Concerning order types, KBO on terms is just as strong as KBO on strings, whereas, concerning complexities, they differ considerably.

There appears to be a subtle relation between the order type of a simplification order proving termination of a TRS and the complexity of the TRS. Cichon (1992) implicitly proposed a connection via *slow growing functions.* It was indicated that (under certain circumstances) the complexity of a simply terminating TRS is bounded from above by a slow growing function with index closely related to the order type of the simplification order used in the termination proof. We call this the *slow growing principle.* Though this principle is valid for both MPO and LPO, it does not hold in general, as it imposes multiple recursive complexity bounds on all simply terminating TRSs. As mentioned above, these bounds have been refuted by Touzet. The principle is also not (directly) valid for KBO, because the slow growing function with index ω^ω is elementary and thus grows much slower than the Ackermann function.

Touzet (1999) adapted the slow growing principle to her results and replaced the slow growing functions with *Hardy functions.* We call this the *Hardy function principle.* Buchholz et al. (1994) established that the upper bounds produced by a version of this principle are correct. Though little is known about the tightness of these bounds, the respective results mentioned so far give strong evidence of the importance of the principle. We collect them in Table 1.1 on the facing page. Here SO is an abbreviation for the class of simplification orders,

class	bound in	bound in Drec$(\cdot)$	order types
MPO(1)	Prec	$<\omega^\omega$	$<\omega_3$
MPO	Prec	$<\omega^\omega$	$<\vartheta(\Omega \cdot \omega)$
KBO(1)	$2^{O(n)}$	ω	$\leqslant\omega^\omega$
KBO	$\mathrm{Ack}(O(n), 0)$	ω^ω	$\leqslant\omega^\omega$
LPO	Mrec	$<\omega_3$	$<\vartheta(\Omega^\omega)$
SO(1)	Mrec	$<\omega_3$	$<\omega_3$
SO	Drec$(<\vartheta(\Omega^\omega))$	$<\vartheta(\Omega^\omega)$	$<\vartheta(\Omega^\omega)$

Table 1.1: *Essentially optimal bounds on size complexities and order types occuring within standard classes of simplification orders and their restrictions to strings.*

and $\mathcal{M}(1)$ indicates the members of the proof method $\mathcal{M}$ living on strings. With Prec the primitive recursive functions are indicated, while Mrec stands for the multiple recursive functions. By Drec(α) we denote the set of α-recursive functions, and the functions which are β-recursive for some $\beta < \alpha$ are collected in Drec$(<\alpha)$. As all bounds are essentially optimal, we get a comprehensive picture.

Computability and Shortest Derivation Lengths

Cichon and Lescanne (1992) considered a measure for the strength of a termination proof method which is sometimes finer than the classification of attainable derivation lengths – the classification of the functions which are *computable* via the method. A k-ary number-theoretic function f is computable by a terminating TRS, if there are a k-ary symbol F, unary (successor) symbols S and P, and constants 0 and $0'$ such that, for all $n_1, \ldots, n_k$, the unique normal form of $F(S^{n_1}0, \ldots, S^{n_k}0)$ is $P^{f(n_1,\ldots,n_k)}0'$. We say f is *computable via* the termination proof method $\mathcal{M}$ if it is computable by a TRS terminating via $\mathcal{M}$.

The distinction between the input successor S and the output successor P is important, for example when computing via polynomial termination (PT). Cichon and Lescanne (1992) established polynomial bounds on the functions computable via PT using only one successor symbol, whereas Bonfante et al. (1999) demonstrated that with two distinct successors the computable functions are exactly the functions computable (on a Turing machine) within double exponential time.

Using the above mentioned size complexity bounds of Hofbauer (1991, 1992), it is easy to see that the functions computable via MPO coincide with the primitive recursive functions, and similarly the bounds of Weiermann (1995) show computation via LPO corresponds to multiple recursion.

It has been open what functions are computable via KBO. Our result bounding the size complexity of a TRS terminating via KBO in $\mathrm{Ack}(O(n), 0)$ can be incorporated to see that the strongest we can hope for are the functions computable with timebound in $\mathrm{Ack}(O(n), 0)$, and indeed KBO possesses such computational power.

Theorem. *The functions computable via KBO and the functions computable (on a Turing machine) with timebound in* $\mathrm{Ack}(O(n), 0)$ *coincide.*

In analogy with PT, the choice of the two successor symbols is crucial. Computations beyond primitive recursion are only possible if the output successor P is the unique so-called special symbol whereas the input successor S is nonspecial. If both successor symbols are special (hence equal), then the computable functions are exactly the primitive recursive ones. Demanding both successors to be nonspecial results in linear bounds on the computed functions, and, even worse, if S is special but P is not, then there are constant bounds on the computed functions.

For the proof we transform the two rules of a TRS Hofbauer (1991) used to show the complexity of a TRS terminating via KBO may live outside of primitive recursion into a related TRS containing fifteen rules. The trick is to cut down the amount of terms a term may reduce to. For the original TRS there are derivation strategies leading quickly to normal forms. Our TRS prevents this and makes certain long derivations *linear* – each rewrite step is the only one possible.

As a by-product we prove a result concerning the *shortest* size complexity, a measure proposed by Hofbauer and Lautemann (1989). It maps a natural number n to the maximum of the lengths of shortest derivations leading from a term of size bounded by n to a normal form.

Theorem. *Any function from* $\mathrm{Ack}(O(n), 0)$ *can be majorized by the shortest size complexity of a TRS terminating via KBO.*

Thus the optimal bounds for both size complexities coincide. The same holds true for both MPO and LPO, while it is unknown for the general case of simple termination. I strongly conjecture that the TRSs $\mathcal{R}_k$ simulating very long Hydra battles can be linearized. Since this has not yet been shown, there are also no optimal results concerning computability via simple termination. However, the $\mathcal{R}_k$ can serve as the nondeterministic engine for the simulation of a timebounded Turing machine. There are various ways to define the function computed by a nondeterministic TRS. We use a standard definition of Krentel (1988), which has later been taken up by Grädel and Gurevich (1995) and by Bonfante et al. (2001). The computed value corresponds to the maximum of those normal forms of the input term which denote numbers.

Theorem. *The functions nondeterministically computable via simple termination are the* $<\vartheta(\Omega^\omega)$*-recursive functions.*

Very Long Size-controlled Derivations

Inspired by results of Harvey Friedman (see Simpson (1985)), Loebl and Matoušek (1987), and Weiermann (2000) concerning the lengths of size controlled sequences of trees related to Kruskal's Tree Theorem, we investigate a similar notion for TRSs. Consider a noncycling TRS $\mathcal{R}$. For each $f\colon \mathbb{N} \times \mathbb{N}^{+} \to \mathbb{N}$ we denote by $K_f^{\mathcal{R}}$ the function on $\mathbb{N}$ which maps n to the first N such that there is no derivation $s_1 \to_{\mathcal{R}} \ldots \to_{\mathcal{R}} s_N$ with the sizes of the s_m bounded by $f(n, m)$.

We are interested in the control function $n + r \log_2(m)$ for fixed $r \in \mathbb{R}$. An easy counting argument shows that if r is too small, then $K_f^{\mathcal{R}}$ is bounded by an exponential function. But if r is only a little larger, the enormous growth rates we encountered before are in reach. By extending the TRSs $\mathcal{R}_k$ which simulate long Hydra battles below Δ_k we get the following result.

Theorem. *Consider $f(n, m) := n + r \log_2(m)$.*

i. *If $\mathcal{R}$ is a noncycling TRS over a signature containing S symbols and if $r < \log_{S+1}(2)$, then $K_f^{\mathcal{R}}$ is bounded by an exponential function.*
ii. *Let $S > 12$. If k satisfies $1 \leqslant k \leqslant S - 12$ and if $r > \log_{S-(10+k)}(2)$, then there exists a totally terminating TRS $\mathcal{R}$ over a signature containing S symbols such that $K_f^{\mathcal{R}}$ eventually dominates all $<\!\Delta_k$-recursive functions.*

A closer look shows that growth rates beyond $<\!\varepsilon_0$-recursion are already possible with $S > 9$ symbols, provided that $r > \log_{S-8}(2)$. These results are visualized in Figure 7.1 on page 161. Even for quite low values of r, size-controlled derivations of totally terminating TRSs get as long as uncontrolled derivations, and with a growing amount of symbols the region between small and enormous growth rates gets arbitrarily small.

Road Map

The text is largely selfcontained, only basic notions like "timebounded computability by a Turing machine" are presupposed. Knowledge of ordinals, subrecursive hierarchies, term rewriting systems, and complexity classes will do no harm. The text combines various fields of mathematics and computer science. Many definitions and results which are usually left out in one field are written down because members of other fields may not be familiar with them. The informed reader is invited to skip freely over these passages, but a look at the entry "convention" in the index may be indicated. Skipping a convention might cause trouble. Unknown symbols or notations are listed (and explained) in the glossary of notation.

In Chapter 2 we present basic definitions and results. After fixing conventions and notations we consider orders, ordinals, and order types in Section 2.3. Subrecursive hierarchies, ordinal notation systems and Bachmann systems are presented in Section 2.4, while Section 2.5 contains the introduction of complexity classes and various concepts of computation. Everything here is fairly standard.

Chapter 3 is devoted to term rewriting. We first present the basics, and then we quickly treat abstract orders on terms, including Kruskal's Tree Theorem for terms and simplification orders in Section 3.2. This leads to a closer look at semantic orders as induced by interpretations and Σ-algebras in Section 3.3. A major part of the *termination hierarchy* of Zantema (1993, 1994, 1999, 2001) is presented, as well as an important Theorem of Touzet (1998b) about connections between weak total termination and total termination. Section 3.4 introduces the famous syntactic simplification orders MPO, LPO, and KBO. We present a series of new results concerning KBO. They will be used in the following chapters. In the final section of this chapter we are concerned with computability via a TRS and computability via some termination proof method. The former gets connected to computability on Turing machines, and we mention a relation between the size complexity and the computed function.

Order types of simplification orders play the leading rôle in Chapter 4. The well-known results concerning MPO, LPO, and the general case are joined by our new characterization of the order types of KBOs.

Chapter 5 may be regarded as the heart of this text. It starts with a review of complexity bounds occuring within ω-termination, including termination via MPO and LPO. Section 5.2 is devoted to KBO, it contains our new interpretation of KBO into $\mathbb{N}$, which is then used to show termination via KBO implies ω^2-termination. Simple termination in general is cared for in Section 5.3. After reviewing some results, we turn to the construction of totally terminating TRSs with longest possible derivations. This requires a lengthy and technical proof. Once this is accomplished, we go on about various principles concerning connections between order types, complexities, and subrecursive hierarchies. In Section 5.4 we utilize our freshly constructed TRSs in the context of (generalized) LPO-controlled derivations, a concept introduced by Harvey Friedman (1999). The final section of this chapter is devoted to differences between closed and open versions of syntactic simplification orders.

We treat computability via simple termination in Chapter 6. After shortly reviewing the results dealing with PT, MPO, and LPO, we turn to computability via KBO, present our new results, and say a little about shortest derivation lengths. The final section of this chapter is concerned with nondeterministic computability via simple termination.

Chapter 7 contains the construction of TRSs having very long size-controlled derivations. This affords a closer look at results from Section 5.3. Finally, a short summary is given in Chapter 8. It also contains open problems, conjectures, and proposals to further research.

Final Remarks

Before we start, a few more or less official remarks are in order.

Most of the new material is intended to also appear elsewhere. The results of Section 5.3 treating the simulation of long Hydra battles by totally terminating TRSs are contained in Lepper (1999), while Lepper (2000b) presents the complexity bounds for termination via KBO and the classification of the order types of KBOs, thus most of Sections 5.2 and 4.2. Computability via KBO as presented in Section 6.3 is the topic of Lepper (2000a). The work on the first two papers was partially supported by grant WE 2178/2–1 of the Deutsche Forschungsgemeinschaft (DFG).

Finally, the epigraphs are selected from fortune cookies* I encountered during the preparation of this text.

* This is for the German speaking connoisseur: Die Originalübersetzungen der Sprüche lauten in der Reihenfolge ihres Erscheinens:

- ❖ Schätze die wahre Freundschaft, denn sie ist kostenlos.
- ❖ Es wartet eine freudige Überraschung auf Sie.
- ❖ Das ganze Leben ist ein ewiges Wiederanfangen.
- ❖ Wer in seinen Beutel lügt, niemand als sich selbst betrügt.
- ❖ Bravo! Sie lösen im Nu Probleme.
- ❖ Du bist auf dem richtigen Weg, mach weiter!
- ❖ Ein Ding ist nicht bös, wenn man's gut versteht.
- ❖ Es lohnt sich, einen flüchtigen Gedanken zu verfolgen.
- ❖ Ihre eigenen Anstrengungen versprechen viel Erfolg.

Leider habe ich keine Gelegenheit gefunden, meinen eigentlichen Favoriten unterzubringen. Die deutsche Fassung von »Too many cooks will spoil the dinner« lautet »Zuviele Arbeiter richten wenig aus«.

2 Preliminaries

The whole life is
a steady new beginning.

This chapter contains definitions and results which will serve as a basis for everything we will consider later. After settling a few conventions in Section 2.1, we will encounter basic set theoretic definitions in Section 2.2. These are fundamental for the observations in Section 2.3, which will be concerned with partial orders and well-orders, order types, ordinals, and partial well-orders. Section 2.4 treats various sets and hierarchies known from subrecursion theory, and finally complexity classes are introduced in Section 2.5.

2.1 Conventions

Let us first settle a few conventions. Throughout this text, natural numbers are denoted by lowercase Latin letters ranging from a to d and i to q. Sometimes we also use uppercase versions of these letters, but these are usually reserved for various kinds of sets. Ordinals are denoted by the Greek letters $\alpha, \beta, \gamma, \delta, \mu$, λ, and Λ. The common names of terms are s and t, but sometimes r and u show up as well. For variables we use x, y, and z, while constants are called c or e. Functions or function symbols are represented by f, g, and h.

In combination with the Knuth–Bendix order, parts of these conventions are canceled. Instead we consider a special function symbol called i, and, though very rarely, we also denote real numbers by α, β, and γ.

Finite sequences of similar objects are abbreviated using a bar, for example $\bar{s}$ is a shortcut for $s_1, \ldots, s_n$, and $\bar{0}$ abbreviates $0, \ldots, 0$. The length of such a sequence should always be clear from the context. Empty sequences are allowed and will occur soon. If the length of a sequence is not that obvious, we indicate it as a superscript. So n^k sometimes represents k consecutive occurrences of n. This will not be mixed with exponentiation.

We occasionally violate these conventions, provided that it seems appropriate to do so.

2.2 Basic definitions

We write $A \subseteq B$ if A is a subset of B, while $A \subsetneq B$ indicates that A is a proper subset of B. The set theoretic *difference* $A \setminus B$ of A and B is $\{x \in A : x \notin B\}$.

Nonnegative integers are called *natural numbers* and get collected in $\mathbb{N}$, while the set of *real numbers* is $\mathbb{R}$. The set $\mathbb{N}^+$ contains the positive natural numbers, the positive reals make up $\mathbb{R}^+$, and $\mathbb{R}_0^+$ is the set of nonnegative reals. For $\{n, n+1, \ldots, m\}$ we also write $[n, m]$.

By $\operatorname{card}(W)$ we denote the *cardinality* (number of elements) of a set W.

The (finite ordered) *tuple* of $a_1, \ldots, a_n$ is $(a_1, \ldots, a_n)$, its *length* $|(a_1, \ldots, a_n)|$ is n. If all a_i are located in a set A, then $(a_1, \ldots, a_n)$ is a *tuple over* A. The *Cartesian product* of the sets $A_1, \ldots, A_n$ is

$$A_1 \times \cdots \times A_n := \{(a_1, \ldots, a_n) : (\forall i \in [1, n])(a_i \in A_i)\} .$$

If all the A_i coincide with A, then we write A^n for this product. Note that A^0 contains the empty tuple $(\,)$. The set $A^* := \bigcup_{n \in \mathbb{N}} A^n$ contains the tuples over A. We will extend this notion to infinite sequences in Definition 2.16.

The *disjoint union* of $A_1, \ldots, A_n$ is

$$\biguplus_{1 \leqslant i \leqslant n} A_i := \{(i, a) : 1 \leqslant i \leqslant n \wedge a \in A_i\} .$$

We assume the notions of (n-ary) *relation* (on some set A) and *function* (from A to B) to be known. For $n = 1, 2, 3$, these are called *unary*, *binary*, and *ternary*. If R is a binary relation, then we usually write $a\,R\,b$ instead of $R(a, b)$.

By $f\colon X \to Y$ we indicate that f is a function from X to Y. A function f is *number-theoretic* if we have $f\colon \mathbb{N}^k \to \mathbb{N}$ for some $k \in \mathbb{N}$. The set of functions from X to Y is denoted by ${}^X Y$. If $f\colon A \to B$ and $g\colon B \to C$, then $g \circ f$ denotes the function from A to C which maps a to $g(f(a))$. For $f\colon A \to A$, the n^{th} *iteration* $f^n\colon A \to A$ of f (applied to a) is given by

$$f^0(a) := a \quad \text{and} \quad f^{n+1}(a) := f(f^n(a)) . \tag{2.1}$$

We will make heavy use of this notation for functions of higher arities where all but one arguments are kept fixed. In these cases a “$\cdot$” indicates the free position. So, for example,

$$g(\cdot, b)^2(c) = g(g(\cdot, b)^1(c), b) = g(g(g(\cdot, b)^0(c), b), b) = g(g(c, b), b) .$$

The *factorial function* $n!$ is as usual defined by

$$0! := 1 \quad \text{and} \quad (n+1)! := (n+1) \cdot n! ,$$

while *modified subtraction* $\dot{-}$ is given by

$$n \dot{-} m := \begin{cases} n - m & \text{if } n - m \geqslant 0 \ , \\ 0 & \text{otherwise.} \end{cases}$$

For $r \in \mathbb{R}_0^+$ we denote the unique $n \in \mathbb{N}$ satisfying $n \leqslant r < n + 1$ by $\lfloor r \rfloor$, while $\lceil r \rceil$ is the unique $n \in \mathbb{N}$ with $n - 1 < r \leqslant n$. The *logarithm* of $r > 0$ with base $p > 0$ is denoted by $\log_p(r)$, thus $p^{\log_p(r)} = r$.

We are going to introduce multisets and the basic operations on them. A multiset is quite like a finite set, but multiple appearances of its elements are counted.

Definition 2.1. For a set A, a *multiset* over A is a function $M \colon A \to \mathbb{N}$ with finite $\{a \in A : M(a) \neq 0\}$. By $\mathsf{mul}(A)$ we denote the set of multisets over A. We will sometimes use the notation $\wr \ldots \wr$ for multisets, so $\wr 0, 0, 1, 2, 2, 2 \wr$ represents the multiset M satisfying $(\forall n \geqslant 3)(M(n) = 0)$ and $M(0) = 2$, $M(1) = 1$, $M(2) = 3$. The empty multiset is the function mapping each element of A to 0, and it is ambiguously denoted by $\emptyset$. It should be always clear from the surrounding symbols if the multiset $\emptyset$ is meant.

If $a \in A$ and M is a multiset, then we use $a \in M$ for $M(a) > 0$. The *union* of the multisets M and N is denoted by $M \uplus N$ and satisfies

$$(\forall a \in A)((M \uplus N)(a) = M(a) + N(a)) \ ,$$

while the notion of subset is transferable via

$$M \Subset N \ :\Longleftrightarrow \ (\forall a \in A)(M(a) \leqslant N(a)) \ .$$

By $M \setminus\!\!\setminus N$ we denote the *difference* of the multisets M and N, which is defined by $(M \setminus\!\!\setminus N)(a) := M(a) \dot{-} N(a)$.

2.3 Orders and Order types

We are now going to introduce partial orders, well-foundedness, well-orders, and ordinals. Some basic facts and various connections between well-founded orders and ordinals are collected. We further present the first version of Kruskal's Tree Theorem. The following definitions and results are mostly taken from Jech (1978) and Pohlers (1989).

If R is a binary relation on a set P, then $p_0, \ldots, p_n \, R \, q_0, \ldots, q_m$ abbreviates $(\forall i \leqslant n)(\forall j \leqslant m)(p_i \, R \, q_j)$. We extend this notion to quantification, hence, for example, $(\forall p, q, r \in P)F(p, q, r)$ represents $(\forall p \in P)(\forall q \in P)(\forall r \in P)F(p, q, r)$.

Definition 2.2. A binary relation R on a set P is

- *reflexive* if $(\forall p \in P)(p \mathrel{R} p)$,
- *irreflexive* if $(\forall p \in P)(\neg p \mathrel{R} p)$,
- *transitive* if $(\forall p, q, r \in P)((p \mathrel{R} q \wedge q \mathrel{R} r) \Rightarrow p \mathrel{R} r))$,
- *symmetrical* if $(\forall p, q \in P)(p \mathrel{R} q \Rightarrow q \mathrel{R} p)$,
- *antisymmetrical* if $(\forall p, q \in P)((p \neq q \wedge p \mathrel{R} q) \Rightarrow \neg q \mathrel{R} p)$,
- *linear* or *total* if $(\forall p, q \in P)(p \mathrel{R} q \vee p = q \vee q \mathrel{R} p)$.

We call an ordered pair $(P, \preccurlyeq)$ where $\preccurlyeq$ is a binary reflexive and transitive relation on P a *preorder* or *quasiorder*. The *strict part* of $(P, \preccurlyeq)$ is $(P, \prec)$ with

$$p \prec q \ :\Longleftrightarrow\ p \preccurlyeq q \wedge q \not\preccurlyeq p \ .$$

A *partial order* is an antisymmetrical preorder.

If $(P, \preccurlyeq)$ is a partial order, then we have

$$(\forall p, q \in P)(p \preccurlyeq q \iff p \prec q \vee p = q) \ .$$

Hence the strict part $(P, \prec)$ is irreflexive and transitive. On the other hand, if $(P, \lhd)$ is irreflexive and transitive, then $(P, \trianglelefteqslant)$ with $p \trianglelefteqslant q \ :\Longleftrightarrow\ p \lhd q \vee p = q$ is a partial order. Therefore we will frequently introduce a partial order $(P, \preccurlyeq)$ by displaying its irreflexive and transitive strict part $(P, \prec)$, and we will not hesitate to call $(P, \prec)$ a partial order. It is common practice to write $q \succ p$ for $p \prec q$.

Definition 2.3. Let $(P, \prec)$ be a partial order and $n \geqslant 1$. We say $F \colon P^n \to P$

- is *monotone in the i^{th} argument* (with $i \in [1, n]$) if

  $$F(p_1, \ldots, p_n) \succ F(q_1, \ldots, q_n)$$

  holds for all $\bar{p}, \bar{q} \in P$ which satisfy $p_i \succ q_i$ and $p_j = q_j$ for $j \neq i$,
- is *monotone* if it is monotone in all arguments,
- is *weakly monotone in the i^{th} argument* (with $i \in [1, n]$) if

  $$F(p_1, \ldots, p_n) \succcurlyeq F(q_1, \ldots, q_n)$$

  holds for all $\bar{p}, \bar{q} \in P$ which satisfy $p_i \succcurlyeq q_i$ and $p_j = q_j$ for $j \neq i$,
- is *weakly monotone* if it is weakly monotone in all arguments, and it
- has the *(weak) subterm property* if, for all $\bar{p} \in P$ and all $i \in [1, n]$ we have $F(p_1, \ldots, p_n) \succ p_i$ (respectively $F(p_1, \ldots, p_n) \succcurlyeq p_i$).

It will become apparent later why we speak of the “subterm property”.

Definition 2.4. A mapping $o\colon P \to P'$ of the partial order $(P, \prec)$ into the partial order $(P', \prec')$ satisfying

$$(\forall p, q \in P)(p \prec q \implies o(p) \prec' o(q))$$

is called an *embedding*. If o even satisfies

$$(\forall p, q \in P)(p \prec q \iff o(p) \prec' o(q)) \ ,$$

then it is *order-preserving*. Such a mapping is an *order isomorphism* if it is bijective. Two partial orders are *similar* or *order isomorphic* if there exists an order isomorphism between them.

Definition 2.5. Consider a partial order $(P, \prec)$ and $X \subseteq P$. We call $p \in P$

- a *minimum of* X if we have $p \in X$ and $(\forall q \in X)(q \not\prec p)$,
- an *upper bound of* X if we have $(\forall q \in X)(q \preccurlyeq p)$,
- the *least element of* X if we have $p \in X$ and $(\forall q \in X)(p \preccurlyeq q)$, and
- the *supremum of* X if it is the least of the upper bounds of X.

The dual notions are *maximum*, *lower bound*, *greatest element*, and *infimum*. Existence (and uniqueness) of either maximum, minimum, supremum, or infimum of X provided, we abbreviate it by $\max X$, $\min X$, $\sup X$, and $\inf X$.

We call $Y \subseteq X$ *cofinal* (in X) if $(\forall p \in X)(\exists q \in Y)(p \preccurlyeq q)$ holds.

If we consider subsets of $\mathbb{N}$, we will sometimes use the convention $\max \emptyset = 0$.

Definition 2.6. A partial order $(P, \prec)$ is

- *well-founded* if every nonempty subset of P contains a minimum, and
- a *well-order* if it is well-founded and linear.

There is a very important collection of well-orders.

Definition 2.7. A set α is an *ordinal* if

- $(\alpha, \in)$ is a well-order and
- α is a *transitive set*, which means $(\forall \beta \in \alpha)(\beta \subseteq \alpha)$ holds.

By common practice we usually identify α with $(\alpha, \in)$ and use lowercase Greek letters $\alpha, \beta, \gamma, \dots$ to denote ordinals. The *proper class*[*] of ordinals is called On. If α and β are ordinals, then we usually write $\alpha < \beta$ for $\alpha \in \beta$. We extend the convention $\max \emptyset = 0$ to sets of ordinals.

Ordinals are the canonical representatives of equivalence classes of well-orders modulo similarity.

[*] This means On is too big to be a set.

Theorem 2.8.

i. $0 := \emptyset$ *is an ordinal.*

ii. *If α is an ordinal and $b \in \alpha$, then b is an ordinal.*

iii. $(\mathsf{On}, <)$ *is a linear order (using the obvious extension of "linear order" to classes).*

iv. *For each α we have $\alpha = \{\beta : \beta < \alpha\}$.*

v. *If X is a set of ordinals, then $\bigcup X$ is an ordinal. It coincides with* $\sup X$.

vi. *For every α, the set $\alpha \cup \{\alpha\}$ is an ordinal. It is equal to* $\inf\{\beta : \beta > \alpha\}$.

vii. *Every well-order is order-isomorphic to a unique ordinal.*

There are three kinds of ordinals.

Definition 2.9. If $\alpha = \beta \cup \{\beta\}$ for some β, then α is called a *successor ordinal* or *successor*, and we define $\beta + 1 := \beta \cup \{\beta\}$. Any ordinal which is not 0 and not a successor is called a *limit ordinal* or simply a *limit.* The least limit ordinal is called ω, and the proper class of all limit ordinals is denoted by Lim. Ordinals below ω are called *finite*, the remaining are *infinite.* An ordinal is *countable* if it can be mapped one-to-one into ω, and otherwise it is *uncountable.* The first uncountable ordinal is denoted by Ω.

The ordinal ω is order-isomorphic to $(\mathbb{N}, <)$.

Lemma 2.10. *We have $\alpha \in \mathsf{Lim}$ if and only if $\alpha \neq 0$ and $(\forall \beta < \alpha)(\beta + 1 < \alpha)$.*

Ordinals provide a tool for the classification of well-founded partial orders.

Theorem 2.11. *For any well-founded partial order $(P, \prec)$ there exist a unique ordinal* $\mathrm{otype}(P, \prec)$*, called the* order type *of $(P, \prec)$, and a unique embedding* $\mathrm{otype}^P_\prec \colon P \to \mathrm{otype}(P, \prec)$ *satisfying*

$$(\forall p \in P)\left(\mathrm{otype}^P_\prec(p) = \sup\left\{\mathrm{otype}^P_\prec(q) + 1 : q \in P \wedge q \prec p\right\}\right)$$

and

$$\mathrm{otype}(P, \prec) = \sup\left\{\mathrm{otype}^P_\prec(p) + 1 : p \in P\right\}.$$

If $(P, \prec)$ is a well-order, then $\mathrm{otype}^P_\prec$ is the order isomorphism of Theorem 2.8.vii.

Definition 2.12. Let $(P, \prec)$ be a well-order. The inverse function of $\mathrm{otype}^P_\prec$ is called the *enumerating function* of $(P, \prec)$ and is denoted by $\mathrm{enum}^P_\prec$.

The notions of order type and enumerating function are transferable to classes $\mathsf{C} \subseteq \mathsf{On}$, ordered of course by $<$. We have $\mathrm{otype}(\mathsf{C}, <) = \mathsf{On}$ if and only if C is a proper class.

Lemma 2.13. *If we can embed a partial oder $(P, \prec)$ into a well-founded partial order $(P', \prec')$, then $(P, \prec)$ is well-founded, and* $\operatorname{otype}(P, \prec) \leqslant \operatorname{otype}(P', \prec')$ *holds.*

Proof. Let o be the embedding. Since any nonempty subset of P without a $\prec$-minimal element could be transformed by o into a corresponding subset of P', our $(P, \prec)$ has to be well-founded. An induction on $\operatorname{otype}^P_\prec(p)$ immediately yields $\operatorname{otype}^P_\prec(p) \leqslant \operatorname{otype}^{P'}_{\prec'}(o(p))$. □

Definition 2.14. Let $(P, \prec)$ be a partial order and $Q \subseteq P$. By $\prec^{\restriction Q}$ we denote the *restriction* of $\prec$ to Q, which is defined by

$$p \prec^{\restriction Q} q \;:\Longleftrightarrow\; p, q \in Q \wedge p \prec q \; .$$

Although this might be a bit misleading, we will sometimes write $(Q, \prec)$ instead of $(Q, \prec^{\restriction Q})$.

Lemma 2.15. *Let $(P, \prec)$ be a partial order and $Q \subseteq P$.*

i. *$(Q, \prec)$ is a partial order.*
ii. *If $(P, \prec)$ is well-founded, then so is $(Q, \prec)$, and further* $\operatorname{otype}(Q, \prec) \leqslant \operatorname{otype}(P, \prec)$ *holds.*

Proof. While (i) amounts to a trivial calculation, (ii) follows from Lemma 2.13 and the fact that the identity function on Q embeds $(Q, \prec)$ into $(P, \prec)$. □

We are going to extend finite tuples to infinite sequences. This is achieved by identifying the tuples over A we already know with functions from finite ordinals into A.

Definition 2.16. A *sequence* (over a nonempty set A) is a function whose domain is an ordinal α (and whose range is a subset of A). This α is called the *length* of the sequence. Such a sequence is usually displayed as $(a_\iota)_{\iota<\alpha}$. The sequence is *finite* if α is finite, and otherwise it is *infinite*. If β is an ordinal and A is a nonempty set, then $A^{<\beta}$ denotes the set of sequences over A having length below β.

There is an obvious isomorphism between $A^{<\omega}$ and A^*.

Well-foundedness of a partial order is usually formulated in terms of infinite descending chains.

Definition 2.17. Let $(P, \prec)$ be a partial order. A sequence $p = (p_i)_{i<\omega}$ over P is an *infinite chain in P* if $(\forall i < \omega)(p_i \prec p_{i+1})$, and it is an *infinite descending chain in P* if $(\forall i < \omega)(p_i \succ p_{i+1})$.

Using the axiom of choice for one direction we get the announced equivalence.

Lemma 2.18. *A partial order is well-founded if and only if it does not admit an infinite descending chain.*

Corollary 2.19. *Any partial order $(P, \prec)$ with finite P is well-founded.*

Lemma 2.20. *Let $(P, \prec)$ be a well-order. Any monotone $F \colon P^n \to P$ has the weak subterm property.*

Proof. It suffices to consider the case $n = 1$. Suppose for a contradiction that we can find a p satisfying $p \succ F(p)$. An iterated application of the fact that F is an embedding leads (relying on the linearity and transitivity of $(P, \prec)$) to an infinite descending chain $p \succ F(p) \succ F(F(p)) \succ \dots$, and this contradicts, via Lemma 2.18, the well-foundedness of $(P, \prec)$. □

Similar to the way one recursively defines functions with domain $\mathbb{N}$ we can define functions with domain On by *transfinite recursion.*

Theorem 2.21. *If G is a (class) function whose domain contains all sequences, then there is a unique function F with domain On such that, for all ordinals α, $F(\alpha) = G((F(\iota))_{\iota<\alpha})$ holds.*

Definition 2.22. Let $F \colon \mathsf{On} \to \mathsf{On}$ be given. We say

- F is *continuous* if it satisfies

$$(\forall \lambda \in \mathsf{Lim})(F(\lambda) = \sup\{F(\alpha) : \alpha < \lambda\}) ,$$

- F is *normal* if it is a continuous embedding, and
- α is a *fixed point* of F if $F(\alpha) = \alpha$ holds.

Lemma 2.23. *A normal function has arbitrarily large fixed points.*

Proof. Let F be a normal function and pick an arbitrary β. We recursively define $G \colon \omega \to \mathsf{On}$ by $G(0) := \beta$ and $G(n+1) := F(G(n))$. Our intention is to show $\alpha := \sup\{G(n) : n < \omega\}$ is a fixed point. From Lemma 2.20 we infer $G(n+1) = F(G(n)) \geqslant G(n)$, and $\alpha \geqslant \beta$ follows. If there is an n satisfying $G(n+1) = G(n)$, then we get $\alpha = G(n) = F(G(n))$, and otherwise α is a limit ordinal and we have

$$F(\alpha) = F(\sup_{n<\omega} G(n)) = \sup_{n<\omega} F(G(n)) = \sup_{n<\omega} G(n+1) = \alpha$$

since F is continuous. □

A very common way of defining ordinal functions is by distinguishing between the three kinds of ordinals. This procedure is legalized by Theorem 2.21.

Proposition 2.24. *For functions* $G\colon \mathsf{On} \to \mathsf{On}$ *and* $H\colon \mathsf{On}^2 \to \mathsf{On}$ *there exists* $F\colon \mathsf{On}^2 \to \mathsf{On}$ *satisfying*

$$F(\alpha,\beta) = \begin{cases} G(\alpha) & \text{if } \beta = 0\ , \\ H(F(\alpha,\beta'),\alpha) & \text{if } \beta = \beta' + 1\ , \\ \sup\{F(\alpha,\beta') : \beta' < \beta\} & \text{if } \beta \in \mathsf{Lim}\ . \end{cases}$$

Definition 2.25. The (binary) *ordinal addition* $\alpha + \beta$ is defined by

$$\alpha + \beta := \begin{cases} \alpha & \text{if } \beta = 0\ , \\ (\alpha + \beta') + 1 & \text{if } \beta = \beta' + 1\ , \\ \sup\{\alpha + \beta' : \beta' < \beta\} & \text{if } \beta \in \mathsf{Lim}\ . \end{cases}$$

Likewise, *ordinal multiplication* $\alpha \cdot \beta$ is generated with

$$\alpha \cdot \beta := \begin{cases} 0 & \text{if } \beta = 0\ , \\ (\alpha \cdot \beta') + \alpha & \text{if } \beta = \beta' + 1\ , \\ \sup\{\alpha \cdot \beta' : \beta' < \beta\} & \text{if } \beta \in \mathsf{Lim}\ . \end{cases}$$

Finally, *ordinal exponentiation* α^β is given by

$$\alpha^\beta := \begin{cases} 1 & \text{if } \beta = 0\ , \\ \alpha^{\beta'} \cdot \alpha & \text{if } \beta = \beta' + 1\ , \\ \sup\{\alpha^{\beta'} : 0 < \beta' < \beta\} & \text{if } \beta \in \mathsf{Lim}\ . \end{cases}$$

It is easy to see that these functions extend the usual functions on natural numbers to ordinals.

Lemma 2.26. *Let* α, β, γ *be ordinals.*

i. *Ordinal addition is associative, but for* $\alpha \geqslant \omega$ *we have* $1 + \alpha = \alpha < \alpha + 1$.
ii. *Ordinal multiplication is associative, but* $2 \cdot \omega = \omega < \omega + \omega = \omega \cdot 2$.
iii. *We have* $\alpha \cdot (\beta + \gamma) = \alpha \cdot \beta + \alpha \cdot \gamma$, *but* $(\omega + 1) \cdot 2 = \omega \cdot 2 + 1 < \omega \cdot 2 + 2$.
iv. *We have* $\alpha^\beta \cdot \alpha^\gamma = \alpha^{\beta+\gamma}$.
v. *The ordinal functions* $\delta \mapsto \alpha + \delta$ *(with arbitrary* α*),* $\delta \mapsto \alpha \cdot \delta$ *(with* $\alpha > 0$*), and* $\delta \mapsto \alpha^\delta$ *(with* $\alpha > 1$*) are normal.*

By Lemma 2.26.v and Lemma 2.23, there are arbitrarily large ordinals λ satisfying $\lambda = \omega^\lambda$. Similarly, there are arbitrarily large λ which are closed under addition, i.e. they satisfy $(\forall \alpha, \beta < \lambda)(\alpha + \beta < \lambda)$.

Definition 2.27.

- Ordinals $\lambda > 0$ which are closed under addition are called *principal ordinals.* They are collected in the proper class H.[†]
- Ordinals λ satisfying $\lambda = \omega^{\lambda}$ are called *epsilons* and collected in the proper class E. The α^{th} epsilon number is called ε_{α}.
- We introduce *ω-towers* ω_n by $\omega_0 := 1$ and $\omega_{n+1} := \omega^{\omega_n}$.

Lemma 2.28.

i. *The enumerating function of* H *is* $\alpha \mapsto \omega^{\alpha}$.

ii. *We have* $\varepsilon_0 = \sup\{\omega_n : n \in \omega\}$.

Proposition 2.29. *For every ordinal α there are uniquely determined principal ordinals $\alpha_1 \geqslant \ldots \geqslant \alpha_n$ such that $\alpha = \alpha_1 + \cdots + \alpha_n$ holds. This is called the* additive normal form *of α, and we sometimes write $\alpha =_{\mathrm{NF}} \alpha_1 + \cdots + \alpha_n$.*

Note that the sum may be empty, yielding 0. We may combine the Proposition with Lemma 2.28.i.

Corollary 2.30. *For every ordinal α there are uniquely determined ordinals $\alpha_1 \geqslant \ldots \geqslant \alpha_n$ such that $\alpha = \omega^{\alpha_1} + \cdots + \omega^{\alpha_n}$ holds. This is called the* Cantor normal form *of α, and we sometimes write $\alpha =_{\mathrm{CNF}} \omega^{\alpha_1} + \cdots + \omega^{\alpha_n}$.*

Based on the additive normal form, it is possible to define an alternative ordinal addition which is associative and commutative.

Definition 2.31. The *natural sum* $\alpha \oplus \beta$ of two ordinals $\alpha =_{\mathrm{NF}} \gamma_1 + \cdots + \gamma_n$ and $\beta =_{\mathrm{NF}} \gamma_{n+1} + \cdots + \gamma_{n+m}$ is given by $\alpha \oplus \beta := \gamma_{\pi(1)} + \cdots + \gamma_{\pi(n+m)}$, where π is any permutation of $[1, n+m]$ with $(\forall i \in [1, n+m-1])(\gamma_{\pi(i)} \geqslant \gamma_{\pi(i+1)})$.

Lemma 2.32. *The natural sum is commutative, associative, and monotone.*

Definition 2.33. The operation $*$ *concatenates* two sequences $a = (a_{\iota})_{\iota<\alpha}$ and $b = (b_{\iota})_{\iota<\beta}$. By $a * b$ we denote the sequence $c = (c_{\iota})_{\iota<\alpha+\beta}$ satisfying

$$c_{\iota} = \begin{cases} a_{\iota} & \text{if } \iota < \alpha \;, \\ b_{\xi} & \text{if } \iota = \alpha + \xi \;. \end{cases}$$

We say $a' = (a'_{\iota})_{\iota<\alpha'}$ is an *extension* of $a = (a_{\iota})_{\iota<\alpha}$, abbreviated by $a \leqslant_{\mathsf{ext}} a'$, if $\alpha \leqslant \alpha'$ and $(\forall \iota < \alpha)(a_{\iota} = a'_{\iota})$.

Lemma 2.34. *The class of sequences is partially ordered by $\leqslant_{\mathsf{ext}}$. We have $a \leqslant_{\mathsf{ext}} a'$ if and only if there is a sequence b with $a' = a * b$.*

[†] H stems from the German translation *Hauptzahlen.*

Definition 2.35. The *concatenation* of partial orders $(P_1, \prec_1), \dots, (P_n, \prec_n)$ is $(\biguplus_{1 \leqslant i \leqslant n} P_i, \prec^{1,n})$ with

$$(i, p) \prec^{1,n} (j, q) \ :\Longleftrightarrow\ i < j \vee (i = j \wedge p \prec_i q)\ .$$

Lemma 2.36.

i. *The concatenation of (linear) partial orders is a (linear) partial order.*

ii. *The concatenation of well-founded partial orders is a well-founded partial order, and its order type is the sum of the order types of the basic orders.*

iii. *The concatenation of well-orders is a well-order.*

Definition 2.37.

❖ The *lexicographic product* of the partial orders $(P_1, \prec_1), \dots, (P_n, \prec_n)$ is $(P_1 \times \cdots \times P_n, \prec_{\mathsf{lex}}^{1,n})$ where $(p_1, \dots, p_n) \prec_{\mathsf{lex}}^{1,n} (q_1, \dots, q_n)$ holds if

$$(\exists i \in [1, n])(p_i \prec_i q_i \wedge (\forall j \in [1, i-1])(p_j = q_j))\ .$$

❖ If all $(P_k, \prec_k)$ coincide with $(P, \prec)$, we write $\prec_{\mathsf{lex}}^{n}$ for $\prec_{\mathsf{lex}}^{1,n}$ and call the resulting $(P^n, \prec_{\mathsf{lex}}^{n})$ the *n-fold* lexicographic product.

❖ The *lexicographic order* $(P^*, \prec_{\mathsf{lex}}^{*})$ based on a partial order $(P, \prec)$ is defined by

$$p \prec_{\mathsf{lex}}^{*} q \ :\Longleftrightarrow\ |p| < |q| \vee (|p| = |q| \wedge p \prec_{\mathsf{lex}}^{|p|} q)\ .$$

If two sequences of equal lengths are considered, we will often write $<_{\mathsf{lex}}$ instead of $<_{\mathsf{lex}}^{n}$ or $\prec_{\mathsf{lex}}^{*}$.

Proposition 2.38.

i. *The lexicographic product of (linear) partial orders is a (linear) partial order.*

ii. *The lexicographic product of well-founded partial orders is a well-founded partial order, and its order type is the reverse product of the order types of the basic orders:*

$$\mathrm{otype}(P_1 \times \cdots \times P_n, \prec_{\mathsf{lex}}^{1,n}) = \mathrm{otype}(P_n, \prec_n) \cdot \cdots \cdot \mathrm{otype}(P_1, \prec_1)\ .$$

iii. *The lexicographic product of well-orders is a well-order.*

iv. *The lexicographic order based on a (linear) partial order is a (linear) partial order.*

v. *The lexicographic order based on a well-founded partial order is a well-founded partial order.*

vi. *The lexicographic order based on a well-order is a well-order.*

Lemma 2.39. *If* $(P, \prec)$ *is a well-founded partial order and* $\mathrm{otype}(P, \prec) \geqslant \omega$*, then* $\mathrm{otype}(P^*, \prec^*_{\mathsf{lex}}) = \mathrm{otype}(P, \prec)^\omega$*.*

Proof. We put $\alpha := \mathrm{otype}(P, \prec) \geqslant \omega$. Because of Lemma 2.26 we have

$$\alpha^n + \alpha^{n+1} = \alpha^n \cdot 1 + \alpha^n \cdot \alpha = \alpha^n \cdot (1 + \alpha) = \alpha^n \cdot \alpha = \alpha^{n+1}$$

for all n, hence Proposition 2.38.ii implies $\mathrm{otype}(P^n, \prec^n_{\mathsf{lex}}) = \alpha^n$. As $(P^*, \prec^*_{\mathsf{lex}})$ corresponds to the infinite concatenation of the $(P^n, \prec^n_{\mathsf{lex}}) = \alpha^n$, we reach

$$\begin{aligned} \mathrm{otype}(P^*, \prec^*_{\mathsf{lex}}) &= \sup_{n<\omega} \big(\mathrm{otype}(P^0, \prec^0_{\mathsf{lex}}) + \cdots + \mathrm{otype}(P^n, \prec^n_{\mathsf{lex}}) \big) \\ &= \sup_{n<\omega} \big(\alpha^0 + \cdots + \alpha^n \big) = \sup_{n<\omega} \alpha^n = \alpha^\omega \ , \end{aligned}$$

using Lemma 2.36.ii and the definition of ordinal exponentiation. □

Corollary 2.40. *We have* $\mathrm{otype}(\mathbb{N}^*, <^*_{\mathsf{lex}}) = \omega^\omega$*.*

Dershowitz and Manna (1979) transmogrified the properties of principal ordinals into a well-order which does not refer to ordinals.

Definition 2.41. The *multiset extension* of a given partial order $(P, \prec)$ is $(\mathsf{mul}(P), \prec_{\mathsf{mul}})$ with

$$\begin{aligned} M \prec_{\mathsf{mul}} N \ :\Longleftrightarrow \ (\exists X, Y \in \mathsf{mul}(P))(&\emptyset \neq X \Subset N \\ &\wedge M = (N \setminus\!\!\setminus X) \uplus Y \\ &\wedge (\forall y \in Y)(\exists x \in X)(y \prec x)) \ . \end{aligned}$$

It is common practice to accompany this definition with a more perspicuous equivalent formulation.

Lemma 2.42. *If* $(\mathsf{mul}(P), \prec_{\mathsf{mul}})$ *is the multiset extension of* $(P, \prec)$*, then we have, for all* $M, N \in \mathsf{mul}(P)$*,*

$$M \prec_{\mathsf{mul}} N \iff M \neq N \wedge (\forall y \in M \setminus\!\!\setminus N)(\exists x \in N \setminus\!\!\setminus M)(y \prec x) \ .$$

Proposition 2.43.

i. *The multiset extension of a (linear) partial order is a (linear) partial order.*
ii. *The multiset extension of a well-founded partial order is a well-founded partial order.*
iii. *The multiset extension of a well-order is a well-order.*

A proof of the following folklore Theorem can be extracted, for example, from Weiermann (1992), or from Ferreira (1995, Remark 5.22).

Theorem 2.44. *If* $(P, \prec)$ *is a well-founded partial order, then we have*

$$\operatorname{otype}(\mathsf{mul}(P), \prec_{\mathsf{mul}}) = \omega^{\operatorname{otype}(P,\prec)} .$$

We are going to introduce an ordinal notation system which is able to pin down the *small Veblen number*. Our approach mimics Rathjen and Weiermann (1993) and Weiermann (1994). For an explanation of the concepts behind the following definitions the reader is invited to also consult Veblen (1908), Bachmann (1950), Schütte (1977), or Pohlers (1989).

Definition 2.45. For any $\alpha < \varepsilon_{\Omega+1}$ we introduce the set $E_\Omega(\alpha)$ of countable epsilon numbers which are needed to represent α by

$$E_\Omega(\alpha) := \begin{cases} \emptyset & \text{if } \alpha = 0 \vee \alpha = \Omega \ , \\ \{\alpha\} & \text{if } \alpha \in \mathsf{E} \ , \\ E_\Omega(\alpha_1) \cup \ldots \cup E_\Omega(\alpha_n) & \text{if } \alpha =_{\mathrm{NF}} \alpha_1 + \cdots + \alpha_n > \alpha_1 \ , \\ E_\Omega(\alpha_1) & \text{if } \alpha = \omega^{\alpha_1} > \alpha_1 \ . \end{cases}$$

Additionally we put $\alpha^\star := \max E_\Omega(\alpha)$ (and recall $\max \emptyset = 0$).

Definition 2.46. For $\alpha < \varepsilon_{\Omega+1}$ and $\beta < \Omega$ we define sets $C_n(\alpha, \beta)$ and $C(\alpha, \beta)$ of ordinals and the ordinal $\vartheta(\alpha)$ by main recursion on α and secondary recursion on $n < \omega$ via

- $\{0, \Omega\} \cup \beta \subseteq C_n(\alpha, \beta)$,
- $\xi, \eta \in C_n(\alpha, \beta) \implies \omega^\xi + \eta \in C_{n+1}(\alpha, \beta)$,
- $\xi \in C_n(\alpha, \beta) \cap \alpha \implies \vartheta(\xi) \in C_{n+1}(\alpha, \beta)$,
- $C(\alpha, \beta) := \bigcup_{n<\omega} C_n(\alpha, \beta)$,
- $\vartheta(\alpha) := \min\{\xi < \Omega : C(\alpha, \xi) \cap \Omega \subseteq \xi \wedge \alpha \in C(\alpha, \xi)\}$.

Theorem 2.47.

i. *The function* $\vartheta \colon \varepsilon_{\Omega+1} \to \mathsf{E} \cap \Omega$ *is one-to-one.*

ii. *For* $\alpha, \beta < \varepsilon_{\Omega+1}$ *we have*

$$\vartheta(\alpha) < \vartheta(\beta) \iff (\alpha < \beta \wedge \alpha^\star < \vartheta(\beta)) \vee (\beta < \alpha \wedge \vartheta(\alpha) \leqslant \beta^\star) .$$

Definition 2.48. The ordinal $\vartheta(\Omega^\omega)$ is called the *small Veblen number*, while $\vartheta(\Omega^\Omega)$ runs as the *big Veblen number*.

We will encounter the small Veblen number several times. This is due to the fact that it is the proof-theoretic ordinal of Kruskal's Tree Theorem (see Rathjen and Weiermann (1993)) and hence the first ordinal which is not the order type of a simplification order (see Definition 3.12). An alternative approach to $\vartheta(\Omega^\omega)$ from below will be presented in Section 5.3.1.

For ordinals below $\vartheta(\Omega^\omega)$ there is an effective criterion for comparisons, which is an immediate consequence of Theorem 2.47.ii.

Lemma 2.49. *Let*

$$\alpha = \vartheta(\Omega^n \cdot \alpha_0 + \cdots + \Omega^0 \cdot \alpha_n) \quad \textit{and} \quad \beta = \vartheta(\Omega^m \cdot \beta_0 + \cdots + \Omega^0 \cdot \beta_m)$$

with $\bar{\alpha}, \bar{\beta} < \vartheta(\Omega^\omega)$ *and* $\alpha_0, \beta_0 \neq 0$. *We have* $\alpha < \beta$ *if and only if*

- $(\exists i \leqslant m)(\alpha \leqslant \beta_i)$, *or*
- $n < m$ *and* $(\forall i \leqslant n)(\alpha_i < \beta)$, *or*
- $n = m$; $(\alpha_1, \ldots, \alpha_n) <_{\mathsf{lex}} (\beta_1, \ldots, \beta_n)$, *and* $(\forall i \leqslant n)(\alpha_i < \beta)$.

Definition 2.50. A partial order $(P, \prec)$ is a *partial well-order* (PWO) if any partial order $(P, \prec')$ satisfying $\prec \subseteq \prec'$ is well-founded.

The concept of PWOs occurs in various fields of mathematics. Its importance is witnessed by the fact that several people independently discovered it.

A trivial consequence of Corollary 2.19 is the following.

Lemma 2.51. *Any partial order* $(P, \prec)$ *with finite* P *is a PWO.*

Lemma 2.52. *A partial order* $(P, \prec)$ *is a PWO if and only if, for every sequence* $(p_i)_{i<\omega}$ *over* P, *there exist* $i < j$ *such that* $p_i \preccurlyeq p_j$.

A proof can be found, for example, in Fraïssé (1986, 4.3.2), or in Middeldorp and Zantema (1997).

Definition 2.53. Let A be a nonempty set. We call the elements of A *labels.*

- The set $\mathcal{B}(A)$ of (finite rooted labeled ordered) *trees* is the smallest set closed under the condition that, if $T_1, \ldots, T_n$ (with $n \geqslant 0$) are trees and if a is a label, then $(a, T_1, \ldots, T_n)$ is also a tree.
- The *size* $|\cdot|$ of such a tree is given by $|(a, T_1, \ldots, T_n)| := 1 + |T_1| + \cdots + |T_n|$.

Definition 2.54. Let $(P, \prec)$ be a partial order. By *homeomorphic embedding* we denote the partial order $(\mathcal{B}(P), \prec_{\mathsf{hemb}})$, where $T \prec_{\mathsf{hemb}} T'$ holds if and only if $T = (p, T_1, \ldots, T_n)$, $T' = (q, T'_1, \ldots, T'_m)$, and

- $T \preccurlyeq_{\mathsf{hemb}} T'_k$ for some k, or
- $p \prec q$ and there are $j_1 < \cdots < j_n$ in $[1, m]$ with $(\forall l \in [1, n])(T_l \preccurlyeq_{\mathsf{hemb}} T'_{j_l})$.

We are now prepared to state Kruskal's famous Tree Theorem, which is quite surprising in its generality. It is an extension of an earlier result of Higman (1952), which is called *Higman's Lemma* (and treats trees with bounded outdegree). The Tree Theorem is the backbone to a whole class of termination results in term rewriting theory, see Definition 3.12. Its first proof was published by Kruskal (1960), and later a rather short proof was given by Nash-Williams (1963), see also Middeldorp and Zantema (1997).

Theorem 2.55 (Kruskal's Tree Theorem). *If $(P, \prec)$ is a PWO, then so is $(\mathcal{B}(P), \prec_{\mathsf{hemb}})$.*

We will sometimes need an alternative notion of trees which includes infinite trees. This is achieved by considering as trees over A those nonempty sets of finite sequences over A which are closed under initial sequences.

Definition 2.56. Let A be a nonempty set. A (rooted) *tree over A* is any nonempty $T \subseteq A^{<\omega}$ satisfying

$$(\forall c \in T)(\forall d \in A^{<\omega})(d <_{\mathsf{ext}} c \Rightarrow d \in T) .$$

Such a T contains an (infinite) *branch* if there is a sequence $(a_i)_{i<\omega}$ over A satisfying $(\forall n)((a_i)_{i<n} \in T)$. We call T *finitely branching* if, for all $c \in T$, the set $\{a \in A : c * (a) \in T\}$ of immediate T-successors of c is finite.

König (1927) stated an important property of finitely branching trees.

Lemma 2.57 (König's Lemma). *A finitely branching tree is infinite if and only if it contains an infinite branch.*

2.4 Subrecursive Hierarchies

During the efforts to fully classify the recursive functions (see page 38 for a definition), people tried to approach large initial parts from below by defining *subrecursive hierarchies* and by singling out various principles of constructing new functions from known ones. The most prominent of these principles are *primitive* and *multiple* recursion.

Definition 2.58. A function $f\colon \mathbb{N} \to \mathbb{N}$ *(eventually) dominates* $g\colon \mathbb{N}^n \to \mathbb{N}$ if

$$g(m_1, \ldots, m_n) < f(m_1 + \cdots + m_n) \tag{2.2}$$

holds for (almost) all $m_1, \ldots, m_n$, where "almost all" means "all but finitely many". We abbreviate domination of g by f as $g <_{\mathrm{d}} f$, while the more important concept of eventual domination is abbreviated by $g <_{\mathrm{ed}} f$. The canonical variant using $\leqslant$ in (2.2) is called $\leqslant_{\mathrm{ed}}$.

We extend the notion to sets X and Y of number-theoretic functions, where $X \leqslant_{\mathrm{ed}} Y$ means $(\forall g \in X)(\exists f \in Y)(g \leqslant_{\mathrm{ed}} f)$. If additionally $Y \leqslant_{\mathrm{ed}} X$ holds, then we mark this by $X \approx_{\mathrm{ed}} Y$.

Domination of g by f is sometimes based on the sharper condition

$$g(m_1, \ldots, m_n) < f(\max\{m_1, \ldots, m_n\}) .$$

This makes not much difference because

$$\max\{m_1, \ldots, m_n\} \leqslant m_1 + \cdots + m_n \leqslant n \cdot \max\{m_1, \ldots, m_n\} ,$$

but it is of course important if sets of very slow growing functions are considered. The sets of functions we will encounter are not sensitive about this difference.

If $g \colon \mathbb{N}^m \to \mathbb{N}$ and $h_i \colon \mathbb{N}^n \to \mathbb{N}$ for all $i \in [1, m]$, then the function generated by *substitution* (or *composition*) of the h_i in g is $\mathsf{Sub}(g, h_1, \ldots, h_m) \colon \mathbb{N}^n \to \mathbb{N}$, defined via

$$\mathsf{Sub}(g, h_1, \ldots, h_m)(\bar{x}) := g(h_1(\bar{x}), \ldots, h_m(\bar{x})) .$$

Definition 2.59 (Kalmár 1943). The set ELEM of *elementary functions* is the smallest set of number-theoretic functions which contains the zero function, the successor, the projections, addition, multiplication, and modified subtraction $\dot{-}$ and is closed under substitution and bounded sums and products.

If we have some property of number-theoretic functions like being elementary, we can extend this property to sets (or relations) $P \subseteq \mathbb{N}^k$ by looking at the *characteristic function* $\chi_P \colon \mathbb{N}^k \to \{0, 1\}$, which is defined by

$$\chi_P(\bar{m}) := \begin{cases} 1 & \text{if } P(\bar{m}) , \\ 0 & \text{otherwise.} \end{cases}$$

We say that P has the property if χ_P has.

Elementary functions have rather small growth rates. For $n \in \mathbb{N}$ we recursively define $2_n \colon \mathbb{N} \to \mathbb{N}$ by

$$2_0(m) := m \quad \text{and} \quad 2_{n+1}(m) := 2^{2_n(m)} .$$

The next result is folklore, cf., for example, Monk (1976, Lemma 2.44).

Lemma 2.60. *If f is elementary, then there is $n \in \mathbb{N}$ such that $f \leqslant_{\mathrm{ed}} 2_n$.*

Thus the function $n \mapsto 2_n(n)$ is not elementary.

Skolem (1923) explicitly introduced the scheme of *primitive recursion.* For an n-ary g and an $n+2$-ary h we define the $n+1$-ary $\mathsf{PRec}(g,h)$ by

$$\begin{aligned}\mathsf{PRec}(g,h)(0,\bar{x}) &:= g(\bar{x})\;,\\ \mathsf{PRec}(g,h)(y+1,\bar{x}) &:= h(y,\bar{x},\mathsf{PRec}(g,h)(y,\bar{x}))\;.\end{aligned}$$

If $f = \mathsf{PRec}(g,h)$ also satisfies $f(y,\bar{x}) \leqslant j(y,\bar{x})$ then f is defined by *limited recursion* based on g, h, and j.

Definition 2.61. The set $\textsc{Prec}$ of *primitive recursive functions* is the smallest set of number-theoretic functions which contains zero functions of arbitrary arities, the successor, and the projections and is closed under substitution and primitive recursion.

Ackermann (1928) presented a general construction of a functions leaving a given countable set of functions, yielding the first example of a (recursive) function which grows too fast to be primitive recursive. Recall from (2.1) that $G^n(m)$ denotes the n-fold application of G to m.

Definition 2.62 (Ritchie 1965). The (binary) *Ackermann function* Ack is given by $\mathrm{Ack}(n,m) := \mathrm{Ack}_n(m)$, with its branches $\mathrm{Ack}_n\colon \mathbb{N}\to\mathbb{N}$ generated via

$$\mathrm{Ack}_0(m) := m+1 \quad\text{and}\quad \mathrm{Ack}_{n+1}(m) := \mathrm{Ack}_n^{m+1}(1)\;.$$

Lemma 2.63.

i. Ack *is the usual binary Ackermann function from Péter (1935):*

$$\begin{aligned}\mathrm{Ack}(0,m) &= m+1\;,\\ \mathrm{Ack}(n+1,0) &= \mathrm{Ack}(n,1)\;,\ \textit{and}\\ \mathrm{Ack}(n+1,m+1) &= \mathrm{Ack}(n,\mathrm{Ack}(n+1,m))\;.\end{aligned}$$

ii. Ack *is monotone in both arguments.*

iii. *The* Ack_n *are primitive recursive.*

iv. *For any primitive recursive function* $f\colon \mathbb{N}^k\to\mathbb{N}$ *there is an* n *such that*

$$f(m_1,\dots,m_k) < \mathrm{Ack}_n(\max\{m_1,\dots,m_k\})$$

holds for all $\bar{m}\in\mathbb{N}$*. Hence* $f <_{\mathrm{d}} \mathrm{Ack}_n$*.*

v. *We have* $\textsc{Prec} \approx_{\mathrm{ed}} \{\mathrm{Ack}_n : n\in\mathbb{N}\}$*.*

vi. Ack *is not primitive recursive.*

vii. *For all* n *we have* $\mathrm{Ack}(n,n) \leqslant \mathrm{Ack}(n+2,0)$*.*

Proof. Most of the results are well known, cf., for example, Péter (1936, 1951) or Rose (1984). To see why (vii) holds, we note that monotonicity of Ack yields $n \leqslant \mathrm{Ack}_{n+1}(0)$. This implies

$$\mathrm{Ack}(n,n) \leqslant \mathrm{Ack}(n, \mathrm{Ack}(n+1,0)) = \mathrm{Ack}(n+1,1) = \mathrm{Ack}(n+2,0)\ . \quad \square$$

For $k \geqslant 2$, Péter (1936) introduced the concept of *k-recursion*, which yields a strong scheme of nested recursions. Here a function F is generated from given functions g and f^i_j (with $i \in [1,k]$ and $j \in [1,k-1]$) of appropriate arities by

$$\begin{aligned} F(x,y_1,\dots,y_k) &:= 0 \quad \text{if } y_1 \cdot y_2 \cdot \ldots y_k = 0\ , \\ F(x,y_1+1,\dots,y_k+1) &:= g(x,y_1,\dots,y_k,F_1,\dots,F_k) \quad \text{else,} \end{aligned}$$

where the F_i for $i \in [1,k]$ are given by

$$F_i := F(x, y_1+1,\dots,y_{i-1}+1, y_i, F^i_1,\dots,F^i_{k-i})$$

using the abbreviation

$$F^n_m := f^n_m(x,y_1,\dots,y_k,F(x,y_1+1,\dots,y_{k-1}+1,y_k))\ .$$

Definition 2.64 (Péter 1936). A function is *k-recursive* (with $k \geqslant 2$) if it is definable using elementary functions and finitely many *k-recursions*. We collect the k-recursive functions in $\mathcal{M}_k$.

The set Mrec of *multiple recursive functions* is the smallest set of number-theoretic functions which contains zero functions of arbitrary arities, the successor, and the projections and is closed under substitution and all k-recursions.

As an example of k-recursion we show one way of extending the binary Ackermann function to higher arities.

Definition 2.65. For $k > 2$ the *k-ary Ackermann function* is defined by

$$\begin{aligned} \mathrm{Ack}(\bar{0},m) &:= m+1\ , \\ \mathrm{Ack}(\bar{l},n+1,0) &:= \mathrm{Ack}(\bar{l},n,1)\ , \\ \mathrm{Ack}(\bar{l},n+1,m+1) &:= \mathrm{Ack}(\bar{l},n,\mathrm{Ack}(\bar{l},n+1,m))\ , \text{ and} \\ \mathrm{Ack}(l_1,\dots,l_{i-1},l_i+1,0,\bar{0},m) &:= \mathrm{Ack}(\bar{l},m,\bar{0},m)\ . \end{aligned}$$

Theorem 2.66 (Péter 1936).

i. *For any $k > l \geqslant 2$, the k-ary Ackermann function is k-recursive but not l-recursive.*

ii. *Any multiple recursive function is eventually dominated by almost all functions $n \mapsto \mathrm{Ack}(n,\bar{0})$ where* Ack *is k-ary:*

$$\text{Mrec} \approx_{\text{ed}} \{n \mapsto \mathrm{Ack}(n,0^k) : k \geqslant 2\}\ .$$

The following definition of the Grzegorczyk classes, which is a slight variation on the original one, is taken from Rose (1984).

Definition 2.67 (Grzegorczyk 1953). The n^{th} *Grzegorczyk class* $\mathcal{E}_n$ is the smallest set of functions closed under substitution and limited recursion, containing the zero function, the successor, the projections, and, provided that $n = m + 1$, the function E_m, where E_m is defined by

$$\begin{aligned} \mathrm{E}_0(x, y) &:= x + y \ , \\ \mathrm{E}_1(x) &:= x^2 + 2 \ , \text{ and} \\ \mathrm{E}_{m+2}(x) &:= \mathrm{E}_{m+1}^{x}(2) \ . \end{aligned}$$

For a proof of the following Theorem which is tailored for the above definition, see Rose (1984, Section 2.2).

Theorem 2.68 (Grzegorczyk 1953).

i. *Each function in $\mathcal{E}_2$ is eventually dominated by a polynomial.*
ii. *For each k-ary $f \in \mathcal{E}_n$ and each $i \in [1, k]$, the $k+1$-ary function*

$$(y, m_1, \ldots, m_k) \mapsto f(m_1, \ldots, m_{i-1}, \cdot, m_{i+1}, \ldots, m_k)^y(m_i)$$

is in $\mathcal{E}_{n+1}$.

iii. *For all n we have $\mathcal{E}_n \subsetneq \mathcal{E}_{n+1}$.*
iv. *We have* $\textsc{Elem} = \mathcal{E}_3$ *and* $\textsc{Prec} = \bigcup_{n \in \mathbb{N}} \mathcal{E}_n$.

Various convenient ways of introducing subrecursive hierarchies are based on ordinals, more precisely on *ordinal notation systems.* The ordinals below some countable (limit) ordinal Λ can be generated from below by the interplay of finitely many closure processes. Thus they can be encoded as a subset of $\mathbb{N}$, together with appropriate relations and functions representing $<$, $+$, $\cdot$, $\omega^{\cdot}$ and some further related functions. Friedman and Sheard (1995) gave a particularly thorough definition which presumably covers all standard ordinal notation systems considered so far.

Definition 2.69 (Friedman and Sheard 1995, 1.1). An *elementary recursive ordinal notation system* (ERONS) is a tuple $(A, <, +, \cdot, \omega^{\cdot})$ such that

- $A \subseteq \mathbb{N}$ is infinite and elementary,
- $(A, <)$ is a linear order and $<$ is elementary,
- $+$, $\cdot$ and $\omega^{\cdot}$ are elementary functions (of appropriate arities) on A, not necessarily defined‡ everywhere,

‡ Although the functions are of course total, it may happen that (codes of) ordinals are not mapped to ordinals.

- "all the usual order and algebraic properties" of an initial segment of ordinals are satisfied (see Friedman and Sheard (1995) for a detailed list of seventeen properties),
- the function mapping n to the n-th element of the order A is elementary in both directions,
- for every α (which is coded in A) there exists a unique expansion $\alpha = \omega^{\alpha_1} + \cdots + \omega^{\alpha_n}$ with $\alpha_1 \geqslant \ldots \geqslant \alpha_n$, and the correspondence between α and $(\alpha_1, \ldots, \alpha_n)$ is elementary in both directions, and finally
- there are no infinite elementary $<$-descending sequences over A.

Technical reasons force us to add the following two conditions which are not present in the original definition (with $<_{\mathbb{N}}$ being the standard order on $\mathbb{N}$):

- if $\alpha = \omega^{\beta_1} + \cdots + \omega^{\beta_n}$ with $\beta_1 \geqslant \ldots \geqslant \beta_n$, then $n, \omega^{\beta_1}, \ldots, \omega^{\beta_n} \leqslant_{\mathbb{N}} \alpha$,
- for any $m \in \mathbb{N}$, the code for m is smaller (with respect to $<_{\mathbb{N}}$) than the code for ω^m.

The last two conditions are introduced to meet the demands of Buchholz et al. (1994, Theorem 2), which corresponds to Theorem 2.82. As the underlying coding of sequences usually has the subterm property, it is not difficult to satisfy these conditions.

For the remainder of this subsection we assume that the ordinals below some limit ordinal Λ are given as an ERONS.

Definition 2.70. Let $\Lambda \geqslant \varepsilon_0$ be a countable limit ordinal. For $\alpha < \Lambda$ and any function h from $\mathbb{N}^2$ into the (codes of) ordinals below α, we introduce the *count function* count-h: $\mathbb{N} \to \mathbb{N}$ via

$$\text{count-}h(m) := \min\{n : h(m,n) \leqslant h(m,n+1)\} .$$

Based on this we can define the scheme of *descent recursion.* For h as above and any binary number-theoretic function g we introduce the unary function $\mathsf{DRec}(g,h)$ by

$$\mathsf{DRec}(g,h)(m) := g(m, \text{count-}h(m)) .$$

The *α-recursive functions* are all such functions $\mathsf{DRec}(g,h)$ with g and h being elementary. They are collected in the set $\textsc{Drec}(\alpha)$. A function which is α-recursive for some α is called *ordinal recursive* or *descent recursive.* For convenience we call a function *$<\alpha$-recursive* if it is β-recursive for some $\beta < \alpha$. The set of those functions is called $\textsc{Drec}(<\alpha)$.

It should be mentioned that our $\textsc{Drec}(<\alpha)$ coincides with the $DR(\alpha)$ of Friedman and Sheard (1995), while our $\textsc{Drec}(\alpha)$ is their $DR(\alpha+1)$. Our ordinal recursive functions are not to be mixed up with those of Kreisel (1952).

There is a slight danger of confusing α-recursion and k-recursion. But as α-recursion is only interesting for $\alpha \geqslant \omega$, it should always be clear from the context which kind of recursion is meant.

While (i) of the following Theorem is immediate, the remaining items can also be found, for example, in Friedman and Sheard (1995).

Theorem 2.71 (Robbin 1965).

i. *For* $\beta < \alpha < \Lambda$ *we have* $\textsc{Drec}(\beta) \subseteq \textsc{Drec}({<}\alpha) \subseteq \textsc{Drec}(\alpha)$.
ii. $\textsc{Elem} \approx_{\mathrm{ed}} \textsc{Drec}(\omega) = \textsc{Drec}({<}\omega^2)$
iii. $\textsc{Prec} \approx_{\mathrm{ed}} \textsc{Drec}({<}\omega^\omega) \subsetneq \textsc{Drec}(\omega^\omega)$
iv. *If* $\alpha \geqslant \omega^2$, *then for every* α*-recursive function* f *there is an elementary* h *mapping into* α *such that we have* $f(m) \leqslant \text{count-}h(m)$ *for all* m.

With the aim of classifying (a part of) the recursive functions, a variety of number-theoretic hierarchies has been introduced. We present the most prominent examples, which are defined in the same way. Starting at an initial function (which gets the index 0), we define the function with index $\alpha + 1$ in a uniform manner using the function with index α, and finally for limit ordinals λ the function with index λ is constructed via diagonalization of its predecessors along *fundamental sequences*.

Definition 2.72. Fix some countable limit ordinal Λ (as usual, represented by an ERONS). An assignment of *fundamental sequences* for Λ is an elementary function $\cdot[\cdot]\colon \Lambda \times \mathbb{N} \to \Lambda$ which satisfies, for all $n \in \mathbb{N}$,

- $0[n] = 0$,
- $(\alpha + 1)[n] = \alpha$ for all $\alpha < \Lambda$,
- $\alpha[n] < \alpha[n+1] < \alpha$ for all limit ordinals $\alpha < \Lambda$,
- $(\alpha + \beta)[n] = \alpha + \beta[n]$ if $\alpha + \beta = \alpha \oplus \beta < \Lambda$,
- $\omega^{\alpha+1}[n] = \omega^\alpha \cdot (n+1)$ if $\omega^{\alpha+1} < \Lambda$, and finally
- $\omega^\alpha[n] = \omega^{\alpha[n]}$ if $\alpha < \omega^\alpha < \Lambda$.

If additionally the *Bachmann property*, demanding

$$(\forall \alpha, \beta < \Lambda)(\alpha[n] < \beta < \alpha \implies \alpha[n] \leqslant \beta[0]) ,$$

holds, $(\Lambda, \cdot[\cdot])$ is called a *Bachmann system.*

The second half of the six conditions above is usually not demanded explicitly, but almost all assignments of fundamental sequences occurring in the literature meet them. In Bachmann (1967) the Bachmann property appeared first.

Lemma 2.73. *If* $(\Lambda, \cdot[\cdot])$ *is a Bachmann system, then we have* $\alpha = \sup_{n \in \mathbb{N}} \alpha[n]$ *for all limit ordinals* $\alpha < \Lambda$.

Proof. Assume for a contradiction that we have $\beta = \sup_{n\in\mathbb{N}} \alpha[n]$ for some $\beta < \alpha$. Take an arbitrary n. As $\alpha[n] < \alpha[n+1]$ holds, we may conclude $\alpha[n] < \beta < \alpha$, and see $\beta[0] < \beta$. Now the Bachmann property yields $\alpha[n] \leqslant \beta[0]$, hence we arrive at the contradiction $\beta = \sup_{n\in\mathbb{N}} \alpha[n] \leqslant \beta[0] < \beta$. □

Theorem 2.74 (Bachmann 1967, Schmidt 1976a). *Let us drop for the moment the restrictions concerning elementary functions in Definition 2.70 and Definition 2.72. Under this provision we get:*

i. *Every countable ordinal carries a Bachmann system.*
ii. *The first uncountable ordinal does not carry a Bachmann system.*

Proof. For proofs of (i) see Schmidt (1976a), Rose (1984, Theorem 3.2) or the elegant Buchholz et al. (1994). We can find (ii) in Bachmann (1967) and Schmidt (1976b). □

For the remainder of this subsection we fix a Bachmann system $(\Lambda, \cdot[\cdot])$. This will be the basis for the following definitions and observations.

Definition 2.75. Based (and depending) on our Bachmann system $(\Lambda, \cdot[\cdot])$, we recursively define four hierarchies of number-theoretic functions. Consider $\alpha, \lambda < \Lambda$ with $\lambda \in \mathsf{Lim}$.

The *fast growing functions* $(\mathrm{F}_\gamma)_{\gamma<\Lambda}$ are based on iterations and defined by

$$\mathrm{F}_0(m) := 3^{m+1}, \quad \mathrm{F}_{\alpha+1}(m) := \mathrm{F}_\alpha^{m+1}(m), \quad \text{and} \quad \mathrm{F}_\lambda(m) := \mathrm{F}_{\lambda[m]}(m)\ .$$

Of slightly slower growth are the *Hardy functions* $(\mathrm{H}_\gamma)_{\gamma<\Lambda}$, introduced by

$$\mathrm{H}_0(m) := m, \quad \mathrm{H}_{\alpha+1}(m) := \mathrm{H}_\alpha(m+1), \quad \text{and} \quad \mathrm{H}_\lambda(m) := \mathrm{H}_{\lambda[m]}(m+1)\ ,$$

and the related *counting functions* $(\mathrm{L}_\gamma)_{\gamma<\Lambda}$, which are constructed via

$$\mathrm{L}_0(m) := 0, \quad \mathrm{L}_{\alpha+1}(m) := \mathrm{L}_\alpha(m+1) + 1, \quad \text{and} \quad \mathrm{L}_\lambda(m) := \mathrm{L}_{\lambda[m]}(m+1) + 1\ .$$

Finally, the *slow growing functions* $(\mathrm{G}_\gamma)_{\gamma<\Lambda}$ are given by

$$\mathrm{G}_0(m) := 0, \quad \mathrm{G}_{\alpha+1}(m) := \mathrm{G}_\alpha(m) + 1, \quad \text{and} \quad \mathrm{G}_\lambda(m) := \mathrm{G}_{\lambda[m]}(m)\ .$$

Hardy (1904) used the H_γ to construct a set of real numbers of cardinality $\aleph_1$, and their first appearance in the field of subrecursive hierarchies is Wainer (1972), with the slightly different choice $\mathrm{H}_{\lambda[m]}(m)$ for $\mathrm{H}_\lambda(m)$. This does not affect growth rates, cf. Buchholz et al. (1994). Robbin (1965) investigated the fast growing functions up to ω^ω as the canonical extension of (variants of) the Grzegorczyk functions E_m to the transfinite. Later this approach was extended to ε_0, see Löb and Wainer (1970), Schwichtenberg (1971), and Wainer (1970).

Weiermann (1997b) observed that the choice of the underlying assignment of fundamental sequences is vital to the slow growing hierarchy. In contrast to this, the remaining three hierarchies are not that much affected by changes of the assignment. The choice of the nonzero initial functions is not really important, provided that they are honest. Just about any monotone elementary function will do. We chose an exponential function as the initial fast growing function in favor of the usual successor function because the slightly faster growth rates it induces will be useful in Section 5.2. To give a taste of the easier proofs in this field, we establish those properties of the F_n we will use later.

Lemma 2.76. *Let natural numbers n, m, and a be given.*

i. *Each F_n is primitive recursive, monotone, and has the subterm property.*
ii. $\mathrm{F}_n(0) = 3$, $\mathrm{F}_{n+1}(m+1) > \mathrm{F}_n(m+1)$, *and* $\mathrm{F}_n^a(m) \geqslant a$.
iii. $\mathrm{F}_n(m+1) > \mathrm{Ack}_n(m)$ *and* $\mathrm{Ack}_{n+3}(3m) > \mathrm{F}_n(m)$.
iv. F_ω *eventually dominates all primitive recursive functions.*
v. *If $a \geqslant 2$, then* $\mathrm{F}_{n+1}^a(m) > \mathrm{F}_n^{a^2}(m)$.
vi. $\mathrm{F}_n^{a+1}(m) > a \cdot \mathrm{F}_n^a(m)$
vii. *If $a \geqslant 3$, then* $\mathrm{F}_{n+1}^{a-1}(m) > \mathrm{F}_n^{a+1}(m)$.

Proof. The proofs for (i) and (ii) are standard and therefore left out. For $m > 0$ we show $\mathrm{F}_n(m) > \mathrm{Ack}_n(m)$ by induction on n. The interesting part here is the induction step:

$$\mathrm{F}_{n+1}(m) = \mathrm{F}_n^{m+1}(m) \geqslant \mathrm{F}_n^{m+1}(1) > \mathrm{Ack}_n^{m+1}(1) = \mathrm{Ack}_{n+1}(m) \ .$$

From this the first half of (iii) follows. By inductions we can show

$$\mathrm{Ack}_1(m) = m + 2, \quad \mathrm{Ack}_2(m) = 2m + 3 \ ,$$

and

$$\mathrm{Ack}_3(m) > 2^{m+2} > 3m \ . \tag{2.3}$$

Building on this, the second half of (iii) is shown by induction on n. To start with, we see

$$\mathrm{Ack}_3(3m) > 2^{3m+2} = 4 \cdot 8^m > 3 \cdot 3^m = \mathrm{F}_0(m) \ .$$

In the induction step we first treat the case $m = 0$. Here we have

$$\mathrm{Ack}_{n+4}(0) > \mathrm{Ack}_3(0) = 5 > 3 = \mathrm{F}_{n+1}(0) \ .$$

For the case $m > 0$ we present two auxiliary results. A direct consequence of monotonicity is the property $\mathrm{Ack}_{n+3}^{m-1}(1) \geqslant m$. The induction hypothesis implies

$$\mathrm{Ack}_{n+3}^2(l) \geqslant \mathrm{Ack}_{n+3}(\mathrm{Ack}_3(l)) > \mathrm{Ack}_{n+3}(3l) > \mathrm{F}_n(l)$$

for $l \in \mathbb{N}$, hence for all $k > 0$ we have $\mathrm{Ack}^{2k}_{n+3}(l) > \mathrm{F}^{k}_{n}(l)$. A combination of these results yields

$$\begin{aligned}\mathrm{Ack}_{n+4}(3m) &= \mathrm{Ack}^{3m+1}_{n+3}(1) = \mathrm{Ack}^{2(m+1)}_{n+3}(\mathrm{Ack}^{m-1}_{n+3}(1)) \\ &\geqslant \mathrm{Ack}^{2(m+1)}_{n+3}(m) > \mathrm{F}^{m+1}_{n}(m) = \mathrm{F}_{n+1}(m) \ .\end{aligned}$$

Let us take care of (iv). By definition and by (ii), F_ω eventually dominates all F_n, hence, using (iii), it eventually dominates all Ack_n. Now Lemma 2.63.v implies all primitive recursive functions are eventually dominated by F_ω.

We show (v) by induction on $a \geqslant 2$. If $a = 2$, then we may rely on (i) and $\mathrm{F}_{n+1}(m) \geqslant 3$, and for the induction step we have

$$\mathrm{F}^{a+1}_{n+1}(m) > \mathrm{F}_n^{\mathrm{F}_n^{a^2}(m)+1}(\mathrm{F}_n^{a^2}(m)) \geqslant \mathrm{F}_n^{2a+1}(\mathrm{F}_n^{a^2}(m))$$

by the induction hypothesis and by $\mathrm{F}_n^{a^2}(m) \geqslant a^2 \geqslant 2a$. To prove (vi) we recall $3^b > b^2$ holds for all $b \in \mathbb{N}$. This yields

$$\mathrm{F}^{a+1}_{n}(m) = \mathrm{F}_n(\mathrm{F}^a_n(m)) > 3^{\mathrm{F}^a_n(m)} > \mathrm{F}^a_n(m) \cdot \mathrm{F}^a_n(m) \geqslant a \cdot \mathrm{F}^a_n(m) \ .$$

Finally, (vii) is established by

$$\mathrm{F}^{a-1}_{n+1}(m) = \mathrm{F}^2_{n+1}(\mathrm{F}^{a-3}_{n+1}(m)) > \mathrm{F}^4_n(\mathrm{F}^{a-3}_n(m)) = \mathrm{F}^{a+1}_n(m) \ ,$$

which follows from (v). □

Lemma 2.77. *For any $\alpha < \Lambda$ and any $m \in \mathbb{N}$ we have*

$$\mathrm{H}_\alpha(m) = \mathrm{H}_{\mathrm{L}_\alpha(m)}(m) = m + \mathrm{L}_\alpha(m) \ .$$

Proof. The second equation is easily established by noting that for finite n we have $\mathrm{H}_n(m) = n + m$. We prove the first equation by induction on α. The case $\alpha = 0$ follows from $\mathrm{L}_0(n) = 0$. For $\alpha \neq 0$ we put $\gamma := \alpha[m]$ and gladly see

$$\mathrm{H}_\alpha(m) = \mathrm{H}_\gamma(m+1) = \mathrm{H}_{\mathrm{L}_\gamma(m+1)}(m+1) = \mathrm{H}_{\mathrm{L}_\gamma(m+1)+1}(m) = \mathrm{H}_{\mathrm{L}_\alpha(m)}(m)$$

suffices. □

The next result is standard. For a proof see, e.g., Buchholz et al. (1994).

Lemma 2.78. *Let $\alpha, \beta < \Lambda$ be given.*

i. *Each H_α is strictly monotone.*
ii. *If $\alpha + \beta = \alpha \oplus \beta < \Lambda$, then for each m we have $\mathrm{H}_{\alpha+\beta}(m) = \mathrm{H}_\alpha(\mathrm{H}_\beta(m))$.*
iii. *If $\beta < \alpha$, then $\mathrm{H}_\beta <_{\mathrm{ed}} \mathrm{H}_\alpha$.*

In some sense, the Hardy function H_α is a paradigmatic α-recursive function. According to, e.g., Friedman and Sheard (1995) we have the following result.

Proposition 2.79. *For any $\alpha < \Lambda$, the function H_α is α-recursive.*

Corollary 2.80. $\{\mathrm{H}_\beta : \beta < \alpha\} \subseteq \textsc{Drec}(<\alpha)$ *and* $\{\mathrm{H}_\beta : \beta \leqslant \alpha\} \subseteq \textsc{Drec}(\alpha)$.

In order to reach the opposite direction of the above Corollary, a further seemingly restrictive condition on the underlying Bachmann system has to be imposed. Apparently all standard notation systems satisfy this.

Definition 2.81 (Zemke 1977; Friedman and Sheard 1995, 1.37). The Bachmann system $(\Lambda, \cdot[\cdot])$ is *tame* if there exists a unary elementary function g satisfying

$$\beta < \alpha < \Lambda \implies \beta \leqslant \alpha[g(\beta)] \ .$$

A similar condition (demanding instead that g is primitive recursive) has been thoroughly investigated by Buchholz et al. (1994).

Theorem 2.82. *Let $(\Lambda, \cdot[\cdot])$ be a tame Bachmann system. If there exists $\lambda' > 0$ with $\lambda = \omega^\omega \cdot \lambda' \leqslant \Lambda$, then*

$$\textsc{Drec}(<\lambda) \approx_{\mathrm{ed}} \{\mathrm{H}_\alpha : \alpha < \lambda\} \ .$$

Proof. One inclusion is part of Corollary 2.80. For a proof of the opposite inclusion see Friedman and Sheard (1995, 1.42) (where only ordinals below ε_0 are explicitly treated), or Buchholz et al. (1994, Theorem 2) (which has to be combined with Theorem 2.71.iv). □

We are now prepared to supplement Theorem 2.71.

Theorem 2.83. *If $(\Lambda, \cdot[\cdot])$ is a tame Bachmann system, then we have*

$$\textsc{Prec} \approx_{\mathrm{ed}} \{\mathrm{F}_\alpha : \alpha < \omega\} \approx_{\mathrm{ed}} \{\mathrm{H}_\alpha : \alpha < \omega^\omega\} \approx_{\mathrm{ed}} \textsc{Drec}(<\omega^\omega)$$

and

$$\textsc{Mrec} \approx_{\mathrm{ed}} \{\mathrm{F}_\alpha : \alpha < \omega^\omega\} \approx_{\mathrm{ed}} \{\mathrm{H}_\alpha : \alpha < \omega_3\} \approx_{\mathrm{ed}} \textsc{Drec}(<\omega_3) \ .$$

Proof. As the F_m with $m \in \mathbb{N}$ are just a version of Grzegorczyk's E_m, his Theorem 2.68.iv establishes $\textsc{Prec} \approx_{\mathrm{ed}} (\mathrm{F}_\alpha)_{\alpha<\omega}$. The connection between $\textsc{Mrec}$ and the fast growing hierarchy was established by Robbin (1965). He also showed that the function $\mathrm{F}_{\omega^\omega}$ is not multiple recursive, and proved the results concerning descent recursion. Later Wainer (1972) used a connection between fast growing and Hardy functions which suffices for our purposes (roughly speaking, for $\alpha \leqslant \varepsilon_0$ the growth rates of F_α and $\mathrm{H}_{\omega^\alpha}$ are the same). □

Under certain additional assumptions concerning the underlying Bachmann system, Girard (1981) (see also Cichon and Wainer (1983)) showed

$$\textsc{Mrec} \approx_{\mathrm{ed}} \{\mathrm{G}_\alpha : \alpha < \vartheta(\Omega^\omega)\} \ . \tag{2.4}$$

2.5 Complexity Classes

Originating with Turing (1936), a multitude of equivalent models of computation has been introduced. One such model, which is particularly well suited to be simulated by TRSs, is the *register machine*, sometimes also called *random access machine*.

Definition 2.84 (Shepherdson and Sturgis 1963). A (k-)*register machine* (RM) consists of a counter (which is used to store the line number of a program running on the machine) and $k \geqslant 2$ registers containing natural numbers. Such a machine is equipped with a *program*, which is a sequence $\mathcal{P} = (a_0, \ldots, a_m)$ of *instructions*. Each instruction a_p is either of the form $(j, +, q)$, $(j, -, q)$, or (j, q, r), with $j \in [1, k]$ and $q, r \in [0, m+1]$. The *configuration* of a k-RM is described by $(p; b_1, \ldots, b_k)$, where $p \in [0, m+1]$ denotes the current instruction number and the $\bar{b}$ are the numbers stored in the registers.

To run the program $\mathcal{P}$ with *input* $n_1, \ldots, n_k$, we start with the configuration $(0; n_1, \ldots, n_k)$. The machine stops with *output* $\bar{b}$ if it reaches the configuration $(m+1; \bar{b})$. Otherwise, the current configuration is $(p; \bar{b})$, and p corresponds to an instruction a_p. If this is of the form $(j, +, q)$, then we make a transition to the configuration $(q; b_1, \ldots, b_j + 1, \ldots, b_k)$. Similarly, with $(j, -, q)$ we continue at $(q; b_1, \ldots, b_j \dot{-} 1, \ldots, b_k)$. The instruction (j, q, r) tests the content of the register j. If $b_j = 0$, then we continue at $(q; \bar{b})$, and otherwise at $(r; \bar{b})$.

Definition 2.85. Let $k \geqslant l$. A function $f \colon \mathbb{N}^l \to \mathbb{N}$ is *computable* on a k-RM with (*timebound* $g \colon \mathbb{N} \to \mathbb{N}$ and) program $\mathcal{P}$ if, for any $n_1, \ldots, n_l$, the machine with input $n_1, \ldots, n_l, 0^{k-l}$ outputs $f(n_1, \ldots, n_l), 0^{k-1}$ (and if additionally the function counting the number of program steps is dominated by g).

Our definition forces programs to be *deterministic*, indicating that the next configuration is always uniquely determined. Sometimes it is advisable to consider an alternative definition of program where, instead of one next program line, a set of next lines is considered. The number of the next line is randomly picked out of this set. Using this approach, the next configuration is no longer uniquely determined. This provides good reason to call such a program *nondeterministic*. It is a matter of taste what is to be understood as *the* computed function in this context. One possible choice is presented in Definition 6.22.

The *Turing machine* (TM), introduced by Turing (1936), is the mathematical model of computation which occurs most frequently. For an explicit definition of TMs and the concept of function computed by such a machine, the reader may consult any standard text on computation, such as Davis (1958), to name but one. The notions of *program*, *instruction*, *configuration*, and *computable function* (with some *timebound*) are easily transferable to TMs. If a function is computable by a TM, it is called *recursive*.

Alternatively, a computation with timebound g can be defined by allowing for at most $g(\log(n+1))$ computation steps, see, e.g., Börger (1989). Our results are not affected by this.

Various subclasses of the recursive functions have been studied over the last decades. The most prominent ones are concerned with *timebounds*, measuring the complexity of a function by counting the program steps needed to compute it, and *spacebounds*, which are based on the number of band cells touched during the computation. We will later encounter the computational classes collecting the functions computable (on a TM) in *polynomial*, *exponential*, and *double exponential* time. These are called $\textsc{Ptime}$, $\textsc{Etime}$, and $\mathrm{E}_2\textsc{time}$, while $\textsc{Linspace}$ denotes the functions computable under a linear spacebound.

The *O-calculus* provides an important tool for complexity characterizations.

Definition 2.86. For $f\colon \mathbb{N} \to \mathbb{N}$ we say that a number-theoretic function g is in $O(f)$ if there are $c, d \in \mathbb{N}$ satisfying $g \leqslant_{\mathrm{ed}} cf + d$.

We will frequently extend this notion to contexts. For example, $\mathrm{Ack}(O(n), 0)$ contains the functions which are eventually dominated by $n \mapsto \mathrm{Ack}(f(n), 0)$ for some f in $O(n)$.

Definition 2.87. In $\textsc{Atime}$ we collect the functions computable (on a TM) with timebound from $\mathrm{Ack}(O(n), 0)$.

Though the principle of eventual domination proves useful for many observations, it is not necessary for some prominent sets of functions.

Lemma 2.88. *Let $\mathcal{M}$ be either* $\textsc{Elem}$*,* $\textsc{Prec}$*,* $\textsc{Mrec}$*, or* $\mathrm{Ack}(O(n), 0)$*. If f is a number-theoretic function such that there is $g \in \mathcal{M}$ with $f \leqslant_{\mathrm{ed}} g$, then there is $h \in \mathcal{M}$ with $f <_{\mathrm{d}} h$.*

Proof. As there are only finitely many inputs where f may be larger than g, there is $a \in \mathbb{N}$ such that $f <_{\mathrm{d}} g + a$. Now the result becomes obvious for all classes but $\mathrm{Ack}(O(n), 0)$, as these are closed under addition with constants. For $\mathrm{Ack}(O(n), 0)$ it suffices to note that, by monotonicity of Ack, we have $\mathrm{Ack}(cn + d, 0) + a \leqslant \mathrm{Ack}(cn + d + a, 0)$. □

As models of computation, TMs and RMs are (mostly) equivalent. The following result is folklore.

Theorem 2.89. *If f is computable by a TM (resp. RM) with timebound g, then there is a unary exponential function p such that f is computable on a RM (resp. TM) with timebound $p \circ g$.*

We introduce a property of sets of functions which reflects the above property. This notion occurs, for example, in Handley and Wainer (1994, 4.16).

Definition 2.90. A set $\mathcal{M}$ of number-theoretic functions *accommodates exponentiation* if

- for any k-ary f in $\mathcal{M}$, there is a unary g in $\mathcal{M}$ with $f \leqslant_{\mathrm{ed}} g$, and
- for any unary f and g in $\mathcal{M}$, there is a unary h in $\mathcal{M}$ which eventually dominates $n \mapsto f(n)^{g(n)}$.

Many well-known sets of functions accommodate exponentiation.

Lemma 2.91. *The sets of functions* $\textsc{Elem}$*,* $\textsc{Prec}$*,* $\textsc{Mrec}$*, and* $\mathrm{Ack}(O(n),0)$ *accommodate exponentiation.*

Proof. For the first three sets this is easily seen, as they contain addition and exponentiation, and are closed under substitution.

The different structure of $\mathrm{Ack}(O(n),0)$ deserves a closer look. We first note that, for any $n \geqslant 4$, monotonicity of the Ackermann function and (2.3) yield

$$\begin{aligned} 2^{\mathrm{Ack}_n(0)} &< \mathrm{Ack}_3(\mathrm{Ack}_n(0)) = \mathrm{Ack}_3(\mathrm{Ack}_{n-1}(1)) \\ &\leqslant \mathrm{Ack}_{n-1}^2(1) = \mathrm{Ack}_n(1) = \mathrm{Ack}_{n+1}(0)\ . \end{aligned}$$

It suffices to show that, for any $c, d > 0$ there is a' such that, for almost all n,

$$\mathrm{Ack}(cn,0)^{\mathrm{Ack}(dn,0)} < \mathrm{Ack}(a'n,0)$$

holds. Given c and d, we put $a := \max\{c,d\}$. As soon as $an \geqslant 4$, we get

$$\begin{aligned} \mathrm{Ack}(cn,0)^{\mathrm{Ack}(dn,0)} &\leqslant \mathrm{Ack}(an,0)^{\mathrm{Ack}(an,0)} \leqslant \left(2^{\mathrm{Ack}(an,0)}\right)^{\mathrm{Ack}(an,0)} \\ &= 2^{\mathrm{Ack}(an,0)^2} \leqslant 2^{2^{\mathrm{Ack}(an,0)}} < \mathrm{Ack}(an+2,0)\ , \end{aligned}$$

using $(\forall m \geqslant 4)(m^2 \leqslant 2^m)$ and $\mathrm{Ack}(an,0) > 4$. □

The following result is an immediate consequence of Theorem 2.89.

Lemma 2.92. *Let* $\mathcal{M} \supseteq \textsc{Elem}$ *accommodate exponentiation. A function is computable on a TM with timebound from* $\mathcal{M}$ *if and only if it is computable on a RM with timebound from* $\mathcal{M}$*.*

A fundamental result of recursion theory is Kleene's *normal form theorem.*

Theorem 2.93 (Kleene 1936). *For any* $k > 0$ *there are primitive recursive functions* $U\colon \mathbb{N} \to \mathbb{N}$ *and* $C\colon \mathbb{N}^{k+2} \to \mathbb{N}$ *such that for any* $f\colon \mathbb{N}^k \to \mathbb{N}$ *which is computable on a TM with some timebound* g*, there is* $e \in \mathbb{N}$ *satisfying*

$$(\forall \bar{n})(f(n_1,\ldots,n_k) = U(C(e,n_1,\ldots,n_k,g(n_1+\cdots+n_k))))\ .$$

The Theorem can be combined with the closure under substitutions of both $\textsc{Prec}$ and $\textsc{Mrec} \supseteq \textsc{Prec}$.

Corollary 2.94. *If* f *is computable on a TM with timebound from* $\textsc{Prec}$ *(resp.* $\textsc{Mrec}$*), then* f *is in* $\textsc{Prec}$ *(resp.* $\textsc{Mrec}$*).*

3 Term Rewriting

A liar only fools himself.

The concept of *term rewriting* was developed at the end of the 1960s. An early reference of its practical importance is Knuth and Bendix (1970). The basic idea is to turn a set of term equations into directed equations, telling which side of the equation is to be regarded as more complex. A *term rewriting system* (TRS) is a finite[*] set of such equations. The emerging *rewrite relation* is the closure of the TRS under substitutions and contexts. Provided that the TRS satisfies some additional properties (it has to be confluent and terminating), the question whether two terms are equal with respect to the given (undirected) equations becomes decidable, as it amounts to iterated rewriting of both terms until no further rules can be applied, followed by a comparison of the resulting terms. This makes TRSs a valuable tool in automated deduction. Additionally term rewriting constitutes an interesting paradigm for nondeterministic computation, importing the term notion for free.

Termination is one major topic in computation. A TRS is said to *terminate* if all derivations induced by its rewrite relation are finite. Just as in computation theory, the problem of establishing termination of a given TRS is nontrivial. Various approaches to it have been made. A common pattern is the association of some well-founded order to the rewrite relation in order to establish well-foundedness of the rewrite relation. The basic definitions and results dealing with this will be the main topic of Section 3.2. Two main directions can be distinguished: *semantic* and *syntactic* orders. For orders of the first kind, which will be presented in Section 3.3, the main idea consists in mapping each term to a member of a well-founded partial order $\prec$. If this can be done by embedding the (transitive closure of the) rewrite relation into $\prec$, termination is achieved. A very convenient way of doing so is by means of Σ-*algebras*. The basis of a Σ-algebra is a partial order $(P, \prec)$. For any symbol of the signature, this is joined to a function on P of the appropriate arity. Roughly speaking, this corresponds

[*] It is common to consider also infinite systems, but for our aims this makes no sense.

to a $\Sigma \cup \{<\}$-structure in first order logic. If the partial order is well-founded and if certain additional properties (like compatibility and monotonicity) are established, termination is proven.

The main difficulty in showing termination by the semantical approach consists in finding an appropriate well-founded order. As the general question of termination of TRS is undecidable, this problem necessarily is hard.

It is fruitful to study certain subclasses of the orders which can be used to establish termination. The most prominent such subclass consists of the *simplification orders* of Dershowitz (1979), which will be the key object of this text. Well-foundedness of simplification orders is a consequence of their compatibility with the homeomorphic embeddability relation used in Kruskal's beautiful Tree Theorem. Another consequence of this Theorem is that compatibility of a TRS with a monotone Σ-algebra which also satisfies the subterm property implies termination. Such a TRS is called *simply terminating*. Zantema (1993, 1994) speaks of *total termination* if the Σ-algebra is based on a (total) well-order. This constitutes an important proper subclass of simple termination.

As any well-order is order-isomorphic to an ordinal, it is possible to classify totally terminating TRSs according to the ordinals on which compatible Σ-algebras live. This leads to a rich structure of subhierarchies. Inside the set of TRSs whose termination is provable on the ordinal ω we can further differentiate, depending on the functions used for interpreting the function symbols of the Σ-algebra. The canonical approach here is a classification along various subrecursive hierarchies, and it unveils a complex structure.

Neither simple nor total termination (nor even termination on ω) are decidable. Syntactic simplification orders, which will be introduced in Section 3.4, provide a partial remedy to this. Depending on a *precedence* (a partial order on the underlying signature), which is sometimes joined by additional structure, one uniformly defines by recursion a simplification order on the terms. Instead of searching within the infinitude of simplification orders one simply has to try out the orders generated by the (finitely many) partial orders of the signature. Termination via such an order is decidable, provided that the "additional structure" is also of finitary nature. Important examples of such syntactic orders are the multiset path order (MPO) of Plaisted (1978), the lexicographic path order (LPO) of Kamin and Lévy (1980), and the Knuth–Bendix order (KBO) of Knuth and Bendix (1970). It will turn out that these orders cover but a very small part of total termination.

In Section 3.5 we take a look at computability. TRSs provide a powerful and natural model of nondeterministic computation, because the term concept, which is ubiquitous in mathematics, is built in. It is fairly obvious that TMs can simulate computation by TRSs. We will see that the simulation is possible at a not too high expenditure.

3.1 Basic Definitions

The following definitions and results are variations on the texts of Baader and Nipkow (1998), Dershowitz and Jouannaud (1990), Ferreira and Zantema (1993), Ferreira (1995), Hofbauer (1991), and Touzet (1998b).

A *signature* is a *finite* set Σ equipped with a mapping $\#\colon \Sigma \to \mathbb{N}$. The elements of Σ are *function symbols.* For each $f \in \Sigma$, we call $\#(f)$ its (unique) *arity* and equivalently speak of f as a $\#(f)$-ary function symbol. Function symbols of arities 1, 2, 3, resp., are *unary, binary,* and *ternary.* The set of function symbols in Σ having arity n ($\geqslant n$, $\leqslant n$, resp.) is denoted by $\Sigma^{(n)}$ ($\Sigma^{(\geqslant n)}$, $\Sigma^{(\leqslant n)}$, resp.).

In order to avoid unpleasant situations we demand $\emptyset \neq \Sigma^{(0)} \neq \Sigma$.

Members of $\Sigma^{(0)}$ are *constants.* The *maximal arity* of Σ, denoted by $\mathrm{Ar}(\Sigma)$, is the largest n such that $\Sigma^{(n)} \neq \emptyset$. For any set $\mathcal{X}$ which is disjoint from Σ, we can consider the *term algebra* $\mathcal{T}(\Sigma, \mathcal{X})$ over Σ and $\mathcal{X}$, which is the smallest set $\mathcal{T} \supseteq \mathcal{X}$ closed under the condition

$$n \in \mathbb{N} \land f \in \Sigma^{(n)} \land s_1, \dots, s_n \in \mathcal{T} \implies f(s_1, \dots, s_n) \in \mathcal{T} \ .$$

Note that $n = 0$ is allowed here. In this case, $f()$ is simply f. Elements of $\mathcal{T}(\Sigma, \mathcal{X})$ are called *terms (over Σ and $\mathcal{X}$).* The *variables* $\mathcal{V}$ form a countably infinite set which is disjoint from Σ. A *symbol* is a member of $\Sigma \cup \mathcal{V}$. We usually consider terms which are members of $\mathcal{T}(\Sigma, \mathcal{V})$, for various signatures Σ. The *root symbol* $\mathrm{root}(s)$ of a term s is defined by $\mathrm{root}(x) := x$ for $x \in \mathcal{V}$ and $\mathrm{root}(f(\dots)) := f$. Symbols *occuring* in a term s are collected in $\mathcal{O}(s)$, which is defined by $\mathcal{O}(x) := \{x\}$ for $x \in \mathcal{V}$ and

$$\mathcal{O}(f(s_1, \dots, s_n)) := \{f\} \cup \mathcal{O}(s_1) \cup \ldots \cup \mathcal{O}(s_n) \ .$$

The variables occurring in s are $\mathcal{V}(s) := \mathcal{O}(s) \cap \mathcal{V}$. Any term containing no variables is *closed* or *ground*, and $\mathcal{T}(\Sigma)$ is the set of closed terms. Of course, $\mathcal{T}(\Sigma)$ may be identified with $\mathcal{T}(\Sigma, \emptyset)$. The set of *subterms* of a term s is called $\mathcal{S}(s)$ and defined by $\mathcal{S}(x) := \{x\}$ for $x \in \mathcal{V}$ and

$$\mathcal{S}(f(s_1, \dots, s_n)) := \{f(s_1, \dots, s_n)\} \cup \mathcal{S}(s_1) \cup \ldots \cup \mathcal{S}(s_n) \ .$$

Any subterm t of s satisfying $t \neq s$ is a *proper subterm* of s. Being a proper subterm constitutes a partial order $(\mathcal{T}(\Sigma, \mathcal{V}), <_{\mathsf{sub}})$.

Before we can introduce the number of occurrences of a symbol in a term, for $u \in \Sigma \cup \mathcal{V}$ we denote by δ_u the function from $\Sigma \cup \mathcal{V}$ to $\{0, 1\}$ which is 1 only if the input is u. We say u *occurs* $|s|_u$ times in the term s, where $|x|_u := \delta_u(x)$ for $x \in \mathcal{V}$ and

$$|f(s_1, \dots, s_n)|_u = \delta_u(f) + |s_1|_u + \cdots + |s_n|_u \ .$$

Similarly $|s|$, the *size* or *length* of s, is defined. We put $|x| := 1$ for $x \in \mathcal{V}$ and

$$|f(s_1, \ldots, s_n)| := 1 + |s_1| + \cdots + |s_n| \ .$$

Of course, $|s|$ is the sum of the $|s|_u$ with $u \in \Sigma \cup \mathcal{V}$. By $\mathrm{dp}(s)$ we denote the *depth* of s, defined by $\mathrm{dp}(x) := 1$ for $x \in \mathcal{V}$ and

$$\mathrm{dp}(f(s_1, \ldots, s_n)) := 1 + \max\{\mathrm{dp}(s_j) : 1 \leqslant j \leqslant n\}.$$

For all terms s, size and depth are related via

$$\mathrm{dp}(s) \leqslant |s| \leqslant (\mathrm{Ar}(\Sigma) + 1)^{\mathrm{dp}(s)} \ . \tag{3.1}$$

The following result remains valid even if we allowed for infinite signatures.

Lemma 3.1 (Raney 1960). *Let Σ be a signature. For $n \in \mathbb{N}$ and $s \in \mathcal{T}(\Sigma)$, let $|s|_n$ denote the sum of the $|s|_f$ with $f \in \Sigma^{(n)}$. We have*

$$|s|_0 = 1 + \sum_{i=1}^{\infty} (i-1)|s|_i \ .$$

Proof. The proof is by induction on $|s|$. For $s \in \Sigma^{(0)}$, there is not much to do. So consider $s = f(s_1, \ldots, s_n)$ with $n \geqslant 1$. The induction hypothesis yields

$$\begin{aligned}
|f(s_1, \ldots, s_n)|_0 &= \sum_{j=1}^{n} |s_j|_0 \\
&= \sum_{j=1}^{n} \left(1 + \sum_{i=1}^{\infty} (i-1)|s_j|_i\right) \\
&= n + \sum_{j=1}^{n} \sum_{i=1}^{\infty} (i-1)|s_j|_i \\
&= 1 + (n-1) + \sum_{i=1}^{\infty} (i-1) \sum_{j=1}^{n} |s_j|_i \\
&= 1 + \sum_{i=1}^{\infty} (i-1) \left(\delta_{in} + \sum_{j=1}^{n} |s_j|_i\right) \\
&= 1 + \sum_{i=1}^{\infty} (i-1)|f(s_1, \ldots, s_n)|_i \ ,
\end{aligned}$$

where δ_{kl} is *Kronecker's* δ, which equals 1 if $k = l$ and 0 otherwise. Since almost all summands are 0, exchanging the summations is a legal action. □

Just as in (2.1), for a unary function symbol f and a term s, we adapt the n^{th} *iteration* $f^n(s)$, and we similarly extend this to function symbols of higher arities, using "$\cdot$" again to mark the position where the iterations happen.

A *substitution* is a mapping $\sigma\colon \mathcal{V} \to \mathcal{T}(\Sigma, \mathcal{V})$. It is *ground* if all variables are mapped to ground terms. We can extend σ to a mapping from $\mathcal{T}(\Sigma, \mathcal{V})$ to $\mathcal{T}(\Sigma, \mathcal{V})$ via

$$\sigma(f(s_1, \ldots, s_n)) := f(\sigma(s_1), \ldots, \sigma(s_n))$$

and abbreviate $\sigma(s)$ by $s\sigma$. If σ and τ are substitutions, then so is $\tau \circ \sigma$. Note that $\tau \circ \sigma$ is ground if σ or τ is.

Lemma 3.2. *For any term s and substitutions σ, τ we have $(s\sigma)\tau = s(\tau \circ \sigma)$.*

Definition 3.3. Let R be a binary relation on $\mathcal{T}(\Sigma, \mathcal{X})$, with $\mathcal{X} = \mathcal{V}$ or $\mathcal{X} = \emptyset$. We say that R is

i. *closed under (ground) contexts* if we have, for all n, for all $f \in \Sigma^{(n+1)}$, and for all (ground) terms $s, t, \bar{s}$,

$$s \mathrel{R} t \implies f(s_1, \ldots, s, \ldots, s_n) \mathrel{R} f(s_1, \ldots, t, \ldots, s_n)\ ,$$

ii. *closed under (ground) substitutions* if we have

$$s \mathrel{R} t \implies s\sigma \mathrel{R} t\sigma$$

for all terms s, t and all (ground) substitutions σ.

Of course, for $\mathcal{X} = \emptyset$ there is no need to explicitly consider closed versions as there are only closed terms.

Definition 3.4. For $l, r \in \mathcal{T}(\Sigma, \mathcal{V})$, the ordered pair (l, r) is a *rewrite rule* if $\mathcal{V}(l) \supseteq \mathcal{V}(r)$ and $l \notin \mathcal{V}$ holds. A *term rewriting system* (TRS) (over $\mathcal{T}(\Sigma, \mathcal{V})$ or, less precise, over Σ) is a *finite* set of rewrite rules. The *rewrite relation* induced by a TRS $\mathcal{R}$, denoted by $\to_{\mathcal{R}}$, is the smallest binary relation on $\mathcal{T}(\Sigma, \mathcal{V})$ which contains $\mathcal{R}$ and is closed under contexts and substitutions. By $\xrightarrow{+}_{\mathcal{R}}$ we denote the transitive closure of $\to_{\mathcal{R}}$, while the transitive and reflexive closure runs under $\xrightarrow{*}_{\mathcal{R}}$. We write $s \xrightarrow{\scriptscriptstyle\triangleright}_{\mathcal{R}} t$ if $s \to_{\mathcal{R}} t$ and t is the unique t' satisfying $s \to_{\mathcal{R}} t'$. The transitive closure of $\xrightarrow{\scriptscriptstyle\triangleright}_{\mathcal{R}}$ is $\xrightarrow{\triangleright}_{\mathcal{R}}$.

It is not difficult to see that $\to_{\mathcal{R}}$ is the smallest binary relation on $\mathcal{T}(\Sigma, \mathcal{V})$ which is closed under contexts and satisfies $l\sigma \to_{\mathcal{R}} r\sigma$ for all substitutions σ and all $(l, r) \in \mathcal{R}$.

When we focus on a TRS $\mathcal{R}$, we will usually drop a subscript $\mathcal{R}$. For example we write $\to$ instead of $\to_{\mathcal{R}}$. If (A) is a (named) rule from $\mathcal{R}$, then $\to_{\mathrm{A}}$ indicates a rewrite step due to this rule, i.e. a rewrite step in the TRS which solely contains the rule (A).

Definition 3.5. Let $\mathcal{R}$ be a TRS over Σ.

- A term s is *in normal form* (with respect to $\mathcal{R}$) if there is no term t satisfying $s \to_{\mathcal{R}} t$, and s is *a normal form* of s' if $s' \xrightarrow{*}_{\mathcal{R}} s$ and s is in normal form.
- $\mathcal{R}$ is *confluent for the term* s if, for all terms s' and s'', we have

$$s \xrightarrow{*}_{\mathcal{R}} s' \wedge s \xrightarrow{*}_{\mathcal{R}} s'' \implies (\exists t)(s' \xrightarrow{*}_{\mathcal{R}} t \wedge s'' \xrightarrow{*}_{\mathcal{R}} t) \ .$$

- $\mathcal{R}$ is *confluent* if it is confluent for all terms s.

Definition 3.6. A rewrite rule (l, r) over $\mathcal{T}(\Sigma, \mathcal{V})$ is called

- *duplicating* if there is a variable x with $|l|_x < |r|_x$,
- *increasing* if there is a substitution σ such that $|l\sigma| < |r\sigma|$,
- *size-preserving* if we have $|l\sigma| = |r\sigma|$ for all substitutions σ, and
- *embedding* if we have $l = f(x_1, \ldots, x_n)$, with $f \in \Sigma^{(\geqslant 1)}$ and pairwise distinct variables $\bar{x}$, and $r = x_i$. The set $\mathrm{Emb}(\Sigma)$ contains a representative of each embedding rule.

If all rules of a TRS $\mathcal{R}$ share one of the above properties, then $\mathcal{R}$ inherits the property. So, if all rules in $\mathcal{R}$ are duplicating, then $\mathcal{R}$ is duplicating.

Definition 3.7. A signature Σ which solely consists of unary symbols and exactly one constant is *monadic.*

We call a TRS over a monadic signature a *string rewriting system* (SRS) if the only constant of the signature does not occur in the rules. Since in this context terms may be identified with strings over the alphabet $\Sigma^{(1)}$, we will display them as strings, hence we drop the parentheses, leave out the constant resp. variable, and use ε for the *empty string.* In addition we will usually not mention the constant when introducing a monadic signature.

An $\mathcal{R}$-*derivation* (or simply *derivation*, or *reduction*) is any sequence $(s_i)_{1 \leqslant i < \alpha}$ over $\mathcal{T}(\Sigma, \mathcal{V})$ satisfying $1 < \alpha \leqslant \omega$ and, for all $i > 0$ with $i + 1 < \alpha$, $s_i \to_{\mathcal{R}} s_{i+1}$. If $\alpha = \omega$, then we call the derivation *infinite* and say it has *length* ω, and otherwise the derivation is *finite* and has length $\alpha - 1$. If the above derivation is finite, then it is a derivation *from* s_1 *to* $s_{\alpha-1}$. Of course, $s \xrightarrow{*} t$ is equivalent to the existence of a derivation from s to t, and if we add the condition $\alpha > 2$, then this is equivalent to $s \xrightarrow{+} t$.

A derivation $(s_i)_{1 \leqslant i < \alpha}$ is *bounded* by n if we have $|s_i| \leqslant n$ for all i satisfying $1 \leqslant i < \alpha$. We write $s \xrightarrow{*}^{\geqslant} t$ if there is a derivation from s to t which is bounded by $|s|$, we similarly use $\xrightarrow{*}^{\leqslant}$ if a derivation is bounded by $|t|$, and by $\xrightarrow{*}^{=}$ we indicate that all terms in a derivation have the same size. In the same sense $\xrightarrow{+}^{\leqslant}$, $\to^{\leqslant}$ and their relatives are defined. We should note that $\xrightarrow{+}^{\leqslant}$ is not the transitive closure of $\rightarrow^{\leqslant}$. The derivation is *linear* if we always have $s_i \overset{\rhd}{\rightarrow}_{\mathcal{R}} s_{i+1}$, and it is *noncycling* if $1 \leqslant i < j < \alpha$ implies $s_i \neq s_j$. We call $\rightarrow_{\mathcal{R}}$ (and $\mathcal{R}$) *noncycling* if all derivations are noncycling, and $\rightarrow_{\mathcal{R}}$ is *Noetherian* if all derivations are finite. In the latter case we say that $\mathcal{R}$ *terminates* or is *terminating.*

It is possible to transform any TM into an SRS in such a way that the TM halts for all configurations if and only if the associated SRS terminates.

Theorem 3.8. *Termination is an undecidable property of SRSs (and hence of TRSs). The question remains undecidable even if only TRSs containing exactly one rule are considered.*

Proof. Huet and Lankford (1978) demonstrated how to transform a TM $\mathcal{M}$ into an equivalent SRS $\mathcal{R}_{\mathcal{M}}$ as described above. The number of rules contained in $\mathcal{R}_{\mathcal{M}}$ depends on the number of instructions of $\mathcal{M}$. Dershowitz (1987b) showed that termination of TRSs containing only two rules is already undecidable, and later Dauchet (1992) (see also Lescanne (1994)) found a way to transform TMs into equivalent TRSs containing only one rule. □

The worst-case behavior of a terminating TRS is measured by the *derivation length function* or *(depth) complexity* $\mathrm{Dl}_{\mathcal{R}} \colon \mathbb{N} \to \mathbb{N}$. First one defines $\mathrm{dl}_{\mathcal{R}} \colon \mathcal{T}(\Sigma) \to \mathbb{N}$ (using the earlier introduced convention $\max \emptyset = 0$) by

$$\mathrm{dl}_{\mathcal{R}}(s) := \max\{\mathrm{dl}_{\mathcal{R}}(t) + 1 : s \rightarrow_{\mathcal{R}} t\}. \tag{3.2}$$

So $\mathrm{dl}_{\mathcal{R}}(s)$ is the maximal length of a derivation from s. This is well-defined since the finiteness of $\mathcal{R}$ implies $\rightarrow_{\mathcal{R}}$ is finitely branching. Now we put

$$\mathrm{Dl}_{\mathcal{R}}(n) := \max\{\mathrm{dl}_{\mathcal{R}}(s) : s \in \mathcal{T}(\Sigma) \wedge \mathrm{dp}(s) \leqslant n\}.$$

The condition $\mathrm{dp}(s) \leqslant n$ guarantees weak monotonicity of $\mathrm{Dl}_{\mathcal{R}}$. Note that it suffices to consider only closed terms, as any derivation may be transformed, under preservation of depths and sizes, to a derivation of equal length containing only closed terms: simply apply a substitution which maps all variables to constants.

One glance at Theorem 2.11 shows $\mathrm{dl}_{\mathcal{R}}$ corresponds to a certain order type.

Lemma 3.9. *For any terminating TRS $\mathcal{R}$ we have* $\mathrm{otype}(\mathcal{T}(\Sigma), \overset{+}{\leftarrow}_{\mathcal{R}}) \leqslant \omega$ *since* $\mathrm{dl}_{\mathcal{R}}(s)$ *is just the order type of s in the partial order* $(\mathcal{T}(\Sigma), \overset{+}{\leftarrow}_{\mathcal{R}})$.

An alternative (finer) measure is the *size complexity* $\mathrm{Dc}_{\mathcal{R}} \colon \mathbb{N} \to \mathbb{N}$, given by

$$\mathrm{Dc}_{\mathcal{R}}(n) := \max\{\mathrm{dl}_{\mathcal{R}}(s) : s \in \mathcal{T}(\Sigma) \wedge |s| \leqslant n\}.$$

Following (3.1), for all $n \in \mathbb{N}$ we have

$$\mathrm{Dc}_{\mathcal{R}}(n) \leqslant \mathrm{Dl}_{\mathcal{R}}(n) \leqslant \mathrm{Dc}_{\mathcal{R}}((\mathrm{Ar}(\Sigma) + 1)^n)\ . \tag{3.3}$$

If $\Sigma = \Sigma^{(\leqslant 1)}$, then $\mathrm{dp}(s) = |s|$ holds for all terms s, hence $\mathrm{Dl}_{\mathcal{R}}$ and $\mathrm{Dc}_{\mathcal{R}}$ coincide.

Hofbauer and Lautemann (1989) proposed a complementary measure:

> *"Particularly interesting would be the length of a shortest terminating derivation (in accordance with the usual complexity measure for nondeterministic models)."*

We take this up and define the *shortest derivation length function* or *shortest (depth) complexity* $\mathrm{SDl}_{\mathcal{R}}$ and the *shortest size complexity* $\mathrm{SDc}_{\mathcal{R}}$ like $\mathrm{Dl}_{\mathcal{R}}$ and $\mathrm{Dc}_{\mathcal{R}}$, yet both times $\mathrm{dl}_{\mathcal{R}}$ is replaced with $\mathrm{sdl}_{\mathcal{R}}$, which measures the length of a shortest derivation to a normal form:

$$\mathrm{sdl}_{\mathcal{R}}(s) := \min \{\mathrm{sdl}_{\mathcal{R}}(t) + 1 : s \to_{\mathcal{R}} t\}.$$

In many cases upper bounds on $\mathrm{SDl}_{\mathcal{R}}$ are located far below $\mathrm{Dl}_{\mathcal{R}}$. Chapter 6 contains TRSs for whom these two complexity measures are located in the same (rather high) complexity classes.

3.2 Abstract Orders on Terms

A general method to prove termination of a TRS is the construction of a well-founded order on the terms which is compatible with the rewrite relation. Since any infinite derivation corresponds to an infinite descending chain in the well-founded order, we immediately get termination. It turns out that it suffices to consider only those orders which are closed under contexts and substitutions. For obvious reasons these are called *rewrite orders*, while *reduction orders* are well-founded rewrite orders. A TRS terminates if and only if it is compatible with some reduction order.

Dershowitz (1979) introduced *simplification orders*, which are rewrite orders having the subterm property. They are the key object of this text. As a consequence of Kruskal's beautiful Tree Theorem, any such order is well-founded and hence a reduction order. Unfortunately there are terminating TRSs which are not compatible with any simplification order, as the SRS $ff \to fgf$ shows. It terminates because each rewrite step decreases the number of occurrences of ff, yet for any compatible simplification order $\prec$ we would get $ff \succ fgf \succ ff$.

Most of the following three sections is taken from Ferreira (1995), Hofbauer (1991), Middeldorp and Zantema (1997), and Zantema (1999, 2001).

Definition 3.10. Let $\mathcal{X}$ be equal to $\mathcal{V}$ or $\emptyset$. A partial order $(\mathcal{T}(\Sigma, \mathcal{X}), \prec)$

- is a *rewrite order* if it is closed under contexts and substitutions,
- is a *reduction order* if it is a well-founded rewrite order,

- is *compatible* with a TRS $\mathcal{R}$ if $\rightarrow_{\mathcal{R}} \subseteq \succ$, that is, if $s \rightarrow_{\mathcal{R}} t$ implies $s \succ t$,
- *normalizes* a TRS $\mathcal{R}$ if $l\sigma \succ r\sigma$ holds for all rules $(l, r) \in \mathcal{R}$ and all substitutions σ, and it
- has the *(weak) subterm property* if $s \succ t$ (respectively $s \succcurlyeq t$) holds as soon as t is a proper subterm of s.

By transitivity, $(\mathcal{T}(\Sigma, \mathcal{X}), \prec)$ has the subterm property if we have

$$f(s_1, \dots, s_n) \succ s_i$$

for all $n \geqslant 1$, for all $f \in \Sigma^{(n)}$, for all terms $\bar{s}$, and for all $i \in [1, n]$. A similar observation can be made for the weak subterm property.

The next Lemma contains some easy consequences of the above definitions.

Lemma 3.11. *Let $\mathcal{R}$ be a TRS over Σ.*

i. *If $\mathcal{R}$ is noncycling, then $(\mathcal{T}(\Sigma, \mathcal{V}), \overset{+}{\leftarrow}_{\mathcal{R}})$ is the least rewrite order living on $\mathcal{T}(\Sigma, \mathcal{V})$ which is compatible with $\mathcal{R}$.*
ii. *If $\mathcal{R}$ is terminating, then $(\mathcal{T}(\Sigma, \mathcal{V}), \overset{+}{\leftarrow}_{\mathcal{R}})$ is the least reduction order on $\mathcal{T}(\Sigma, \mathcal{V})$ compatible with $\mathcal{R}$.*
iii. *If a well-founded partial order $(\mathcal{T}(\Sigma, \mathcal{V}), \prec)$ is compatible with $\mathcal{R}$, then $\mathcal{R}$ terminates.*
iv. *$\mathcal{R}$ terminates if and only if there is a reduction order $(\mathcal{T}(\Sigma, \mathcal{V}), \prec)$ which is compatible with $\mathcal{R}$.*
v. *A rewrite order $(\mathcal{T}(\Sigma, \mathcal{V}), \prec)$ is compatible with $\mathcal{R}$ if $\mathcal{R} \subseteq \succ$, that is, if we have $l \succ r$ for all rules (l, r) of $\mathcal{R}$.*

Proof. For (i) transitivity of $\overset{+}{\leftarrow}$ is obvious, while its irreflexivity follows from the fact that $\mathcal{R}$ is noncycling. Thus $\overset{+}{\leftarrow}$ is a partial order on $\mathcal{T}(\Sigma, \mathcal{V})$. Any rewrite order compatible with $\mathcal{R}$ contains $\overset{+}{\leftarrow}$. From this we can easily infer (ii), as termination of $\mathcal{R}$ implies $\mathcal{R}$ is noncycling. It suffices for (iii) to note that any infinite $\mathcal{R}$-derivation is an infinite $\prec$-descending chain. One direction of (iv) is contained in (ii), while the opposite direction is a direct consequence of (iii). For (v) we just have to recall that, by definition, $\rightarrow_{\mathcal{R}}$ is the closure of $\mathcal{R}$ under contexts and substitutions. □

Terms over a signature Σ can be considered as trees labeled with elements from Σ. This observation leads to *simplification orders*, the most prominent application of Kruskal's Tree Theorem within term rewriting theory.

Definition 3.12 (Dershowitz 1979). A rewrite order $(\mathcal{T}(\Sigma, \mathcal{V}), \prec)$ having the subterm property is a *simplification order*. We call a TRS *simplifying* if it is compatible with a simplification order.

We should mention that Kaplan (1987) introduced the phrase “simplifying”. Recall from Definition 3.6 that $\mathrm{Emb}(\Sigma)$ is the set of embedding rules.

Definition 3.13. We call $(\mathcal{T}(\Sigma, \mathcal{V}), \overset{*}{\leftarrow}_{\mathrm{Emb}(\Sigma)})$ *embedding* and write $\leqslant_{\mathsf{emb}}$ for $\overset{*}{\leftarrow}_{\mathrm{Emb}(\Sigma)}$.

The set of closed terms over Σ can be identified with a subset of the finite rooted trees with labels from Σ. If we order Σ by the empty binary relation $\prec$, thus making distinct symbols incomparable, and restrict the induced homeomorphic embedding relation $\prec_{\mathsf{hemb}}$ to the trees representing closed terms, we end up with $<_{\mathsf{emb}}$.

Lemma 3.14. *Let Σ be a signature.*

i. *A rewrite order $(\mathcal{T}(\Sigma, \mathcal{V}), \prec)$ has the subterm property if and only if it is compatible with* $\mathrm{Emb}(\Sigma)$*, i.e. if* $\mathrm{Emb}(\Sigma) \subseteq \succ$.
ii. $(\mathcal{T}(\Sigma, \mathcal{V}), <_{\mathsf{emb}})$ *is the smallest rewrite order on $\mathcal{T}(\Sigma, \mathcal{V})$ having the subterm property.*

There are various incarnations of Kruskal’s Tree Theorem in term rewriting theory. We present the most simple finite version. A thorough presentation of more advanced (and stronger) versions is Middeldorp and Zantema (1997).

Theorem 3.15 (Kruskal’s Tree Theorem, finite version). *For any sequence $(s_i)_{i<\omega}$ over $\mathcal{T}(\Sigma)$ there exist $i < j$ with $s_i \leqslant_{\mathsf{emb}} s_j$.*

Proof. Following the remark below Definition 3.13, $<_{\mathsf{emb}}$ can be seen as a restriction of $\prec_{\mathsf{hemb}}$ to $\mathcal{T}(\Sigma)$, where Σ is ordered by the empty binary relation $\prec$. As Σ is finite, Lemma 2.51 ensures this partial order is a PWO. Theorem 2.55, the general version of Kruskal’s Tree Theorem, implies the induced homeomorphic embedding relation, whose restriction to $\mathcal{T}(\Sigma)$ coincides with $\leqslant_{\mathsf{emb}}$, is a PWO. Now Lemma 2.52 applies. □

Corollary 3.16. *Simplification orders are well-founded.*

From this and Lemma 3.11.iv we can infer the following fundamental result.

Theorem 3.17 (Dershowitz 1979). *Any simplifying TRS is terminating.*

Dershowitz (1982) considers more general simplification orders, which are based on preorders. It should be obvious how to transfer the occuring notions from partial orders to preorders. The following result shows that, in a term rewriting setting, a transition to preorders has no effect.

Theorem 3.18 (Middeldorp and Zantema 1997). *A TRS $\mathcal{R}$ over signature Σ is simplifying if and only if there exists a preorder $(\mathcal{T}(\Sigma, \mathcal{V}), \prec)$ which is closed under contexts, has the subterm property, and normalizes $\mathcal{R}$.*

An easy application of Lemma 3.14 yields the following very useful result.

Lemma 3.19. *Let $\mathcal{R}$ be a TRS over Σ. $\mathcal{R}$ is simplifying if and only if the TRS $\mathcal{R} \cup \mathrm{Emb}(\Sigma)$ is terminating.*

We can use this result to show not all terminating TRSs are simplifying.

Lemma 3.20. *There is a terminating SRS which is not simplifying.*

Proof. Consider the SRS $\mathcal{R}$ over signature $\{f, g\}$ containing only $ff \to fgf$. Its termination is achieved by showing that every rewrite step decreases the number of occurrences of the substring ff. However, if we add to $\mathcal{R}$ the embedding rule $g \to \varepsilon$, the cycling derivation $ff \to fgf \to ff$ is possible. Lemma 3.19 shows $\mathcal{R}$ is not simplifying. □

3.3 Semantic Orders on Terms

A common approach to the construction of compatible reduction orders is to look at the semantics of the TRS. It is sometimes possible to give an *interpretation* into a well-known object which happens to be equipped with a well-founded order. This order can be lifted to $\mathcal{T}(\Sigma, \mathcal{V})$, inducing a well-founded order on $\mathcal{T}(\Sigma, \mathcal{V})$. The obvious aim is to find an interpretation such that the induced order is a reduction order compatible with the TRS in consideration.

In a related approach one tries to provide a compatible Σ-*algebra*. Such an object consists of a set P which is equipped with a partial order and with interpreting functions $[f]$ (of appropriate arities) living on P, for all $f \in \Sigma$. This induces a canonical homomorphic interpretation. A Σ-algebra is called *monotone* if all $[f]$ are. It is a good idea to take a closer look at monotone Σ-algebras, as here the induced order on $\mathcal{T}(\Sigma, \mathcal{V})$ is a rewrite order and the validity of a termination proof solely relies on the well-foundedness of the underlying partial order. This approach is general, i.e. a TRS is terminating if and only it is compatible with a well-founded monotone Σ-algebra.

Depending on the Σ-algebras we can classify the terminating TRSs. Of special importance are those compatible with monotone Σ-algebras having the subterm property, as these are exactly the TRSs compatible with simplification orders. Zantema (1993, 1994) introduced the important subclass of *totally terminating* TRSs, collecting the TRSs compatible with a monotone Σ-algebra living on some well-order. As well-orders are order-isomorphic to ordinals, there is always some ordinal α which can serve as the universe of the Σ-algebra. Under these circumstances we also speak of α-termination. The ordinals α can be used to classify the totally terminating TRSs. Roughly speaking, the larger a limit ordinal α is the more TRSs are α-terminating. The first nontrivial member of the total termination hierarchy, the set of ω-terminating TRSs, already carries a very rich structure. Since here the interpreting functions are number-theoretic, we can make use of subrecursive hierarchies as a tool for classification.

3.3.1 Interpretations

Interpretations are a canonical means to prove termination.

Definition 3.21. Let $(P, \prec)$ be a partial order.

- A mapping $\mathcal{I}\colon \mathcal{T}(\Sigma) \to P$ is an *interpretation* of $\mathcal{T}(\Sigma)$ in $(P, \prec)$.
- The interpretation $\mathcal{I}$ induces a binary relation $\prec_{\mathcal{I}}$ on $\mathcal{T}(\Sigma)$ via
$$s \prec_{\mathcal{I}} t \;:\Longleftrightarrow\; \mathcal{I}(s) \prec \mathcal{I}(t)\ .$$
This can be lifted to a binary relation on $\mathcal{T}(\Sigma, \mathcal{V})$ via
$$\begin{aligned} s \prec_{\mathcal{I}} t \;&:\Longleftrightarrow \mathcal{I}(s\sigma) \prec \mathcal{I}(t\sigma) \text{ for all ground substitutions } \sigma \\ &\Longleftrightarrow s\sigma \prec_{\mathcal{I}} t\sigma \text{ for all ground substitutions } \sigma\ . \end{aligned}$$

Note that $\prec_{\mathcal{I}}$ is not total on $\mathcal{T}(\Sigma, \mathcal{V})$ because distinct variables are always incomparable. Even if $(P, \prec)$ is total, $\prec_{\mathcal{I}}$ need not be total on $\mathcal{T}(\Sigma)$ since distinct terms may be mapped to the same member of P.

Lemma 3.22. *Let $\mathcal{I}$ be an interpretation of $\mathcal{T}(\Sigma)$ in the partial order $(P, \prec)$, and let $(\mathcal{T}(\Sigma, \mathcal{V}), \prec_{\mathcal{I}})$ be the induced relation.*

i. *$\mathcal{I}$ is an order preserving mapping from $(\mathcal{T}(\Sigma), \prec_{\mathcal{I}})$ to $(P, \prec)$.*
ii. *$(\mathcal{T}(\Sigma, \mathcal{V}), \prec_{\mathcal{I}})$ is a partial order which is well-founded if $(P, \prec)$ is.*
iii. *$(\mathcal{T}(\Sigma, \mathcal{V}), \prec_{\mathcal{I}})$ is closed under substitutions.*
iv. *If $(\mathcal{T}(\Sigma), \prec_{\mathcal{I}})$ is closed under (ground) contexts, then $(\mathcal{T}(\Sigma, \mathcal{V}), \prec_{\mathcal{I}})$ is closed under contexts.*
v. *If $(\mathcal{T}(\Sigma), \prec_{\mathcal{I}})$ has the subterm property, then so does $(\mathcal{T}(\Sigma, \mathcal{V}), \prec_{\mathcal{I}})$.*
vi. *If $(\mathcal{T}(\Sigma), \prec_{\mathcal{I}})$ is well-founded, then so is $(\mathcal{T}(\Sigma, \mathcal{V}), \prec_{\mathcal{I}})$.*

Proof. For (i), everything immediately follows from the definition of $\prec_{\mathcal{I}}$, while transitivity and irreflexivity needed in (ii) are inherited from $(P, \prec)$. Via $\mathcal{I}$, any infinite $\prec_{\mathcal{I}}$-descending chain corresponds to an infinite $\prec$-descending chain. In order to treat (iii), we consider a substitution τ and $s, t \in \mathcal{T}(\Sigma, \mathcal{V})$ with $s \succ_{\mathcal{I}} t$. Recall that $\sigma \circ \tau$ is a ground substitution if σ is. By Lemma 3.2 we get

$$\begin{aligned} s \succ_{\mathcal{I}} t \;&\Longleftrightarrow s\sigma \succ_{\mathcal{I}} t\sigma \text{ for all ground substitutions } \sigma \\ &\Longrightarrow s(\sigma \circ \tau) \succ_{\mathcal{I}} t(\sigma \circ \tau) \text{ for all ground substitutions } \sigma \\ &\Longleftrightarrow (s\tau)\sigma \succ_{\mathcal{I}} (t\tau)\sigma \text{ for all ground substitutions } \sigma \\ &\Longleftrightarrow s\tau \succ_{\mathcal{I}} t\tau\ . \end{aligned}$$

For (iv), let $(\mathcal{T}(\Sigma), \prec_{\mathcal{I}})$ be closed under ground contexts and take $f \in \Sigma^{(n+1)}$, a ground substitution σ, and $s, t, \bar{s} \in \mathcal{T}(\Sigma, \mathcal{V})$ satisfying $s \succ_{\mathcal{I}} t$. Here we get

$$\begin{aligned} s \succ_{\mathcal{I}} t &\implies s\sigma \succ_{\mathcal{I}} t\sigma \\ &\implies f(s_1\sigma, \dots, s\sigma, \dots, s_n\sigma) \succ_{\mathcal{I}} f(s_1\sigma, \dots, t\sigma, \dots, s_n\sigma) \\ &\iff f(s_1, \dots, s, \dots, s_n)\sigma \succ_{\mathcal{I}} f(s_1, \dots, t, \dots, s_n)\sigma \ , \end{aligned}$$

which, by arbitrarity of σ, implies $f(s_1, \dots, s, \dots, s_n) \succ_{\mathcal{I}} f(s_1, \dots, t, \dots, s_n)$. The proof of (v) should meanwhile be trivial. For (vi) we just have to incorporate (iii): the application of a ground substitution transforms any infinite descending chain over $\mathcal{T}(\Sigma, \mathcal{V})$ into an infinite descending chain over $\mathcal{T}(\Sigma)$. □

Definition 3.23. Let $\mathcal{I}$ be an interpretation of $\mathcal{T}(\Sigma)$ in the partial order $(P, \prec)$. We say that

- $\mathcal{I}$ is *monotone* if, for all n, all $f \in \Sigma^{(n+1)}$, and all $s, t, \bar{s} \in \mathcal{T}(\Sigma)$, we have

$$\mathcal{I}(s) \succ \mathcal{I}(t) \implies \mathcal{I}(f(s_1, \dots, s, \dots, s_n)) \succ \mathcal{I}(f(s_1, \dots, t, \dots, s_n)) \ ,$$

- $\mathcal{I}$ is *weakly monotone* if, for all $n, f, s, t, \bar{s}$ as above, we get

$$\mathcal{I}(s) \succcurlyeq \mathcal{I}(t) \implies \mathcal{I}(f(s_1, \dots, s, \dots, s_n)) \succcurlyeq \mathcal{I}(f(s_1, \dots, t, \dots, s_n)) \ ,$$

- $\mathcal{I}$ has the *subterm property* if $\mathcal{I}(s) \succ \mathcal{I}(t)$ holds as soon as $s >_{\mathsf{sub}} t$, and
- $\mathcal{I}$ is a *normalization* of a TRS $\mathcal{R}$ if, for all rules (l, r) in $\mathcal{R}$ and for all ground substitutions σ, we have $\mathcal{I}(l\sigma) \succ \mathcal{I}(r\sigma)$.

Again, $\mathcal{I}$ has the subterm property if and only if

$$\mathcal{I}(f(s_1, \dots, s_n)) \succ \mathcal{I}(s_i)$$

holds for all $n \geqslant 1$, all $f \in \Sigma^{(n)}$, all $\bar{s} \in \mathcal{T}(\Sigma)$, and all $i \in [1, n]$.

Lemma 3.24. *Let $\mathcal{I}$ be an interpretation of $\mathcal{T}(\Sigma)$ in the partial order $(P, \prec)$, and let $\mathcal{R}$ be a TRS over Σ.*

i. *If $\mathcal{I}$ is monotone, then $(\mathcal{T}(\Sigma, \mathcal{V}), \prec_{\mathcal{I}})$ is a rewrite order. Additional well-foundedness of $(P, \prec)$ implies $(\mathcal{T}(\Sigma, \mathcal{V}), \prec_{\mathcal{I}})$ is a reduction order.*

ii. *If $\mathcal{I}$ has the subterm property, then so does $(\mathcal{T}(\Sigma, \mathcal{V}), \prec_{\mathcal{I}})$.*

iii. *$\mathcal{I}$ normalizes $\mathcal{R}$ if and only if $R \subseteq \succ_{\mathcal{I}}$.*

iv. *If $\mathcal{I}$ is a monotone normalization of $\mathcal{R}$, then $\mathcal{I}$ embeds $(\mathcal{T}(\Sigma), \xleftarrow{+}_{\mathcal{R}})$ into $(P, \prec)$, and $\mathcal{R}$ is compatible with $(\mathcal{T}(\Sigma, \mathcal{V}), \prec_{\mathcal{I}})$. Additional well-foundedness of $(P, \prec)$ implies termination of $\mathcal{R}$.*

Proof. We start with (i). Lemma 3.22.ii shows $(\mathcal{T}(\Sigma, \mathcal{V}), \prec_{\mathcal{I}})$ is a partial order, and the monotonicity of $\mathcal{I}$ implies $(\mathcal{T}(\Sigma), \prec_{\mathcal{I}})$ is closed under (ground) contexts. By Lemma 3.22.iv,iii, $(\mathcal{T}(\Sigma, \mathcal{V}), \prec_{\mathcal{I}})$ is a rewrite order, and by Lemma 3.22.ii it is even a reduction order, provided that $(P, \prec)$ is well-founded. In a similar way (ii) follows from Lemma 3.22.v. As the definition of normalization is just a copy of the definition of $\prec_{\mathcal{I}}$, (iii) is obvious. Finally, (iv) is established by (i), (iii), Lemma 3.22.i and Lemma 3.11. □

Lemma 3.25. *Let $\mathcal{R}$ be a terminating TRS over the signature Σ.*

i. *The function $\mathrm{dl}_{\mathcal{R}}$ is the minimal embedding of $(\mathcal{T}(\Sigma), \xleftarrow{+}_{\mathcal{R}})$ into $(\mathbb{N}, <)$: for any such embedding $\mathcal{I}$ we have $(\forall s \in \mathcal{T}(\Sigma))(\mathcal{I}(s) \geqslant \mathrm{dl}_{\mathcal{R}}(s))$.*
ii. *The function $\mathrm{dl}_{\mathcal{R}}$ is minimal among the monotone interpretations of $\mathcal{T}(\Sigma)$ in $(\mathbb{N}, <)$ which normalize $\mathcal{R}$.*

Proof. The function $\mathrm{dl}_{\mathcal{R}} \colon \mathcal{T}(\Sigma) \to \mathbb{N}$ from (3.2) is well-defined and satisfies

$$(\forall s, t \in \mathcal{T}(\Sigma))(s \to_{\mathcal{R}} t \implies \mathrm{dl}_{\mathcal{R}}(s) \geqslant \mathrm{dl}_{\mathcal{R}}(t) + 1) .$$

Thus it is an embedding of $(\mathcal{T}(\Sigma), \xleftarrow{+}_{\mathcal{R}})$ into $(\mathbb{N}, <)$. The minimality follows by induction on $\mathrm{dl}_{\mathcal{R}}(s)$. For (ii) we use Lemma 3.24.iv and (i). □

The Lemma shows it actually suffices to consider only monotone interpretations in the well-order $(\mathbb{N}, <)$. However, this often turns out to be much more difficult than to construct an interpretation in some other well-founded order.

3.3.2 Σ-algebras and the Termination Hierarchy

A pleasant way of defining an interpretation is by Σ-algebras.

Definition 3.26 (Lankford 1975, Zantema 1994, 1999, 2001). Let Σ be a signature. A Σ-*algebra* is a triple $(A, \prec, \mathcal{F})$ such that $(A, \prec)$ is a partial order and $\mathcal{F}$ contains, for all $f \in \Sigma^{(n)}$, a function $[f]_A \colon A^n \to A$. The Σ-algebra

- is *(weakly) monotone* if all $[f]_A$ (with $f \in \Sigma^{(\geqslant 1)}$) are (weakly) monotone,
- is *total* if $(A, \prec)$ is total,
- is *well-founded* if $(A, \prec)$ is well-founded,
- has the *(weak) subterm property* if all $[f]_A$ (with $f \in \Sigma^{(\geqslant 1)}$) have it,
- is *simple monotone* if it is monotone and has the subterm property, and
- is *weak monotone* if it is weakly monotone and has the weak subterm property.

In analogy with Definition 2.7, for ordinals α we abbreviate $(\alpha, <, \mathcal{F})$ by $(\alpha, \mathcal{F})$.

By convention, our signatures contain at least one constant. Thus the partial order underlying the Σ-algebra is nonempty. Zantema (1999) calls monotone Σ-algebras with the *weak* subterm property "simple monotone". For our purposes, both two approaches are equivalent, see Proposition 3.35.

A Σ-algebra is what is called a $\Sigma \cup \{<\}$*-structure* in *first order logic* (with $\prec$ playing the rôle of $<$), and the operations of the algebra are called interpretations (of the function symbols of the underlying language) there. We are now going to define appropriate restrictions of *assignments* and *satisfaction* $\vDash$.

Definition 3.27. Let $(A, \prec, \mathcal{F})$ be a Σ-algebra. The function

$$[\![\cdot,\cdot]\!]_A \colon \mathcal{T}(\Sigma, \mathcal{V}) \times {}^{\mathcal{V}}\!A \to A$$

is defined via $[\![x,\rho]\!]_A := \rho(x)$ and

$$[\![f(s_1,\dots,s_n),\rho]\!]_A := [f]_A([\![s_1,\rho]\!]_A,\dots,[\![s_n,\rho]\!]_A) \ .$$

This induces a binary relation $\prec_A$ on $\mathcal{T}(\Sigma,\mathcal{V})$ by

$$s \prec_A t \ :\Longleftrightarrow \ (\forall \rho \in {}^{\mathcal{V}}\!A)([\![s,\rho]\!]_A \prec [\![t,\rho]\!]_A) \ .$$

For $s \in \mathcal{T}(\Sigma)$, the value of $[\![s,\rho]\!]_A$ does not depend on ρ. We call this value $[\![s]\!]_A$ and thus get an interpretation $[\![\cdot]\!]_A$ of $\mathcal{T}(\Sigma)$ in $(A,\prec)$.

Just as with interpretations, totality of $(A,\prec)$ does not imply $\prec_A$ is total on $\mathcal{T}(\Sigma)$, as distinct terms may be mapped to the same member of A.

If it is clear from the context to which Σ-algebra we refer, we will usually drop the subscript A in $[\![\cdot]\!]_A$ and $[\cdot]_A$.

Definition 3.28. A TRS $\mathcal{R}$ over Σ and a Σ-algebra $(A,\prec,\mathcal{F})$ are *compatible* if we have $\mathcal{R} \subseteq \succ_A$, i.e. if $l \succ_A r$ holds for all rules (l,r) of $\mathcal{R}$.

We will soon see that, for monotone Σ-algebras, it is legal to speak of "compatibility" here. In continuation of our digression to first order logic, $(A,\prec,\mathcal{F})$ is compatible with $\mathcal{R}$ if and only if it is a *model* of the rules, that is, if we have $(A,\prec,\mathcal{F}) \vDash l > r$ for all $(l,r) \in \mathcal{R}$.

The following result establishes a connection between the interpretation $[\![\cdot]\!]_A$ (in the sense of Definition 3.23) and the underlying Σ-algebra.

Lemma 3.29. *Let $(A,\prec,\mathcal{F})$ be a Σ-algebra.*

i. *If $(A,\prec,\mathcal{F})$ is (weakly) monotone, then so is the interpretation $[\![\cdot]\!]_A$.*
ii. *If $(A,\prec,\mathcal{F})$ has the (weak) subterm property, then so does $[\![\cdot]\!]_A$.*
iii. *If a TRS $\mathcal{R}$ is compatible with $(A,\prec,\mathcal{F})$, then $[\![\cdot]\!]_A$ normalizes $\mathcal{R}$.*

We can now present a counterpart to Lemma 3.22 and Lemma 3.24, with almost identical proofs.

Lemma 3.30. *Let $(A, \prec, \mathcal{F})$ be a Σ-algebra, further let $\mathcal{R}$ be a TRS over Σ.*

i. *$(\mathcal{T}(\Sigma, \mathcal{V}), \prec_A)$ is a partial order which is well-founded if $(A, \prec)$ is.*
ii. *$(\mathcal{T}(\Sigma, \mathcal{V}), \prec_A)$ is closed under substitutions.*
iii. *If $(A, \prec, \mathcal{F})$ is monotone, then $(\mathcal{T}(\Sigma, \mathcal{V}), \prec_A)$ is a rewrite order. It is a reduction order if $(A, \prec)$ is also well-founded.*
iv. *If $(A, \prec, \mathcal{F})$ has the (weak) subterm property, then so does $(\mathcal{T}(\Sigma, \mathcal{V}), \prec_A)$.*
v. *If $(A, \prec, \mathcal{F})$ is monotone and compatible with $\mathcal{R}$, then $(\mathcal{T}(\Sigma, \mathcal{V}), \prec_A)$ is compatible with $\mathcal{R}$.*

At first sight, Σ-algebras may appear less powerful than arbitrary interpretations, as $[\![s]\!]_A = [\![t]\!]_A$ already implies $[\![f(\dots, s, \dots)]\!]_A = [\![f(\dots, t, \dots)]\!]_A$, whereas interpretations are allowed to be less uniform. The following result shows that, concerning termination, it suffices to consider only Σ-algebras.

Proposition 3.31. *Let $\mathcal{R}$ be a TRS over Σ. $\mathcal{R}$ terminates if and only if it is compatible with a well-founded monotone Σ-algebra.*

Proof. First let $\mathcal{R}$ be compatible with a well-founded monotone Σ-algebra. By Lemma 3.30.iii,v, this ensures $\mathcal{R}$ is compatible with a reduction order, and now Lemma 3.11.iv tells us $\mathcal{R}$ terminates.

Let us turn to the opposite direction and consider a terminating $\mathcal{R}$. By Lemma 3.11.ii, $(\mathcal{T}(\Sigma, \mathcal{V}), \overset{+}{\leftarrow}_{\mathcal{R}})$ is a reduction order on $\mathcal{T}(\Sigma, \mathcal{V})$ compatible with $\mathcal{R}$. Thus the Σ-algebra $(\mathcal{T}(\Sigma), \overset{+}{\leftarrow}_{\mathcal{R}}, \mathcal{F})$ using the operations

$$[f](s_1, \dots, s_n) := f(s_1, \dots, s_n) \tag{3.4}$$

is well-founded and monotone. We get compatibility of $\mathcal{R}$ with the Σ-algebra because the mappings ρ from $\mathcal{V}$ to $\mathcal{T}(\Sigma)$ coincide with the ground substitutions and because $[\![s, \rho]\!]_A = s\rho$ holds for all terms s. □

Zantema (1993, 1994, 1999, 2001) introduced and considered a whole hierarchy of termination proof methods. We present an extended selection of it.

Definition 3.32. Let $\mathcal{R}$ be a TRS over Σ. We call $\mathcal{R}$

- *simply terminating* if it is compatible with a simple monotone Σ-algebra,
- *(weakly) α-terminating* for an ordinal α if it is compatible with a (weak) monotone Σ-algebra $(\alpha, \mathcal{F})$, and
- *(weakly) totally terminating* if it is (weakly) α-terminating for some α.

In this context we will often speak of *simple termination*, *(weak) total termination*, and *(weak) α-termination.* Stimulated by Definition 2.70, a TRS is called *$<\alpha$-terminating* if it is β-terminating for some $\beta < \alpha$.

Inside of ω-termination there is a further hierarchy (mainly due to Hofbauer (1991)), focusing on the interpreting functions of the Σ-algebra. If they are linear we get *linear termination* (LT), and for polynomials we have *polynomial termination* (PT). This can be extended to *exponential*, *elementary*, *$\mathcal{E}_n$-*, *primitive recursive*, *$\mathcal{M}_n$-*, *multiple recursive*, *$<\alpha$-recursive*, and *α-recursive termination.*

We mentioned above that, by convention, our Σ-algebras are nonempty. Thus for us there is no such thing as 0-termination. The phrase "simple termination" was coined by Kurihara and Ohuchi (1990). Zantema (1993, 1994) introduced total termination as compatibility with a total well-founded monotone Σ-algebra. By Theorem 2.8.vii, this is equivalent to our definition. Simple termination is directly connected to simplification orders, hence termination is easily achieved.

Proposition 3.33. *A TRS is simply terminating if and only if it is compatible with a simplification order.*

Proof. Let $\mathcal{R}$ be a TRS over Σ whose simple termination is witnessed by the simple monotone Σ-algebra $(A, \prec, \mathcal{F})$. By Lemma 3.30.iii,iv, $(\mathcal{T}(\Sigma, \mathcal{V}), \prec_A)$, is a rewrite order which has the subterm property, hence a simplification order. Lemma 3.30.v yields compatibility of $\mathcal{R}$ and $(\mathcal{T}(\Sigma, \mathcal{V}), \prec_A)$.

Now let $\mathcal{R}$ be compatible with the simplification order $(\mathcal{T}(\Sigma, \mathcal{V}), \prec)$. It is easy to see that the canonical Σ-algebra $(\mathcal{T}(\Sigma), \prec, \mathcal{F})$ (where the Σ-operations are defined as in (3.4)) is simple monotone and compatible with $\mathcal{R}$. □

We can combine this result and Corollary 3.16.

Corollary 3.34. *Simple termination implies termination.*

It is possible to weaken the conditions we required for simple termination.

Proposition 3.35 (Zantema 1999). *A TRS is simply terminating if and only if it is compatible with a monotone Σ-algebra having the weak subterm property.*

Simple and total termination may appear independent. This is deceiving.

Proposition 3.36 (Zantema 1994). *Total termination implies simple termination.*

Proof. Let $\mathcal{R}$ be a TRS over signature Σ whose total termination is witnessed by $(\alpha, \mathcal{F})$. By $\mathcal{P}$ we denote the lexicographic product of α and $(\mathbb{N}, <)$. This is turned into a Σ-algebra using the operations

$$[\![f]\!]_{\mathcal{P}}((\gamma_1, m_1), \dots, (\gamma_n, m_n)) := ([\![f]\!]_{\alpha}(\bar{\gamma}), m_1 + \cdots + m_n + 1) .$$

A moment's reflection shows this Σ-algebra is simple monotone. □

On the other hand, simple and total termination do not coincide. The TRS containing the two rules $f(a) \to f(b)$ and $g(b) \to g(a)$ is simply terminating but not totally terminating (since a and b have to be incomparable in a compatible monotone Σ-algebra).

In Theorem 3.8 we saw that termination is an undecidable property of TRSs. There are further undecidability results which relate to some of the recently introduced subclasses and, yet again, to TRSs containing only one rule.

Theorem 3.37.

i. *Simple termination is undecidable, even for one-rule TRSs.*

ii. *Total termination is undecidable.*

iii. *ω-termination is undecidable, even within total termination.*

Proof. Caron (1991) transfered Theorem 3.8 to the context of simple termination by reducing simple termination to the *Correspondence Problem* for linear bounded automata posed by Post (1947). Middeldorp and Gramlich (1995) refined the approach of Dauchet (1992) (see Theorem 3.8) to linear bounded automata and simple termination of TRSs containing only one rule. Zantema (1995) applied yet another transformation of Post's Correspondence Problem to prove undecidability of total termination. The (relative) undecidability of ω-termination was established by Geser (1997). □

Lemma 3.38. *Weak α-termination follows from α-termination. Hence total termination implies weak total termination.*

Proof. Following Zantema (1999), it suffices to show that any total well-founded monotone Σ-algebra is already weak monotone. As monotonicity of a function implies its weak monotonicity, it remains to establish its weak subterm property. This is done by an invocation of Lemma 2.20. □

Proposition 3.31 shows total termination is an honest concept, as it implies termination. Various natural questions arise:

- Depending on the signature, which α carry an α-terminating TRS?
- For which β does (weak) α-termination imply β-termination?
- Which α carry an α-terminating TRS which is not $<\alpha$-terminating?

We will present a collection of partial answers to these question on the following pages. They will later be joined by Corollary 4.11, which shows that the α of the third question are to be found below $\vartheta(\Omega^\omega)$.

Lemma 3.39. *If a TRS $\mathcal{R}$ over signature Σ is compatible with a simplification order $(\mathcal{T}(\Sigma, \mathcal{V}), \prec)$ which is linear on ground terms, then $\mathcal{R}$ is* $\mathrm{otype}(\mathcal{T}(\Sigma), \prec)$*-terminating.*

Proof. Consider the Σ-algebra $(\mathcal{T}(\Sigma), \prec, \mathcal{F})$, where the $[f]$ are defined canonically as in (3.4). It is easy to see this Σ-algebra is monotone, total (by premise), and well-founded (by simplification). Its compatibility with $\mathcal{R}$ follows from the compatibility of $\mathcal{R}$ with $(\mathcal{T}(\Sigma, \mathcal{V}), \prec)$. Indeed, by the latter we have $l\sigma \succ r\sigma$ for all ground substitutions σ. The same argumentation as in the proof of Proposition 3.31 yields $[\![l, \sigma]\!] \succ [\![r, \sigma]\!]$. Thus we get otype$(\mathcal{T}(\Sigma), \prec)$-termination. □

Proposition 3.40 (Ferreira 1995, 5.33, 5.25). *Let $\mathcal{R}$ be an α-terminating TRS over signature Σ.*

i. *If $\mathcal{R}$ is a nonempty SRS, then α is a limit ordinal.*
ii. *If $\Sigma^{(\geqslant 2)} \neq \emptyset$, then α is a principal ordinal.*

Recall from Lemma 2.28.i that the principal ordinals are just the ordinals of the shape ω^β with $\beta \geqslant 0$. A direct consequence of the above Proposition is that, as soon as a signature contains a function symbol of arity greater than 1, any TRS over this signature is not $\alpha + \alpha$-terminating, as $\alpha + \alpha$ is not principal.

Proposition 3.41 (Ferreira 1995, 5.34). *For any $\alpha \leqslant \omega$, there is a SRS which is ω^α-terminating but not $<\omega^\alpha$-terminating. Hence total termination does not imply ω-termination.*

Proof (sketch). The following examples can also be found in Ferreira and Zantema (1996), and the case $\alpha = 2$ was treated before by Zantema (1992, 1994).

For $\alpha = 0$ the empty SRS suffices, while for $\alpha = 1$ we just have to consider the SRS $\{f \to \varepsilon\}$. Let us turn to the case $1 < \alpha < \omega$. Define, over the signature $\Sigma_\alpha := \{f_1, \ldots, f_\alpha\}$, the SRS $\mathcal{R}_\alpha$ containing the rules $f_i f_{i+1} \to f_{i+1} f_i f_i$ for all i with $1 \leqslant i < \alpha$. $\mathcal{R}_\alpha$ is ω^α-terminating but not $<\omega^\alpha$-terminating. Finally, the SRS consisting of $fg \to gff$ and $fh \to hg$ suffices for the case $\alpha = \omega$. □

Proposition 3.42 (Zantema 1999). *The SRS $\{fg \to gff\}$ is weakly ω-terminating but not ω-terminating.*

Proof. We know from Proposition 3.41 that the SRS is not ω-terminating. Its weak ω-termination is established by putting $[f](n) := 1 + 2\lfloor n/2 \rfloor$ and $[g](n) := 2 + 2\lfloor n/2 \rfloor$. These functions are weakly monotone, have the weak subterm property, and satisfy $[f]([g](n)) = 3 + 2\lfloor n/2 \rfloor > 2 + 2\lfloor n/2 \rfloor = [g]([f]([f](n)))$. □

Though weak ω-termination is more general than ω-termination, it does not cover the whole of ω^2-termination.

Lemma 3.43. *There is a SRS which is ω^2-terminating but not weakly ω-terminating (and not $<\omega^2$-terminating).*

Proof. Consider the SRS over $\{f, g\}$ containing $\mathrm{Emb}(\Sigma)$ and, yet again, the rule $fg \to gff$. Since the two embedding rules force the interpreting functions to be monotone, weak α-termination and α-termination are equivalent in this case. The result follows from Proposition 3.41. □

Theorem 3.44 (Ferreira 1995, 5.28, 5.29).

i. *Let $\beta > 0$ and $\mathcal{R}$ be an α-terminating TRS over signature Σ. If $\Sigma = \Sigma^{(\leqslant 1)}$ or β is principal, then $\mathcal{R}$ is $\beta \cdot \alpha$-terminating.*

ii. *α-termination implies ω^α-termination.*

One main obstacle to proving α-termination (and hence total termination) of a given TRS is to exhibit a compatible Σ-algebra $(\alpha, \mathcal{F})$ which is *strictly* monotone, whereas naturally arising candidates are often *weakly* monotone. The following result of Touzet (1998b), which is of central importance for Theorem 5.51, one of the main results of this text, shows that weak α-termination implies total termination. Hence strict monotonicity can be dropped in favor of the combination of weak monotonicity and the weak subterm property (on a possibly larger ordinal).

Theorem 3.45 (Touzet 1998b). *Weak α-termination implies ω^α-termination. Hence weak total termination implies total termination.*

Proof. Let $\mathcal{R}$ be a TRS over signature Σ whose weak α-termination is witnessed by $(\alpha, \mathcal{F})$. By $\mathcal{P}$ we denote the partial order $(\mathsf{mul}(\alpha) \setminus \{\emptyset\}, <_{\mathsf{mul}})$. An application of Proposition 2.43.iii and Theorem 2.44 shows $\mathcal{P}$ is a well-order with order type ω^α (omitting the empty multiset does not hurt since ω^α is infinite). As the elements of $\mathcal{P}$ are nonempty multisets over α, each of them contains a unique maximal ordinal. Thus the following definition of algebra operations is legal. For $f \in \Sigma^{(n)}$ and members $a_1, \dots, a_n$ of $\mathcal{P}$, we put

$$[f]_{\mathcal{P}}(a_1, \dots, a_n) := \wr [f]_\alpha(\max a_1, \dots, \max a_n) \int \uplus a_1 \uplus \dots \uplus a_n \ .$$

Any $\beta \in a_i$ satisfies $\beta \leqslant \max a_i \leqslant [f]_\alpha(\max a_1, \dots, \max a_n)$ because of the weak subterm property of $[f]_\alpha$, hence

$$[f]_\alpha(\max a_1, \dots, \max a_n) = \max\, [f]_{\mathcal{P}}(a_1, \dots, a_n) \ . \tag{3.5}$$

Each $\rho\colon \mathcal{V} \to \mathcal{P}$ induces $\rho'\colon \mathcal{V} \to \alpha$ by $\rho'(x) := \max \rho(x)$. We intend to show

$$(\forall s \in \mathcal{T}(\Sigma, \mathcal{V}))(\max [\![s, \rho]\!]_{\mathcal{P}} = [\![s, \rho']\!]_\alpha) \ . \tag{3.6}$$

This is done by induction on s. For variables

$$\max [\![x, \rho]\!]_{\mathcal{P}} = \max \rho(x) = \rho'(x) = [\![x, \rho']\!]_\alpha$$

suffices. The remaining case is a bit more involved. Relying on (3.5) and the induction hypothesis we get

$$\begin{aligned}\max [\![f(s_1,\dots,s_n),\rho]\!]_{\mathcal{P}} &= \max\,[f]_{\mathcal{P}}([\![s_1,\rho]\!]_{\mathcal{P}},\dots,[\![s_n,\rho]\!]_{\mathcal{P}})\\ &= [f]_\alpha(\max [\![s_1,\rho]\!]_{\mathcal{P}},\dots,\max [\![s_n,\rho]\!]_{\mathcal{P}})\\ &= [f]_\alpha([\![s_1,\rho']\!]_\alpha,\dots,[\![s_n,\rho']\!]_\alpha)\\ &= [\![f(s_1,\dots,s_n),\rho']\!]_\alpha\ .\end{aligned}$$

Now we are prepared to show $\mathcal{R}$ is compatible with our new Σ-algebra. Consider ρ as above and a rule (l,r) in $\mathcal{R}$. As $\mathcal{R}$ is compatible with $(\alpha,\mathcal{F})$, we know $[\![l,\rho']\!]_\alpha > [\![r,\rho']\!]_\alpha$. Using (3.6) twice we arrive at

$$\max [\![r,\rho]\!]_{\mathcal{P}} = [\![r,\rho']\!]_\alpha < [\![l,\rho']\!]_\alpha \in [\![l,\rho]\!]_{\mathcal{P}}\ ,$$

and now a look at Lemma 2.42 yields $[\![r,\rho]\!]_{\mathcal{P}} <_{\mathsf{mul}} [\![l,\rho]\!]_{\mathcal{P}}$. As ρ was arbitrary, we see $\mathcal{R}$ is compatible with our new Σ-algebra. It remains to show that the $[f]_{\mathcal{P}}$ with $f \in \Sigma^{(\geqslant 1)}$ are monotone. Let $a_1,\dots,a_n,b$ be members of $\mathcal{P}$ such that there is an $i \in [1,n]$ satisfying $a_i >_{\mathsf{mul}} b$. This implies $\max a_i \geqslant \max b$, and hence, by weak monotonicity of $[f]_\alpha$,

$$\begin{aligned}\gamma &:= [f]_\alpha(\max a_i,\dots,\max a_i,\dots,\max a_n)\\ &\geqslant [f]_\alpha(\max a_i,\dots,\max b,\dots,\max a_n) =: \beta\end{aligned}$$

holds. Finally we arrive at

$$\begin{aligned}[f]_{\mathcal{P}}(a_1,\dots,a_n) &= \ \wr\gamma\int \uplus a_1 \uplus \dots \uplus a_i \uplus \dots \uplus a_n\\ &\geqslant_{\mathsf{mul}} \wr\beta\int \uplus a_1 \uplus \dots \uplus a_i \uplus \dots \uplus a_n\\ &>_{\mathsf{mul}} \wr\beta\int \uplus a_1 \uplus \dots \uplus b \uplus \dots \uplus a_n\ ,\end{aligned}$$

using Definition 2.41. □

Touzet's original version of the Theorem states that total termination of $\mathcal{R}$ follows from the existence of a weakly monotone interpretation of $\mathcal{T}(\Sigma)$ in a well-order $(P,\prec)$ such that $\mathcal{I}$ has the subterm property and normalizes $\mathcal{R}$. The extension to Σ-algebras is due to Zantema (1999, 2001). The latter text contains a short alternative proof of the second half of the Theorem.

Even if we know a certain TRS over Σ is totally terminating, this does not necessarily lead to an appropriate well-order of $\mathcal{T}(\Sigma)$, because two distinct terms may have the same interpretation. Nevertheless there do exist satisfactory well-orders on $\mathcal{T}(\Sigma)$.

Theorem 3.46 (Ferreira 1995, 5.40). *A TRS $\mathcal{R}$ is totally terminating if and only if there exists a well-order $(\mathcal{T}(\Sigma),\prec)$ which is closed under contexts and normalizes $\mathcal{R}$.*

We learnt that ω-termination, (weak) total termination, simple termination, and termination constitute a proper hierarchy within the class of TRSs. There are also proper hierarchies within ω-termination.

Proposition 3.47 (Zantema 1994). *The SRS $\{fgh \to gfhg\}$ is ω-terminating but not polynomially terminating.*

In Section 5.1 we will encounter further separation results and see that $\mathcal{E}_n$-termination, $\mathcal{E}_{n+1}$-termination, primitive recursive termination, $\mathcal{M}_k$-termination (for $k \geqslant 2$), $\mathcal{M}_{k+1}$-termination, and multiple termination constitute a properly increasing hierarchy within ω-termination.

3.4 Syntactic Orders on Terms

An important group of termination proof methods are the syntactic orders on terms. Before we present their most prominent examples, we introduce some nomenclature which will later be useful.

Definition 3.48. Let $\mathcal{M}$ be a collection of reduction orders.

- By $\mathcal{M}(1)$ we denote the collection of those members of $\mathcal{M}$ which live on a signature $\Sigma = \Sigma^{(\leqslant 1)}$.
- A TRS is said to *terminate via* $\mathcal{M}$ if it is compatible with some $\prec \in \mathcal{M}$.

By Lemma 3.11.v, a TRS $\mathcal{R}$ is compatible with a reduction order $\prec$ if and only if $\mathcal{R} \subseteq \succ$ holds. Lemma 3.11.iv shows termination via $\mathcal{M}$ implies termination.

Definition 3.49. Let Σ be a signature.

- A partial order on Σ is called a *precedence.*
- Consider a family of partial orders $(\mathcal{T}(\Sigma, \mathcal{V}), \lhd_{\Sigma,\prec})$ in the (displayed) parameters Σ and $\prec$, the latter being a precedence on Σ. This family is called *incremental with respect to signatures* if, for any signature $\Sigma' \supseteq \Sigma$, $\lhd_{\Sigma',\prec}$ is an extension of $\lhd_{\Sigma,\prec}$ (with base $\mathcal{T}(\Sigma', \mathcal{V})$), and it is called *incremental with respect to precedences* if $\prec' \supseteq \prec$ (both being precedences on Σ) implies $\lhd_{\Sigma,\prec'}$ is an extension of $\lhd_{\Sigma,\prec}$ (with base $\mathcal{T}(\Sigma, \mathcal{V})$).

3.4.1 Multiset Path Orders

Plaisted (1978) introduced a syntactic order on terms which roughly consists of comparing terms by first comparing their root symbols according to a given precedence, and, in case of equality, by recursively comparing the multisets of their immediate subterms. As these multisets ignore the positions of the subterms, this order is not sensitive to permutations of subterms.

Definition 3.50. We define a binary relation $\sim$ on $\mathcal{T}(\Sigma, \mathcal{V})$ by $s \sim t$ if $s = f(s_1, \ldots, s_n)$, $t = f(t_1, \ldots, t_n)$, and there is a permutation π of $[1, n]$ such that $(\forall i \in [1, n])(s_i \sim t_{\pi(i)})$. The relation is called *permutative equivalence.*

Definition 3.51 (Plaisted 1978, Dershowitz 1982). Let Σ be a signature equipped with a precedence $\succ$. The *multiset path order* (MPO) $\succ_{\mathsf{mpo}}$ (based on Σ and $\succ$) of $\mathcal{T}(\Sigma, \mathcal{V})$ is defined by: $s \succ_{\mathsf{mpo}} t$ if $s = f(s_1, \ldots, s_n)$ and

i. $(\exists i \in [1, n])(s_i \succ_{\mathsf{mpo}} t \vee s_i \sim t)$, or
ii. $t = g(t_1, \ldots, t_m)$, $f \succ g$, and $(\forall i \in [1, m])(s \succ_{\mathsf{mpo}} t_i)$, or
iii. $t = f(t_1, \ldots, t_n)$ and $\wr s_1, \ldots, s_n \int \succ_{\mathsf{mpo}}^{\mathsf{mul}} \wr t_1, \ldots, t_n \int$.

Theorem 3.52.

i. *MPOs are incremental with respect to both signature and precedence.*
ii. *Each MPO is a simplification order.*
iii. *Termination via MPO is decidable.*
iv. *Termination via MPO implies total termination.*

Proof. While (i) can easily be read off the definition, (ii) is due to Dershowitz (1982). As there are only finitely many precedences on Σ, decidability of termination via MPO is immediate. Ferreira (1995, 4.40) established (iv) (in a much more general setting). □

Canonical TRSs terminating via MPO are those computing primitive recursive functions, see Theorem 6.4. Theorem 5.6 will tell us that these are essentially the most complex examples MPO can cope with.

3.4.2 Lexicographic Path Orders

The difference between lexicographic path orders and MPOs is a lexicographic comparison of the subterms instead of using the multiset order. As we will see later, this results in a considerable gain of complexity. Because we compare lexicographically, we do not need permutative equivalence.

Definition 3.53 (Kamin and Lévy 1980). Let Σ be a signature equipped with a precedence $\succ$. The *lexicographic path order* (LPO) $\succ_{\mathsf{lpo}}$ (based on Σ and $\succ$) of $\mathcal{T}(\Sigma, \mathcal{V})$ is defined by: $s \succ_{\mathsf{lpo}} t$ if $s = f(s_1, \ldots, s_n)$ and

i. $(\exists i \in [1, n])(s_i \succeq_{\mathsf{lpo}} t)$, or
ii. $t = g(t_1, \ldots, t_m)$, $f \succ g$, and $(\forall i \in [1, m])(s \succ_{\mathsf{lpo}} t_i)$, or
iii. $t = f(t_1, \ldots, t_n)$, $(s_1, \ldots, s_n) \succ_{\mathsf{lpo}}^{\mathsf{lex}} (t_1, \ldots, t_n)$, and $(\forall i \in [1, n])(s \succ_{\mathsf{lpo}} t_i)$.

For the proof of the following result we can proceed literally as in the proof of Theorem 3.52, though of course the real proofs differ.

Theorem 3.54.

i. *LPOs are incremental with respect to both signature and precedence.*

ii. *Each LPO is a simplification order.*

iii. *Termination via LPO is decidable.*

iv. *Termination via LPO implies total termination.*

If there are at most unary function symbols, multiset and lexicographic comparison of the direct subsets are identical, and permutative equivalence boils down to equality.

Lemma 3.55. *If* $\Sigma = \Sigma^{(\leqslant 1)}$ *and* $\prec$ *is a precedence on* Σ*, then* $\prec_{\mathsf{mpo}}$ *and* $\prec_{\mathsf{lpo}}$ *coincide. Hence MPO(1) and LPO(1) are identical, and termination via MPO(1) is equivalent to termination via LPO(1).*

Canonical examples of complex TRSs terminating via LPO are those computing the k-ary Ackermann functions of Definition 2.65, see also Theorem 6.6. Thus LPO allows for much longer derivations than MPO, but even termination via LPO does not comprise PT, as the SRS $\{ff \to g, g \to f\}$, which is taken from Middeldorp and Zantema (1997, p. 144), shows. It is polynomially terminating (with $[\![f]\!](n) := n+2$ and $[\![g]\!](n) := n+3$), yet neither LPO nor MPO can show its termination as we cannot have both $f \succ g$ and $g \succ f$.

MPO is sometimes called *recursive path order* (RPO) or *RPO with multiset status*, while LPO is referred to as *RPO with lexicographic status*. The latter names indicate it is possible to combine MPO and LPO in a generalized order, *RPO with status*. This was introduced by Kamin and Lévy (1980).

3.4.3 Knuth–Bendix Orders

We are going to define Knuth–Bendix orders, which were introduced by Knuth and Bendix (1970) in a slightly restricted form. Our definition follows Baader and Nipkow (1998). The signature is equipped with a precedence and a *weight function*, which associates a nonnegative real number with each symbol. We extend this function to a weight on terms by adding the weights of the symbols. Very roughly speaking, terms are compared by first comparing their weights, then their root symbols, and finally, by recursion, their subterms. Under certain additional assumptions this results in a simplification order.

In the original definition weights had to be natural numbers. We will soon see how the extension to reals results in new orders which are not equivalent to any old order. However, concerning termination of TRSs nothing is gained. According to a recent result of Korovin and Voronkov (2001), termination via a new order implies termination via some old order.

Definition 3.56. Let Σ be a signature equipped with a precedence $\succ$. We call $\mu\colon \Sigma \cup \mathcal{V} \to \mathbb{R}_0^+$ *compatible* if there is $\gamma \in \mathbb{R}^+$ such that the following three conditions are met:

- For all $e \in \Sigma^{(0)}$ we have $\mu(e) \geqslant \gamma$.
- For all $x \in \mathcal{V}$ we have $\mu(x) = \gamma$.
- If $f \in \Sigma^{(1)}$ and $\mu(f) = 0$, then $f \succ g$ holds for all $g \in \Sigma \setminus \{f\}$.

A (necessarily unique) unary symbol f satisfying $\mu(f) = 0$ is called *special*, and we will soon see it indeed deserves special treatment. We will stick to the convention that, if there is a special symbol in Σ, then it is the symbol i. By abuse of notation, a compatible μ generates a *weight function* $\mu\colon \mathcal{T}(\Sigma, \mathcal{V}) \to \mathbb{R}^+$ via

$$\mu(g(s_1, \ldots, s_n)) := \mu(g) + \sum_{k=1}^{n} \mu(s_k) \ .$$

An important property of the weight function is

$$\mu(s) = \gamma \cdot \sum_{x \in \mathcal{V}} |s|_x + \sum_{g \in \Sigma} \mu(g) \cdot |s|_g \ . \tag{3.7}$$

More sophisticated weight functions have been considered. Lankford (1979) introduced polynomials with natural number coefficients, and a very general approach is from Dershowitz (1987b). Although some of the following results can be extended to these weight functions, we will not treat them in any detail.

Definition 3.57. The *Knuth–Bendix order* (KBO) $\succ_{\mathsf{kbo}}$ (based on Σ, $\succ$, and compatible μ) of $\mathcal{T}(\Sigma, \mathcal{V})$ is defined by: $s \succ_{\mathsf{kbo}} t$ if $(\forall x \in \mathcal{V})(|s|_x \geqslant |t|_x)$ and

i. $\mu(s) > \mu(t)$ or
ii. $\mu(s) = \mu(t)$ and
   a) $s = i^a(x)$ with special i, $t = x$ for some $a > 0$, and $x \in \mathcal{V}$, or
   b) $s = f(s_1, \ldots, s_n)$, $t = g(t_1, \ldots, t_m)$, and $f \succ g$, or
   c) $s = f(s_1, \ldots, s_n)$, $t = f(t_1, \ldots, t_n)$, and $(s_1, \ldots, s_n) \succ_{\mathsf{kbo}}^{\mathsf{lex}} (t_1, \ldots, t_n)$.

Theorem 3.58.

i. *KBOs are incremental with respect to both signature and precedence (under the assumption of compatibility).*
ii. *If $\prec$ is total on Σ, then $(\mathcal{T}(\Sigma), \prec_{\mathsf{kbo}})$ is total.*
iii. *Each KBO is a simplification order.*
iv. *Termination via KBO implies total termination.*
v. *If a TRS terminates via KBO, then it is nonduplicating.*

Proof. The proof of (iii) is lengthy and quite involved. An ambitious reader may consult Baader and Nipkow (1998), whose presentation is based on work of Dick et al. (1990). All remaining items are (more or less) immediate consequences of the definition. □

It should be stressed that closure under substitutions is lost as soon as one allowed for a constant c with $\mu(x) > \mu(c)$, since this implied $x \succ_{\mathsf{kbo}} c$ and, substituting c for x, $c \succ_{\mathsf{kbo}} c$. Similarly, dropping the condition that a unary f with weight 0 has to be $\prec$-maximal is lethal. Suppose there is some g of arity n satisfying $g \succ f$. Put $s := g(c, \ldots, c)$ with c a constant. The $s_n := f^n(s)$ provide an infinite $\prec_{\mathsf{kbo}}$-descending sequence.

Knuth and Bendix introduced KBO in the context of group theory, and it is indeed particularly well suited for TRSs occuring there. Consider, over the signature containing e, $-$, and $+$ of arities 0, 1, and 2, the TRS containing

$$x + e \to x\ , \quad x + (-x) \to e\ , \text{and} \quad -(-x) \to x\ .$$

It terminates via any KBO with special $-$ and $\mu(e) = \mu(x)$.

Definition 3.59. A KBO based on Σ, $\succ$, and μ is called

- an *NKBO* if μ takes only values in $\mathbb{N}$, and it is
- a *KBO*$^-$ if there is no special symbol.

Any KBO whose weight takes only values in the rational numbers is an NKBO, because for $\alpha \in \mathbb{R}^+$ the KBO based on weight μ and the KBO based on weight $\alpha \cdot \mu$ coincide.

Proposition 3.60. *There are KBOs which are not equivalent to any NKBO.*

Proof. Let Σ contain the constant c and the unary symbols f and g. For the KBO $\succ_{\mathsf{kbo}}$ based on the empty precedence, $\mu(c) := \mu(f) := 1$, and

$$\mu(g) := 1.0100100010000100000010\ldots01\underbrace{0\ldots0}_{n}1\underbrace{0\ldots0}_{n+1}10\ldots$$

we get $f^2(c) \succ_{\mathsf{kbo}} g^1(c) \succ_{\mathsf{kbo}} f^1(c)$, $f^{11}(c) \succ_{\mathsf{kbo}} g^{10}(c) \succ_{\mathsf{kbo}} f^{10}(c)$, $f^{102}(c) \succ_{\mathsf{kbo}} g^{100}(c) \succ_{\mathsf{kbo}} f^{101}(c)$, $f^{1011}(c) \succ_{\mathsf{kbo}} g^{1000}(c) \succ_{\mathsf{kbo}} f^{1010}(c)$, and so on. Note that we have $\inf \{\mu(s) - \mu(t) > 0 : s, t \in \mathcal{T}(\Sigma, \mathcal{V})\} = 0$. No NKBO is able to make such fine distinctions. □

So there are more KBOs than NKBOs. Does this have an effect on the amount of TRSs whose termination can be shown by KBO? The finite character of the rewrite relation, which is due to the fact that (for us) a TRS is finite, prevents us from making use of the full power of real-valued KBO.

Theorem 3.61 (Korovin and Voronkov 2001). *A TRS terminates via KBO if and only if it terminates via NKBO.*

In contrast to both MPO and LPO, decidability of termination via KBO is not immediate, because one has to choose from infinitely many weight functions. Using systems of homogeneous linear inequalities, this choice process can be made effective.

Theorem 3.62 (Dick et al. 1990, Korovin and Voronkov 2001). *It is decidable (in polynomial time) whether a TRS terminates via KBO.*

Proof. Decidability was shown by Dick et al. (1990), and Korovin and Voronkov (2001) provided the timebound (in the more general setting of termination via closed KBO, see Section 5.5). □

From now on, in connection with KBO signatures will be tacitly equipped with precedence and (compatible) weight.

Termination via KBO is incomparable with both PT and termination via LPO. The SRS $\{fg \to ggf\}$ (from Ferreira (1995, p. 114), see also Middeldorp and Zantema (1997, p, 148)) is not terminating via KBO as weight considerations imply g has to be special, whereas precedence considerations demand $f \succ g$. In contrast to this, we get both termination via LPO (using $f \succ g$) and PT (by putting $[\![f]\!](n) := 3n$ and $[\![g]\!](n) := n + 1$). The following result prepares complementary results in favor of KBO.

Lemma 3.63. *There is an SRS terminating via KBO which is ω^2-terminating but not $<\omega^2$-terminating and not weakly ω-terminating.*

Proof. In Lemma 3.43 we observed that the SRS containing $fg \to gff$ and $\mathrm{Emb}(\{f, g\})$ is ω^2-terminating but neither $<\omega^2$-terminating nor weakly ω-terminating. Making f special, we easily establish termination via KBO. □

Thus termination via KBO does not imply PT. We will later (in Theorems 5.6 and 5.7) see that termination via either MPO or LPO does imply ω-termination. Hence, in contrast to KBO, neither MPO nor LPO are able to cope with the rule $fg \to gff$. An essentially optimal connection between termination via KBO and α-termination will be established in Theorem 5.17.

The rôle of the special symbol i in the above standard definition of KBO is not that obvious. A KBO on $\mathcal{T}(\Sigma, \mathcal{V})$ with $i \in \Sigma$ can be considered as an order on the *labeled* members of $\mathcal{T}(\Sigma \setminus \{i\}, \mathcal{V})$. A term is labeled by adjoining natural numbers to each of its symbols (including the variables). Two labeled terms are compared by lexicographically comparing the weights, the labels of the root symbols, the root symbols, and, recursively, the direct subterms. We want to be more explicit here, because we believe this is an important observation.

There is good reason to refrain from formally introducing labeled terms, since for them substitutions are a bit nonstandard (the labels of the variable and of the root symbol of the substituted term have to be added ...) Instead, we give a reformulation of "s' is labeled with a".

Definition 3.64. For $s, s' \in \mathcal{T}(\Sigma, \mathcal{V})$, $a \in \mathbb{N}$, and the special symbol i we put

$$s \equiv i^a s' \;:\Longleftrightarrow\; s = i^a s' \wedge \mathrm{root}(s') \neq i$$

and say that s is s' *labeled with* a.

This will now be used for an equivalent reformulation of KBO. We define $(\mathcal{T}(\Sigma, \mathcal{V}), \prec_{\mathsf{kbo2}})$ like $(\mathcal{T}(\Sigma, \mathcal{V}), \prec_{\mathsf{kbo}})$ in Definition 3.57, but replace (ii) with

(ii') $\mu(s) = \mu(t)$, $s \equiv i^a s'$, and $t \equiv i^b t'$ with special i, and

- ❖ $a > b$ or
- ❖ $a = b$, $s' = f(s_1, \dots, s_n)$, $t' = g(t_1, \dots, t_m)$, and
  - a) $f \succ g$, or
  - b) $f = g$ and $(s_1, \dots, s_n) \succ^{\mathsf{lex}}_{\mathsf{kbo2}} (t_1, \dots, t_n)$.

Note that the formulation is valid even if Σ contains no special symbol, since then $u \equiv i^0 u$ holds for all terms u.

Lemma 3.65. *The orders* $(\mathcal{T}(\Sigma, \mathcal{V}), \prec_{\mathsf{kbo}})$ *and* $(\mathcal{T}(\Sigma, \mathcal{V}), \prec_{\mathsf{kbo2}})$ *coincide.*

Proof. First of all, if $i \in \Sigma$, then for $\succ_{\mathsf{kb}}$ equal to either $\succ_{\mathsf{kbo}}$ or $\succ_{\mathsf{kbo2}}$,

$$(\forall s, t \in \mathcal{T}(\Sigma, \mathcal{V}))\big(s \succ_{\mathsf{kb}} t \Leftrightarrow i(s) \succ_{\mathsf{kb}} i(t)\big) \tag{3.8}$$

holds. The equivalence of $s \succ_{\mathsf{kbo}} t$ and $s \succ_{\mathsf{kbo2}} t$ is now shown by induction on $\mathrm{dp}(s) + \mathrm{dp}(t)$. In the interesting case we have $\mu(s) = \mu(t)$. It is now easy to see that $s \succ_{\mathsf{kbo}} t$ implies $s \succ_{\mathsf{kbo2}} t$, partly relying on (3.8). On the other hand, $s \succ_{\mathsf{kbo2}} t$ yields $s \equiv i^a s'$, $t \equiv i^b t'$ and $a \geqslant b$. If $a = 0$ or $b > 0$, then (3.8) and the induction hypothesis imply $s \succ_{\mathsf{kbo}} t$. So it remains to treat the case $a > b = 0$. If $t' = x \in \mathcal{V}$, then, due to the variable condition $|s'|_x \geqslant |t'|_x$ and weight considerations, we get $s' = x$ and may conclude $s \succ_{\mathsf{kbo}} t$. Otherwise, we have $\mathrm{root}(s) = i \succ \mathrm{root}(t)$. This too implies $s \succ_{\mathsf{kbo}} t$. □

We will tacitly use $\succ_{\mathsf{kbo2}}$ whenever comparing terms by KBO.

In order to approach KBOs with number-theoretic functions we have to replace the $\mathbb{R}^+$-valued weight function with an equivalent function into $\mathbb{N}^+$. This takes some effort.

Proposition 3.66. *For each weight μ there is $\nu\colon \mathcal{T}(\Sigma,\mathcal{V}) \to \mathbb{N}^+$ satisfying*

$$(\forall s,t \in \mathcal{T}(\Sigma,\mathcal{V}))(\mu(s) > \mu(t) \iff \nu(s) > \nu(t)) \ .$$

We can choose ν such that there is an $a \in \mathbb{N}$ with $(\forall s \in \mathcal{T}(\Sigma,\mathcal{V}))(\nu(s) < a^{\mathrm{dp}(s)})$.

Proof. Note that we may put $\nu := \mu$ if μ takes only values in $\mathbb{N}$. In this case, for any $a > \mathrm{Ar}(\Sigma)$ satisfying $(\forall f \in \Sigma)(a > \mu(f))$, we can show $\nu(s) < a^{\mathrm{dp}(s)}$ by induction on $s \in \mathcal{T}(\Sigma,\mathcal{V})$.

In the general case things are more involved. Our aim is to show

$$\nu(s) := \mathrm{card}(\{\mu(t) : \mu(t) < \mu(s)\}) + 1$$

is well-defined and has the desired properties. We may assume that the nonspecial symbols of Σ are $f_1,\ldots,f_k$. Each $t \in \mathcal{T}(\Sigma,\mathcal{V})$ is associated with a tuple $p(t)$ of length $k+1$ via

$$p(t) := (|t|_{f_1},\ldots,|t|_{f_k},\sum_{x\in\mathcal{V}}|t|_x) \ .$$

From (3.7) we can infer that $p(t) = p(t')$ implies $\mu(t) = \mu(t')$. This observation can be used to impose a limit on the weights of terms of bounded size. For arbitrary m we introduce $T_m := \{\mu(t) : |t| < m\}$ and get

$$\mathrm{card}(T_m) \leqslant \mathrm{card}(\{p(t) : |t| < m\}) \leqslant \mathrm{card}(\{(a_0,\ldots,a_k) : \bar{a} < m\}) \leqslant m^{k+1} \ .$$

In a next step we put

$$W := \{\mu(u) > 0 : u \in \Sigma \cup \mathcal{V}\}, \quad w := \min W, \quad \text{and} \quad w' := \max W \ .$$

Let us fix some $s \in \mathcal{T}(\Sigma,\mathcal{V})$. A close look at (3.1) and (3.7) shows

$$\mu(s) \leqslant w' \cdot |s| \leqslant w' \cdot b^{\mathrm{dp}(s)} \tag{3.9}$$

for $b := \mathrm{Ar}(\Sigma) + 1$. Assume we are given $t \in \mathcal{T}(\Sigma,\mathcal{V})$ satisfying $\mu(t) < \mu(s)$. As we are only interested in the weight of t, we may safely assume t contains no special symbol. Hence t consists of elements of $X := \mathcal{V} \cup (\Sigma^{(\leqslant 1)} \setminus \{i\})$ and symbols from $\Sigma^{(\geqslant 2)}$. Due to the weight conditions for KBOs, any occurrence of an element of X in t contributes at least w to $\mu(t)$. By Lemma 3.1, symbols from $\Sigma^{(\geqslant 2)}$ have less occurrences in t than elements of X, so we get $\frac{1}{2}w|t| \leqslant \mu(t)$ using (3.7). Hence (3.9) implies

$$|t| \leqslant 2w^{-1}\mu(t) < 2w^{-1}\mu(s) \leqslant 2w^{-1}w'b^{\mathrm{dp}(s)} \ ,$$

and thus $|t| < c^{\mathrm{dp}(s)}$ holds for any $c \geqslant 2w^{-1}w'b$. This yields

$$\mathrm{card}(\{\mu(t) : \mu(t) < \mu(s)\}) \leqslant \mathrm{card}(T_{c^{\mathrm{dp}(s)}}) \leqslant \left(c^{\mathrm{dp}(s)}\right)^{k+1} = \left(c^{k+1}\right)^{\mathrm{dp}(s)} \ ,$$

so ν is well-defined and owns the announced growth bound.

Obviously $\mu(s) > \mu(t)$ holds if and only if $\nu(s) > \nu(t)$. □

3.5 Functions Computable by a TRS

Because the powerful concept of "term" is already present, TRSs provide a strong and natural model of computation.

Definition 3.67. Let $f\colon \mathbb{N}^l \to \mathbb{N}$ be given.

- We say f is *computable* by the terminating TRS $\mathcal{R}$ if there are $F \in \Sigma^{(l)}$, $S, P \in \Sigma^{(1)}$ (not necessarily distinct), and $0, 0' \in \Sigma^{(0)}$ (also not necessarily distinct) such that, for all natural numbers $n_1, \ldots, n_l$, the unique normal form of $F(S^{n_1}0, \ldots, S^{n_l}0)$ is $P^{f(n_1,\ldots,n_l)}0'$. The symbol S is called the *input successor*, while P is the *output successor*.
- If $\mathcal{M}$ is a termination proof method, then we say that f is *computable via* $\mathcal{M}$ if f is computable by a TRS terminating via $\mathcal{M}$.
- By $\textsc{Comp}(\mathcal{M})$ we denote the set of functions computable via $\mathcal{M}$, while $\textsc{Comp}_1(\mathcal{M})$ collects the functions computable via $\mathcal{M}$ using only one successor symbol.

The input successor and the output successor do not have to be distinct. However, Bonfante et al. (1999) observed that using only one successor symbol may sometimes impose restrictions on the amount of functions computable via $\mathcal{M}$ – see Sections 6.1 and 6.3.

Though in the above definition we demand confluence on the input terms, the rewriting process is intrinsically nondeterministic. In Definition 6.22 the notion of computability by a TRS will be extended to the nonconfluent case.

The concept of computation by a TRS is quite different from computation by either a TM or a RM. There is a k depending on the TRS such that in one rewrite step $s \to t$ the size of t may get k times larger than the size of s, whereas for both RM and TM one step changes just one symbol. Forthermore applicability of a rewrite rule is based on a rather complex definition. The following Theorem shows that computation by a TRS can be simulated by both TM and RM with the usual overhead.

Theorem 3.68. *Let $\mathcal{M} \supseteq \textsc{Elem}$ be a set of number-theoretic functions which accommodates exponentiation. If f is computable by a TRS $\mathcal{R}$ such that $\mathrm{dl}_{\mathcal{R}}$ is eventually dominated by a member of $\mathcal{M}$, then f is computable by both TM and RM with timebound in $\mathcal{M}$.*

Proof. Handley and Wainer (1994, 7.11) investigate the simulation of a TRS by a nondeterministic multi-tape TM with special focus on the time complexity and show that the time taken by the TM to simulate a derivation is exponential in the length of the derivation. As it is possible to turn a nondeterministic multi-tape TM with timebound in $\mathcal{M}$ into a deterministic one-tape TM with timebound in $\mathcal{M}$ (see Handley and Wainer (1994, 7.6 and 7.7)), we are through with the TMs. The result for RMs follows from Lemma 2.92. □

We already mentioned (somewhere around Theorem 3.8) that Huet and Lankford (1978) showed it is possible to transform a TM into a TRS, under preservation of termination. Hence we can flank the above result by the following.

Corollary 3.69. *The recursive functions are the functions computable by TRSs.*

Under a certain assumption, we can directly employ the derivation length function of $\mathcal{R}$ to construct a bound on the computed function.

Lemma 3.70. *If $f\colon \mathbb{N}^l \to \mathbb{N}$ is computable by a nonduplicating TRS $\mathcal{R}$ (with symbols as above), then there are $a \in \{0,1\}$ and $b \in \mathbb{N}$ satisfying*

$$f(n_1,\dots,n_l) \leqslant 1 + a\cdot(n_1+\cdots+n_l) + b\cdot \mathrm{dl}_{\mathcal{R}}(F(S^{n_1}0,\dots,S^{n_l}0))$$

for all $n_1,\dots,n_l$. The use of two distinct successor symbols implies $a = 0$.

Proof. Let S and P be the input resp. output successor. We put

$$b := \max\{|r|_P \dot{-} |l|_P : (l,r) \in \mathcal{R}\}.$$

Since $\mathcal{R}$ is nonduplicating, $s \to_{\mathcal{R}} t$ implies $|t|_P \dot{-} |s|_P \leqslant b$. Hence for

$$s_{\text{in}} := F(S^{n_1}0,\dots,S^{n_l}0) \quad \text{and} \quad s_{\text{out}} := P^{f(n_1,\dots,n_l)}0'$$

we get

$$f(n_1,\dots,n_l) = |s_{\text{out}}|_P \leqslant |s_{\text{in}}|_P + b\cdot \mathrm{dl}_{\mathcal{R}}(s_{\text{in}})\ .$$

If S and P differ, then F is the only possible occurrence of P in s_{in}. □

A similar result holds for arbitrary $\mathcal{R}$, however multiplication (with some b) has to be replaced with exponentiation.

4 Order Types

You are on the right way.
Go on!

We have been told by Lemma 3.9 that, for any terminating TRS $\mathcal{R}$, the order type of $(\mathcal{T}(\Sigma), \overset{+}{\leftarrow}_{\mathcal{R}})$ cannot exceed ω. In contrast to this, the order types of simplification orders which are compatible with $\mathcal{R}$ may get quite large. According to Schmidt (1979) their supremum is the small Veblen number $\vartheta(\Omega^\omega)$. By a result of de Jongh and Parikh (1977), the supremum of the order types of simplification orders over monadic signatures is the much smaller ordinal ω_3. The maximal order types of simplification orders over both terms and strings are already attained by prominent representatives – by LPOs.

In general, the larger the order type of a simplification order the more TRSs are compatible with it, hence the more TRSs can be shown to terminate via this order. Thus the strength of a termination proof method consisting of a collection of simplification orders (like one of the syntactic orders on terms we encountered in Section 3.4) can be measured by figuring out the supremum of the occuring order types. The larger the occuring ordinals are the longer derivations may be expected. This measure has to be taken with a grain of salt as, due to its finitary character, a TRS may not be able to exhaust the full power of the termination proof method because distinct branches of the compatible order are too far apart for the locally operating rewrite relation. Later (in Chapter 5 and Chapter 6) we will see that, concerning complexities, MPO is weaker than KBO, although the order types occuring within MPO are much larger than ω^ω, which we will prove to be the maximal order type of a KBO. This order type is already attainable by a KBO over a monadic signature, provided there is a special symbol. If a KBO does not contain a special symbol, then its order type is just ω.

So, as a rule of thumb, it is justified to regard the occuring order types as an indicator for the strength of a termination proof method, but this tool has to be used with care.

4.1 Order Types of MPOs and LPOs

While the order types of MPOs are already rather large, the order types of LPOs are enormous – they exhaust the small Veblen number $\vartheta(\Omega^\omega)$.

Theorem 4.1 (Dershowitz and Okada 1988). *Let Σ be a signature equipped with a precedence $\prec$.*

i. *For the induced MPO $\prec_{\mathsf{mpo}}$ we have*

$$\mathrm{otype}(\mathcal{T}(\Sigma), \prec_{\mathsf{mpo}}) \leqslant \vartheta(\Omega \cdot \mathrm{card}(\Sigma)) < \vartheta(\Omega \cdot \omega) \ .$$

ii. *For the induced LPO $\prec_{\mathsf{lpo}}$ we have*

$$\mathrm{otype}(\mathcal{T}(\Sigma), \prec_{\mathsf{lpo}}) \leqslant \vartheta(\Omega^{\mathrm{Ar}(\Sigma)} \cdot \mathrm{card}(\Sigma)) < \vartheta(\Omega^\omega) \ .$$

Proof. Our presentation follows Weiermann (1992).

Associate to each symbol f a natural number $a_f < \mathrm{card}(\Sigma)$ such that $f \prec g$ implies $a_f < a_g$ (the order type of f with respect to $\prec$ will do). For MPO we define an embedding $o\colon (\mathcal{T}(\Sigma), \prec_{\mathsf{mpo}}) \to \vartheta(\Omega \cdot \mathrm{card}(\Sigma))$ via

$$o(f(s_1, \ldots, s_n)) := \vartheta(\Omega \cdot a_f + o(s_1) \oplus \cdots \oplus o(s_n))$$

where $\oplus$ is the natural sum of ordinals from Definition 2.31. An invocation of Lemma 2.49 proves the claim.

The proof for LPO is similar. Here the the embedding o of $(\mathcal{T}(\Sigma), \prec_{\mathsf{lpo}})$ into $\vartheta(\Omega^{\mathrm{Ar}(\Sigma)} \cdot \mathrm{card}(\Sigma))$ is defined by

$$o(f(s_1, \ldots, s_n)) := \vartheta(\Omega^{\mathrm{Ar}(\Sigma)} \cdot a_f + \Omega^{n-1} \cdot o(s_1) + \cdots + \Omega^0 \cdot o(s_n)) \ ,$$

and again Lemma 2.49 does the job. □

It is a folklore result (cf. Hasegawa (1994, p. 161)) that, on the other hand, $\vartheta(\Omega \cdot \omega)$ and $\vartheta(\Omega^\omega)$ are the respective suprema of the order types of MPOs and LPOs.

If we restrict ourselves to monadic signatures, much lower bounds on the order types occur. Recall from Lemma 3.55 that, for monadic signatures, MPO and LPO coincide.

Theorem 4.2 (Sakai 1984, see also Hasegawa 1994). *Let Σ be a monadic signature which contains exactly n unary symbols and is equipped with a precedence $\prec$. For both the induced MPO $\prec_{\mathsf{mpo}}$ and the induced LPO $\prec_{\mathsf{lpo}}$ we have*

$$\mathrm{otype}(\mathcal{T}(\Sigma), \prec_{\mathsf{mpo}}) = \mathrm{otype}(\mathcal{T}(\Sigma), \prec_{\mathsf{lpo}}) \leqslant \omega^{\omega^{n-1}} \ .$$

If $\prec$ is total, then equality holds.

Theorem 4.9 will show this result already presents the maximal order types for simplification orders over monadic signatures.

4.2 Order Types of KBOs

In this section we develop a complete classification of the order types of KBOs. For various reasons, the literature concerned with order types of reduction orders focuses on the order types of the orders restricted to closed terms. We will join in, but since our classification is painlessly extendable to the unrestricted orders, we will present this result as well as a digression to infinite signatures. First of all, we exploit the literature.

Theorem 4.3. *Let $\prec_{\mathsf{kbo}}$ be an NKBO over the signature Σ.*

i. *If $\prec^{\Gamma}_{\mathsf{kbo}}$ is the NKBO over the signature Γ containing $i \succ \circ \succ e$ of weights* 0*,* 1*,* 1 *and arities* 1, 2, 0*, then we can embed $(\mathcal{T}(\Sigma), \prec_{\mathsf{kbo}})$ into $(\mathcal{T}(\Gamma), \prec^{\Gamma}_{\mathsf{kbo}})$.*
ii. *If $\Sigma = \Sigma^{(\leqslant 1)}$ and $\Sigma^{(1)}$ contains a special symbol and at least one more symbol, then* $\operatorname{otype}(\mathcal{T}(\Sigma), \prec_{\mathsf{kbo}}) = \omega^{\omega}$*.*
iii. *We have* $\operatorname{otype}(\mathcal{T}(\Sigma), \prec_{\mathsf{kbo}}) \leqslant \vartheta(\Omega^{\mathrm{Ar}(\Sigma)+1} \cdot \omega)$*.*
iv. *We have* $\operatorname{otype}(\mathcal{T}(\Sigma), \prec_{\mathsf{kbo}}) \leqslant \vartheta(\Omega^{3} \cdot \omega)$*.*

Proof. The result (i) is from Hofbauer (1991, 5.7) (or, equivalently, Hofbauer (2000)), while (ii) is from Touzet (1997, 5.2.20). We can find (iii) in Touzet (1997, 4.2.23), and (iv) is just a combination of (ii) and (iii). □

Lemma 4.4. *If $\prec_{\mathsf{kbo}}$ is a KBO over Σ and $\Sigma^{(\geqslant 1)}$ contains a special symbol and at least one more symbol, then* $\operatorname{otype}(\mathcal{T}(\Sigma), \prec_{\mathsf{kbo}}) \geqslant \omega^{\omega}$*.*

Proof. Let $\prec^{\Gamma}_{\mathsf{kbo}}$ be the NKBO over the signature Γ consisting of $i \succ g \succ e$ with weights 0, 1, 1 and arities 1, 1, 0. Since we have $\operatorname{otype}(\mathcal{T}(\Gamma), \prec^{\Gamma}_{\mathsf{kbo}}) = \omega^{\omega}$ by Theorem 4.3.ii, it suffices to construct an embedding o from $(\mathcal{T}(\Gamma), \prec^{\Gamma}_{\mathsf{kbo}})$ into $(\mathcal{T}(\Sigma), \prec_{\mathsf{kbo}})$. By our convention, Σ contains a constant e', and the premise gives us a $k+1$-ary symbol $f \neq i$. We put

$$o(e) := e', \quad o(i(s)) := i(o(s)), \quad \text{and} \quad o(g(s)) := f(o(s), e', \dots, e') \ .$$

An induction on $s \in \mathcal{T}(\Gamma)$ shows we have $\mu_{\Sigma}(o(s)) = \alpha \cdot (\mu_{\Gamma}(s) - 1) + \mu_{\Sigma}(e')$, with $\alpha := \mu_{\Sigma}(f) + k \cdot \mu_{\Sigma}(e')$. The weight conditions for KBO imply $\alpha > 0$, hence, concerning weights, o is order preserving. Next we observe that $s \equiv i^{a} s'$ leads to $o(s) \equiv i^{a} o(s')$. By induction on $\operatorname{dp}(s) + \operatorname{dp}(t)$ we can now show $s \succ^{\Gamma}_{\mathsf{kbo}} t$ implies $o(s) \succ_{\mathsf{kbo}} o(t)$. □

We know from Theorems 4.1 and 4.2 that the order types in reach of either MPO or LPO over monadic signatures are located far below the bounds for arbitrary signatures. Quite surprisingly, this is not the case with KBO. The maximal order type is ω^{ω}, and it is already attainable over a monadic signature with just two unary symbols.

Theorem 4.5. *If $\prec_{\mathsf{kbo}}$ is a KBO over Σ, then* $\operatorname{otype}(\mathcal{T}(\Sigma), \prec_{\mathsf{kbo}}) \leqslant \omega^\omega$.

Proof. Because of $\operatorname{otype}(\mathbb{N}^*, <^*_{\mathsf{lex}}) = \omega^\omega$ (see Corollary 2.40) it suffices to find a mapping o which embeds $(\mathcal{T}(\Sigma), \prec_{\mathsf{kbo}})$ into $(\mathbb{N}^*, <^*_{\mathsf{lex}})$. We may suppose

$$\Sigma = \{f_1, \dots, f_m, i\} \quad \text{and} \quad f_m \succ \dots \succ f_1 \ .$$

Take the function ν from Proposition 3.66 and put $b := \max\{\operatorname{Ar}(\Sigma), 3\} + 1$. We recursively define o for $s = i^a f_j(s_1, \dots, s_n)$ by

$$o(s) := (\nu(s), a, j) * o(s_1) * \dots * o(s_n) * q$$

where q is a (nonempty) sequence of zeros of such a length that $|o(s)| = b^{\nu(s)+1}$ holds. An induction on $\operatorname{dp}(s)$ shows $o(s)$ is well-defined – for the unique $p \in \mathbb{N}^*$ with $o(s) = p * q$ we get

$$|p| = 3 + \sum_{k=1}^{n} |o(s_k)| = 3 + \sum_{k=1}^{n} b^{\nu(s_k)+1} \leqslant 3 + n \cdot b^{\nu(s)} < b^{\nu(s)+1} \ .$$

Thus $\mu(s) > \mu(t)$ yields $o(s) >^*_{\mathsf{lex}} o(t)$, while $\mu(s) = \mu(t)$ implies $|o(s)| = |o(t)|$. Before we can establish that o is an embedding we show

$$\big(s \succ_{\mathsf{kbo}} t \wedge |o(s) * r| = |o(t) * r'|\big) \implies o(s) * r >^*_{\mathsf{lex}} o(t) * r' \qquad (4.1)$$

for all $r, r' \in \mathbb{N}^*$ by induction on $\operatorname{dp}(s) + \operatorname{dp}(t)$. Let s, t, r, r' fulfill the premise. If $\mu(s) > \mu(t)$, then we have $\nu(s) > \nu(t)$, which by definition of o implies (4.1), and if $\mu(s) = \mu(t)$, then the cases in which $o(s)$ and $o(t)$ differ in the second or third component can be handled in the same way. It remains to treat the case $s = i^a f_j(s_1, \dots, s_n)$ and $t = i^a f_j(t_1, \dots, t_n)$ with $(s_1, \dots, s_n) \succ^{\mathsf{lex}}_{\mathsf{kbo}} (t_1, \dots, t_n)$. Let k be minimal such that $s_k \succ_{\mathsf{kbo}} t_k$. There are $p, p_s, p_t \in \mathbb{N}^*$ such that $o(s) = p * o(s_k) * p_s$ and $o(t) = p * o(t_k) * p_t$. Putting $q := p_s * r$ and $q' := p_t * r'$ we get $|o(s_k) * q| = |o(t_k) * q'|$ since $|o(s) * r| = |o(t) * r'|$. The induction hypothesis yields $o(s_k) * q >^*_{\mathsf{lex}} o(t_k) * q'$, and this implies (4.1).

It remains to show that o is an embedding. If $s \succ_{\mathsf{kbo}} t$, then we either have $\mu(s) > \mu(t)$, which was already treated above, or $\mu(s) = \mu(t)$, which implies $|o(s)| = |o(t)|$. In this case (4.1) with $r = r' = (\,)$ suffices. □

In a similar but a little more involved way it is possible to embed $(\mathcal{T}(\Sigma), \prec_{\mathsf{kbo}})$ into $(\mathcal{T}(\Gamma), \prec^{\Gamma}_{\mathsf{kbo}})$ from Lemma 4.4.

Theorem 4.6. *Let Σ be a signature whose special symbol, provided that it exists, is i. For the KBO $\prec_{\mathsf{kbo}}$ based on Σ we have*

$$\operatorname{otype}(\mathcal{T}(\Sigma), \prec_{\mathsf{kbo}}) = \begin{cases} \omega & \text{if } i \notin \Sigma \ , \\ \omega \cdot \operatorname{card}(\{\mu(e) : e \in \Sigma^{(0)}\}) & \text{if } \Sigma = \Sigma^{(0)} \cup \{i\} \ , \\ \omega^\omega & \text{otherwise.} \end{cases}$$

Proof. In absence of a special symbol only finitely many closed terms share the same weight. Since $\Sigma^{(0)} \neq \Sigma$ by convention, the set of possible weights is infinite. According to Proposition 3.66 it is ordered like $\mathbb{N}$. This yields an order type of ω. Let us turn to the case $\Sigma = \Sigma^{(0)} \cup \{i\}$. If the weights of $c_1, \dots, c_m \in \Sigma^{(0)}$ coincide, then we end up with a simple lexicographic comparison:

$$i^a c_j \prec_{\mathsf{kbo}} i^b c_k \iff a < b \vee (a = b \wedge c_j \prec c_k) \; .$$

By Proposition 2.38.ii this yields

$$\operatorname{otype}(\mathcal{T}(\{i, c_1, \dots, c_m\}), \prec_{\mathsf{kbo}}) = \operatorname{otype}(\{c_1, \dots, c_m\}, \prec) \cdot \omega = \omega \; .$$

Thus for this KBO we get an order type of $\omega \cdot n$ where n is the number of distinct weights of the constants. In the remaining case, the order type is determined by Lemma 4.4 and Theorem 4.5. □

Corollary 4.7.

i. *Termination via KBO implies ω^ω-termination.*

ii. *Termination via KBO$^-$ implies ω-termination.*

Proof. Let the TRS $\mathcal{R}$ be compatible with some KBO $\prec_{\mathsf{kbo}}$ whose underlying $\prec$ is total. By Theorem 3.58.ii, $\prec_{\mathsf{kbo}}$ is total on ground terms. Under these circumstances Lemma 3.39 applies and shows $\mathcal{R}$ is $\operatorname{otype}(\mathcal{T}(\Sigma), \prec_{\mathsf{kbo}})$-terminating. Since we can safely add symbols, we are free to arrive at ω^ω-termination. □

This result will be improved later, see Theorem 5.17.

Even if we allow for variables, order types beyond ω^ω are not in reach. Any KBO $\prec_{\mathsf{kbo}}$ on Σ can be extended to a KBO on $\Gamma := \Sigma \cup \{e\}$ where e is a fresh constant with $\mu(e) := \mu(x)$ and $f \succ e$ for all $f \in \Sigma$. The mapping from $\mathcal{T}(\Sigma, \mathcal{V})$ into $\mathcal{T}(\Gamma)$ which simply replaces all variables in a term with e embeds $(\mathcal{T}(\Sigma, \mathcal{V}), \prec_{\mathsf{kbo}})$ into $(\mathcal{T}(\Gamma), \prec_{\mathsf{kbo}}^{\Gamma})$. Combining this with Theorem 4.6 we get

$$\operatorname{otype}(\mathcal{T}(\Sigma, \mathcal{V}), \prec_{\mathsf{kbo}}) = \begin{cases} \omega \cdot \operatorname{card}(\{\mu(u) : u \in \Sigma^{(0)} \cup \mathcal{V}\}) & \text{if } \Sigma = \Sigma^{(0)} \cup \{i\} \; , \\ \operatorname{otype}(\mathcal{T}(\Sigma), \prec_{\mathsf{kbo}}) & \text{otherwise.} \end{cases}$$

Though we did not introduce this, it is common practice in rewriting theory to consider terms and orders over infinite signatures. Just as the embeddings of Touzet we presented in Theorem 4.1 are extendible to infinite signatures (as precedences get infinite the natural numbers a_f have to be replaced with possibly infinite ordinals α_f), our Theorem 4.5 can be extended to well-founded KBOs over infinite signatures. Yet again we see that KBO is not able to generate large order types.

Theorem 4.8. *If $\prec_{\mathsf{kbo}}$ is a well-founded KBO over the possibly infinite signature Σ (based on the well-founded precedence $\prec$), then*

$$\operatorname{otype}(\mathcal{T}(\Sigma), \prec_{\mathsf{kbo}}) \leqslant \max\{\omega, \operatorname{otype}(\Sigma, \prec)\}^\omega \; .$$

4.3 Order Types of Simplification Orders

The following two counterparts to Theorems 4.2 and 4.1 show that, concerning order types, LPOs are essentially the most complex simplification orders.

Theorem 4.9 (de Jongh and Parikh 1977). *If $(\mathcal{T}(\Sigma), \prec)$ is a simplification order over a monadic signature which contains exactly n unary symbols, then*

$$\mathrm{otype}(\mathcal{T}(\Sigma), \prec) \leqslant \omega^{\omega^{n-1}} < \omega_3 \ .$$

Theorem 4.10 (Schmidt 1979). *If $(\mathcal{T}(\Sigma), \prec)$ is a simplification order, then*

$$\mathrm{otype}(\mathcal{T}(\Sigma), \prec) < \vartheta(\Omega^\omega) \ .$$

The Theorem is but a small part of a more general result of Schmidt (1979).

Corollary 4.11.

i. *A TRS is totally terminating if and only if it is $<\vartheta(\Omega^\omega)$-terminating.*

ii. *An SRS is totally terminating if and only if it is $<\omega_3$-terminating.*

Proof. Let $\mathcal{R}$ be a totally terminating TRS over the signature Σ. The proof of Proposition 3.36 shows $\mathcal{R}' := \mathcal{R} \cup \mathrm{Emb}(\Sigma)$ is also totally terminating. By Theorem 3.46 there exists a well-order $(\mathcal{T}(\Sigma), \prec)$ which is closed under contexts and normalizes $\mathcal{R}'$. Just as in (3.4) (see the proof of Proposition 3.31), we can extend $(\mathcal{T}(\Sigma), \prec)$ to a monotone Σ-algebra which is compatible with $\mathcal{R}'$. Because $\mathrm{Emb}(\Sigma)$ is contained in $\mathcal{R}'$, the underlying order $(\mathcal{T}(\Sigma), \prec)$ is a total simplification order. Now we may invoke Theorem 4.10 and identify the closed terms with ordinals below some $\alpha < \vartheta(\Omega^\omega)$ (respectively Theorem 4.9 for (ii) and ordinals below ω_3). □

In this chapter we saw that the order types of simplification orders (on terms) can get huge, and that LPO exhausts these order types, for terms as well as for strings. In contrast to this, any KBO has a surprisingly tiny order type, while MPO resides between the other two termination proof methods. The following chapter contains a comparison of these three termination proof methods concerning the maximal complexities TRSs terminating via one of them can attain. As a main results we will see that KBO and MPO have to exchange positions.

Some of the results we presented in this chapter are collected, together with complexity results, in Table 5.2 on page 112.

The reader interested in further results which also treat infinite signatures is invited to consult Hasegawa (1994).

5 Derivation Lengths

Bravo! You solve problems
in a flash.

We know from Theorem 3.8 that any TM can be turned into an equivalent SRS, under preservation of termination. The computation process is transferable into a corresponding derivation with at least as many rewrite steps as computation steps. Hence any recursive function is dominated by the derivation length function of some terminating TRS. As the field of recursive functions is far too wide, we restrict our attention to smaller collections of recursive functions.

It is a natural question to ask what complexity bound is imposed by a termination proof via some given collection of simplification orders. This question is a difficult one, and only partial answers have been given.

There is a rich hierarchy of function classes which can be characterized by corresponding termination proof methods. Most of the results about upper bounds on derivation length functions we can find in the literature are concerned with ω-termination. This is mainly due to the fact that a Σ-algebra witnessing ω-termination can effectively be used to impose an upper bound on the derivation length function of the TRS in question. Apart from obvious members of ω-termination such as PT, we will also encounter termination via MPO, and even termination via LPO. This is quite surprising if we take a look at the huge order types of LPOs we met in Theorem 4.1.

One of the central results of this chapter states that MPO is much weaker than LPO. According to Hofbauer (1991, 1992), termination via MPO implies primitive recursive termination, which imposes primitive recursive bounds on the derivation length functions. In contrast to this, Weiermann (1995) showed, by means of the hierarchy of fast growing functions with indices below ω^ω, that LPO corresponds to multiple recursive termination, ending up with multiple recursive complexity bounds. Both two bounds are essentially optimal.

As usual, things are quite different with KBO. We know from Lemma 3.63 that, due to the flexibility supplied by the special symbol, KBO is able to prove termination of TRSs which are not $<\omega^2$-terminating. This may be a reason why no optimal bounds on the complexities occuring within termination via KBO have been obtained so far. Hofbauer and Lautemann (1989) constructed a tricky TRS terminating via KBO whose complexity grows slightly faster than (the diagonalization of) the binary Ackermann function, while Hofbauer (1991, 2000) described a 4-recursive function whose branches served as uniform upper bounds on the complexities which occur within termination via KBO.

We will close the gap between these two bounds by showing that the size complexity of a TRS terminating via KBO can always be bounded from above by a member of $\mathrm{Ack}(O(n), 0)$, while $\mathrm{Ack}(2^{O(n)}, 0)$ suffices for the depth complexity. Both two bounds are shown to be essentially optimal. In the proof some of the techniques used by Weiermann (1995) in his treatment of LPO can be incorporated. One central issue is the use of fast growing functions with indices up to ω. Our approach is general enough to cover certain extensions of KBO like KBO with polynomial weight functions. This may however lead to bounds beyond $\mathrm{Ack}(2^{O(n)}, 0)$.

A slight transformation of the interpretation into $\mathbb{N}$ which is used in our construction shows that termination via KBO implies ω^2-termination. Hence termination via either PT, MPO, LPO, or KBO is contained in ω^2-termination. Compare this with the rather simple SRSs $\mathcal{R}_\alpha$ of Proposition 3.41 which are ω^α-terminating but not $<\omega^\alpha$-terminating for any $\alpha \leqslant \omega$. Since all well-known syntactic simplification orders live at the very beginning of the α-termination hierarchy, there is some need for stronger simplification orders.

Let us now turn to simple termination in general. Cichon and Tahhan Bittar (1998) showed that the complexity of any simply terminating SRS is bounded by a multiple recursive function. The construction made heavy use of the (somewhat extended) Hardy hierarchy up to ω_3. At the same time, Weiermann (1994) proved that for any simply terminating TRS there is $\alpha < \vartheta(\Omega^\omega)$ such that the complexity is dominated by the Hardy function H_α. Thus there are always $<\vartheta(\Omega^\omega)$-recursive complexity bounds.

For quite a while these upper bounds appeared to be far too high. Hofbauer's SRSs computing branches of the Ackermann function (see Proposition 5.5) which terminated via MPO supplied the largest known complexities of SRSs. They exhausted but remained inside the growth rates occuring in PREC, the collection of primitive recursive functions, contrasting the multiple recursive upper bounds mentioned above. The situation for terms was similar. Here the largest known complexities accompanied TRSs terminating via LPO which compute the k-ary Ackermann functions. As these functions are cofinal (with respect to $<_{\mathrm{ed}}$) in the multiple recursive functions, they are $<\omega_3$-recursive.

Connections between these lower bounds, the order types of simplification orders, and the slow growing functions are known from proof theory. This led to the formulation of the *slow growing principle*, implicit in Cichon (1992), stating that the complexity of a TRS terminating via a simplification order of order type α is eventually dominated by a slow growing function with index closely related to α. The principle is valid for both MPO and LPO. If it was also valid in general, then the multiple recursive lower bounds would be upper bounds, too. Touzet constructed counterexamples for both SRSs and TRSs.

The TRS used by Hofbauer and Lautemann to prove that termination via KBO allows for complexities similar to the Ackermann function was transformed by Touzet (1997) into a simply (yet not totally) terminating SRS, showing that SRSs with complexities beyond the primitive recursive functions exist. Touzet (1999) improved this result later and proved that the upper bound of Cichon and Tahhan Bittar is essentially optimal. The complexities of simply (and even totally) terminating SRS are thus cofinal in the multiple recursive functions.

Turning to TRSs, Touzet (1998b) left multiple recursion by proving that, for each n, there is a totally terminating TRS whose complexity is not $<\omega_n$-recursive. The proofs of the latter two results rely on Hardy functions, just as the proofs of the upper bounds do.

With the TRSs of Touzet one still had to live within $<\varepsilon_0$-recursion. Touzet (1999) conjectured it is possible to extend her approach to all ordinals below the small Veblen number $\vartheta(\Omega^\omega)$, showing Weiermann's huge upper bound is essentially optimal. One of our chief results is the validation of this conjecture. By a stepwise extension of Touzet's approach to the Hardy hierarchy below $\vartheta(\Omega^\omega)$ we prove that the complexities of simply (even totally) terminating TRSs are cofinal in the $<\vartheta(\Omega^\omega)$-recursive functions, showing that enormous growth rates are possible here. These growth rates are by far not reached by standard simplification orders like MPO, LPO, and KBO.

This closes the case for problem 81 in the RTA list of open problems, described in Dershowitz et al. (1995), whose revision in Dershowitz and Treinen (1998) asks what maximal complexity can be reached by simply terminating TRSs. The original problem was posed by Weiermann.

After proving that the slow growing principle does not always hold, Touzet (1999) proposed to replace the slow growing functions with Hardy functions, arriving at the *Hardy function principle*. From Buchholz et al. (1994) we know that the upper bounds described by a version of this principle are correct. Only little is known about the tightness of these bounds. It appears to be wise not to consider single TRSs but to focus on sets of TRSs terminating via certain termination proof methods. For general simple termination we will soon confirm the announced bounds are essentially optimal.

This optimality is (partially) lost if we consider the three standard examples of syntactic simplification orders. Although the order types of MPOs are only bounded by $\vartheta(\Omega \cdot \omega)$, MPO is far from being able to make use of the full computational power of such large order types, as only primitive recursive complexities occur. This corresponds to $<\omega^\omega$-recursion. Likewise LPO, whose order types are cofinal in the order types of simplification orders, gets stuck at $<\omega_3$-recursion, far below $<\vartheta(\Omega^\omega)$-recursion. At least KBO is able to exhaust its tiny maximal order type ω^ω, as $\mathrm{H}_{\omega^\omega}$ is a version of the Ackermann function.

Table 5.2 on page 112 collects the main results of this and the former chapter.

5.1 ω-termination

The most natural – though by far not trivial – termination proof methods are contained in ω-termination. We will meet some old friends like polynomial termination and multiple recursive termination, but surprisingly also termination via MPO and LPO appear. It is a very pleasant feature of ω-termination that the witnessing Σ-algebra can immediately be used to find a bound on the corresponding derivation length function.

Theorem 5.1 (Hofbauer 1991, 2.19). *Let $\mathcal{R}$ be a TRS over signature Σ whose ω-termination is witnessed by the Σ-algebra $(\omega, \mathcal{F})$. For any monotone $p\colon \mathbb{N} \to \mathbb{N}$ satisfying*

$$p(m) \geqslant [f](m, \ldots, m)$$

for all $f \in \Sigma$ and all $m \in \mathbb{N}$ we have

i. $\mathrm{dl}_{\mathcal{R}}(s) \leqslant p^{\mathrm{dp}(s)}(0)$ *for all $s \in \mathcal{T}(\Sigma)$, and*

ii. $\mathrm{Dc}_{\mathcal{R}}(n) \leqslant \mathrm{Dl}_{\mathcal{R}}(n) \leqslant p^n(0)$ *for all $n \in \mathbb{N}$.*

Proof. By Lemma 3.29, $[\![\cdot]\!]$ is a monotone interpretation of $\mathcal{T}(\Sigma)$ in $(\mathbb{N}, <)$ which normalizes $\mathcal{R}$. Thus we get, using Lemma 3.25.ii, $\mathrm{dl}_{\mathcal{R}}(s) \leqslant [\![s]\!]$ for all closed terms s. To reach (i) we show $[\![s]\!] \leqslant p^{\mathrm{dp}(s)}(0)$ by induction on s. Frequently relying on the weak subterm property of p, which is a consequence of Lemma 2.20, for $s = f(s_1, \ldots, s_n)$ we get

$$\begin{aligned}[\![f(s_1, \ldots, s_n)]\!] &= [f]([\![s_1]\!], \ldots, [\![s_n]\!]) \leqslant [f](p^{\mathrm{dp}(s_1)}(0), \ldots, p^{\mathrm{dp}(s_n)}(0)) \\ &\leqslant [f](p^{\mathrm{dp}(s)-1}(0), \ldots, p^{\mathrm{dp}(s)-1}(0)) \leqslant p(p^{\mathrm{dp}(s)-1}(0)) = p^{\mathrm{dp}(s)}(0)\ .\end{aligned}$$

From (i) and (3.3) we directly infer (ii). □

If $\mathcal{C}$ is one of the function classes introduced in Section 2.4, it is now an easy task to impose bounds on the derivation lengths which are possible within $\mathcal{C}$-termination. We just have to see where we can find the iteration of the maximum function taken over finitely many elements of $\mathcal{C}$.

Theorem 5.2 (Geupel 1988, Lautemann 1988). *Polynomial termination of a TRS $\mathcal{R}$ implies $\mathrm{Dc}_{\mathcal{R}}$ and $\mathrm{Dl}_{\mathcal{R}}$ are in $2^{2^{O(n)}}$. This result is essentially optimal.*

Theorem 5.3 (Hofbauer and Lautemann 1989). *Linear termination of a TRS $\mathcal{R}$ implies $\mathrm{Dc}_{\mathcal{R}}$ and $\mathrm{Dl}_{\mathcal{R}}$ are in $2^{O(n)}$. This result is essentially optimal.*

More generally, as a direct consequence of Theorem 5.1 and Theorem 2.68, we get the following Theorem. Note that the upper bound parts of (i) and Theorem 5.2 are just special cases of (ii).

Theorem 5.4 (Hofbauer 1991, 2.20, 2.21). *Let $\mathcal{R}$ be a TRS.*

i. *Exponential or even elementary termination of $\mathcal{R}$ implies $\mathrm{Dc}_{\mathcal{R}}$ and $\mathrm{Dl}_{\mathcal{R}}$ are in $\mathcal{E}_4$. This result is essentially optimal.*
ii. *For $m \geqslant 2$, $\mathcal{E}_m$-termination of $\mathcal{R}$ implies $\mathrm{Dc}_{\mathcal{R}}$ and $\mathrm{Dl}_{\mathcal{R}}$ are in $\mathcal{E}_{m+1}$.*
iii. *Primitive recursive termination of $\mathcal{R}$ implies $\mathrm{Dc}_{\mathcal{R}}$ and $\mathrm{Dl}_{\mathcal{R}}$ are in* Prec.
iv. *Multiple recursive termination of $\mathcal{R}$ implies $\mathrm{Dc}_{\mathcal{R}}$ and $\mathrm{Dl}_{\mathcal{R}}$ are in* Mrec.

Within termination via MPO derivation lengths beyond any given primitive recursive function are possible – even for SRSs.

Proposition 5.5 (Hofbauer 1991, 1992). *For every $f \in$* Prec *there is an SRS $\mathcal{R}$ terminating via MPO whose complexity eventually dominates f.*

Proof. We present a variant from Touzet (1997, 4.2.15), which is directly linked to the Ackermann function. Since there is, by Lemma 2.63.iv, a branch of the Ackermann function which dominates f, it suffices to show we can simulate all these branches by SRSs which terminate via MPO. For each n, consider the SRS $\mathcal{R}_n$ over $\{0, s, A_0, \ldots, A_n\}$ containing the rule $A_0 \to s$ and, for all $i \leqslant n$, the rules $A_{i+1}0 \to A_i s0$ and $A_{i+1}s \to A_i A_{i+1}$. The SRS terminates via the MPO based on $A_n \succ \cdots \succ A_0 \succ s \succ 0$. For all $j \leqslant n$ and all m we can show

$$\mathrm{dl}_{\mathcal{R}_n}(A_j s^m 0) \geqslant \mathrm{Ack}_j(m) \quad \text{and} \quad A_j s^m 0 \xrightarrow{*}_{\mathcal{R}} s^{\mathrm{Ack}_j(m)} 0$$

by simultaneous induction on j (with secondary induction on m). □

Hofbauer (1991, 1992) showed even a bit more – any Grzegorczyk class $\mathcal{E}_n$ is inhibited by the size complexity of an SRS terminating via MPO. The Proposition will get its final form in Theorem 6.4, which states that all primitive recursive functions are computable via MPO.

Theorem 5.6 (Hofbauer 1991, 1992). *Termination via MPO implies primitive recursive termination, hence there is always a primitive recursive bound on the derivation length function. This result is essentially optimal.*

We can combine Proposition 5.5 and Theorem 5.6 to see the Grzegorczyk classes $\mathcal{E}_n$ can be used to measure derivational complexity: $\mathcal{E}_n$-termination for $n \in \mathbb{N}$ constitutes a proper hierarchy inside ω-termination. Termination via LPO allows for much larger complexities.

Theorem 5.7 (Weiermann 1995). *Termination via LPO implies multiple recursive termination, hence there is always a multiple recursive bound on the derivation length function. This result is essentially optimal.*

A very elegant alternative proof of Theorem 5.6 and Theorem 5.7, using techniques from proof theory, is due to Buchholz (1995), and yet another nice proof of the latter Theorem is from Arai (1998).

A further hierarchy within ω-termination is provided by $\mathcal{M}_k$-termination (which is based on k-recursive functions). As the primitive recursive functions form a proper subset of the 2-recursive functions, this new hierarchy is located much higher than $\mathcal{E}_n$-termination.

It is unknown which α is minimal such that ω-termination implies $<\alpha$-recursive termination. From Theorems 5.7, 5.20, and 2.83 we know $\omega_3 \leqslant \alpha \leqslant \vartheta(\Omega^\omega)$.

5.2 Termination via KBO

We first present the best complexity bounds for TRSs terminating via KBO that are known so far. Note that these results are concerned with the size complexity function $\mathrm{Dc}_\mathcal{R}$ whose relation to $\mathrm{Dl}_\mathcal{R}$ is elucidated by (3.3).

Theorem 5.8. *Let $\mathcal{R}$ be a TRS over some signature Σ.*

i. *There is a TRS terminating via KBO for whom no primitive recursive upper bound on the size complexity exists.*
ii. *Let $\succ_{\mathsf{kbo}}$ be an NKBO based on Σ. The function mapping n to the maximal length of a sequence $s_1 \to_\mathcal{R} \ldots \to_\mathcal{R} s_m$ satisfying $s_1 \succ_{\mathsf{kbo}} \ldots \succ_{\mathsf{kbo}} s_m$ and $|s_1| \leqslant n$ has a 4-recursive bound.*
iii. *If $\mathcal{R}$ terminates via KBO, then $\mathrm{Dc}_\mathcal{R}$ has a 4-recursive bound.*
iv. *If $\mathcal{R}$ terminates via KBO$^-$ or, more general, via KBO with a special symbol i such that for all $(l,r) \in \mathcal{R}$ we have $|l|_i \geqslant |r|_i$, then $\mathrm{Dc}_\mathcal{R}$ is in $2^{O(n)}$. This bound is essentially optimal.*
v. *If $\mathcal{R}$ terminates via KBO(1), then $\mathrm{Dc}_\mathcal{R}$ is in $2^{O(n)}$. This bound is essentially optimal.*

Proof. The first proof of (i) is from Hofbauer and Lautemann (1989), a slightly improved version of this is Hofbauer (1991, 5.9). For a detailed proof see Theorem 6.13. Recall from Theorem 3.61 that termination via KBO is equivalent to termination via NKBO. The result (iii) for termination via NKBO is an easy consequence of (ii), which again was established by Hofbauer (1991, 5.20), see also Hofbauer (2000) for an improved version. In Hofbauer (1991, 5.10, 5.14) we can find (iv) and (v). For the latter it is shown that termination via KBO(1) implies termination via LT, thus making Theorem 5.3 applicable. □

For (iii), Touzet (1997, 4.2.30) independently showed that there is always a k-recursive bound for some $k \in \mathbb{N}$ which depends on the maximal arity of the function symbols contained in Σ. Her approach remains valid even if polynomial weight functions over $\mathbb{N}$ are considered.

Hofbauer (1991) remarks on Theorem 5.8.i,iii that

> *"the question if* 2- *or* 3-*recursive upper bounds (lower bounds resp.) exist is still an open problem."*

We will establish fairly low 2-recursive upper bounds on the complexities of TRSs terminating via KBO. Since we are interested in results on $\mathrm{Dl}_{\mathcal{R}}$, too, we have to adapt Theorem 5.8.i to this setting.

Lemma 5.9. *For any* $a \in \mathbb{N}$ *there is a TRS* $\mathcal{R}$ *terminating via KBO such that we have* $\mathrm{Dl}_{\mathcal{R}}(n) > \mathrm{Ack}(a^n, 0)$ *for all* $n \geqslant 3$.

For a better exposition the proof is deferred to Section 6.3.3, see page 125.

We are now going to construct 2-recursive complexity bounds for all TRSs terminating via KBO. This construction is in the spirit of Weiermann (1995), as it makes use of functions from the fast growing hierarchy. The letters A, D, and K will be used for certain natural numbers that are kept fixed throughout most of this section.

Let Σ be a signature equipped with weight μ and precedence $\succ$. Since, by Theorem 3.58, KBOs are incremental with respect to both signature and precedence, we may safely assume $i \in \Sigma$ holds and $\succ$ is linear:

$$\Sigma = \{f_1, \dots, f_K, i\} \quad \text{and} \quad i \succ f_K \succ \cdots \succ f_1 \, . \tag{5.1}$$

By ν we denote the function accompanying μ according to Proposition 3.66. Theorem 3.61 shows termination via KBO coincides with termination via NKBO. Hence, following the remark made at the top of the proof of Proposition 3.66, there appears to be no need to consider ν instead of μ. We do so because this way our proof can better be adapted to more sophisticated weight functions like real-valued polynomial functions or those mentioned above Definition 3.57. For KBOs based on these weights it appears to be unknown if it suffices to consider only weight functions living on $\mathbb{N}$. In order to impose an appropriate bound on the size complexities we will later switch to μ, though.

Suppose for the moment that we have a TRS $\mathcal{R}$ over Σ terminating via KBO. Our aim is to construct an embedding from $(\mathcal{T}(\Sigma), \overset{+}{\leftarrow}_{\mathcal{R}})$ into $(\mathbb{N}, <)$, under guidance of the definition of $\prec_{\mathsf{kbo}}$. Just as with MPO and LPO, order type considerations (cf. Lemma 2.13 and Theorem 4.6) show it is usually not possible to embed $(\mathcal{T}(\Sigma), \prec_{\mathsf{kbo}})$ into $(\mathbb{N}, <)$. We saw in Lemma 3.63 that, in contrast to the other two standard groups of syntactic simplification orders, termination via KBO does not imply ω-termination. Thus there is some $\mathcal{R}$ terminating via KBO with no compatible Σ-algebra $(\omega, \mathcal{F})$. If we intend to construct bounds on the derivation length function of $\mathcal{R}$, we have to come up with a new approach and can not fall back on Σ-algebras over some ordinal number, simply because we can not expect to extract information about $\mathrm{Dl}_{\mathcal{R}}$ from functions living on some (limit) ordinal $\lambda > \omega$.

For the construction of an embedding from $(\mathcal{T}(\Sigma), \overset{+}{\leftarrow}_{\mathcal{R}})$ into $(\mathbb{N}, <)$, the results of Section 4.2 suggest that a function is required which mostly behaves like an embedding of the finite sequences over $\mathbb{N}$ with lengths bounded by some m into the natural numbers. We construct such a function using the fast growing functions F_n from Definition 2.75, but first we put

$$A := 2 \cdot (\mathrm{Ar}(\Sigma) + 1) \geqslant 4$$

and fix some D which satisfies

$$D > K \quad \text{and} \quad D > \mathrm{Ar}(\Sigma) + 2\ .$$

Definition 5.10. For $0 \leqslant m < D$ such that $n - 2m \geqslant 0$ we recursively define

$$\mathcal{A}_n((a_1, \dots, a_m), c) := \begin{cases} c & \text{if } m = 0\ , \\ \mathrm{F}_n^{a_1 \cdot D}(\mathcal{A}_{n-2}((a_2, \dots, a_m), c)) & \text{otherwise.} \end{cases}$$

We will use $\mathcal{A}_n(a, c)$ as an abbreviation of $\mathcal{A}_n((a), c)$.

A list of all properties of $\mathcal{A}$ we will need is given in the following Lemma.

Lemma 5.11. *Consider m with $0 < m < D$.*

i. *We have $\mathcal{A}_n((a_1, \dots, a_m), c) > a_1, \dots, a_m, c$ as soon as $a_j > 0$ for some j.*
ii. *The function mapping $n, \bar{a}$, and c to $\mathcal{A}_n((a_1, \dots, a_m), c)$ is monotone in c, in each a_j and, if $c > 0$, in n. For $c = 0$ weak monotonicity in n holds.*
iii. *If $a_m > 0$, then $\mathcal{A}_n((a_1, \dots, a_m), c) > \mathcal{A}_n((a_1, \dots, a_{m-1}), c)$, and otherwise we get $\mathcal{A}_n((a_1, \dots, a_m), c) = \mathcal{A}_n((a_1, \dots, a_{m-1}), c)$.*
iv. *We have $\mathcal{A}_n(b, \mathcal{A}_n((a_1, \dots, a_m), c)) = \mathcal{A}_n((b + a_1, a_2, \dots, a_m), c)$.*
v. *If $\mathcal{A}_l(b, c) \geqslant a_1, \dots, a_m$ and $b > 0$, then $\mathcal{A}_{l+1}(b, c) > \sum_{k=1}^{m} a_k$.*
vi. *If $j \leqslant m$; $\mathcal{A}_l(b, c) \geqslant a_1, \dots, a_j$; $b > 0$; $\mathcal{A}_{l+1}(b, c) > c'$, and $n - 2j \geqslant l$, then $\mathcal{A}_{n+1}(b, c) > \mathcal{A}_n((a_1, \dots, a_j), c')$.*
vii. *If $n' > n$, then $\mathcal{A}_{n'}(a + 1, \mathcal{A}_n(b, c)) > \mathcal{A}_{n'}(a, \mathcal{A}_n(b + D, c))$.*
viii. *$\mathcal{A}_n((a + 1, b), c) > \mathcal{A}_n((a, b + D), c)$ holds.*
ix. *If $n \geqslant 4$, then $\mathcal{A}_n(a + b, c) \geqslant \mathcal{A}_n((a, b), c)$.*

Proof. The items (i) to (iii) are consequences of Lemma 2.76.i,ii, while (iv) is obvious. Because of Lemma 2.76.vii,vi, $D \geqslant 3$, and $D > m$ we can show (v):

$$\mathcal{A}_{l+1}(b,c) > \mathrm{F}_{l+1}^{bD-1}(c) > \mathrm{F}_{l}^{bD+1}(c) > \mathcal{A}_l(b,c) \cdot D \geqslant \sum_{k=1}^{m} a_k \;.$$

An induction on the $j \leqslant m$ such that $n - 2j \geqslant l$ yields (vi). For $j = 0$ there is not much to do, and for $j > 0$ we see

$$\begin{aligned}\mathcal{A}_{n+1}(b,c) &= \mathrm{F}_{n+1}(\mathrm{F}_{n+1}^{bD-1}(c)) > \mathrm{F}_{n+1}(\mathrm{F}_{n-1}^{bD+1}(c))\\ &> \mathrm{F}_{n}^{\mathrm{F}_l^{bD+1}(c)}(\mathrm{F}_{n-1}^{bD}(c)) \geqslant \mathrm{F}_{n}^{\mathrm{F}_l^{bD}(c)\cdot D}(\mathrm{F}_{n-1}^{bD}(c))\\ &= \mathcal{A}_n(\mathcal{A}_l(b,c), \mathcal{A}_{n-1}(b,c)) \geqslant \mathcal{A}_n(a_1, \mathcal{A}_{n-1}(b,c))\\ &> \mathcal{A}_n(a_1, \mathcal{A}_{n-2}((a_2,\dots,a_j),c')) = \mathcal{A}_n((a_1,\dots,a_j),c') \;,\end{aligned}$$

where we used Lemma 2.76.vii,vi and the induction hypothesis. Lemma 2.76.v yields (vii), a special case of whom is (viii). Finally, (iv) and (ii) imply (ix). □

As we explained above, it is usually not possible to define a Σ-algebra $(\omega, \mathcal{F})$ which is compatible with $\mathcal{R}$. Thus we have to deviate from the approaches of Theorems 5.6 and 5.7 and directly define an interpretation of $\mathcal{T}(\Sigma)$ in $(\mathbb{N}, <)$.

Definition 5.12. We define $\mathcal{I}\colon \mathcal{T}(\Sigma) \to \mathbb{N}$ for $s = i^a f_j(s_1,\dots,s_m)$ by

$$\mathcal{I}(s) := \mathcal{A}_{A\nu(s)}((a+1, j, \mathcal{I}(s_1), \dots, \mathcal{I}(s_{m-1})), \sum_{k=1}^{m} \mathcal{I}(s_k)) \;.$$

Note that $\mathcal{I}$ is well-defined by definition of A, since $A - 2(m+1) \geqslant 0$ and $A\nu(s) \geqslant A$. We mention without proof that there usually are $s, t \in \mathcal{T}(\Sigma)$ with $\mathcal{I}(s) > \mathcal{I}(t)$ and $\mathcal{I}(f(\dots,s,\dots)) < \mathcal{I}(f(\dots,t,\dots))$ (since termination via KBO does not imply ω-termination). This is only possible if $\mu(s) < \mu(t)$.

Lemma 5.13. *Let σ be a ground substitution and $s, t \in \mathcal{T}(\Sigma, \mathcal{V})$. We define $n := \nu(s\sigma)$ and, for $u \in \mathcal{T}(\Sigma, \mathcal{V})$, $S(u) := \sum_{x \in u} \mathcal{I}(x\sigma)$.*

i. *We have $\mathcal{I}((i^a s)\sigma) = \mathcal{A}_{An}(a, \mathcal{I}(s\sigma))$.*
ii. *$\mathcal{I}$ has the subterm property and satisfies $\mathcal{I}(s\sigma) > 0$.*
iii. *If $\mathcal{I}(s\sigma) > \mathcal{I}(t\sigma)$; $\mu(s\sigma) \geqslant \mu(t\sigma)$, and $f \in \Sigma^{(\geqslant 1)}$, then*

$$\mathcal{I}(f(\dots,s\sigma,\dots)) > \mathcal{I}(f(\dots,t\sigma,\dots)) \;.$$

iv. *$\mathcal{I}(s\sigma) \geqslant S(s)$ holds.*
v. *If $s = f_j(s_1,\dots,s_m)$, then $\mathcal{I}(s\sigma) \geqslant \mathcal{A}_{An}((1,j), S(s)) \geqslant \mathcal{A}_{An}((1,1), S(s))$.*

vi. *We have* $\mathcal{A}_{An}(\mathrm{dp}(s)+1, S(s)) > \mathcal{I}(s\sigma) \geqslant \mathcal{A}_{An}((1,1),0)$.

vii. *If* $s = f_j(s_1,\ldots,s_m)$ *and* $\mathrm{dp}(s) < D$*, then*

$$\mathcal{A}_{An}(2, S(s)) > \mathcal{A}_{An}((1, j+1), S(s)) > \mathcal{I}(s\sigma)\ .$$

viii. *If* $s = f_j(s_1,\ldots,s_m)$*;* $\mathrm{dp}(s) < D$*;* $1 \leqslant l < m$*, and* $n' := An - 2(l+1)$*, then*

$$\mathcal{A}_{n'}(\mathcal{I}(s_l\sigma)+1, S(s)) > \mathcal{A}_{n'}((\mathcal{I}(s_l\sigma),\ldots,\mathcal{I}(s_{m-1}\sigma)), \sum_{k=1}^{m}\mathcal{I}(s_k\sigma))\ .$$

Proof. The point (i) follows from Lemma 5.11.iv, while (ii) is an implication of Lemma 5.11.i, and (iii) is shown using (i) and Lemma 5.11.ii. By induction on s we prove (iv). This is easy if $s \in \mathcal{V}$ or $s = i(s')$, and if $s = f_j(s_1,\ldots,s_m)$, then

$$c := \sum_{k=1}^{m}\mathcal{I}(s_k\sigma) \geqslant \sum_{k=1}^{m} S(s_k) \geqslant S(s)$$

holds by the induction hypothesis, leading to

$$\mathcal{I}(s\sigma) \geqslant \mathcal{A}_{An}((1,j),c) \geqslant \mathcal{A}_{An}((1,j),S(s)) > S(s)$$

via Lemma 5.11.iii,ii,i. We can utilize this to show (v) as well as the second half of (vi), since $s\sigma = i^a f_j(\ldots)$. The first half of (vi) deserves an induction on s. If $s \in \mathcal{V}$, then $\mathcal{I}(s\sigma) = S(s)$, and for $s = i(s')$ we get the statement by virtue of (i), the induction hypothesis, and $S(s') = S(s)$. It remains to treat the case $s = f_j(s_1,\ldots,s_m)$. Because f_j is not special, for $1 \leqslant k \leqslant m$ the induction hypothesis implies

$$\mathcal{A}_{A(n-1)}(\mathrm{dp}(s), S(s)) \geqslant \mathcal{A}_{A\nu(s_k\sigma)}(\mathrm{dp}(s_k)+1, S(s_k)) > \mathcal{I}(s_k\sigma)\ ,$$

hence by Lemma 5.11.v we get $\mathcal{A}_{A(n-1)+1}(\mathrm{dp}(s), S(s)) > \sum_{k=1}^{m}\mathcal{I}(s_k\sigma)$. We use this and Lemma 5.11.vi,ii,viii,ix to get through the following calculation:

$$\begin{aligned}\mathcal{I}(s\sigma) &= \mathcal{A}_{An}((1,j), \mathcal{A}_{An-4}((\mathcal{I}(s_1\sigma),\ldots,\mathcal{I}(s_{m-1}\sigma)), \sum_{k=1}^{m}\mathcal{I}(s_k\sigma)))\\ &< \mathcal{A}_{An}((1,j), \mathcal{A}_{An-3}(\mathrm{dp}(s), S(s)))\\ &\leqslant \mathcal{A}_{An}((1, D+\mathrm{dp}(s)-1), S(s))\\ &< \mathcal{A}_{An}((2, \mathrm{dp}(s)-1), S(s))\\ &\leqslant \mathcal{A}_{An}(\mathrm{dp}(s)+1, S(s))\ .\end{aligned} \tag{5.2}$$

To prove (vii) we travel via (5.2) and see

$$\begin{aligned}\mathcal{I}(s\sigma) &< \mathcal{A}_{An}((1,j), \mathcal{A}_{An-3}(\mathrm{dp}(s), S(s))) < \mathcal{A}_{An}((1,j), \mathcal{A}_{An-3}(D, S(s)))\\ &< \mathcal{A}_{An}((1,j+1), S(s)) \leqslant \mathcal{A}_{An}((1,D), S(s)) < \mathcal{A}_{An}(2, S(s))\end{aligned}$$

by Lemma 5.11.vii,viii, since $\mathrm{dp}(s), j+1 \leqslant D$. It remains to prove (viii). Similar to the proof of (vi), for $1 \leqslant k \leqslant m$ we have $\mathcal{A}_{A(n-1)}(\mathrm{dp}(s), S(s)) > \mathcal{I}(s_k\sigma)$ and $\mathcal{A}_{A(n-1)+1}(\mathrm{dp}(s), S(s)) > \sum_{k=1}^{m} \mathcal{I}(s_k\sigma)$. Hence Lemma 5.11.vi implies

$$\mathcal{A}_{n'-1}(\mathrm{dp}(s), S(s)) > \mathcal{A}_{n'-2}((\mathcal{I}(s_{l+1}\sigma), \ldots, \mathcal{I}(s_{m-1}\sigma)), \sum_{k=1}^{m} \mathcal{I}(s_k\sigma)) =: b \; .$$

Relying on this and $D > \mathrm{dp}(s)$,

$$\mathcal{A}_{n'}(\mathcal{I}(s_l\sigma) + 1, S(s)) > \mathcal{A}_{n'}(\mathcal{I}(s_l\sigma), \mathcal{A}_{n'-1}(D, S(s))) > \mathcal{A}_{n'}(\mathcal{I}(s_l\sigma), b)$$

follows from Lemma 5.11.vii. □

Let $\rhd$ be the closure of $\{(s,t) : s \succ_{\mathsf{kbo}} t \wedge \mathrm{dp}(t) < D\}$ under ground substitutions. The next Lemma shows $\mathcal{I}$ is able to embed $(\mathcal{T}(\Sigma), \lhd)$ into $(\mathbb{N}, <)$. This result is essentially optimal, since the whole of $\succ_{\mathsf{kbo}}$ is not embeddable.

Lemma 5.14. *Let σ be a ground substitution and $s, t \in \mathcal{T}(\Sigma, \mathcal{V})$ with $\mathrm{dp}(t) < D$. If $s \succ_{\mathsf{kbo}} t$, then $\mathcal{I}(s\sigma) > \mathcal{I}(t\sigma)$.*

Proof. The proof is by induction on $\mathrm{dp}(s) + \mathrm{dp}(t)$. Let n and S be defined as in Lemma 5.13, and put $n' := \nu(t\sigma)$. Since $s \succ_{\mathsf{kbo}} t$ we have $s \notin \mathcal{V}$, $n \geqslant n'$, and, due to $\mathcal{V}(s) \supseteq \mathcal{V}(t)$, $S(s) \geqslant S(t)$. We first treat the case $n > n'$. Because of $s \notin \mathcal{V}$ we get $\mathcal{I}(s\sigma) \geqslant \mathcal{A}_{An}(1, S(s))$ by Lemma 5.13.i,iv,v, and

$$\begin{aligned} \mathcal{A}_{An}(1, S(s)) > \mathcal{A}_{An-2}(D, S(s)) &\geqslant \mathcal{A}_{An'}(D, S(t)) \\ &\geqslant \mathcal{A}_{An'}(\mathrm{dp}(t) + 1, S(t)) > \mathcal{I}(t\sigma) \end{aligned}$$

follows with the help of Lemma 5.11.viii,ii and Lemma 5.13.vi.

If $n = n'$, then we have $s \equiv i^a s'$ and $t \equiv i^b t'$ with $a \geqslant b$. First we take care of the case $a > b$. If $s' \in \mathcal{V}$, then weight considerations imply $t' \in \mathcal{V} \cup \Sigma^{(0)}$. For $t' \in \mathcal{V}$ we can infer $t' = s'$, which yields $\mathcal{I}(s\sigma) > \mathcal{I}(t\sigma)$ since t is a proper subterm of s. If $t' \in \Sigma^{(0)}$, then we get $S(t) = 0$, and Lemma 5.13.i,vi,vii lead to

$$\begin{aligned} \mathcal{I}(s\sigma) \geqslant \mathcal{A}_{An}((a+1, 1), 0) > \mathcal{A}_{An}(a+1, 0) &\geqslant \mathcal{A}_{An}(b+2, S(t)) \\ = \mathcal{A}_{An}(b, \mathcal{A}_{An}(2, S(t'))) &> \mathcal{A}_{An}(b, \mathcal{I}(t'\sigma)) = \mathcal{I}(t\sigma) \; . \end{aligned} \tag{5.3}$$

For $s' = f_j(s_1, \ldots, s_m)$ we get $t' = f_{j'}(t_1, \ldots, t_{m'})$, since $t' \in \mathcal{V}$ is impossible. From observations very similar to the ones needed in (5.3) we infer

$$\mathcal{I}(s\sigma) \geqslant \mathcal{A}_{An}((a+1, 1), S(s)) > \mathcal{A}_{An}(b+2, S(t)) > \mathcal{I}(t\sigma) \; .$$

The case $a = b$ is next. Yet again, $s' = f_j(s_1, \ldots, s_m)$ and $t' = f_{j'}(t_1, \ldots, t_{m'})$ with $j \geqslant j'$ appear. In view of Lemma 5.13.i it suffices to show $\mathcal{I}(s'\sigma) > \mathcal{I}(t'\sigma)$. If $j > j'$, then Lemma 5.13.v,vii yield

$$\mathcal{I}(s'\sigma) \geqslant \mathcal{A}_{An}((1, j), S(s')) \geqslant \mathcal{A}_{An}((1, j'+1), S(t')) > \mathcal{I}(t'\sigma) \; ,$$

while $j = j'$ implies $m = m'$ as well as $(s_1, \dots, s_m) \succ_{\mathsf{kbo}}^{\mathsf{lex}} (t_1, \dots, t_m)$. Let l be minimal such that $s_l \succ_{\mathsf{kbo}} t_l$. By the induction hypothesis $\mathcal{I}(s_l\sigma) > \mathcal{I}(t_l\sigma)$ holds. If $m = l$, then Lemma 5.13.i easily gives $\mathcal{I}(s'\sigma) > \mathcal{I}(t'\sigma)$, and otherwise we have $1 \leqslant l < m$. We put $n'' := An - 2(l+1)$ and get

$$\begin{aligned} &\mathcal{A}_{n''}((\mathcal{I}(s_l\sigma), \dots, \mathcal{I}(s_{m-1}\sigma)), \sum_{k=1}^{m} \mathcal{I}(s_k\sigma)) \\ \geqslant\ &\mathcal{A}_{n''}(\mathcal{I}(s_l\sigma), S(s)) \\ \geqslant\ &\mathcal{A}_{n''}(\mathcal{I}(t_l\sigma) + 1, S(t)) \\ >\ &\mathcal{A}_{n''}((\mathcal{I}(t_l\sigma), \dots, \mathcal{I}(t_{m-1}\sigma)), \sum_{k=1}^{m} \mathcal{I}(t_k\sigma)) \end{aligned}$$

by Lemma 5.13.iv,viii. This implies $\mathcal{I}(s'\sigma) > \mathcal{I}(t'\sigma)$. □

Theorem 5.15. *If a TRS $\mathcal{R}$ terminates via KBO, then its complexity* $\mathrm{Dl}_{\mathcal{R}}$ *is a member of* $\mathrm{Ack}(2^{O(n)}, 0)$*. This bound is essentially optimal.*

Proof. We add $D > \max\{\mathrm{dp}(r) : (l, r) \in \mathcal{R}\}$ to our former conditions on D. Lemma 5.14 shows that $\mathcal{I}$ normalizes $\mathcal{R}$, hence Lemma 5.13.iii implies $\mathcal{I}$ is an embedding of $(\mathcal{T}(\Sigma), \xleftarrow{+}_{\mathcal{R}})$ into $(\mathbb{N}, <)$. Lemma 3.25.i yields $\mathrm{dl}_{\mathcal{R}}(s) \leqslant \mathcal{I}(s)$ for all $s \in \mathcal{T}(\Sigma)$. We borrow an $a \in \mathbb{N}$ from Proposition 3.66. For all $s \in \mathcal{T}(\Sigma)$, putting $n := \mathrm{dp}(s) \geqslant 1$ we get

$$\begin{aligned} \mathrm{dl}_{\mathcal{R}}(s) &\leqslant \mathcal{I}(s) \\ &< \mathcal{A}_{A\nu(s)}(n+1, 0) && \text{by Lemma 5.13.vi} \\ &= \mathrm{F}_{A\nu(s)}^{D(n+1)}(0) \\ &< \mathrm{F}_{A\nu(s)+1}(D(n+1)) \\ &< \mathrm{Ack}(A\nu(s) + 4, 3D(n+1)) && \text{by Lemma 2.76.iii} \qquad (5.4) \\ &< \mathrm{Ack}(Aa^n + 4, 3D(n+1)) && \text{by Proposition 3.66.} \end{aligned}$$

As $A \geqslant 4$, any $b \geqslant a, 2D$ fulfills $Ab^n + 4 > 3D(n+1)$. Assembling everything, for $c := Ab + 6$ we get

$$\begin{aligned} \mathrm{dl}_{\mathcal{R}}(s) &< \mathrm{Ack}(Aa^n + 4, 3D(n+1)) \\ &< \mathrm{Ack}(Ab^n + 4, Ab^n + 4) \leqslant \mathrm{Ack}(Ab^n + 6, 0) \leqslant \mathrm{Ack}(c^n, 0) \end{aligned}$$

using Lemma 2.63.vii. Lemma 5.9 shows this bound is essentially optimal. □

The methods presented so far suffice to treat more sophisticated weight functions, provided that they have the subterm property for all symbols except the special one, and provided further that the weight can be replaced with a function into $\mathbb{N}$ like the ν of Proposition 3.66. Such a ν may however be located in a higher complexity class, resulting in bounds different from the ones of Theorem 5.15.

In order to impose optimal bounds on the size complexities occuring within termination via KBO we have to be a bit more picky about the weight function, simply because we have to rely on the equivalence of termination via KBO and termination via NKBO. This equivalence is only known for the usual weight.

Theorem 5.16. *If a TRS $\mathcal{R}$ terminates via KBO, then its size complexity* $\mathrm{Dc}_{\mathcal{R}}$ *is a member of* $\mathrm{Ack}(O(n), 0)$. *This bound is essentially optimal.*

Proof. Theorem 3.61 tells us $\mathcal{R}$ terminates via NKBO. As we already remarked at the top of the proof of Proposition 3.66, in case of an NKBO the weight μ can play the part of the ν of this Proposition. By B we denote the maximum of $\{\mu(g) : g \in \Sigma\}$. For all $s \in \mathcal{T}(\Sigma)$ we get $\mu(s) \leqslant B \cdot |s|$ by (3.7), while (3.1) includes $\mathrm{dp}(s) \leqslant |s|$. With A and D chosen as in the proof of Theorem 5.15, for any $C \geqslant AB, 3D$ we can bring the crops in and see

$$\begin{aligned}
\mathrm{dl}_{\mathcal{R}}(s) &< \mathrm{Ack}(A\mu(s) + 4, 3D(\mathrm{dp}(s) + 1)) && \text{by (5.4)}\\
&\leqslant \mathrm{Ack}(AB(|s| + 1), 3D(|s| + 1))\\
&\leqslant \mathrm{Ack}(C(|s| + 1), C(|s| + 1))\\
&\leqslant \mathrm{Ack}(C|s| + C + 2, 0) && \text{by Lemma 2.63.vii.}
\end{aligned}$$

The optimality of this bound is established as in Lemma 5.9. □

Our next Theorem is the promised improvement of Corollary 4.7.

Theorem 5.17. *Termination via KBO implies ω^2-termination. If no special symbol is involved, then we even get ω-termination.*

Proof. Since the second half is a part of Corollary 4.7, it suffices to establish ω^2-termination. Let $\mathcal{R}$ be a TRS over a signature Σ which terminates via KBO. By Theorem 3.61, $\mathcal{R}$ terminates via some NKBO $\prec_{\mathsf{kbo}}$ with weight μ and (total) precedence $\prec$. Because extending the signature does not hurt, we may again suppose there is a special symbol i and Σ is as in (5.1). Furthermore we may assume $\Sigma^{(\geqslant 1)}$ contains a symbol which is not special. This is needed to get an infinitude of distinct weights.

Let $(P, \lhd)$ be the restriction of the well-order $(\mathbb{N} \times \mathbb{N}, <_{\mathsf{lex}})$ to

$$\{(\mu(s), \mathcal{I}(s)) : s \in \mathcal{T}(\Sigma)\} ,$$

where $\mathcal{I}$ is the interpretation from Theorem 5.15. Recall from Proposition 3.66 that the ν used in Theorem 5.15 is just the weight μ, as we treat an NKBO. We have $\mathrm{otype}(P, \lhd) = \omega^2$ because there are infinitely many distinct weights and because Lemma 5.13.i tells us that, for each term $s \in \mathcal{T}(\Sigma)$, the set

$$\{(\mu(i^a s), \mathcal{I}(i^a s)) : a \in \mathbb{N}\} = \{(\mu(s), \mathcal{I}(i^a s)) : a \in \mathbb{N}\}$$

is infinite. We are about to introduce the Σ-algebra $(\mathcal{P}, \lhd, \mathcal{F})$. For $f_j \in \Sigma^{(m)}$ and $p_1, \ldots, p_m \in \mathcal{P}$ with $p_l = (b_l, c_l)$ we put $b' := \mu(f_j) + b_1 + \cdots + b_m$ and

$$[f_j](\bar{p}) := (b', \mathcal{A}_{Ab'}((1, j, c_1, \ldots, c_{m-1}), \sum_{k=1}^{m} c_k)) \; .$$

The special symbol is handled by

$$[i]((b, c)) := (b, \mathcal{A}_{Ab}(1, c)) \; .$$

This Σ-algebra is monotone. Indeed, if $p \rhd p'$ because of the first component, then monotonicity is obvious, and otherwise the monotonicity of $\mathcal{A}$, which we established in Lemma 5.11.ii, comes into play.

An induction on $s \in \mathcal{T}(\Sigma)$, which partially relies on Lemma 5.11.iv, shows $[\![s]\!] = (\mu(s), \mathcal{I}(s))$. This can be generalized. For any $\rho\colon \mathcal{V} \to \mathcal{P}$ there is, by definition of $\mathcal{P}$, a ground substitution σ satisfying $\rho(x) = (\mu(x\sigma), \mathcal{I}(x\sigma))$. An induction on $s \in \mathcal{T}(\Sigma, \mathcal{V})$ establishes

$$[\![s, \rho]\!] = (\mu(s\sigma), \mathcal{I}(s\sigma)) \; . \tag{5.5}$$

We intend to show that $\mathcal{R}$ is compatible with the Σ-algebra. Pick (l, r) from $\mathcal{R}$ and $\rho\colon \mathcal{V} \to \mathcal{P}$ with an associated σ as in (5.5). From $l \succ_{\mathsf{kbo}} r$ we infer $\mu(l\sigma) \geqslant \mu(r\sigma)$ and, via Lemma 5.14, $\mathcal{I}(l\sigma) > \mathcal{I}(r\sigma)$. Hence we get, by another invocation of (5.5),

$$[\![l, \rho]\!] = (\mu(l\sigma), \mathcal{I}(l\sigma)) \geqslant_{\mathsf{lex}} (\mu(r\sigma), \mathcal{I}(l\sigma)) >_{\mathsf{lex}} (\mu(r\sigma), \mathcal{I}(r\sigma)) = [\![r, \rho]\!] \; ,$$

and thus $[\![l, \rho]\!] \rhd [\![r, \rho]\!]$. We had to take a digression to $(\mathbb{N} \times \mathbb{N}, <_{\mathsf{lex}})$ because $(\mu(r\sigma), \mathcal{I}(l\sigma))$ may not be an element of $\mathcal{P}$. □

Recall from Lemma 3.63 that the SRS containing $fg \to gff$, $f \to \varepsilon$, and $g \to \varepsilon$ terminates via KBO, is ω^2-terminating, not $<\omega^2$-terminating, and also not weakly ω-terminating. In a certain sense this is a most complex example, as termination via KBO implies ω^2-termination.

Theorem 5.18. *Termination via either MPO, LPO, or KBO implies ω^2-termination but not $<\omega^2$-termination.*

Proof. We just treated termination via KBO. Termination via either MPO or LPO implies, by Theorems 5.6 and 5.7, ω-termination. Via Theorem 3.44.i and $\omega \cdot \omega = \omega^2$ this yields ω^2-termination. □

The last Theorem clarifies that the usual syntactic simplification orders are located at the very beginning of the α-termination hierarchy. An obvious research task is to find natural simplification orders which do not suffer from such harsh limitations.

5.3 Simple Termination is Complex*

We already mentioned that the complexities of terminating SRSs are cofinal (with respect to $<_{\text{ed}}$) in the recursive functions. Our main intention in this section is to demonstrate that the complexities of simply terminating TRSs are, though considerably lower, still inconceivably large. Most of the results we will present rely on techniques developed in proof theory. For example, the construction behind the following result is based on a generalized Hardy hierarchy below ω_3.

Theorem 5.19 (Cichon and Tahhan Bittar 1998). *The complexity of any simply terminating SRS is dominated by a multiple recursive function.*

With the forerunner Rathjen and Weiermann (1993), the whole of simple termination was covered by Weiermann. The following Theorem is only a part of a more general result, see Theorem 5.54.

Theorem 5.20 (Weiermann 1994). *The complexity of any simply terminating TRS is dominated by a* $<\vartheta(\Omega^\omega)$*-recursive function.*

For quite a while, Hofbauer's SRSs which are related to branches of the Ackermann function (see Proposition 5.5) and terminate via MPO supplied the largest known complexities of simply terminating SRSs. Their complexities exhaust but remain inside the primitive recursive functions.

The situation for terms was similar. Here the largest known complexities were those of TRSs terminating via LPO which compute the k-ary Ackermann functions. By Theorem 2.66.ii, these functions exhaust but remain inside the multiple recursive functions. Now on the one hand $\vartheta(\Omega^\omega)$ is the supremum of both the order types of simplification orders and the order types of LPOs (see Theorem 4.10 and Theorem 4.1), and on the other hand we have (2.4) on page 37 telling us that, under certain additional assumptions, $\textsc{Mrec} \approx_{\text{ed}} (\mathrm{G}_\beta)_{\beta<\vartheta(\Omega^\omega)}$ holds with the G_β being slow growing functions. This led to the formulation of the *slow growing principle*, implicit in Cichon (1992), stating that a termination proof by a simplification order of order type α imposes as an upper complexity bound some G_β with $\beta < \vartheta(\Omega^\omega)$ closely related to α. A

[*] This title is inspired by Middeldorp and Gramlich (1993).

consequence of this is the existence of multiple recursive complexity bounds for all simply terminating TRSs. It was further conjectured that the complexities of all simply terminating SRSs have primitive recursive bounds. However, in both cases Touzet constructed counterexamples.

Touzet (1997) transformed the TRS Hofbauer and Lautemann used to show that termination via KBO leads to complexities beyond primitive recursion (see Theorem 5.8.i and Theorem 6.13) into a simply terminating SRS.

Theorem 5.21 (Touzet 1997, 5.3, Touzet 1998a). *There is a simply (yet not totally) terminating SRS whose complexity is not primitive recursive.*

This result has been improved later, see Theorem 5.23. Recall from Theorem 2.83 that the multiple recursive functions and the $<\omega_3$-recursive functions coincide. The conjectured multiple recursive complexity bounds linked with simple termination were pulverized by Touzet.

Theorem 5.22 (Touzet 1998b). *For any $n \in \mathbb{N}$ there is a totally terminating TRS whose complexity is not $<\omega_n$-recursive.*

The TRS used for $n+1$ can be identified with a proper extension (concerning both symbols and rules) of the one used for n. As the TRSs have to be finite, it is still not possible to leave the $<\varepsilon_0$-recursive functions with this construction. We will extend Touzet's approach in the following subsections and show that the bound of Theorem 5.20 is optimal.

Touzet established the optimality of the upper complexity bound for simply terminating SRSs described in Theorem 5.19. This classifies the strength of Higman's Lemma.

Theorem 5.23 (Touzet 1999). *For any multiple recursive function f there is a totally terminating SRS whose complexity eventually dominates f.*

The construction behind both Theorem 5.22 and Theorem 5.23 is twofold. In a first step the ordinals below ε_0 and ω_3, resp., are encoded by terms. The second step provides rewrite rules which simulate, for each encoded ordinal α, a process called the *battle of Hercules and the Hydra*, which first appeared in Kirby and Paris (1982). This corresponds to computing the Hardy functions[†] H_α. It then suffices to give a proof of total termination in order to show complexities similar to those of the corresponding Hardy functions are attainable. For example, Theorem 5.23 is established by encoding (larger and larger subsets of) the ordinals below ω_3. By Theorem 2.83, the Hardy functions below ω_3 and the multiple recursive functions match up. Note that here ω_3 necessarily has to

[†] More precisely, the closely related counting function L_α is mimicked.

be approached from below by larger and larger subsets (and larger and larger TRSs), because a simulation of the battle for all ordinals below ω_3 at once leads to a complexity similar to H_{ω_3}. This function grows too fast to be multiple recursive, see Lemma 2.78.iii.

For TRSs Touzet (1999) conjectured it is possible to extend this approach to all ordinals below the small Veblen number $\vartheta(\Omega^\omega)$, showing Weiermann's huge upper bound of Theorem 5.20 is essentially optimal. The validation of this conjecture will be our main concern in the next subsections.

We used the function ϑ of Definition 2.46 mainly to pin down $\vartheta(\Omega^\omega)$. Since ϑ is very powerful and, due to its definition using involved closure processes, not easy to handle by TRSs, it seems advisable for our purposes to use k-ary Veblen functions φ and their fixed point free variants ψ instead of ϑ. The small Veblen number $\vartheta(\Omega^\omega)$ is the limit (for k approaching ω) of the ordinals that can be denoted by the k-ary ψ augmented by 0 and (binary) addition $+$. We prefer ψ to φ since the absence of fixed points significantly simplifies calculations.

The canonical approach consists in defining TRSs $\mathcal{R}_k$ which simulate Hydra battles for all ordinals below $\psi(1,0,\dots,0)$ where ψ is $k+1$-ary. This is done in Section 5.3.4, where we use an encoding for the ordinals below $\psi(1,0,\dots,0)$ introduced in Section 5.3.3. Total termination of $\mathcal{R}_k$ is then established in Section 5.3.5 using Touzet's technically smooth characterization of total termination from Theorem 3.45. The $\mathcal{R}_k$ are given in a uniform manner, and for $k > l$ the TRS $\mathcal{R}_k$ can be regarded as a proper extension of $\mathcal{R}_l$. Thus the $\mathcal{R}_k$ constitute a hierarchy of totally terminating TRSs, and the complexity of any simply terminating TRS is eventually dominated by the complexities of almost all $\mathcal{R}_k$. Just as for SRSs, this stepwise approach from below is inevitable as it is not possible to define a simply terminating TRS which is able to simulate Hydra battles for all ordinals below $\vartheta(\Omega^\omega)$.

5.3.1 The Fixed Point Free Veblen Function

We are going to introduce a part of the *Veblen (1908) function* φ and its fixed point free variant ψ. This part suffices to reach $\vartheta(\Omega^\omega)$.

For $\alpha_1,\dots,\alpha_k \in \mathsf{On}$ with $k > 0$ we intend to recursively define the branch $\varphi_{\bar\alpha}\colon \mathsf{On} \to \mathsf{On}$ of the *Veblen function.* It is advisable to interchangeably use $\varphi_{\bar\alpha}(\beta)$ and $\varphi(\bar\alpha,\beta)$, thus regarding φ as a function from the ordinal sequences of lengths larger than 1 into the ordinals. The principal ordinals H are enumerated by $\varphi_{\bar 0}$. If $\alpha_k > 0$, then $\varphi_{\bar\alpha}$ is the enumerating function of the proper class

$$\{\beta : (\forall\gamma < \alpha_k)(\varphi(\alpha_1,\dots,\alpha_{k-1},\gamma,\beta) = \beta)\},$$

and otherwise we have $(\alpha_1,\dots,\alpha_k) = (\alpha_1,\dots,\alpha_i,\bar 0,0)$ with $\alpha_i > 0$. Here we let

$\varphi_{\bar{\alpha}}$ be the enumerating function of the proper class

$$\{\beta : (\forall \gamma < \alpha_i)(\varphi(\alpha_1, \dots, \alpha_{i-1}, \gamma, \beta, \bar{0}, 0) = \beta)\}.$$

Obviously $\varphi_{\bar{0},\bar{\alpha}} = \varphi_{\bar{\alpha}}$ holds, and by definition $\varphi_{\bar{0},1}$ enumerates the epsilons. The φ function lacks the subterm property since it admits fixed points. Therefore we concentrate on ψ, the *fixed point free* version of φ. We let $\psi(\alpha_1, \dots, \alpha_k, \beta)$ be $\varphi(\bar{\alpha}, \beta + 1)$ if $\beta = \beta_0 + n$ for some $n \in \mathbb{N}$ and $\beta_0 \in \mathsf{Lim} \cup \{0\}$ with $\varphi(\bar{\alpha}, \beta_0) \in \{\alpha_1, \dots, \alpha_k, \beta_0\}$, and otherwise $\psi(\bar{\alpha}, \beta)$ is just $\varphi(\bar{\alpha}, \beta)$.

For almost all of the remainder of this section we keep some $k > 0$ fixed and focus on the $k + 1$-ary ψ.

By Δ_k we denote the first infinite ordinal closed under $+$ and the $k+1$-ary ψ:

$$\Delta_k := \varphi(1, \underbrace{0, \dots, 0}_{k+1 \text{ times}}) \,. \tag{5.6}$$

Thus Δ_2 coincides with the ordinal Γ_0 celebrated by Gallier (1991). The connection between ψ and ϑ was illuminated by Schmidt (1979). All weed need to know about this here is where we can find the small Veblen number.

Theorem 5.24 (Schmidt 1979). $\vartheta(\Omega^\omega) = \sup\{\Delta_k : k \in \mathbb{N}\}$.

By Proposition 2.29, for every ordinal $\alpha > 0$ there are uniquely determined principal ordinals $\alpha_0 \geqslant \dots \geqslant \alpha_n$ such that $\alpha = \alpha_0 + \dots + \alpha_n$ holds. In addition, for every principal $\alpha < \Delta_k$ there are uniquely determined $\alpha_1, \dots, \alpha_{k+1}$ below α satisfying $\alpha = \psi(\bar{\alpha})$, cf. Buchholz (1993). So every $\alpha < \Delta_k$ can be associated with a unique representation solely built up from 0, $+$ and the $k+1$-ary ψ. We call this the *k-normal form* of α.

The next Lemma lists some of the basic properties of ψ. Recall from Definition 2.37 that the lexicographic order of ordinal tuples having the same length is denoted by $<_{\mathsf{lex}}$.

Lemma 5.25. *Let $\alpha_1, \dots, \alpha_{k+1}$ and $\gamma_1, \dots, \gamma_{k+1}$ be given.*

i. *Each $\psi(\bar{\alpha})$ is a principal ordinal and, except for $\psi(\bar{0}) = 1$, a limit ordinal.*
ii. *The function ψ has the subterm property and is monotone.*
iii. *$\psi(\bar{\alpha}) > \psi(\bar{\gamma})$ is equivalent to*

$$\big((\bar{\alpha}) >_{\mathsf{lex}} (\bar{\gamma}) \wedge \psi(\bar{\alpha}) > \gamma_1, \dots, \gamma_{k+1}\big) \vee (\exists i \in [1, k+1])(\alpha_i \geqslant \psi(\bar{\gamma})) \,.$$

Recall from (2.1) the notion $f(\bar{x}, \cdot, \bar{y})^n(z)$ of the n^{th} iteration of the unary function $v \mapsto f(\bar{x}, v, \bar{y})$ on z. The following Lemma contains all properties of ψ we will rely on later.

Lemma 5.26. *Let $i \in [1,k]$; $m,n \in \mathbb{N}$, and ordinals $\alpha_1,\dots,\alpha_{k+1}$, $\gamma_1,\dots,\gamma_{k+1}$, δ,δ' be given.*

i. *If $n > m$ and $\alpha_j \neq 0$ for some j, then $\psi(\bar{\alpha}) \cdot n > \psi(\bar{\alpha}) \cdot m$.*

ii. *If $\psi(\bar{\alpha}) > \psi(\bar{\gamma})$, then $\psi(\bar{\alpha}) > \psi(\bar{\gamma}) \cdot n + m$.*

iii. *If $n \geqslant m$; $\delta \geqslant \delta'$, and at least one of the inequalities is proper, then*

$$\psi(\alpha_1,\dots,\alpha_i,\cdot,\alpha_{i+1},\dots,\alpha_k)^n(\delta) > \psi(\alpha_1,\dots,\alpha_i,\cdot,\alpha_{i+1},\dots,\alpha_k)^m(\delta') \ .$$

iv. *If $(\gamma_1,\dots,\gamma_i) >_{\mathsf{lex}} (\alpha_1,\dots,\alpha_i)$ and $\psi(\bar{\gamma}) > \alpha_1,\dots,\alpha_k,\delta$, then*

$$\psi(\bar{\gamma}) > \psi(\alpha_1,\dots,\alpha_i,\cdot,\alpha_{i+1},\dots,\alpha_k)^n(\delta) \ .$$

Proof. The first point follows from $\psi(\bar{\alpha}) > 1$. Under the conditions of (ii), $\psi(\bar{\alpha})$ is a principal limit ordinal. For (iii) we utilize the subterm property and the monotonicity of ψ, respectively. Finally, (iv) is established by induction on n using Lemma 5.25.iii. □

5.3.2 Fundamental Sequences and Hydrae below Δ_k

Fundamental sequences and Bachmann systems have been introduced and studied in Section 2.4. We are going to introduce a Bachmann system for Δ_k, but before we can do so, we have to wade through some technical definitions.

Definition 5.27. Let $\alpha_1,\dots,\alpha_k < \Delta_k$. The set of *fixed points* of $\varphi_{\bar{\alpha}}$ is

$$\mathsf{Fix}(\bar{\alpha}) := \{\psi(\bar{\gamma},\delta) : (\bar{\gamma}) >_{\mathsf{lex}} (\bar{\alpha}) \wedge \psi(\bar{\gamma},\delta) > \alpha_1,\dots,\alpha_k\} \ .$$

For $\beta < \Delta_k$ we need the auxiliary notation

$$\psi(\bar{\alpha},\beta)^* := \begin{cases} \psi(\bar{\alpha},\beta_0) & \text{if } \beta = \beta_0 + 1 \ , \\ \beta & \text{otherwise,} \end{cases}$$

and for $\gamma < \Delta_k$ in k-normal form we are going to define the set $\mathrm{IS}_{\bar{\alpha}}(\gamma)$ of (relative to $\bar{\alpha}$) *interesting subterms* of γ by recursion. We start with $\mathrm{IS}_{\bar{\alpha}}(0) := \{0\}$ and $\mathrm{IS}_{\bar{\alpha}}(\gamma_1 + \cdots + \gamma_m) := \bigcup_{1 \leqslant i \leqslant m} \mathrm{IS}_{\bar{\alpha}}(\gamma_i)$ for $m > 1$. The nontrivial case is given by

$$\mathrm{IS}_{\bar{\alpha}}(\psi(\bar{\gamma})) := \begin{cases} \{\psi(\bar{\gamma})\} & \text{if } (\gamma_1,\dots,\gamma_k) \geqslant_{\mathsf{lex}} (\alpha_1,\dots,\alpha_k) \ , \\ \bigcup\limits_{1 \leqslant i \leqslant k+1} \mathrm{IS}_{\bar{\alpha}}(\gamma_i) & \text{otherwise.} \end{cases}$$

Now the (relative to $\bar{\alpha}$) *maximal interesting subterm* $\mathrm{MS}_{\bar{\alpha}}(\bar{\delta})$ of a nonempty sequence $\bar{\delta}$ is defined to be the maximum of the ordinals occurring in the $\mathrm{IS}_{\bar{\alpha}}(\delta_i)$.

The following definition corresponds to the assignment of fundamental sequences for ordinals below Δ_k from Weiermann (1997a), which is based on work of Buchholz (1980).

Definition 5.28. We define $\alpha[n]$ by recursion on $\alpha < \Delta_k$ in k-normal form:

$$\begin{aligned}
0[n] &:= 0 \\
(\alpha_1 + \cdots + \alpha_m)[n] &:= \alpha_1 + \cdots + \alpha_{m-1} + \alpha_m[n] \qquad \text{if } m > 1 \\
\psi(\bar{0})[n] &:= 0 \\
\psi(\bar{\alpha}, \lambda)[n] &:= \psi(\bar{\alpha}, \lambda[n]) \qquad \text{if } \lambda \in \mathsf{Lim} \setminus \mathsf{Fix}(\bar{\alpha}) \\
\psi(\bar{0}, \beta)[n] &:= \alpha^* \cdot (n+1) \\
\psi(\alpha_1, \ldots, \alpha_i + 1, \bar{0}, \beta)[n] &:= \psi(\alpha_1, \ldots, \alpha_i, \cdot, \bar{0})^{n+1}(\alpha^*) \\
\psi(\alpha_1, \ldots, \alpha_i, \bar{0}, 0)[n] &:= \psi(\alpha_1, \ldots, \alpha_i[n], \bar{0}, \mathrm{MS}_{\bar{\alpha},\bar{0}}(\bar{\alpha})) \\
\psi(\alpha_1, \ldots, \alpha_i, \bar{0}, \beta)[n] &:= \psi(\alpha_1, \ldots, \alpha_i[n], \bar{0}, \alpha^*) \; .
\end{aligned}$$

It should be clear that $\alpha_i \in \mathsf{Lim}$ is demanded in the last two lines. For $m \in \mathbb{N}$ we further introduce

$$\alpha[n,m] := \begin{cases} \alpha & \text{if } n > m \; , \\ (\alpha[n, m-1])[m] & \text{otherwise.} \end{cases} \tag{5.7}$$

Thus $\alpha[n,m] = (\ldots((\alpha[n])[n+1])\ldots)[m]$. The main results concerning subrecursive hierarchies in Section 2.4 are related to the tame Bachmann systems of Definition 2.81. We are pleased to see the assignment of fundamental sequences defined above is tame. Of central importance is the use of $\mathrm{MS}_{\bar{\alpha},\bar{0}}(\bar{\alpha})$ in the penultimate line above. Replacing the ordinal with 0 would spoil convergence. A proof of the following result can be extracted from Weiermann (1997a).

Theorem 5.29. $(\Delta_k, \cdot[\cdot])$ *is a tame Bachmann system.*

The *battle of Hercules and the Hydra* from Kirby and Paris (1982) is closely connected to the Hardy functions. This is emphasized by the fact that *Hardy* is an anagram for *Hydra*.

Definition 5.30. A *Hydra* is an ordinal below Δ_k. For each Hydra α the ordered pair $c := (\alpha, n)$ is called a *configuration*. The *next configuration* c^+ for c is $(\alpha[n], n+1)$, and the *Hydra battle* for the configuration c is the sequence $(c_m)_{m<\omega}$ of configurations with $c_0 = c$ and $c_{m+1} = c_m^+$. The minimal m such that the Hydra in c_m is 0 is called the *length* of the battle.

An immediate consequence of Theorem 5.29 is $(\forall \alpha > 0)(\forall n)(\alpha > \alpha[n])$, thus the length of the battle is well-defined for each configuration. The following Lemma gathers a few basic facts about Hydra battles, and it links the length of a battle to the counting functions L_α of Definition 2.75.

Lemma 5.31. *Let* $c := (\alpha, n)$ *be a configuration and consider its battle* $(c_m)_{m \in \mathbb{N}}$.

i. *For all* m *we have* $c_m = (\alpha[n, n+m-1], n+m)$.

ii. *The length of the battle is the minimal* m *such that* $\alpha[n, n+m-1] = 0$.

iii. *The length of the battle is* $\mathrm{L}_\alpha(n)$.

Proof. We can show (i) by an induction on m, and (ii) is an easy consequence of this. An induction on α establishes (iii). □

The counting functions are tailored exactly for the Hydra battle, and hence the Hardy functions are also closely related to the battle.

Proposition 5.32. *If there is an* α_0 *such that* $\alpha = \omega^\omega \cdot \alpha_0 < \Delta_k$, *then the function which maps* n *to the length of the battle for the configuration* (α, n) *eventually dominates all* $<\!\alpha$*-recursive functions.*

Proof. We combine Lemma 2.77 and Lemma 5.31.iii to see that the length of the battle for (α, n) is $\mathrm{H}_\alpha(n) - n$. By Theorem 5.29, we deal with a tame Bachmann system, hence we may incorporate Theorem 2.82 to see that any $<\!\alpha$-recursive function is eventually dominated by some H_β with $\beta < \alpha$. Taking a look at Lemma 2.78.iii we know that H_α eventually dominates H_β. Thus the claim is established. □

5.3.3 Encoding all Hydrae below Δ_k

We intend to encode all Hydrae below Δ_k by terms. For these terms formal fundamental sequences will be defined.

Definition 5.33. The signature Σ_0 consists of

- the constant $\mathsf{0}$,
- the unary (successor) S,
- the binary $+$, and
- the $k+1$-ary P, which represents ψ.

Each $s \in \mathcal{T}(\Sigma_0)$ has a *value* $\mathrm{val}(s) < \Delta_k$, which is calculated by interpreting $\mathsf{0}$, S, $+$, P with 0, the successor function, the ordinal sum, and ψ, respectively.

The symbol S is not really needed for encoding Hydrae since the terms $\mathsf{S}s$ and $+(s, \mathsf{P}(\bar{\mathsf{0}}))$ have the same value. We use S as a syntactic indicator of successor ordinals, and additionally its presence will simplify various calculations.

Obviously, each ordinal below Δ_k can be denoted by terms of $\mathcal{T}(\Sigma_0)$. To mimic fundamental sequences on terms we introduce a set of *standard terms*. Because our $+$ has fixed arity and because we do not want to bother about distinguishing between $+(+(s,t),u)$ and $+(s,+(t,u))$, there will usually be distinct standard terms denoting the same ordinal. Furthermore, for $\gamma < \Delta_k$ denoted by the standard term s and for $n > 0$ we intend to denote $\gamma + n$ by $\mathsf{S}^n s$, thus using $+$ only for certain additions of (standard terms for) limit ordinals. This will be useful later when we treat derivations of standard terms.

Recall from Definition 2.31 that $\oplus$ denotes the natural sum of two ordinals. A pair (λ, μ) of limit ordinals is called *compatible* if $\lambda + \mu = \lambda \oplus \mu$ holds.

Definition 5.34. The set $\mathcal{D} \subseteq \mathcal{T}(\Sigma_0)$ of *standard terms* is the smallest superset of $\{\mathsf{0}\}$ which is closed under S and these rules:

- $\bar{s} \in \mathcal{D}$ and $\bar{s} \neq \bar{\mathsf{0}} \implies \mathsf{P}(\bar{s}) \in \mathcal{D}$,
- $s, t \in \mathcal{D}$ and $(\mathrm{val}(s), \mathrm{val}(t))$ compatible $\implies +(s,t) \in \mathcal{D}$.

By $\mathcal{D}(\alpha)$ we denote the collection of standard terms with value α.

The following Lemma should be no surprise.

Lemma 5.35.

i. *If s' is a proper subterm of $s \in \mathcal{D}(\alpha)$, then there is $\beta < \alpha$ with $s' \in \mathcal{D}(\beta)$.*
ii. *For all $\alpha < \Delta_k$ we have $\mathcal{D}(\alpha) \neq \emptyset$.*
iii. *$\mathcal{D}$ is the union of the $\mathcal{D}(\alpha)$ with $\alpha < \Delta_k$.*

Definition 5.36. A formal *multiplication* for a term s and $n > 0$ is defined by

$$s \times n := +(\cdot, s)^{n-1}(s) .$$

It will be important that the recursion occurs in the first argument.

Our aim is now to mimic the definition of $\alpha[n]$ for the members of $\mathcal{D}$. We encounter some difficulties on the way. Though in Definition 5.28 the ordinals on the left are supposed to be in k-normal form, the ordinals on the right are not always in k-normal form. In the same way, a formal equivalent to fundamental sequences for standard terms will not always produce standard terms. For example, if we defined $\mathsf{P}(\bar{\mathsf{0}}, \mathsf{S}s)[n]$ to be $\mathsf{P}(\bar{\mathsf{0}}, s) \times (n+1)$, this would result in occurrences of the nonstandard term $\mathsf{P}(\bar{\mathsf{0}})$ for $s = \mathsf{0}$. We overcome this obstacle for $d \in \mathcal{D}(\alpha)$ by simultaneously defining $d\langle n\rangle$ and $d[n]$, where $d\langle n\rangle$ is a formal equivalent to $\alpha[n]$ which need not be a standard term but has a uniform definition, while $d[n]$ is a refinement of $d\langle n\rangle$ and an element of $\mathcal{D}(\alpha[n])$. Later we will work in a TRS which is able to reduce $d\langle n\rangle$ to $d[n]$.

Definition 5.37. By $\mathcal{D}(\mathsf{Lim})$ we denote the set of standard terms whose values are limit ordinals, while the analogies to $\mathsf{Fix}(\bar{\alpha})$ and $\mathrm{MS}_{\bar{\alpha}}(\bar{\beta})$ on $\mathcal{D}$ are called $\mathsf{Fix}(\bar{s})$ and $\mathrm{MS}_{\bar{s}}(\bar{t})$.

For the sake of transparency those cases of the definition of $\alpha[n]$ which involve the notation α^* are split up. They will be handled by distinct rewrite rules.

Definition 5.38. For $d \in \mathcal{D}$ and $n \in \mathbb{N}$ we simultaneously define $d\langle n\rangle$ and $d[n]$, both members of $\mathcal{T}(\Sigma_0)$, by recursion on $\mathcal{D}$:

$$\begin{aligned}
\mathsf{0}\langle n\rangle &:= \mathsf{0} && (5.8a)\\
\mathsf{S}s\langle n\rangle &:= s && (5.8b)\\
+(s,t)\langle n\rangle &:= +(s,t[n]) && (5.8c)\\
\mathsf{P}(\bar{s},t)\langle n\rangle &:= \mathsf{P}(\bar{s},t[n]) \qquad \text{if } t \in \mathcal{D}(\mathsf{Lim}) \setminus \mathsf{Fix}(\bar{s}) && (5.8d)\\
\mathsf{P}(\bar{\mathsf{0}},\mathsf{S}t)\langle n\rangle &:= \mathsf{P}(\bar{\mathsf{0}},t) \times (n+1) && (5.8e)\\
\mathsf{P}(\bar{\mathsf{0}},t)\langle n\rangle &:= t \times (n+1) && (5.8f)\\
\mathsf{P}(s_1,\dots,\mathsf{S}s_i,\bar{\mathsf{0}},\mathsf{S}t)\langle n\rangle &:= \mathsf{P}(\bar{s},\cdot,\bar{\mathsf{0}})^{n+1}(\mathsf{P}(s_1,\dots,\mathsf{S}s_i,\bar{\mathsf{0}},t)) && (5.8g)\\
\mathsf{P}(s_1,\dots,\mathsf{S}s_i,\bar{\mathsf{0}},t)\langle n\rangle &:= \mathsf{P}(\bar{s},\cdot,\bar{\mathsf{0}})^{n+1}(t) && (5.8h)\\
\mathsf{P}(s_1,\dots,s_i,\bar{\mathsf{0}},\mathsf{0})\langle n\rangle &:= \mathsf{P}(s_1,\dots,s_i[n],\bar{\mathsf{0}},\mathrm{MS}_{\bar{s},\bar{\mathsf{0}}}(\bar{s})) && (5.8i)\\
\mathsf{P}(s_1,\dots,s_i,\bar{\mathsf{0}},\mathsf{S}t)\langle n\rangle &:= \mathsf{P}(s_1,\dots,s_i[n],\bar{\mathsf{0}},\mathsf{P}(s_1,\dots,s_i,\bar{\mathsf{0}},t)) && (5.8j)\\
\mathsf{P}(s_1,\dots,s_i,\bar{\mathsf{0}},t)\langle n\rangle &:= \mathsf{P}(s_1,\dots,s_i[n],\bar{\mathsf{0}},t)\ . && (5.8k)
\end{aligned}$$

Similar to Definition 5.28, $s_i \neq \mathsf{0}$ is required for (5.8i)–(5.8k).

If $d = +(s,t)$ and $t[n] = \mathsf{S}^i t'$ where i is as large as possible, we put

$$+(s,t)[n] := \begin{cases} \mathsf{S}^i s & \text{if } t' = \mathsf{0}\ , \\ \mathsf{S}^i{+}(s,t') & \text{otherwise.} \end{cases}$$

Moreover, we demand $\mathsf{P}(\bar{\mathsf{0}},\mathsf{S0})[n] := \mathsf{S}^{n+1}\mathsf{0}$ as well as

$$\mathsf{P}(\mathsf{0},\dots,\mathsf{S0},\bar{\mathsf{0}},\mathsf{0})[n] := \mathsf{P}(\mathsf{0},\dots,\mathsf{0},\cdot,\bar{\mathsf{0}})^n(\mathsf{S0})\ ,$$

and in all remaining cases we put $d[n] := d\langle n\rangle$. For $m \in \mathbb{N}$ we further introduce $d[n,m]$ in analogy with the $\alpha[n,m]$ of (5.7) in Definition 5.28.

We now show that this definition is correct and meets our requirements.

Lemma 5.39. *Let $\alpha < \Delta_k$ and $n \in \mathbb{N}$. For $d \in \mathcal{D}(\alpha)$ we have* $\mathrm{val}(d\langle n\rangle) = \alpha[n]$ *and $d[n] \in \mathcal{D}(\alpha[n])$.*

Proof by induction on $\mathcal{D}$. As the definition of $d\langle n\rangle$ just copies Definition 5.28, the first statement is immediate from the induction hypothesis. Because in the above definition recursion is only used for standard terms s and t denoting limit ordinals, we have $s[n] \neq \mathsf{0}$ and $t[n] \neq \mathsf{0}$ according to the induction hypothesis and Theorem 5.29. Since $\mathrm{MS}_{\bar{s},\bar{\mathsf{0}}}(\bar{s})$, being a subterm of some $s_j \in \mathcal{D}$, cannot be $\mathsf{P}(\bar{\mathsf{0}})$, the only possible occurrences of $\mathsf{P}(\bar{\mathsf{0}})$ in $d\langle n\rangle$ are the ones we gave special treatment in the definition of $d[n]$. In both cases it is obvious that $d\langle n\rangle$ and $d[n]$ have the same value and that $d[n]$ is standard.

Now let $d = +(s,t)$ and let i, t' be as in the definition of $d[n]$. Since d is standard, we know the pair (τ,μ) with $\tau := \mathrm{val}(s)$ and $\mu := \mathrm{val}(t)$ is compatible. The statement obviously holds if $t' = \mathsf{0}$. So let t' denote a limit ordinal, say μ'. By Definition 5.34 we have to show (τ,μ') is compatible. This is done by proving $\mu' \leqslant \mu$. The induction hypothesis yields $t[n]$ is a standard term with $\mathrm{val}(t[n]) = \mu[n]$. Because μ is a limit, Theorem 5.29 implies $\mu > \mu[n]$, thus $\mu' \leqslant \mu[n] < \mu$. So $d[n]$ is standard and has the correct value. In the remaining cases the statement easily follows from the induction hypothesis. □

5.3.4 Simulating all Hydra Battles below Δ_k

We are now prepared to gradually define the TRS $\mathcal{R}$, which is intended to simulate all Hydra battles below Δ_k. Therefore Σ_0 has to be enlarged by new symbols whose meaning will be elucidated in the following definitions.

Definition 5.40. The signature Σ consists of Σ_0 enriched by

- the unary $\bullet$, $\circ$, and $[]$,
- the $k+1$-ary M,
- the $i+1$-ary J_i, for $1 \leqslant i \leqslant k$,
- the $i+1$-ary Q_{ij}, for $1 \leqslant j \leqslant i \leqslant k$, and
- the $i+2$-ary R_i, for $1 \leqslant i \leqslant k$.

It will sometimes be necessary to reduce a term to one of its subterms. Since $\mathcal{R}$ is intended to be totally and thus simply terminating, we may introduce, for all symbols $f \in \Sigma^{(n)}$ with $n > 0$ and for $1 \leqslant i \leqslant n$, embedding rules

$$(\mathrm{S}_i f) \quad f(x_1,\dots,x_n) \to x_i \ .$$

Because $\mathsf{0}$ is the only constant, the following result is blatantly trivial, even though we did not yet define the whole of $\mathcal{R}$.

Lemma 5.41. *$\mathcal{R}$ is confluent, and each $s \in \mathcal{T}(\Sigma)$ reduces in less than* $\mathrm{dp}(s)$ *steps to its unique normal form* $\mathsf{0}$. *If s' is a subterm of s, then $s \xrightarrow{*} s'$.*

The promised rules which enable us to reduce $d\langle n\rangle$ to $d[n]$ are

$$(\mathrm{F1}) \quad \mathsf{P}(\bar{\mathsf{0}}) \to \mathsf{S0}\ , \qquad (\mathrm{F2}) \quad +(x,\mathsf{S}y) \to \mathsf{S}{+}(x,y)\ .$$

Lemma 5.42. *For $d \in \mathcal{D}$ we have $d\langle n\rangle \xrightarrow{*} d[n]$.*

Proof. The difference between $d\langle n\rangle$ and $d[n]$ for $d = \mathsf{P}(\bar{\mathsf{0}},\mathsf{S0},\bar{\mathsf{0}},\mathsf{0})$ consists of one single $\mathsf{P}(\bar{\mathsf{0}})$ which is replaced with $\mathsf{S0}$. This can be handled by (F1). For $d = \mathsf{P}(\bar{\mathsf{0}},\mathsf{S0})$ and $n > 0$ we have to show $\mathsf{P}(\bar{\mathsf{0}}) \times n \xrightarrow{+} \mathsf{S}^n\mathsf{0}$, which is done by induction on n. The case $n = 1$ is again established by (F1), while the induction hypothesis yields $\mathsf{P}(\bar{\mathsf{0}}) \times (n+1) \xrightarrow{+} +(\mathsf{S}^n\mathsf{0}, \mathsf{P}(\bar{\mathsf{0}}))$. Now we get

$$+(\mathsf{S}^n\mathsf{0}, \mathsf{P}(\bar{\mathsf{0}})) \to_{\mathrm{F1}} +(\mathsf{S}^n\mathsf{0}, \mathsf{S0}) \to_{\mathrm{F2}} \mathsf{S}{+}(\mathsf{S}^n\mathsf{0}, \mathsf{0}) \to_{\mathrm{S_1+}} \mathsf{S}^{n+1}\mathsf{0}\ .$$

It suffices for the remaining case $d = +(s,t)$ to note that

$$+(s, \mathsf{S}^i t') \xrightarrow{*}_{\mathrm{F2}} \mathsf{S}^i{+}(s,t') \to_{\mathrm{S_1+}} \mathsf{S}^i s$$

is possible for arbitrary i and t'. □

Following Touzet (1998b), $\mathcal{R}$ is to regard $\bullet[]^{n+1}d$ with $d \in \mathcal{D}$ as a term which encodes the battle configuration $(\mathrm{val}(d), n)$. Since we want to simulate Hydra battles at full length we intend, for $d \neq \mathsf{0}$, to make possible derivations $\bullet[]^{n+1}d \xrightarrow{+} \bullet[]^{n+2}d[n]$, which can then be iterated until $\bullet[]^{n+m}\mathsf{0}$ is reached.

For some calculations it will be necessary to facilitate $\bullet[]^{n+1}d \xrightarrow{+} []^{n+1}\bullet^{n+1}d$, so that $\bullet^{n+1}$ may be moved to the top of subterms of d as material which can be deleted in subderivations. When we reach a point where d can safely be modified into something close to $d\langle n\rangle$, we do so and put a $\circ$ on top of the new subterm. This $\circ$ will enable us to create $\bullet[]$ in front of $[]^{n+1}d[n]$, furthermore, recursions like the one needed for (5.8c) can be simulated. The required rules are variations on rules of Touzet (1998b):

$$\begin{array}{llll} (\mathrm{N1}) & \bullet[]x \to []\bullet\bullet x\ , & (\mathrm{N2}) & []\circ x \to \circ[][]x\ , \\ (\mathrm{N3}) & \circ x \to []x\ , & (\mathrm{N4}) & []x \to \bullet x\ . \end{array}$$

Lemma 5.43. *For $n > 0$ and $s \in \mathcal{T}(\Sigma)$ we have*

i. $[]^n s \xrightarrow{+} \bullet^n s \xrightarrow{+} s$

ii. $\bullet[]^n s \xrightarrow{+} []^n\bullet^n s$

iii. $[]^n\circ s \xrightarrow{+} \bullet[]^{n+1}s.$

Proof. For (i) we rely on (N4) and ($\mathrm{S_1}\bullet$), while (ii) follows from (i) and $\bullet[]^n s \xrightarrow{+} []^n\bullet^{2^n}s$, which is shown by induction on n using

$$\bullet^m[]s \xrightarrow{*}_{\mathrm{N1}} []\bullet^{2m}s\ , \tag{5.9}$$

which in turn is shown by induction on $m \geqslant 0$ using (N1). We get (iii) from

$$[]^n\circ s \xrightarrow{+}_{\mathrm{N2}} \circ[]^{2n}s \xrightarrow{*} \circ[]^{n+1}s \to_{\mathrm{N3}} [][]^{n+1}s \to_{\mathrm{N4}} \bullet[]^{n+1}s\ .$$

Its first step is won like (5.9), and the second one relies on $2n \geqslant n+1$ and (i). □

To simulate cases like (5.8c) we have to import $\bullet$ and $[]$ into standard terms. For $f \in \{\mathsf{S}, +, \mathsf{P}\}$ with arity n and for $1 \leqslant i \leqslant n$ we thus introduce the rule

$$(\mathrm{D}_i f)\quad \bullet f(\bar{x}) \to f(x_1, \ldots, []x_i, \ldots, x_n)\ .$$

Lemma 5.44. *For $s, t, \bar{s} \in \mathcal{T}(\Sigma)$; $n \geqslant 0$, and $1 \leqslant i \leqslant k+1$ we have*

i. $\bullet^{n+1}{+}(s,t) \xrightarrow{+} {+}(s, \bullet^{n+1}t)$

ii. $\bullet^{n+1}\mathsf{P}(\bar{s}) \xrightarrow{*} \bullet\mathsf{P}(s_1, \dots, \bullet^{n}s_i, \dots, s_{k+1}) \xrightarrow{+} \mathsf{P}(s_1, \dots, \bullet^{n+1}s_i, \dots, s_{k+1})$

iii. $\bullet^{n+1}\mathsf{P}(\bar{s}) \xrightarrow{+} \mathsf{P}(s_1, \dots, \Box^{n+1}s_i, \dots, s_{k+1})$

iv. $\bullet^{n+1}\mathsf{P}(s_1, \dots, \mathsf{S}s_i, \dots, s_{k+1}) \xrightarrow{+} \mathsf{P}(s_1, \dots, \mathsf{S}\Box^{n+1}s_i, \dots, s_{k+1})$.

Proof. To settle (i), (D$_2$+) is applied $n+1$ times, and afterwards we rely on Lemma 5.43.i. With little changes, using (D$_i\mathsf{P}$) and (D$_1\mathsf{S}$) instead of (D$_2$+), the remaining points follow. □

As mentioned earlier, importing $\bullet^n$ or $\Box^n$ shall enable us to locally reduce until it is safe to create a $\circ$ on top of the subterm we treated. Sometimes such a $\circ$ has to be exported. This is achieved by these rules:

$$
\begin{array}{lrl}
(\mathrm{E}_2{+}) & {+}(x, \circ y) \to \circ{+}(x,y) & \\
(\mathrm{E}_i\mathsf{P}) & \mathsf{P}(x_1, \dots, \circ x_i, \dots, x_{k+1}) \to \circ\mathsf{P}(\bar{x}) & \text{for } 1 \leqslant i \leqslant k+1\ .
\end{array}
$$

In order to simulate (5.8e)–(5.8h) we need rewrite rules for a special kind of multiplication and for iterations of P. Since multiplication amounts to iterating $+$, the rules are very similar:

$$
\begin{array}{lrl}
(\mathrm{R}\mathsf{M}) & \mathsf{M}(\bar{x}, \Box y) \to {+}(\mathsf{M}(\bar{x}, y), \mathsf{P}(\bar{x}, y)) & \\
(\mathrm{R}\mathsf{J}_i) & \mathsf{J}_i(x_1, \dots, \Box x_i, y) \to \mathsf{P}(\bar{x}, \mathsf{J}_i(\bar{x}, y), \bar{0}) & \text{for } 1 \leqslant i \leqslant k\ .
\end{array}
$$

Lemma 5.45. *For $\bar{s}, t \in \mathcal{T}(\Sigma)$; $n > 0$, and $1 \leqslant i \leqslant k$ we have*

i. $\mathsf{M}(\bar{s}, \Box^n t) \xrightarrow{+} \mathsf{P}(\bar{s}, t) \times n$

ii. $\mathsf{J}_i(s_1, \dots, \Box^n s_i, t) \xrightarrow{+} \mathsf{P}(s_1, \dots, s_i, \cdot, \bar{0})^n(t)$.

Proof. As both statements are treated similarly by induction on n, we only prove (i) in detail. For the start we have

$$
\mathsf{M}(\bar{s}, \Box t) \to_{\mathrm{R}\mathsf{M}} {+}(\mathsf{M}(\bar{s}, t), \mathsf{P}(\bar{s}, t)) \to_{\mathrm{S}_2+} \mathsf{P}(\bar{s}, t)\ ,
$$

and the induction step is

$$
\begin{aligned}
\mathsf{M}(\bar{s}, \Box^{n+1}t) &\to_{\mathrm{R}\mathsf{M}} {+}(\mathsf{M}(\bar{s}, \Box^n t), \mathsf{P}(\bar{s}, \Box^n t)) \\
&\xrightarrow{+} \quad {+}(\mathsf{M}(\bar{s}, \Box^n t), \mathsf{P}(\bar{s}, t)) \\
&\xrightarrow{+} \quad {+}(\mathsf{P}(\bar{s}, t) \times n, \mathsf{P}(\bar{s}, t))\ ,
\end{aligned}
$$

where we used Lemma 5.43.i and the induction hypothesis for the last two steps. Statement (ii) relies on (RJ_i) and (S$_{i+1}\mathsf{J}_i$) instead of (RM) and (S$_2$+). □

We now present the rules intended to carry out the transformations prescribed by Definition 5.38. Because it is not easy to distinguish between cases like (5.8d) and (5.8k), Lemma 5.44 and the rules ($\mathrm{E}_i\mathsf{P}$) facilitate both possible transformations. Hence $\mathcal{R}$ is also able to simulate wrong battles, i.e. battles based on assignments of fundamental sequences which differ from our assignment. In the wrong cases the ordinals denoted are smaller, thus this shall pose no problem to our intended Σ-algebra. Likewise, transformations as the one needed in (5.8h) are made possible for arbitrary terms t, and the t in (5.8f) is supposed to be an element of $\mathsf{Fix}(\bar{\mathsf{0}})$ whereas the associated rule (H3) below treats arbitrary terms beginning with P. The final rules are:

$$\begin{array}{lrl}
(\mathrm{H1}) & \bullet\mathsf{S}x & \to \circ x \\
(\mathrm{H2}) & \mathsf{P}(\bar{\mathsf{0}},\mathsf{S}y) & \to \circ\mathsf{M}(\bar{\mathsf{0}},y) \\
(\mathrm{H3}) & \mathsf{P}(\bar{\mathsf{0}},\mathsf{P}(\bar{x},y)) & \to \circ\mathsf{M}(\bar{x},y) \\
(\mathrm{H}_i4) & \mathsf{P}(x_1,\dots,\mathsf{S}x_i,\bar{\mathsf{0}},y) & \to \circ\mathsf{J}_i(\bar{x},y) \\
(\mathrm{H}_i5) & \mathsf{P}(x_1,\dots,\mathsf{S}x_i,\bar{\mathsf{0}},\mathsf{S}y) & \to \circ\mathsf{J}_i(\bar{x},\mathsf{P}(x_1,\dots,\mathsf{S}x_i,\bar{\mathsf{0}},y)) \\
(\mathrm{H}_{ij}6) & \bullet\mathsf{P}(x_1,\dots,x_i,\bar{\mathsf{0}},\mathsf{0}) & \to \mathsf{Q}_{ij}(x_1,\dots,\bullet x_i,x_j) \\
(\mathrm{R}\mathsf{Q}_{ij}) & \mathsf{Q}_{ij}(x_1,\dots,\circ x_i,y) & \to \circ\mathsf{P}(\bar{x},\bar{\mathsf{0}},y) \\
(\mathrm{H}_i7) & \bullet\mathsf{P}(x_1,\dots,x_i,\bar{\mathsf{0}},\mathsf{S}y) & \to \mathsf{R}_i(x_1,\dots,\bullet x_i,x_i,y) \\
(\mathrm{R}\mathsf{R}_i) & \mathsf{R}_i(x_1,\dots,\circ x_i,y,z) & \to \circ\mathsf{P}(\bar{x},\bar{\mathsf{0}},\mathsf{P}(x_1,\dots,x_{i-1},y,\bar{\mathsf{0}},z))
\end{array}$$

for i and j with $1 \leqslant j \leqslant i \leqslant k$.

Proposition 5.46. *For $d \in \mathcal{D}$ with $d \neq \mathsf{0}$ and for $n \geqslant 0$ we have*

$$\bullet\square^{n+1}d \xrightarrow{+} \square^{n+1}\bullet^{n+1}d \xrightarrow{+} \square^{n+1}\circ d[n] \xrightarrow{+} \begin{cases} \circ d[n] \\ \bullet\square^{n+2}d[n] \ . \end{cases}$$

Before we prove this, we would like to point out its main implication. By Lemma 5.39, $\mathcal{R}$ is able to simulate Hydra battles for all configurations (α, n) with $0 < \alpha < \Delta_k$ at full length:

$$\bullet\square^{n+1}d \xrightarrow{+} \bullet\square^{n+2}d[n] \xrightarrow{+} \bullet\square^{n+3}d[n,n+1] \xrightarrow{+} \dots \xrightarrow{+} \bullet\square^{n+l}\mathsf{0} \xrightarrow{+} \mathsf{0} \ .$$

It is obviously not wise to strive after a similar result for $d = \mathsf{0}$.

Proof by induction on $\mathcal{D}$. As mentioned before, a close look at Definition 5.38 shows that whenever $u[n]$ (with u being a subterm of d) is used to define $d[n]$ we have $u \neq \mathsf{0}$. This observation enables us to rely on the induction hypothesis when required.

A quick glance at Lemmata 5.42 and 5.43 assures us that it suffices to show

$$\bullet^{n+1}d \xrightarrow{+} \circ d\langle n\rangle \ .$$

For (5.8b) we employ Lemma 5.43.i to get $\bullet^{n+1}\mathsf{S}s \xrightarrow{*} \bullet\mathsf{S}s \to_{\mathrm{H1}} \circ s = \circ d\langle n\rangle$, while (5.8c) is handled by Lemma 5.44 and the induction hypothesis:

$$\bullet^{n+1}{+}(s,t) \xrightarrow{+} {+}(s,\bullet^{n+1}t) \xrightarrow{+} {+}(s,\circ t[n]) \to_{\mathrm{E}_2+} \circ{+}(s,t[n]) \ .$$

The treatment of (5.8d) and (5.8k) is very close to this, relying on ($\mathrm{E}_i\mathsf{P}$) instead of (E_2+). For (5.8f) we note that $d = \mathsf{P}(\bar{\mathsf{0}},u)$ with $u \in \mathsf{Fix}(\bar{\mathsf{0}})$ holds. According to Definition 5.27, u is some $\mathsf{P}(\bar{s},t)$. Applying Lemmata 5.44.ii,iii and 5.45.i,

$$\begin{aligned}
\bullet^{n+1}\mathsf{P}(\bar{\mathsf{0}},\mathsf{P}(\bar{s},t)) &\xrightarrow{+} \quad \mathsf{P}(\bar{\mathsf{0}},\mathsf{P}(\bar{s},\Box^{n+1}t)) \\
&\to_{\mathrm{H3}} \circ\mathsf{M}(\bar{s},\Box^{n+1}t) \\
&\xrightarrow{+} \quad \circ(\mathsf{P}(\bar{s},t)\times(n+1))
\end{aligned}$$

follows. The proof of (5.8e) is very similar (using (H2)) and therefore left out here. For (5.8h) we need Lemmata 5.44.iv and 5.45.ii:

$$\begin{aligned}
\bullet^{n+1}\mathsf{P}(s_1,\dots,\mathsf{S}s_i,\bar{\mathsf{0}},t) &\xrightarrow{+} \quad \mathsf{P}(s_1,\dots,\mathsf{S}\Box^{n+1}s_i,\bar{\mathsf{0}},t) \\
&\to_{\mathrm{H}_i4} \circ\mathsf{J}_i(s_1,\dots,\Box^{n+1}s_i,t) \\
&\xrightarrow{+} \quad \circ\mathsf{P}(s_1,\dots,s_i,\cdot,\bar{\mathsf{0}})^{n+1}(t) \ ,
\end{aligned}$$

while we care for (5.8g) in much the same way, replacing (H_i4) by (H_i5). The treatment of $\mathrm{MS}_{\bar{s},\bar{\mathsf{0}}}(\bar{s})$ in (5.8i) requires a new idea, since $\mathcal{R}$ does not know which of the s_j has $\mathrm{MS}_{\bar{s},\bar{\mathsf{0}}}(\bar{s})$ as a subterm. By virtue of Lemma 5.44.ii and the induction hypothesis, for each j with $1 \leqslant j \leqslant i$ and for $s_i \neq \mathsf{0}$, we can show

$$\begin{aligned}
\bullet^{n+1}\mathsf{P}(s_1,\dots,s_i,\bar{\mathsf{0}},\mathsf{0}) &\xrightarrow{*} \quad \bullet\mathsf{P}(s_1,\dots,\bullet^{n}s_i,\bar{\mathsf{0}},\mathsf{0}) && (5.10\mathrm{a}) \\
&\xrightarrow{+} \quad \mathsf{Q}_{ij}(s_1,\dots,\bullet^{n+1}s_i,s_j) && (5.10\mathrm{b}) \\
&\xrightarrow{+} \quad \mathsf{Q}_{ij}(s_1,\dots,\circ s_i[n],s_j) \\
&\to_{\mathrm{R}\mathsf{Q}_{ij}} \circ\mathsf{P}(s_1,\dots,s_i[n],\bar{\mathsf{0}},s_j) \ . && (5.10\mathrm{c})
\end{aligned}$$

To get from (5.10a) to (5.10b), we make use of ($\mathrm{H}_{ij}6$) and, in case of $j = i$, Lemma 5.43.i. Now we may incorporate Lemma 5.41 to reduce the second s_j in (5.10c) to any of its subterms. Since $\mathrm{MS}_{\bar{s},\bar{\mathsf{0}}}(\bar{s})$ is a subterm of some s_j, we reach our goal. When we use (H_i7) and (RR_i) instead of ($\mathrm{H}_{ij}6$) and ($\mathrm{R}\mathsf{Q}_{ij}$), the result for (5.8j) is easily obtained. □

Corollary 5.47. *If $\mathcal{R}$ terminates, then both* $\mathrm{Dl}_{\mathcal{R}}$ *and* $\mathrm{Dc}_{\mathcal{R}}$ *eventually dominate all $<\!\Delta_k$-recursive functions.*

Proof. As mentioned in (5.6), $\Delta_k = \psi(1, \bar{0})$ holds where $\bar{0}$ has length $k+1$. Proposition 5.32 tells us the function which maps n to the length of the Hydra battle for $c_n := (\Delta_k[n], n+1)$ eventually dominates all $<\Delta_k$-recursive functions. We take a short digression from our fixed k to $k+1$, recall $\psi_{0,\bar{\alpha}} = \psi_{\bar{\alpha}}$, and see

$$\Delta_k[n] = \psi(0, \cdot, \bar{0})^{n+1}(0) = \psi(\cdot, \bar{0})^{n+1}(0)$$

holds where the first ψ is $k+2$-ary and the second one is, as usual, $k+1$-ary. Since $s_n := \bullet[]^{n+2}\mathsf{P}(\cdot, \bar{\mathsf{0}})^n(\mathsf{S0})$ encodes c_n, the length of the battle for c_n is majorized by $\mathrm{dl}_{\mathcal{R}}(s_n)$. Because of $\mathrm{dp}(s_{n+1}) = \mathrm{dp}(s_n) + 2$, $\mathrm{Dl}_{\mathcal{R}}$ grows too fast to be $<\Delta_k$-recursive. The calculation for $\mathrm{Dc}_{\mathcal{R}}$ is similar. □

Since the complexity of a TRS terminating via either MPO, KBO, or LPO is bounded by a multiple recursive function (see Theorems 5.6, 5.7, and 5.15), termination of $\mathcal{R}$ cannot be established by one of these orders.

We want to demonstrate where LPO fails. $\mathcal{R}$ owes much of its strength to the interplay between $\bullet$, $[]$, and $\circ$ on the one hand and $+$ and P on the other hand. No LPO is able to prove termination of a TRS containing the rules (N3), (N4), ($\mathrm{D_1P}$), and ($\mathrm{E_1P}$), since the first three rules require $\circ \succ [] \succ \bullet \succ \mathsf{P}$ while the fourth rule implies $\mathsf{P} \succ \circ$.

Touzet (1999) conjectured the approach taken in the proof of Theorem 5.23 is transferable to TRSs. There an interpretation was defined by mapping a term s to a pair (u, v) of terms, where u (mostly) contains the part of s which corresponds to its denoted ordinal, while v contains the auxiliary symbols (like $\bullet$ and $[]$) of s. Comparison of terms was achieved by the lexicographic product of $\prec_{\mathsf{mpo}}$ and an order $\prec$ tailored to meet the demands of the rules importing resp. exporting auxiliary symbols. Such a lexicographic product is needed to destroy the uniformity of termination via MPO, as, according to Theorem 5.6, $\prec_{\mathsf{mpo}}$ alone is doomed to stick with primitive recursive complexity bounds. On the other hand, it is reasonable to make MPOs a part of the termination proof, as their order types exhaust the order types of simplification orders on strings (see Theorems 4.2 and 4.9). For terms, the weapon of choice appears to be LPO. The final remark of Touzet (1999) is

> *"We believe that this approach would apply to term rewriting systems, using the lexicographic path ordering on terms instead of the recursive path ordering* [i.e. MPO] *on strings: the order type of the lexicographic path ordering reaches the maximal order type of the homeomorphic embedding of Kruskal's theorem."*

In order to make this a bit more explicit we recall that $\vartheta(\Omega^\omega)$, the supremum of the order types of simplification orders, is also the supremum of the order types of LPOs, see Theorems 4.1 and 4.10. Just as termination via MPO on strings is too weak, a direct approach using LPO on terms is impossible since we run aground at multiple recursive complexity bounds, see Theorem 5.7.

Though the approach of Touzet (1999) is promising, we will take a different and more direct route, which is inspired by the earlier paper Touzet (1998b).

5.3.5 Proof of Total Termination

We order $\mathcal{P} := (\Delta_k \setminus \{0\}) \times \omega \times \omega$ by $\prec$, which is the lexicographic product of the usual $<$ on these sets of ordinals. By Proposition 2.38.ii, $(\mathcal{P}, \prec)$ is a well-order with (using Lemma 2.26 and the fact that Δ_k is an epsilon)

$$\mathrm{otype}(\mathcal{P}, \prec) = \omega \cdot \omega \cdot \Delta_k = \omega^2 \cdot \omega^{\Delta_k} = \omega^{2+\Delta_k} = \omega^{\Delta_k} = \Delta_k \ .$$

We identify $(\alpha, 0, 0) \in \mathcal{P}$ and α to avoid lengthy notations. Thus $\alpha > \beta$ implies $\alpha \succ (\beta, m, n) \succcurlyeq \beta$. Our main intention is to define a weak monotone Σ-algebra which is compatible with $\mathcal{R}$.

Definition 5.48. The operations of the Σ-algebra $(\mathcal{P}, \prec, \mathcal{F})$ are defined as follows. For arbitrary elements $p = (\alpha, m, n)$, $p_l = (\alpha_l, m_l, n_l)$ (with $1 \leqslant l \leqslant k$), $q = (\beta, m', n')$ and $r = (\gamma, m'', n'')$ of $\mathcal{P}$ we put

$$\begin{aligned}
[\mathsf{0}] &:= 1 \\
[\mathsf{S}](p) &:= \alpha + 1 \\
[+](p, q) &:= \alpha \oplus \beta \oplus \beta \\
[\mathsf{P}](\bar{p}, q) &:= \psi(\bar{\alpha}, \beta) \\
[\bullet](p) &:= (\alpha, m, n+1) \\
[\square](p) &:= (\alpha, m+1, 0) \\
[\circ](p) &:= \alpha + 1 \\
[\mathsf{M}](\bar{p}, q) &:= \psi(\bar{\alpha}, \beta) \cdot (3m' + 1) \\
[\mathsf{J}_i](p_1, \ldots, p_i, q) &:= \psi(\alpha_1, \ldots, \alpha_i, \cdot, \bar{1})^{2m_i+1}(\beta) \\
[\mathsf{Q}_{ij}](p_1, \ldots, p_i, q) &:= \psi(\alpha_1, \ldots, \max\{\alpha_j, \beta\}, \ldots, \alpha_i, \bar{1}) \\
[\mathsf{R}_i](p_1, \ldots, p_i, q, r) &:= \psi(\alpha_1, \ldots, \max\{\alpha_i, \beta\}, \bar{1}, \gamma + 1) \ .
\end{aligned}$$

Note that some components of q, r and p_l are never used. Furthermore the $[f]$ with $f \in \Sigma_0$ intentionally forget the two trailing components of p, q and p_l. Hence, for $\rho \colon \mathcal{V} \to \mathcal{P}$ we have

$$\begin{aligned}
&\llbracket \mathsf{S}\square x, \rho \rrbracket = \llbracket \mathsf{S}x, \rho \rrbracket = \llbracket \circ x, \rho \rrbracket = \llbracket \circ \square x, \rho \rrbracket \ , \\
&\llbracket +(\square x, y), \rho \rrbracket = \llbracket +(x, \square y), \rho \rrbracket = \llbracket +(x, y), \rho \rrbracket \ , \text{ and} \qquad (5.11) \\
&\llbracket \mathsf{P}(x_1, \ldots, \square x_i, \ldots, x_{k+1}), \rho \rrbracket = \llbracket \mathsf{P}(\bar{x}), \rho \rrbracket \ .
\end{aligned}$$

This remains true when $[\!]$ is replaced with $\bullet$.

Both the use of m' in $[\mathsf{M}]$ and the use of m_i in $[\mathsf{J}_i]$ are based on $[[\!]]$, as m' and m_i count the appearances of $[\!]$ at the positions important for (RM) and (RJ_i), respectively. The definitions of $[\mathsf{Q}_{ij}]$ and $[\mathsf{R}_i]$ reflect the duplication of x_j and x_i in (H$_{ij}$6) and (H$_i$7), respectively. Here we profit from Theorem 3.45, since taking the maximum violates monotonicity. Note that only $[\bullet]$ is monotone, as the other functions ignore the third component of their first arguments.

Lemma 5.49. *The Σ-algebra $(\mathcal{P}, \prec, \mathcal{F})$ is weakly monotone and has the subterm property, hence it is weak monotone.*

Proof. We start with the subterm property and treat only the cases which are not that obvious. The proof for $[+]$ uses the fact that the first components of elements of $\mathcal{P}$ are larger than 0, and the functions involving ψ rely on the subterm property and the monotonicity of ψ, hence on Lemma 5.25.ii.

Weak monotonicity is obvious for all interpretations from $[\mathsf{0}]$ to $[\circ]$, and a moment's reflection establishes it for $[\mathsf{Q}_{ij}]$ and $[\mathsf{R}_i]$. The result for $[\mathsf{M}]$ and $[\mathsf{J}_i]$ relies on Lemma 5.26. □

For the reader's convenience we collect the rules of $\mathcal{R}$ scattered over the last pages in Table 5.1 on the following page.

Theorem 5.50. *$\mathcal{R}$ is Δ_k-terminating, thus totally terminating.*

Proof. Our aim is to establish weak Δ_k-termination of $\mathcal{R}$. As soon as we reach this, Theorem 3.45 yields ω^{Δ_k}-termination. Because Δ_k is an epsilon, this is just what we are longing for.

By Lemma 5.49 it is sufficient to show $\mathcal{R}$ is compatible with $(\mathcal{P}, \prec, \mathcal{F})$. Let $\rho\colon \mathcal{V} \to \mathcal{P}$ be given, and let the values under ρ of x, x_i (with $1 \leqslant i \leqslant k+1$), y and z be (α, m, n), (α_i, m_i, n_i), (β, m', n') and (γ, m'', n''). Our goal will be established by showing

$$[\![l, \rho]\!] \succ [\![r, \rho]\!] \quad \text{for all } (l, r) \in \mathcal{R} \ .$$

For all rules (S$_i f$) we can fall back upon the subterm property of the interpreting functions under consideration. The subterm properties of $[\bullet]$ and $[\circ]$, sometimes in combination with (5.11), settle (N2), (N3), (D$_i f$), (H1), (H$_{ij}$6), and (H$_i$7). For example, we can treat (N3) by

$$[\![\circ x, \rho]\!] = [\![\circ[\!]x, \rho]\!] \succ [\![[\!]x, \rho]\!] \ ,$$

and for (H$_i$7)

$$\begin{aligned} [\![\bullet\mathsf{P}(x_1, \ldots, x_i, \bar{\mathsf{0}}, \mathsf{S}y), \rho]\!] &\succ [\![\mathsf{P}(x_1, \ldots, x_i, \bar{\mathsf{0}}, \mathsf{S}y), \rho]\!] \\ &= [\![\mathsf{P}(x_1, \ldots, \bullet x_i, \bar{\mathsf{0}}, \mathsf{S}y), \rho]\!] \\ &= [\![\mathsf{R}_i(x_1, \ldots, \bullet x_i, x_i, y), \rho]\!] \end{aligned}$$

$(\mathrm{S}_i f)$	$f(x_1,\dots,x_n) \to x_i$	$[f \in \Sigma]$
(F1)	$\mathsf{P}(\bar{\mathsf{0}}) \to \mathsf{S0}$	
(F2)	$+(x,\mathsf{S}y) \to \mathsf{S}{+}(x,y)$	
(N1)	$\bullet\Box x \to \Box\bullet\bullet x$	
(N2)	$\Box\circ x \to \circ\Box\Box x$	
(N3)	$\circ x \to \Box x$	
(N4)	$\Box x \to \bullet x$	
$(\mathrm{D}_i f)$	$\bullet f(\bar{x}) \to f(x_1,\dots,\Box x_i,\dots,x_n)$	$[f \in \{\mathsf{S},+,\mathsf{P}\}]$
$(\mathrm{E}_2{+})$	$+(x,\circ y) \to \circ{+}(x,y)$	
$(\mathrm{E}_i\mathsf{P})$	$\mathsf{P}(x_1,\dots,\circ x_i,\dots,x_{k+1}) \to \circ\mathsf{P}(\bar{x})$	
$(\mathrm{R}\mathsf{M})$	$\mathsf{M}(\bar{x},\Box y) \to +(\mathsf{M}(\bar{x},y),\mathsf{P}(\bar{x},y))$	
$(\mathrm{R}\mathsf{J}_i)$	$\mathsf{J}_i(x_1,\dots,\Box x_i,y) \to \mathsf{P}(\bar{x},\mathsf{J}_i(\bar{x},y),\bar{\mathsf{0}})$	
(H1)	$\bullet\mathsf{S}x \to \circ x$	
(H2)	$\mathsf{P}(\bar{\mathsf{0}},\mathsf{S}y) \to \circ\mathsf{M}(\bar{\mathsf{0}},y)$	
(H3)	$\mathsf{P}(\bar{\mathsf{0}},\mathsf{P}(\bar{x},y)) \to \circ\mathsf{M}(\bar{x},y)$	
(H_i4)	$\mathsf{P}(x_1,\dots,\mathsf{S}x_i,\bar{\mathsf{0}},y) \to \circ\mathsf{J}_i(\bar{x},y)$	
(H_i5)	$\mathsf{P}(x_1,\dots,\mathsf{S}x_i,\bar{\mathsf{0}},\mathsf{S}y) \to \circ\mathsf{J}_i(\bar{x},\mathsf{P}(x_1,\dots,\mathsf{S}x_i,\bar{\mathsf{0}},y))$	
$(\mathrm{H}_{ij}6)$	$\bullet\mathsf{P}(x_1,\dots,x_i,\bar{\mathsf{0}},\mathsf{0}) \to \mathsf{Q}_{ij}(x_1,\dots,\bullet x_i,x_j)$	
$(\mathrm{R}\mathsf{Q}_{ij})$	$\mathsf{Q}_{ij}(x_1,\dots,\circ x_i,y) \to \circ\mathsf{P}(\bar{x},\bar{\mathsf{0}},y)$	
(H_i7)	$\bullet\mathsf{P}(x_1,\dots,x_i,\bar{\mathsf{0}},\mathsf{S}y) \to \mathsf{R}_i(x_1,\dots,\bullet x_i,x_i,y)$	
$(\mathrm{R}\mathsf{R}_i)$	$\mathsf{R}_i(x_1,\dots,\circ x_i,y,z) \to \circ\mathsf{P}(\bar{x},\bar{\mathsf{0}},\mathsf{P}(x_1,\dots,x_{i-1},y,\bar{\mathsf{0}},z))$	

Table 5.1: *The rules of $\mathcal{R}$ (in order of their appearance)*

suffices. In the same way we build on $\llbracket\Box x,\rho\rrbracket = \llbracket\Box\bullet x,\rho\rrbracket$ to treat (N1) and (N4), while (F2) and $(\mathrm{E}_2{+})$ both rely on

$$\alpha \oplus (\beta+1) \oplus (\beta+1) = \alpha \oplus \beta \oplus \beta + 2 > \alpha \oplus \beta \oplus \beta + 1 \ .$$

Since (F1), $(\mathrm{E}_i\mathsf{P})$, and (H2) can be handled similarly, it suffices to show what to do with the latter. By monotonicity of ψ we get

$$\llbracket\mathsf{P}(\bar{\mathsf{0}},\mathsf{S}y),\rho\rrbracket = \psi(\bar{1},\beta+1) \succ \psi(\bar{1},\beta) \ ,$$

and now Lemma 5.26.ii yields

$$\psi(\bar{1},\beta+1) \succ \psi(\bar{1},\beta)\cdot(3m'+1)+1 = \llbracket\circ\mathsf{M}(\bar{\mathsf{0}},y),\rho\rrbracket \ .$$

We can handle (H3) in almost the same way, using the subterm property of ψ instead of its monotonicity. The proof for $(\mathrm{R}\mathsf{M})$ boils down to establishing

$$\psi(\bar{\alpha},\beta)\cdot(3(m'+1)+1) > \psi(\bar{\alpha},\beta)\cdot(3(m'+1)) \ ,$$

which is shown via Lemma 5.26.i, and (RJ$_i$) is taken care of by Lemma 5.26.iii:

$$\psi(\alpha_1,\ldots,\alpha_i,\cdot,\bar{1})^{2(m_i+1)+1}(\beta) > \psi(\alpha_1,\ldots,\alpha_i,\cdot,\bar{1})^{2m_i+2}(\beta)\ .$$

To settle (H$_i$5) we use the subterm property and the monotonicity of ψ, and see

$$\psi(\alpha_1,\ldots,\alpha_i+1,\bar{1},\beta+1) > \bar{\alpha},1,\psi(\alpha_1,\ldots,\alpha_i+1,\bar{1},\beta) =: \delta\ .$$

This is joined by Lemma 5.26.iv,ii to produce

$$\psi(\alpha_1,\ldots,\alpha_i+1,\bar{1},\beta+1) > \psi(\alpha_1,\ldots,\alpha_i,\cdot,\bar{1})^{2m_i+1}(\delta)+1\ .$$

Along this line of reasoning we can get rid of (H$_i$4). The key observation is

$$\psi(\alpha_1,\ldots,\alpha_i+1,\bar{1},\beta) > \psi(\alpha_1,\ldots,\alpha_i,\cdot,\bar{1})^{2m_i+1}(\beta)+1\ .$$

It remains to take care of (RQ_{ij}) and (RR_i). For the former we introduce $p := [\![\mathsf{Q}_{ij}(x_1,\ldots,\circ x_i,y),\rho]\!]$ and get $p \succcurlyeq \psi(\alpha_1,\ldots,\alpha_i+1,\bar{1})$ by weak monotonicity of ψ, whereas the subterm property of ψ yields $p \succ \alpha_1,\ldots,\alpha_i,1,\beta$. Under these conditions Lemma 5.25.iii implies

$$p \succ \psi(\alpha_1,\ldots,\alpha_i,\bar{1},\beta)+1 = [\![\circ\mathsf{P}(\bar{x},\bar{0},y),\rho]\!]\ .$$

Via similar reasoning we can handle (RR_i), getting

$$p := \psi(\alpha_1,\ldots,\max\{\alpha_i+1,\beta\},\bar{1},\gamma+1) \succ \bar{\alpha},1,\psi(\alpha_1,\ldots,\alpha_{i-1},\beta,\bar{1},\gamma) =: \delta$$

first and $p \succ \psi(\bar{\alpha},\bar{1},\delta)+1$ afterwards, using Lemma 5.25.iii and, yet again, Lemma 5.26.ii. □

Combining Corollary 5.47, Theorem 5.50, and Theorem 5.20 we arrive at an essentially optimal result.

Theorem 5.51.

i. *For every $\alpha < \vartheta(\Omega^\omega)$ there exists a simply (and even totally) terminating TRS whose complexity eventually dominates all α-recursive functions.*

ii. *For every simply terminating TRS $\mathcal{R}$ there exists a $<\!\vartheta(\Omega^\omega)$-recursive function which dominates the complexity of $\mathcal{R}$.*

In conclusion we learn that simply terminating TRSs are blessed with an enormous computational strength. This strength is by no means exhausted by the common classes of simplification orders like MPO, LPO, or KBO. The question arises whether there is a natural class of simplification orders whose corresponding TRSs may attain complexities beyond multiple recursion.

class	bound in	order types	see Theorems
MPO(1)	Prec	$<\omega_3$	5.6/4.2
MPO	Prec	$<\vartheta(\Omega \cdot \omega)$	5.6/4.1
KBO$^-$	$2^{O(n)}$	ω	5.8/4.6
KBO(1)	$2^{O(n)}$	$\leqslant\omega^\omega$	
KBO	$\mathrm{Ack}(O(n), 0)$	$\leqslant\omega^\omega$	5.16/4.6
LPO	Mrec	$<\vartheta(\Omega^\omega)$	5.7/4.1
SO(1)	Mrec	$<\omega_3$	5.23/4.9
SO	Drec($<\vartheta(\Omega^\omega)$)	$<\vartheta(\Omega^\omega)$	5.51/4.10

Table 5.2: *Size complexity bounds and order types occuring within classes of simplification orders.*

Table 5.2 contains a collection of the most important results presented in this and the previous chapter. We use the abbreviation SO for the class of simplification orders. Recall from Definition 3.48 that $\mathcal{C}(1)$ contains the members of a class $\mathcal{C}$ of simplification orders which live on a signature Σ satisfying $\Sigma = \Sigma^{(\leqslant 1)}$. There is no need to list LPO(1) as, by Lemma 3.55, it coincides with MPO(1).

What is the reason for the frequent appearance of the Hardy functions in this section? Recall the slow growing principle, which states that a termination proof via a simplification order of order type α implies as an upper complexity bound a slow growing function with index $\beta < \vartheta(\Omega^\omega)$, where β is related to α. After the construction of first counterexamples to this principle, Touzet (1999) adapted it to her results and proposed that

> *"the Hardy hierarchy is the right tool for connecting derivation length and order type."*

We call this the *Hardy function principle.* From Buchholz et al. (1994, Theorem 1) we can infer that a termination proof using a simplification order of order type α implies the size complexity is dominated by some H_β with $\beta < \alpha + \omega^\omega$. Thus there is a valid instance of this principle. We do however know far too little about the tightness of the described bounds, even if we focus on sets of TRSs terminating via certain termination proof methods instead of considering single TRSs.

The results presented in this chapter give strong evidence of the importance of the Hardy function principle. In the general case of a simply terminating SRS, the order type is below ω_3, and we can always find a $\beta < \omega_3$ such that H_β bounds the complexity of the SRS. For terms the situation is similar. The TRSs $\mathcal{R}_k$ (depending on $k > 0$) simulating the Hydra battle below Δ_k we considered in this section are in some sense the most complex simply terminating TRSs, and the size complexity of $\mathcal{R}_k$ is closely related to H_{Δ_k}.

A further outstanding example is KBO with a maximal order type of ω^ω and size complexities cofinal in $\mathrm{Ack}(O(n), 0)$. The Hardy function $\mathrm{H}_{\omega^\omega}$ is a version of the (binary) Ackermann function Ack (cf. Theorem 2.83). Less convincing examples are MPO and LPO, which are not able to make full use of the huge order types of their members. For the size complexities occuring within termination via MPO, the H_α with $\alpha < \omega^\omega$ suffice, while those of termination via LPO are controlled by the H_α with $\alpha < \omega_3$.

The TRSs simulating the Hydra battle will show up again in Chapter 7, where we ask what complexities are possible if further restrictions are imposed on the sizes of the terms occurring in the derivations.

5.4 A Digression: LPO-Controlled Derivations

Harvey Friedman (1999) investigated a more general notion of derivation length which is also applicable to nonterminating TRSs.

Definition 5.52. A *well-founded ordered TRS* is a triple $(\mathcal{R}, \Sigma, \prec)$ such that $\mathcal{R}$ is a TRS over the signature Σ and $(\mathcal{T}(\Sigma), \prec)$ is a well-founded partial order. The $\mathcal{R}$-derivation $(s_i)_{i<n}$ is a $\prec$*-derivation* if we have $s_i \succ s_{i+1}$ for all $i < n - 1$.

Note that, by the well-foundedness of $\prec$, it makes little sense to consider infinite derivations as well. Because $\rightarrow_\mathcal{R}$ is finitely branching we can safely transport the derivation length function to this new context.

Definition 5.53. Let $(\mathcal{R}, \Sigma, \prec)$ be as above. By $\mathrm{dl}^{\prec}_{\mathcal{R}}(s)$ we denote the length of a longest $\prec$-derivation starting with $s \in \mathcal{T}(\Sigma)$. Based on this we can define $\mathrm{Dl}^{\prec}_{\mathcal{R}}$ almost as we defined $\mathrm{Dl}_\mathcal{R}$.

This notion extends the usual one since, for terminating $\mathcal{R}$, $\mathrm{dl}_\mathcal{R}$ and $\mathrm{dl}^{\prec}_{\mathcal{R}}$ with $\prec \,:=\, \overset{+}{\leftarrow}_\mathcal{R}$ coincide. We may however also consider nonterminating TRSs. Before we do so, we take a look at terminating TRSs and find out that the behavior of $\mathrm{Dl}^{\prec}_{\mathcal{R}}$ is already known if $\prec$ is a simplification order.

Theorem 5.54 (Weiermann 1994). *If $(\mathcal{R}, \Sigma, \prec)$ is a well-founded ordered TRS and $(\mathcal{T}(\Sigma), \prec)$ is (the closed part of) a simplification order, then there exists a $<\!\vartheta(\Omega^\omega)$-recursive function which dominates $\mathrm{Dl}^{\prec}_{\mathcal{R}}$.*

A complementary result is the following consequence of Theorem 5.51.

Theorem 5.55. *For every $\alpha < \vartheta(\Omega^\omega)$ there exists a well-founded ordered TRS $(\mathcal{R}, \Sigma, \prec)$ such that $(\mathcal{T}(\Sigma), \prec)$ is (the closed part of) a simplification order and $\mathrm{Dl}^{\prec}_{\mathcal{R}}$ eventually dominates all α-recursive functions.*

We already pointed out why it is impossible for the $\mathcal{R}$ of the Theorem to terminate via LPO, although the order types of LPOs are cofinal in the order types of simplification orders. If we use well-founded ordered rewriting, LPO becomes strong enough – presumably at the cost of termination.

Theorem 5.56 (Friedman 1999). *For every $\alpha < \vartheta(\Omega^\omega)$ there exists a (non-terminating) well-founded ordered TRS $(\mathcal{R}, \Sigma, \prec)$ such that $\prec$ is (the closed part of) an LPO and $\mathrm{Dl}_{\mathcal{R}}^{\prec}$ eventually dominates all α-recursive functions.*

Friedman showed that these growth rates are attained even if we restrict ourselves to head derivations, i.e. those derivations $s \to t$ for whom there exist a rule (l, r) and a substitution σ satisfying $s = l\sigma$ and $t = r\sigma$.

It is appealing to generalize the notion of well-founded ordered TRSs by introducing a control function $g\colon \mathbb{N}^2 \to \mathbb{N}$ (whose branches $m \mapsto g(n, m)$ are called g_n). A derivation $(s_i)_{i<n}$ is called a *$\prec$,g-derivation* if there is a monotone function $f \leqslant_{\mathrm{d}} g_{|s_0|}$ such that $s_{f(i)} \succ s_{f(i+1)}$ holds for all i with $f(i+1) < n$.

Obviously, for the g defined by $g(n, m) := m$ the $\prec$-derivations and the $\prec$,g-derivations coincide. It is of course not advisable to choose a g with $g(n, 0)$ too close to $\mathrm{dl}_{\mathcal{R}}(n)$, and additionally an elementary g should suffice.

Let us return to Section 5.3.4 and its TRS $\mathcal{R}$ which is able to simulate all Hydra battles below Δ_k for some fixed k. According to Proposition 5.46, for all standard terms $d \in \mathcal{D}$ with $d \neq \mathsf{0}$ and all $n \geqslant 0$ we have

$$\bullet[\,]^{n+1} d \xrightarrow{+} \bullet[\,]^{n+2} d[n] \;.$$

The LPO $\succ_{\mathsf{lpo}}$ based on $\mathsf{P} \succ + \succ \mathsf{S} \succ \mathsf{0} \succ \bullet \succ [\,]$ satisfies

$$\bullet[\,]^{n+1} d \succ_{\mathsf{lpo}} \bullet[\,]^{n+2} d[n] \;.$$

In the proof of Corollary 5.47 we used terms $s_n = \bullet[\,]^{n+2}\mathsf{P}(\cdot, \bar{\mathsf{0}})^n(\mathsf{S0})$ and found out that the function mapping n to $\mathrm{dl}_{\mathcal{R}}(s_n)$ is not $<\Delta_k$-recursive. If we now define $g\colon \mathbb{N}^2 \to \mathbb{N}$ to be such that $g_{|s_n|}$ enumerates the positions of (codes for) battle configurations in a derivation of minimal length simulating the Hydra battle starting with s_n, then this derivation is $\prec_{\mathsf{lpo}}$,g-controlled. Hence the function which maps n to the longest $\prec_{\mathsf{lpo}}$,g-controlled derivation starting with a term of size n is not $<\Delta_k$-recursive. Even the function which maps n to the first m such that $g_n(m)$ exceeds the lengths of all $\prec_{\mathsf{lpo}}$,g-controlled derivations starting with a term of size n is not $<\Delta_k$-recursive.

In Chapter 7 we will take a closer look at a variant g' of g (for a variant of $\mathcal{R}$) and see that there is an elementary upper bound on g'. Roughly speaking, the length of the derivation from $\bullet[\,]^{n+1}d$ to $\bullet[\,]^{n+2}d[n]$ depends on n, $\mathrm{dp}(d)$ and $|d|$. Proposition 7.25 shows $|d[n]| < (|d| - 1)(\mathrm{dp}(d) + n)$, while Proposition 7.28 yields $\mathrm{dp}(d[n]) \leqslant \mathrm{dp}(d) + n + 1$. These results can be used to define an elementary function whose branches impose upper bounds on the minimal number of rewrite steps between two encoded battle configurations.

5.5 Another Digression: Ground Termination

There is a concept of proving termination using a syntactic simplification order which is slightly broader than the usual one. We introduce it for KBO. If $\prec_{\mathsf{kbo}}$ is a KBO over a signature Σ, then we define *ground KBO*, $\prec_{\mathsf{gkbo}}$, on $\mathcal{T}(\Sigma, \mathcal{V})$ by

$$s \prec_{\mathsf{gkbo}} t \;:\Longleftrightarrow\; s\sigma \prec_{\mathsf{kbo}} t\sigma \text{ for all ground substitutions } \sigma \ .$$

Termination via KBO obviously implies termination via ground KBO.

Theorem 5.57 (Korovin and Voronkov 2001). *Termination via ground KBO is decidable in polynomial time.*

Ground KBO is not incremental with respect to signatures, as the introduction of a new constant having weight less than all other constants may be lethal. The latter concept is stronger than the former. Consider, over the signature Σ containing the symbols e, f, and g of arity 0, 1, and 2, resp., the TRS consisting solely of the rule

$$g(x, g(f(e), e)) \to g(e, g(e, f^2(e))) \ . \tag{5.12}$$

Although it does not terminate via KBO (because x and e are incomparable and f must be special), it does terminate via ground KBO by making f special and choosing legal weights for e and g. Korovin and Voronkov (2000) presented the example

$$h(x, a, b) \to h(b, b, a) \ , \tag{5.13}$$

which does not terminate via KBO but terminates via ground KBO if $\mu(a) > \mu(b)$ or $\mu(b) = \mu(a)$ and $a \succ b$.

A certain drawback of both two examples is the fact that the rewrite relations they induce correspond to rewrite relations of TRSs terminating via KBO. We simply have to replace, for all function symbols f' of arity n, the occuring variables with $f'(x_1, \ldots, x_n)$ where $x_1, \ldots, x_n$ are new variables. For (5.13), this leads to the TRS

$$\begin{aligned} h(h(x_1, x_2, x_3), a, b) &\to h(b, b, a) \\ h(a, a, b) &\to h(b, b, a) \\ h(b, a, b) &\to h(b, b, a) \ , \end{aligned}$$

which terminates via KBO with weight and precedence as above.

Such a transformation is successful if we have termination via a ground KBO which contains no special symbol. Thus termination via ground KBO$^-$ and termination via KBO$^-$ coincide. If a special symbol is required, it is not always possible to replace a TRS terminating via ground KBO with an equivalent TRS terminating via KBO. Consider, over the signature Σ used in (5.12), the TRS $\mathcal{R}$ solely consisting of the rule

$$g(f(g(e,e)),g(x,x)) \to g(g(x,e),f^2(g(e,e))) \; .$$

Weight considerations show f has to be special. For any valid choice of weight and precedence, $\mathcal{R}$ terminates via ground KBO. Even a repeated application of the above technique of closure under certain substitutions does not work here because of the variable condition of KBO. In fact, for any j we have $f(g(e,e)) \not\succ_{\mathsf{kbo}} g(f^j(x),e)$, because x does occur only in the term which is supposed to be smaller. A moment's reflection shows that, for every TRS terminating via KBO, there is a j such that

$$g(f(g(e,e)),g(f^j(e),f^j(e))) \not\to g(g(f^j(e),e),f^2(g(e,e)))$$

holds. Hence termination via ground KBO allows for rewrite relations which are not possible within termination via KBO.

A sharp complexity bound for termination via ground KBO is unknown. The lower bound from Theorem 5.8.i remains valid, and Hofbauer's 4-recursive upper bound from Theorem 5.8.ii applies. I conjecture the upper bounds are the same as those for the usual KBO and are always to be found in $\mathrm{Ack}(2^{O(n)},0)$. The main construction of Korovin and Voronkov (2001) implicitly contains a purely syntactic recursive definition of $\prec_{\mathsf{gkbo}}$ similar to the one of $\prec_{\mathsf{kbo}}$ in Definition 3.57, yet more involved. It should be possible to treat this definition with an interpretation like the one introduced in Definition 5.12.

The example (5.13) of Korovin and Voronkov (2000) can also be used to separate closed LPO from open LPO. It seems likely that the complexity bounds for closed MPO resp. closed LPO coincide with those of the open versions. After all, little is known about the closed versions of KBO, MPO, and LPO. They should be investigated further.

Incidentally it makes no sense to try to separate ground simple termination from simple termination, as these two notions coincide. Lemma 3.22 shows that the closed version of a simplification order on $\mathcal{T}(\Sigma,\mathcal{V})$ (more generally, the extension of any partial order on $\mathcal{T}(\Sigma)$ which has the subterm property and is closed under ground contexts) is again a simplification order on $\mathcal{T}(\Sigma,\mathcal{V})$.

6 Computability

Follow a fleeting thought!

In the preceding chapter we focused on the derivation complexity as one way to measure the strength of a termination proof method. Cichon and Lescanne (1992) studied a different and sometimes finer measure, which goes back to Huet and Oppen (1980) – the functions *computable via* a certain method. We recall from Definition 3.67 that, roughly speaking (and leaving aside assumptions dealing with partial confluence and normal forms), a function f of arity l is computable by a terminating TRS $\mathcal{R}$ if there are symbols $F, S, P, 0$, and $0'$ of appropriate arities such that

$$F(S^{n_1}0, \dots, S^{n_l}0) \xrightarrow{*}_{\mathcal{R}} P^{f(n_1,\dots,n_l)}0'$$

holds for all $n_1, \dots, n_l$. The distinction between the input successor S and the output successor P is important, as sometimes more complex functions are computable using two distinct successors. We recall further that, given a termination proof method $\mathcal{M}$, a function is computable via $\mathcal{M}$ if it is computable by some TRS which terminates via $\mathcal{M}$.

Many well known complexity classes coincide with the classes of functions computable via (restrictions of) standard termination proof methods. Such characterizations often impose lower bounds on the size complexity and even on the best-case behavior of a termination proof method, which is measured by the shortest size complexity. Because the size of an input term for input $n_1, \dots, n_l$ is linear in $n_1 + \cdots + n_l$, we focus on results concerning size complexities instead of (depth) complexities. The latter concept allows for exponential sizes of the input, blurring membership in the low computation classes.

Our definition of computation by TRSs implies that SRSs can only compute unary functions. As we are interested in the characterization of complexity classes containing functions of arbitrary arities, we will scarcely consider SRSs.

In Section 6.1 we will briefly enumerate the results concerning computation via polynomial termination and some of its restrictions, producing complexity classes between Ptime and E_2time. With almost the same brevity both MPO and LPO are considered in Section 6.2. Here the primitive respectively multiple recursive functions appear. Section 6.3 is different, as we put much effort into the treatment of KBO. A new result will be presented, telling that the functions computable via KBO are those computable in time $\mathrm{Ack}(O(n), 0)$. Finally, a quick glance at the general case is taken in Section 6.4.

6.1 Computability via PT

Cichon and Lescanne (1992) and Bonfante et al. (1999) observed that the computational power of polynomial termination mainly depends on the way the successor symbols are interpreted.

Definition 6.1 (Bonfante et al. 1999). Let Σ be a signature containing the input successor S and the output successor P, further let $\mathcal{R}$ be a TRS over Σ. We say $\mathcal{R}$ terminates via PT-L (PT-1) if it is polynomially terminating with linear functions (having leading coefficient 1) interpreting S and P.

Computation via PT using only one successor symbol does not lead far, and a nonlinear polynomial interpreting the successor symbol imposes an even stronger bound on the computed function. Recall from Definition 3.67 that $\mathrm{Comp}(\mathcal{M})$ denotes the set of functions computable via $\mathcal{M}$, while $\mathrm{Comp}_1(\mathcal{M})$ collects the functions computable via $\mathcal{M}$ using only one successor symbol.

Theorem 6.2 (Cichon and Lescanne 1992).

i. *We have* $\mathrm{Comp}_1(\text{PT-1}) \subseteq \mathrm{Ptime}$.

ii. *Each function in* $\mathrm{Comp}_1(\mathrm{PT} \setminus \text{PT-1})$ *has a linear upper bound.*

The use of two distinct successor symbols leads to much more adequate results. We know from Theorem 5.2 (resp. Theorem 5.3) that termination via PT (resp. LT) imposes upper bounds from $2^{2^{O(n)}}$ (resp. $2^{O(n)}$) on the size complexity. This is supplemented by the following Theorem.

Theorem 6.3 (Bonfante et al. 1999). *We have*

i. $\mathrm{Comp}(\text{PT-1}) = \mathrm{Ptime}$,

ii. $\mathrm{Comp}(\text{PT-L}) = \mathrm{Etime}$, *and*

iii. $\mathrm{Comp}(\mathrm{PT}) = \mathrm{E}_2\mathrm{time}$.

As indicated by its accompanying derivation lengths, computability via PT gets stuck at rather low complexity classes. The derivation lengths we encountered when we investigated the usual syntactic simplification orders promise much larger sets of computable functions.

6.2 Computability via MPO and LPO

In Proposition 5.5 we met SRSs terminating via MPO which are able to compute (using a different definition of computability than our Definition 3.67) branches Ack_n of the Ackermann function. This can be considerably generalized.

Theorem 6.4 (Plaisted 1978 and Hofbauer 1991, 1992).

$$\textsc{Comp}(\mathrm{MPO}) = \textsc{Comp}_1(\mathrm{MPO}) = \textsc{Prec}\ .$$

Proof. We know from Theorem 5.6 (of Hofbauer (1991, 1992)) that termination via MPO leads to a primitive recursive bound on the size complexity. Theorem 3.68 yields computability (on a TM) within primitive recursive time, and according to Corollary 2.94 this implies membership in Prec. Hence we get $\textsc{Comp}(\mathrm{MPO}) \subseteq \textsc{Prec}$.

A first proof of the opposite direction is to be found in Plaisted (1978). We consider an arbitrary primitive recursive function $f\colon \mathbb{N}^k \to \mathbb{N}$. As the primitive recursive functions are defined by a closure of certain basic functions under substitution and primitive recursion, only finitely many basic functions, substitutions $\mathsf{Sub}(g, h_1, \ldots, h_m)$ and primitive recursions $\mathsf{PRec}(g, h)$ occur in the equations defining f. These equations can easily be oriented and turned into a TRS $\mathcal{R}_f$ over a signature containing the constant 0, the successor S, symbols for the occuring basic functions, and symbols like $\mathsf{Sub}_{g,h_1,\ldots,h_m}$ and $\mathsf{PRec}_{g,h}$ representing the composed functions occurring in the definition of f. For example, the n-ary zero function 0_n (with $n > 0$) is simulated via $0_n(x_1, \ldots, x_n) \to 0$, substitutions are managed by

$$\mathsf{Sub}_{g,h_1,\ldots,h_m}(\bar{x}) \to g(h_1(\bar{x}), \ldots, h_m(\bar{x}))\ ,$$

and finally primitive recursion is handled by the two rules

$$\begin{aligned} \mathsf{PRec}_{g,h}(0, \bar{x}) &\to g(\bar{x})\ , \\ \mathsf{PRec}_{g,h}(S(y), \bar{x}) &\to h(y, \bar{x}, \mathsf{PRec}_{g,h}(y, \bar{x}))\ . \end{aligned}$$

It is immediate that $\mathcal{R}_f$ computes f using only one successor symbol. Termination is provided by any MPO which is based on a precedence $\prec$ satisfying

$$0^n \succ 0, \quad \mathsf{Sub}_{g,h_1,\ldots,h_m} \succ g, h_1, \ldots, h_m, \quad \text{and} \quad \mathsf{PRec}_{g,h} \succ g, h\ .$$

Note that these conditions can be regarded as a kind of subterm property. □

Corollary 6.5. *If a TRS $\mathcal{R}$ terminates via MPO, then its shortest size complexity* $\mathrm{SDc}_{\mathcal{R}}$ *has a primitive recursive bound. This result is essentially optimal.*

Just as MPO is related to primitive recursion, LPO is connected to multiple recursion. We omit the proof because it very much resembles the proof of Theorem 6.4 (though it relies on Theorem 5.7). It is a nice exercise to check that any k-ary Ackermann function (see Definition 2.65) can be transformed into a TRS terminating via LPO.

Theorem 6.6 (Weiermann 1995). $\textsc{Comp}(\mathrm{LPO}) = \textsc{Comp}_1(\mathrm{LPO}) = \textsc{Mrec}$.

Corollary 6.7. *If a TRS $\mathcal{R}$ terminates via LPO, then its shortest size complexity $\mathrm{SDc}_{\mathcal{R}}$ has a multiple recursive bound. This result is essentially optimal.*

6.3 Computability via KBO

Recently the computational power of certain restrictions of KBO has been investigated. Aiming at small complexity classes, Bonfante (2000) considered computation via KBO(1) and KBO$^-$ (which was introduced in Definition 3.59) using a model of computation which significantly differs from ours. According to Theorem 5.8.v,iv, the size complexity of a TRS terminating via either KBO(1) or KBO$^-$ is in $2^{O(n)}$. Bonfante (2000) showed that for his model of computation these two restrictions of KBO are separable. In fact, in this model the functions computable under KBO(1) are just the members of Linspace, while the collection of functions computable under KBO$^-$ coincides with Etime.

We intend to classify computation via (unrestricted) KBO. In analogy with the observations made above concerning computation via PT, the need for two distinct successor symbols figures prominently in the following result.

Proposition 6.8. *Let $f\colon \mathbb{N}^l \to \mathbb{N}$ be a function which is computable (with input successor S and output successor P) via KBO.*

i. *If S is special and P is not, then f is bounded by a constant function.*
ii. *If P is not special, then f is bounded by a linear function.*
iii. *If both S and P are special, then f has a primitive recursive bound.*
iv. *There is an upper bound on f in $\mathrm{Ack}(O(n), 0)$.*

Proof. Let $\mathcal{R}$ be a TRS over signature Σ which computes f and terminates via KBO (based on the weight μ). There is an $F \in \Sigma^{(l)}$ such that, for all $n_1, \ldots, n_l$,

$$F(S^{n_1}0, \ldots, S^{n_l}0) \xrightarrow{*} P^{f(n_1,\ldots,n_l)}0'$$

holds. Weight considerations and (3.7) yield

$$(n_1 + \cdots + n_l) \cdot \mu(S) + \mu(F) + l \cdot \mu(0) - \mu(0') \geqslant f(n_1, \ldots, n_l) \cdot \mu(P) \ .$$

If P is not special, then we have $\mu(P) > 0$. This immediately imposes a linear upper bound on f. We even end up with a constant bound on f if additionally S is supposed to be special. These observations settle (i) and (ii).

Let us turn to a different approach. Define $g\colon \mathbb{N}^l \to \mathbb{N}$ by

$$g(n_1, \dots, n_l) := \mathrm{dl}_{\mathcal{R}}(F(S^{n_1}0, \dots, S^{n_l}0)) \ .$$

According to Theorem 3.58.v $\mathcal{R}$ is nonduplicating, hence we can incorporate Lemma 3.70 and find $a \in \{0, 1\}$ and $b \in \mathbb{N}$ with

$$f(n_1, \dots, n_l) \leqslant 1 + a \cdot (n_1 + \dots + n_l) + b \cdot g(n_1, \dots, n_l) \ ,$$

and, provided that S and P differ, $a = 0$. Theorem 5.16 establishes (iv), as it gives us an upper bound on g which can be found in $\mathrm{Ack}(O(n), 0)$. For the special S considered in (iii) we take a closer look and define $h\colon \mathbb{N}^l \to \mathbb{N}$ by

$$h(n_1, \dots, n_l) := \mathcal{I}(F(S^{n_1}0, \dots, S^{n_l}0))$$

with $\mathcal{I}$ being the interpretation used in the proof of Theorem 5.15. Because all terms S^n0 share the same weight, a close inspection of Definition 5.12 (where $\mathcal{I}$ is introduced) shows the function $n \mapsto \mathcal{I}(S^n0)$ is primitive recursive. This can be used to show h is primitive recursive, too. As h is an upper bound on g, we have a primitive recursive upper bound on f. □

Taking a less direct approach we get stronger results.

Proposition 6.9. *We have*

i. $\textsc{Comp}(\text{KBO}) \subseteq \textsc{Atime}$ *, and*

ii. $\textsc{Comp}_1(\text{KBO}) \subseteq \textsc{Prec}$.

Proof. We just mentioned that computability via KBO implies computability by a TRS with timebound in $\mathrm{Ack}(O(n), 0)$. As $\mathrm{Ack}(O(n), 0)$ accommodates exponentiation (see Lemma 2.91), Theorem 3.68 yields computability with timebound in $\mathrm{Ack}(O(n), 0)$, hence membership in $\textsc{Atime}$.

If only one successor symbol is present, then we may proceed along the same results to get computability with timebound in $\textsc{Prec}$. Corollary 2.94 says this implies membership in $\textsc{Prec}$. □

As we intend to compute nonelementary functions and to even go beyond primitive recursion, Proposition 6.8 tells us we have to focus on the case where S is not special but P is, and a combination of Theorem 5.8.v and the fact that any TRS terminating via KBO is nonduplicating forces us to work with a signature containing at least one symbol with arity greater than 1, even if we compute a unary function. Theorem 6.20 below shows this is the right choice, but before we can present it, we have to establish various auxiliary results.

6.3.1 List Operations Compatible with KBO

The ability of KBO to encode certain operations on lists of natural numbers was used by Hofbauer and Lautemann (1989) to show that the size complexity $\mathrm{Dc}_{\mathcal{R}}$ of a certain TRS $\mathcal{R}$ terminating via KBO cannot be bounded by a primitive recursive function. We will treat a similar TRS in Section 6.3.3, but to do so we have to introduce list operations which are compatible with KBO in a more general setting.

We work with a signature Σ containing at least a binary symbol $\circ$ of weight 0 and a constant e of weight 1. For terms $s_0, \dots, s_n \in \mathcal{T}(\Sigma, \mathcal{V})$ we put

$$[\,] := e \quad \text{and} \quad [s_0, \dots, s_n] := s_0 \circ [s_1, \dots, s_n] \ . \tag{6.1}$$

The *length* of the *list* $[s_1, \dots, s_n]$ is n. By $[s_1, \dots, s_n] * t$ we denote the *concatenation* of the list $[s_1, \dots, s_n]$ and the term t, which we get via

$$[\,] * t := t \quad \text{and} \quad [s_0, \dots, s_n] * t := s_0 \circ ([s_1, \dots, s_n] * t) \ .$$

It is easy to see that, as soon as t is a list, concatenation produces a new list.

We introduce rules operating on these lists. For $t_0, \dots, t_n \in \mathcal{T}(\Sigma, \mathcal{V})$ we may consider the rule $[s_0, \dots, s_n] \to [t_0, \dots, t_n]$. This transforms a list into a list. Due to the structure of (6.1), the rule is also applicable to longer lists. To emphasize this we will introduce such a rule by

$$[\dots, s_0, \dots, s_n] \to [\dots, t_0, \dots, t_n] \ . \tag{6.2}$$

Similarly, we can consider rules operating somewhere within the list like

$$[\dots, s_0, \dots, s_n, \dots] \to [\dots, t_0, \dots, t_n, \dots] \tag{6.3}$$

by concatenating both $[s_0, \dots, s_n]$ and $[t_0, \dots, t_n]$ with a new variable. Comparable rules which operate solely at the top of lists are possible only if there is a symbol above the list which is, unlike $\circ$ and e, not used to construct lists. For example, with a binary symbol $\diamond$ at our disposal, the rule $e \diamond [s_0, \dots, s_n] \to e \diamond [t_0, \dots, t_n]$ is not applicable at arbitrary positions within a list but only at the top of the list, and additionally the list has to be of fixed length $n + 1$. In the same way, rules like $e \diamond [s_0, \dots, s_n, \dots] \to e \diamond [t_0, \dots, t_n, \dots]$ are possible.

Definition 6.10. For a given KBO $\prec_{\mathsf{kbo}}$ on $\mathcal{T}(\Sigma, \mathcal{V})$ a rule $l \to r$ is *lexicographic* if it is built like (6.2) or (6.3) and additionally satisfies these conditions:

- For all $x \in \mathcal{V}$ we have $1 \geqslant |l|_x \geqslant |r|_x$.
- For all $j, k \in [0, n]$, the weights $\mu(s_j)$ and $\mu(t_k)$ coincide.
- We have $(s_0, \dots, s_n) \succ_{\mathsf{kbo}}^{\mathsf{lex}} (t_0, \dots, t_n)$.

Under these conditions the weights of l and r are equal. An induction on the n in (6.2) or (6.3) proves the following Lemma.

Lemma 6.11. *If $l \to r$ is lexicographic for $\prec_{\mathsf{kbo}}$, then we have $l \succ_{\mathsf{kbo}} r$.*

From now on, we will restrict ourselves to two kinds of lists, for whom each element has weight 1. Lists of the first kind are those containing only natural numbers. Here we require the special symbol i to be an element of Σ, and the list elements we consider are called *number terms* and are either of the kind $i^a e$, encoding (and displayed as) $a \in \mathbb{N}$, or $i^a x$ with $x \in \mathcal{V}$, which we will write as $x + a$. The comparison of number terms is completely described by

$$\begin{aligned} a \succ_{\mathsf{kbo}} b &\iff a > b\ , \\ x + a \succ_{\mathsf{kbo}} b &\iff a > b\ , \\ x + a \succ_{\mathsf{kbo}} y + b &\iff x = y \wedge a > b\ . \end{aligned}$$

For example, the following rules, which will occur again later, are lexicographic:

$$\begin{aligned} [\dots, x+1, y, \dots] &\to [\dots, x, y+2, \dots] \\ [\dots, x+1, y, z, \dots] &\to [\dots, x, z, y, \dots]\ . \end{aligned}$$

Lists of the second kind are called *attributed lists.* Their definition is based on a set $A \subseteq \Sigma^{(0)} \setminus \{e\}$ of *attributes*, each attribute having weight 1. An attributed list is of even length and contains an alternating sequence of number terms and attributes, starting with a number term. Since attributes are constants, their comparison is determined by the underlying precedence. For attributes p and q we have

$$\mathrm{p} \succ_{\mathsf{kbo}} \mathrm{q} \iff \mathrm{p} \succ \mathrm{q}\ .$$

By convention, attributes are written as upright subscripts to the number terms, hence we write $(x+1)_{\mathrm{p}}$, or shorter $x + 1_{\mathrm{p}}$, instead of $x + 1, \mathrm{p}$.

Here are two examples of rules with attributes p and q:

$$\begin{aligned} [\dots, x + 4_{\mathrm{q}}, y_{\mathrm{q}}, \dots] &\to [\dots, x + 3_{\mathrm{p}}, y + 42_{\mathrm{p}}, \dots]\ , \\ [\dots, x + 1_{\mathrm{p}}, 0_{\mathrm{q}}, \dots] &\to [\dots, x + 1_{\mathrm{q}}, 17_{\mathrm{p}}, \dots]\ . \end{aligned}$$

The first rule is lexicographic regardless of $\succ$, whereas the second rule is lexicographic only if $\mathrm{p} \succ \mathrm{q}$. We mention without proof that it is possible to simulate attributed lists by unattributed lists – at the cost of simplicity.

6.3.2 Simulating a Timebounded RM

In this and the following subsections we will often use s^n as an abbreviation of n subsequent occurrences of the (possibly attributed) number term s.

A program for a timebounded RM can easily be simulated by a TRS consisting of lexicographic rules over lists of natural numbers. As the program is deterministic, the simulating derivations can be arranged to be linear.

Lemma 6.12. *If $f\colon \mathbb{N}^l \to \mathbb{N}$ is computable on a k-RM with timebound g and program $\mathcal{P} = (a_0, \dots, a_m)$, then there exists a TRS $\mathcal{R}_f$ over the signature $\Sigma = \{i, \circ, e\}$ such that $\mathcal{R}_f$ terminates via KBO and, for all $n_1, \dots, n_l$, we have*

$$[g(n_1 + \cdots + n_l), 0, n_1, \dots, n_l, 0^{k-l}] \xrightarrow{\triangleright} [0, m+1, f(n_1, \dots, n_l), 0^{k-1}] \ .$$

Proof. We show how to transform each instruction a_p of the program into one or two rules. If $a_p = (j, +, q)$, then we enrich $\mathcal{R}_f$ by the rule

$$[\dots, z+1, p, x_1, \dots, x_j, \dots, x_k] \to [\dots, z, q, x_1, \dots, x_j + 1, \dots, x_k] \ .$$

To treat the case $a_p = (j, -, q)$ we add the rules

$$\begin{aligned} [\dots, z+1, p, x_1, \dots, x_j + 1, \dots, x_k] &\to [\dots, z, q, x_1, \dots, x_j, \dots, x_k] \ , \\ [\dots, z+1, p, x_1, \dots, 0, \dots, x_k] &\to [\dots, z, q, x_1, \dots, 0, \dots, x_k] \end{aligned}$$

to $\mathcal{R}_f$, and if $a_p = (j, q, r)$ then $\mathcal{R}_f$ is extended by the rules

$$\begin{aligned} [\dots, z+1, p, x_1, \dots, x_j + 1, \dots, x_k] &\to [\dots, z, r, x_1, \dots, x_j + 1, \dots, x_k] \ , \\ [\dots, z+1, p, x_1, \dots, 0, \dots, x_k] &\to [\dots, z, q, x_1, \dots, 0, \dots, x_k] \ . \end{aligned}$$

We need one additional rule to simulate the halting condition:

$$[\dots, z+1, m+1, x_1, \dots, x_k] \to [\dots, 0, m+1, x_1, \dots, x_k] \ .$$

All rules of $\mathcal{R}_f$ are lexicographic, hence, by Lemma 6.11, $\mathcal{R}_f$ terminates via KBO. Each transition corresponds to exactly one rewrite step, and, as the program is deterministic, the considered derivations are linear. □

This transformation is also applicable to nondeterministic programs, but of course we can no longer guarantee that derivations are linear.

6.3.3 Long Linear Derivations

Hofbauer and Lautemann (1989) constructed a TRS terminating via KBO whose size complexity is in $\mathrm{Ack}(O(n), 0)$ but has no primitive recursive bound. Later Hofbauer (1991) introduced an improved version of this, which we call $\mathcal{H}$. It (mainly) contains the rules

$$\begin{aligned} &(\mathrm{H1}) \quad [\dots, x+1, y, \dots] \to [\dots, x, y+2, \dots] \\ &(\mathrm{H2}) \quad [\dots, x+1, y, z, \dots] \to [\dots, x, z, y, \dots] \ , \end{aligned}$$

which are, as we already know, lexicographic.

Theorem 6.13 (Hofbauer 1991, 5.9). *The TRS $\mathcal{H}$ terminates via KBO, and there is no primitive recursive upper bound on* $\mathrm{Dc}_{\mathcal{H}}$.

Proof. From Lemma 6.11 we learn termination via KBO. The shape of the rules makes it possible to reduce in layers. If we have $[a, 0^n, 0] \xrightarrow{*} [0, 0^n, a']$ then, using the same rewrite rules at same positions, we also get $[a + b, 0^n, 0] \xrightarrow{*} [b, 0^n, a']$. We already mentioned that for lists s and t with $t \xrightarrow{*} t'$ we have $s * t \xrightarrow{*} s * t'$, too. Both observations will be used below. Now we define $h\colon \mathbb{N}^2 \to \mathbb{N}$ by

$$h(n, m) := \max\{k : [2(m + 1), 0^{n+1}] \xrightarrow{*}_{\mathcal{H}} [0^{n+1}, k + 1]\}$$

and intend to show

$$(\forall n, m)(h(n, m) \geqslant \mathrm{Ack}(n, m)) \ . \tag{6.4}$$

Because of $[2(m+1), 0] \xrightarrow{*}_{\mathrm{H1}} [0, 4(m+1)]$ we have $h(0, m) \geqslant m+1$, and similarly $[2, 0^{n+2}] \xrightarrow{*}_{\mathrm{H1}} [0, 4, 0^{n+1}]$ yields $h(n + 1, 0) \geqslant h(n, 1)$. The derivation

$$\begin{aligned}
[2(m + 2), 0^{n+2}] = {} & [2 + 2(m + 1), 0^{n+2}] \\
\xrightarrow{+}_{\mathcal{H}} {} & [2, 0^{n+1}, h(n + 1, m) + 1] \\
\xrightarrow{*}_{\mathrm{H1}} {} & [1^n, 2, 0, h(n + 1, m) + 1] \\
\to_{\mathrm{H2}} {} & [1^{n+1}, h(n + 1, m) + 1, 0] \\
\xrightarrow{+}_{\mathrm{H1}} {} & [1^{n+1}, 0, 2(h(n + 1, m) + 1)] \\
\xrightarrow{+}_{\mathrm{H2}} {} & [0, 2(h(n + 1, m) + 1), 0^{n+1}] \\
\xrightarrow{+}_{\mathcal{H}} {} & [0^{n+2}, h(n, h(n + 1, m)) + 1]
\end{aligned}$$

implies $h(n+1, m+1) \geqslant h(n, h(n+1, m))$, hence we established (6.4). A close inspection of these derivations also shows

$$\mathrm{dl}_{\mathcal{H}}([2(m + 1), 0^{n+1}]) \geqslant \mathrm{Ack}(n, m) \ . \tag{6.5}$$

From this, $|[2(m + 1), 0^{n+1}]| = 2(n + 2) + 1 + 2(m + 1)$, and Lemma 2.63.iv one easily infers $\mathrm{Dc}_{\mathcal{H}}$ has no primitive recursive upper bound. □

The time has come to give the long deferred proof of Lemma 5.9. It states that, for any $a \in \mathbb{N}$, there is a TRS $\mathcal{R}$ terminating via KBO such that we have

$$(\forall n \geqslant 3)(\mathrm{Dl}_{\mathcal{R}}(n) > \mathrm{Ack}(a^n, 0)) \ .$$

Proof of Lemma 5.9. We may suppose $a \geqslant 2$. This time the signature Σ contains the symbols used for building lists of number terms and also the a-ary f. The TRS $\mathcal{R}$ consists of the two rules from $\mathcal{H}$ and the three rules

$$\begin{aligned}
&(\mathrm{H3}) && (x \circ y) \circ z \to x \circ (y \circ z) \\
&(\mathrm{H4}) && [\dots, x, y, z, \dots] \to [\dots, y + 2, z, \dots] \\
&(\mathrm{H5}) && f(x_1, \dots, x_a) \to [0^{a^2-a}, x_1, \dots, x_a] \ ,
\end{aligned}$$

where (H4) includes a little abuse of notation. With (H3) we can reduce terms over $\{\circ, e\}$ to lists of zeros, while (H4) will be used once to reduce $[0^{n+2}]$ to $[2, 0^n]$. Via (H5) the high arity of the symbol f allows for generating a list of length larger than a^n out of a term of depth n.

$\mathcal{R}$ terminates via NKBO for special i, $\mu(e) := 1$, $\mu(\circ) := 0$ and $\mu(f) := a^2$, since for (H5) we may use $\mu([k_1, \ldots, k_n]) = n + 1$. Thus $\mathrm{dl}_{\mathcal{R}}$ is well-defined, and a quick glance at (6.5) shows

$$\mathrm{dl}_{\mathcal{R}}([2(m+1), 0^{n+1}]) \geqslant \mathrm{Ack}(n, m) \ . \tag{6.6}$$

For $n > 0$ we recursively define terms s_n by $s_1 := e$ and $s_{n+1} := f(s_n, \ldots, s_n)$. Obviously $\mathrm{dp}(s_n) = n$, and an induction on $n \geqslant 2$ shows $|s_n|_f \geqslant a^{n-2} + n - 2$. Using (H5) as often as possible we arrive at $s_n \xrightarrow{*} t_n$ for some $t_n \in \mathcal{T}(\{e, \circ\})$. Since each application of (H5) introduces a^2 occurrences of $\circ$, for $n \geqslant 2$ we get $|t_n|_\circ = |s_n|_f \cdot a^2 \geqslant a^n + a^2(n-2)$. Because the rule (H3) only rearranges the symbols of a term, using it as often as possible transforms t_n into a list of zeros of length $|t_n|_\circ$, thus $t_n \xrightarrow{*} [0^{|t_n|_\circ}]$. Applying (H4) once we arrive at $[2, 0^{|t_n|_\circ - 2}]$. This yields

$$\mathrm{dl}_{\mathcal{R}}(s_n) > \mathrm{dl}_{\mathcal{R}}\,[2, 0^{|t_n|_\circ - 2}] \geqslant \mathrm{Ack}(|t_n|_\circ - 3, 0)$$

by (6.6). Any $n \geqslant 3$ satisfies $|t_n|_\circ - 3 > a^n$. □

Although derivations in $\mathcal{H}$ may get very long, there are strategies leading quickly to normal forms. If we start with some term s, then any strategy which applies (H1) only if (H2) is not applicable arrives in less than $|s|$ steps at a normal form. Thus $\mathrm{SDc}_{\mathcal{H}}$ is bounded by a linear function. This prevents us from using the output of $\mathcal{H}$ as a timebound for some sort of abstract machine.

Our aim is to transform $\mathcal{H}$ into a TRS $\mathcal{A}$ operating on attributed lists. Any $\mathcal{A}$-derivation starting from the term representing $[2(m+1), 0^{n+1}]$ is to be very long and *linear*. This transformation is a bit involved. We use attributed lists over the attribute set

$$A := \{\mathrm{a}, \mathrm{b}, \mathrm{d}, \mathrm{f}, \mathrm{p}, \mathrm{r}, \mathrm{s}, \mathrm{w}\} \ .$$

The attributes are abbreviations for “active”, “back”, “double”, “fetch”, “passive”, “return”, “stop”, and “wait”. Since usually most of the elements of our lists are passive, we will leave out the attribute p. The attributes p, s, and w are called *sleeping*, while the other attributes are *awake*. A (possibly empty) list whose numbers have only sleeping attributes is also called *sleeping*, and a list is *good* if it contains exactly one number having an awake attribute.

We intend to work with good lists, using the only awake attribute to mark the position of the only possible rewrite step. Under these circumstances, our choice of the rewrite rules of $\mathcal{A}$ will force derivations to be linear. The rules of $\mathcal{A}$ are displayed in Table 6.1 on the facing page. It is admittedly not that easy to see the fifteen rules of $\mathcal{A}$ are closely related to the two rules of $\mathcal{R}$.

$$\begin{array}{rr}
(1) & [\dots, 1_{\mathrm{a}}, y] \to [\dots, 0, y + 2_{\mathrm{a}}] \\
(2) & [\dots, x + 2_{\mathrm{a}}, y] \to [\dots, x + 1_{\mathrm{a}}, y + 2] \\
(3) & [\dots, 1_{\mathrm{d}}, y] \to [\dots, 0, y + 2_{\mathrm{d}}] \\
(4) & [\dots, x + 2_{\mathrm{d}}, y] \to [\dots, x + 1_{\mathrm{d}}, y + 2] \\
(5) & [\dots, x + 1_{\mathrm{a}}] \to [\dots, x + 1_{\mathrm{r}}] \\
(6) & [\dots, 0, x_{\mathrm{r}}, \dots] \to [\dots, 0_{\mathrm{r}}, x, \dots] \\
(7) & [\dots, x + 2, 0_{\mathrm{r}}, y + 1] \to [\dots, x + 1, y + 1_{\mathrm{d}}, 0] \\
(8) & [\dots, x + 1, 0, y + 1_{\mathrm{d}}] \to [\dots, x, y + 1_{\mathrm{a}}, 0] \\
(9) & [\dots, x + 2, 0_{\mathrm{r}}, 0, \dots] \to [\dots, x + 1_{\mathrm{s}}, 2_{\mathrm{f}}, 0, \dots] \\
(10) & [\dots, 2_{\mathrm{f}}, 0, 0, \dots] \to [\dots, 1_{\mathrm{w}}, 2_{\mathrm{f}}, 0, \dots] \\
(11) & [\dots, 2_{\mathrm{f}}, 0, x + 1] \to [\dots, 1_{\mathrm{w}}, x + 1_{\mathrm{d}}, 0] \\
(12) & [\dots, 1_{\mathrm{w}}, 0, x + 1_{\mathrm{d}}] \to [\dots, 1_{\mathrm{w}}, 0, x + 1_{\mathrm{b}}] \\
(13) & [\dots, 1_{\mathrm{w},} 0, x + 1_{\mathrm{b}}, \dots] \to [\dots, 0, x + 1_{\mathrm{b}}, 0, \dots] \\
(14) & [\dots, x + 1_{\mathrm{s}}, 0, y + 1_{\mathrm{b}}, \dots] \to [\dots, x, y + 1_{\mathrm{a}}, 0, \dots] \\
(15) & [\dots, x + 2_{\mathrm{a}}, 0, 0, \dots] \to [\dots, x, 4_{\mathrm{a}}, 0, \dots]\ .
\end{array}$$

Table 6.1: *The rules of $\mathcal{A}$*

Lemma 6.14. *$\mathcal{A}$ terminates via KBO.*

Proof. By Lemma 6.11 it suffices to find a precedence which makes all rules lexicographic. Except for (5), (6), and (12), the rules are already lexicographic regardless of the precedence. The three remaining rules are lexicographic for any precedence $\prec$ satisfying $\mathrm{a} \succ \mathrm{r}$, $\mathrm{p} \succ \mathrm{r}$ and $\mathrm{d} \succ \mathrm{b}$. □

Lemma 6.15. *If $s \xrightarrow{+} t$ and s is a good list, then t is a good list and $s \xrightarrow{\rhd} t$ holds. For any sleeping list s' we have $s' * s \xrightarrow{\rhd} s' * t$.*

Proof. The rules of $\mathcal{A}$ do not modify the amount of awake attributes, so t is a good list. All rules require the presence of an awake attribute, and a moment's reflection shows that for good lists the rules do not overlap. Hence derivations of good lists are linear. Now the statement involving s' easily follows, as the rewrite relation is closed under contexts and as $s' * s$ is a good list. □

Lemma 6.16. *For $b, c, k \in \mathbb{N}$ we have*

i. $[b_{\mathrm{a}}, c] \xrightarrow{+} [0, c + 2b_{\mathrm{a}}]$ *if* $b > 0$,

ii. $[b_{\mathrm{d}}, c] \xrightarrow{+} [0, c + 2b_{\mathrm{d}}]$ *if* $b > 0$,

iii. $[0^{k+1}, b_{\mathrm{r}}] \xrightarrow{+} [0_{\mathrm{r}}, 0^k, b]$, *and*

iv. $[b + 2, 0^{k+1}, c_{\mathrm{a}}] \xrightarrow{+} [b, 2c_{\mathrm{a}}, 0^{k+1}]$ *if* $c > 0$.

Proof. We establish (i) by induction on $b > 0$, starting with $[1_{\mathrm{a}}, c] \to_1 [0, c+2_{\mathrm{a}}]$. For $b \geqslant 2$ the induction hypothesis yields

$$[b_{\mathrm{a}}, c] = [(b-2) + 2_{\mathrm{a}}, c] \to_2 [b - 1_{\mathrm{a}}, c+2] \xrightarrow{+} [0, c + 2 + 2(b-1)_{\mathrm{a}}]\ .$$

The point (ii) is shown in the same way (using (3) and (4)). An iterated application of the rule (6) yields (iii). For (iv) we fall back on (iii) and see

$$[b+2, 0^{k+1}, c_{\mathrm{a}}] \to_5 [b+2, 0^{k+1}, c_{\mathrm{r}}] \xrightarrow{+} [b+2, 0_{\mathrm{r}}, 0^k, c] =: t\ .$$

If $k = 0$, then we may infer

$$t = [b+2, 0_{\mathrm{r}}, c] \to_7 [b+1, c_{\mathrm{d}}, 0] \xrightarrow{+} [b+1, 0, 2c_{\mathrm{d}}] \to_8 [b, 2c_{\mathrm{a}}, 0]\ ,$$

relying on (ii) (and Lemma 6.15), and otherwise we have $k = l+1$. Here we get

$$\begin{array}{llllll} t = & [b+2, 0_{\mathrm{r}}, 0^k, c] & \to_9 & [b + 1_{\mathrm{s}}, 2_{\mathrm{f}}, 0^k, c] & \xrightarrow{*}_{10} & [b+1_{\mathrm{s}}, 1^l_{\mathrm{w}}, 2_{\mathrm{f}}, 0, c] \\ \to_{11} & [b+1_{\mathrm{s}}, 1^k_{\mathrm{w}}, c_{\mathrm{d}}, 0] & \xrightarrow{+} & [b+1_{\mathrm{s}}, 1^k_{\mathrm{w}}, 0, 2c_{\mathrm{d}}] & & \text{via (ii)} \\ \to_{12} & [b+1_{\mathrm{s}}, 1^k_{\mathrm{w}}, 0, 2c_{\mathrm{b}}] & \xrightarrow{+}_{13} & [b+1_{\mathrm{s}}, 0, 2c_{\mathrm{b}}, 0^k] & \to_{14} & [b, 2c_{\mathrm{a}}, 0^{k+1}]\ . \end{array}$$

□

Lemma 6.17. *Let $n, m, k \in \mathbb{N}$ with $n \geqslant 1$.*

i. *There is $p > \mathrm{Ack}(0, m)$ such that $[2(m+1)_{\mathrm{a}}, 0] \xrightarrow{+} [0, p_{\mathrm{a}}]$.*
ii. *There is $p > \mathrm{Ack}(n, m)$ with $[k + 2(m+1)_{\mathrm{a}}, 0^{n+1}] \xrightarrow{+} [k, 0^n, p_{\mathrm{a}}]$.*

Proof. For (i) we rely on Lemma 6.16.i to get $[2(m+1)_{\mathrm{a}}, 0] \xrightarrow{+} [0, 4(m+1)_{\mathrm{a}}]$ and observe $4(m+1) > m+1 = \mathrm{Ack}(0, m)$.

We establish (ii) by induction on n and secondary induction on m. If $m = 0$, then there is some $p > \mathrm{Ack}(n-1, 1) = \mathrm{Ack}(n, 0)$ such that

$$[k + 2_{\mathrm{a}}, 0^{n+1}] \to_{15} [k, 4_{\mathrm{a}}, 0^n] = [k, 0 + 2 \cdot (1+1)_{\mathrm{a}}, 0^n] \xrightarrow{+} [k, 0^n, p_{\mathrm{a}}]$$

holds. For $n = 1$ the last step is possible due to (i), while for $n > 1$ we may use the induction hypothesis. In the case $m = m' + 1$ we get

$$\begin{array}{ll} \ [k + 2(m+1)_{\mathrm{a}}, 0^{n+1}] & \\ = [k + 2 + 2(m'+1)_{\mathrm{a}}, 0^{n+1}] & \\ \xrightarrow{+} [k+2, 0^n, q_{\mathrm{a}}] & \text{for some } q > \mathrm{Ack}(n, m') \text{ by s.i.h.} \\ \xrightarrow{+} [k, 2q_{\mathrm{a}}, 0^n] & \text{by Lemma 6.16.iv} \\ = [k, 0 + 2((q-1)+1)_{\mathrm{a}}, 0^n] & \\ \xrightarrow{+} [k, 0^n, p_{\mathrm{a}}] & \end{array}$$

for some $p > \mathrm{Ack}(n-1, q-1) \geqslant \mathrm{Ack}(n-1, \mathrm{Ack}(n, m')) = \mathrm{Ack}(n, m)$. Again, the last rewrite step is legal by (i) and the induction hypothesis. □

Proposition 6.18. *Given n and m, we have $[2(m+1)_\mathrm{a}, 0^{n+1}] \overset{\rhd}{\to}_{\mathcal{A}} [0_\mathrm{r}, 0^n, p]$ for some $p > \mathrm{Ack}(n, m)$.*

Proof. By Lemma 6.17 there is a $p > \mathrm{Ack}(n, m)$ such that $[2(m+1)_\mathrm{a}, 0^{n+1}] \overset{+}{\to} [0, 0^n, p_\mathrm{a}] \to_5 [0, 0^n, p_\mathrm{r}] \overset{+}{\to} [0_\mathrm{r}, 0^n, p]$ holds. Lemma 6.16.iii justifies the last step, and Lemma 6.15 ensures the linearity of the derivation. □

We combine this with Theorem 5.16 and Lemma 6.14.

Theorem 6.19. *If a TRS $\mathcal{R}$ terminates via KBO, then its shortest size complexity $\mathrm{SDc}_{\mathcal{R}}$ is a member of $\mathrm{Ack}(O(n), 0)$. This result is essentially optimal.*

Let us consider a terminating TRS $\mathcal{R}$. We mentioned before that, in some sense, $\mathrm{Dc}_{\mathcal{R}}$ measures the worst-case behavior of $\mathcal{R}$, whereas $\mathrm{SDc}_{\mathcal{R}}$ measures its best-case behavior. Often $\mathrm{SDc}_{\mathcal{R}}$ is in a smaller complexity class than $\mathrm{Dc}_{\mathcal{R}}$. However, if $\mathcal{R}$ is able to compute a function f in some fair way, then f may be transformed into a lower bound for $\mathrm{SDc}_{\mathcal{R}}$.

If we consider termination via either MPO or LPO, the tight lower bounds on the size complexities established in Corollary 6.5 and Corollary 6.7 stem from TRSs computing primitive resp. multiple recursive functions. The bound for KBO we present in Theorem 6.19 is reached by a somewhat different approach. We do not calculate the function behind the p of Proposition 6.18, instead, knowing p exceeds $\mathrm{Ack}(n, m)$ suffices. We will soon establish a computability result for KBO which is very similar to those dealing with MPO and LPO. The tool behind this result is our TRS $\mathcal{A}$ which will serve as the engine of a RM.

6.3.4 Hard Computation via KBO

We are about to leave the realm of primitive recursion. Proposition 6.8 forces us to focus on computation with a nonspecial input successor but a special output successor. The following Theorem shows this is the right choice. If we restrict ourselves to computation using only one successor symbol, then we still get quite far, provided that the successor symbol is special.

Theorem 6.20. *Let $f \colon \mathbb{N}^l \to \mathbb{N}$.*

i. *If f is computable on a RM with timebound from $\mathrm{Ack}(O(n), 0)$, then f is computable via KBO.*
ii. *If f is computable on a RM in primitive recursive time, then f is computable via KBO using the special symbol as the only one successor.*

Proof. For expository reasons we treat the case $l = 1$ in detail. Below we will show how the approach can be extended to higher arities.

First we take care of (i), as (ii) is won by minor modifications of the TRS considered there. The premise and Lemma 2.88 yield the existence of a program $\mathcal{P} = (a_0, \dots, a_m)$, of b and $c > 0$ such that, for all n, $f(n)$ is computable by $\mathcal{P}$ in less than $\mathrm{Ack}(bn + c, 0)$ steps. The main task is the construction of a TRS $\mathcal{R}$ based on Lemma 6.12 and Proposition 6.18.

We throw into Σ the symbols occuring there (assuming that the symbols $\circ$ used for the two kinds of lists are distinct), and add to $\mathcal{R}$ the corresponding rewrite rules from $\mathcal{R}_f$ and $\mathcal{A}$. The unary symbols S, F, and G and a binary symbol $\diamond$ are also included in Σ. In order to do without too many parentheses we abbreviate $s\diamond(s'\diamond s'')$ by $s\diamond s'\diamond s''$. Our aim is to show $(\forall n)(F(S^n e) \overset{\triangleright}{\longrightarrow} i^{f(n)}e)$. Keeping this in mind, we adjoin to $\mathcal{R}$ the six rules

$$\begin{array}{lr@{\;\to\;}l}
\text{(A)} & F(w) & G\big(w \diamond [0_{\mathrm{p}}^{c+2}] \diamond [0^{k+1}]\big) \\
\text{(B)} & G\big(S(w) \diamond v \diamond [0, x, \dots]\big) & G\big(w \diamond ([0_{\mathrm{p}}^{b}] * v) \diamond [0, x+1, \dots]\big) \\
\text{(C)} & G\big(e \diamond [x_{\mathrm{p}}, \dots] \diamond v\big) & [x + 2_{\mathrm{a}}, \dots] \diamond v \\
\text{(D)} & [0_{\mathrm{r}}, x_{\mathrm{p}}, \dots] \diamond v & [x_{\mathrm{r}}, \dots] \diamond v \\
\text{(E)} & [x + 1_{\mathrm{r}}] \diamond v & [x + 1] * v \\
\text{(F)} & [\dots, 0, m+1, x_1, \dots, x_k] & x_1 \ ,
\end{array}$$

where we violate our convention and display the occurrences of the otherwise dropped attribute p. So, for example, the list on the right hand side of the rule (E) is not attributed. We use $*$ here twice as an abbreviation of a certain term and not as an operation which affords infinitely many rules. It is fairly easy to see that all we need are just two terms containing the two incarnations of $\circ$.

The new rules do not affect our results about $\mathcal{R}_f$ and $\mathcal{A}$, because, with the exception of (F), they use the new symbol $\diamond$, and the rule (F) can only be used after the computation done by $\mathcal{R}_f$ halted.

Given n, put $d := bn + c$. By Proposition 6.18 there is a $p > \mathrm{Ack}(d, 0)$ such that $[2_{\mathrm{a}}, 0^{d+1}] \xrightarrow{*} [0_{\mathrm{r}}, 0^d, p]$. Now the following (linear) derivation is possible:

$$\begin{aligned}
F(S^n e) &\to_{\mathrm{A}} G\big(S^n e \diamond [0^{c+2}] \diamond [0^{k+1}]\big) \\
&\xrightarrow{*}_{\mathrm{B}} G\big(e \diamond [0^{bn+c+2}] \diamond [0, n, 0^{k-1}]\big) \\
&\to_{\mathrm{C}} [2_{\mathrm{a}}, 0^{d+1}] \diamond [0, n, 0^{k-1}] && \text{(6.7a)} \\
&\xrightarrow{+} [0_{\mathrm{r}}, 0^d, p] \diamond [0, n, 0^{k-1}] && \text{by Proposition 6.18} \\
&\xrightarrow{+}_{\mathrm{D}} [p_{\mathrm{r}}] \diamond [0, n, 0^{k-1}] \\
&\to_{\mathrm{E}} [p, 0, n, 0^{k-1}] && \text{(6.7b)} \\
&\xrightarrow{+} [0, m+1, f(n), 0^{k-1}] && \text{by Lemma 6.12} \\
&\to_{\mathrm{F}} i^{f(n)} e \ .
\end{aligned}$$

We should make a few remarks here. The rules from $\mathcal{A}$ are not applicable until (6.7a) is reached, because before that the attributed lists are sleeping, and similarly the rules from $\mathcal{R}_f$ are useless until we arrive at (6.7b) because they operate on lists with at least $k+2$ elements. As the unattributed numbers are shared by both kinds of lists, the use of (E) is legal. It is routine to check that all presented derivations are linear and all terms $i^a e$ are in normal form.

In order to show that $\mathcal{R}$ terminates via KBO it remains to orient the six new rules, since the weights and precedences for $\mathcal{R}_f$ and $\mathcal{A}$ given in Lemmata 6.12 and 6.14 are compatible. The symbols $\diamond$ and G get the smallest possible weights, 0 and 1. Weight considerations imply the rules (C) to (F) are oriented correctly, and for the remaining two rules we may safely choose sufficiently large weights for F and S.

We turn to the treatment of (ii) and consider f computable on a k-RM in primitive recursive time. By Lemma 2.63.iv, for any unary primitive recursive timebound g there exists $c > 0$ such that $g(n) \leqslant \operatorname{Ack}(c, n)$ holds for all n.

Applying only two changes, we proceed along the lines of the previous proof. Proposition 6.8.ii shows that there is little hope to compute f via KBO with a nonspecial output successor. Therefore we have to replace S with i. The rule (B), which eliminates one S and increases the list length, is only admissible if the weight of S is positive. We replace this rule with a rule which eliminates one i and increases the first number of the attributed list by 2. This rule is oriented correctly by KBO, and using it we get

$$F(i^n e) \xrightarrow{+} [2(n+1)_{\mathrm{a}}, 0^{c+1}] \diamond [0, n, 0^{k-1}] =: t \;.$$

Proposition 6.18 yields a $p > \operatorname{Ack}(c, n)$ with $[2(n+1)_{\mathrm{a}}, 0^{c+1}] \xrightarrow{+} [0_{\mathrm{r}}, 0^c, p]$. Thus we arrive at $t \xrightarrow{*} i^{f(n)} e$. The linearity of these derivations is obvious.

It was promised to show how the above approach can be extend to higher arities. We focus on (i), the proof of (ii) is again very close to this. The main task here is to preserve the linearity of the derivations and to modify the rules from (A) to (C) in such a way that, for $d := b(n_1 + \cdots + n_l) + c$ and some $p > \operatorname{Ack}(d, 0)$, the derivation

$$F(S^{n_1} e, \ldots, S^{n_l} e) \xrightarrow{*} [0_{\mathrm{r}}, 0^d, p] \diamond [0, n_1, \ldots, n_l, 0^{k-l}]$$

is possible. Anything else remains unchanged. The replacement of (A) is

$$(\mathrm{A}') \quad F(w_1, \ldots, w_l) \to G\big(w_1 \diamond \cdots \diamond w_l \diamond [0_{\mathrm{p}}^{c+2}] \diamond [0^{k+1}]\big) \;,$$

while the rule (B) has to be split into l rules (B_j) which delete one S at the j^{th} position in G (with $\diamond$ separating these positions) and add list length and one special symbol. The rule (B_j) has to take care that it can only be applied if all rules ($\mathrm{B}_{j'}$) with $j' < j$ are no longer applicable. This is achieved by deleting the occurrences of S from left to right, preserving linearity of the derivation. Hence the rule (B_j) simply reduces

$$G\bigl(e \diamond \dots e \diamond S(w_j) \diamond w_{j+1} \diamond \dots \diamond w_l \diamond v \diamond [0, x_1, \dots, x_j, \dots]\bigr)$$

(with $j - 1$ leading e) to

$$G\bigl(e \diamond \dots e \diamond w_j \diamond \dots \diamond w_l \diamond ([0^b_{\mathrm{p}}] * v) \diamond [0, x_1, \dots, x_{j-1}, x_j + 1, \dots]\bigr) \ .$$

Finally we have to modify (C) via

$$(\mathrm{C}') \quad G\bigl(e \diamond \dots \diamond e \diamond [x_{\mathrm{p}}, \dots] \diamond v\bigr) \to [x + 2_{\mathrm{a}}, \dots] \diamond v \ ,$$

with l leading e on the left hand side of the rule. It is not difficult to see the new rules act just as intended. Termination via KBO of the new rules is established as before. □

Theorem 6.21. *We have*

i. Comp(KBO) = Atime *, and*

ii. Comp$_1$(KBO) = Prec *.*

Proof. One direction is Proposition 6.9, and the opposite direction is just a combination of Theorem 6.20, Lemmata 2.91 and 2.92, and Corollary 2.94. □

Functions computable by nondeterministic TRSs have been studied, for example, by Bonfante et al. (2001) and Bonfante (2000), see also Definition 6.22. Our results on KBO can easily be extended to this setting. This observation is only for the record, since the distinction between deterministic and nondeterministic programs is only important for the small complexity classes.

In term rewriting it is possible to take the alternative approach of computation on words or even terms instead of computation on numbers, as was done by Bonfante et al. (1999) and Bonfante (2000). A computation on words amounts to having many distinct input and output successor symbols. We know from Proposition 6.8 and Theorem 6.21 how important it is to use the unique special symbol as *the* output successor symbol when computing via KBO. Thus the different approach does not fit well to computation via unrestricted KBO. However, if one is interested in small complexity classes only, this approach is quite attractive, cf. Bonfante (2000).

6.4 Computability via Simple Termination

It is an open problem to characterize the functions computable via simple termination. Because our TRSs of Section 5.3.4 simulating the Hydra battle contain the embedding rules, it is not possible (well, at least not directly) to use them, like $\mathcal{A}$ in Section 6.3, for the linear generation of a large amount of symbols which serve as a counter for the number of program steps. This problem is worse than having to compute nondeterministic programs since here a computation may even stop at a term which is not a number. It should however be possible to reshape these TRSs, drop the embedding rules in favor of more complicated rules, and add symbols and rules for an effective comparison of encoded ordinals so that it is possible to compute a maximal subterm or to find out if a term is a fixed point for the ψ-function based on other terms. For this reason I conjecture that the functions computable via simple termination coincide with the $<\vartheta(\Omega^\omega)$-recursive functions.

It is of course also possible to consider nondeterministic computations. This affords a viable concept of function computable by a TRS as one has to cope with normal forms which do not look like P^m0', and one has to decide which of possibly many normal forms representing a number should be taken as *the* output. The following definition is in the spirit of Krentel (1988), Grädel and Gurevich (1995), and Bonfante et al. (2001).

Definition 6.22. We say $f\colon \mathbb{N}^l \to \mathbb{N}$ is *nondeterministically computable* by the terminating TRS $\mathcal{R}$ if there are $F \in \Sigma^{(l)}$, unary symbols S and P, and constants 0 and $0'$ such that, for all $n_1, \dots, n_l$, we have

$$f(n_1, \dots, n_l) = \max \{m : F(S^{n_1}0, \dots, S^{n_l}0) \xrightarrow{*}_{\mathcal{R}} P^m0' \text{ in normal form}\}.$$

If f is nondeterministically computable by a TRS terminating via a termination proof method $\mathcal{M}$, then f is *nondeterministically computable via* $\mathcal{M}$.

For this approach Bonfante et al. (2001) established the nondeterministic counterpart to Theorem 6.3: the functions nondeterministically computable via either PT-1, PT-L, or PT coincide with the respective members of the nondeterministic versions of PTIME, ETIME, and E$_2$TIME. Nondeterministic computation via simple termination corresponds to a set of functions we met before.

Theorem 6.23. *The functions which are nondeterministically computable via simple termination coincide with the* $<\vartheta(\Omega^\omega)$*-recursive functions.*

Proof (sketch). If a function is nondeterministically computable by the simply terminating TRS $\mathcal{R}$, then, according to Theorem 5.51, $\mathrm{Dc}_{\mathcal{R}}$ is dominated by a $<\vartheta(\Omega^\omega)$-recursive function. Just as nondeterministic computation (on TMs) can be transformed into (longer) deterministic computation, we can find another $<\vartheta(\Omega^\omega)$-recursive function which computes all possible derivations and outputs the maximum of the well-formed results.

$\mathcal{M}$	Comp($\mathcal{M}$)	established by	see Theorem
PT-1	Ptime	Bonfante et al. (1999)	6.3
PT-L	Etime		
PT	E_2time		
KBO	Atime	Lepper	6.21
MPO	Prec	Hofbauer (1991, 1992)	6.4
LPO	Mrec	Weiermann (1995)	6.6

Table 6.2: *Complexity classes characterized by the functions computable (using two successor symbols) via the termination proof method $\mathcal{M}$*

Let us turn to the opposite direction and consider a $<\vartheta(\Omega^\omega)$-recursive function f. This function is computable on some k-RM with a $<\vartheta(\Omega^\omega)$-recursive timebound g. From (the proof of) Corollary 5.47 and Theorem 5.50 we know that there is a TRS $\mathcal{R}$, simulating large enough Hydra battles, which is able to reduce an input uniformly depending on n to $\mathsf{S}^{h(n)}\mathsf{0}$ for some $h >_{\mathrm{d}} g$. Thus we are able to produce sufficiently many incarnations of S to control the simulation of a k-RM computing f. We have to take care of two main differences to Section 6.3.4.

First of all the program itself may be nondeterministic. We already mentioned below Lemma 6.12 that this is no real problem, as our transformation of a program into a TRS is easily extendible to the nondeterministic case.

The second difference is the nonlinearity of $\mathcal{R}$. It may well happen that the stock of symbols S runs empty before the simulation stops. This is taken care of by a rule like the rule (F) of Theorem 6.20, hence something like

$$[\ldots, 0, m+1, x_1, \ldots, x_k] \to x_1 \ .$$

We are not able to reduce to the contents of the output register unless the program reaches the terminal state $m + 1$. Thus the output of the derivation is not well-formed and is ignored in Definition 6.22. On the other hand, any correct output of the program will be output by the computation. □

The results presented in this chapter show that function classes well known from complexity theory occur as classes of functions computable via (restrictions of) the usual termination proof methods. We collect some of the results presented in this chapter in Table 6.2.

7 Very Long Size-Controlled Derivations

Truely understood,
nothing can be bad.

The size complexity maps n to the maximal length of a derivation starting with a term whose size is bounded by n. Apart from this condition on the size of the first term, there are no restrictions on the terms occuring in the derivation. In this chapter we consider the lengths of certain special derivations. The sizes of the terms occurring in these derivations are controlled by a given function. Similar sequences have been studied in connection with the embedding of trees. We state those results of this field which influenced our investigations.

To keep our notions short we denote by $\lhd$ the homeomorphic embedding relation $\prec_{\mathsf{hemb}}$ (as introduced in Definition 2.54) which is based on a label set containing exactly one element. As the label is always the same, it is safe to omit it. For every infinite sequence $(T_i)_{i<\omega}$ of trees Kruskal's Tree Theorem implies there are $i < j$ with $T_i \trianglelefteq T_j$. Harvey Friedman (see Simpson (1985)) investigated how long a finite sequence of trees satisfying $T_i \ntrianglelefteq T_j$ for $i < j$ can get under the additional restriction of a function bounding the size of the trees. It is not hard to see that without further restrictions the sequences may get arbitrarily long.

Lemma 7.1. *For each $f\colon \mathbb{N} \times \mathbb{N}^+ \to \mathbb{N}$ and each $n \in \mathbb{N}$ there is $N \in \mathbb{N}$ such that, for all sequences $(T_1, \dots, T_N)$ of trees satisfying*

$$(\forall m \in [1, N])(|T_m| \leqslant f(n, m)) \ ,$$

there exist $i, j \in [1, N]$ with $i < j$ and $T_i \trianglelefteq T_j$.

Proof. Let S_n collect the (possibly empty) sequences $(T_1, \dots, T_k)$ of trees with

$$(\forall m \in [1, k])(|T_m| \leqslant f(n, m)) \quad \text{and} \quad (\forall i, j \in [1, k])(i < j \Rightarrow T_i \ntrianglelefteq T_j) \ .$$

We partially order S_n by the extension relation $<_{\mathsf{ext}}$ (of Definition 2.33). As there are only finitely many trees of bounded size, $<_{\mathsf{ext}}$ is finitely branching. There are no infinite descending $<_{\mathsf{ext}}$-chains in S_n because otherwise the "union" of these sequences would violate Kruskal's Tree Theorem. An application of König's Lemma shows S_n is finite, hence we can choose N to be any number larger than the maximal length of the sequences occuring in S_n. □

Definition 7.2. With each $f \colon \mathbb{N} \times \mathbb{N}^+ \to \mathbb{N}$ we associate $K_f^{\trianglelefteq} \colon \mathbb{N} \to \mathbb{N}$ by making $K_f^{\trianglelefteq}(n)$ be the first N such that, for all sequences $(T_1, \ldots, T_N)$ of trees satisfying $(\forall m \in [1,N])(|T_m| \leqslant f(n,m))$, there exist $i, j \in [1,N]$ with $i < j$ and $T_i \trianglelefteq T_j$.

The items of the following Theorem are listed in chronological order.

Theorem 7.3. *Put $g(n,m) := n+m$ and, for $r \in \mathbb{R}$, $f_r(n,m) := n + r\log_2(m)$. By c we denote $\log_2(a)^{-1}$ where $a = 2.9557652856\ldots$ is the* tree constant *of Otter (1948).*

i. *$K_g^{\trianglelefteq}$ is not $<\varepsilon_0$-recursive.*
ii. *If $r \leqslant \frac{1}{2}$, then $K_{f_r}^{\trianglelefteq}$ is $<\varepsilon_0$-recursive, but if $r \geqslant 4$, then $K_{f_r}^{\trianglelefteq}$ is not.*
iii. *If $r < c$, then $K_{f_r}^{\trianglelefteq}$ is $<\varepsilon_0$-recursive, but if $r > c$, then $K_{f_r}^{\trianglelefteq}$ is not.*

Proof. Friedman (see Simpson (1985)) established (i), (ii) is due to Loebl and Matoušek (1987), and (iii) is from Weiermann (2000). □

We try to achieve similar results for simply terminating TRSs $\mathcal{R}$, replacing sequences of trees by $\mathcal{R}$-derivations in $\mathcal{T}(\Sigma)$ which obey certain size bounds. Such reductions are called *size-controlled.* The analogy to tree embedding is obvious once one recalls that within simple termination $s \to_{\mathcal{R}} t$ implies $s \not\trianglelefteq_{\mathsf{emb}} t$. In contrast to tree embedding, the restriction to derivations has a certain constructive aspect. We will encounter surprisingly long derivations almost immediately after leaving the field of those size bounds which are trivially too tight. The problem is vaguely related to term enumerating problems (for an overview see Odlyzko (1995)), as it amounts to enumerating, without repetitions, sufficiently many (though not all) terms of size bounded by n before stepping to terms of size beyond n.

Definition 7.4. With each TRS $\mathcal{R}$ and to each $f \colon \mathbb{N} \times \mathbb{N}^+ \to \mathbb{N}$ we associate $K_f^{\mathcal{R}} \colon \mathbb{N} \to \mathbb{N} \cup \{\infty\}$ by making $K_f^{\mathcal{R}}(n)$ be the first N such that there is no $\mathcal{R}$-derivation $(s_i)_{1 \leqslant i \leqslant N}$ in $\mathcal{T}(\Sigma)$ satisfying $(\forall m \in [1,N])(|s_m| \leqslant f(n,m))$, if such an N exists, and ∞ otherwise.

Lemma 7.5. *If $\mathcal{R}$ is terminating, then $K_f^{\mathcal{R}}$ is number-theoretic, and we have*

$$(\forall n)(K_f^{\mathcal{R}}(n) \leqslant \mathrm{Dc}_{\mathcal{R}}(f(n,1)) + 1) \; .$$

Proof. Termination of $\mathcal{R}$ implies $\mathrm{Dc}_{\mathcal{R}}$ is well-defined. As $\mathrm{Dc}_{\mathcal{R}}(f(n,1))$ is the maximal length of an $\mathcal{R}$-derivation starting with a term of size bounded by $f(n,1)$, we can easily impose a bound on $K^{\mathcal{R}}_{f}(n)$. □

If f grows rather slow, a low upper bound on $K^{\mathcal{R}}_{f}$ can be obtained by counting terms having bounded size. In general, investigating the asymptotic behavior of the function mapping n to the number of terms with size bounded by n is a surprisingly hard problem (of analytical combinatorics). These functions heavily depend on the underlying signature. It is good fortune that we do not require optimal bounds – a rough upper bound suffices.

Lemma 7.6. *For all n we have* $\mathrm{card}(\{s \in \mathcal{T}(\Sigma) : |s| \leqslant n\}) < (\mathrm{card}(\Sigma)+1)^{n}$.

Proof. We can identify the closed terms over Σ with their representations as strings over the alphabet Σ in, say, Polish notation. If we introduce a new symbol $\hat{\varepsilon}$ denoting an empty position, then the strings in Polish notation belonging to the closed terms of size bounded by n form a proper subset of the set of strings of length n over the alphabet $\Sigma \cup \{\hat{\varepsilon}\}$. □

For $n > 0$ the above result can easily be strengthened to

$$\mathrm{card}(\{s \in \mathcal{T}(\Sigma) : |s| \leqslant n\}) < (\mathrm{card}(\Sigma)+1)^{n-1} \cdot \mathrm{card}(\Sigma^{(0)}) ,$$

as the last symbol of any considered string is a constant. Similarly, for $n > 1$ the first symbol of any string is either $\hat{\varepsilon}$ or a member of $\Sigma^{(\geqslant 1)}$. Further slight improvements are possible using Lemma 3.1, but these results are still far from the real functions.

Lemma 7.7. *For $r \in \mathbb{R}$ we put $f_r(n,m) := n + r\log_2(m)$. Let Σ be a signature with $S := \mathrm{card}(\Sigma)+1 \geqslant 2$. If $r < \log_S(2)$ and $\mathcal{R}$ is a noncycling TRS over Σ, then $K^{\mathcal{R}}_{f_r}$ is bounded by an exponential function.*

Proof. Pick $a \in \mathbb{R}$ with $0 < a < 2$ such that $r \leqslant \log_S(a)$. Put $b := \log_2(a) < 1$ and $c := \frac{\log_2(S)}{1-b} > 0$. We have

$$rc \leqslant \frac{\log_S(a) \cdot \log_2(S)}{1-b} = \frac{b}{1-b} . \tag{7.1}$$

Assume for a contradiction that there is an n such that $2^{cn} < K^{\mathcal{R}}_{f_r}(n)$. Hence we find $s_1, \ldots, s_{2^{cn}} \in \mathcal{T}(\Sigma)$ with $s_1 \to_{\mathcal{R}} \ldots \to_{\mathcal{R}} s_{2^{cn}}$ and

$$(\forall m \in [1, 2^{cn}])(|s_m| \leqslant n + r\log_2(m) \leqslant n + rcn) .$$

Since $\mathcal{R}$ is noncycling, the s_m are mutually distinct terms of size bounded by $n + rcn$. Under these circumstances Lemma 7.6 and (7.1) yield

$$2^{cn} < S^{n+rcn} \leqslant S^{\frac{1}{1-b}n} = 2^{cn} ,$$

a contradiction. □

Theorem 7.35, the main result of this chapter, contains a complementary result. We combine the TRSs of Section 5.3, which simulate the Hydra battle, with certain SRSs containing sequences of p-adic representations of iterated logarithms in order to slow down term growth under preservation of derivation lengths. A careful calculation of the sizes of the terms occuring in the simulation leads to this result:

> *For any $p > 1$ and $k > 0$ there is a totally terminating TRS $\mathcal{R}$ over a signature Σ with $\operatorname{card}(\Sigma) = p+8+\frac{1}{2}k(k+5)$ such that $K^{\mathcal{R}}_{f_r}$ eventually dominates all $<\Delta_k$-recursive functions as soon as $r > \log_p(2)$.*

Later we will sketch how to boil down the signature to $p+10+k$ symbols, and we show how to get by with even less symbols if $k = 1$. There we arrive at a signature with $p + 9$ symbols and a very close bound – at most exponential growth for $r < \log_{p+10}(2)$ in contrast to growth beyond $<\Delta_1$-recursion for $r > \log_p(2)$. Those who are content with leaving $<\varepsilon_0$-recursion get by with $p + 8$ symbols. Though these bounds are rather tight, they are not optimal. This leaves us with an open problem – the quest for optimal bounds.

It may be instructive to learn a bit more about the line of proof we will follow. Recall from Section 5.3 that a Hydra battle configuration (α, n) is encoded by the term $\bullet[]^{n+1}d$ with $d \in \mathcal{D}$, the set of standard terms, and $\operatorname{val}(d) = \alpha$. Now, for any such $d \neq \mathsf{0}$, our TRS $\mathcal{R}$ is able to reduce the configuration to the next configuration: $\bullet[]^{n+1}d \xrightarrow{+} \bullet[]^{n+2}d[n]$. The main part of this derivation consists in reducing $\bullet^{n+1}d$ to $\circ d[n]$. We will see in Section 7.2 that, roughly speaking, the sizes of all terms involved in a certain derivation from $\bullet^{n+1}d$ to $\circ d[n]$ can be bounded by $|d|(\operatorname{dp}(d) + n)$. The Hydra battle requires an iteration of such derivations, and iterating without protection obviously exceeds the size bounds imposed by f_r. We delay these derivations by doing other harmless things as long as this is possible.

In the proof of Theorem 7.3.ii, Loebl and Matoušek considered the length of iterated p-adic representations of $n \in \mathbb{N}$. These are defined via

$$|n|_p^{(0)} := n \quad \text{and} \quad |n|_p^{(i+1)} := ||n|_p^{(i)}|_p \ ,$$

where $|m|_p$ is the length of the p-adic representation of m. In Section 7.1 we consider a monadic signature containing symbols for the digits below p and, apart from other auxiliary symbols, the symbol $\vartriangle$, and the constant $\blacktriangle$. By $\operatorname{code}_p(n)$ we denote the string which is the p-adic representation of n, and further we put

$$\operatorname{logs}_{p,i}(n) := \operatorname{code}_p(|n|_p^{(i)}) \vartriangle \dots \vartriangle \operatorname{code}_p(|n|_p^{(0)}) \blacktriangle \ .$$

Based on these terms we define a totally terminating SRS $\mathcal{R}_p$ which (mostly executes p-adic addition by 1 and) is able to reduce $\operatorname{logs}_{p,i}(n)$ to $\operatorname{logs}_{p,i}(n+1)$, provided that $|n|_p^{(i+1)} = |n+1|_p^{(i+1)}$. This way we get quite long derivations with rather few increases in size.

The main idea is carried out in Section 7.3. It consists in combining $\mathcal{R}_p$ with the system $\mathcal{R}$ having a complexity beyond $<\Delta_k$-recursion, producing a new totally terminating TRS. Now we may reduce in $\mathcal{R}_p$ until it is safely possible to step from one Hydra battle configuration to the next without violating the size bounds imposed by f_r. To be more explicit, in Proposition 7.32 we put

$$d_m := \circ[]^{n+m-1} d[n, n+m-2]$$

for $n \in \mathbb{N}$; $m > 0$, and $d \in \mathcal{D} \setminus \{0\}$. For $i \in \mathbb{N}$ we further introduce

$$u_m := \star(d_{|m|_p^{(i+1)}}, \text{logs}_{p,i}(m)) ,$$

where $\star$ is a binary symbol. There is an L larger than the length of the battle for the configuration $(\text{val}(d), n)$ such that a certain derivation

$$u_1 \xrightarrow{+} u_2 \xrightarrow{+} \ldots \xrightarrow{+} u_m \xrightarrow{+} u_{m+1} \xrightarrow{+} \ldots \xrightarrow{+} u_L$$

suffices rather strong size conditions related to f_r. In fact, we simply reduce $\text{logs}_{p,i}(m)$ to $\text{logs}_{p,i}(m+1)$ using $\mathcal{R}_p$ as long as this is possible, requiring very little space, and otherwise we step from one configuration to the next using $\mathcal{R}$, arranging things so that $\text{logs}_{p,i}(m+1)$ appears, too. For i large enough ($i > 2$ suffices), the latter case is so rare that its effects on size growth are harmless.

7.1 Sequences of Iterated Logarithms

In this section we keep $p \geqslant 2$ and $a := p-1$ fixed. We are going to define numbers, strings, and SRSs dealing with (codes of) iterated p-adic logarithms.

Definition 7.8. For $n \in \mathbb{N}$ we define $|n|_p$, the *p-adic length* of n, by

$$|n|_p := \lceil \log_p(n+1) \rceil \ .$$

We iterate this for $i \in \mathbb{N}$ and put $|n|_p^{(0)} := n$ and $|n|_p^{(i+1)} := ||n|_p^{(i)}|_p$.

Lemma 7.9. *Let $n, i \in \mathbb{N}$ be given.*

i. *If $n > 0$, then $|n|_p \leqslant \log_p(n) + 1$ holds.*
ii. *We have $|n|_p \leqslant |n+1|_p \leqslant |n|_p + 1$, and $|n|_p < |n+1|_p$ holds if and only if $n = p^m - 1$ for some $m \geqslant 0$.*
iii. *We have $|n|_p^{(i+1)} \leqslant |n|_p^{(i)}$. If $f(x)$ is a polynomial, then $f(|n|_p^{(i+1)}) < |n|_p^{(i)}$ holds for almost all n.*

Proof. For (i) fix the $m \in \mathbb{N}$ satisfying

$$m \leqslant \log_p(n) < m + 1 \; . \tag{7.2}$$

If $\log_p(n+1) < m+1$, then $|n|_p = m + 1 \leqslant \log_p(n) + 1$ holds, and otherwise we have $m + 1 \leqslant \log_p(n+1)$ and get

$$n = p^{\log_p(n)} < p^{m+1} \leqslant p^{\log_p(n+1)} = n + 1 \; .$$

Since $p^{m+1} \in \mathbb{N}$ this yields $p^{m+1} = n + 1$, hence, recalling (7.2), we infer

$$|n|_p = \lceil m + 1 \rceil = m + 1 \leqslant \log_p(n) + 1 \; .$$

The proof of (ii) is similar and therefore left out, while the first part of (iii) is immediate and the second part relies on the well known fact that $n \mapsto a^n$ (with $a > 1$) grows faster than any polynomial. □

Definition 7.10. We introduce the signature

$$\Sigma_p := \{\underline{0}, \underline{1}, \ldots, \underline{a}, \overline{0}, \triangle, \blacktriangle\}$$

where all symbols except for the constant $\blacktriangle$ are unary.

We should briefly describe the intended meaning of these symbols. The digits of the p-adic representation are $\underline{0}, \underline{1}, \ldots, \underline{a}$, while $\overline{0}$ represents a 0 carrying an overflow of 1 which resulted from adding 1 to a. We use $\triangle$ to separate distinct representations, and $\blacktriangle$ is the necessary constant.

Definition 7.11. For $n \in \mathbb{N}$ we define $\mathrm{code}_p(n)$ to be the (possibly empty) string over $\{\underline{0}, \underline{1}, \ldots, \underline{a}\}$ which corresponds to the p-adic representation of n. To be precise, we put $\mathrm{code}_p(0) := \varepsilon$; and if $n = n' \cdot p + m > 0$ with $m < p$, then $\mathrm{code}_p(n) := \mathrm{code}_p(n')\underline{m}$.

Lemma 7.12. *For all $n \in \mathbb{N}$ we have $|\mathrm{code}_p(n)\blacktriangle| = |n|_p + 1$.*

We are now going to define terms $\mathrm{logs}_{p,i}(n)$ containing the $\triangle$-divided sequence from $\mathrm{code}_p(|n|_p^{(i)})$ to $\mathrm{code}_p(|n|_p^{(0)})$. Our intention is to provide an SRS $\mathcal{R}_p$ which is able to reduce $\mathrm{logs}_{p,i}(n)$ to $\mathrm{logs}_{p,i}(n+1)$ if $|n|_p^{(i+1)} = |n+1|_p^{(i+1)}$. This will be done by trying to add 1 to $\mathrm{code}_p(|n|_p^{(0)})$. If there is no overflow, we succeed, and otherwise we enlarge the representation by a leading 1 and add 1 to $\mathrm{code}_p(|n|_p^{(1)})$. We iterate this and usually succeed. The only problem occurs if there is an overflow in the leading position, i.e. if $|n|_p^{(i+1)} \neq |n+1|_p^{(i+1)}$, leaving us with $\overline{0}\,\underline{0}^k$ for $k = |n|_p^{(i+1)} - 1$ at the top of the term. In both cases, the term resulting from adding 1 is called $\mathrm{logs}^{+}_{p,i}(n)$.

Definition 7.13. For $i \in \mathbb{N}$ and $n > 0$, we put $\mathrm{code}_{p,i}(n) := \mathrm{code}_p(|n|_p^{(i)})$ and

$$\mathrm{code}^+_{p,i}(n) := \begin{cases} \mathrm{code}_{p,i}(n+1) & \text{if } |n|_p^{(i+1)} = |n+1|_p^{(i+1)} \ , \\ \overline{0}\underline{0}^{|n|_p^{(i+1)}-1} & \text{otherwise.} \end{cases}$$

We will frequently abbreviate $\mathrm{code}^+_{p,0}(n)$ by $\mathrm{code}^+_p(n)$. Building on these we recursively define $\mathrm{logs}_{p,i}(n) \in \mathcal{T}(\Sigma_p)$ by

$$\mathrm{logs}_{p,i}(n) := \begin{cases} \mathrm{code}_{p,0}(n)\blacktriangle & \text{if } i = 0 \ , \\ \mathrm{code}_{p,i}(n)\vartriangle \mathrm{logs}_{p,i'}(n) & \text{if } i = i' + 1 \ , \end{cases}$$

and finally we introduce $\mathrm{logs}^+_{p,i}(n) \in \mathcal{T}(\Sigma_p)$ by

$$\mathrm{logs}^+_{p,i}(n) := \begin{cases} \mathrm{code}^+_{p,0}(n)\blacktriangle & \text{if } i = 0 \ , \\ \mathrm{code}^+_{p,i}(n)\vartriangle \mathrm{logs}_{p,i'}(n+1) & \text{if } i = i' + 1 \ . \end{cases}$$

Note that in the last line we intentionally use logs and not logs^+.

Lemma 7.14. *Let $n > 0$.*

i. *If $|n|_p^{(i+1)} = |n+1|_p^{(i+1)}$ then $\mathrm{logs}^+_{p,i}(n) = \mathrm{logs}_{p,i}(n+1)$.*

ii. *We have $|\mathrm{logs}_{p,i}(n)| = i + 1 + \sum_{l=1}^{i+1} |n|_p^{(l)}$.*

Proof. For (i) we just have to check that the condition implies $\mathrm{code}^+_{p,i}(n) = \mathrm{code}_{p,i}(n+1)$. We establish (ii) by induction on i, using Lemma 7.12. □

We are now going to define the TRS $\mathcal{R}_p$ which is able to reduce $\mathrm{logs}_{p,i}(n)$ to $\mathrm{logs}^+_{p,i}(n)$. Everything it has to do is to simulate addition by 1 and to continue at a higher iteration stage as long as there are overflows.

Definition 7.15. The TRS $\mathcal{R}_p$ over the signature Σ_p contains the rules

$$(\mathrm{S}a)\ \ \underline{a}\blacktriangle \to \overline{0}\blacktriangle \ , \qquad (\mathrm{C}a)\ \ \underline{a}\overline{0}x \to \overline{0}\underline{0}x \ , \qquad (\mathrm{O}a)\ \ \underline{a}\vartriangle\overline{0}x \to \overline{0}\vartriangle\underline{1}\,\underline{0}x$$

and, for each $m < a$ and $m' := m + 1$, the rules

$$(\mathrm{S}m)\ \ \underline{m}\blacktriangle \to \underline{m'}\blacktriangle \qquad (\mathrm{C}m)\ \ \underline{m}\overline{0}x \to \underline{m'}\,\underline{0}x \qquad (\mathrm{O}m)\ \ \underline{m}\vartriangle\overline{0}x \to \underline{m'}\vartriangle\underline{1}\,\underline{0}x \ .$$

Lemma 7.16. *$\mathcal{R}_p$ is totally terminating.*

Proof. Fix a well-order $(\mathcal{P}, \prec)$ with a mapping $S\colon \mathcal{P} \to \mathcal{P}$ which is weakly monotone and has the subterm property. For example, any limit ordinal with $\alpha \mapsto \alpha + 1$ playing the rôle of S will do. In Proposition 7.31 the well-order will be the one used in Theorem 5.50.

Let $\prec_{\mathsf{kbo}}$ be the KBO based on $\Sigma_p^- := \Sigma_p \setminus \{\triangle\}$ with the (total) precedence

$$\underline{0} \succ \underline{1} \succ \cdots \succ \underline{a} \succ \overline{0} \succ \blacktriangle$$

and the weight function which assigns 1 to each symbol. Note that $\prec_{\mathsf{kbo}}$ is just the lexicographic comparison of strings based on $\prec$. Let $(\mathcal{Q}, \lhd)$ be the lexicographic product of $(\mathcal{P}, \prec)$ and $(\mathcal{T}(\Sigma_p^-), \prec_{\mathsf{kbo}})$. Since (partially by Theorem 3.58) these two orders are well-orders, so is $(\mathcal{Q}, \lhd)$. Keeping Theorem 3.45 in mind, it suffices to find a weakly monotone Σ_p-algebra $(\mathcal{Q}, \lhd, \mathcal{F}_p)$ which has the subterm property and is compatible with $\mathcal{R}_p$.

Let q_0 be $\prec$-minimal in $\mathcal{P}$. We define the operations of the Σ_p-algebra by

$$\begin{aligned} [\blacktriangle] &:= (q_0, \blacktriangle) \ , \\ [f]((q,s)) &:= (q, fs) \qquad \text{for } f \in \Sigma_p^- \setminus \{\blacktriangle\} \ , \\ [\triangle]((q,s)) &:= (S(q), \blacktriangle) \ . \end{aligned}$$

Let us first establish the subterm property. For $[f]$ as above this follows from $fs \succ_{\mathsf{kbo}} s$, while for $[\triangle]$ the subterm property of S is vital.

Consider $(q,s) \unrhd (q',s')$ for weak monotonicity. With $[f]$ as above we get the desired $(q, fs) \unrhd (q', fs')$ because any KBO is closed under contexts, while the proof for $[\triangle]$ relies on the weak monotonicity of S.

It remains to treat compatibility. We grab $\rho\colon \mathcal{V} \to \mathcal{Q}$ with $\rho(x) = (q,s)$. The rule (Sa) is now handled by

$$[\![\underline{a}\blacktriangle, \rho]\!] = (q_0, \underline{a}\blacktriangle) \rhd (q_0, \overline{0}\blacktriangle) = [\![\overline{0}\blacktriangle, \rho]\!] \ ,$$

while the correct orientation of (Ca) is confirmed by

$$[\![\underline{a}\overline{0}x, \rho]\!] = (q, \underline{a}\overline{0}s) \rhd (q, \overline{0}\underline{0}s) = [\![\overline{0}\underline{0}x, \rho]\!] \ .$$

The most interesting rule is (Oa), which produces

$$[\![\underline{a}\triangle\overline{0}x, \rho]\!] = (S(q), \underline{a}\blacktriangle) \rhd (S(q), \overline{0}\blacktriangle) = [\![\overline{0}\triangle\underline{1}\,\underline{0}x, \rho]\!] \ .$$

Just as $\underline{a} \succ \overline{0}$ lurked behind the preceding rules, the remaining rules (Cm), (Om), and (Sm) with $m < a$ are based on $\underline{m} \succ \underline{m+1}$. □

Lemma 7.17. *For $i \in \mathbb{N}$ and $n > 0$ we have*

$$\mathrm{logs}_{p,i}(n) \xrightarrow{+}{}^{\leqslant}_{\mathcal{R}_p} \mathrm{logs}^+_{p,i}(n)$$

Proof. It suffices to show $\mathrm{logs}_{p,i}(n) \xrightarrow{+} \mathrm{logs}^+_{p,i}(n)$, since the rules of $\mathcal{R}_p$ are not decreasing. First of all, by induction on $n > 0$, we show

$$\mathrm{code}_p(n)\overline{0}x \xrightarrow{+} \mathrm{code}^+_p(n)\underline{0}x \ . \tag{7.3}$$

There exist $m < p$ and $n' \in \mathbb{N}$ such that $n = n' \cdot p + m$. Hence we see

$$\mathrm{code}_p(n)\overline{0}x = \mathrm{code}_p(n')\underline{m}\overline{0}x \rightarrow_{\mathrm{C}m} \begin{cases} \mathrm{code}_p(n')\underline{m+1}\,\underline{0}x & \text{if } m < a\ , \\ \mathrm{code}_p(n')\overline{0}\underline{0}x & \text{if } m = a\ . \end{cases}$$

If $m < a$, then we get $\mathrm{code}_p(n')\underline{m+1} = \mathrm{code}_p(n+1) = \mathrm{code}^+_p(n)$, so the case $m = a$ is a bit more interesting. It remains to consider $n' > 0$. The induction hypothesis yields

$$\mathrm{code}_p(n')\overline{0}\underline{0}x \xrightarrow{+} \mathrm{code}^+_p(n')\underline{0}\,\underline{0}x = \begin{cases} \mathrm{code}_p(n'+1)\underline{0}\,\underline{0}x & \text{if } |n'|_p = |n'+1|_p\ , \\ \overline{0}\underline{0}^{|n'|_p-1}\underline{0}\,\underline{0}x & \text{if } |n'|_p < |n'+1|_p\ . \end{cases}$$

If $|n'|_p = |n'+1|_p$, then we have $|n|_p = |n+1|_p$ and

$$\mathrm{code}_p(n'+1)\underline{0}y = \mathrm{code}_p((n'+1) \cdot p)y = \mathrm{code}_p(n+1)y = \mathrm{code}^+_p(n)y\ ,$$

while else we get $|n|_p = |n'|_p + 1$, so $\overline{0}\underline{0}^{|n'|_p-1}\underline{0}y = \overline{0}\underline{0}^{|n|_p-1}y = \mathrm{code}^+_p(n)y$ holds.

In a second step we intend to establish

$$\mathrm{code}_p(n)\blacktriangle \xrightarrow{+} \mathrm{code}^+_p(n)\blacktriangle \tag{7.4}$$

and

$$\mathrm{code}_p(n)\triangle\overline{0}x \xrightarrow{+} \mathrm{code}^+_p(n)\triangle\underline{1}\,\underline{0}x\ . \tag{7.5}$$

For $n = n' \cdot p + m$ as above we see

$$\mathrm{code}_p(n)\blacktriangle = \mathrm{code}_p(n')\underline{m}\blacktriangle \rightarrow_{\mathrm{S}m} \begin{cases} \mathrm{code}_p(n')\underline{m+1}\blacktriangle & \text{if } m < a\ , \\ \mathrm{code}_p(n')\overline{0}\blacktriangle & \text{if } m = a\ . \end{cases}$$

If $m < a$, then we may proceed just as above, and otherwise we can show (7.4) either by Definition 7.11 (if $n' = 0$) or by (7.3). We prove (7.5) in a similar manner, using (Om) instead of (Cm).

Now we are prepared to show $\mathrm{logs}_{p,i}(n) \xrightarrow{+} \mathrm{logs}^+_{p,i}(n)$ by induction on i. The case $i = 0$ is just (7.4), and if $i = i' + 1$, then the induction hypothesis yields

$$\mathrm{logs}_{p,i}(n) = \mathrm{code}_{p,i}(n)\triangle\,\mathrm{logs}_{p,i'}(n) \xrightarrow{+} \mathrm{code}_{p,i}(n)\triangle\,\mathrm{logs}^+_{p,i'}(n) =: t\ .$$

If $|n|_p^{(i'+1)} = |n+1|_p^{(i'+1)}$, then $\mathrm{code}_{p,i}(n) = \mathrm{code}_{p,i}(n+1) = \mathrm{code}^+_{p,i}(n)$ and, by Lemma 7.14.i, $\mathrm{logs}^+_{p,i}(n) = \mathrm{logs}_{p,i}(n+1)$. Hence $t = \mathrm{logs}^+_{p,i}(n)$. Things get a bit more involved if $|n+1|_p^{(i'+1)} = |n|_p^{(i'+1)} + 1$. There exists a closed term s with

$$\begin{aligned} t =\ & \mathrm{code}_{p,i}(n)\triangle\overline{0}\underline{0}^{|n|_p^{(i'+1)}-1}s \\ \xrightarrow{+}\ & \mathrm{code}^+_{p,i}(n)\triangle\underline{1}\,\underline{0}\,\underline{0}^{|n|_p^{(i'+1)}-1}s && \text{by (7.5)} \\ =\ & \mathrm{code}^+_{p,i}(n)\triangle\,\mathrm{code}_{p,i'}(n+1)s =: t'\ . \end{aligned}$$

Depending on i we either have $s = \blacktriangle$ or $s = \triangle\,\mathrm{logs}_{p,i'-1}(n+1)$, and in both cases $t' = \mathrm{logs}^+_{p,i}(n)$ is valid, just as expected. □

7.2 Hydra Battles, Revisited

We have to take a closer look at TRSs simulating Hydra battles. The focus is on results concerning the sizes of the terms involved in a complete simulation of such a battle. In order to get pleasing results, a few rules which prevent the creation of too large intermediate terms are added.

Just as we did in Section 5.3 (on page 96), we fix some $k > 0$ and treat the $k+1$-ary ψ. We even recycle the signature Σ of Definition 5.40 and the TRS $\mathcal{R}$ (cf. Table 5.1 on page 110) which simulates all Hydra battles below Δ_k.

Definition 7.18. The TRS $\mathcal{R}'$ over the signature Σ consists of the rules from $\mathcal{R}$ and these rules:

$$\begin{array}{lrcll}
(\mathrm{N1}') & \bullet\Box x & \to & \Box\bullet x & \\
(\mathrm{N2}') & \Box\circ x & \to & \circ\Box x & \\
(\mathrm{R}\mathsf{M}') & \mathsf{M}(\bar{x}, \Box y) & \to & \mathsf{P}(\bar{x}, y) & \\
(\mathrm{R}\mathsf{J}_i') & \mathsf{J}_i(x_1, \ldots, \Box x_i, y) & \to & \mathsf{P}(\bar{x}, y, \bar{\mathsf{0}}) & \quad \text{for } i \in [1, k] \,.
\end{array}$$

Proposition 7.19. *$\mathcal{R}'$ is totally terminating.*

Proof. We intend to show that the Σ-algebra $(\mathcal{P}, \prec, \mathcal{F}_{\mathcal{P}})$ we used in Theorem 5.50 (it was called $(\mathcal{P}, \prec, \mathcal{F})$ there) to show that $\mathcal{R}$ is totally terminating remains valid for $\mathcal{R}'$. As we have already shown there and in Lemma 5.49, it is weakly monotone, has the subterm property, and is compatible with $\mathcal{R}$. Hence compatibility with the new rules remains to be established. This is easily achieved, since the new rules are just short cuts for special combinations of two old rules. For example, (N1′) can be simulated by using (N1) and ($\mathrm{S}_1\bullet$) via $\bullet\Box x \to_{\mathrm{N1}} \Box\bullet\bullet x \to_{\mathrm{S}_1\bullet} \Box\bullet x$. In the same manner we can treat (N2′) (as a combination of (N2) and ($\mathrm{S}_1\Box$)), (RM′) (as a combination of (RM) and (S_2+)), and (RJ_i') (as a combination of (RJ_i) and ($\mathrm{S}_{i+1}\mathsf{J}_i$)). □

We are going to reprove certain results of Section 5.3.4 with the focus on the sizes of the terms involved and begin with Lemmata 5.41 and 5.42. Recall from page 46 that, for example, $s \xrightarrow{*}^{\leqslant} t$ indicates that there is a derivation from s to t such that the sizes of all occuring terms are bounded by $|t|$, and recall further that $\xrightarrow{+}^{\leqslant}$ is not the transitive closure of $\to^{\leqslant}$.

Lemma 7.20. *If s' is a subterm of $s \in \mathcal{T}(\Sigma)$, then $s \xrightarrow{*}^{\geqslant} s'$.*

Proof. The embedding rules ($\mathrm{S}_i f$) are not increasing. □

Lemma 7.21. *For $d \in \mathcal{D}$ we have $d\langle n\rangle \xrightarrow{*}^{\geqslant} d[n]$.*

Proof. All rules used in the proof of Lemma 5.42 are not increasing. □

The next result is an adaptation of Lemma 5.43 to our new setting.

Lemma 7.22. *For $n > 0$ and $s \in \mathcal{T}(\Sigma)$ we have*

i. $[]^n s \xrightarrow{+}{}^{=} \bullet^n s \xrightarrow{+}{}^{\geqslant} s$

ii. $\bullet[]^n s \xrightarrow{+}{}^{\leqslant} []^n \bullet^n s$

iii. $[]^n \circ s \xrightarrow{+}{}^{=} \circ[]^n s$.

Proof. As in Lemma 5.43.i we establish (i) using (N4) followed by ($\mathrm{S}_1\bullet$). For (ii) we recall that, according to the proof of Lemma 5.43.ii, we can apply (N1) exactly $2^n - 1$ times to $\bullet[]^n s$, reaching $[]^n \bullet^{2^n} s$. We are a bit more modest here and apply (N1) only $n-1$ times, generating n occurrences of $\bullet$ somewhere in front of s. These are moved to adequate positions by using (N1′) as often as possible. An iterated application of (N2′) establishes (iii). □

Lemma 7.23. *For $s, t, \bar{s} \in \mathcal{T}(\Sigma)$; $n \geqslant 0$, and $i \in [1, k+1]$ we have*

i. $\bullet^{n+1}{+}(s,t) \xrightarrow{+}{}^{=} {+}(s, \bullet^{n+1} t)$

ii. $\bullet^{n+1}\mathsf{P}(\bar{s}) \xrightarrow{*}{}^{=} \bullet\mathsf{P}(s_1, \dots, \bullet^n s_i, \dots, s_{k+1}) \xrightarrow{+}{}^{=} \mathsf{P}(s_1, \dots, \bullet^{n+1} s_i, \dots, s_{k+1})$

iii. $\bullet^{n+1}\mathsf{P}(\bar{s}) \xrightarrow{+}{}^{=} \mathsf{P}(s_1, \dots, []^{n+1} s_i, \dots, s_{k+1})$

iv. $\bullet^{n+1}\mathsf{P}(s_1, \dots, \mathsf{S}s_i, \dots, s_{k+1}) \xrightarrow{+}{}^{=} \mathsf{P}(s_1, \dots, \mathsf{S}[]^{n+1} s_i, \dots, s_{k+1})$.

Proof. A look at Lemma 5.44 suffices to see that everything can be based on the harmless rules ($\mathrm{D}_i f$) and the first half of Lemma 7.22.i. □

Lemma 7.24. *For $\bar{s}, t \in \mathcal{T}(\Sigma)$; $n > 0$, and $i \in [1, k]$ we have*

i. $\mathsf{M}(\bar{s}, []^n t) \xrightarrow{+} \mathsf{P}(\bar{s}, t) \times n$ *bounded by* $(|\mathsf{M}(\bar{s}, t)| + 1) \cdot n + 1$

ii. $\mathsf{J}_i(s_1, \dots, []^n s_i, t) \xrightarrow{+} \mathsf{P}(s_1, \dots, s_i, \cdot, \bar{\mathsf{0}})^n(t)$ *bounded by*

$$(2 + |s_1| + \cdots + |s_i| + |\bar{\mathsf{0}}|) \cdot n + |t| \ .$$

Proof. Let us first treat (i) by induction on $n > 0$. For $n = 1$ we see

$$\mathsf{M}(\bar{s}, []t) \to^{\geqslant}_{\mathsf{RM}'} \mathsf{P}(\bar{s}, t) = \mathsf{P}(\bar{s}, t) \times 1$$

and $|\mathsf{M}(\bar{s}, []t)| = |\mathsf{M}(\bar{s}, t)| + 1$. The step from n to $n+1$ is done as in Lemma 5.45:

$$\begin{aligned} \mathsf{M}(\bar{s}, []^{n+1} t) &\to^{\leqslant}_{\mathsf{RM}} {+}(\mathsf{M}(\bar{s}, []^n t), \mathsf{P}(\bar{s}, []^n t)) \xrightarrow{+}{}^{\geqslant} {+}(\mathsf{M}(\bar{s}, []^n t), \mathsf{P}(\bar{s}, t)) \\ &\xrightarrow{+} \quad {+}(\mathsf{P}(\bar{s}, t) \times n, \mathsf{P}(\bar{s}, t)) = \mathsf{P}(\bar{s}, t) \times (n+1) \ , \end{aligned}$$

using Lemma 7.22.i. The first peak is estimated by

$$\begin{aligned} |{+}(\mathsf{M}(\bar{s}, []^n t), \mathsf{P}(\bar{s}, []^n t))| &= 1 + |\mathsf{M}(\bar{s}, t)| \cdot 2 + 2n \\ &= (|\mathsf{M}(\bar{s}, t)| + 1) \cdot 2 + 2(n-1) + 1 \\ &\leqslant (|\mathsf{M}(\bar{s}, t)| + 1) \cdot 2 + (|\mathsf{M}(\bar{s}, t)| + 1)(n-1) + 1 \\ &= (|\mathsf{M}(\bar{s}, t)| + 1)(n+1) + 1 \ , \end{aligned}$$

relying on $n-1 \geqslant 0$ and $|\mathsf{M}(\bar{s},t)| \geqslant 3$, and the the induction hypothesis tells us the last rewrite step is bounded by

$$1 + (|\mathsf{M}(\bar{s},t)|+1)\cdot n + 1 + |\mathsf{P}(\bar{s},t)| = 1 + (|\mathsf{M}(\bar{s},t)|+1)(n+1)\ .$$

In a similar way we show (ii) by induction on $n > 0$. For $n = 1$ we have

$$\mathsf{J}_i(s_1,\dots,[\!]s_i,t) \rightarrow_{\mathrm{RJ}'_i} \mathsf{P}(s_1,\dots,s_i,t,\bar{\mathsf{0}}) = \mathsf{P}(s_1,\dots,s_i,\cdot,\bar{\mathsf{0}})^1(t)\ ,$$

and it is easily checked that the derivation has the announced bound. The step from n to $n+1$ is achieved by Lemma 7.22.i and the induction hypothesis via

$$\begin{aligned}\mathsf{J}_i(s_1,\dots,[\!]^{n+1}s_i,t) &\rightarrow^{\leqslant}_{\mathrm{RJ}_i} \mathsf{P}(s_1,\dots,[\!]^{n}s_i,\mathsf{J}_i(s_1,\dots,[\!]^{n}s_i,t),\bar{\mathsf{0}})\\ &\xrightarrow{+}{}^{\geqslant}\ \mathsf{P}(s_1,\dots,s_i,\mathsf{J}_i(s_1,\dots,[\!]^{n}s_i,t),\bar{\mathsf{0}})\\ &\xrightarrow{+}\ \mathsf{P}(s_1,\dots,s_i,\mathsf{P}(s_1,\dots,s_i,\cdot,\bar{\mathsf{0}})^n(t),\bar{\mathsf{0}})\\ &=\ \mathsf{P}(s_1,\dots,s_i,\cdot,\bar{\mathsf{0}})^{n+1}(t)\ ,\end{aligned}$$

Obeying the latter, the last step is bounded by

$$1 + \sum_{j=1}^{i}|s_j| + |\bar{\mathsf{0}}| + (2 + \sum_{j=1}^{i}|s_j| + |\bar{\mathsf{0}}|)\cdot n + |t|\ ,$$

which is within our bound, while the size of the second peak is

$$\begin{aligned}&(1+|s_1|+\cdots+|s_i|)\cdot 2 + 2n + |t| + |\bar{\mathsf{0}}|\\ \leqslant\ &(2+|s_1|+\cdots+|s_i|+|\bar{\mathsf{0}}|)\cdot 2 + 2(n-1) + |t|\\ \leqslant\ &(2+|s_1|+\cdots+|s_i|+|\bar{\mathsf{0}}|)(n-1+2) + |t|\ ,\end{aligned}$$

thus also controlled by our bound. □

Now we are prepared to prove the size controlled analog to Proposition 5.46.

Proposition 7.25. *For $d \in \mathcal{D}\setminus\{\mathsf{0}\}$ and $n \geqslant 0$ we have*

$$\bullet^{n+1}d \xrightarrow{+} \circ d[n] \qquad \textit{bounded by } |d|(\mathrm{dp}(d)+n)\ .$$

Proof by induction on $\mathcal{D}$. In accordance with the proof of Proposition 5.46 and Lemma 7.21, it suffices to show

$$\bullet^{n+1}d \xrightarrow{+} \circ d\langle n\rangle \qquad \text{bounded by } |d|(\mathrm{dp}(d)+n)\ ,$$

and again we are free to fall back on the induction hypothesis whenever $u[n]$ (with u being a subterm of d) is used to define $d\langle n\rangle$, since in these cases we have $u \neq \mathsf{0}$. Due to $d \neq \mathsf{0}$ we can freely rely on the important properties

$$\mathrm{dp}(d) \geqslant 2 \quad \text{and} \quad |d| \geqslant 2\ , \tag{7.6}$$

which immediately yield

$$|\bullet^{n+1}d| = |d| + n + 1 < |d| + |d|n + |d| \leqslant |d|(\mathrm{dp}(d) + n) \ . \tag{7.7}$$

We are going to treat the different shapes of $d\langle n\rangle$ (mostly) in order of their appearance in Definition 5.38. The frequent use of Lemma 7.22.i and Lemma 7.23 will not be indicated. In (5.8b) we have $d = \mathsf{S}s$ and get

$$\bullet^{n+1}\mathsf{S}s \xrightarrow{*}{}^{\geqslant} \bullet\mathsf{S}s \to_{\mathrm{H1}}^{\geqslant} \circ s \ ,$$

which is correctly bounded via (7.7). For (5.8c) we use the induction hypothesis to see

$$\bullet^{n+1}{+}(s,t) \xrightarrow{+}{}^{=} {+}(s, \bullet^{n+1}t) \xrightarrow{+} {+}(s, \circ t[n]) \to_{\mathrm{E}_2+}^{=} \circ{+}(s, t[n]) \ ,$$

and to confirm the bound

$$1 + |s| + |t|(\mathrm{dp}(t) + n) \leqslant (1 + |s| + |t|)(\mathrm{dp}(t) + n) < |d|(\mathrm{dp}(d) + n)$$

on the second part of the derivation. The first part is again controlled by (7.7), and since the use of this result is routine by now we will not always refer to it in the remainder of this proof. As we can treat (5.8d) and (5.8k) in a manner very similar to (5.8c), there is no urge to display them here. Let us turn to (5.8e) and $d = \mathsf{P}(\bar{\mathsf{0}}, \mathsf{S}t)$. With the help of Lemma 7.24.i we reach

$$\bullet^{n+1}\mathsf{P}(\bar{\mathsf{0}}, \mathsf{S}t) \xrightarrow{+}{}^{=} \mathsf{P}(\bar{\mathsf{0}}, \mathsf{S}[]^{n+1}t) \to_{\mathrm{H2}}^{=} \circ\mathsf{M}(\bar{\mathsf{0}}, []^{n+1}t) \xrightarrow{+} \circ(\mathsf{P}(\bar{\mathsf{0}}, t) \times (n+1))$$

with the bound

$$(|\mathsf{P}(\bar{\mathsf{0}}, t)| + 1)(n+1) + 2 = |d|(n+1) + 2 \leqslant |d|(\mathrm{dp}(d) + n) \ .$$

In the similar case (5.8f) we recall from the proof of Proposition 5.46 that $d = \mathsf{P}(\bar{\mathsf{0}}, \mathsf{P}(\bar{s}, t))$ holds. We can again rely on Lemma 7.24.i, which reveals

$$\begin{aligned} \bullet^{n+1}\mathsf{P}(\bar{\mathsf{0}}, \mathsf{P}(\bar{s}, t)) \xrightarrow{+}{}^{=} \ & \mathsf{P}(\bar{\mathsf{0}}, \mathsf{P}(\bar{s}, []^{n+1}t)) \\ & \to_{\mathrm{H3}}^{\geqslant} \circ\mathsf{M}(\bar{s}, []^{n+1}t) \xrightarrow{+} \circ(\mathsf{P}(\bar{s}, t) \times (n+1)) \ . \end{aligned}$$

For anything not covered by (7.7), the induction hypothesis yields the bound

$$(|\mathsf{P}(\bar{s}, t)| + 1)(n+1) + 2 < |d|(n+1) + 2 \leqslant |d|(\mathrm{dp}(d) + n) \ .$$

Before we take care of (5.8g) we have a look at the easier (5.8h). Here $d = \mathsf{P}(s_1, \ldots, \mathsf{S}s_i, \bar{\mathsf{0}}, t)$ holds, and Lemma 7.24.ii yields

$$\begin{aligned} \bullet^{n+1}\mathsf{P}(s_1, \ldots, \mathsf{S}s_i, \bar{\mathsf{0}}, t) \xrightarrow{+}{}^{=} \ & \mathsf{P}(s_1, \ldots, \mathsf{S}[]^{n+1}s_i, \bar{\mathsf{0}}, t) \\ & \to_{\mathrm{H}_i4}^{\geqslant} \circ\mathsf{J}_i(s_1, \ldots, []^{n+1}s_i, t) \xrightarrow{+} \circ\mathsf{P}(s_1, \ldots, s_i, \cdot, \bar{\mathsf{0}})^{n+1}(t) \ , \end{aligned}$$

while the bound for the interesting part is

$$\begin{aligned}(2+|s_1|+\cdots+|s_i|+|\bar{\mathsf{0}}|)(n+1)+|t|+1 &\leqslant |d|(n+1)+1\\ &< |d|(\mathrm{dp}(d)+n)\ .\end{aligned}$$

For (5.8g) the shape of d is $\mathsf{P}(s_1,\dots,\mathsf{S}s_i,\bar{\mathsf{0}},\mathsf{S}t)$. With the help of Lemma 7.24.ii,

$$\begin{aligned}\bullet^{n+1}d &\xrightarrow{+}{}^{=} && \mathsf{P}(s_1,\dots,\mathsf{S}[\!]^{n+1}s_i,\bar{\mathsf{0}},\mathsf{S}t)\\ &\to^{\leqslant}_{\mathrm{H}_i5} && \circ\mathsf{J}_i(s_1,\dots,[\!]^{n+1}s_i,\mathsf{P}(s_1,\dots,\mathsf{S}[\!]^{n+1}s_i,\bar{\mathsf{0}},t)) \qquad =: u\\ &\xrightarrow{+}{}^{\geqslant} && \circ\mathsf{J}_i(s_1,\dots,[\!]^{n+1}s_i,\mathsf{P}(s_1,\dots,\mathsf{S}s_i,\bar{\mathsf{0}},t))\\ &\xrightarrow{+} && \circ\mathsf{P}(s_1,\dots,s_i,\cdot,\bar{\mathsf{0}})^{n+1}(\mathsf{P}(s_1,\dots,\mathsf{S}s_i,\bar{\mathsf{0}},t))\end{aligned}$$

is possible. Thus we have to focus on two peaks. For the first one we get

$$\begin{aligned}|u| &= (2+|s_1|+\cdots+|s_i|)\cdot 2+2(n+1)+|\bar{\mathsf{0}}|+|t|\\ &< (3+|s_1|+\cdots+|s_i|+|\bar{\mathsf{0}}|+|t|)\cdot 2+2n\\ &= |d|\cdot 2+2n < |d|(2+n) < |d|(\mathrm{dp}(d)+n)\ ,\end{aligned}$$

while via Lemma 7.24.ii the second one is controlled by

$$\begin{aligned}&(2+|s_1|+\cdots+|s_i|+|\bar{\mathsf{0}}|)(n+1)+|\mathsf{P}(s_1,\dots,\mathsf{S}s_i,\bar{\mathsf{0}},t)|+1\\ &= (2+|s_1|+\cdots+|s_i|+|\bar{\mathsf{0}}|)(n+2)+|t|+1\\ &\leqslant (2+|s_1|+\cdots+|s_i|+|\bar{\mathsf{0}}|+|t|)(n+2) < |d|(\mathrm{dp}(d)+n)\ .\end{aligned}$$

Turning to (5.8i) we get $d=\mathsf{P}(s_1,\dots,s_i,\bar{\mathsf{0}},\mathsf{0})$ with $s_i\neq\mathsf{0}$. As mentioned in the proof of Proposition 5.46, there is $j\in[1,i]$ such that $\mathrm{MS}_{\bar{s},\bar{\mathsf{0}}}(\bar{s})$ is a subterm of s_j. The induction hypothesis and Lemma 7.20 yield

$$\begin{aligned}\bullet^{n+1}\mathsf{P}(s_1,\dots,s_i,\bar{\mathsf{0}},\mathsf{0}) &\xrightarrow{*}{}^{=} && \bullet\mathsf{P}(s_1,\dots,\bullet^{n}s_i,\bar{\mathsf{0}},\mathsf{0})\\ &\to_{\mathrm{H}_{ij}6} && \begin{cases}\mathsf{Q}_{ij}(s_1,\dots,\bullet^{n+1}s_i,s_j) & \text{if } j<i\ ,\\ \mathsf{Q}_{ij}(s_1,\dots,\bullet^{n+1}s_i,\bullet^{n}s_j) & \text{if } j=i\end{cases}\\ &\xrightarrow{*}{}^{\geqslant} && \mathsf{Q}_{ij}(s_1,\dots,\bullet^{n+1}s_i,s_j) \qquad\qquad (7.8a)\\ &\xrightarrow{+} && \mathsf{Q}_{ij}(s_1,\dots,\circ s_i[n],s_j)\\ &\to^{\leqslant}_{\mathrm{R}\mathsf{Q}_{ij}} && \circ\mathsf{P}(s_1,\dots,s_i[n],\bar{\mathsf{0}},s_j) \qquad\qquad (7.8b)\\ &\xrightarrow{*}{}^{\geqslant} && \circ\mathsf{P}(s_1,\dots,s_i[n],\bar{\mathsf{0}},\mathrm{MS}_{\bar{s},\bar{\mathsf{0}}}(\bar{s}))\ .\end{aligned}$$

First we consider, even for $j\neq i$,

$$\begin{aligned}|\mathsf{Q}_{ij}(s_1,\dots,\bullet^{n+1}s_i,\bullet^{n}s_j)| &= 1+|s_1|+\cdots+|s_i|+2n+1+|s_j|\\ &\leqslant (1+|s_1|+\cdots+|s_i|)\cdot 2+2n\\ &< |d|\cdot 2+2n < |d|(\mathrm{dp}(d)+n)\ ,\end{aligned}$$

while in a second step we care for the more difficult part from (7.8a) to (7.8b). Trusting the induction hypothesis, everything there is bounded by

$$\begin{aligned} &1 + |s_1| + \cdots + |s_{i-1}| + |s_i|(\mathrm{dp}(s_i) + n) + |\bar{\mathsf{0}}| + |s_j| \\ \leqslant\ &(1 + |s_1| + \cdots + |s_{i-1}| + |\bar{\mathsf{0}}|) \cdot 2 + |s_i|(\mathrm{dp}(s_i) + n + 1) < |d|(\mathrm{dp}(d) + n) \ . \end{aligned}$$

In a very similar way we can cover the remaining case (5.8j). There we have $d = \mathsf{P}(s_1, \ldots, s_i, \bar{\mathsf{0}}, \mathsf{S}t)$. The induction hypothesis yields

$$\begin{aligned} \bullet^{n+1}\mathsf{P}(s_1, \ldots, s_i, \bar{\mathsf{0}}, \mathsf{S}t) &\xrightarrow{*}{}^{=} \ \bullet\mathsf{P}(s_1, \ldots, \bullet^{n} s_i, \bar{\mathsf{0}}, \mathsf{S}t) \\ &\to_{\mathrm{H}_i 7} \mathsf{R}_i(s_1, \ldots, \bullet^{n+1} s_i, \bullet^{n} s_i, t) \\ &\xrightarrow{*}{}^{\geqslant} \ \mathsf{R}_i(s_1, \ldots, \bullet^{n+1} s_i, s_i, t) \\ &\xrightarrow{+} \ \mathsf{R}_i(s_1, \ldots, \circ s_i[n], s_i, t) \\ &\to^{\leqslant}_{\mathrm{R}\mathsf{R}_i} \circ\mathsf{P}(s_1, \ldots, s_i[n], \bar{\mathsf{0}}, \mathsf{P}(s_1, \ldots, s_i, \bar{\mathsf{0}}, t)) \ , \end{aligned}$$

so again we have to check two peaks. For the first one we get

$$\begin{aligned} |\mathsf{R}_i(s_1, \ldots, \bullet^{n+1} s_i, \bullet^{n} s_i, t)| &= 1 + |s_1| + \cdots + |s_i| + 2n + 1 + |s_i| + |t| \\ &\leqslant (1 + |s_1| + \cdots + |s_i| + |t|) \cdot 2 + 2n \\ &< |d|(\mathrm{dp}(d) + n) \ , \end{aligned}$$

while the second one is bounded by

$$\begin{aligned} &(1 + |s_1| + \cdots + |s_{i-1}| + |\bar{\mathsf{0}}|) \cdot 2 + |s_i|(\mathrm{dp}(s_i) + n) + |s_i| + |t| \\ \leqslant\ &(1 + |s_1| + \cdots + |s_{i-1}| + |\bar{\mathsf{0}}| + |t|) \cdot 2 + |s_i|(\mathrm{dp}(s_i) + n + 1) \\ <\ &|d|(\mathrm{dp}(d) + n) \ . \end{aligned}$$ □

Recall from Definition 5.38 that

$$d[n, m] = (\ldots((d[n])[n+1])\ldots)[m] \ .$$

Corollary 7.26. *For $d \in \mathcal{D}$ and $n, m \in \mathbb{N}$ we have*

$$|d[n, n+m-1]| \leqslant |d| \cdot \prod_{j=n}^{n+m-1} (\mathrm{dp}(d[n, j-1]) + j) \ .$$

Proof by induction on m. The case $m = 0$ is trivial as here $d[n, n+m-1] = d$ holds. Let us turn to the step from m to $m+1$. Since

$$d[n, n+m] = (d[n, n+m-1])[n+m] \ ,$$

this is only interesting if $d[n, n+m-1] \neq \mathsf{0}$, and in this case Proposition 7.25 and the induction hypothesis yield

$$\begin{aligned}
&|d[n, n+m-1][n+m]| \\
&< |d[n, n+m-1]|(\mathrm{dp}(d[n, n+m-1]) + n + m) \\
&\leqslant \left(|d| \cdot \prod_{j=n}^{n+m-1} (\mathrm{dp}(d[n, j-1]) + j) \right) (\mathrm{dp}(d[n, n+m-1]) + n + m) \\
&= |d| \cdot \prod_{j=n}^{n+m} (\mathrm{dp}(d[n, j-1]) + j)\ . \qquad \square
\end{aligned}$$

Corollary 7.26 indicates a need to impose a bound on $\mathrm{dp}(d[n, n+j])$. This gets a bit more involved than it should be, partially because of the way the set $\mathcal{D}$ of terms in normal form is defined.

Lemma 7.27. *For $s, \bar{s}, t \in \mathcal{D}$ and $n \in \mathbb{N}$ we have*

i. $\mathrm{dp}(s \times (n+1)) = \mathrm{dp}(s) + n$

ii. $\mathrm{dp}(\mathsf{P}(\bar{s}, \cdot, \bar{\mathsf{0}})^{n+1}(t)) = \mathrm{dp}(\mathsf{P}(\bar{s}, t, \bar{\mathsf{0}})) + n.$

Proof. As both items are shown by similar inductions on n, we focus on the shorter (i). Here the case $n = 0$ is trivial since $s \times 1 = s$ holds, and for $n > 0$ Definition 5.36 and the induction hypothesis yield

$$\begin{aligned}
\mathrm{dp}(s \times (n+1)) &= \mathrm{dp}(+(s \times n, s)) \\
&= 1 + \max\{\mathrm{dp}(s) + n - 1, \mathrm{dp}(s)\} = \mathrm{dp}(s) + n\ . \qquad \square
\end{aligned}$$

Proposition 7.28. *Let $d \in \mathcal{D}$ and $n \in \mathbb{N}$.*

i. *If $d \in \mathcal{D}(\mathsf{Lim})$ and $d[n] = \mathsf{S}^i d'$ where i is as large as possible, then either $i = 0$ or $i = n+1$ holds.*

ii. *We have* $\mathrm{dp}(d[n]) \leqslant \mathrm{dp}(d) + n + 1$.

Proof. We show (i) by induction on $\mathcal{D}$. Recall from Definition 5.37 that $\mathcal{D}(\mathsf{Lim})$ contains those elements of $\mathcal{D}$ whose values are limit ordinals. Looking at Definition 5.38 we first note that $d\langle n \rangle$ does not start with S, which leaves us with the three cases where $d\langle n \rangle \neq d[n]$. In the first one we have $d = \mathsf{P}(\bar{\mathsf{0}}, \mathsf{S0})$ and $d[n] = \mathsf{S}^{n+1}\mathsf{0}$, which is just as we promised. For the second case, $d = \mathsf{P}(\mathsf{0}, \ldots, \mathsf{S0}, \bar{\mathsf{0}}, \mathsf{0})$ and $d[n] = \mathsf{P}(\mathsf{0}, \ldots, \mathsf{0}, \cdot, \bar{\mathsf{0}})^n(\mathsf{S0})$ hold. This is only of interest if $n = 0$. Since this implies $d[n] = \mathsf{S0} = \mathsf{S}^{n+1}\mathsf{0}$, everything is correct here. It remains to treat the case $d = +(s, t)$. According to Definition 5.34, both s and t are members of $\mathcal{D}(\mathsf{Lim})$. From the induction hypothesis we know that if $t[n] = \mathsf{S}^j t'$ holds with j as large as possible, then either $j = 0$ or $j = n+1$. Depending on t',

$d[n]$ either equals $\mathsf{S}^j s$ or $\mathsf{S}^j{+}(s,t')$. While the second term has the announced shape, we have to check for the first term that s does not start with S. Since $s \in \mathcal{D}(\mathsf{Lim})$ holds, we are safe.

We show (ii) by induction on $\mathcal{D}$. If $d = \mathsf{0}$ or $d = \mathsf{S}s$, then $\mathrm{dp}(d[n]) \leqslant \mathrm{dp}(d)$ holds. Let us turn to the cases where $d\langle n\rangle$ and $d[n]$ may differ. We treat them in the same order as in (i). For $d = \mathsf{P}(\bar{\mathsf{0}}, \mathsf{S0})$ it is not difficult to see

$$\mathrm{dp}(d[n]) = \mathrm{dp}(\mathsf{S}^{n+1}\mathsf{0}) = 1 + n + 1 < \mathrm{dp}(d) + n + 1 \ ,$$

while for $d = \mathsf{P}(\mathsf{0}, \dots, \mathsf{S0}, \bar{\mathsf{0}}, \mathsf{0})$ we rely on Lemma 7.27.ii and get

$$\mathrm{dp}(d[n]) = \mathrm{dp}(\mathsf{P}(\bar{\mathsf{0}}, \cdot, \bar{\mathsf{0}})^n(\mathsf{S0})) = \mathrm{dp}(d) + n - 1 \ .$$

In the case $d = {+}(s,t)$ the induction hypothesis yields $\mathrm{dp}(t[n]) \leqslant \mathrm{dp}(t) + n + 1$. If $d[n] = {+}(s, t[n])$, then the claim easily follows from

$$\begin{aligned}\mathrm{dp}({+}(s,t[n])) &\leqslant 1 + \max\{\mathrm{dp}(s), \mathrm{dp}(t) + n + 1\}\\ &\leqslant 1 + \max\{\mathrm{dp}(s), \mathrm{dp}(t)\} + n + 1 = \mathrm{dp}(d) + n + 1 \ ,\end{aligned}$$

while otherwise (i) implies $t[n] = \mathsf{S}^{n+1}t'$. This yields $\mathrm{dp}(t') \leqslant \mathrm{dp}(t)$ and either $d[n] = \mathsf{S}^{n+1}s$ (iff $t' = \mathsf{0}$) or $d[n] = \mathsf{S}^{n+1}{+}(s,t')$. We arrive at

$$\mathrm{dp}(d[n]) \leqslant \mathrm{dp}(\mathsf{S}^{n+1}{+}(s,t')) \leqslant n + 1 + \mathrm{dp}(d) \ .$$

It remains to check the cases where $d\langle n\rangle = d[n]$ holds directly by definition. Since (5.8d) and (5.8k) are of similar shape, we only demonstrate how to treat the former. Here we have $d = \mathsf{P}(\bar{s}, t)$, and the induction hypothesis yields

$$\begin{aligned}\mathrm{dp}(d[n]) &= \mathrm{dp}(\mathsf{P}(\bar{s}, t[n]))\\ &\leqslant 1 + \max\{\mathrm{dp}(s_1), \dots, \mathrm{dp}(s_k), \mathrm{dp}(t) + n + 1\}\\ &\leqslant 1 + \max\{\mathrm{dp}(s_1), \dots, \mathrm{dp}(s_k), \mathrm{dp}(t)\} + n + 1 = \mathrm{dp}(d) + n + 1 \ .\end{aligned}$$

For (5.8e) we have $d = \mathsf{P}(\bar{\mathsf{0}}, \mathsf{S}t)$ and get

$$\mathrm{dp}(d[n]) = \mathrm{dp}(\mathsf{P}(\bar{\mathsf{0}}, t) \times (n+1)) = \mathrm{dp}(\mathsf{P}(\bar{\mathsf{0}}, t)) + n < \mathrm{dp}(d) + n \ ,$$

where we relied on Lemma 7.27.i. The treatment of (5.8f) is completely analogous, so we omit it. In the case (5.8g) with $d = \mathsf{P}(s_1, \dots, \mathsf{S}s_i, \bar{\mathsf{0}}, \mathsf{S}t)$ we have to be more verbose. Using Lemma 7.27.ii we see

$$\begin{aligned}\mathrm{dp}(d[n]) &= \mathrm{dp}(\mathsf{P}(\bar{s}, \cdot, \bar{\mathsf{0}})^{n+1}(\mathsf{P}(s_1, \dots, \mathsf{S}s_i, \bar{\mathsf{0}}, t)))\\ &= \mathrm{dp}(\mathsf{P}(\bar{s}, \mathsf{P}(s_1, \dots, \mathsf{S}s_i, \bar{\mathsf{0}}, t), \bar{\mathsf{0}})) + n\\ &= 1 + \mathrm{dp}(\mathsf{P}(s_1, \dots, \mathsf{S}s_i, \bar{\mathsf{0}}, t)) + n \leqslant 1 + \mathrm{dp}(d) + n \ .\end{aligned}$$

The proof for (5.8h) is immediate, hence we may turn to (5.8i) with $d = \mathsf{P}(s_1, \dots, s_i, \bar{\mathsf{0}}, \mathsf{0})$, invocate the induction hypothesis, and get

$$\begin{aligned} \mathrm{dp}(d[n]) &= \mathrm{dp}(\mathsf{P}(s_1, \dots, s_i[n], \bar{\mathsf{0}}, \mathrm{MS}_{\bar{s},\bar{\mathsf{0}}}(\bar{s}))) \\ &\leqslant \max\{\mathrm{dp}(d), 1 + \mathrm{dp}(s_i[n])\} \\ &\leqslant \max\{\mathrm{dp}(d), 1 + \mathrm{dp}(s_i) + n + 1\} \leqslant \mathrm{dp}(d) + n + 1 \;, \end{aligned}$$

since $\mathrm{MS}_{\bar{s},\bar{\mathsf{0}}}$ is a subterm of some s_j. Finally, (5.8j) with $d = \mathsf{P}(s_1, \dots, s_i, \bar{\mathsf{0}}, \mathsf{S}t)$ is handled once more by the induction hypothesis:

$$\begin{aligned} \mathrm{dp}(d[n]) &= \mathrm{dp}(\mathsf{P}(s_1, \dots, s_i[n], \bar{\mathsf{0}}, \mathsf{P}(s_1, \dots, s_i, \bar{\mathsf{0}}, t))) \\ &= 1 + \max\{\mathrm{dp}(s_i[n]), \mathrm{dp}(\mathsf{P}(s_1, \dots, s_i, \bar{\mathsf{0}}, t))\} \\ &\leqslant 1 + \max\{\mathrm{dp}(s_i) + n + 1, \mathrm{dp}(d)\} \\ &\leqslant 1 + \max\{\mathrm{dp}(d) + n, \mathrm{dp}(d)\} = 1 + \mathrm{dp}(d) + n \;. \end{aligned}$$ □

Corollary 7.29. *For $d \in \mathcal{D}$ and $n, m \in \mathbb{N}$ we have*

$$\mathrm{dp}(d[n, n+m-1]) \leqslant \mathrm{dp}(d) + \sum_{j=n}^{n+m-1} (j+1) \;.$$

Proof by induction on m. If $m = 0$, then $d[n, n+m-1] = d$, hence there is nothing to do. For the step from m to $m+1$ we see

$$\begin{aligned} \mathrm{dp}(d[n, n+m]) &= \mathrm{dp}((d[n, n+m-1])[n+m]) \\ &\leqslant \mathrm{dp}(d[n, n+m-1]) + n + m + 1 \\ &\leqslant \mathrm{dp}(d) + \sum_{j=n}^{n+m-1} (j+1) + n + m + 1 = \mathrm{dp}(d) + \sum_{j=n}^{n+m} (j+1) \;, \end{aligned}$$

relying on Proposition 7.28.ii and the induction hypothesis. □

7.3 Lengthened Derivations

We still keep the $k > 0$ of Section 7.2 fixed. Everything is prepared to combine the signatures Σ and Σ_p (from Definitions 5.40 and 7.10) and to combine the TRSs $\mathcal{R}'$ and $\mathcal{R}_p$ (from Definitions 7.18 and 7.15), generating totally terminating TRSs with very long size-controlled derivations.

Definition 7.30. For $p > 1$ (and our fixed k) we introduce the signature $\Sigma'_p := \Sigma \cup \Sigma_p$. The TRS $\mathcal{R}'_p$ over Σ'_p comprises $\mathcal{R}'$ and $\mathcal{R}_p$, and the rule

$$\text{(Z0)} \quad +(\circ x, \bar{0}\, y) \to +(\bullet\square x, \underline{1}\, \underline{0}\, y) \;.$$

Proposition 7.31. *For any $p > 1$, the TRS $\mathcal{R}'_p$ is totally terminating.*

Proof. The main idea consists in combining the weak monotone algebras we constructed for the separate TRSs as witnesses of weak total termination.

In Proposition 7.19 we used the Σ-algebra $(\mathcal{P}, \prec, \mathcal{F}_\mathcal{P})$ (which was introduced in Theorem 5.50) to establish total termination of $\mathcal{R}'$. Total termination of $\mathcal{R}_p$ was demonstrated in Lemma 7.16 by constructing a weak monotone Σ_p-algebra $(\mathcal{Q}, \lhd, \mathcal{F}_p)$, where $(\mathcal{Q}, \lhd)$ was the lexicographic product of a suitable well-order and a KBO $(\mathcal{T}(\Sigma_p^-), \prec_{\mathsf{kbo}})$. An inspection of the proof of Lemma 7.16 shows the well-order $(\mathcal{P}, \prec)$ is suitable, with the canonical successor function $(\alpha, m, n) \mapsto (\alpha, m, n+1)$, better known as $[\bullet]_\mathcal{P}$, serving as the S needed there. Of course, the $\prec$-minimal element of $\mathcal{P}$ is $(1, 0, 0) = [\![\mathsf{0}]\!]_\mathcal{P}$.

Let $(\mathcal{Q}, \lhd)$ be the well-order which originates from using $(\mathcal{P}, \prec)$ in the proof of Lemma 7.16. We have to provide a set $\mathcal{F}$ of Σ'_p-operations which make up a Σ'_p-algebra compatible with $\mathcal{R}'_p$. For the symbols in Σ_p this is a fairly easy task, since for them we can simply take the members of $\mathcal{F}_p$. In the proof of Lemma 7.16 we showed they are weakly monotone and have the subterm property, and moreover any Σ'_p-algebra which is based on these functions is compatible with $\mathcal{R}_p$.

It remains to treat the symbols of Σ. For any $f \in \Sigma$ we put

$$[f]((q_1, t_1), \ldots, (q_n, t_n)) := ([f]_\mathcal{P}(\bar{q}), \blacktriangle) \ ,$$

thus we completely ignore the t_i. A moment's reflection shows $[f]$ inherits weak monotonicity and the subterm property from $[f]_\mathcal{P}$.

Fix $\rho\colon \mathcal{V} \to \mathcal{Q}$ and let ρ' be the associated mapping from $\mathcal{V}$ to $\mathcal{P}$ satisfying

$$(\forall x \in \mathcal{V})(\exists t \in \mathcal{T}(\Sigma_p^-))(\rho(x) = (\rho'(x), t)) \ .$$

An induction on $s \in \mathcal{T}(\Sigma, \mathcal{V})$ shows that we can always find $t \in \mathcal{T}(\Sigma_p^-)$ such that

$$[\![s, \rho]\!] = \begin{cases} ([\![s, \rho']\!]_\mathcal{P}, t) & \text{if } s \in \mathcal{V} \ , \\ ([\![s, \rho']\!]_\mathcal{P}, \blacktriangle) & \text{otherwise} \ . \end{cases} \tag{7.10}$$

This is no surprise as, for $f \in \Sigma$, the $[f]_\mathcal{P}$ ignore and delete the effects symbols like $\overline{0}$ had on their input.

In Proposition 7.19 we showed that $(\mathcal{P}, \prec, \mathcal{F}_\mathcal{P})$ is compatible with $\mathcal{R}'$, and this means we have $l \succ_\mathcal{P} r$ for all rules $(l, r) \in \mathcal{R}'$. This implies $[\![l, \rho']\!]_\mathcal{P} \succ [\![r, \rho']\!]_\mathcal{P}$. We can rely on this to establish compatibility of $(\mathcal{Q}, \lhd, \mathcal{F})$ and $\mathcal{R}'$, since for $(l, r) \in \mathcal{R}'$ we get, using (7.10), t_l and $t_r \in \mathcal{T}(\Sigma_p^-)$ satisfying

$$[\![l, \rho]\!] = ([\![l, \rho']\!]_\mathcal{P}, t_l) \rhd ([\![r, \rho']\!]_\mathcal{P}, t_r) = [\![r, \rho]\!] \ .$$

In a final effort we have to take care of (Z0). For $\rho(x) = ((\alpha, m, n), t)$ and $\rho(y) = ((\alpha', m', n'), t')$ we just calculate:

$$\begin{aligned}
\llbracket +(\circ x, \bar{0}\, y), \rho \rrbracket &= [+](((\alpha+1, 0, 0), \blacktriangle), ((\alpha', m', n'), \bar{0}\, t')) \\
&= ([+]_{\mathcal{P}}((\alpha+1, 0, 0), (\alpha', m', n')), \blacktriangle) \\
&= (((\alpha+1) \oplus \alpha' \oplus \alpha', 0, 0), \blacktriangle) \\
&\rhd ((\alpha \oplus \alpha' \oplus \alpha', 0, 0), \blacktriangle) \\
&= ([+]_{\mathcal{P}}((\alpha, 1, 1), (\alpha', m', n')), \blacktriangle) \\
&= [+](((\alpha, 1, 1), \blacktriangle), ((\alpha', m', n'), \underline{1}\, \underline{0}\, t')) \\
&= \llbracket +(\bullet\Box x, \underline{1}\, \underline{0}\, y), \rho \rrbracket \; .
\end{aligned}$$

As usual, Theorem 3.45 implies $\mathcal{R}'_p$ is totally terminating. □

Recall from Lemma 5.31.ii and Lemma 5.39 that the length of the battle for the configuration $(\mathrm{val}(d), n)$ is the minimal m which satisfies $d[n, n+m-1] = \mathsf{0}$. We will construct very long derivations which are controlled in a special way by some function B. Once we achieve this, the rest of the time is spent showing that, for rather low r, the size bounding function f_r is compatible with B.

Proposition 7.32. *Let $p > 1$ and $i, n \in \mathbb{N}$. For $d \in \mathcal{D} \setminus \{\mathsf{0}\}$ and $m > 0$ we put*

$$d_m := \circ\Box^{n+m-1} d[n, n+m-2] \quad \textit{and} \quad u_m := +(d_{|m|_p^{(i+1)}}, \mathrm{logs}_{p,i}(m)) \; .$$

Further let L be the length of the battle for the configuration $(\mathrm{val}(d), n)$, and let L' be minimal such that $|L'|_p^{(i+1)} = L + 1$. There exists a derivation

$$u_1 \xrightarrow{+} u_2 \xrightarrow{+} \ldots \xrightarrow{+} u_m \xrightarrow{+} u_{m+1} \xrightarrow{+} \ldots \xrightarrow{+} u_{L'}$$

in $\mathcal{R}'_p$ such that, for all $m \in [1, L']$, the part $u_1 \xrightarrow{*} u_m$ is bounded by

$$B(m) := 1 + n + |m|_p^{(i+1)} + |\mathrm{logs}_{p,i}(m)| + |d| \cdot \prod_{j=n}^{n+|m|_p^{(i+1)}-2} (\mathrm{dp}(d[n, j-1]) + j) \; .$$

Proof. By induction on $m \in [1, L']$ we show that there is a derivation $u_1 \xrightarrow{*} u_m$ which is bounded by $B(m)$. The case $m = 1$ only affords an estimation. Because of $|1|_p^{(i+1)} = 1$ we get

$$|u_1| = |{+}(\circ\Box^n d, \mathrm{logs}_{p,i}(1))| = 1 + 1 + n + |d| + |\mathrm{logs}_{p,i}(1)| = B(1) \; .$$

Let us turn to the step from m to $m+1$. By the induction hypothesis we have a derivation $u_1 \xrightarrow{*} u_m$ which is bounded by $B(m) \leqslant B(m+1)$, so it remains to find a suitable derivation $u_m \overset{+}{\rightarrow} u_{m+1}$. In the easier case $|m+1|_p^{(i+1)} = |m|_p^{(i+1)}$ Lemma 7.14.i and Lemma 7.17 imply

$$\mathrm{logs}_{p,i}(m) \overset{+}{\rightarrow}^{\leqslant} \mathrm{logs}^{+}_{p,i}(m) = \mathrm{logs}_{p,i}(m+1) \ ,$$

yielding

$$\begin{aligned} u_m = {} & +(d_{|m|_p^{(i+1)}}, \mathrm{logs}_{p,i}(m)) = +(d_{|m+1|_p^{(i+1)}}, \mathrm{logs}_{p,i}(m)) \\ \overset{+}{\rightarrow}^{\leqslant} {} & +(d_{|m+1|_p^{(i+1)}}, \mathrm{logs}_{p,i}(m+1)) = u_{m+1} \ . \end{aligned}$$

To confirm the bound we take a look at the function C which splits up B via

$$C(j) := B(j) - |\mathrm{logs}_{p,i}(j)| \ .$$

We already know $|u_m| \leqslant B(m)$, and additionally the only difference between u_m and u_{m+1} is the part containing $\mathrm{logs}_{p,i}$. Thus $|u_{m+1}| \leqslant C(m) + |\mathrm{logs}_{p,i}(m+1)|$ holds. Because of $C(m) = C(m+1)$ we find the derivation from u_m to u_{m+1} is bounded by $B(m+1)$.

Let us now consider the remaining case $|m+1|_p^{(i+1)} = |m|_p^{(i+1)} + 1$. We put $l := |m|_p^{(i+1)}$. By Definition 7.13 and Lemma 7.17 there exists a term t such that

$$\mathrm{logs}_{p,i}(m) \overset{+}{\rightarrow}^{\leqslant} \mathrm{logs}^{+}_{p,i}(m) = \mathrm{code}^{+}_{p,i}(m)t = \overline{0}\,\underline{0}^{l-1}t \qquad (7.11a)$$

and

$$\mathrm{logs}_{p,i}(m+1) = \underline{1}\,\underline{0}^{l}t \ . \qquad (7.11b)$$

Now we are ready to go:

$$\begin{aligned} u_m = {} & +(\circ\square^{n+l-1}d[n, n+l-2], \mathrm{logs}_{p,i}(m)) & \\ \overset{+}{\rightarrow}^{\leqslant} {} & +(\circ\square^{n+l-1}d[n, n+l-2], \overline{0}\,\underline{0}^{l-1}t) & \text{by (7.11a)} \\ \rightarrow^{\leqslant}_{\mathrm{Z0}} {} & +(\bullet\square^{n+l}d[n, n+l-2], \underline{1}\,\underline{0}^{l}t) & \\ = {} & +(\bullet\square^{n+l}d[n, n+l-2], \mathrm{logs}_{p,i}(m+1)) & \text{by (7.11b)} \\ \overset{+}{\rightarrow}^{\leqslant} {} & +(\square^{n+l}\bullet^{n+l}d[n, n+l-2], \mathrm{logs}_{p,i}(m+1)) & \text{by Lemma 7.22.ii} \\ \overset{+}{\rightarrow} {} & +(\square^{n+l}\circ d[n, n+l-1], \mathrm{logs}_{p,i}(m+1)) & \text{by Proposition 7.25} \\ \overset{+}{\rightarrow}^{=} {} & +(\circ\square^{n+l}d[n, n+l-1], \mathrm{logs}_{p,i}(m+1)) & \text{by Lemma 7.22.iii} \\ = {} & u_{m+1} \ . & \end{aligned}$$

We want to take a closer look at why the use of Proposition 7.25 was legal. Because of $m < L'$ and the minimality condition on L' we have $l < L+1$, hence $l-2 < L-1$. Since L is minimal satisfying $d[n, n+L-1] = \mathsf{0}$ we get

$d[n,n+l-2] \neq \mathsf{0}$, which is exactly what we needed to safely apply Proposition 7.25. This Proposition is also useful for establishing the upper bound, since according to it all terms in the above derivation are bounded by

$$\begin{aligned} C := {} & 1+n+l+|\mathrm{logs}_{p,i}(m+1)| \\ & + |d[n,n+l-2]|(\mathrm{dp}(d[n,n+l-2])+n+l-1) \ . \end{aligned}$$

By Corollary 7.26 we get

$$\begin{aligned} & |d[n,n+l-2]|(\mathrm{dp}(d[n,n+l-2])+n+l-1) \\ \leqslant & \left(|d| \cdot \prod_{j=n}^{n+l-2} (\mathrm{dp}(d[n,j-1])+j) \right) (\mathrm{dp}(d[n,n+l-2])+n+l-1) \\ = & |d| \cdot \prod_{j=n}^{n+l-1} (\mathrm{dp}(d[n,j-1])+j) \ . \end{aligned}$$

Since $l = |m+1|_p^{(i+1)} - 1$, the above bound C is in turn bounded by

$$1+n+|m+1|_p^{(i+1)} + |\mathrm{logs}_{p,i}(m+1)| + |d| \cdot \prod_{j=n}^{n+|m+1|_p^{(i+1)}-2} (\mathrm{dp}(d[n,j-1])+j) \ ,$$

and this is exactly $B(m+1)$. □

The next Lemma gathers various auxiliary results we will need as the proof of Proposition 7.34 evolves.

Lemma 7.33. *Let $n, m \in \mathbb{N}$.*

i. *The equation $\sum\limits_{k=0}^{n} k = \frac{1}{2}n(n+1)$ is valid.*
ii. *We have $n(n+1) \leqslant 2n^2$.*
iii. *If $n \geqslant 1$, then $n! \leqslant n^n$.*
iv. *If $n \geqslant 2$ and $m \geqslant 4$, then $m^2 \leqslant n^m$.*
v. *For $a, b \in \mathbb{R}$ we have $2ab \leqslant a^2 + b^2$.*
vi. *For almost all n we have $(\forall m \geqslant 1)((n+m)^3 \leqslant \frac{1}{8}n^4 + 82m^4)$.*

Proof. The well known (i) is shown by induction on n, while (ii) immediately follows from $n \leqslant n^2$. The proof of (iii) is immediate, whereas (iv) is treated by induction on m. In the case $m = 4$ we should take $4^2 = 2^4 \leqslant n^4$ for granted, and the step from m to $m+1$ is achieved by

$$(m+1)^2 = m^2 + 2m + 1 \leqslant m^2 + 4m \leqslant 2m^2 \leqslant 2n^m \leqslant nn^m = n^{m+1} \ .$$

For (v) we rely on $0 \leqslant (a-b)^2 = a^2 + b^2 - 2ab$, and finally (vi) is won by

$$\begin{aligned}(n+m)^3 &= n^3 + 3n^2m + 3nm^2 + m^3 \leqslant n^3 + 6n^2m^2 + m^3 \\ &= n^3 + 2(\tfrac{1}{3}n^2)(9m^2) + m^3 \leqslant n^3 + (\tfrac{1}{3}n^2)^2 + (9m^2)^2 + m^3 \\ &= n^3 + \tfrac{1}{9}n^4 + 81m^4 + m^3 \leqslant \tfrac{1}{8}n^4 + 82m^4 \ ,\end{aligned}$$

where we used (v) and $n \geqslant 72$. □

Recall from Definition 7.4 that $K_f^{\mathcal{R}}$ maps n to the first N for whom there is no sequence $s_1 \to_{\mathcal{R}} \ldots \to_{\mathcal{R}} s_N$ satisfying $(\forall m \in [1,N])(|s_m| \leqslant f(n,m))$. We are going to show that such functions grow rather fast.

Proposition 7.34. *Let $p > 1$. For real $r > 1$, put $g_r(n,m) := n + r\log_p(m)$ and write K_r for $K_{g_r}^{\mathcal{R}'_p}$. This K_r eventually dominates all $<\Delta_k$-recursive functions.*

Proof. Take $r > 1$ and fix $i > 2$. For $n \in \mathbb{N}$ we put $t_n := \mathsf{P}(\cdot, \bar{0})^n(\mathsf{S}0)$. Since P is $k+1$-ary, a short induction and Lemma 7.27.ii show

$$|t_n| = (k+1)\cdot n + 2 \quad \text{and} \quad \mathrm{dp}(t_n) = n + 2 \ . \tag{7.12}$$

Let L be the function which maps n to the length of the battle for the configuration $(\mathrm{val}(t_n), n+1)$. In the proof of Corollary 5.47 we used the fact that L eventually dominates all $<\Delta_k$-recursive functions. Our aim is now to establish

$$K_r(p^{n^4}) > L(n) \quad \text{for almost all } n \ . \tag{7.13}$$

Since exponential functions are elementary, α-recursiveness is not affected by substituting exponential functions. This immediately implies K_r eventually dominates all $<\Delta_k$-recursive functions.

Fix some n. By Proposition 7.32 (substituting d with t_n and n with $n+1$), there exist $L' > L(n)$ and a derivation

$$u_1 \xrightarrow{+} u_2 \xrightarrow{+} \ldots \xrightarrow{+} u_m \xrightarrow{+} u_{m+1} \xrightarrow{+} \ldots \xrightarrow{+} u_{L'}$$

in $\mathcal{R}'_p$ whose part $u_1 \xrightarrow{*} u_m$ is bounded by

$$B_n(m) := 2 + n + |m|_p^{(i+1)} + |\mathrm{logs}_{p,i}(m)| + |t_n| \cdot \prod_{j=n+1}^{n+|m|_p^{(i+1)}-1} (\mathrm{dp}(t_n[n+1, j-1]) + j) \ .$$

Since $B_n(m) \leqslant B_n(m+1)$ holds if $1 \leqslant m < L'$, we may focus on the first $L(n)$ steps in this derivation. Thus we consider the derivation

$$u'_1 \to u'_2 \to \ldots \to u'_m \to u'_{m+1} \to \ldots \to u'_{L(n)}$$

where $u'_1 \xrightarrow{*} u'_m$ is still bounded by $B_n(m)$, so especially

$$(\forall m \in [1, L(n)])(|u'_m| \leqslant B_n(m))$$

holds. If we succeed in proving

$$(\text{for almost all } n)(\forall m \in [1, L(n)])(B_n(m) \leqslant g_r(p^{n^4}, m)) \ , \tag{7.14}$$

we immediately get (7.13). This appears to be a good reason to show (7.14).

From now on, let n be large (what large means will become apparent as the proof evolves), and fix $m \in [1, L(n)]$. As in the proof of Proposition 7.32 we put $l := |m|_p^{(i+1)}$, and additionally we set $o := |\mathrm{logs}_{p,i}(m)|$. Lemma 7.14.ii implies

$$o = i + 1 + \sum_{j=1}^{i+1} |m|_p^{(j)} \ . \tag{7.15}$$

By (7.12) and Corollary 7.29, any n satisfies

$$\begin{aligned}
&|t_n| \cdot \prod_{j=n+1}^{n+l-1} (j + \mathrm{dp}(t_n[n+1, j-1])) \\
&\leqslant ((k+1) \cdot n + 2) \cdot \prod_{j=n+1}^{n+l-1} (j + n + 2 + \sum_{j'=n+1}^{j-1} (j'+1)) \\
&= ((k+1) \cdot n + 2) \cdot \prod_{j=n+1}^{n+l-1} \sum_{j'=n}^{j} (j'+1) \ .
\end{aligned}$$

For large n this implies

$$\begin{aligned}
B_n(m) &\leqslant 2 + n + l + o + ((k+1) \cdot n + 2) \cdot \prod_{j=n+1}^{n+l-1} \sum_{j'=n}^{j} (j'+1) \\
&\leqslant l + o + ((k+2) \cdot n + 4) \cdot \prod_{j=n+1}^{n+l-1} \sum_{j'=n}^{j} (j'+1) \\
&\leqslant l + o + \prod_{j=n-1}^{n+l-1} \sum_{j'=n-1}^{j} (j'+1) && (7.16a) \\
&\leqslant l + o + \prod_{j=n-1}^{n+l-1} \sum_{j'=1}^{j} (j'+1) \\
&= l + o + \prod_{j=n-1}^{n+l-1} \tfrac{1}{2}(j+1)(j+2) && (7.16b) \\
&< l + o + (\tfrac{1}{2}(n+l)(n+l+1))! \ . && (7.16c)
\end{aligned}$$

The step towards (7.16a) is justified because $((k+2)\cdot n+4) \leqslant n(2n+1)$ holds if $k+2 \leqslant n$, while the step to (7.16b) is a consequence of Lemma 7.33.i. We safely reach (7.16c), as the components of the product form a nonrepeating and nonexhaustive sequence of numbers bounded by $\frac{1}{2}(n+l)(n+l+1)$. The factorial in this bound is now replaced with something more pleasant:

$$\begin{aligned}
\big(\tfrac{1}{2}(n+l)(n+l+1)\big)! &\leqslant \big((n+l)^2\big)! && \text{by Lemma 7.33.ii}\\
&\leqslant \big((n+l)^2\big)^{(n+l)^2} && \text{by Lemma 7.33.iii}\\
&\leqslant p^{(n+l)^3} && \text{by Lemma 7.33.iv}\\
&< p^{\frac{1}{8}n^4+82l^4} && \text{by Lemma 7.33.vi}\\
&< p^{\frac{1}{4}n^4+\frac{1}{2}(|m|_p^{(i)}-1)} && \text{by Lemma 7.9.iii}\\
&= p^{\frac{1}{4}n^4}\cdot p^{\frac{1}{2}(|m|_p^{(i)}-1)}\\
&\leqslant \big(p^{\frac{1}{4}n^4}\big)^2+\big(p^{\frac{1}{2}(|m|_p^{(i)}-1)}\big)^2 && \text{by Lemma 7.33.v}\\
&= p^{\frac{1}{2}n^4}+p^{|m|_p^{(i)}-1}\\
&\leqslant p^{\frac{1}{2}n^4}+p^{\log_p(|m|_p^{(i-1)})} && \text{by Lemma 7.9.i}\\
&= p^{\frac{1}{2}n^4}+|m|_p^{(i-1)}\ .
\end{aligned}$$

Maybe the above use of Lemma 7.9.iii should be explained in more detail. By the Lemma we have

$$82(|m|_p^{(i+1)})^4 < \tfrac{1}{2}(|m|_p^{(i)}-1) \quad \text{for almost all } m\ ,$$

hence there exists $N \in \mathbb{N}$ with

$$(\forall m)(82(|m|_p^{(i+1)})^4 < \tfrac{1}{2}(|m|_p^{(i)}-1)+N)\ .$$

Since n is large, we may safely assume $N < \frac{1}{8}n^4$.

We combine the results presented so far to arrive at

$$\begin{aligned}
&B_n(m)\\
&< l+o+p^{\frac{1}{2}n^4}+|m|_p^{(i-1)} && \text{by (7.16)}\\
&= p^{\frac{1}{2}n^4}+|m|_p^{(i+1)}+|m|_p^{(i-1)}+i+1+\sum_{j=1}^{i+1}|m|_p^{(j)} && \text{by (7.15)}\\
&\leqslant p^{\frac{1}{2}n^4}+(i+2)|m|_p^{(2)}+i+1+|m|_p && \text{by Lemma 7.9.iii}\\
&\leqslant p^{\frac{1}{2}n^4}+(i+2)(|m|_p^{(2)}+1)+\log_p(m) && \text{by Lemma 7.9.i}\\
&< p^{n^4}+r\log_p(m) && \text{by Lemma 7.9.iii}\\
&= g_r(p^{n^4},m)\ ,
\end{aligned}$$

and this is (7.14). The first use of Lemma 7.9.iii is justified by $i > 2$ because, except for $|m|_p^{(1)} = |m|_p$, all occurrences of $|m|_p^{(i')}$ satisfy $i' \geqslant 2$. In the second use of this property, $(i+2)(|m|_p^{(2)} + 1)$ is compared to $(r-1)\log_p(m)$. For almost all m we know that the former is smaller than the latter. The remaining values of m are covered by the largeness of n and stepping from $p^{\frac{1}{2}n^4}$ to p^{n^4}. Note that this is the only time we use the premise $r > 1$. □

Finally, the time has come to bring in the crop.

Theorem 7.35. *For $r \in \mathbb{R}$ we put $f_r(n,m) := n + r\log_2(m)$.*

i. *Let Σ be a signature with $S := \operatorname{card}(\Sigma) + 1 \geqslant 2$. If $r < \log_S(2)$ and $\mathcal{R}$ is a noncycling TRS over Σ, then $K^{\mathcal{R}}_{f_r}$ is bounded by an exponential function.*
ii. *For any $p > 1$ and $k > 0$ there exists a totally terminating TRS $\mathcal{R}$ over a signature Σ with $\operatorname{card}(\Sigma) = p + 8 + \frac{1}{2}k(k+5)$ such that $K^{\mathcal{R}}_{f_r}$ eventually dominates all $<\!\Delta_k$-recursive functions as soon as $r > \log_p(2)$.*

Proof. Lemma 7.7 and (i) coincide.

The $\mathcal{R}$ of (ii) is our $\mathcal{R}'_p$ associated with $p > 1$ and $k > 0$. From Proposition 7.31 we know $\mathcal{R}$ is totally terminating, while Proposition 7.34 shows $K^{\mathcal{R}}_{f_r}$ eventually dominates all $<\!\Delta_k$-recursive functions if $r > \log_p(2)$ (this relies on $\log_p(m) = \log_p(2) \cdot \log_2(m)$). By Definition 7.30, the signature of $\mathcal{R}$ is the union of the disjoint signatures Σ and Σ_p. According to Definition 7.10, we have $\operatorname{card}(\Sigma_p) = p + 2$, and the Definitions 5.33 and 5.40 yield $\operatorname{card}(\Sigma) = 8 + 2k + \frac{1}{2}k(k+1)$. As the symbols S and $\circ$ have identical interpreting functions, we can simply replace $\circ$ with S. The same is possible for $[]$ and $\triangle$, hence we arrive at a total of $p + 8 + \frac{1}{2}k(k+5)$ symbols. □

With growing p and fixed k the bounds $\log_S(2)$ and $\log_p(2)$ of the two parts of the Theorem get arbitrarily close. We wonder what happens for r between the two bounds. Maybe there is, similar to the results of Weiermann (2000) mentioned in Theorem 7.3, one distinct r that separates exponential growth from enormous growth, maybe there is even an increasing sequence of such r, each a border between two different rates of growth.

In (ii), for $k = 1$ the number of symbols in Σ can be lowered a bit. Since the rules ($\mathrm{H}_{ij}6$) and (H_i7) introducing Q_{ij} and R_i need a P which is at least ternary, these symbols can be dropped. This leads to a signature with $p + 9$ symbols and a very close bound: at most exponential growth for $r < \log_{p+10}(2)$ in contrast to growth beyond $<\!\Delta_1$-recursion for $r > \log_p(2)$.

The bounds can get even closer. All ordinals below ε_0 are encoded by those standard terms whose occurrences of P are all of the shape $\mathsf{P}(\mathsf{0}, t)$, encoding $\omega^{\operatorname{val}(t)}$. Thus we do not need J_i, leaving us with $p + 8$ symbols and a function which grows too fast to be $<\!\varepsilon_0$-recursive and hence is not provably total in

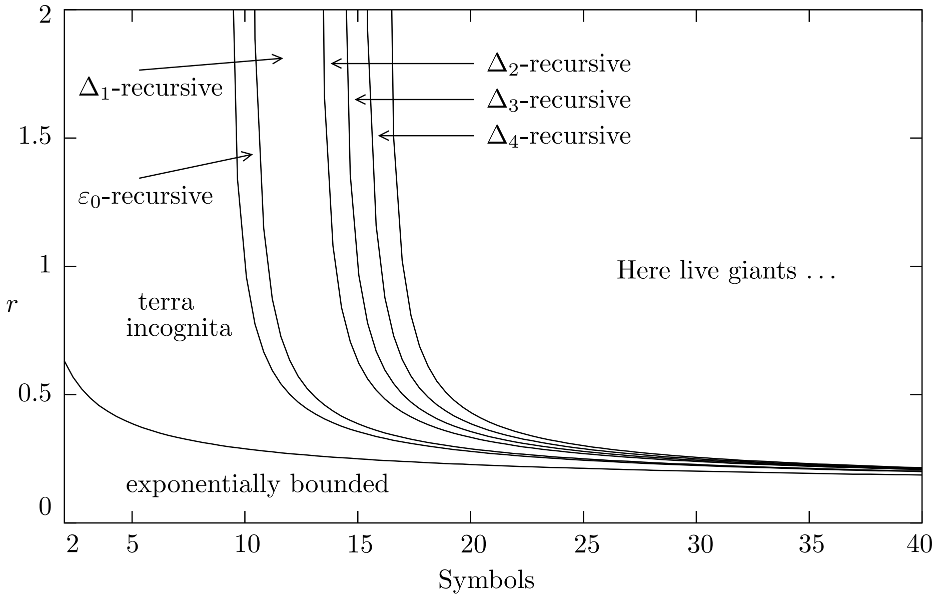

Figure 7.1: *Maximal known growth rates of $K_f^{\mathcal{R}}$ for $f(m,n) = n + r\log_2(m)$ depending on the number of symbols in the signature and on r.*

Peano arithmetic. It should also be possible here to combine the symbols $+$ and P into one binary symbol representing $\omega^x + y$.

Even for arbitrary $k > 0$ we can reach smaller signatures. Instead of considering k symbols R_i, we focus on one $(2k+1)$-ary symbol R and rules like

$$\bullet\mathsf{P}(x_1,\dots,x_i,\bar{0},\mathsf{S}y) \to \mathsf{R}(x_1,\dots,\bullet x_i,\bar{0},x_1,\dots,x_i,\bar{0},y)$$

and

$$\mathsf{R}(x_1,\dots,\circ x_i,\bar{0},y_1,\dots,y_i,\bar{0},z) \to \circ\mathsf{P}(\bar{x},\bar{0},\mathsf{P}(\bar{y},\bar{0},z))\ .$$

The interpreting function is

$$[\mathsf{R}](\bar{p},\bar{q},r) := \psi(\max\{\alpha_1,\beta_1\},\dots,\max\{\alpha_k,\beta_k\},\gamma+1)$$

with $p_i = (\alpha_i,\dots)$, $q_i = (\beta_i,\dots)$, and $r = (\gamma,\dots)$. In very much the same way we can replace the $\frac{1}{2}k(k+1)$ many Q_{ij} with one Q. Unfortunately there appears to be no way to get rid of the J_i, so we arrive at $p+10+k$ symbols. Of course, the new rules and symbols afford adapted versions of Proposition 7.28 and its relatives. We hope to prevent complete boredom by omitting these. In Figure 7.1 we try to depict the different regions of growth we met depending on the number of symbols and on r.

It may be instructive to sketch how similar results for SRSs can be achieved. In Theorem 5.23 we presented a result of Touzet (1999), stating that there is a hierarchy $(\mathcal{A}_k)_{k<\omega}$ of SRSs such that any multiple recursive function is eventually dominated by the complexities of almost all $\mathcal{A}_k$. Each $\mathcal{A}_k$ simulates all Hydra battles below ω^{ω^k}. Since the transition to a new battle configuration in $\mathcal{A}_k$ is done by a rewrite step at the root symbol, it is possible to combine (adapted versions of) our SRSs $\mathcal{R}_p$ with the $\mathcal{A}_k$. We do not go into more detail here, because the whole proof of Theorem 5.23 had to be repeated.

8 Conclusion

Your own efforts promise
a lot of success.

The time has come to summarize the results presented in the preceding text and to make some concluding remarks. Our main focus has been on complexities and order types. We have filled some gaps that had been left open in the big picture. The new picture of the world of simplification orders is presented in Table 8.1 on the following page. As before, SO is an abbreviation for the class of simplification orders. Since all bounds are essentially optimal, we get a comprehensive picture. The upper complexity bound for KBO and the lower one for SO are new results, just as the classification of the order types occuring within KBO and its subsets. All results indicate the importance of the Hardy function principle, introduced by Touzet (1999). It states that the complexity of a TRS terminating via some simplification order is bounded by a Hardy function with index closely related to the order type of the simplification order. We got a valid instance of this principle from Buchholz et al. (1994). The announced bounds are often not optimal, though. This leads us to the first problem.

Problem 1. Find and prove strong instances of the Hardy function principle.

The upper complexity bounds are, though essentially optimal, not that useful in practice. Furthermore, we have only considered simple termination, although many natural TRSs are not simplifying. Hofbauer (2001) has recently proposed a promising new approach which sometimes directly yields optimal bounds and which is even applicable to TRSs which are terminating but not simply terminating.

Concerning the shortest complexity – the function measuring the best-case behavior of a TRS – we have seen that for KBO, MPO, and LPO the usual complexity bounds are transferable. The result concerning KBO is new. In the general case, the situation is unclear. I strongly conjecture that here the $<\vartheta(\Omega^\omega)$-recursive bound is optimal, too.

Problem 2. What shortest complexities occur within simple termination?

class	bound in	bound in $\textsc{Drec}(\cdot)$	order types
MPO(1)	$\textsc{Prec}$	$<\omega^\omega$	$<\omega_3$
MPO	$\textsc{Prec}$	$<\omega^\omega$	$<\vartheta(\Omega \cdot \omega)$
KBO$^-$	$2^{O(n)}$	ω	ω
KBO(1)	$2^{O(n)}$	ω	$\leqslant\omega^\omega$
KBO	$\mathrm{Ack}(O(n), 0)$	ω^ω	$\leqslant\omega^\omega$
LPO	$\textsc{Mrec}$	$<\omega_3$	$<\vartheta(\Omega^\omega)$
SO(1)	$\textsc{Mrec}$	$<\omega_3$	$<\omega_3$
SO	$\textsc{Drec}(<\vartheta(\Omega^\omega))$	$<\vartheta(\Omega^\omega)$	$<\vartheta(\Omega^\omega)$

Table 8.1: *Essentially optimal bounds on size complexities and order types occuring within classes of simplification orders and their restrictions to strings.*

In Section 5.5 we touched upon ground versions of the three standard syntactic simplification orders and met, as a new result, a TRS which terminates via ground KBO and does not correspond to any TRS terminating via KBO. No optimal complexity bounds are known for the ground versions. I conjecture the bounds are just the same as those for the open versions.

Problem 3. What are the optimal complexity bounds for the ground versions of KBO, MPO, and LPO?

Termination via either MPO or LPO implies ω-termination, whereas termination via KBO is not contained in $<\omega^2$-termination. One of our new results states that termination via KBO is a part of ω^2-termination. Thus the three standard simplification orders are located at the very beginning of the α-termination hierarchy, and this leaves us with the task of finding stronger orders.

Problem 4. Construct syntactic simplification orders which reach beyond ω^2-termination.

The next problem is also concerned with α-termination. It is the natural extension of old questions to the termination hierarchy.

Problem 5. What complexity bounds are imposed by α-termination?

Questions like this have not yet been considered. We only know that the complexities occuring within ω-termination exhaust the multiple recursive functions, but it is already unknown if this result is optimal.

Let us turn to computability considerations. The central results here are collected in Table 8.2 on the next page. Both two entries concerning KBO are new. We see that it is usually a good idea to use two distinct successor symbols. For the general case I conjecture the functions computable via simple termination

$\mathcal{M}$	COMP($\mathcal{M}$)	COMP$_1$($\mathcal{M}$)
PT	E$_2$TIME	$\subseteq$ (PTIME $\cup\, O(n)$)
KBO	ATIME	PREC
MPO	PREC	PREC
LPO	MREC	MREC

Table 8.2: *The sets of functions computable via standard termination proof methods $\mathcal{M}$ using two respectively one successor symbol(s).*

coincide with the $<\vartheta(\Omega^{\omega})$-recursive functions, hence, as we have shown, with the functions nondeterministically computable via simple termination.

Problem 6. What functions are computable via simple termination?

In Chapter 7 we considered the length of maximal f_r-controlled derivations, depending on $r \in \mathbb{R}$ and the size of the signature, with $f_r(n, m) = n + r \log_2(m)$. We found out that if the signature contains $S - 1$ symbols and if $r < \log_S(2)$, then the lengths of f_r-controlled derivations (as a function of n) are bounded by exponential functions. On the other hand, if the signature contains $p + 10 + k$ symbols (with $p > 1$) and if $r > \log_p(2)$, then lengths of f_r-controlled derivations may leave the $<\Delta_k$-recursive functions. I feel these things should be investigated further, and maybe the amount of required rules should also be taken into consideration.

Problem 7. Consider triples $(n, m, r) \in \mathbb{N} \times \mathbb{N} \times \mathbb{R}$. Over signatures containing n symbols, what maximal f_r-controlled derivation lengths are possible for TRSs containing m rules?

These open problems show there is still a lot to do ...

Bibliography

ACKERMANN, W., 1928. Zum Hilbertschen Aufbau der reellen Zahlen. *Mathematische Annalen*, 99:118–133. → 29

ARAI, T., 1998. Some results on cut-elimination, provable well-orderings, induction and reflection. *Annals of Pure and Applied Logic*, 95(1–3):93–184. → 84

BAADER, F. and T. NIPKOW, 1998. *Term Rewriting and All That*. Cambridge University Press, Cambridge. → 43, 64, 66

BACHMANN, H., 1950. Die Normalfunktionen und das Problem der ausgezeichneten Folgen von Ordnungszahlen. *Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich*, 95:115–147. → 25

——, 1967. *Transfinite Zahlen*. Springer-Verlag, Berlin. → 33, 34

BONFANTE, G., 2000. Complexity characterisations of restrictions of KBO. Preprint. → 120, 132

BONFANTE, G., A. CICHON, J.-Y. MARION, and H. TOUZET, 1999. Complexity classes and rewrite systems with polynomial interpretation. In *Computer Science Logic*, vol. 1584 of *Lecture Notes in Computer Science*, pp. 372–384. Springer-Verlag, Berlin. → 7, 70, 118, 132, 134

——, 2001. Algorithms with polynomial interpretation termination proof. *Journal of Functional Programming*, 11(1):33–53. → 8, 132, 133

BÖRGER, E., 1989. *Computability, Complexity, Logic*, vol. 128 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, Amsterdam. → 39

BUCHHOLZ, W., 1980. Fundamentalfolgen für $\bar{\theta}(\nu)$. Handwritten notes in German. → 98

——, 1993. Kruskals Theorem impliziert $WF(\vartheta(\Omega^\omega)$. Notes in German. → 96

——, 1995. Proof-theoretic analysis of termination proofs. *Annals of Pure and Applied Logic*, 75:57–65. → 84

BUCHHOLZ, W., E. A. CICHON, and A. WEIERMANN, 1994. A uniform approach to fundamental sequences and hierarchies. *Mathematical Logic Quarterly*, 40:273–286. → 6, 32, 34, 36, 37, 81, 112, 163

CARON, A.-C., 1991. Linear bounded automata and rewrite systems: Influence of initial configuration on decision properties. In *Proceedings of the*

Colloquium on Trees in Algebra and Programming, vol. 493 of *Lecture Notes in Computer Science*, pp. 75–89. Springer-Verlag, Berlin. → 58

Cichon, E. A., 1992. Termination orderings and complexity characterisations. In P. Aczel, H. Simmons, and S. S. Wainer, eds., *Proof Theory*, pp. 171–193. Cambridge University Press, Cambridge. → 6, 80, 93

Cichon, E. A. and P. Lescanne, 1992. Polynomial interpretations and the complexity of algorithms. In *Automated Deduction—CADE-11*, vol. 607 of *Lecture Notes in Artificial Intelligence*, pp. 139–147. Springer-Verlag, Berlin. → 7, 117, 118

Cichon, E. A. and E. Tahhan Bittar, 1998. Ordinal recursive bounds for Higman's theorem. *Theoretical Computer Science*, 201(1–2):63–84. → 4, 80, 81, 93

Cichon, E. A. and S. S. Wainer, 1983. The slow-growing and the Grzegorczyk hierarchies. *Journal of Symbolic Logic*, 48(2):399–408. → 37

Dauchet, M., 1989. Simulation of Turing machines by a left-linear rewrite rule. In Dershowitz (1989), pp. 109–120. → 168

——, 1992. Simulation of Turing machines by a regular rewrite rule. *Theoretical Computer Science*, 103(2):409–420. A preliminary version appeared as Dauchet (1989). → 47, 58

Davis, M., 1958. *Computability and Unsolvability*. McGraw Hill, New York. → 38

Dershowitz, N., 1979. A note on simplification orderings. *Information Processing Letters*, 9(5):212–215. → 42, 48, 49, 50

——, 1982. Orderings for term-rewriting systems. *Theoretical Computer Science*, 17:279–301. → 50, 63

——, 1987a. Corrigendum to termination of rewriting. *Journal of Symbolic Computation*, 4:409–410. → 168

——, 1987b. Termination of rewriting. *Journal of Symbolic Computation*, 3(1–2):69–115. See also Dershowitz (1987a). → 47, 65

Dershowitz, N., ed., 1989. *Rewriting Techniques and Applications*, vol. 355 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin. → 168, 170

Dershowitz, N. and J.-P. Jouannaud, 1990. Rewrite systems. In J. van Leeuwen, ed., *Handbook of Theoretical Computer Science*, vol. B, pp. 243–320. Elsevier. → 43

Dershowitz, N., J.-P. Jouannaud, and J. W. Klop, 1995. Problems in rewriting III. In J. Hsiang, ed., *Rewriting Techniques and Applications*, vol. 914 of *Lecture Notes in Computer Science*, pp. 457–471. Springer-Verlag, Berlin. → 5, 81

Dershowitz, N. and Z. Manna, 1979. Proving termination with multiset orderings. *Communications of the ACM*, 22(8):465–476. → 24

Dershowitz, N. and M. Okada, 1988. Proof-theoretic techniques for term rewriting theory. In *Third Annual Symposium on Logic in Computer Science*, pp. 104–111. IEEE Computer Society. → 74

Dershowitz, N. and R. Treinen, 1998. An on-line problem database. In T. Nipkow, ed., *Rewriting Techniques and Applications*, vol. 1379 of *Lecture Notes in Computer Science*, pp. 332–342. Springer-Verlag, Berlin. The list is available at `http://www.lri.fr/~rtaloop`. → 5, 81

de Jongh, D. H. J. and R. J. Parikh, 1977. Well-partial orderings and hierarchies. *Indagationes Mathematicae*, 39(3):195–206. → 6, 73, 78

Dick, J., J. Kalmus, and U. Martin, 1990. Automating the Knuth Bendix ordering. *Acta Informatica*, 28(2):95–119. → 66, 67

Ferreira, M. d. C. F., 1995. *Termination of term rewriting: Well-foundedness, totality and transformations*. Ph.D. thesis, Universiteit Utrecht. `http://www.library.uu.nl/digiarchief/dip/diss/01847318/inhoud.htm`. → 4, 25, 43, 48, 59, 60, 61, 63, 67

Ferreira, M. d. C. F. and H. Zantema, 1993. Total termination of term rewriting. In Kirchner (1993), pp. 213–227. → 43, 169

——, 1996. Total termination of term rewriting. *Applicable Algebra in Engineering, Communication and Computing*, 7(2):133–162. A preliminary version appeared as Ferreira and Zantema (1993). → 59

Fraïssé, R., 1986. *Theory of Relations*, vol. 118 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, Amsterdam. → 26

Friedman, H., 1999. FOM: 55:Term rewriting/proof theory. `http://www.math.psu.edu/simpson/fom/postings/9908/msg00066.html`. Mailing list entry from August 27th. An improved and corrected version of this is the unpublished preprint "Term Rewriting and Computational Complexity". → 10, 113, 114

——, 2000. FOM: Urbana thoughts. `http://www.math.psu.edu/simpson/fom/postings/0006/msg00041.html`. Mailing list entry from June 10th. → 1

Friedman, H. and M. Sheard, 1995. Elementary descent recursion and proof theory. *Annals of Pure and Applied Logic*, 71(1):1–45. → 31, 32, 33, 37

Gallier, J. H., 1991. What's so special about Kruskal's theorem and the ordinal Γ_0? A survey of some results in proof theory. *Annals of Pure and Applied Logic*, 53(3):199–260. → 5, 96

Geser, A., 1997. Omega-termination is undecidable for totally terminating term rewriting systems. *Journal of Symbolic Computation*, 23(4):399–411. → 58

Geupel, O., 1988. *Terminationsbeweise bei Termersetzungssystemen*. Diplomarbeit, Technische Universität Dresden. → 3, 82

Girard, J.-Y., 1981. Π^1_2-logic I. Dilators. *Annals of Mathematical Logic*, 21:75–219. → 37

GRÄDEL, E. and Y. GUREVICH, 1995. Tailoring recursion for complexity. *Journal of Symbolic Logic*, 60(3):952–969. → 8, 133

GRZEGORCZYK, A., 1953. Some classes of recursive functions. *Rozprawy Matematyczne*, 4. → 31, 37, 83

HANDLEY, W. G. and S. S. WAINER, 1994. Equational derivation vs. computation. *Annals of Pure and Applied Logic*, 70(1):17–49. → 39, 70, 71

HARDY, G. H., 1904. A theorem concerning the infinite cardinal numbers. *Quarterly Journal of Mathematics*, 35:87–94. → 34

HASEGAWA, R., 1994. Well-ordering of algebras and Kruskal's theorem. In N. D. Jones, M. Hagiya, and M. Sato, eds., *Logic, Language and Computation*, vol. 792 of *Lecture Notes in Computer Science*, pp. 133–172. Springer-Verlag, Berlin. → 74, 78

HIGMAN, G., 1952. Ordering by divisibility in abstract algebras. *Proceedings of the London Mathematical Society, 3rd series*, 2:326–336. → 27

HOFBAUER, D., 1991. *Termination Proofs and Derivation Lengths in Term Rewriting Systems*. Ph.D. thesis, Technische Universität Berlin. → 3, 6, 7, 8, 43, 48, 57, 75, 79, 80, 82, 83, 84, 85, 93, 119, 124, 125, 134

——, 1992. Termination proofs by multiset path orderings imply primitive recursive derivation lengths. *Theoretical Computer Science*, 105(1):129–140. → 3, 7, 79, 83, 119, 134

——, 2000. *An Upper Bound on the Derivational Complexity of Knuth-Bendix Orderings*. Tech. Rep. 28/00, Mathematische Schriften Kassel, Fachbereich Mathematik/Informatik, Universität Gesamthochschule Kassel. To appear in "Information and Computation". → 3, 75, 80, 84, 116

——, 2001. Termination proofs by context-dependent interpretations. In Middeldorp (2001), pp. 108–121. → 163

HOFBAUER, D. and C. LAUTEMANN, 1989. Termination proofs and the length of derivations. In Dershowitz (1989), pp. 167–177. → 3, 8, 48, 80, 81, 83, 84, 94, 122, 124

HUET, G. and D. S. LANKFORD, 1978. *On the Uniform Halting Problem for Term Rewriting Systems*. Report 283, Institut de Recherche en Informatique et en Automatique, Le Chesnay, France. → 47, 71

HUET, G. and D. C. OPPEN, 1980. Equations and rewrite rules: A survey. In R. Book, ed., *Formal Language Theory: Perspectives and Open Problems*, pp. 349–405. Academic Press, New York. → 117

JECH, T. J., 1978. *Set Theory*. No. 79 in Pure and Applied Mathematics. Academic Press, New York. → 15

KALMÁR, L., 1943. Egyzzerü példa eldönthetetlen aritmetikai problémára. *Matematikai és Fizikai Lapok*, 50:1–23. → 28

KAMIN, S. and J.-J. LÉVY, 1980. Attempts for generalizing the recursive path orderings. Unpublished manuscript. → 1, 42, 63, 64

Kaplan, S., 1987. Simplifying conditional term rewriting systems: Unification, termination and confluence. *Journal of Symbolic Computation*, 4(3):295–334. → 50

Kirby, L. and J. Paris, 1982. Accessible independence results for Peano arithmetic. *Bulletin of the London Mathematical Society*, 14:285–293. → 5, 94, 98

Kirchner, C., ed., 1993. *Rewriting Techniques and Applications*, vol. 690 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin. → 169, 172

Kleene, S. C., 1936. General recursive functions of natural numbers. *Mathematische Annalen*, 112:727–742. → 40

Knuth, D. E. and P. B. Bendix, 1970. Simple word problems in universal algebras. In J. Leech, ed., *Computational Problems in Abstract Algebra*, pp. 263–297. Pergamon Press, Oxford. → 1, 41, 42, 64, 66

König, D., 1927. Über eine Schlussweise aus dem Endlichen ins Unendliche. *Acta litteraria Academiae Scientiarum Hungaricae*, 3:121–130. → 27, 136

Korovin, K. and A. Voronkov, 2000. A decision procedure for the existential theory of term algebras with the Knuth-Bendix ordering. In *Proceedings of the 15th Annual IEEE Symposium on Logic in Computer Science, LICS'00*, pp. 291–304. IEEE Computer Society, Washington. → 115, 116

——, 2001. Verifying orientability of rewrite rules using the Knuth-Bendix order. In Middeldorp (2001), pp. 137–153. → 64, 67, 115, 116

Kreisel, G., 1952. On the interpretation of non-finitist proofs. II. Interpretation of number theory. Applications. *Journal of Symbolic Logic*, 17:43–58. → 32

Krentel, M. W., 1988. The complexity of optimization problems. *Journal of Computer and System Sciences*, 36(3):490–509. → 8, 133

Kruskal, J. B., 1960. Well-quasi-orderings, the tree theorem, and Vázsonyi's conjecture. *Transactions of the American Mathematical Society*, 95:210–225. → 27, 42, 48, 49, 50, 135, 136

Kurihara, M. and A. Ohuchi, 1990. Modularity of simple termination of term rewriting systems. *Journal of Information Processing*, 31(5):633–642. → 57

Lankford, D. S., 1975. *Canonical Algebraic Simplification in Computational Logic*. Memo ATP-25, Automatic Theorem Proving Project, University of Texas, Austin. → 54

——, 1979. *On Proving Term Rewriting Systems are Noetherian*. Memo MTP-3, Mathematics Department, Louisiana Tech. University, Ruston, LA. Revised October 1979. → 65

Lautemann, C., 1988. A note on polynomial interpretation. *Bulletin of the European Association for Theoretical Computer Science*, 36:129–131. → 3, 82

LEPPER, I., 1999. Simply terminating rewrite systems with long derivations. Submitted to "Archive for Mathematical Logic". → 11

——, 2000a. The computational strength of the Knuth–Bendix order. Preliminary, available via http://wwwmath.uni-muenster.de/logik/publ/pre/7.html. → 11

——, 2000b. Derivation lengths and order types of Knuth–Bendix orders. To appear in "Theoretical Computer Science", see also http://wwwmath.uni-muenster.de/logik/publ/pre/5.html. → 11

LESCANNE, P., 1994. On termination of one rule rewrite systems. *Theoretical Computer Science*, 132(1–2):395–401. → 47

LÖB, M. H. and S. S. WAINER, 1970. Hierarchies of number-theoretic functions I. *Archiv für Mathematische Logik*, 13:39–51. → 34

LOEBL, M. and J. MATOUŠEK, 1987. On undecidability of the weakened Kruskal theorem. In S. G. Simpson, ed., *Logic and combinatorics*, vol. 65 of *Contemporary Mathematics*, pp. 275–280. Transactions of the American Mathematical Society, Providence, RI. → 9, 136, 138

MIDDELDORP, A., ed., 2001. *Rewriting Techniques and Applications*, vol. 2051 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin. → 170, 171

MIDDELDORP, A. and B. GRAMLICH, 1993. Simple termination is difficult. In Kirchner (1993), pp. 228–242. → 93, 172

——, 1995. Simple termination is difficult. *Applicable Algebra in Engineering, Communication and Computing*, 6(2):115–128. A preliminary version appeared as Middeldorp and Gramlich (1993). → 58

MIDDELDORP, A. and H. ZANTEMA, 1997. Simple termination of rewrite systems. *Theoretical Computer Science*, 175(1):127–158. → 26, 27, 48, 50, 64, 67

MONK, J. D., 1976. *Mathematical Logic*, vol. 37 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin. → 28

NASH-WILLIAMS, C. S. J. A., 1963. On well-quasi-ordering finite trees. *Proceedings of the Cambridge Philosophical Society*, 59(4):833–835. → 27

ODLYZKO, A. M., 1995. Asymptotic enumeration methods. In R. L. Graham, M. Grötschel, and L. Lovász, eds., *Handbook of Combinatorics*, vol. 2, pp. 1063–1229. Elsevier, Amsterdam. → 136

OTTER, R., 1948. The number of trees. *Annals of Mathematics*, 49:583–599. → 136

PÉTER, R., 1935. Konstruktion nichtrekursiver Funktionen. *Mathematische Annalen*, 111:42–60. → 29

——, 1936. Über die mehrfache Rekursion. *Mathematische Annalen*, 113:489–527. → 30

——, 1951. *Rekursive Funktionen*. Akadémiai Kiadó, Budapest. See also the English translation Péter (1967). → 30

——, 1967. *Recursive Functions*. Academic Press, New York. → 172

Plaisted, D. A., 1978. *A Recursively Defined Ordering for Proving Termination of Term Rewriting Systems*. Report R-78-943, Department of Computer Science, University of Illinois, Urbana, IL. → 1, 42, 62, 63, 119

Pohlers, W., 1989. *Proof Theory*. No. 1407 in Lecture Notes in Mathematics. Springer-Verlag, Berlin. → 15, 25

Post, E. L., 1947. Recursive unsolvability of a problem by Thue. *Journal of Symbolic Logic*, 12(1):1–11. → 58

Raney, G. N., 1960. Functional composition patterns and power series reversion. *Transactions of the American Mathematical Society*, 94:441–451. → 44

Rathjen, M. and A. Weiermann, 1993. Proof-theoretic investigations on Kruskal's theorem. *Annals of Pure and Applied Logic*, 60(1):49–88. → 4, 25, 26, 93

Ritchie, R. W., 1965. Classes of recursive functions based on Ackermann's function. *Pacific Journal of Mathematics*, 15:1027–1044. → 29

Robbin, J. W., 1965. *Subrecursive Hierarchies*. Ph.D. thesis, Princeton University, Princeton. → 33, 34, 37

Rose, H. E., 1984. *Subrecursion: Functions and Hierarchies*. Clarendon Press, Oxford. → 30, 31, 34

Sakai, K., 1984. *Knuth-Bendix algorithm for Thue system based on Kachinuki ordering*. ICOT Technical Memorandum TM-0087, ICOT, Institute for New Generation Computer Technology, Tokyo, Japan. → 74

Schmidt, D., 1976a. Built-up systems of fundamental sequences and hierarchies of number-theoretic functions. *Archiv für Mathematische Logik*, 18:47–53. → 34

——, 1976b. Postscript to "Built-up systems of fundamental sequences and hierarchies of number-theoretic functions". *Archiv für Mathematische Logik*, 18:145–146. → 34

——, 1979. *Well-Partial Orderings and their Maximal Order Types*. Habilitationsschrift, Fakultät für Mathematik der Ruprecht-Karl-Universität Heidelberg. → 5, 6, 73, 78, 96

Schütte, K., 1977. *Proof Theory*. Springer-Verlag, Berlin. → 25

Schwichtenberg, H., 1971. Eine Klassifikation der ε_0-rekursiven Funktionen. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 17:61–74. → 34

Shepherdson, J. C. and H. E. Sturgis, 1963. Computability of recursive functions. *Journal of the Association for Computing Machinery*, 10:217–255. → 38

Simpson, S. G., 1985. Non-provability of certain combinatorial properties of finite trees. In L. A. Harrington, M. D. Morley, A. Scedrov, and S. G. Simpson, eds., *Harvey Friedman's Research on the Foundations of Mathematics*, vol.

117 of *Studies in Logic and the Foundations of Mathematics*, pp. 87–117. North-Holland, Amsterdam. → 9, 135, 136

Skolem, T. A., 1923. Begründung der elementaren Arithmetik durch die rekurrierende Denkweise ohne Anwendung scheinbarer Veränderlichen mit unendlichem Ausdehnungsbereich. *Videnskapsselskapets Skrifter, Matematisk-Naturvidenskapelig Klasse*, 6:1–38. English translation in van Heijenoort (1967). → 28

Touzet, H., 1997. *Propriétés combinatoires pour la terminaison de systèmes de réécriture*. Ph.D. thesis, Université Henri Poincaré, Nancy. `http://www.lifl.fr/~touzet/Publications/publi.html`. → 3, 4, 6, 75, 77, 81, 83, 85, 94

——, 1998a. A complex example of a simplifying rewrite system. *Lecture Notes in Computer Science*, 1443:507–517. → 94

——, 1998b. Encoding the Hydra battle as a rewrite system. In L. Brim, J. Gruska, and J. Zlatuška, eds., *Mathematical Foundations of Computer Science*, vol. 1450 of *Lecture Notes in Computer Science*, pp. 267–276. Springer-Verlag, Berlin. → 4, 5, 6, 10, 43, 60, 61, 81, 94, 95, 103, 108

——, 1999. A characterization of multiply recursive functions with Higman's lemma. In P. Narendran and M. Rusinowitch, eds., *Rewriting Techniques and Applications*, vol. 1631 of *Lecture Notes in Computer Science*, pp. 163–174. Springer-Verlag, Berlin. → 4, 6, 81, 94, 95, 107, 108, 112, 161, 163

Turing, A. M., 1936. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society, 2nd series*, 42:230–265. See also Turing (1937). → 38

——, 1937. On Computable Numbers, with an Application to the Entscheidungsproblem. A correction. *Proceedings of the London Mathematical Society, 2nd series*, 43:544–546. → 174

van Heijenoort, J., 1967. *From Frege to Gödel*. Harvard University Press, Cambridge. → 174

Veblen, O., 1908. Continuous increasing functions of finite and transfinite ordinals. *Transactions of the American Mathematical Society*, 9:280–292. → 4, 25, 81, 95

Wainer, S. S., 1970. A classification of the ordinal recursive functions. *Archiv für Mathematische Logik*, 13:136–153. → 34

——, 1972. Ordinal recursion, and a refinement of the extended Grzegorczyk hierarchy. *Journal of Symbolic Logic*, 37(2):281–292. → 34, 37

Weiermann, A., 1992. Proving termination for term rewriting systems. In E. Börger, G. Jäger, H. K. Büning, and M. M. Richter, eds., *Computer Science Logic. 5th Workshop, CSL'91*, vol. 626 of *Lecture Notes in Computer Science*, pp. 419–428. Springer-Verlag, Berlin. → 25, 74

——, 1994. Complexity bounds for some finite forms of Kruskal's theorem. *Journal of Symbolic Computation*, 18(5):463–488. → 4, 5, 25, 80, 81, 93, 95, 113

——, 1995. Termination proofs for term rewriting systems by lexicographic path orderings imply multiply recursive derivation lengths. *Theoretical Computer Science*, 139(1–2):355–362. → 3, 7, 79, 80, 84, 85, 120, 134

——, 1997a. Ordinalzahlbezeichnungssysteme und subrekursive Hierarchien. Handwritten notes in German. → 98

——, 1997b. Sometimes slow growing is fast growing. *Annals of Pure and Applied Logic*, 90(1–3):91–99. → 34

——, 2000. An application of graphical enumeration to PA. Submitted to "Journal of Symbolic Logic". → 9, 136, 160

ZANTEMA, H., 1992. *Termination of Term Rewriting by Interpretation*. Tech. Rep. RUU-CS-92-14, Department of Computer Science, Utrecht University. → 3, 59, 175

——, 1993. Termination of term rewriting by interpretation. In M. Rusinowitch and J. Rémy, eds., *Conditional Term Rewriting Systems, Third International Workshop, CTRS-92*, vol. 656 of *Lecture Notes in Computer Science*, pp. 155–167. Springer-Verlag, Berlin. Full version appeared as Zantema (1992). → 2, 10, 42, 51, 56, 57

——, 1994. Termination of term rewriting: Interpretation and type elimination. *Journal of Symbolic Computation*, 17(1):23–50. → 2, 10, 42, 51, 54, 56, 57, 59, 62

——, 1995. Total termination of term rewriting is undecidable. *Journal of Symbolic Computation*, 20(1):43–60. → 58

——, 1999. *The Termination Hierarchy for Term Rewriting*. Tech. Rep. UU-CS-1999-22, Department of Computer Science, Utrecht University. http://www.cs.uu.nl/docs/research/publication/TechList2.html. Report version of Zantema (2001). → 10, 48, 54, 55, 56, 57, 58, 59, 61

——, 2001. The termination hierarchy for term rewriting. *Applicable Algebra in Engineering, Communication and Computing*, 12(1–2):3–19. → 10, 48, 54, 56, 61, 175

ZEMKE, F., 1977. P.R.-regulated systems of notation and the subrecursive hierarchy equivalence property. *Transactions of the American Mathematical Society*, 234:89–118. → 37

Glossary of Notation

$\bar{s}$	an abbreviation for $s_1, \dots, s_n$, 13
$\bar{0}$	an abbreviation for $0, \dots, 0$, 13
n^k	k consecutive occurrences of n, 13
$A \subseteq B$	A is a subset of B, 14
$A \subsetneq B$	A is a subset of $B \neq A$, 14
$A \setminus B$	set theoretic difference: $\{x \in A : x \notin B\}$, 14
$\mathbb{N}$	$\{0, 1, 2, \dots\}$, the nonnegative integers, 14
$\mathbb{R}$	the reals, 14
$\mathbb{N}^+$	$\{1, 2, \dots\}$, the positive integers, 14
$\mathbb{R}^+$	the positive reals, 14
$\mathbb{R}_0^+$	the nonnegative reals, 14
$[n, m]$	the set $\{n, n+1, \dots, m\}$, 14
$\mathrm{card}(W)$	the cardinality of W, 14
$(a_1, \dots, a_n)$	the (finite ordered) tuple of $a_1, \dots, a_n$, 14
$\lvert\cdot\rvert$	the length of a tuple, 14
$\times$	the Cartesian product, 14
A^n	the n-fold product of A, 14
A^*	the set of (finite) tuples over A, 14
$\uplus$	the disjoint union, 14
$f \colon X \to Y$	f is a function from X to Y, 14
${}^X Y$	the set of functions from X to Y, 14
$g \circ f$	the function which maps a to $g(f(a))$, 14
$f^n(a)$	the n^{th} iteration of f, applied to a, 14
$g(\cdot, b)^n(c)$	$h^n(c)$ for the function h mapping a to $g(a, b)$, 14
$n!$	the factorial function, 14
$n \dot{-} m$	$\max\{n - m, 0\}$, 15
$\lfloor r \rfloor$	the unique $n \in \mathbb{N}$ with $n \leqslant r < n + 1$, 15
$\lceil r \rceil$	the unique $n \in \mathbb{N}$ with $n - 1 < r \leqslant n$, 15

$\log_p(r)$	the logarithm of r with base p, 15
$\Lbag\dots\Rbag$	a multiset, 15
$M \Cup N$	multiset union, 15
$M \Subset N$	multi-subset, 15
$M \mathbin{\backslash\!\backslash} N$	multiset difference, 15
$\bar{p}\,R\,\bar{q}$	$p_i\,R\,q_j$ for all affected i and j, 15
$(P, \preccurlyeq)$	a reflexive pre- or partial order, 16
$(P, \prec)$	the strict part of a pre- or partial order, 16
$\max X$	the maximum of X, with the convention $\max\emptyset = 0$, 17
$\min X$	the minimum of X, 17
$\sup X$	the supremum of X, 17
$\inf X$	the infimum of X, 17
On	the proper class of ordinals, 17
$\beta + 1$	the ordinal successor of β, 18
ω	the least limit ordinal, 18
Lim	the class of limit ordinals, 18
Ω	the first uncountable ordinal, 18
$\mathrm{otype}(P, \prec)$	the order type of $(P, \prec)$, 18
$\mathrm{otype}^P_\prec$	the canonical embedding of a well-founded $(P, \prec)$ into $\mathrm{otype}(P, \prec)$, 18
$\mathrm{enum}^P_\prec$	the enumerating function of the well-order $(P, \prec)$, 18
$\prec^{\restriction Q}$	the restriction of $\prec$ to Q, 19
$(a_\iota)_{\iota<\alpha}$	a sequence of length α, 19
$A^{<\beta}$	the sequences over A having length below β, 19
$\alpha + \beta$	ordinal addition, 21
$\alpha \cdot \beta$	ordinal multiplication, 21
α^β	ordinal exponentiation, 21
H	the class of principal ordinals, 22
E	the epsilon numbers: λ with $\lambda = \omega^\lambda$, 22
ε_α	the α^{th} epsilon number, 22
ω_n	an ω-tower: $\omega_0 = 1$ and $\omega_{n+1} = \omega^{\omega_n}$, 22
$=_{\mathrm{NF}}$	additive normal form, 22
$=_{\mathrm{CNF}}$	Cantor normal form, 22
$\oplus$	the natural sum of ordinals, 22
$*$	the concatenation of sequences, 22
$\leqslant_{\mathsf{ext}}$	if the sequence a' extends a, then $a \leqslant_{\mathsf{ext}} a'$, 22

$\prec^{1,n}$	the concatenation of $\prec_1, \dots, \prec_n$, 23
$\prec_{\mathsf{lex}}^{1,n}$	the lexicographic product of $\prec_1, \dots, \prec_n$, 23
$\prec_{\mathsf{lex}}^{n}$	the n-fold lexicographic product of $\prec$, 23
$\prec_{\mathsf{lex}}^{*}$	the lexicographic order based on $\prec$, 23
$<_{\mathsf{lex}}$	abbreviation for $<_{\mathsf{lex}}^{n}$ (if n is obvious) and $\prec_{\mathsf{lex}}^{*}$, 23
$\prec_{\mathsf{mul}}$	the multiset extension of $(P, \prec)$, 24
$E_\Omega(\alpha)$	the countable epsilon numbers needed to represent $\alpha < \varepsilon_{\Omega+1}$, 25
$\alpha^\star$	$\max E_\Omega(\alpha)$, 25
ϑ	a very pleasant unary one-to-one function from $\varepsilon_{\Omega+1}$ to $\mathsf{E} \cap \Omega$, 25
$\vartheta(\Omega^\omega)$	the small Veblen number, 25
$\mathcal{B}(A)$	the set of (finite rooted ordered) trees with labels from A, 26
$\lvert\cdot\rvert$	the size of a tree, 26
$\prec_{\mathsf{hemb}}$	homeomorphic embedding of labeled trees, 26
$g <_{\mathrm{d}} f$	the function g is (everywhere) dominated by f, 27
$g <_{\mathrm{ed}} f$	the function g is eventually dominated by f, 27
$g \leqslant_{\mathrm{ed}} f$	the variant of $<_{\mathrm{ed}}$ using $\leqslant$, 27
$X \leqslant_{\mathrm{ed}} Y$	any member of X is eventually dominated by a member of Y, 28
$X \approx_{\mathrm{ed}} Y$	$X \leqslant_{\mathrm{ed}} Y$ and $Y \leqslant_{\mathrm{ed}} X$, 28
$\mathsf{Sub}(g, \bar{h})$	substitution of the h_i in g, 28
$\textsc{Elem}$	the elementary functions, 28
χ_P	characteristic function for the set P, 28
2_n	n times iterated exponentiation by 2, 28
$\mathsf{PRec}(g, h)$	primitive recursion based on g and h, 29
$\textsc{Prec}$	the primitive recursive functions, 29
Ack	the binary Ackermann function, 29
$\mathcal{M}_k$	the k-recursive functions, 30
$\textsc{Mrec}$	the multiple recursive functions, 30
$\mathcal{E}_n$	the n^{th} Grzegorczyk class, 31
E_m	the m^{th} Grzegorczyk function, 31
Λ	a countable limit ordinal, 31
count-h	the count function for h, 32
$\mathsf{DRec}(g, h)$	the function defined from g and h via descent recursion, 32
$\textsc{Drec}(\alpha)$	the α-recursive functions, 32
$\textsc{Drec}(<\alpha)$	the $<\alpha$-recursive functions, 32
$\alpha[n]$	the n^{th} member of the fundamental sequence associated with α, 33

$(\Lambda, \cdot[\cdot])$	a Bachmann system, 33
F_α	the α^{th} fast growing function, 34
H_α	the α^{th} Hardy function, 34
L_α	the α^{th} counting function, 34
G_α	the α^{th} slow growing function, 34
$\mathcal{P}$	a program for a RM, 38
$(j, +, q)$	an addition instruction for a RM, 38
$(j, -, q)$	a subtraction instruction for a RM, 38
(j, q, r)	a jump instruction for a RM, 38
$(p; \bar{b})$	a configuration of a RM, 38
Ptime	the functions computable in polynomial time, 39
Etime	the functions computable in exponential time, 39
E$_2$time	the functions computable in double exponential time, 39
Linspace	the functions computable using linear space, 39
$O(f)$	the functions g with $g \leqslant_{\mathrm{ed}} cf + d$ for some c, d, 39
Atime	the functions computable with timebound in $\mathrm{Ack}(O(n), 0)$, 39
Σ	a signature, 43
$\#(f)$	the arity of f, 43
$\Sigma^{(n)}$	the members of Σ with arity n, 43
$\Sigma^{(\geqslant n)}$	the members of Σ with arity $\geqslant n$, 43
$\Sigma^{(\leqslant n)}$	the members of Σ with arity $\leqslant n$, 43
$\mathrm{Ar}(\Sigma)$	maximal arity of (the function symbols in) Σ, 43
$\mathcal{T}(\Sigma, \mathcal{X})$	the term algebra over Σ and $\mathcal{X}$, 43
$\mathcal{V}$	the variables, 43
$\mathrm{root}(s)$	the root symbol of s, 43
$\mathcal{O}(s)$	the symbols occuring in s, 43
$\mathcal{V}(s)$	the variables occurring in s, 43
$\mathcal{T}(\Sigma)$	the closed terms over Σ, 43
$\mathcal{S}(s)$	the subterms of s, 43
$<_{\mathsf{sub}}$	the (proper) subterm relation, 43
δ_u	spits out 1 iff the input equals u, otherwise 0, 43
$\lvert s \rvert_u$	the number of occurrences of u in s, 43
$\lvert s \rvert$	the size or length (number of symbols) of s, 44
$\mathrm{dp}(s)$	the depth of the term s, 44
δ_{kl}	Kronecker's δ: 1 if $k = l$, 0 else, 44

$f^n(s)$	iteration of function symbols, 45
σ	a substitution, 45
$s\sigma$	abbreviation for $\sigma(s)$, 45
(l, r)	a rewrite rule, 45
$\mathcal{R}$	a term rewriting system, 45
$\to_{\mathcal{R}}$	the rewrite relation, 45
$\xrightarrow{+}_{\mathcal{R}}$	the transitive closure of $\to_{\mathcal{R}}$, 45
$\xrightarrow{*}_{\mathcal{R}}$	the transitive and reflexive closure of $\to_{\mathcal{R}}$, 45
$\xrightarrow{\dot\triangleright}_{\mathcal{R}}$	$s \xrightarrow{\dot\triangleright}_{\mathcal{R}} t$ iff $s \to_{\mathcal{R}} t$ and t is unique, 45
$\xrightarrow{\triangleright}_{\mathcal{R}}$	the transitive closure of $\xrightarrow{\dot\triangleright}_{\mathcal{R}}$, 45
$\to_{\mathrm{A}}$	a rewrite step due to the rule (A), 45
$\mathrm{Emb}(\Sigma)$	the set of (representatives of) embedding rules, 46
ε	the empty string, 46
$\xrightarrow{*}^{\geqslant}$	$s \xrightarrow{*}^{\geqslant} t$ if $s \xrightarrow{*} t$ is bounded by $\lvert s\rvert$, 46
$\xrightarrow{*}^{\leqslant}$	$s \xrightarrow{*}^{\leqslant} t$ if $s \xrightarrow{*} t$ is bounded by $\lvert t\rvert$, 46
$\xrightarrow{*}^{=}$	$s \xrightarrow{*}^{=} t$ if there is a derivation from s to t where all terms have the same size, 46
$\mathrm{Dl}_{\mathcal{R}}$	the (depth) complexity of $\mathcal{R}$, mapping n to the maximal $\mathrm{dl}_{\mathcal{R}}(s)$ with $\mathrm{dp}(s) \leqslant n$, 47
$\mathrm{dl}_{\mathcal{R}}$	$\mathrm{dl}_{\mathcal{R}}$ maps a closed term s to the maximal length of a derivation starting with s, 47
$\mathrm{Dc}_{\mathcal{R}}$	the size complexity of $\mathcal{R}$, defined like $\mathrm{Dl}_{\mathcal{R}}$, but based on $\lvert s\rvert$, 47
$\mathrm{SDl}_{\mathcal{R}}$	the shortest (depth) complexity of $\mathcal{R}$, 48
$\mathrm{SDc}_{\mathcal{R}}$	the shortest size complexity of $\mathcal{R}$, 48
$\mathrm{sdl}_{\mathcal{R}}$	$\mathrm{sdl}_{\mathcal{R}}$ maps a closed term s to the minimal length of a derivation to normal form starting with s, 48
$\leqslant_{\mathsf{emb}}$	the embedding relation, 50
$\mathcal{I}$	an interpretation, mapping closed terms to a partial order, 52
$\prec_{\mathcal{I}}$	the partial order on $\mathcal{T}(\Sigma, \mathcal{V})$ induced by $\mathcal{I}$, 52
$(A, \prec, \mathcal{F})$	a Σ-algebra over $(A, \prec)$, 54
$[f]_A$	a function from A^n to A, where n is the arity of f, interpreting f in a Σ-algebra based on $(A, \prec)$, 54
$(\alpha, \mathcal{F})$	the Σ-algebra $(\alpha, <, \mathcal{F})$, 54
$\models$	satisfaction in first order logic, 55
$[\![\cdot, \cdot]\!]_A$	the pre-interpretation based on a Σ-algebra, 55
$\prec_A$	the partial order on $\mathcal{T}(\Sigma, \mathcal{V})$ induced by the Σ-algebra $(A, \prec, \mathcal{F})$, 55

$[\![\cdot]\!]_A$	the interpretation based on a Σ-algebra, 55
$[\![\cdot]\!]$	$[\![\cdot]\!]_A$, if $(A, \prec)$ is canonical, 55
$[\cdot]$	$[\cdot]_A$, if $(A, \prec)$ is canonical, 55
$\mathcal{M}$	a collection of reduction orders, 62
$\mathcal{M}(1)$	the members of $\mathcal{M}$ with $\Sigma = \Sigma^{(\leqslant 1)}$, 62
$\sim$	permutative equivalence on terms, 63
$\prec_{\mathsf{mpo}}$	MPO based on the precedence $\prec$, 63
$\prec_{\mathsf{lpo}}$	LPO based on the precedence $\prec$, 63
i	the special symbol of a KBO, 65
μ	the weight function of a KBO, 65
$\prec_{\mathsf{kbo}}$	a KBO based on the precedence $\prec$, 65
$s \equiv i^a s'$	s is s' labeled with $a \in \mathbb{N}$, 68
$\prec_{\mathsf{kbo2}}$	an alternative (equivalent) definition of $\prec_{\mathsf{kbo}}$, 68
ν	a companion to the weight function, but mapping into $\mathbb{N}^+$, 69
S	the input successor, 70
P	the output successor, 70
$\textsc{Comp}(\mathcal{M})$	the functions computable by a TRS terminating via $\mathcal{M}$, 70
$\textsc{Comp}_1(\mathcal{M})$	the functions computable by a TRS terminating via $\mathcal{M}$ using only one successor symbol, 70
f_i	in Section 5.2: a member of the fixed signature, 85
$\mathcal{A}_n$	a function used to encode tuples; based on fast growing functions, 86
$\mathcal{A}_n(a, c)$	an abbreviation of $\mathcal{A}_n((a), c)$, 86
$\mathcal{I}$	in Section 5.2: an interpretation based on $\mathcal{A}$, 87
$\varphi_{\bar{\alpha}}$	a branch of the Veblen φ function, 95
φ	the Veblen function, 95
ψ	the fixed point free Veblen function, 96
k	in Section 5.3, $k > 0$ is fixed and the $k + 1$-ary ψ is considered, 96
Δ_k	the first infinite ordinal closed under $+$ and the $k + 1$-ary ψ, 96
$\mathsf{Fix}(\bar{\alpha})$	the fixed points of $\varphi_{\bar{\alpha}}$, 97
$\psi(\bar{\alpha}, \beta)^*$	$\psi(\bar{\alpha}, \beta_0)$, if $\beta = \beta_0 + 1$, and β otherwise, 97
$\mathrm{IS}_{\bar{\alpha}}(\gamma)$	the (relative to $\bar{\alpha}$) interesting subterms of γ, 97
$\mathrm{MS}_{\bar{\alpha}}(\bar{\delta})$	the (relative to $\bar{\alpha}$) maximal interesting subterm of $\bar{\delta}$, 97
$\alpha[n]$	the n^{th} member of the standard fundamental sequence for $\alpha < \Delta_k$, 98
$\alpha[n, m]$	$(\ldots((\alpha[n])[n+1])\ldots)[m]$, 98
c^+	the configuration next to c, 98
Σ_0	the signature $\{\mathsf{0}, \mathsf{S}, +, \mathsf{P}\}$ used to denote ordinals below Δ_k, 99

$\mathsf{0}$	represents the ordinal 0, 99
S	represents the ordinal successor, 99
$\mathsf{+}$	represents $+$, 99
P	represents ψ, 99
$\mathrm{val}(s)$	the canonical value of a term in $\mathcal{T}(\Sigma_0)$, 99
$\mathcal{D}$	the set of standard terms; subset of $\mathcal{T}(\Sigma_0)$, 100
$\mathcal{D}(\alpha)$	the collection of standard terms with value α, 100
$s \times n$	$\mathsf{+}(\cdot, s)^{n-1}(s)$, 100
$\mathcal{D}(\mathsf{Lim})$	the set of standard terms denoting limit ordinals, 100
$\mathsf{Fix}(\bar{s})$	the analog to $\mathsf{Fix}(\bar{\alpha})$ on standard terms, 100
$\mathrm{MS}_{\bar{s}}(\bar{t})$	the analog to $\mathrm{MS}_{\bar{\alpha}}(\bar{\beta})$ on standard terms, 100
$d\langle n\rangle$	the (possibly nonstandard) term equivalent to $(\mathrm{val}(d))[n]$, 101
$d[n]$	the standard term equivalent to $(\mathrm{val}(d))[n]$, 101
$\mathcal{R}$	the TRS defined in Section 5.3.4; it simulates all Hydra battles below Δ_k, 102
Σ	in Section 5.3.4: the signature $\mathcal{R}$ is based on, 102
$\bullet$	a unary member of Σ, used to encode battle configurations, 102
$\circ$	a unary member of Σ, used at the end of a battle step, 102
$\llbracket\rrbracket$	a unary member of Σ, used to encode battle configurations, 102
M	a $k+1$-ary member of Σ, used for multiplications, 102
J_i	an $i+1$-ary member of Σ, used for iterations of P, 102
Q_{ij}	an $i+1$-ary member of Σ, used for (5.8i), 102
R_i	an $i+2$-ary member of Σ, used for (5.8j), 102
$\bullet\llbracket\rrbracket^{n+1}d$	the term which encodes the battle configuration $(\mathrm{val}(d), n)$, 103
$(\mathcal{P}, \prec)$	the lexicographic product of the usual $<$ on $\Delta_k \setminus \{0\}$, ω and ω, 108
α	in Section 5.3.5: an abbreviation for $(\alpha, 0, 0) \in \mathcal{P}$, 108
$(\mathcal{P}, \prec, \mathcal{F})$	the Σ-algebra used to establish total termination of $\mathcal{R}$, 108
$(\mathcal{R}, \Sigma, \prec)$	a well-founded ordered TRS, 113
$\mathrm{dl}^{\prec}_{\mathcal{R}}(s)$	like $\mathrm{dl}_{\mathcal{R}}(s)$, yet considering only $\prec$-derivations, 113
$\mathrm{Dl}^{\prec}_{\mathcal{R}}$	like $\mathrm{Dl}_{\mathcal{R}}$, but based on $\mathrm{dl}^{\prec}_{\mathcal{R}}$, 113
$\prec_{\mathsf{gkbo}}$	the ground version of $\prec_{\mathsf{kbo}}$, 115
$[s_1, \ldots, s_n]$	the term representing the list containing $s_1, \ldots, s_n$, 122
$s * t$	concatenation of the list s with the term t, 122
p	an attribute, 123
$x + 1_{\mathrm{p}}$	the list element $x + 1$, attributed by p, 123
s^n	n occurrences of the (possibly attributed) number term s in a list, 123

$\mathcal{R}_f$	a TRS terminating via KBO, simulating an RM-computation of f, 124
$\mathcal{H}$	Hofbauer's TRS which terminates via KBO and has size complexity beyond primitive recursion, 124
A	the attribute set used by $\mathcal{A}$, 126
$\mathcal{A}$	a TRS terminating via KBO whose shortest size complexity function grows beyond primitive recursion, 126
$\lhd$	homeomorphic embedding with only one label, 135
$K_f^{\lhd}$	the function mapping n to the first N such that, for every sequence of trees T_i of length N, there are $i < j$ with $T_i \trianglelefteq T_j$, 136
f_r	a control function with $r \in \mathbb{R}$; we have $f_r(n,m) = n + r \log_2(m)$, 136
$K_f^{\mathcal{R}}$	the function mapping n to the first N such that there is no $\mathcal{R}$-derivation of length N controlled by f, 136
p	in Section 7.1 p-adic representations (with $p \geqslant 2$) are considered, 139
a	in Section 7.1 we have $a = p - 1$, 139
$\lvert n\rvert_p$	the p-adic length of n, 139
i	in Section 7.1 p-adic representations get iterated i times, 139
$\lvert n\rvert_p^{(i)}$	the i times iterated p-adic length of n, 139
Σ_p	a signature containing $\underline{0}, \underline{1}, \ldots, \underline{a}, \overline{0}, \triangle, \blacktriangle$, 140
$\underline{m}$	the member of Σ_p representing the digit $m \in [0, a]$, 140
$\overline{0}$	the member of Σ_p representing the digit 0 carrying an overflow, 140
$\triangle$	the member of Σ_p used to separate two p-adic representations, 140
$\blacktriangle$	the constant in Σ_p, 140
$\mathrm{code}_p(n)$	the p-adic representation of n, 140
$\mathrm{code}_{p,i}(n)$	$\mathrm{code}_p(\lvert n\rvert_p^{(i)})$, 141
$\mathrm{code}^+_{p,i}(n)$	$\mathrm{code}_{p,i}(n+1)$, if it has the same size as $\mathrm{code}_{p,i}(n)$, and otherwise $\overline{0}\underline{0}^{\lvert n\rvert_p^{(i+1)}-1}$, 141
$\mathrm{code}^+_p(n)$	abbreviation of $\mathrm{code}^+_{p,0}(n)$, 141
$\mathrm{logs}_{p,i}(n)$	the $\triangle$-divided sequence from $\mathrm{code}_p(\lvert n\rvert_p^{(i)})$ to $\mathrm{code}_p(\lvert n\rvert_p^{(0)})$, 141
$\mathrm{logs}^+_{p,i}(n)$	$\mathrm{code}^+_{p,i}(n)$, followed by $\triangle\, \mathrm{logs}_{p,i-1}(n+1)$ or $\blacktriangle$, 141
$\mathcal{R}_p$	a totally terminating TRS over Σ_p which reduces $\mathrm{logs}_{p,i}(n)$ to $\mathrm{logs}^+_{p,i}(n)$, if $n > 0$, 141
$\mathcal{R}'$	a minor modification of the Hydra battle simulator $\mathcal{R}$, 144
Σ'_p	the union of the signatures Σ and Σ_p, 152
$\mathcal{R}'_p$	a TRS over Σ'_p, slightly extending both $\mathcal{R}'$ and $\mathcal{R}_p$, 152
K_r	an abbreviation of $K_{g_r}^{\mathcal{R}'_p}$, 157

Index

Curriculum Vitae

Ingo Lepper

geboren am 1. 10. 1969 in Münster,

ledig,

Vater Paul Lepper,
Mutter Maria Lepper, geb. Köper

Schulbildung

1976–1979	Grundschule in Everswinkel
1979–1988	Gymnasium in Warendorf
31. 5. 1988	Abitur in Warendorf

Studium

10/1988	Beginn des Studiums der Mathematik an der Westfälischen Wilhelms-Universität Münster
7. 10. 1997	Diplom im Fach Mathematik
06/1998	Beginn der Dissertation am Institut für mathematische Logik und Grundlagenforschung, betreut von Herrn Prof. Dr. Andreas Weiermann

Tätigkeiten

05/1996–03/1997	studentische Hilfskraft am Institut für mathematische Logik und Grundlagenforschung
01/1998–12/1999	Mitarbeit am DFG-Projekt »Beweistheorie und Rekursionstheorie«
seit 01/2000	wissenschaftlicher Mitarbeiter am Institut für mathematische Logik und Grundlagenforschung