

STAR GAMES AND HYDRAS

JÖRG ENDRULLIS^a, JAN WILLEM KLOP^b, AND ROY OVERBEEK^a

^a Vrije Universiteit Amsterdam, Amsterdam
e-mail address: j.endrullis@vu.nl, r.overbeek@vu.nl

^b Centrum voor Wiskunde en Informatica (CWI), Amsterdam
e-mail address: JWKlop1945@kpnmail.nl

ABSTRACT. The recursive path ordering is an established and crucial tool in term rewriting to prove termination. We revisit its presentation by means of some simple rules on trees (or corresponding terms) equipped with a ‘star’ as control symbol, signifying a command to make that tree (or term) smaller in the order being defined. This leads to star games that are very convenient for proving termination of many rewriting tasks. For instance, using already the simplest star game on finite unlabeled trees, we obtain a very direct proof of termination of the famous Hydra battle, direct in the sense that there is not the usual mention of ordinals. We also include an alternative road to setting up the star games, using a proof method of Buchholz, adapted by van Oostrom, resulting in a quantitative version of the star as control symbol. We conclude with a number of questions and future research directions.

DEDICATION

After graduating, Jos Baeten started his scientific work in the realm of *the higher infinite* with his Ph.D. thesis, appearing in the bibliography of Kanamori’s book [Kan94] about that realm (see Baeten [Bae84]). The second author cherishes many precious memories of the cooperation with Jos and Jan Bergstra in our years of process algebra around 1984.

The present note is dedicated to Jos in friendship, and includes several questions relating to the higher infinite of ordinals and cardinals.

Key words and phrases: well-quasi-ordering, recursive path ordering, iterative path ordering, Hydra battle, termination, Kruskal’s tree theorem, Buchholz Hydra.

1. INTRODUCTION

To understand difficult notions in mathematics, logic or informatics it sometimes helps to develop a “game view” on the notion in question. A typical example is the notion of bisimulation. The second author of this note, at the start of his new job around Christmas 1980 at the CWI¹, in trying to understand the then newly proposed notion of the *recursive path ordering* (RPO), found that his understanding was much eased by a “gamification”, consisting of some simple rules for moving a control symbol \star on finite labeled trees.

The star game version of RPO was called IPO, *iterative path order*. IPO was used fruitfully by Bergstra & Klop [BK85] to yield termination of a rewrite system evaluating process expressions in ACP, the *algebra of communicating processes*. Together with Jan Bergstra and Aart Middeldorp, the second author of the current paper wrote a course book (in Dutch) *Termherschrijfsystemen* [BKM89], while employed, respectively, in 1987 at the University of Amsterdam and the Vrije Universiteit Amsterdam. There the IPO star game is included, together with a proof of the crucial lemma of acyclicity and *Kruskal’s tree theorem*. This proof of the acyclicity lemma is published in the present paper for the first time (in English, the mentioned course book is in Dutch). The survey chapter [Klo92] did not include a proof of the acyclicity lemma.

The iterative path order has been extended in various directions. Klop, van Oostrom and de Vrijer [KvOdV06] have extended it to the *iterative lexicographic path order* (ILPO) after including a well-known rule for lexicographic extensions of RPO², see also [Ges90, BN98, Der82, KL80]. In subsequent studies, van Raamsdonk and Kop [KvR08, KvR09] have extended the framework of ILPO to higher-order rewriting [KvR08] and to deal with the recursive path order with *status* [KvR09]. Nowadays, the recursive path order is a standard ingredient of every leading termination prover. In particular, it is an important part of the termination prover *Jambox* [End, EZ15] developed by the first author. The recursive path order has also been applied fruitfully to prove confluence by decreasing diagrams [FvO13]. Decreasing diagrams [vO94a, EK13, EKO18] is one of the strongest techniques for proving confluence of (abstract) rewrite systems.

A peculiar matter with the IPO star game is that on the auxiliary objects, being finite trees adorned with control stars, the game reduction is *not* terminating (SN), while it is SN when restricted to the proper, unstarred, objects. Vincent van Oostrom [KvOdV06] discovered a very interesting variant of the IPO star game where the stars are given a *quantity of energy*, a natural number or ordinal. This variant had the remarkable property to be terminating even on the intermediate auxiliary trees (see Figure 1) while inducing the same reduction relation on unstarred terms. The termination proof in [KvOdV06] did not use Kruskal’s tree theorem plus acyclicity, but instead a very different proof method [Buc95], developed by Buchholz, in the framework of proof theory investigations concerned with powerful independence theorems and fast-growing hierarchies. The method was adapted by van Oostrom in [KvOdV06]. In our paper we have devoted a full section to this proof calling it ‘magical’, which of course is a subjective appraisal, but which just means that we find this route surprisingly elegant and effective, saving us from the more laborious details of the symbol tracing proof of acyclicity.

¹The Centrum Wiskunde & Informatica of which Jos Baeten is the current director. In 1980 the CWI was still called Mathematical Centre Amsterdam.

²For stars it was mentioned in [BKM89, Klo92]. For RPO it was defined in Dershowitz [DJ90] and several introductions to RPO, e.g., Zantema’s chapter on Termination in [Zan03].

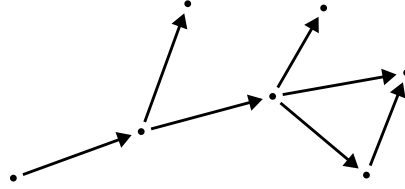
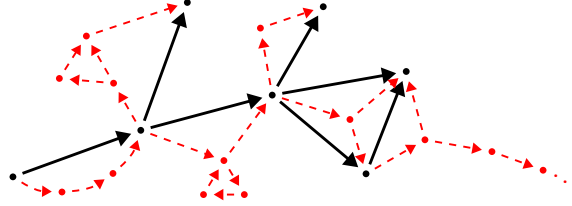
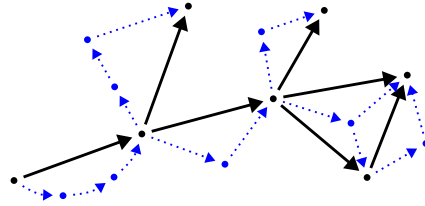
(a) Original terms with reduction relation \rightarrow_1 , to be proved SN.(b) Original terms together with auxiliary terms and reduction relation \rightarrow_2 such that $\rightarrow_1 \subseteq \rightarrow_2^+$, and \rightarrow_2^+ restricted to the original terms is SN; \rightarrow_2 itself need not be SN. It follows that \rightarrow_1 is SN. This is the situation of ILPO *with stars*.(c) Original terms together with auxiliary terms and reduction relation \rightarrow_3 such that \rightarrow_3 is SN, and $\rightarrow_2^+ = \rightarrow_3^+$, restricted to original terms. It follows that \rightarrow_1 is SN. This is the situation of ILPO *with number labels*.

FIGURE 1. Rewriting of objects simulated by star rules (b), red and dashed, and by number star rules (c), blue and dotted. Note that the red auxiliary rewriting relation is not always terminating, while the blue one is. We note that the non-terminating red parts of the reduction are not simply cut away; their effect is in a more subtle way after finite unwinding subsumed in the blue terminating reduction.

Recently, in a draft book on abstract rewriting by the first two authors, we were including the Kirby-Paris Hydra battle [KP82], a paradigm example in the termination area [Buc07, DM07, Mos09, Tou98]. The terminator Hercules succeeds in killing the monstrous Hydra, in the Greek mythology helped by the golden sword given to him by goddess Athena. While including this killer example of termination, a question, thirty years overdue, occurred to us: if the IPO star game is so flexible and versatile as it sometimes appeared to be, should it maybe also prove the victory of Hercules? And *lo and behold*, the termination proof of the Kirby-Paris Hydra battle via a simulation in the IPO game, is almost immediately obvious, as we shall demonstrate in Section 6. Additionally, we design an even more monstrous Hydra, and show that it also is no match for IPO.

Remarkably, the usual proofs for termination of the Kirby-Paris Hydra battle use ordinals in the form of Cantor normal forms, or even the more intricate ordinal diagrams [Tak13],

but the proof via Kruskal’s tree theorem and the star game in this paper does not refer to ordinals at all. Actually, ordinals are hiding in the background: the Kirby-Paris Hydra example requires only trees labelled with 0, i.e., unlabelled finite trees, and this game is paramount to the nested multiset ordering over natural numbers, which has, as is well-known, ordinal strength ϵ_0 [Klo92]. The more general star game, employing natural numbers as node labels, is much stronger, reaching the ordinal $\phi_\omega(0)$ in the Veblen notation [Der87].

However, there are monsters and giant monsters, some of which are much harder to conquer than the Kirby-Paris Hydra. The most well-known giant monster is the Buchholz Hydra [HO98, Oka88, Buc87], described in this paper. It is far beyond a simulation by the IPO star game as presented here, because its astronomically large ordinal dwarfs even the ordinals in the Veblen hierarchy.

Outline. In the introduction we give a lengthy historical account of sources leading up to the current survey paper on star games as a termination framework. In Section 2 we recapitulate the main notions and notations for reading this paper, including a nutshell account of some of the first large ordinals. Section 3 presents the main star game of this paper, the star game called *iterative path order* (IPO).

In Section 4 we establish termination of IPO using Kruskal’s tree theorem and acyclicity. Section 4.1 gives a brief introduction to Kruskal’s tree theorem and Higman’s lemma. In Section 4.3 we prove the pivotal lemma of acyclicity.

Section 5 contains a different route towards the same result of SN of IPO, via a refinement of stars, and a proof method that is interesting in its own right.

Section 6 presents three Hydras: the “classical” one by Kirby and Paris; a super Hydra of our making, still provably terminating by our IPO star game; and a hyper Hydra of Buchholz, that is beyond the power of IPO.

The final section, Section 7, presents several directions of further research.

2. PRELIMINARIES

In this section we will briefly enumerate and describe some of the main notions and notations figuring in this paper. We will work with *unranked* terms (basically labelled, ordered trees) where the symbols of the signature do *not* have a fixed arity. We consider unordered trees (algebraic trees) as equivalence classes of terms modulo the permutation of arguments.

2.1. Abstract rewriting.

Definition 2.1 (Abstract rewriting). *An abstract reduction system (ARS) $\mathcal{A} = \langle A, \rightarrow \rangle$ consists of a set A of objects and a binary rewrite or reduction relation $\rightarrow \subseteq A \times A$.*

Let $\mathcal{A} = \langle A, \rightarrow \rangle$ be an ARS.

Definition 2.2 (Rewrite sequence). *A rewrite sequence (or reduction) in \mathcal{A} is a sequence of rewrite steps $a_0 \rightarrow a_1 \rightarrow a_2 \rightarrow \dots$ where $a_0, a_1, \dots \in A$. If the rewrite sequence is finite, say $a_0 \rightarrow a_1 \rightarrow \dots \rightarrow a_n$ for $n \geq 0$, the sequence is said to have length n and to end in a_n . If $a_0 = a_n$ and $n > 0$, then we speak of a reduction cycle. If there is a finite rewrite sequence from a and to (ending in) b we write $a \rightarrow^* b$ or $a \twoheadrightarrow b$.*

Definition 2.3. *An object $a \in A$ is a normal form if there is no $b \in A$ such that $a \rightarrow b$.*

Definition 2.4. *The ARS \mathcal{A} is said to be*

- (1) terminating (or strongly normalizing), denoted $\text{SN}(\mathcal{A})$, if there are no infinite reductions;
- (2) weakly normalizing if for every $a \in A$ there exists a normal form b such that $a \rightarrow b$;
- (3) confluent (has the Church-Rosser property), denoted $\text{CR}(\mathcal{A})$, if $\leftarrow \cdot \rightarrow \subseteq \rightarrow \cdot \leftarrow$;
- (4) acyclic if there are no reduction cycles.

2.2. Term rewriting. We introduce some basic notions of term rewriting and unranked terms. For background on term rewriting with unranked terms, we refer to [Kut02, Kut04]. For general background in (term) rewriting we refer to Terese [Ter03], Klop [Klo92], Dershowitz-Jouannaud [DJ90], Baader-Nipkow [BN98].

Let Σ be a (possibly infinite) set of symbols, called the (unranked) signature. Let $\mathcal{X} = \{x, y, z, \dots\}$ be an infinite set of variables such that $\mathcal{X} \cap \Sigma = \emptyset$.

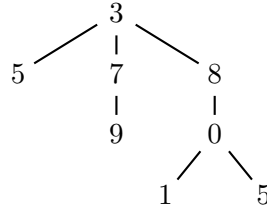
Definition 2.5 (Terms). *The set of (unranked) terms $\mathcal{T}(\Sigma, \mathcal{X})$ over the signature Σ and set of variables \mathcal{X} is inductively defined by:*

- (i) $\mathcal{X} \subseteq \mathcal{T}(\Sigma, \mathcal{X})$, and
- (ii) $f(t_1, \dots, t_n) \in \mathcal{T}(\Sigma, \mathcal{X})$ whenever $f \in \Sigma$ and $t_1, \dots, t_n \in \mathcal{T}(\Sigma, \mathcal{X})$.

We denote terms $f()$ by f for short.

As we only consider unranked terms, we will simply speak of terms.

Remark 2.6. *The set of terms $\mathcal{T}(\Sigma, \mathcal{X})$ is isomorphic to the set of ordered, rooted finite trees with inner nodes labeled by elements from Σ and leaves labelled by elements from $\Sigma \cup \mathcal{X}$. Here “ordered” means that there is an order between node siblings. In illustrations, the sibling-order is the left-to-right. We frequently depict terms by ordered trees. For instance,*



depicts the term $3(5, 7(9), 8(0(1, 5)))$.

Definition 2.7 (Variables). *The set of variables $\text{Var}(t) \subseteq \mathcal{X}$ of $t \in \mathcal{T}(\Sigma, \mathcal{X})$ is given by:*

$$\text{Var}(x) = \{x\} \quad \text{Var}(f(t_1, \dots, t_n)) = \text{Var}(t_1) \cup \dots \cup \text{Var}(t_n)$$

Definition 2.8 (Positions). *The set of positions $\text{Pos}(t) \subseteq \mathbb{N}^*$ of $t \in \mathcal{T}(\Sigma, \mathcal{X})$ is defined by:*

$$\text{Pos}(x) = \{\epsilon\} \quad \text{Pos}(f(t_1, \dots, t_n)) = \{\epsilon\} \cup \{ip \mid 1 \leq i \leq n, p \in \text{Pos}(t_i)\}$$

For $p \in \text{Pos}(t)$, the subterm $t|_p$ of t at position p is defined by:

$$t|_\epsilon = t \quad f(t_1, \dots, t_n)|_{ip} = t_i|_p$$

Definition 2.9 (Substitutions). *A substitution σ is a map $\sigma : \mathcal{X} \rightarrow \mathcal{T}(\Sigma, \mathcal{X})$. We extend the domain of substitutions σ to terms $\mathcal{T}(\Sigma, \mathcal{X})$ as follows:*

$$\sigma(f(t_1, \dots, t_n)) = f(\sigma(t_1), \dots, \sigma(t_n))$$

We write $\{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}$ for the substitution σ given by $\sigma(x_1) = t_1, \dots, \sigma(x_n) = t_n$ and $\sigma(y) = y$ for every $y \in \mathcal{X} - \{x_1, \dots, x_n\}$.

Definition 2.10 (Contexts). A context C is a term $\mathcal{T}(\Sigma, \mathcal{X} \cup \{\square\})$ containing precisely one occurrence of \square , that is, there is precisely one position $p \in \text{Pos}(C)$ such that $C|_p = \square$. For a context C and a term t , we write $C[t]$ for the term $\{\square \mapsto t\}(C)$.

Definition 2.11 (Rewriting). A rewrite rule $\ell \rightarrow r$ over Σ and \mathcal{X} is a pair $(\ell, r) \in \mathcal{T}(\Sigma, \mathcal{X}) \times \mathcal{T}(\Sigma, \mathcal{X})$ of terms such that the left-hand side ℓ is not a variable ($\ell \notin \mathcal{X}$), and all variables in the right-hand side r occur in ℓ ($\text{Var}(r) \subseteq \text{Var}(\ell)$).

A term rewrite system (TRS) \mathcal{R} over Σ and \mathcal{X} is a set of rewrite rules over Σ and \mathcal{X} . The rewrite relation $\rightarrow_{\mathcal{R}} \subseteq \mathcal{T}(\Sigma, \mathcal{X}) \times \mathcal{T}(\Sigma, \mathcal{X})$ generated by \mathcal{R} consist of all pairs

$$C[\sigma(\ell)] \rightarrow_{\mathcal{R}} C[\sigma(r)]$$

for contexts C , rules $\ell \rightarrow r \in \mathcal{R}$, and substitutions $\sigma : \mathcal{X} \rightarrow \mathcal{T}(\Sigma, \mathcal{X})$. Furthermore, we write $\rightarrow_{\mathcal{R},p}$ whenever additionally $C|_p = \square$. We drop the subscript \mathcal{R} in $\rightarrow_{\mathcal{R}}$ and $\rightarrow_{\mathcal{R},p}$ whenever \mathcal{R} is clear from the context.

Notation 2.12 (Argument vector notation). We use expressions of the form \vec{x} to denote a finite (possibly empty) sequence of distinct variables (x_1, \dots, x_n) for some $n \in \mathbb{N}$. For example, the rule schema

$$f(\vec{x}, y, \vec{z}) \rightarrow f(\vec{x}, \vec{z})$$

enumerates rules

$$f(x_1, \dots, x_n, y, z_1, \dots, z_m) \rightarrow f(x_1, \dots, x_n, z_1, \dots, z_m)$$

for every $m, n \geq 0$. The rule schema thus allows deleting an arbitrary subtree y under head symbol f .

2.3. Tree rewriting. We introduce (finite, rooted, labelled, unordered) trees as equivalence classes of terms modulo the permutation of arguments. In our setting this view is convenient as we have already introduced terms, we use terms to denote trees, and we will switch from trees to terms by “freezing” in order to make use of positions and tracing.

Definition 2.13 (Swapping). Let \mathcal{R}_{\simeq} be the TRS consisting of the rules

$$f(\vec{u}, x, y, \vec{w}) \rightarrow f(\vec{u}, y, x, \vec{w})$$

for every $f \in \Sigma$. We write \rightarrow_{\simeq} for the rewrite relation $\rightarrow_{\mathcal{R}_{\simeq}}$, and we use \simeq to denote the equivalence relation generated by \rightarrow_{\simeq} .

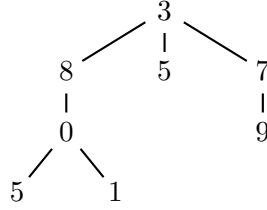
Note that \simeq coincides with the transitive closure of \rightarrow_{\simeq} , that is, $\simeq = \rightarrow_{\simeq}^*$.

Definition 2.14 (Trees). The set of (unordered) trees $\mathbb{T}(\Sigma, \mathcal{X})$ over Σ and \mathcal{X} is $\mathcal{T}(\Sigma, \mathcal{X})/\simeq$, the equivalence classes of terms modulo \simeq . For a term $t \in \mathcal{T}(\Sigma, \mathcal{X})$, we write t_{\simeq} for the equivalence class of t with respect to \simeq . So $\mathbb{T}(\Sigma, \mathcal{X}) = \{t_{\simeq} \mid t \in \mathcal{T}(\Sigma, \mathcal{X})\}$.

Example 2.15. In a tree the subtrees of a node may be permuted without changing the tree. So $3(5, 7(9), 8(0(1, 5)))_{\simeq}$ denotes the same tree as $3(8(0(5, 1)), 5, 7(9))_{\simeq}$.

In this paper we speak of terms whenever the arguments are ordered, and we speak of trees when the arguments are unordered. Most of this paper will be concerned with trees. In the proof in Section 4.3, however, we “freeze” trees, obtaining terms, in order to make use of positions and tracing throughout rewrite sequences. This freezing procedure was also used in [Klo92].

Remark 2.16. *We depict trees in the same way as we depict terms:*



It will be clear from the context whether such illustration depicts a term (ordered) or a tree (unordered).

We rewrite trees using term rewriting systems.

Definition 2.17. *Let \mathcal{R} be a term rewriting system. We extend $\rightarrow_{\mathcal{R}}$ from terms $\mathcal{T}(\Sigma, \mathcal{X})$ to trees $\mathbb{T}(\Sigma, \mathcal{X})$ by defining $s_{\simeq} \rightarrow_{\mathcal{R}} t_{\simeq}$ whenever $s, t \in \mathcal{T}(\Sigma, \mathcal{X})$ and $s \rightarrow_{\mathcal{R}} t$.*

So $\{(s_{\simeq}, t_{\simeq}) \mid s, t \in \mathcal{T}(\Sigma, \mathcal{X}), s \rightarrow_{\mathcal{R}} t\}$ is the rewrite relation induced by \mathcal{R} on trees.

A term rewrite system \mathcal{R} thus induces abstract rewrite systems on terms and trees, namely $\langle \mathcal{T}(\Sigma, \mathcal{X}), \rightarrow_{\mathcal{R}} \rangle$ and $\langle \mathbb{T}(\Sigma, \mathcal{X}), \rightarrow_{\mathcal{R}} \rangle$, respectively. Hence, the notions of rewrite sequences, reduction cycles, normal forms, termination, confluence, acyclicity and so forth, carry over from abstract rewriting to term and tree rewriting.

2.4. Orders. We also presuppose some familiarity with the basics of order theory, partial orders, well-founded orders, and trees. Particularly important in this paper is the notion of a *well-quasi-order (wqo)*, an order that often proves fruitful in the termination setting [SSW20].

Definition 2.18. *(Well-quasi-order) A quasi-order is a binary relation that is reflexive and transitive. A well-quasi-order (wqo) is a quasi-order \preceq such that any infinite sequence of elements $a_0, a_1, \dots, a_i, \dots, a_j, \dots$ contains an increasing pair $a_i \preceq a_j$ with $i < j$.*

Proposition 2.19. *Any wqo is well-founded.*

2.5. Ordinals. An *ordinal* [Rub67] is a transitive set whose elements are again transitive sets. A set A is *transitive* if $x \subseteq A$ for every $x \in A$. In this view, every ordinal α is the set of all ordinals smaller than α ; thus $\alpha = \{\beta \mid \beta < \alpha\}$. So the class of all ordinals is totally ordered by the membership relation. The least ordinal is $\{\}$, called 0. The *successor* α^+ of an ordinal α is the ordinal $\alpha^+ = \alpha \cup \{\alpha\}$. A *limit* ordinal is an ordinal that is neither 0 nor a successor ordinal. So the smallest ordinals are

$$0 = \{\}, \quad 1 = 0^+ = \{\{\}\}, \quad 2 = 1^+ = \{\{\}, \{\{\}\}\}, \quad \dots, \quad \omega, \dots$$

The ordinal ω is the least infinite ordinal, $\omega = \{0, 1, 2, 3, \dots\}$.

The Kirby-Paris Hydra and the Buchholz Hydra treated in this paper have traditionally been studied and used in the context of ordinal analysis [Rat06], a branch of proof theory that measures the strength of mathematical theories in terms of ordinals. More specifically, the *ordinal strength* (or *proof-theoretic ordinal*) of such a theory T can be characterized as the smallest ordinal that T cannot prove well-founded. In this paper, we will say that a Hydra has *ordinal strength* α if a proof of its termination requires a theory with ordinal strength at least α .

While we will not perform any ordinal analysis in this paper, we will ask questions pertaining to it, in particular in relation to our newly introduced Star Hydra. For this reason, we provide the following overview of large ordinals, assuming only an understanding of the limit ordinal ω and ordinal arithmetic [Rub67].

2.5.1. *Ordinals up to ϵ_0 .* Any ordinal can be written as a polynomial

$$\omega^{\alpha_1} c_1 + \omega^{\alpha_2} c_2 + \cdots + \omega^{\alpha_n} c_n,$$

where $\alpha_1 > \alpha_2 > \cdots > \alpha_n \geq 0$ are ordinals and the c_i are positive integers. This representation is called Cantor normal form (CNF).

Using ordinals $\leq \omega$ as elementary symbols, one can express ordinals lying on the spectrum $\omega, \omega^\omega, \omega^{\omega^\omega}, \dots$ in CNF. The limit ordinal for this sequence is the first solution to the fixed point equation $\alpha = \omega^\alpha$, called ϵ_0 . The weakest Hydra treated in this paper, the Kirby-Paris Hydra, has ordinal strength ϵ_0 .

The equation $\alpha = \omega^\alpha$ in fact has infinitely many fixed points. The ι -th ordinal that is a solution to $\alpha = \omega^\alpha$ is called ϵ_ι . If we similarly generalize the subscript of ϵ , the limit for what we can express is the first fixed point solution to $\alpha = \epsilon_\alpha$. Call this ordinal ζ_0 . Once again generalizing the subscript, we can express ordinals up to the first fixed point solution to $\alpha = \zeta_\alpha$. If we were to continue in this fashion, we would soon exhaust the Greek alphabet. Veblen functions provide a systematic solution to this problem.

2.5.2. *Ordinals up to Γ_0 .* The Veblen hierarchy $\phi_\alpha(\gamma)$ can be defined by two clauses:

- for $\alpha = 0$, $\phi_\alpha(\gamma) = \omega^\gamma$, and
- for $\alpha > 0$, $\phi_\alpha(\gamma)$ is the $(1 + \gamma)$ -th common fixed point of the functions ϕ_β for all $\beta < \alpha$.

So in particular, we obtain $\phi_1(0) = \epsilon_0$ and $\phi_2(0) = \zeta_0$. The first solution to the fixed point equation $\phi_\alpha(0) = \alpha$ is the Feferman–Schütte ordinal, and it is denoted by Γ_0 [Veb08, Sch65].

2.5.3. *Ordinals beyond Γ_0 .* Multiple definitions exist that allow us to go beyond the Veblen hierarchy. Buchholz’s ψ functions [Buc86] are one example. The ordinal strength of the strongest Hydra in our paper, the powerful Buchholz Hydra, exceeds even all the ordinals expressible using the ψ functions.

3. STAR GAMES

Proving termination (SN) of various rewriting sequences is of major importance. Not only is it a central aspect of program correctness, it is also a stepping stone towards confluence, using Newman’s lemma. Although it is in general undecidable whether a (string or term) rewrite system is terminating, in many instances termination can be proven, and various techniques have been developed to do so.

In this paper we present one of the most powerful of such termination proof techniques: the *recursive path order* [Der82] as developed by Dershowitz. We will use the presentation as in [KvOdV06], where the inductive definitions of the usual presentation are replaced by a reduction procedure that is easier to grasp to our taste. We develop this application, employing trees with annotations, natural numbers or ordinals at the nodes, and an auxiliary

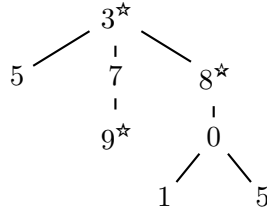


FIGURE 2. A tree with stars.

marker \star at some nodes. It turns out that we will arrive at vast generalizations of the well-known *multiset orders* [Man80, vO94b].

Definition 3.1 (Starred signature). $\Sigma^\star = \{f^\star \mid f \in \Sigma\}$ is the starred signature. It is assumed that Σ is unstarred, i.e., it does not contain a symbol of the form x^\star .

Notation 3.2. We abbreviate $\mathcal{T}(\Sigma \cup \Sigma^\star, \mathcal{X})$ as $\mathcal{T}(\Sigma, \mathcal{X})^\star$, and $\mathbb{T}(\Sigma \cup \Sigma^\star, \mathcal{X})$ as $\mathbb{T}(\Sigma, \mathcal{X})^\star$.

Definition 3.3 (Iterative path order). Let $> \subseteq \Sigma \times \Sigma$ be a well-founded partial order, often called the precedence ordering. The iterative path order (IPO) on trees $\mathbb{T}(\Sigma, \mathcal{X})$ is

$$\rightarrow^+ \cap (\mathbb{T}(\Sigma, \mathcal{X}) \times \mathbb{T}(\Sigma, \mathcal{X})).$$

Here \rightarrow is the rewrite relation induced on $\mathbb{T}(\Sigma, \mathcal{X})$ by the TRS *Star* consisting of the rules

$$\begin{array}{ll}
 \text{(PUT)} & f(\vec{x}) \rightarrow f^\star(\vec{x}) \\
 \text{(SELECT)} & f^\star(\vec{x}, y, \vec{z}) \rightarrow y \\
 \text{(COPY)} & f^\star(\vec{x}) \rightarrow g(\underbrace{f^\star(\vec{x}), \dots, f^\star(\vec{x})}_{0 \text{ or more}}) \quad \text{if } f > g \\
 \text{(DOWN)} & f^\star(\vec{x}, g(\vec{y}), \vec{z}) \rightarrow f(\vec{x}, \underbrace{g^\star(\vec{y}), \dots, g^\star(\vec{y})}_{0 \text{ or more}}, \vec{z})
 \end{array}$$

for every $f, g \in \Sigma$.

Overloading names and notation, we also use \rightarrow to denote the rewrite relation on $\mathcal{T}(\Sigma, \mathcal{X})^\star$ induced by these rules.

Remark 3.4. Although they are rule schemas in the technical sense, we will often refer to PUT, SELECT, COPY, and DOWN simply as rules. Similarly for many other rule schemas in this paper.

The idea behind *Star* is that marking the root symbol of a tree, by means of the PUT rule, corresponds to the obligation to make that tree *smaller*, whereas the other rules correspond to ways in which this can be brought about:

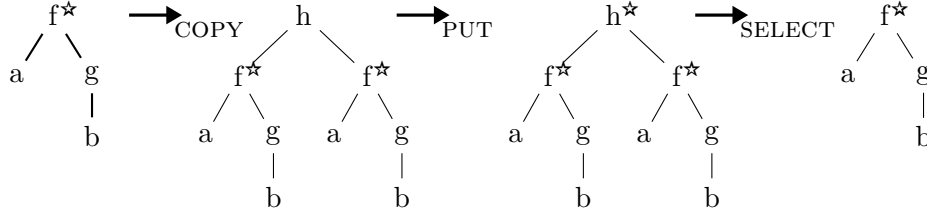
- (1) The SELECT rule expresses that selecting one of the children of a tree makes it smaller.
- (2) The COPY rule expresses that a tree t can be made smaller by putting finitely many copies of trees smaller than t below a root symbol g which is less heavy than the root symbol f of t .
- (3) The DOWN rule expresses that a tree can be made smaller by replacing one of its subtrees with finitely many smaller subtrees.

Remark 3.5 (DOWN vs. LEX). *In the literature on iterative lexicographic path orders, one frequently finds the LEX rule [Klo92, Ges90, KvOdV06, KvR09] instead of the DOWN rule:*

$$(LEX) \quad f^\star(\vec{x}, g(\vec{y}), \vec{z}) \rightarrow f(\vec{x}, g^\star(\vec{y}), \underbrace{\ell, \dots, \ell}_{0 \text{ or more}}) \quad \text{where } \ell = f^\star(\vec{x}, g(\vec{y}), \vec{z})$$

The LEX rule is stronger than the DOWN rule in the sense that the latter can be derived using the former (in combination with the rules PUT and SELECT). However, the LEX rule is only valid on terms, but it leads to non-termination in the setting of (unordered) trees. The DOWN rule, on the other hand, is valid also on trees, and this is the setting that we are focusing on in this paper.

Example 3.6 (A cyclic reduction). *A cyclic reduction*



in *Star*. (Here $h < f$.)

Clearly, *Star* is neither SN, nor CR. However, we are interested in termination on star-free trees. It turns out that *Star* does not admit infinite rewrite sequences that contain an infinite number of star-free trees. In other words, the restriction of the transitive closure \rightarrow^+ to $\mathbb{T}(\Sigma, \mathcal{X})$ is terminating.

Theorem 3.7. *The relation \rightarrow^+ , restricted to $\mathbb{T}(\Sigma, \mathcal{X})$, is SN.*

We will present two proofs of this theorem.

The first proof makes use of Kruskal's tree theorem in combination with acyclicity of the reduction relation \rightarrow . This route has also been used in the seminal paper [Der82] introducing the recursive path order. Nevertheless, the proof that we present has a very different flavour. In [Der82] the recursive path ordering is defined via an inductive definition, and the crucial step in the termination proof is establishing transitivity of the defined order. In our setting, transitivity of the iterative path order is immediate since the order is defined via the transitive closure of \rightarrow . Our proof, presented in Section 4, establishes acyclicity by tracing symbol occurrences throughout \rightarrow reductions.

The second proof, presented in Section 5, is based on well-founded induction, as in [Buc95, KvOdV06]. This proof gives a stronger result than the first proof in the presence of infinite signatures Σ , as it establishes termination of the iterative path order for arbitrary well-founded partial orders $(\Sigma, <)$. For the first proof, (Σ, \leq) needs to be a wqo due to the use of Kruskal's tree theorem.

Theorem 3.7 is of significant practical interest because of the following corollary, which intuitively states that a relation R is terminating if it can be simulated by \rightarrow^+ .

Corollary 3.8 (Termination by simulation). *If $R \subseteq \mathbb{T}(\Sigma, \mathcal{X}) \times \mathbb{T}(\Sigma, \mathcal{X})$ and $R \subseteq \rightarrow^+$, then $\text{SN}(R)$.*

We illustrate this corollary by applying it to a simple word rewrite system.

Example 3.9 (*Star in practice*). Consider the rewrite system with signature $\Sigma = \{1, 0\}$ and the single rewrite rule

$$1(0(x)) \rightarrow 0(0(1(x))) .$$

By defining the precedence ordering $1 > 0$, we obtain the IPO reduction

$$\begin{aligned} 1(0(x))_{\simeq} &\rightarrow_{\text{PUT}} 1^{\star}(0(x))_{\simeq} \\ &\rightarrow_{\text{COPY}} 0(1^{\star}(0(x)))_{\simeq} \\ &\rightarrow_{\text{COPY}} 0(0(1^{\star}(0(x))))_{\simeq} \\ &\rightarrow_{\text{DOWN}} 0(0(1(0^{\star}(x))))_{\simeq} \\ &\rightarrow_{\text{SELECT}} 0(0(1(x)))_{\simeq} \end{aligned}$$

on unstarred trees. Since the conditions of Corollary 3.8 are met, it follows that \rightarrow is terminating.

A more interesting example is given in Section 6, where *Star* is used to show that the Kirby-Paris Hydra is terminating.

4. THE FIRST PROOF: THE SCENIC ROUTE

In this section, we present the first proof of Theorem 3.7 based on Kruskal's tree theorem, which presents a well-known way to establishing termination of the star games [Klo92]. Due to the use of Kruskal's tree theorem, this proof requires that (Σ, \leq) is a well-quasi-order (wqo). The proof in this section is centered around the following simple, but pivotal, proposition.

Proposition 4.1. *Let $\mathcal{A} = \langle A, \rightarrow \rangle$ be an ARS and suppose that*

- (i) $\leq \subseteq A \times A$ is a wqo,
- (ii) $a \leq b \implies b \rightarrow^+ a$, and
- (iii) \rightarrow is acyclic.

Then the reduction \rightarrow is SN.

Proof. Assume for a contradiction that there is an infinite rewrite sequence $a_0 \rightarrow a_1 \rightarrow \dots$. Then since \leq is a wqo on A , there exists a pair $a_i \rightarrow^+ a_j$ with $a_i \leq a_j$. Thus by (2), $a_j \rightarrow a_i$. Hence $a_i \rightarrow^+ a_i$, contradicting that \rightarrow is acyclic. So \rightarrow must be SN. \square

The idea behind Proposition 4.1 is illustrated in Figure 3.³

The remainder of this section is devoted to establishing conditions (i)–(iii) of Proposition 4.1 for $A = \mathbb{T}(\Sigma, \mathcal{X})$ and $\rightarrow = \rightarrow^+ \cap (\mathbb{T}(\Sigma, \mathcal{X}) \times \mathbb{T}(\Sigma, \mathcal{X}))$. Then termination of \rightarrow^+ on unstarred trees follows. In Subsection 4.1, we recall a well-known embedding relation \preceq on trees, which is a wqo as stated by the beautiful Kruskal's tree theorem [NW67, RW93, Wei94, Gal91]. In Subsection 4.2, we clarify the relation between \preceq and \rightarrow^+ , and show that $s \preceq t \implies t \rightarrow^+ s$. In Subsection 4.3, finally, we demonstrate that \rightarrow^+ is acyclic on unstarred trees by means of an elaborate symbol tracing proof.

³Proposition 4.1 can be viewed as a generalization of [MZ97, Theorem 4.4], the termination of simplification orders, to abstract reduction systems. In [MZ97, Theorem 4.4], the objects are terms, and the wqo \leq is the homeomorphic embedding.

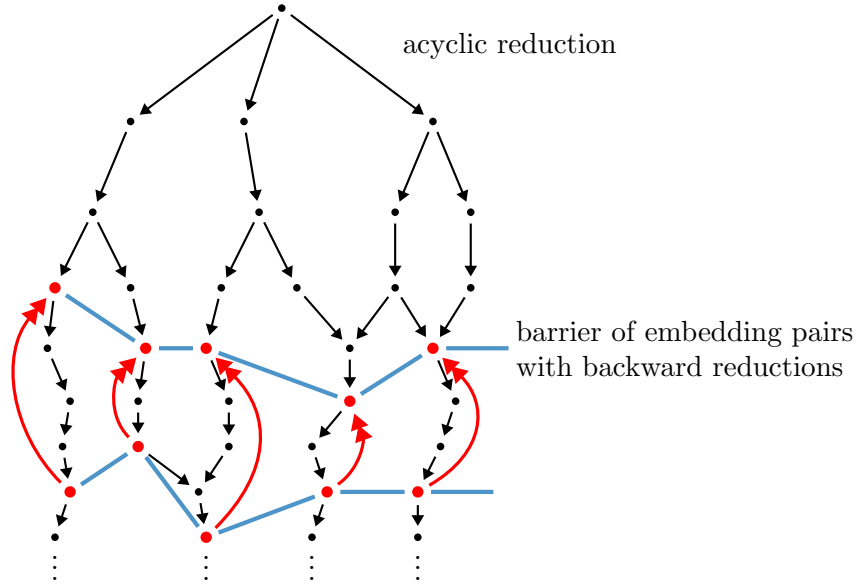


FIGURE 3. A wqo provides a barrier for an acyclic reduction, if it subsumes the converse of the wqo embedding \triangleleft . The well-founded part of the reduction tree cut off by the wqo barrier has an ordinal height that measures how soon the barrier with its embedded pairs is manifesting itself.

4.1. A reminder of Kruskal’s tree theorem. A standard embedding relation on trees exists, called the *homeomorphic embedding*, for which we need the following auxiliary definitions. Let α and β be nodes in a tree. We write $\alpha \ll \beta$ if β lies within the subtree rooted at α and $\alpha \neq \beta$. We write $\alpha \wedge \beta$ for the supremum of α and β with respect to \ll . In other words, $\alpha \wedge \beta$ is the unique node that lies on the intersection of the path from α to β , the path from α to the root and the path from β to the root.

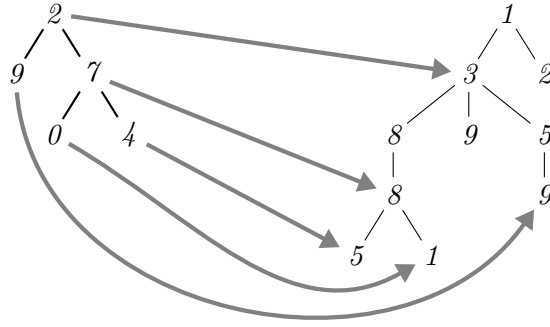
Definition 4.2 (Tree embedding [Gal91]). A tree $s \in \mathbb{T}(\Sigma, \mathcal{X})$ is homeomorphically embedded in tree $t \in \mathbb{T}(\Sigma, \mathcal{X})$, notation $s \preceq t$, if there is an injective function $\varphi : \text{nodes}(s) \rightarrow \text{nodes}(t)$ such that

- (1) φ is supremum preserving ($\varphi(\alpha \wedge \beta) = \varphi(\alpha) \wedge \varphi(\beta)$), and
- (2) φ is label increasing ($\text{label}(\alpha) \leq \text{label}(\varphi(\alpha))$).

Here \leq is the wqo on the node labels Σ .

For the labels, we will typically use natural numbers with the usual order \leq .

Example 4.3. We have $2(9, 7(0, 4))_{\simeq} \preceq 1(3(8(8(5, 1)), 9, 5(9)), 2)_{\simeq}$ due to the mapping:



Clearly, \preceq is a partial order on $\mathbb{T}(\Sigma, \mathcal{X})$. Moreover it satisfies the following remarkable property:

Theorem 4.4 (Kruskal’s tree theorem [Kru60]). *Let t_0, t_1, t_2, \dots be an infinite sequence of trees in $\mathbb{T}(\Sigma, \mathcal{X})$, and \leq a wqo on the labels Σ . Then $t_i \preceq t_j$ for some $i < j$.*

Remark 4.5. *Kruskal’s tree theorem can be phrased as follows: if (Σ, \leq) is a wqo, then $(\mathbb{T}(\Sigma, \mathcal{X}), \preceq)$ is a wqo.*

Kruskal’s tree theorem generalizes Higman’s lemma, a similar result originally stated for words over finite alphabets.

Corollary 4.6 (Higman’s lemma [Hig52]). *Let (A, \leq) be a (possibly infinite) wqo alphabet. Then the set of words (A^*, \leq_*) is a wqo, where $w \leq_* v$ if w can be obtained from v by decreasing and erasing an arbitrary number of letters in v .*

4.2. Embeddings and IPO. We now clarify the relation between the homeomorphic embedding \preceq and \rightarrow .

First of all, we believe that the homeomorphic embedding relation can be more conveniently defined via term rewriting as follows.

Definition 4.7 (Tree embedding via term rewriting). *Define the TRS \mathcal{E} to consist of the following rules:*

$$\begin{aligned}
 & \text{(removing a subtree)} && f(\vec{x}, y, \vec{z}) \rightarrow f(\vec{x}, \vec{y}) \\
 & \text{(selecting a subtree)} && f(\vec{x}, y, \vec{z}) \rightarrow y \\
 & \text{(decreasing a label)} && f(\vec{x}) \rightarrow g(\vec{x}) \quad \text{if } f > g
 \end{aligned}$$

for $f, g \in \Sigma$. Here, $f > g$ means that $f \geq g$ and not $f \leq g$. We use $\rightarrow_{\mathcal{E}}$ to denote the rewrite relation induced by \mathcal{E} on trees. We say that s can be homeomorphically embedded in t (via tree rewriting) if $t \rightarrow_{\mathcal{E}}^* s$.

Remark 4.8. *The term rewrite system \mathcal{E} can be used to define both the homeomorphic embedding on trees and on terms. Note that the rewrite system does not permute arguments. So it preserves the argument order (sibling order) when applied to terms.*

Lemma 4.9 (Equivalence of embedding relations). $\preceq = \leftarrow_{\mathcal{E}}^*$.

Proof. We provide a proof for \subseteq , which is the only direction relied on in this paper.

Assume $s \preceq t$ is established by some witnessing homeomorphism ϕ . For a tree u and a node n in u , we use $u|_n$ to denote the subtree of u rooted at n . It suffices to prove that (\dagger) for node every n in s , $t|_{\phi(n)} \rightarrow^* s|_n$. For then $t \rightarrow^* t|_{\phi(\text{root}(s))}$ using 0 or more selecting steps, and $t|_{\phi(\text{root}(s))} \rightarrow^* s|_{\text{root}(s)} = s$ by (\dagger) . We prove (\dagger) by structural induction on s . For the base case, s is a leaf n . Then $t|_{\phi(n)} \rightarrow^* n = s|_n$ by removing all of the subtrees of $\phi(n)$, and decreasing its label.

For the inductive step, $s = n(s_0, \dots, s_k)_{\simeq}$ for some $k \in \mathbb{N}$. For every $0 \leq i < j \leq k$, the roots of s_i and s_j have n as their supremum in s , and thus $\phi(\text{root}(s_i))$ and $\phi(\text{root}(s_j))$ have $\phi(n)$ as their supremum in t . This implies that $\phi(\text{root}(s_0)), \dots, \phi(\text{root}(s_k))$ are each in distinct subtrees that are rooted at $\phi(n)$. So $t|_{\phi(n)} = \phi(n)(t_0, \dots, t_k, \dots, t_{k+k'})_{\simeq}$ for some $k' \geq 0$ such that $\phi(\text{root}(s_i))$ is among the nodes of t_i for all $i \leq k$. Now we obtain

$$\begin{aligned}
t|_{\phi(n)} &\rightarrow^* \phi(n)(t_0, \dots, t_k)_{\simeq} && \text{removing subtrees} \\
&\rightarrow^* \phi(n)(t|_{\phi(\text{root}(s_0))}, \dots, t|_{\phi(\text{root}(s_k))})_{\simeq} && \text{selecting in the } t_i\text{'s} \\
&\rightarrow^* \phi(n)(s|_{\text{root}(s_0)}, \dots, s|_{\text{root}(s_k)})_{\simeq} && \text{induction hypothesis} \\
&= \phi(n)(s_0, \dots, s_k)_{\simeq} \\
&\rightarrow^* n(s_0, \dots, s_k)_{\simeq} && \text{decreasing } \phi(n)\text{'s label} \\
&= s|_n
\end{aligned}$$

which concludes the proof. □

Example 4.10. We have $2(9, 7(0, 4))_{\simeq} \preceq 1(3(8(8(5, 1)), 9, 5(9)), 2)_{\simeq}$ since

$$\begin{aligned}
&\underline{1(3(8(8(5, 1)), 9, 5(9)), 2)} \\
&\rightarrow_{\mathcal{E}} 3(\underline{8(8(5, 1))}, 9, 5(9)) && \text{selecting a term} \\
&\rightarrow_{\mathcal{E}} 3(8(5, 1), \underline{9}, 5(9)) && \text{selecting a term} \\
&\rightarrow_{\mathcal{E}} 3(8(5, 1), \underline{5(9)}) && \text{removing a subterm} \\
&\rightarrow_{\mathcal{E}} 3(8(5, 1), 9) && \text{selecting a term} \\
&\simeq 3(9, 8(1, 5)) && \text{commutativity} \\
&\rightarrow_{\mathcal{E}}^{\dagger} 2(9, 7(0, 4)) && \text{decreasing labels (multiple times)}
\end{aligned}$$

Example 4.11 (Non-example). We do not have $1(0, 0)_{\simeq} \preceq 1(0(0, 0))_{\simeq}$.

The following proposition allows us to infer the existence of an IPO reduction sequence $t \rightarrow^* s$ given a tree embedding $s \preceq t$.

Proposition 4.12. For $s, t \in \mathbb{T}(\Sigma, \mathcal{X})$, $s \preceq t \implies t \rightarrow^* s$.

Proof. By Proposition 4.9 it suffices to show that $\rightarrow_{\mathcal{E}}^* \subseteq \rightarrow^*$. This follows from the easily observed fact that IPO can simulate any $\rightarrow_{\mathcal{E}}$ step, i.e., that $\rightarrow_{\mathcal{E}} \subseteq \rightarrow^*$. □

4.3. Proving acyclicity. Our final obligation concerns the acyclicity of \rightarrow^+ on star-free trees. The proof that we will give is a microscopic analysis of how the symbols propagate and have descendants in a \rightarrow^+ reduction. Because of this symbol tracing approach, we will have to “freeze” the trees $\mathbb{T}(\Sigma, \mathcal{X})$ of interest, and thus reason on the level of terms $\mathcal{T}(\Sigma, \mathcal{X})$ rather than trees. The precise nature of this proof reduction is clarified and justified by the following proposition.

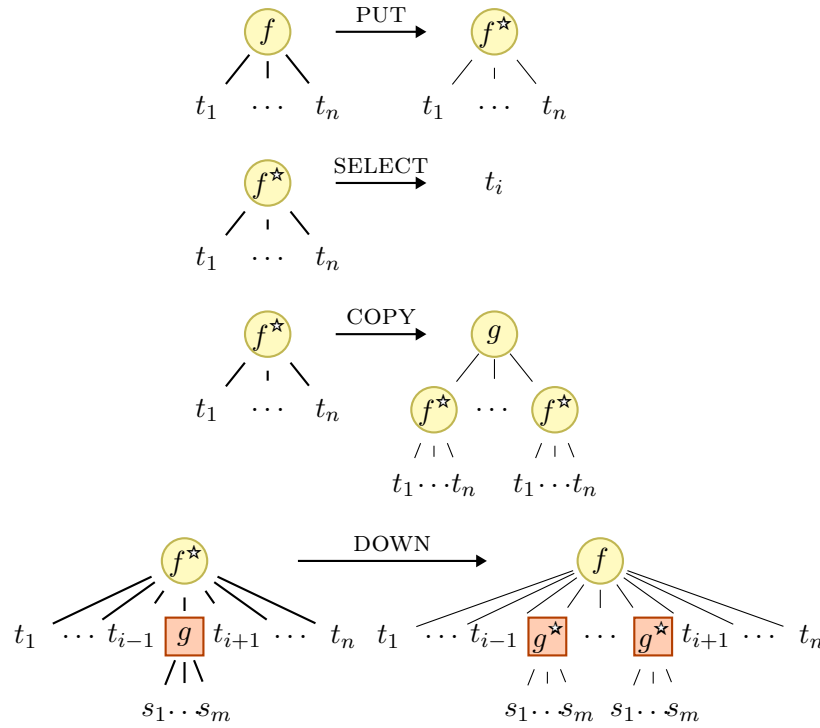


FIGURE 4. Tracing via colors. The symbol occurrences in the right-hand side are descendants of those symbol occurrences in the left-hand side that have the same color. Here red (square) and yellow (round) can be arbitrary colors (they can also be the same color). The subterms $s_1, \dots, s_m, t_1, \dots, t_n$ can contain colored symbol occurrences as well.

Proposition 4.13. *The relation \rightarrow has the following properties:*

- (i) *On $\mathcal{T}(\Sigma, \mathcal{X})$, we have $\rightarrow_{\simeq} \cdot \rightarrow \subseteq \rightarrow \cdot \rightarrow_{\simeq}$.*
- (ii) *If for no term $t \in \mathcal{T}(\Sigma, \mathcal{X})$ we have $t (\rightarrow^+ \cdot \simeq) t$, then for no tree $t' \in \mathbb{T}(\Sigma, \mathcal{X})$ we have $t' \rightarrow^+ t'$.*

Proof.

- (i) It is routine to verify that any swap steps can be postponed.
- (ii) By contradiction. Assume that there is a cycle on trees $t' \rightarrow^+ t'$. Then by the definition of a tree one obtains a sequence on terms $\sigma = t' (\rightarrow_{\simeq}^* \cdot \rightarrow \cdot \rightarrow_{\simeq}^*)^+ t'$. From σ we can construct a sequence $t' (\rightarrow^+ \cdot \rightarrow_{\simeq}^*)^+ t'$ on terms by repeatedly applying (i), contradicting the hypothesis. \square

The tracing can be understood intuitively via coloring. In order to trace a symbol occurrence ρ throughout a reduction, one can color ρ using a unique color c . Then one can apply the colored variant of the rewrite rules as shown in Figure 4. After each rewrite step, the symbol occurrences with color c are the descendants of ρ . This way of proceeding, in which the tracing is expressed via labelling is called *label-tracing* in [Ter03, Section 8.6.4].

Here, we define the descendant relation without the use of colors as follows.

Definition 4.14 (Tracing). *We define in a step $s \rightarrow t$ ($s, t \in \mathcal{T}(\Sigma, \mathcal{X})^\star$) a descendant relation between the occurrences of constant and function symbols of s and t . We will disregard the markers \star . The definition follows the clauses of the definition of \rightarrow .*

- (1) $s = f(\vec{u}) \rightarrow_{\text{PUT}} f^\star(\vec{u}) = t$. *Every symbol occurrence in s has as unique descendant “the same” symbol occurrence in t . (In particular, f^\star descends from f .)*
- (2) $s = f^\star(\vec{u}, v, \vec{w}) \rightarrow_{\text{SELECT}} v = t$. *Every symbol occurrence in v (subterm of s) has as unique descendant “the same” symbol occurrence in $v = t$. The other symbol occurrences of s have no descendant.*
- (3) $s = f^\star(\vec{u}) \rightarrow_{\text{COPY}} g(s, \dots, s) = t$.
 - (a) *Every symbol occurrence in s has as descendants the corresponding symbols in each copy of s in t .*
 - (b) *In addition, the head symbol f^\star of s has the head symbol g of t as descendant. In this case g is called the special descendant of f^\star .*
- (4) $s = f^\star(\vec{u}, v, \vec{w}) \rightarrow_{\text{DOWN}} f(\vec{u}, v^\star, \dots, v^\star, \vec{w}) = t$. *Every symbol occurrence in s has as unique descendant “the same” symbol in t .*
- (5) $s = C[t_1] \rightarrow C[t_2] = t$ with $t_1 \rightarrow t_2$. *The descendants of symbol occurrences in t_1 are given by the clauses above; each symbol occurrence in the context $C[\]$ has as unique descendant “the same” symbol in t .*

In a sequence $t_0 \rightarrow t_1 \rightarrow t_2 \rightarrow \dots \rightarrow t_n$ of several steps ($n \geq 2$) we define the descendant relation between symbol occurrences of t_0 and those of t_n in the obvious way, namely by relational composition. Moreover, if p is a descendant of q , then we call q an ancestor of p .

The descendant relation thus defined has the following three properties that are easily checked.

Lemma 4.15 (Unique ancestors). *Let $s \rightarrow^+ t$, and let q be a symbol occurrence in t . Then q has a unique ancestor in s .*

Lemma 4.16 (Descendants are smaller). *Let p and q be symbol occurrences such that q descends from p . Then $p \geq q$ (in the ordering on S). If one or more steps in the descendant sequence from p to q are special, i.e., are of type Definition 4.14.3(b), then moreover $p > q$.*

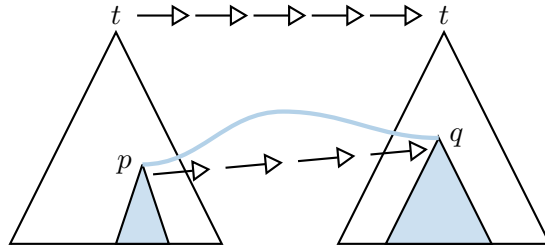


FIGURE 5. Illustration of Lemma 4.17.

The following lemma is illustrated in Figure 5.

Lemma 4.17 (Reductions on subterms). *Let $\alpha : s \rightarrow^+ t$ be a nonempty sequence. Let p be a symbol occurrence in s and q one of its descendants in t . Let $s|_p$ be the subterm of s rooted at symbol occurrence p , and $t|_q$ the subterm of t rooted at symbol occurrence q . Then $s|_p \rightarrow^* t|_q$, in a number of \rightarrow -steps less or equal to the number of steps in the sequence α .*

Proof. This property is easily proved from the particular case of one step $s \rightarrow t$, where for p in s and q in t such that q descends from p , we have $s|_p \rightarrow^= t|_q$. \square

We are now ready to prove acyclicity on terms.

Lemma 4.18 (Acyclicity on terms). *For no term $t \in \mathcal{T}(\Sigma, \mathcal{X})$, $t (\rightarrow^+ \cdot \simeq) t$*

Proof. For a contradiction, suppose there exists a cycle $t (\rightarrow^+ \cdot \simeq) t$ on terms. We call such a term t *cyclic*. Let t be a smallest cyclic term (with respect to the number of symbols). We consider a shortest cycle of t (with respect to the number of steps):

$$\mathcal{C}: t = f(\vec{u}) \rightarrow^+ t' = f(\vec{v}) \simeq t,$$

- (1) Our first observation on the cycle \mathcal{C} is that the head symbol of t' descends from the head symbol of t .

For suppose that the head symbol f of t' descends from a symbol occurrence in t at position $q \neq \epsilon$. Then, by Lemma 4.17, we have $t|_q \rightarrow^* t'$. Since $t \simeq t'$, there is a q' in t' such that $t'|_{q'} \simeq t|_q$, q' not at the root. (That is, $t'|_{q'}$ is a proper subterm of t' .) Therefore $t' \rightarrow^+ t'|_{q'}$ by, e.g., a $(\rightarrow_{\text{PUT}} \cdot \rightarrow_{\text{SELECT}})^+$ sequence at the root. But then we obtain the cycle

$$t|_q \rightarrow^* t' \rightarrow^+ t'|_{q'} \simeq t|_q$$

which contradicts that t that is the smallest cyclic term (Figure 6).

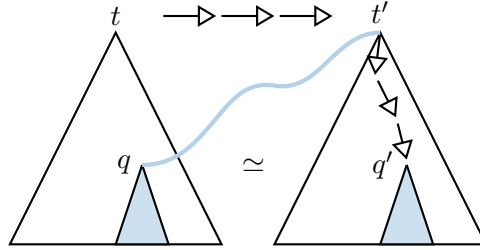


FIGURE 6. Illustration of item (1) in the proof of Proposition 4.18.

- (2) The second observation is that in \mathcal{C} there do not occur steps of type SELECT or COPY at the root.

For suppose there is such a first step in \mathcal{C} . This cannot be a SELECT step, since otherwise the head symbol of t would not have a descendant in t' , contradicting the previous item. So \mathcal{C} must be of the form

$$t = f(\vec{u}) \rightarrow_{\alpha}^+ f^{\star}(\vec{w}) = s \rightarrow_{\text{COPY}} g(s, \dots, s) \rightarrow_{\beta}^* t' \simeq t$$

for some $g < f$, where in part α no SELECT or COPY steps occur at the root. (Observe that $f(\vec{u}) \rightarrow_{\alpha}^+ f^{\star}(\vec{w})$ is nonempty because f must become starred for a COPY step to occur.)

Now the only descendants in $g(s, \dots, s)$ of the head symbol f of t are the head symbol g and the head symbols of the arguments s, \dots, s . By item (1) and the fact that the descendant relation is defined through relational composition, one of these symbols must be the ancestor of the head symbol f of t' . By Lemma 4.16 and $g < f$, this ancestor cannot be g . Thus, it must be the head symbol f of one of the arguments s, \dots, s . By

Lemma 4.17 we then obtain a sequence $s \rightarrow_{\gamma}^* t'$ of length not exceeding the length of part β . Omitting the first COPY step, it follows that

$$t = f(\vec{w}) \rightarrow_{\alpha}^+ f^{\star}(\vec{w}) = s \rightarrow_{\gamma}^* t' \simeq t$$

is a cycle on t shorter than \mathcal{C} , contradicting \mathcal{C} 's minimality.

- (3) Let $\text{DOWN}(T, k)$ denote a DOWN step at a root in \mathcal{C} which creates k starred copies of a subtree T . Our final observation is that any step at a root in \mathcal{C} is either a PUT step or a $\text{DOWN}(k)$ step with $k > 1$, and that there is at least one such a DOWN step. From this it follows that the root of t' has more subterms than the root of t , contradicting $t \simeq t'$, and concluding the proof.

The final observation follows from the following three items:

- (a) At least one DOWN step must occur in \mathcal{C} . Otherwise, it follows from item (2) that all steps in \mathcal{C} happen below the root, implying the existence of a cyclic term smaller than t .
- (b) No $\text{DOWN}(T, 1)$ step takes place. For the step and its corresponding PUT step at the root can be replaced with a single PUT step on the root of T , yielding a cycle smaller than \mathcal{C} .
- (c) No $\text{DOWN}(T, 0)$ step takes place. For a contradiction, suppose otherwise. Since no SELECT and COPY steps take place at the root, T was either originally under the head symbol of t , or it arose due to some earlier $\text{DOWN}(T', k)$ step with $k > 1$ and T a reduct of $(T')^{\star}$. In either case, we know that no local rewrite steps have since taken place in T , since omitting these steps would yield a cycle shorter than \mathcal{C} . In the first case, T can be pruned from the initial term t , yielding a smaller cyclic term. More formally, we have

$$t = f(\vec{w}, T, \vec{x}) \rightarrow^+ f^{\star}(\vec{y}, T, \vec{z}) \rightarrow_{\text{DOWN}(T,0)} f(\vec{y}, \vec{z}) \rightarrow^* t' \simeq t,$$

from which we can construct a sequence

$$f(\vec{w}, \vec{x}) \rightarrow^* f(\vec{y}, \vec{z}) \rightarrow^* t' \simeq t = f(\vec{w}, T, \vec{x}) \rightarrow^+ f(\vec{w}, \vec{x})$$

with $f(\vec{w}, \vec{x})$ the cyclic term smaller than t (postponing the \simeq step).

In the second case, we can make one less copy of T' in the earlier $\text{DOWN}(T', k)$ step, allowing us to construct a cycle shorter than \mathcal{C} by omitting the $\text{DOWN}(T, 0)$ step. More formally, we have

$$t \rightarrow^+ \cdot \rightarrow_{\text{DOWN}(T',k)} f(\vec{w}, (T')^{\star}, \vec{x}) \rightarrow_{\alpha}^+ f^{\star}(\vec{y}, T, \vec{z}) \rightarrow_{\text{DOWN}(T,0)} f(\vec{y}, \vec{z}) \rightarrow^* t' \simeq t$$

from which we can construct a shorter cycle

$$t \rightarrow^+ \cdot \rightarrow_{\text{DOWN}(T',k-1)} f(\vec{w}, \vec{x}) \rightarrow_{\beta}^* f(\vec{y}, \vec{z}) \rightarrow^* t' \simeq t$$

where part β is shorter than α . □

This ends the proof of acyclicity of the iterative path ordering on terms $\mathcal{T}(\Sigma, \mathcal{X})$.

Corollary 4.19 (Acyclicity on trees). *For no tree $t \in \mathbb{T}(\Sigma, \mathcal{X})$ we have $t \rightarrow^+ t$.*

Proof. Immediate from Proposition 4.13(ii) and Lemma 4.18. □

We now have all the ingredients required for proving Theorem 3.7, the claim that \rightarrow^+ restricted to unstarred trees is SN.

Proof of Theorem 3.7. Follows from the basic Proposition 4.1, Kruskal’s tree theorem (Theorem 4.4), Proposition 4.12 and Corollary 4.19. \square

5. THE SECOND PROOF: A MAGICAL PROOF

We now give an alternative proof of Theorem 3.7, by employing an ingenious proof technique due to Buchholz [Buc95] which uses neither Kruskal’s tree theorem nor acyclicity. Throughout this section, $(\Sigma, <)$ is a well-founded partial order.⁴

As we have seen, the star game introduced and studied so far has the peculiarity that it is terminating when restricted to the intended, proper objects, being finite trees with nodes labeled with natural numbers or ordinals, but without presence of the stars as “control agents”. On the auxiliary trees that are equipped with the control stars, we do not have termination, as there may be cycles and spiraling reductions, generating a proper context of the starting tree.

An interesting subsequent step was made by Vincent van Oostrom in [KvOdV06], who noticed that by making the control stars *quantitative*, as if having a certain amount of working energy, conceived as a natural number or ordinal, this peculiarity can be made to disappear, and the auxiliary objects are themselves already terminating. Moreover, the two reductions, with simple stars or with their numerical refinements, are coinciding on the proper objects, trees without control symbols. Figure 1 presents a helicopter view of this situation.⁵

We now state the rules of the number star game.

Definition 5.1 (Energized signature). *Let f^n be a fresh symbol for every $f \in \Sigma$ and $n \in \mathbb{N}$. We define $\Sigma^\omega = \{f^n \mid f \in \Sigma, n \in \mathbb{N}\}$ as the energized signature.*

Notation 5.2. *We abbreviate $\mathbb{T}(\Sigma \cup \Sigma^\omega, \mathcal{X})$ as $\mathbb{T}(\Sigma, \mathcal{X})^\omega$.*

Definition 5.3 (Stars with specified energy). *On $\mathbb{T}(\Sigma, \mathcal{X})^\omega$ we define the rewriting system $Star^\omega$, which induces the reduction relation \rightarrow , as follows:*

$$\begin{aligned}
 (\text{PUT}) \quad & f(\vec{s}) \rightarrow f^n(\vec{s}) \\
 (\text{SELECT}) \quad & f^{n+1}(\vec{s}, t, \vec{u}) \rightarrow t \\
 (\text{COPY}) \quad & f^{n+1}(\vec{s}) \rightarrow g(\underbrace{f^n(\vec{s}), \dots, f^n(\vec{s})}_{0 \text{ or more}}) \quad \text{if } f > g \\
 (\text{DOWN}) \quad & f^{n+1}(\vec{s}, t, \vec{u}) \rightarrow f(\vec{s}, \underbrace{t^n, \dots, t^n}_{0 \text{ or more}}, \vec{u})
 \end{aligned}$$

for every every $n \in \mathbb{N}$ and $f, g \in \Sigma$.

⁴Strictly speaking even transitivity is not needed, an arbitrary well-founded relation suffices. However, without loss of generality, we may assume that $<$ is transitive, for otherwise we can work with $<^+$.

⁵In [KvOdV06] the present second author together with his co-authors chose to focus on the iterative *lexicographic* path order among the material in draft form available at the time. That Buchholz’ proof method could also be adapted to prove termination of related orders such as the iterative *recursive* path order was not fleshed out and ended up only as a claim in the conclusion of [KvOdV06]. We prove the claim applies to IPO (Definition 3.3) as employed here.

A star \star can be seen as an unspecified amount of “energy” while a label n is a precise quantity of energy. We can lift every finite reduction in $Star$ to one in $Star^\omega$ by replacing the stars by energy quanta that are large enough to make the journey all the way. As the energy decreases by at most 1 in every \rightarrow step, the length of the (remaining) reduction bounds the amount of energy that we need.

Proposition 5.4. *The relations \rightarrow^+ and \rightarrow^+ coincide on unmarked trees $\mathbb{T}(\Sigma, \mathcal{X})$. \square*

Example 5.5. *We reconsider Example 3.9 with the single rule $1(0(x)) \rightarrow 0(0(1(x)))$ and the order $1 > 0$ on the signature. In $Star^\omega$ we have*

$$\begin{array}{ll} 1(0(x))_{\simeq} & \rightarrow_{\text{PUT}} 1^4(0(x))_{\simeq} \\ & \rightarrow_{\text{COPY}} 0(1^3(0(x)))_{\simeq} \\ & \rightarrow_{\text{COPY}} 0(0(1^2(0(x))))_{\simeq} \\ & \rightarrow_{\text{DOWN}} 0(0(1(0^1(x))))_{\simeq} \\ & \rightarrow_{\text{SELECT}} 0(0(1(x)))_{\simeq} \end{array}$$

Note that the energy decreases in each step.

We now prove the main theorem, termination of \rightarrow , by employing a proof technique due to Buchholz [Buc95]. As mentioned before, the proof technique was adapted by Vincent van Oostrom, and is included in [KvOdV06]. In the following proof we closely follow the exposition given by van Oostrom in [KvOdV06].

Proposition 5.6. *\rightarrow is terminating.*

Proof. We write $f^\omega(\vec{t})$ to denote the tree $f(\vec{t})$ for an *unmarked* symbol f . This allows us to write *any* tree from $\mathbb{T}(\Sigma, \mathcal{X})^\omega$ uniquely in the form $f^\alpha(\vec{t})$ for some unmarked symbol f , vector of trees \vec{t} and ordinal $\alpha \leq \omega$. In the crucial induction in this proof we will make use of the fact that in the ordering of the ordinals we have $\omega > n$ for each natural number n .

We prove by induction on the construction of trees that any ω -marked tree is terminating. To be precise, we show that any ω -marked tree $f^\alpha(\vec{t})$ is terminating under the assumption (the induction hypothesis) that its arguments \vec{t} are terminating:

$$t_1, \dots, t_n \text{ terminating} \implies f^\alpha(t_1, \dots, t_n) \text{ terminating}$$

for every $f \in \Sigma$ and $\alpha \leq \omega$. We prove this implication by induction on the triple

$$(f, \{\{t_1, \dots, t_n\}\}, \alpha)$$

in the lexicographic order with respect to

- (1) $>$ on the symbols of Σ ,
- (2) the multiset extension $\rightarrow_\#$ of \rightarrow , and
- (3) $>$ on the markers in $\mathbb{N} \cup \{\omega\}$.

Note that $\rightarrow_\#$ is well-founded on multisets that contain only terminating terms.

Clearly, the tree $f^\alpha(\vec{t})$ is terminating if all its one-step reducts are. We prove that the one-step reducts are terminating by distinguishing cases on the type of the reduction step:

- (1) If the step occurs at the root, we perform a further case analysis on the rule applied:
 - (PUT) The first and second component of the triple remain the same. The third component decreases from ω to some $m \in \mathbb{N}$. Thus by the IH, the reduct is terminating.
 - (SELECT) The result follows by the termination assumption for the terms \vec{t} .

(COPY) Then α is of the form $m + 1$ for some $m \in \mathbb{N}$, and the reduct has shape $g^\omega(f^m(\vec{t}), \dots, f^m(\vec{t}))$ for some g such that $f > g$. By the IH for the third component, each of the $f^m(\vec{t})$ is terminating. Hence, by the IH for the first component, the reduct is terminating.

(DOWN) Then α is of the form $m + 1$ for some $m \in \mathbb{N}$, and the reduct is of the form

$$f^\omega(t_1, \dots, t_{i-1}, \underbrace{t_i^m, \dots, t_i^m}_{0 \text{ or more}}, t_{i+1}, \dots, t_k)$$

with $t_i = g^\omega(\vec{s})$ for some trees \vec{s} , and $t_i^m = g^m(\vec{s})$. Each t_1, \dots, t_k is terminating by assumption, and t_i^m is terminating since it is a reduct of t_i . Hence, the reduct is terminating by the IH for the second component.

(2) If the step is a non-head step, then it rewrites some subterm, and the result follows by the IH for the second component. \square

6. HYDRAS

A (mathematical) Hydra is a tree, in which the leaves resemble the Hydra's heads, and the root resembles its body. The objective is to slay the Hydra by chopping off its heads (i.e., removing its leaves) one by one. Like its mythological counterpart, the Hydra may – subject to certain conditions – grow new heads in response to each chop. *Will the Hydra eventually be slain, no matter which strategy we choose?*

Multiple versions of the Hydra exist. One Hydra of particular interest is the *Kirby-Paris (KP) Hydra* [KP82]. It was used by Kirby and Paris to derive a mathematical truth that cannot be proved in Peano Arithmetic (namely, the fact there is no losing strategy against this Hydra). Such independence results started with Gödel's famous theorem; but the endeavour later on was to find independence results with a more “mathematical” content, referring to mathematical objects such as words and trees rather than logical proof systems. (Of course, for us, the latter are also mathematical objects, but maybe not in the eyes of every mathematician.)

Definition 6.1 (Kirby-Paris Hydra). *A Kirby-Paris (KP) Hydra is a finite tree with nodes labeled from $S = \{0, \dagger\}$. The label \dagger labels only the root. Its rewrite relation \rightarrow_{KP} is defined by the TRS consisting of rule schemas*

$$\begin{aligned} \text{(KP}_1\text{)} \quad & \dagger(\vec{t}, \mathbf{0}) \rightarrow \dagger(\vec{t}) \\ \text{(KP}_2\text{)} \quad & f(\vec{t}_1, 0(\vec{t}_2, \mathbf{0})) \rightarrow f(\vec{t}_1, \underbrace{0(\vec{t}_2), \dots, 0(\vec{t}_2)}_{k \in \mathbb{N} \text{ times}}) \end{aligned}$$

for any $f \in S$. In these rules, a boldfaced $\mathbf{0}$ highlights the head 0 that is “chopped off” by the rule.

Remark 6.2. *The Kirby-Paris Hydra is usually described as an unlabeled Hydra – or equivalently – a Hydra labeled from a singleton label set. In this modeling, rule KP_1 is of the form $0(\vec{t}, \mathbf{0}) \rightarrow 0(\vec{t})$, and its applicability must be explicitly restricted to the root.*

Example 6.3 (Slaying a KP Hydra). *One battle with the KP Hydra $\dagger(0(0,0))$ may progress as follows:*

$$\begin{aligned}
\dagger(0(0, \mathbf{0})) &\rightarrow_{KP} \dagger(0(\mathbf{0}), 0(0)) \\
&\rightarrow_{KP} \dagger(0, \mathbf{0}, 0(0)) \\
&\rightarrow_{KP} \dagger(\mathbf{0}, 0(0)) \\
&\rightarrow_{KP} \dagger(0(\mathbf{0})) \\
&\rightarrow_{KP} \dagger(0, \mathbf{0}) \\
&\rightarrow_{KP} \dagger(\mathbf{0}) \\
&\rightarrow_{KP} \dagger
\end{aligned}$$

The KP Hydra of Example 6.3 started out quite small, and made only two copies after each cut not directly below the root. But it is easy to see that the Hydra can grow explosively large even for small starting examples.

Similar to Example 3.9, we can use *Star* to prove that the Kirby-Paris Hydra is eventually slain.

Theorem 6.4. *Any Kirby-Paris Hydra is eventually slain, no matter the strategy. In other words, \rightarrow_{KP} is terminating.*

Proof. For KP_1 we have:

$$\dagger(\vec{t}, 0) \rightarrow_{\text{PUT}} \dagger^{\star}(\vec{t}, 0) \rightarrow_{\text{DOWN}} \dagger(\vec{t})$$

and for rule KP_2 we have (see also Figure 7):

$$\begin{aligned}
x(\vec{t}_1, 0(\vec{t}_2, 0)) \rightarrow_{\text{PUT}} x^{\star}(\vec{t}_1, 0(\vec{t}_2, 0)) &\rightarrow_{\text{DOWN}} x(\vec{t}_1, \underbrace{0^{\star}(\vec{t}_2, 0), \dots, 0^{\star}(\vec{t}_2, 0)}_{k \in \mathbb{N} \text{ times}}) \\
&\rightarrow_{\text{DOWN}}^* x(\vec{t}_1, \underbrace{0(\vec{t}_2), \dots, 0(\vec{t}_2)}_{k \in \mathbb{N} \text{ times}})
\end{aligned}$$

Hence $\rightarrow_{KP} \subseteq \rightarrow^+$, so that termination of \rightarrow_{KP} follows from Corollary 3.8. \square

There exist even more intimidating Hydras. A well-known one is the Buchholz Hydra, in which non-root nodes are labeled with ordinals $\alpha \leq \omega$ rather than 0. For heads labeled with $\alpha = 0$, the Buchholz Hydra acts much like the Kirby-Paris Hydra. More interestingly, for heads labeled with $0 < \alpha < \omega$, the Hydra shifts down a (slightly modified) copy of a supertree towards the chopping site. This allows it to not only grow in width, but also in height. The precise definition of the Buchholz Hydra contains a number of subtleties, given below.

Definition 6.5 (Buchholz Hydra [Buc87]). *A Buchholz Hydra (BH) is a finite tree with nodes labeled from $L = \mathbb{N} \cup \{\omega, \dagger\}$, where*

- (1) \dagger labels only the root,
- (2) every node directly below the root is labeled 0, and
- (3) every remaining node is labeled with an ordinal $\alpha \leq \omega$.

Its rewrite relation \rightarrow_{BH} can be defined as a TRS as induced by the following three rule schemas.

The first rule schema is a generalization of the Kirby-Paris rule schema KP_2 :

$$(\text{BH}_1) \quad x(\vec{t}_1, \alpha(\vec{t}_2, \mathbf{0})) \rightarrow x(\vec{t}_1, \underbrace{\alpha(\vec{t}_2), \dots, \alpha(\vec{t}_2)}_{k \in \mathbb{N} \text{ times}})$$

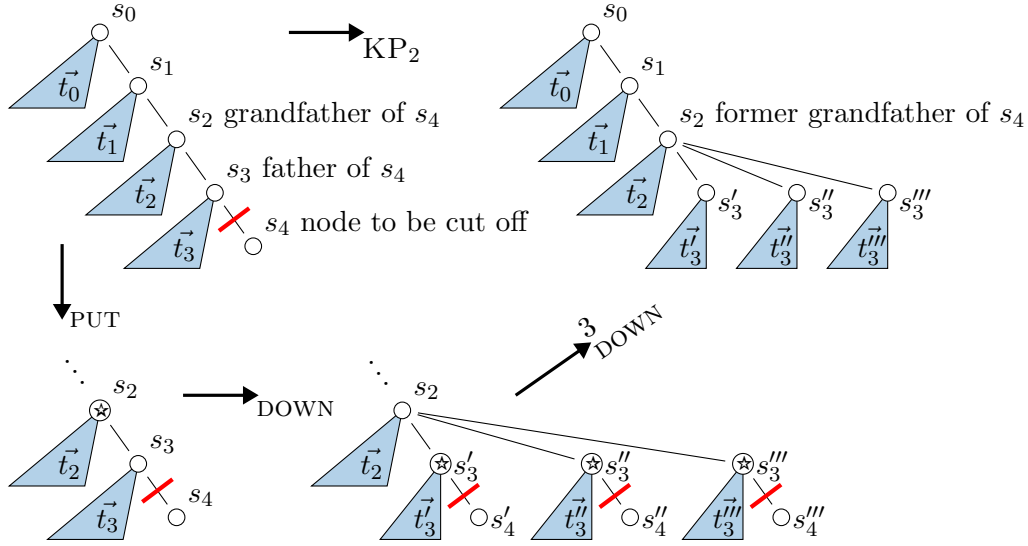


FIGURE 7. Simulating rule KP_2 by star rules. All nodes of the trees have label 0.

for any $x, \alpha \in L$.

The second rule is more intricate. For a targeted leaf node x labeled with a successor ordinal $0 < \alpha < \omega$, find the closest ancestor y such that $\text{label}(y) \leq \alpha$. (By condition (2) above, such an ancestor exists.) Now copy the subtree T rooted at y . In T , set $\text{label}(y) = \alpha - 1$, and set $\text{label}(x) = 0$. Call the result of these two changes T' . Then the second rule is given by

$$(BH_2) \quad z(\vec{t}, \alpha) \rightarrow z(\vec{t}, T')$$

for any $z \in L$.

The third rule, finally, simply relabels a targeted leaf node with label ω to an arbitrary $k \in \mathbb{N}$, i.e.,

$$(BH_3) \quad z(\vec{t}, \omega) \rightarrow z(\vec{t}, k) \quad \text{for } k \in \mathbb{N}$$

for any $z \in L$.

Example 6.6 (Example application of BH_2). Cutting off the boldfaced head in the Buchholz Hydra $\dagger(0(\omega), 0(2, 7(\mathbf{5})))$ results in the Hydra $\dagger(0(\omega), 0(2, 7(T')))$ with $T' = 4(2, 7(0))$.

Buchholz [Buc95] has shown that every Buchholz Hydra can be slain by always chopping the rightmost head.

Theorem 6.7. \rightarrow_{BH} is weakly normalizing.

We conjecture that \rightarrow_{BH} is even strongly normalizing. But even if this is correct, IPO cannot be used to prove this. IPO can simulate rules BH_1 and BH_3 , respectively, by the argument given in the proof to Theorem 6.4, and the fact that IPO can decrease labels. But rule BH_2 cannot generally be simulated, as the proof to the following lemma demonstrates.

Lemma 6.8 (IPO cannot simulate BH). $\rightarrow_{BH} \not\subseteq \rightarrow^*$.

Proof. Assume by contradiction that $\rightarrow_{BH} \subseteq \rightarrow^*$. Then in particular, the BH_2 step $\dagger(0(3(\mathbf{2}))) \rightarrow_{BH} \dagger(0(3(1(3(0))))$ implies the existence of a sequence

$$\begin{array}{ccc} \dagger(0(3(\mathbf{2}))) & \rightarrow^* & \dagger(0(3(1(3(0)))) \\ & \rightarrow_{\text{PUT}} \cdot \rightarrow_{\text{SELECT}} & \dagger(0(3(3(0)))) \\ & \rightarrow_{\text{PUT}} & \dagger(0(3(3^*(0)))) \\ & \rightarrow_{\text{COPY}} & \dagger(0(3(\mathbf{2}))), \end{array}$$

which is a \rightarrow^+ cycle on unmarked trees, contradicting Lemma 4.19. \square

Remark 6.9. *The proof to Lemma 6.8 shows something stronger: any pursuit for a conservative extension of IPO that allows one to simulate the Buchholz Hydra is fruitless. For any such solution results in cyclicity, and therefore non-termination, of \rightarrow^+ on unmarked trees, defeating the purpose of defining \rightarrow in the first place.*

The heart of the problem is that, under minimal conditions, the Buchholz Hydra allows replacing a leaf x in a tree T with a tree T' , where T' contains a subtree Y larger than x . Subsequently, subtree Y can be installed on the position of x using repeated applications of rule SELECT, producing a tree T'' larger than the starting tree T .

Note that the Kirby-Paris Hydra can grow only in width, and never in height, whereas the Buchholz Hydra can grow both in width and in height. We now define an alternative powerful Hydra which can be labeled with any well-ordered set, and which can grow both in width and in height, but which – unlike the Buchholz Hydra – can be proven terminating using IPO. The underlying idea is that a chop can be regarded as prefulfilling an “obligation” to make its supertrees smaller: hence the roots of the supertrees receive “permission” (denoted by underlining) to modify themselves in accordance with the rules of *Star*. For a labeled tree $t = f(\vec{u})$, we write \underline{t} to denote the tree $\underline{f}(\vec{u})$ obtained from t by underlining the root symbol.

Definition 6.10 (Star Hydra). *A Star Hydra (SH) is a finite tree with nodes labeled from a well-founded set $(S, <)$. For every $s \in S$ let \underline{s} be a fresh symbol, and define $\underline{S} = \{\underline{s} \mid s \in S\}$. On finite trees labeled with $S \cup \underline{S}$ we define the relation \rightarrow by the term rewrite system:*

$$\begin{array}{lll} \text{(CHOP)} & n(\alpha, \vec{x}) & \rightarrow \underline{n}(\beta_1, \dots, \beta_k, \vec{x}) \quad k \geq 0, \beta_1, \dots, \beta_k < \alpha \\ \text{(PROPAGATE)} & n(\underline{m}(\vec{x}), \vec{y}) & \rightarrow \underline{n}(\underline{m}(\vec{x}), \vec{y}) \\ \text{(WIDEN)} & \underline{n}(\underline{m}(\vec{x}), \vec{y}) & \rightarrow \underline{n}(\underbrace{\underline{m}(\vec{x}), \dots, \underline{m}(\vec{x})}_{0 \text{ or more}}, \vec{y}) \\ \text{(LENGTHEN)} & \underline{n}(\vec{x}) & \rightarrow \underline{m}(\underline{n}(\vec{x})) \quad m < n \\ \text{(WAIVE)} & \underline{m}(\vec{x}) & \rightarrow m(\vec{x}) \end{array}$$

The Star Hydra rewrite relation \rightarrow_{SH} is defined as the relation

$$\rightarrow_{\text{CHOP}} \cdot \rightarrow_{\text{PROPAGATE}}^! \cdot (\rightarrow_{\text{WIDEN}} \cup \rightarrow_{\text{LENGTHEN}})^* \cdot \rightarrow_{\text{WAIVE}}^!$$

where $R^!$ denotes a maximal reduction of R , i.e., $x R^! y$ if and only if $x R^* y$ and there exists no z such that $y R z$.

Intuitively, the Star Hydra can be described as follows. When Hercules chops off a head, the Hydra enrages, possibly regrowing any smaller number of heads at the wound ($\rightarrow_{\text{CHOP}}$). In addition, it seeks to strengthen the neck from the body to the wound ($\rightarrow_{\text{PROPAGATE}}^!$). It does this by repeatedly lengthening and duplicating subtrees along the neck, in any order

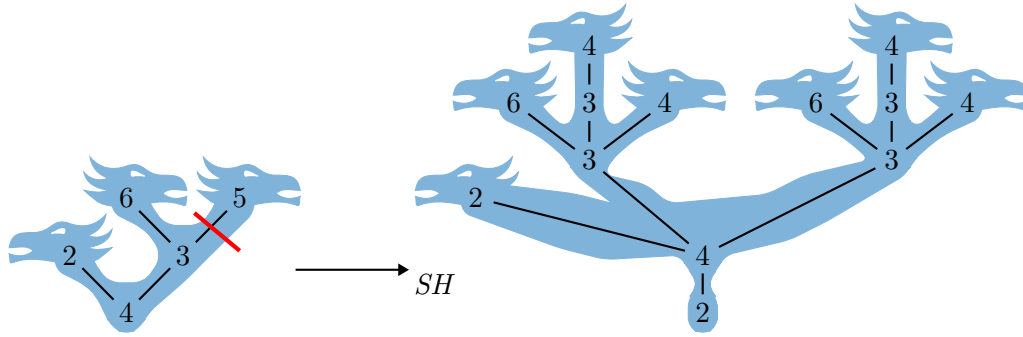


FIGURE 8. One step in the process of killing the Star Hydra. After chopping the head with value 5, the Hydra grows both in width and in height, and Hercules starts running.

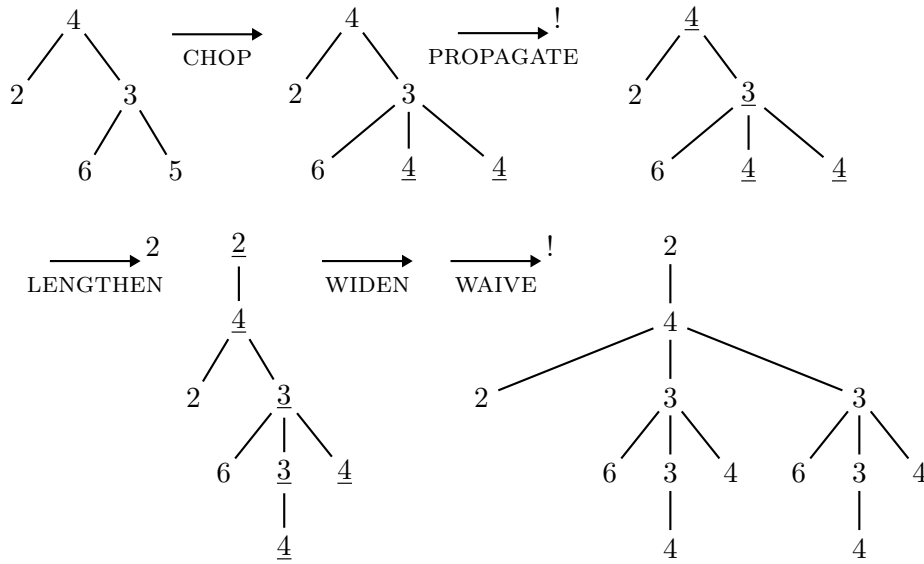


FIGURE 9. Justification of the \rightarrow_{SH} step in Figure 8, detailing the change of the Hydra via the steps in Definition 6.10.

$((\rightarrow_{WIDEN} \cup \rightarrow_{LENGTHEN})^*)$. Eventually it calms down and stabilizes $(\rightarrow_{WAIVE}^!)$, opening up another opportunity for Hercules to strike. An example \rightarrow_{SH} step is given in Figure 8.

In order to prove termination of \rightarrow_{SH} we show that $\rightarrow_{SH} \subseteq \rightarrow^+$. For this purpose, we show that the underlined reduction introduced in Definition 6.10, in particular the rules WIDEN and LENGTHEN, can be simulated by \rightarrow^+ . As a first approximation, these rules can be simulated by

$$\begin{aligned}
 n(m(\vec{x}), \vec{u}) &\rightarrow_{\text{PUT}} n^*(m(\vec{x}), \vec{u}) \\
 &\rightarrow_{\text{DOWN}} n(m^*(\vec{x}), \dots, m^*(\vec{x}), \vec{u})
 \end{aligned}$$

and

$$\begin{aligned} n(\vec{x}) &\rightarrow_{\text{PUT}} n^\star(t, \vec{x}) \\ &\rightarrow_{\text{COPY}} m(n^\star(\vec{x})), \end{aligned}$$

respectively. However, there is one problem: we need to get rid of the stars \star after the \rightarrow simulation. The idea is to enrich the trees such that every underlined symbol has additional “garbage” arguments. A garbage argument *garbage* can then be used to eliminate a \star symbol as follows:

$$n^\star(\vec{t}, \text{garbage}) \rightarrow_{\text{DOWN}} n(\vec{t})$$

To this end, we introduce the notion of garbage collection.

For convenience, when proving $\rightarrow_{SH} \subseteq \rightarrow^+$, we consider \rightarrow over an extended signature

$$S' = S \cup \{\perp\}$$

where \perp is a fresh element that is smaller than all elements in S . So \perp is an auxiliary element that is not used in the Hydra reduction \rightarrow_{SH} .

Definition 6.11 (Garbage collection). *For every $g \geq 1$ we define \hookrightarrow_g on $\mathbb{T}(S' \cup \underline{S'}, \mathcal{X})$ by the following rewrite system*

$$n(\vec{x}, y_1, \dots, y_h) \hookrightarrow_g \underline{n}(\vec{x}) \quad \text{for every } h \geq g$$

Definition 6.12 (Dropping underlining). *For $s \in \mathbb{T}(S' \cup \underline{S'}, \mathcal{X})$ we write $[s]$ for the tree obtained from s by removing underlining, that is, replacing each symbol occurrence \underline{n} by n .*

Garbage collection can be handled by \rightarrow in the following sense.

Lemma 6.13. *If $s \in \mathbb{T}(S', \mathcal{X})$ and $s \hookrightarrow_g^* t$, then $s \rightarrow^* [t]$.*

Proof. Since $s \hookrightarrow_g^* t$, the term $[t]$ is obtained from s by removing at least g subtrees. This can be simulated by repeated applications of $\rightarrow_{\text{PUT}} \cdot \rightarrow_{\text{DOWN}}$. \square

The following lemma states that \rightarrow^* can simulate the chopping of a head and insert garbage arguments along the affected neck (the underlined symbol occurrence).

Lemma 6.14. *If $g \geq 1$, $s \in \mathbb{T}(S, \mathcal{X})$ and $s \rightarrow_{\text{CHOP}} \cdot \rightarrow_{\text{PROPAGATE}}^! t$, then $s \rightarrow s^\star \rightarrow^* v \hookrightarrow_g^* t$ for some $v \in \mathbb{T}(S', \mathcal{X})$.*

Proof. We prove the claim by induction on the depth of the CHOP step:

(1) If the CHOP step occurs at the root, then it is of the form

$$s = n(\alpha, \vec{u}) \rightarrow_{\text{CHOP}} \underline{n}(\beta_1, \dots, \beta_k, \vec{u}) = t$$

with $k \geq 0$, $\beta_1, \dots, \beta_k < \alpha$. We then reduce as follows:

$$\begin{aligned} s = n(\alpha, \vec{u}) &\rightarrow_{\text{PUT}} n^\star(\alpha, \vec{u}) = s^\star \\ &\rightarrow_{\text{DOWN}} \underbrace{n(\alpha^\star, \dots, \alpha^\star, \vec{u})}_{k+g \text{ times}} \\ &\rightarrow_{\text{COPY}}^* \underbrace{n(\beta_1(\perp, \dots, \perp), \dots, \beta_k(\perp, \dots, \perp), \perp, \dots, \perp, \vec{u})}_{g \text{ times}} = v \\ &\hookrightarrow_g^* \underline{n}(\beta_1, \dots, \beta_k, \vec{u}) = t \end{aligned}$$

(2) Let $s = n(s', \vec{u})$ and assume that the CHOP step occurs in s' . Then

$$s' \rightarrow_{\text{CHOP}} \cdot \rightarrow_{\text{PROPAGATE}}^! t' \quad \text{and} \quad n(t', \vec{u}) \rightarrow_{\text{PROPAGATE}} \underline{n}(t', \vec{u}) = t$$

for some t' . By the induction hypothesis $s' \rightarrow \triangleright s'^{\star} \rightarrow^* v' \hookrightarrow_g^* t'$ for some $v' \in \mathbb{T}(S', \mathcal{X})$. Hence

$$\begin{aligned} n(s', \vec{u}) &\rightarrow_{\text{PUT}} n^{\star}(s', \vec{u}) \\ &\rightarrow_{\text{DOWN}} n(\underbrace{s'^{\star}, \dots, s'^{\star}}_{g+1 \text{ times}}, \vec{u}) \\ &\rightarrow^* n(\underbrace{v', \dots, v'}_{g+1 \text{ times}}, \vec{u}) = v \\ &\hookrightarrow_g^* \underline{n}(t', \vec{u}) = t \end{aligned}$$

This concludes the proof. \square

Lemma 6.15. *If $s \in \mathbb{T}(S', \mathcal{X})$ and $s \hookrightarrow_{g+1}^* \cdot \rightarrow_{\text{WIDEN}} u$ for $g \geq 1$, then $s \rightarrow^* v \hookrightarrow_g^* u$ for some $v \in \mathbb{T}(S', \mathcal{X})$.*

Proof. Let $s \in \mathbb{T}(S', \mathcal{X})$ and $s \hookrightarrow_{g+1}^* t \rightarrow_{\text{WIDEN}} u$ with $g \geq 1$. W.l.o.g. we may assume that the WIDEN step occurs at the root (otherwise we consider the relevant subtrees of s, t, u). Then the rewrite sequence is of the form

$$s \hookrightarrow_{g+1}^* t = \underline{n}(\underline{m}(\vec{v}), \vec{w}) \rightarrow_{\text{WIDEN}} \underline{n}(\underline{m}(\vec{v}), \dots, \underline{m}(\vec{v}), \vec{w}) = u$$

for some $n, m \in S'$ and $\vec{v}, \vec{w} \in \mathbb{T}(S' \cup \underline{S}', \mathcal{X})^*$. As a consequence s must be of the form

$$s = n(m(\vec{v}', x_1, \dots, x_{g_1+1}), \vec{w}', y_1, \dots, y_{g_2+1})$$

with $g_1, g_2 \geq g$ and componentwise $\vec{v}' \hookrightarrow_{g+1}^* \vec{v}$ and $\vec{w}' \hookrightarrow_{g+1}^* \vec{w}$. Then

$$\begin{aligned} s &= n(m(\vec{v}', x_1, \dots, x_{g_1+1}), \vec{w}', \vec{y}) \\ &\rightarrow_{\text{PUT}} n^{\star}(m(\vec{v}', x_1, \dots, x_{g_1+1}), \vec{w}', \vec{y}) \\ &\rightarrow_{\text{DOWN}} n(m^{\star}(\vec{v}', x_1, \dots, x_{g_1+1}), \dots, m^{\star}(\vec{v}', x_1, \dots, x_{g_1+1}), \vec{w}', \vec{y}) \\ &\rightarrow_{\text{DOWN}}^* n(m(\vec{v}', x_1, \dots, x_{g_1}), \dots, m(\vec{v}', x_1, \dots, x_{g_1}), \vec{w}', \vec{y}) = v \\ &\hookrightarrow_g^* \underline{n}(\underline{m}(\vec{v}), \dots, \underline{m}(\vec{v}), \vec{w}) \\ &= u \end{aligned}$$

This proves the claim. \square

Lemma 6.16. *If $s \in \mathbb{T}(S', \mathcal{X})$ and $s \hookrightarrow_{g+1}^* \cdot \rightarrow_{\text{LENGTHEN}} u$ for $g \geq 1$, then $s \rightarrow^* v \hookrightarrow_g^* u$ for some $v \in \mathbb{T}(S', \mathcal{X})$.*

Proof. Let $s \in \mathbb{T}(S', \mathcal{X})$ and $s \hookrightarrow_{g+1}^* t \rightarrow_{\text{LENGTHEN}} u$ with $g \geq 1$. Assume that the LENGTHEN step occurs at the root (otherwise we consider the relevant subtrees of s, t, u). The rewrite sequence is of the form

$$s \hookrightarrow_{g+1}^* t = \underline{n}(\vec{w}) \rightarrow_{\text{LENGTHEN}} \underline{m}(\underline{n}(\vec{w})) = u$$

for some $n > m \in S'$ and $\vec{w} \in \mathbb{T}(S' \cup \underline{S}', \mathcal{X})^*$. Hence s must be of the form

$$s = n(\vec{w}', x_1, \dots, x_{g'+1})$$

with $g' \geq g$ and componentwise $\vec{w}' \hookrightarrow_{g+1}^* \vec{w}$. Then

$$\begin{aligned}
s &= n(\vec{w}', x_1, \dots, x_{g'+1}) \\
&\rightarrow_{\text{PUT}} n^\star(\vec{w}', x_1, \dots, x_{g'+1}) \\
&\rightarrow_{\text{COPY}} \underbrace{m(n^\star(\vec{w}', x_1, \dots, x_{g'+1}), \dots, n^\star(\vec{w}', x_1, \dots, x_{g'+1}))}_{g+1 \text{ times}} \\
&\rightarrow_{\text{DOWN}}^* m(n(\vec{w}', x_1, \dots, x_{g'}), \dots, n(\vec{w}', x_1, \dots, x_{g'})) \\
&\hookrightarrow_g^* \underline{m}(n(\vec{w})) \\
&= u
\end{aligned}$$

This concludes the proof. \square

Lemma 6.17. *We have $\rightarrow_{SH} \subseteq \rightarrow^+$ where \rightarrow uses the given order $<$ on S' .*

Proof. Assume that $s \rightarrow_{SH} t$. Then

$$s \rightarrow_{\text{CHOP}} \cdot \rightarrow_{\text{PROPAGATE}}^! s'' (\rightarrow_{\text{WIDEN}} \cup \rightarrow_{\text{LENGTHEN}})^n t' \rightarrow_{\text{WAIVE}}^! t$$

for some $n \geq 0$ and trees s'', t' . By Lemma 6.14 we have $s \rightarrow^+ s' \hookrightarrow_{n+1}^* s''$ for some $s' \in \mathbb{T}(S', \mathcal{X})$. Using induction on n together with Lemmas 6.15 and 6.16, we obtain $s' \rightarrow^* t'' \hookrightarrow_1 t'$ for some $t'' \in \mathbb{T}(S', \mathcal{X})$. Thus we have

$$s \rightarrow^* s' \rightarrow^* t'' \hookrightarrow_1 t' \rightarrow_{\text{WAIVE}}^! t$$

Moreover, by Lemma 6.13 we have $t'' \rightarrow^* [t'] = t$. Hence $s \rightarrow^* t$. \square

Theorem 6.18. *The Star Hydra \rightarrow_{SH} is terminating.*

Proof. A direct consequence of Lemma 6.17. \square

7. CONCLUDING REMARKS AND QUESTIONS

We have revisited star games as an alternative setting for dealing with several termination problems, to wit, of word rewriting systems, term rewriting systems, and other objects, in this paper in particular Hydras. We expect that the scope of this method, consisting of letting some control symbols walk over the objects and transforming them, is wider than we could treat in this paper. We mention a few follow-up matters, that we hope to address in subsequent notes or papers.

- (1) An essential part of the star games is that they cooperate with well-quasi-orders, as vocalized in our simple but basic Proposition 4.1. That requires that the appropriate embedding relations in the wqo can be obtained by a subset of the star game rules, or by derived rules, derived from the star rules. Above we have seen this for the notion of embedding as in Kruskal's tree theorem.

A challenge is to extend the star rules so that they capture the more difficult embedding on words known as *gap embedding*, and likewise the corresponding embedding on trees as proposed by Friedman and studied by Dershowitz and Tzameret [DT03, Tza02].

- (2) In general, it is our endeavour to extend the *expressivity* of the star rules by adding new rules. A typical example, already realized in the literature, is admitting the LEX rule as treated in the paper ILPO [KvOdV06], that actually makes the DOWN rule a derived rule, and that importantly, enables one to prove termination of functions like the one

of Ackermann. See further Geser [Ges90]. However, in the setting of trees (as we have used in this paper), the LEX rule is unsound for proving termination (see Remark 3.5).

- (3) In continuation of the previous item about enhancing expressivity, fruitful results have already been obtained towards connecting the framework of this paper to *higher-order rewriting*, pertaining to various typed and untyped lambda calculi, or combinations of these with first-order rewriting such as in HRSs and CRSs [Klo92].

For the original version of RPO, progress in this direction has been made by Jouannaud and Rubio and others. For the equivalent version of IPO as in [KvOdV06] and the present paper, several developments have been realized by Kop [Kop12] and Van Raamsdonk [KvR09].

- (4) Another direction is the adaptation of the star game for proving infinitary normalization [EGH⁺08] and productivity [End10, EGH09].
- (5) We also would like to enhance the *precision* of the star rules. Can the star game be changed in order to prove *local termination* [EdVW09, EdVW10] on a given set of starting terms, in contrast to *global termination* on all terms?
- (6) The star game framework lends itself to a further syntactical analysis of its fine-structure along the lines of obtaining for instance a *standardization theorem*, well-known in lambda calculus and first order rewriting. Such a theorem was already included and used in [KvOdV06], but as yet without proof, which was subsequently given by Van Oostrom under the name of obtaining *cut elimination*. An interesting corollary pointed out by him in that work is the *decidability* of the star rewrite relation. The point of such a strategy is that computation is going down along the tree with a *wave front* of star symbols, denoting the front of activity, and freezing everything above that front.
- (7) Are there interesting variations of star games for proving termination in infinitary rewriting [EHK12, EHH⁺18], that is, considering infinite trees and rewrite sequences of ordinal length?
- (8) A further question that we wondered about, is obtained by *reflecting a star game in itself*, and continuing to repeat that. That is, we can allow finite ω -labeled trees themselves as labels, and determine the termination ordinal thus obtained. We can reiterate this feeding the system into itself, even in an ordinal long procedure. Here the notion of normal function on ordinals will likely play a role. Can we determine at what ordinal this process closes? We expect that for some of these questions there are already answers in the literature, but we have not encountered such answers explicitly, for several questions.
- (9) The most fascinating question to our mind concerns the extension of star games to *finite graphs*, to start with, undirected and unlabeled as in the original *graph minor theorem (GMT)* by Robertson and Seymour [RS04]. Obvious generalizations here are admitting directed edges, and admitting ordinal labels on the nodes. This is a very powerful generalization of KTT. Ideally, we would like to obtain a star game on such finite graphs, yielding a framework to termination of various graph rewrite problems, by simulation into this terminating graph framework [JK08]. A promising advance in this direction has been made by Dershowitz and Jouannaud, who designed a very general datatype called *drags*, not using star games, but with developments towards setting up RPO, recursive path order, for these objects [DJ19, DJ18].

ACKNOWLEDGMENTS

We thank Nachum Dershowitz for his useful conversations concerning a draft of this paper and many comments and pointers to the literature, during his sabbatical period at CWI Amsterdam around end of 2019 and beginning of 2020. Also we gratefully acknowledge support for our research from CWI Amsterdam and the Section Theory of the Vrije Universiteit Amsterdam. The third author received funding from NWO under the Innovational Research Incentives Scheme (project No. VI.Vidi.192.004). The second author is thankful for monthly funding from the Dutch ABP, the Algemeen Burgerlijk Pensioenfonds, for daily subsistence and cost of living.

Many thanks to Vincent van Oostrom, for a scrutiny of a draft of this paper and numerous comments for improvement. We thank Andreas Weiermann and Toshiyasu Arai for advising us on references for ordinal analysis. We finally thank two anonymous referees, for several more comments and valuable suggestions.

REFERENCES

- [Bae84] Jos Baeten. Filters and Ultrafilters over Definable Subsets of Admissible Ordinals. In *Models and Sets*, pages 1–8. Springer, 1984.
- [BK85] Jan A. Bergstra and Jan Willem Klop. Algebra of Communicating Processes with Abstraction. *Theoretical Computer Science*, 37:77–121, 1985.
- [BKM89] Jan A. Bergstra, Jan Willem Klop, and Aart Middeldorp. *Termherschrijfsystemen*. Kluwer Bedrijfswetenschappen, 1989.
- [BN98] Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.
- [Buc86] W. Buchholz. A new system of proof-theoretic ordinal functions. *Annals of Pure and Applied Logic*, 32:195–207, 1986.
- [Buc87] Wilfried Buchholz. An independence result for $(\Pi_1^1 - CA) + BI$. *Annals of Pure and Applied Logic*, 33:131–155, 1987.
- [Buc95] Wilfried Buchholz. Proof-theoretic analysis of termination proofs. *Annals of Pure and Applied Logic*, 75(1-2):57–65, 1995.
- [Buc07] Wilfried Buchholz. Another rewrite system for the standard Hydra battle. *Oberwolfach Report*, 52(2006):13–15, 2007.
- [Der82] Nachum Dershowitz. Orderings for Term-Rewriting Systems. *Theoretical Computer Science*, 17(3):279–301, 1982.
- [Der87] Nachum Dershowitz. Termination of rewriting. *Journal Symbolic Computation*, 3(1/2):69–116, 1987.
- [DJ90] Nachum Dershowitz and Jean-Pierre Jouannaud. Rewrite Systems. In *Handbook of Theoretical Computer Science*, volume B: Formal Models and Semantics, chapter 15, pages 243–320. North-Holland, Amsterdam, 1990.
- [DJ18] Nachum Dershowitz and Jean-Pierre Jouannaud. Graph Path Orderings. In *Proc. Conf. on Logic for Programming, Artificial Intelligence and Reasoning*, volume 57 of *EPiC Series in Computing*, pages 307–325. EasyChair, 2018.
- [DJ19] Nachum Dershowitz and Jean-Pierre Jouannaud. Drags: A Simple Algebraic Framework For Graph Rewriting. *Theoretical Computer Science*, 777:204–231, 2019.
- [DM07] Nachum Dershowitz and Georg Moser. The Hydra Battle Revisited. In *Rewriting, Computation and Proof*, pages 1–27. Springer, 2007.
- [DT03] Nachum Dershowitz and Iddo Tzameret. Gap Embedding for Well-Quasi-Orderings. *Electronic Notes in Theoretical Computer Science*, 84:80–90, 2003.
- [EdVW09] Jörg Endrullis, Roel C. de Vrijer, and Johannes Waldmann. Local Termination. In *Proc. Conf. on Rewriting Techniques and Applications*, volume 5595 of *Lecture Notes in Computer Science*, pages 270–284. Springer, 2009.

- [EdVW10] Jörg Endrullis, Roel C. de Vrijer, and Johannes Waldmann. Local Termination: Theory and Practice. *Logical Methods in Computer Science*, 6(3), 2010.
- [EGH⁺08] Jörg Endrullis, Clemens Grabmayer, Dimitri Hendriks, Jan Willem Klop, and Roel C. de Vrijer. Proving Infinitary Normalization. In *Proc. Conf. on Types for Proofs and Programs*, volume 5497 of *Lecture Notes in Computer Science*, pages 64–82. Springer, 2008.
- [EGH09] Jörg Endrullis, Clemens Grabmayer, and Dimitri Hendriks. Complexity of Fractran and Productivity. In *Proc. Conf. on Automated Deduction*, volume 5663 of *Lecture Notes in Computer Science*, pages 371–387. Springer, 2009.
- [EHH⁺18] Jörg Endrullis, Helle Hvid Hansen, Dimitri Hendriks, Andrew Polonsky, and Alexandra Silva. Coinductive Foundations of Infinitary Rewriting and Infinitary Equational Logic. *Logical Methods in Computer Science*, 14(1), 2018.
- [EHK12] Jörg Endrullis, Dimitri Hendriks, and Jan Willem Klop. Highlights in Infinitary Rewriting and Lambda Calculus. *Theoretical Computer Science*, 464:48–71, 2012.
- [EK13] Jörg Endrullis and Jan Willem Klop. De Bruijn’s Weak Diamond Property Revisited. *Indagationes Mathematicae*, 24(4):1050 – 1072, 2013. In memory of N.G. (Dick) de Bruijn (1918–2012).
- [EKO18] Jörg Endrullis, Jan Willem Klop, and Roy Overbeek. Decreasing Diagrams with Two Labels Are Complete for Confluence of Countable Systems. In *Proc. Conf. on Formal Structures for Computation and Deduction*, volume 108 of *LIPICs*, pages 14:1–14:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.
- [End] Jörg Endrullis. Jambox: Automated Termination Analysis of Term Rewriting Systems.
- [End10] Jörg Endrullis. *Termination and Productivity*. PhD thesis, Vrije Universiteit Amsterdam, 2010.
- [EZ15] Jörg Endrullis and Hans Zantema. Proving Non-termination by Finite Automata. In *Proc. Conf. on Rewriting Techniques and Applications*, volume 36 of *LIPICs*, pages 160–176. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015.
- [FvO13] Bertram Felgenhauer and Vincent van Oostrom. Proof Orders for Decreasing Diagrams. In *Proc. Conf. on Rewriting Techniques and Applications*, volume 21 of *LIPICs*, pages 174–189. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2013.
- [Gal91] Jean H. Gallier. 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, 1991.
- [Ges90] Alfons Geser. *Relative Termination*. PhD thesis, Universiteit Passau, 1990. Appeared as technical report 91-03, Ulmer Informatik-Berichte, Universität Ulm.
- [Hig52] Graham Higman. Ordering by Divisibility in Abstract Algebras. *Proceedings of the London Mathematical Society*, s3-2(1):326–336, 1952.
- [HO98] Masahiro Hamano and Mitsuhiro Okada. A Direct Independence Proof of Buchholz’s Hydra Game on Finite Labeled Trees. *Archive for Mathematical Logic*, 37(2):67–89, 1998.
- [JK08] Salil Joshi and Barbara König. Applying the Graph Minor Theorem to the Verification of Graph Transformation Systems. In *Proc. Conf. on Computer Aided Verification*, pages 214–226. Springer, 2008.
- [Kan94] Akihiro Kanamori. *The Higher Infinite. Large Cardinals in Set Theory from Their Beginnings*. Perspectives in Mathematical Logic. Springer, 1994.
- [KL80] Samuel Kamin and Jean-Jacques Lévy. Two generalizations of the recursive path ordering. *Unpublished note, Department of Computer Science, University of Illinois, Urbana, IL.*, 1980.
- [Klo92] Jan Willem Klop. Term Rewriting Systems. In *Handbook of Logic in Computer Science*, volume II, pages 1–116. Oxford University Press, 1992.
- [Kop12] Cynthia L.M. Kop. *Higher Order Termination: Automatable Techniques for Proving Termination of Higher-Order Term Rewriting Systems*. PhD thesis, Vrije Universiteit Amsterdam, 2012.
- [KP82] Laurie Kirby and Jeff Paris. Accessible Independence Results for Peano Arithmetic. *Bulletin of the London Mathematical Society*, 14(4):285–293, 1982.
- [Kru60] Joseph B. Kruskal. Well-quasi-ordering, the tree theorem, and vazsonyi’s conjecture. *Transactions of the American Mathematical Society*, 95(2):210–225, 1960.
- [Kut02] Temur Kutsia. *Solving and proving in equational theories with sequence variables and flexible arity symbols*. PhD thesis, Institute RISC-Linz, Johannes Kepler University, Hagenberg, Austria, 2002.

- [Kut04] Temur Kutsia. Solving equations involving sequence variables and sequence functions. In Bruno Buchberger and John Campbell, editors, *Artificial Intelligence and Symbolic Computation*, pages 157–170, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
- [KvOdV06] Jan Willem Klop, Vincent van Oostrom, and Roel C. de Vrijer. Iterative Lexicographic Path Orders. In *Algebra, Meaning, and Computation*, pages 541–554. Springer, 2006.
- [KvR08] Cynthia Kop and Femke van Raamsdonk. A Higher-Order Iterative Path Ordering. In *Proc. Conf. on Logic for Programming, Artificial Intelligence, and Reasoning*, volume 5330 of *Lecture Notes in Computer Science*, pages 697–711. Springer, 2008.
- [KvR09] Cynthia Kop and Femke van Raamsdonk. An Iterative Path Ordering. *Liber Amicorum for Roel de Vrijer’s 60th Birthday*, pages 145–153, 2009.
- [Man80] Zohar Manna. *Lectures on the Logic of Computer Programming*. NSF-CBMS Regional Research Conferences in the Mathematical Sciences. Society for Industrial and Applied Mathematics, 1980.
- [Mos09] Georg Moser. The Hydra battle and Cichon’s principle. *Applicable Algebra in Engineering, Communication and Computing*, 20(2):133–158, 2009.
- [MZ97] Aart Middeldorp and Hans Zantema. Simple termination of rewrite systems. *Theoretical Computer Science*, 175(1):127–158, 1997.
- [NW67] Crispin St. J.A. Nash-Williams. Infinite graphs—A survey. *Journal of Combinatorial Theory*, 3(3):286–301, 1967.
- [Oka88] Mitsuhiro Okada. Note on a Proof of the Extended Kirby—Paris Theorem on Labeled Finite Trees. *European Journal of Combinatorics*, 9(3):249–253, 1988.
- [Rat06] Michael Rathjen. The art of ordinal analysis. In *Proceedings of the International Congress of Mathematicians*, volume 2, pages 45–69, 2006.
- [RS04] Neil Robertson and Paul D. Seymour. Graph Minors. XX. Wagner’s conjecture. *Journal of Combinatorial Theory, Series B*, 92(2):325 – 357, 2004. Special Issue Dedicated to Professor W.T. Tutte.
- [Rub67] Jean E. Rubin. *Set Theory for the Mathematician*. Holden-Day Series in Mathematics. Holden-Day, 1967.
- [RW93] Michael Rathjen and Andreas Weiermann. Proof-theoretic investigations on Kruskal’s theorem. *Annals of Pure and Applied Logic*, 60(1):49–88, 1993.
- [Sch65] Kurt Schütte. Predicative well-orderings. In *Formal Systems and Recursive Functions*, volume 40 of *Studies in Logic and the Foundations of Mathematics*, pages 280 – 303. Elsevier, 1965.
- [SSW20] Peter M. Schuster, Monika Seisenberger, and Andreas Weiermann, editors. *Well-Quasi Orders in Computation, Logic, Language and Reasoning*. Springer International Publishing, 2020.
- [Tak13] Gaisi Takeuti. *Proof Theory*, volume 81. Courier Corporation, 2013.
- [Ter03] Terese, editor. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.
- [Tou98] H elene Touzet. Encoding the Hydra Battle as a Rewrite System. In *Proc. Symp. on Mathematical Foundations of Computer Science*, pages 267–276. Springer, 1998.
- [Tza02] Iddo Tzameret. Kruskal-Friedman Gap Embedding Theorems over Well-Quasi-Orderings. Master’s thesis, The Blavatnik School of Computer Science, Tel Aviv University, 2002.
- [Veb08] Oswald Veblen. Continuous increasing functions of finite and transfinite ordinals. *Transactions of the American Mathematical Society*, 9(3):280–280, 1908.
- [vO94a] Vincent van Oostrom. Confluence by Decreasing Diagrams. *Theoretical Computer Science*, 126(2):259–280, 1994.
- [vO94b] Vincent van Oostrom. *Confluence for Abstract and Higher-Order Rewriting*. PhD thesis, Vrije Universiteit Amsterdam, 1994.
- [Wei94] Andreas Weiermann. Complexity Bounds for Some Finite Forms of Kruskal’s Theorem. *Journal of Symbolic Computation*, 18(5):463–488, 1994.
- [Zan03] Hans Zantema. Termination. In *Term Rewriting Systems*, pages 181–259. Cambridge University Press, 2003.