RWTH Aachen

Department of Computer Science
Technical Report

Inferring Lower Bounds for Runtime Complexity

Florian Frohn, Jürgen Giesl, Jera Hensel, Cornelius Aschermann, Thomas Ströder

ISSN 0935–3232 · Aachener Informatik-Berichte · AIB-2015-15

RWTH Aachen · Department of Computer Science · April 2015 (revised version)

The publications of the Department of Computer Science of *RWTH Aachen University* are in general accessible through the World Wide Web.

http://aib.informatik.rwth-aachen.de/

Inferring Lower Bounds for Runtime Complexity[*]

Florian Frohn, Jürgen Giesl, Jera Hensel, Cornelius Aschermann, and Thomas Ströder

LuFG Informatik 2, RWTH Aachen University, Germany
`{florian.frohn,giesl,hensel,cornelius,stroeder}@informatik.rwth-aachen.de`

Abstract

We present the first approach to deduce lower bounds for innermost runtime complexity of term rewrite systems (TRSs) automatically. Inferring lower runtime bounds is useful to detect bugs and to complement existing techniques that compute upper complexity bounds. The key idea of our approach is to generate suitable families of rewrite sequences of a TRS and to find a relation between the length of such a rewrite sequence and the size of the first term in the sequence. We implemented our approach in the tool AProVE and evaluated it by extensive experiments.

1998 ACM Subject Classification F.1.3 - Complexity Measures and Classes, F.4.2 - Grammars and Other Rewriting Systems, I.2.3 Deduction and Theorem Proving

Keywords and phrases Term Rewriting, Runtime Complexity, Lower Bounds, Induction

1 Introduction

There exist numerous methods to infer *upper bounds* for the runtime complexity of TRSs [3, 11, 13, 16, 20]. We present the first automatic technique to infer *lower bounds* for the innermost[1] runtime complexity of TRSs. *Runtime complexity* [11] refers to the "worst" cases in terms of evaluation length and our goal is to find lower bounds for these cases. While upper complexity bounds help to prove the absence of bugs that worsen the performance of programs, lower bounds can be used to *find* such bugs. Moreover, in combination with methods to deduce upper bounds, our approach can prove *tight* complexity results. In addition to *asymptotic* lower bounds, in many cases our technique can even compute *concrete* bounds.

As an example, consider the following TRS $\mathcal{R}_{\mathsf{qs}}$ for *quicksort*. The auxiliary function $\mathsf{low}(x, xs)$ returns those elements from the list xs that are smaller than x (and high works analogously). To ease readability, we use infix notation for the function symbols $\leq$ and $\mathsf{++}$.

▶ **Example 1** (TRS $\mathcal{R}_{\mathsf{qs}}$ for Quicksort).

$$
\begin{array}{rcll}
\mathsf{qs}(\mathsf{nil}) & \to & \mathsf{nil} & (1)\\
\mathsf{qs}(\mathsf{cons}(x, xs)) & \to & \mathsf{qs}(\mathsf{low}(x, xs)) \mathbin{\mathsf{++}} \mathsf{cons}(x, \mathsf{qs}(\mathsf{high}(x, xs))) & (2)\\
\mathsf{low}(x, \mathsf{nil}) & \to & \mathsf{nil} & \\
\mathsf{low}(x, \mathsf{cons}(y, ys)) & \to & \mathsf{ifLow}(x \leq y, x, \mathsf{cons}(y, ys)) & \\
\mathsf{ifLow}(\mathsf{true}, x, \mathsf{cons}(y, ys)) & \to & \mathsf{low}(x, ys) & \\
\mathsf{ifLow}(\mathsf{false}, x, \mathsf{cons}(y, ys)) & \to & \mathsf{cons}(y, \mathsf{low}(x, ys)) & \\
\mathsf{high}(x, \mathsf{nil}) & \to & \mathsf{nil} & \\
\mathsf{high}(x, \mathsf{cons}(y, ys)) & \to & \mathsf{ifHigh}(x \leq y, x, \mathsf{cons}(y, ys)) & \\
\mathsf{ifHigh}(\mathsf{true}, x, \mathsf{cons}(y, ys)) & \to & \mathsf{cons}(y, \mathsf{high}(x, ys)) & \\
\mathsf{ifHigh}(\mathsf{false}, x, \mathsf{cons}(y, ys)) & \to & \mathsf{high}(x, ys) &
\end{array}
\qquad
\begin{array}{rcll}
\mathsf{zero} \leq x & \to & \mathsf{true} & \\
\mathsf{succ}(x) \leq \mathsf{zero} & \to & \mathsf{false} & \\
\mathsf{succ}(x) \leq \mathsf{succ}(y) & \to & x \leq y & \\
\mathsf{nil} \mathbin{\mathsf{++}} ys & \to & ys & (3)\\
\mathsf{cons}(x, xs) \mathbin{\mathsf{++}} ys & \to & \mathsf{cons}(x, xs \mathbin{\mathsf{++}} ys) &
\end{array}
$$

[*] Supported by the DFG grant GI 274/6-1.

[1] We consider *innermost* rewriting, since TRSs resulting from the translation of programs usually have to be evaluated with an innermost strategy (e.g., [9, 17]). Obviously, lower bounds for innermost reductions are also lower bounds for full reductions (i.e., our approach can also be used for full rewriting).

For any $n \in \mathbb{N}$, let $\gamma_{\mathbf{List}}(n)$ be the term $\overbrace{\mathsf{cons}(\mathsf{zero}, \ldots, \mathsf{cons}(\mathsf{zero}, \mathsf{nil})}^{n \text{ times}} \ldots)$, i.e., the list of length n where all elements have the value zero (we also use the notation "$\mathsf{cons}^n(\mathsf{zero}, \mathsf{nil})$"). To find lower bounds, we automatically generate *rewrite lemmas* that describe families of rewrite sequences. For example, our technique infers the following rewrite lemma automatically.

$$\mathsf{qs}(\gamma_{\mathbf{List}}(n)) \quad \xrightarrow{\mathrm{i}}{}^{3n^2+2n+1} \quad \gamma_{\mathbf{List}}(n) \tag{4}$$

This rewrite lemma means that for each $n \in \mathbb{N}$, there is an innermost rewrite sequence of length $3n^2 + 2n + 1$ that reduces $\mathsf{qs}(\mathsf{cons}^n(\mathsf{zero}, \mathsf{nil}))$ to $\mathsf{cons}^n(\mathsf{zero}, \mathsf{nil})$. From this rewrite lemma, our technique then concludes that the innermost runtime of $\mathcal{R}_{\mathsf{qs}}$ is at least quadratic.

While most methods to infer upper bounds are adaptions of termination techniques, the approach in this paper is related to our technique to prove non-termination of TRSs [7]. Both techniques generate "meta-rules" representing infinitely many rewrite sequences. However, the *rewrite lemmas* in the current paper are more general than the meta-rules in [7], as they can be parameterized by *several* variables $n_1, \ldots, n_m$ of type $\mathbb{N}$.

In Sect. 2 we show how to automatically speculate conjectures that may result in suitable rewrite lemmas. Sect. 3 explains how these conjectures can be verified automatically by induction. From these induction proofs, one can deduce information on the lengths of the rewrite sequences represented by a rewrite lemma, cf. Sect. 4. Thus, the use of induction to infer lower runtime bounds represents a novel application for automated inductive theorem proving. This complements our earlier work on using inductive theorem proving for termination analysis [8]. Finally, Sect. 5 shows how rewrite lemmas can be used to infer lower bounds for the innermost runtime complexity of a TRS.

Sect. 6 discusses an improvement of our approach by pre-processing the TRS before the analysis and Sect. 7 extends our approach to handle rewrite lemmas with arbitrary unknown right-hand sides. We implemented our technique in the tool AProVE [10] and demonstrate its power by an extensive experimental evaluation in Sect. 8. All proofs can be found in the appendix.

2 Speculating Conjectures

We now show how to speculate conjectures (whose validity must be proved afterwards in Sect. 3). See, e.g., [5] for the basics of rewriting, where we only consider finite TRSs. $\mathcal{T}(\Sigma, \mathcal{V})$ is the set of all terms over a (finite) signature Σ and a set of variables $\mathcal{V}$ and $\mathcal{T}(\Sigma) = \mathcal{T}(\Sigma, \varnothing)$ is the set of ground terms. The *arity* of a symbol $f \in \Sigma$ is denoted by $\mathrm{ar}_\Sigma(f)$. As usual, the *defined symbols* of a TRS $\mathcal{R}$ are $\Sigma_{def}(\mathcal{R}) = \{\, \mathrm{root}(\ell) \mid \ell \to r \in \mathcal{R}\,\}$ and the *constructors* $\Sigma_{con}(\mathcal{R})$ are all other function symbols in $\mathcal{R}$. Thus, $\Sigma_{def}(\mathcal{R}_{\mathsf{qs}}) = \{\mathsf{qs}, \mathsf{low}, \mathsf{ifLow}, \mathsf{high}, \mathsf{ifHigh}, \mathsf{++}, \leq\}$ and $\Sigma_{con}(\mathcal{R}_{\mathsf{qs}}) = \{\mathsf{nil}, \mathsf{cons}, \mathsf{zero}, \mathsf{succ}, \mathsf{true}, \mathsf{false}\}$.

Our approach is based on rewrite lemmas containing *generator functions* such as $\gamma_{\mathbf{List}}$ for types like **List**. Hence, in the first step of our approach we compute suitable types for the TRS $\mathcal{R}$ to be analyzed. While ordinary TRSs are defined over untyped signatures Σ, Def. 2 shows how to extend such signatures by (monomorphic) types (see, e.g., [8, 13, 21]).

▶ **Definition 2** (Typing)**.** Let Σ be an (untyped) signature. A many-sorted signature Σ' is a *typed variant* of Σ if it contains the same function symbols as Σ, with the same arities. So $f \in \Sigma$ with $\mathrm{ar}_\Sigma(f) = k$ iff $f \in \Sigma'$ where f's type has the form $\tau_1 \times \ldots \times \tau_k \to \tau$. Similarly, a typed variant $\mathcal{V}'$ of the set of variables $\mathcal{V}$ contains the same variables as $\mathcal{V}$, but now every variable has a type τ. We always assume that for every type τ, $\mathcal{V}'$ contains infinitely many variables of type τ. Given Σ' and $\mathcal{V}'$, $t \in \mathcal{T}(\Sigma, \mathcal{V})$ is a *well-typed* term of type τ iff

- $t \in \mathcal{V}'$ is a variable of type τ or
- $t = f(t_1, \ldots, t_k)$ with $k \geq 0$, where each t_i is a well-typed term of type τ_i, and where $f \in \Sigma'$ has the type $\tau_1 \times \ldots \times \tau_k \to \tau$.

We only permit typed variants Σ' where there exist well-typed ground terms of types $\tau_1, \ldots, \tau_k$ over Σ', whenever some $f \in \Sigma'$ has type $\tau_1 \times \ldots \times \tau_k \to \tau$.[2]

A TRS $\mathcal{R}$ over Σ and $\mathcal{V}$ is *well typed* w.r.t. Σ' and $\mathcal{V}'$ iff for all $\ell \to r \in \mathcal{R}$, we have that ℓ and r are well typed and that they have the same type.[3]

For any TRS $\mathcal{R}$, one can use a standard type inference algorithm to compute a typed variant Σ' such that $\mathcal{R}$ is well typed. Of course, a trivial solution is to use a many-sorted signature with just one sort (then every term and every TRS are trivially well typed). But to make our approach more powerful, it is advantageous to use the most general typed variant where $\mathcal{R}$ is well typed. Here, the set of terms is decomposed into as many types as possible. Then fewer terms are well typed and more useful rewrite lemmas can be generated.

To make $\mathcal{R}_{\mathsf{qs}}$ from Ex. 1 well typed, we obtain a typed variant of its signature with the types **Nats**, **Bool**, and **List**. Here, the function symbols have the following types:

$$\begin{array}{rlrl} \mathsf{nil} : & \mathbf{List} & \mathsf{qs} : & \mathbf{List} \to \mathbf{List} \\ \mathsf{cons} : & \mathbf{Nats} \times \mathbf{List} \to \mathbf{List} & \mathsf{++} : & \mathbf{List} \times \mathbf{List} \to \mathbf{List} \\ \mathsf{zero} : & \mathbf{Nats} & \leq : & \mathbf{Nats} \times \mathbf{Nats} \to \mathbf{Bool} \\ \mathsf{succ} : & \mathbf{Nats} \to \mathbf{Nats} & \mathsf{low}, \mathsf{high} : & \mathbf{Nats} \times \mathbf{List} \to \mathbf{List} \\ \mathsf{true}, \mathsf{false} : & \mathbf{Bool} & \mathsf{ifLow}, \mathsf{ifHigh} : & \mathbf{Bool} \times \mathbf{Nats} \times \mathbf{List} \to \mathbf{List} \end{array}$$

A type τ *depends* on a type τ' (denoted $\tau \sqsupseteq_{dep} \tau'$) iff $\tau = \tau'$ or if there is a $c \in \Sigma'_{con}(\mathcal{R})$ of type $\tau_1 \times \ldots \times \tau_k \to \tau$ where $\tau_i \sqsupseteq_{dep} \tau'$ for some $1 \leq i \leq k$. To ease the presentation, we do not allow mutually recursive types (i.e., if $\tau \sqsupseteq_{dep} \tau'$ and $\tau' \sqsupseteq_{dep} \tau$, then $\tau' = \tau$). To speculate conjectures, we now introduce generator functions γ_τ. For any $n \in \mathbb{N}$, $\gamma_\tau(n)$ is a term from $\mathcal{T}(\Sigma'_{con}(\mathcal{R}))$ where a recursive constructor of type τ is nested n times. A constructor $c : \tau_1 \times \ldots \times \tau_k \to \tau$ is *recursive* iff $\tau_i = \tau$ for some $1 \leq i \leq k$. So for the type **Nats** above, we have $\gamma_{\mathbf{Nats}}(0) = \mathsf{zero}$ and $\gamma_{\mathbf{Nats}}(n+1) = \mathsf{succ}(\gamma_{\mathbf{Nats}}(n))$. If a constructor has a non-recursive argument of type τ', then γ_τ instantiates this argument by $\gamma_{\tau'}(0)$. So for **List**, we get $\gamma_{\mathbf{List}}(0) = \mathsf{nil}$ and $\gamma_{\mathbf{List}}(n+1) = \mathsf{cons}(\mathsf{zero}, \gamma_{\mathbf{List}}(n))$. If a constructor has several recursive arguments, then several generator functions are possible. So for a type **Tree** with the constructors $\mathsf{leaf} : \mathbf{Tree}$ and $\mathsf{node} : \mathbf{Tree} \times \mathbf{Tree} \to \mathbf{Tree}$, we have $\gamma_{\mathbf{Tree}}(0) = \mathsf{leaf}$, but either $\gamma_{\mathbf{Tree}}(n+1) = \mathsf{node}(\gamma_{\mathbf{Tree}}(n), \mathsf{leaf})$ or $\gamma_{\mathbf{Tree}}(n+1) = \mathsf{node}(\mathsf{leaf}, \gamma_{\mathbf{Tree}}(n))$. Similarly, if a type has several non-recursive or recursive constructors, then several different generator functions can be constructed by considering all combinations of non-recursive and recursive constructors.

To ease the presentation, we only consider generator functions for *simply structured* types τ. Such types have exactly two constructors $c, d \in \Sigma'_{con}(\mathcal{R})$, where c is not recursive, d has exactly one argument of type τ, and each argument type $\tau' \neq \tau$ of c or d is simply structured, too. The presented approach can easily be extended to more complex types by applying suitable heuristics to choose one of the possible generator functions.

▶ **Definition 3** (Generator Functions and Equations)**.** Let $\mathcal{R}$ be a TRS that is well typed w.r.t. Σ' and $\mathcal{V}'$. We extend the set of types by a fresh type $\mathbb{N}$. For every type $\tau \neq \mathbb{N}$, let γ_τ be a fresh *generator function symbol* of type $\mathbb{N} \to \tau$. The set $\mathcal{G}_\mathcal{R}$ consists of the following *generator*

---

[2] This is not a restriction, as one can simply add new constants to Σ and Σ'.

[3] W.l.o.g., here one may rename the variables in every rule. Then it is not a problem if the variable x is used with type τ_1 in one rule and with type τ_2 in another rule.

equations for every simply structured type τ with the constructors $c : \tau_1 \times \ldots \times \tau_k \to \tau$ and $d : \rho_1 \times \ldots \times \rho_b \to \tau$, where $\rho_j = \tau$. We write $\mathcal{G}$ instead of $\mathcal{G}_\mathcal{R}$ if $\mathcal{R}$ is clear from the context.

$$\begin{array}{rcl} \gamma_\tau(0) & = & c(\gamma_{\tau_1}(0), \ldots, \gamma_{\tau_k}(0)) \\ \gamma_\tau(n+1) & = & d(\gamma_{\rho_1}(0), \ldots, \gamma_{\rho_{j-1}}(0), \gamma_\tau(n), \gamma_{\rho_{j+1}}(0), \ldots, \gamma_{\rho_b}(0)) \end{array}$$

We extend $\sqsupseteq_{dep}$ to $\Sigma_{def}(\mathcal{R})$ by defining $f \sqsupseteq_{dep} h$ iff $f = h$ or if there is a rule $f(\ldots) \to r$ and a symbol g in t with $g \sqsupseteq_{dep} h$. When speculating conjectures, we take the dependencies between defined symbols into account. If $f \sqsupseteq_{dep} g$ and $g \not\sqsupseteq_{dep} f$, then we first generate a rewrite lemma for g. This lemma can be used when generating a lemma for f afterwards.

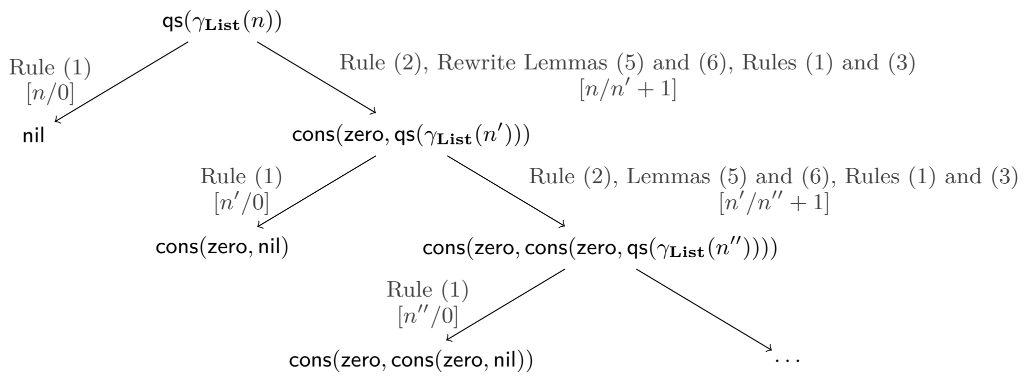
For $f \in \Sigma'_{def}(\mathcal{R})$ of type $\tau_1 \times \ldots \times \tau_k \to \tau$ with simply structured types $\tau_1, \ldots, \tau_k$, our goal is to speculate a conjecture of the form $f(\gamma_{\tau_1}(s_1), \ldots, \gamma_{\tau_k}(s_k)) \xrightarrow{\mathrm{i}}{}^* t$, where the $s_1, \ldots, s_k$ are polynomials over variables $n_1, \ldots, n_m$ of type $\mathbb{N}$. Moreover, t is a term built from Σ, arithmetic expressions, generator functions, and $n_1, \ldots, n_m$. As usual, a rewrite step is *innermost* (denoted $s \xrightarrow{\mathrm{i}}_\mathcal{R} t$ where we omit the index $\mathcal{R}$ if it is clear from the context) if the reduced subterm of s does not have redexes as proper subterms. From the speculated conjecture, we afterwards infer a rewrite lemma $f(\gamma_{\tau_1}(s_1), \ldots, \gamma_{\tau_k}(s_k)) \xrightarrow{\mathrm{i}}{}^{rt(n_1,\ldots,n_m)} t$, where $rt : \mathbb{N}^m \to \mathbb{N}$ describes the <u>r</u>un<u>t</u>ime of the lemma. To speculate a conjecture, we first generate *sample conjectures* that describe the effect of applying f to specific arguments. To this end, we narrow $f(\gamma_{\tau_1}(n_1), \ldots, \gamma_{\tau_k}(n_k))$ where $n_1, \ldots, n_k \in \mathcal{V}$ using the rules of the TRS and the lemmas we have proven so far, taking also the generator equations and integer arithmetic into account.

For any proven rewrite lemma $s \xrightarrow{\mathrm{i}}{}^{rt(\ldots)} t$, let the set $\mathcal{L}$ contain the rule $s \to t$. Moreover, let $\mathcal{A}$ be the infinite set of all valid equalities in the theory of $\mathbb{N}$ with addition and multiplication. Then s *narrows* to t ("$s \rightsquigarrow_{(\mathcal{R}\cup\mathcal{L})/(\mathcal{G}\cup\mathcal{A})} t$" or just "$s \rightsquigarrow t$" if $\mathcal{R}$, $\mathcal{L}$, $\mathcal{G}$ are clear from the context) iff there exist a term s', a substitution σ that maps variables of type $\mathbb{N}$ to arithmetic expressions, a position π, and a variable-renamed rule $\ell \to r \in \mathcal{R} \cup \mathcal{L}$ such that $s\sigma \equiv_{\mathcal{G}\cup\mathcal{A}} s'\sigma$, $s'|_\pi\sigma = \ell\sigma$, and $s'[r]_\pi\sigma = t$. Although checking $s\sigma \equiv_{\mathcal{G}\cup\mathcal{A}} s'\sigma$ (i.e., $\mathcal{G} \cup \mathcal{A} \models s\sigma = s'\sigma$) is undecidable in general, the required narrowing can usually be performed automatically using SMT solvers.

▶ **Example 4** (Narrowing). In Ex. 1 we have $\mathsf{qs} \sqsupseteq_{dep} \mathsf{low}$ and $\mathsf{qs} \sqsupseteq_{dep} \mathsf{high}$. If the lemmas

$$\mathsf{low}(\gamma_{\mathbf{Nats}}(0), \gamma_{\mathbf{List}}(n)) \xrightarrow{\mathrm{i}}{}^{3n+1} \gamma_{\mathbf{List}}(0) \quad (5) \qquad \mathsf{high}(\gamma_{\mathbf{Nats}}(0), \gamma_{\mathbf{List}}(n)) \xrightarrow{\mathrm{i}}{}^{3n+1} \gamma_{\mathbf{List}}(n) \quad (6)$$

were already proved, then the following narrowing tree can be generated to find sample conjectures for qs. The arrows are annotated with the rules and the substitutions used for variables of type $\mathbb{N}$. To save space, some arrows correspond to *several* narrowing steps.



The goal is to get representative rewrite sequences, but not to cover all reductions. So we stop constructing the tree after some steps and choose suitable narrowings heuristically.

After constructing a narrowing tree for f, we collect *sample points* (t, σ, d). Here, t results from a $\rightsquigarrow$-normal form q reached in a path of the tree by normalizing q w.r.t. the generator equations $\mathcal{G}$ applied from right to left. So terms from $\mathcal{T}(\Sigma, \mathcal{V})$ are rewritten to generator symbols with arithmetic expressions as arguments. Moreover, σ is the substitution for variables of type $\mathbb{N}$, and d is the number of applications of recursive f-rules on the path (the *recursion depth*). A rule $f(\ldots) \to r$ is *recursive* iff r contains a symbol g with $g \sqsupseteq_{dep} f$.

▶ **Example 5** (Sample Points)**.** In Ex. 4, we obtain the following set of sample points:[4]

$$S \;=\; \{\, (\gamma_{\mathbf{List}}(0), [n/0], 0), \quad (\gamma_{\mathbf{List}}(1), [n/1], 1), \quad (\gamma_{\mathbf{List}}(2), [n/2], 2) \,\} \tag{7}$$

The sequence from $\mathsf{qs}(\gamma_{\mathbf{List}}(n))$ to nil does not use recursive qs-rules. So its recursion depth is 0 and the $\rightsquigarrow$-normal form nil rewrites to $\gamma_{\mathbf{List}}(0)$ when applying $\mathcal{G}$ from right to left. The sequence from $\mathsf{qs}(\gamma_{\mathbf{List}}(n))$ to $\mathsf{cons}(\mathsf{zero}, \mathsf{nil})$ (resp. $\mathsf{cons}(\mathsf{zero}, \mathsf{cons}(\mathsf{zero}, \mathsf{nil}))$) uses the recursive qs-rule (2) once (resp. twice), i.e., it has recursion depth 1 (resp. 2). Moreover, these $\rightsquigarrow$-normal forms rewrite to $\gamma_{\mathbf{List}}(1)$ (resp. $\gamma_{\mathbf{List}}(2)$) when using $\mathcal{G}$ from right to left.

A sample point (t, σ, d) for a narrowing tree with the root $s = f(\ldots)$ represents the *sample conjecture* $s\sigma \xrightarrow{\mathsf{i}}{}^{*} t$, which stands for a reduction with d applications of recursive f-rules. So for $s = \mathsf{qs}(\gamma_{\mathbf{List}}(n))$, the sample points in (7) represent the sample conjectures $\mathsf{qs}(\gamma_{\mathbf{List}}(0)) \xrightarrow{\mathsf{i}}{}^{*} \gamma_{\mathbf{List}}(0)$, $\mathsf{qs}(\gamma_{\mathbf{List}}(1)) \xrightarrow{\mathsf{i}}{}^{*} \gamma_{\mathbf{List}}(1)$, $\mathsf{qs}(\gamma_{\mathbf{List}}(2)) \xrightarrow{\mathsf{i}}{}^{*} \gamma_{\mathbf{List}}(2)$. Now the goal is to speculate a general conjecture from these sample conjectures (whose validity must be proved afterwards).

In general, we search for a maximal subset of sample conjectures that are suitable for generalization. More precisely, if s is the root of the narrowing tree, then we take a maximal subset S_{max} of sample points such that for all $(t, \sigma, d), (t', \sigma', d') \in S_{max}$, the sample conjectures $s\sigma \xrightarrow{\mathsf{i}}{}^{*} t$ and $s\sigma' \xrightarrow{\mathsf{i}}{}^{*} t'$ are identical up to the occurring natural numbers and the variable names. For instance, $\mathsf{qs}(\gamma_{\mathbf{List}}(0)) \xrightarrow{\mathsf{i}}{}^{*} \gamma_{\mathbf{List}}(0)$, $\mathsf{qs}(\gamma_{\mathbf{List}}(1)) \xrightarrow{\mathsf{i}}{}^{*} \gamma_{\mathbf{List}}(1)$, and $\mathsf{qs}(\gamma_{\mathbf{List}}(2)) \xrightarrow{\mathsf{i}}{}^{*} \gamma_{\mathbf{List}}(2)$ are indeed identical up to the numbers in these sample conjectures. To obtain a general conjecture, we replace all numbers in the sample conjectures by polynomials. So in our example, we want to speculate a conjecture of the form $\mathsf{qs}(\gamma_{\mathbf{List}}(pol^{left})) \xrightarrow{\mathsf{i}}{}^{*} \gamma_{\mathbf{List}}(pol^{right})$. Here, pol^{left} and pol^{right} are polynomials in one variable n (the *induction variable* of the conjecture) that stands for the recursion depth. This facilitates a proof of the resulting conjecture by induction on n.

So in general, in any sample conjecture $s\sigma \xrightarrow{\mathsf{i}}{}^{*} t$ that correspond to a sample point $(t, \sigma, d) \in S_{max}$, we replace the natural numbers in $s\sigma$ and t by polynomials. For any term q, let $\mathrm{pos}(q)$ be the set of its positions and $\Pi^q_{\mathbb{N}} = \{\pi \in \mathrm{pos}(q) \mid q|_\pi \in \mathbb{N}\}$. Then for each $\pi \in \Pi^{s\sigma}_{\mathbb{N}}$ (resp. $\pi \in \Pi^t_{\mathbb{N}}$) with $(t, \sigma, d) \in S_{max}$, we search for a polynomial pol^{left}_π (resp. pol^{right}_π). To this end, for every sample point $(t, \sigma, d) \in S_{max}$, we generate the constraints

$$\text{“}pol^{left}_\pi(d) \;=\; s\sigma|_\pi\text{” for every } \pi \in \Pi^{s\sigma}_{\mathbb{N}} \quad \text{and} \quad \text{“}pol^{right}_\pi(d) \;=\; t|_\pi\text{” for every } \pi \in \Pi^t_{\mathbb{N}}. \tag{8}$$

Here, pol^{left}_π and pol^{right}_π are polynomials with abstract coefficients. So if one searches for polynomials of degree e, then the polynomials have the form $c_0 + c_1 \cdot n + c_2 \cdot n^2 + \ldots + c_e \cdot n^e$ and the constraints in (8) are linear diophantine equations over the unknown coefficients $c_i \in \mathbb{N}$.[5] These equations can easily be solved automatically. Finally, the desired generalized

[4] We always simplify arithmetic expressions in terms and substitutions, e.g., the substitution $[n/0 + 1]$ in the second sample point is simplified to $[n/1]$.

[5] Note that in the constraints (8), n is instantiated by an actual number d. Thus, if $pol^{left}_\pi = c_0 + c_1 \cdot n + c_2 \cdot n^2 + \ldots + c_e \cdot n^e$, then $pol^{left}_\pi(d)$ is a *linear* polynomial over the unknowns $c_0, \ldots, c_e$.

speculated conjecture is obtained from $s\sigma \xrightarrow{\mathrm{i}}{}^* t$ by replacing $s\sigma|_\pi$ with pol_π^{left} for every $\pi \in \Pi_{\mathbb{N}}^{s\sigma}$ and by replacing $t|_\pi$ with pol_π^{right} for every $\pi \in \Pi_{\mathbb{N}}^{t}$.

▶ **Example 6** (Speculating Conjectures). In Ex. 4, we narrowed $s{=}\mathsf{qs}(\gamma_{\mathbf{List}}(n))$ and S_{max} is the set S in (7). For each $(t,\sigma,d) \in S_{max}$, we have $\Pi_{\mathbb{N}}^{s\sigma}{=}\{1.1\}$ and $\Pi_{\mathbb{N}}^{t}{=}\{1\}$. So from the sample conjecture $\mathsf{qs}(\gamma_{\mathbf{List}}(0)) \xrightarrow{\mathrm{i}}{}^* \gamma_{\mathbf{List}}(0)$, where the recursion depth is $d{=}0$, we obtain the constraints $pol_{1.1}^{left}(d){=}pol_{1.1}^{left}(0){=}\mathsf{qs}(\gamma_{\mathbf{List}}(0))|_{1.1}{=}0$ and $pol_1^{right}(d){=}pol_1^{right}(0){=}\gamma_{\mathbf{List}}(0)|_1{=}0$. Similarly, from the two other sample conjectures we get $pol_{1.1}^{left}(1){=}pol_1^{right}(1){=}1$ and $pol_{1.1}^{left}(2){=}pol_1^{right}(2){=}2$. When using $pol_{1.1}^{left}{=}c_0{+}c_1{\cdot}n{+}c_2{\cdot}n^2$ and $pol_1^{right}{=}d_0{+}d_1{\cdot}n{+}d_2{\cdot}n^2$ with the abstract coefficients $c_0,\dots,c_2,d_0,\dots,d_2$, the solution $c_0{=}c_2{=}d_0{=}d_2{=}0$, $c_1{=}d_1{=}1$ (i.e., $pol_{1.1}^{left}{=}n$ and $pol_1^{right}{=}n$) is easily found automatically. So the resulting conjecture is $\mathsf{qs}(\gamma_{\mathbf{List}}(pol_{1.1}^{left})) \xrightarrow{\mathrm{i}}{}^* \gamma_{\mathbf{List}}(pol_1^{right})$, i.e., $\mathsf{qs}(\gamma_{\mathbf{List}}(n)) \xrightarrow{\mathrm{i}}{}^* \gamma_{\mathbf{List}}(n)$.

If S_{max} contains sample points with e different recursion depths, then we generate polynomials of at most degree $e-1$ satisfying the constraints (8) (these polynomials are determined uniquely). Ex. 7 shows how to speculate conjectures with *several* variables.

▶ **Example 7** (Conjecture With Several Variables). The following TRS combines half and plus.

$$\mathsf{hp}(\mathsf{zero}, y) \to y \qquad \mathsf{hp}(\mathsf{succ}(\mathsf{succ}(x)), y) \to \mathsf{succ}(\mathsf{hp}(x, y))$$

Narrowing $s = \mathsf{hp}(\gamma_{\mathbf{Nats}}(n_1), \gamma_{\mathbf{Nats}}(n_2))$ yields the sample points $(\gamma_{\mathbf{Nats}}(n_2), [n_1/0], 0)$, $(\gamma_{\mathbf{Nats}}(n_2+1), [n_1/2], 1)$, $(\gamma_{\mathbf{Nats}}(n_2+2), [n_1/4], 2)$, and $(\gamma_{\mathbf{Nats}}(n_2+3), [n_1/6], 3)$. For the last three sample points (t,σ,d), the only number in $s\sigma$ is at position 1.1 and the polynomial $pol_{1.1}^{left} = 2\cdot n$ satisfies the constraint $pol_{1.1}^{left}(d) = s\sigma|_{1.1}$. Moreover, the only number in t is at position 1.2 and the polynomial $pol_{1.2}^{right} = n$ satisfies $pol_{1.2}^{right} = t|_{1.2}$. Thus, we speculate the conjecture $\mathsf{hp}(\gamma_{\mathbf{Nats}}(2\cdot n), \gamma_{\mathbf{Nats}}(n_2)) \xrightarrow{\mathrm{i}}{}^* \gamma_{\mathbf{Nats}}(n_2+n)$ with the induction variable n.

3 Proving Rewrite Lemmas

If the proof of a speculated conjecture succeeds, then we have found a *rewrite lemma*.

▶ **Definition 8** (Rewrite Lemmas). Let $\mathcal{R}$ be a TRS that is well typed w.r.t. Σ' and $\mathcal{V}'$. For any term q, let $q{\downarrow_{\mathcal{G}/\mathcal{A}}}$ be q's normal form w.r.t. $\mathcal{G}_\mathcal{R}$, where the generator equations are applied from left to right and $\mathcal{A}$-equivalent (sub)terms are considered to be equal. Moreover, let $s \xrightarrow{\mathrm{i}}{}^* t$ be a conjecture with $\mathcal{V}(s) = \{n_1,\dots,n_m\} \neq \varnothing$, where $\overline{n} = (n_1,\dots,n_m)$ are pairwise different variables of type $\mathbb{N}$, s is well typed, $\mathrm{root}(s) \in \Sigma_{def}(\mathcal{R})$, and s has no defined symbol from $\Sigma_{def}(\mathcal{R})$ below the root. Let $rt : \mathbb{N}^m \to \mathbb{N}$. Then $s \to_{\mathcal{R}}^{rt(\overline{n})} t$ is a *rewrite lemma* for $\mathcal{R}$ iff $s\sigma{\downarrow_{\mathcal{G}/\mathcal{A}}} \xrightarrow{\mathrm{i}}{}_{\mathcal{R}}^{rt(\overline{n}\sigma)} t\sigma{\downarrow_{\mathcal{G}/\mathcal{A}}}$ for all $\sigma : \mathcal{V}(s) \to \mathbb{N}$, i.e., $s\sigma{\downarrow_{\mathcal{G}/\mathcal{A}}}$ can be reduced to $t\sigma{\downarrow_{\mathcal{G}/\mathcal{A}}}$ in exactly $rt(n_1\sigma,\dots,n_m\sigma)$ innermost $\mathcal{R}$-steps. We omit the index $\mathcal{R}$ if it is clear from the context.

So the conjecture $\mathsf{qs}(\gamma_{\mathbf{List}}(n)) \xrightarrow{\mathrm{i}}{}^* \gamma_{\mathbf{List}}(n)$ gives rise to a rewrite lemma, since $\sigma(n) = b \in \mathbb{N}$ implies $\mathsf{qs}(\gamma_{\mathbf{List}}(b)){\downarrow_{\mathcal{G}/\mathcal{A}}} = \mathsf{qs}(\mathsf{cons}^b(\mathsf{zero},\mathsf{nil})) \xrightarrow{\mathrm{i}}{}^{3b^2+2b+1} \mathsf{cons}^b(\mathsf{zero},\mathsf{nil}) = \gamma_{\mathbf{List}}(b){\downarrow_{\mathcal{G}/\mathcal{A}}}$.

To prove rewrite lemmas, essentially we use rewriting with $\xrightarrow{\mathrm{i}}_{(\mathcal{R}\cup\mathcal{L})/(\mathcal{G}\cup\mathcal{A})}$.[6] However, this would allow us to prove lemmas that do not correspond to *innermost* rewriting with $\mathcal{R}$, if $\mathcal{R}$ contains rules with overlapping left-hand sides. Consider $\mathcal{R} = \{\mathsf{g}(\mathsf{zero}) \to \mathsf{zero}, \mathsf{f}(\mathsf{g}(x)) \to \mathsf{zero}\}$. We have $\mathsf{f}(\mathsf{g}(\gamma_{\mathbf{Nats}}(n))) \xrightarrow{\mathrm{i}}_{(\mathcal{R}\cup\mathcal{L})/(\mathcal{G}\cup\mathcal{A})} \mathsf{zero}$, but for the instantiation $[n/0]$, this would not be an innermost reduction. To avoid this, we use the following relation $\xrightarrow{\mathrm{i}}\!\!\!\rightharpoonup_{\mathcal{R}} \subseteq \xrightarrow{\mathrm{i}}_{(\mathcal{R}\cup\mathcal{L})/(\mathcal{G}\cup\mathcal{A})}$: We have $s \xrightarrow{\mathrm{i}}\!\!\!\rightharpoonup_{\mathcal{R}} t$ iff there exist a term s', a substitution σ, a position π, and a

[6] Here, we define $\xrightarrow{\mathrm{i}}_{(\mathcal{R}\cup\mathcal{L})/(\mathcal{G}\cup\mathcal{A})}$ to be the relation $\equiv_{\mathcal{G}\cup\mathcal{A}} \circ (\xrightarrow{\mathrm{i}}_{\mathcal{R}} \cup \to_{\mathcal{L}}) \circ \equiv_{\mathcal{G}\cup\mathcal{A}}$. An adaption of our approach to runtime complexity of full rewriting is obtained by considering $\to_{(\mathcal{R}\cup\mathcal{L})/(\mathcal{G}\cup\mathcal{A})}$ instead.

rule $\ell \to r \in \mathcal{R} \cup \mathcal{L}$ such that $s \equiv_{\mathcal{G}\cup\mathcal{A}} s'$, $s'|_\pi = \ell\sigma$ and $s'[r\sigma]_\pi \equiv_{\mathcal{G}\cup\mathcal{A}} t$. Moreover, if $\ell \to r \in \mathcal{R}$, then there must not be any proper non-variable subterm q of $\ell\sigma$, a (variable-renamed) rule $\ell' \to r' \in \mathcal{R}$, and a substitution σ' such that $\ell'\sigma' \equiv_{\mathcal{G}\cup\mathcal{A}} q\sigma'$. Now $\mathsf{f}(\mathsf{g}(\gamma_{\mathbf{Nats}}(n))) \not\xrightarrow{\mathrm{i}}_{\mathcal{R}} \mathsf{zero}$, because the subterm $\mathsf{g}(\gamma_{\mathbf{Nats}}(n))$ unifies with the left-hand side $\mathsf{g}(\mathsf{zero})$ modulo $\mathcal{G} \cup \mathcal{A}$.

When proving a conjecture $s \xrightarrow{\mathrm{i}}{}^{*} t$ by induction, in the step case we try to reduce $s[n/n+1]$ to $t[n/n+1]$, where one may use the rule IH: $s \to t$ as induction hypothesis. Here, the variables in IH may not be instantiated. The reason for not allowing instantiations of the non-induction variables from $\mathcal{V}(s) \setminus \{n\}$ is that such induction proofs are particularly suitable for inferring runtimes of rewrite lemmas, cf. Sect. 4.

Thus, for any rule IH: $\ell \to r$, let $s \mapsto_{\mathrm{IH}} t$ iff there exist a term s' and a position π such that $s \equiv_{\mathcal{G}\cup\mathcal{A}} s'$, $s'|_\pi = \ell$ and $s'[r]_\pi \equiv_{\mathcal{G}\cup\mathcal{A}} t$. Let $\xrightarrow{\mathrm{i}}_{(\mathcal{R},\mathrm{IH})} = \xrightarrow{\mathrm{i}}_{\mathcal{R}} \cup \mapsto_{\mathrm{IH}}$. Moreover, $\xrightarrow{\mathrm{i}}{}^{*}_{\mathcal{R}}$ (resp. $\xrightarrow{\mathrm{i}}{}^{*}_{(\mathcal{R},\mathrm{IH})}$) denotes the transitive-reflexive closure of $\xrightarrow{\mathrm{i}}_{\mathcal{R}}$ (resp. $\xrightarrow{\mathrm{i}}{}^{*}_{(\mathcal{R},\mathrm{IH})}$), where in addition $s \xrightarrow{\mathrm{i}}{}^{*}_{\mathcal{R}} s'$ and $s \xrightarrow{\mathrm{i}}{}^{*}_{(\mathcal{R},\mathrm{IH})} s'$ also hold if $s \equiv_{\mathcal{G}\cup\mathcal{A}} s'$. Thm. 9 shows which rewrite sequences are needed to prove a conjecture $s \xrightarrow{\mathrm{i}}{}^{*} t$ by induction on its induction variable n.

▶ **Theorem 9** (Proving Rewrite Lemmas). *Let $\mathcal{R}$, s, t be as in Def. 8, $n \in \mathcal{V}(s) = \{n_1, \dots, n_m\}$, and $\overline{n} = (n_1, \dots, n_m)$. If $s[n/0] \xrightarrow{\mathrm{i}}{}^{*}_{\mathcal{R}} t[n/0]$ and $s[n/n+1] \xrightarrow{\mathrm{i}}{}^{*}_{(\mathcal{R},\mathrm{IH})} t[n/n+1]$, where* IH *is the rule $s \to t$, then there is an $rt : \mathbb{N}^m \to \mathbb{N}$ such that $s \xrightarrow{\mathrm{i}}{}^{rt(\overline{n})} t$ is a rewrite lemma for $\mathcal{R}$.*

▶ **Example 10** (Proof of Rewrite Lemma). Assume that we have already proved the rewrite lemmas (5) and (6). To prove the conjecture $\mathsf{qs}(\gamma_{\mathbf{List}}(n)) \xrightarrow{\mathrm{i}}{}^{*} \gamma_{\mathbf{List}}(n)$, in the induction base we show $\mathsf{qs}(\gamma_{\mathbf{List}}(0)) \xrightarrow{\mathrm{i}}_{\mathcal{R}} \gamma_{\mathbf{List}}(0)$ and in the induction step, we obtain $\mathsf{qs}(\gamma_{\mathbf{List}}(n+1)) \xrightarrow{\mathrm{i}}{}^{*}_{\mathcal{R}} \mathsf{nil} \mathbin{+\!+} \mathsf{cons}(\mathsf{zero}, \mathsf{qs}(\gamma_{\mathbf{List}}(n))) \mapsto_{\mathrm{IH}} \mathsf{nil} \mathbin{+\!+} \mathsf{cons}(\mathsf{zero}, \gamma_{\mathbf{List}}(n)) \xrightarrow{\mathrm{i}}_{\mathcal{R}} \gamma_{\mathbf{List}}(n+1)$. Thus, there is a rewrite lemma $\mathsf{qs}(\gamma_{\mathbf{List}}(n)) \xrightarrow{\mathrm{i}}{}^{rt(n)} \gamma_{\mathbf{List}}(n)$. Sect. 4 will clarify how to find the function rt.

4 Inferring Bounds for Rewrite Lemmas

Now we show how to infer the function rt for a rewrite lemma $s \xrightarrow{\mathrm{i}}{}^{rt(\overline{n})} t$ from its proof. If $n \in \overline{n}$ was the induction variable and the induction hypothesis was applied ih times in the induction step, then we get the following recurrence equations for rt where $\widetilde{n}$ is $\overline{n}$ without the variable n:

$$rt(\overline{n}[n/0]) = ib(\widetilde{n}) \qquad \text{and} \qquad rt(\overline{n}[n/n+1]) = ih \cdot rt(\overline{n}) + is(\overline{n}) \tag{9}$$

Here, $ib(\widetilde{n})$ is the length of the reduction $s[n/0]{\downarrow_{\mathcal{G}/\mathcal{A}}} \xrightarrow{\mathrm{i}}{}^{*}_{\mathcal{R}} t[n/0]{\downarrow_{\mathcal{G}/\mathcal{A}}}$, which must exist due to the induction base. The addend $is(\overline{n})$ is the length of $s[n/n+1]{\downarrow_{\mathcal{G}/\mathcal{A}}} \xrightarrow{\mathrm{i}}{}^{*}_{\mathcal{R}} t[n/n+1]{\downarrow_{\mathcal{G}/\mathcal{A}}}$, but without those subsequences that are covered by the induction hypothesis IH. Since the non-induction variables were not instantiated in IH, $rt(\overline{n})$ is the runtime for each application of IH. To compute ib and is, for each previous rewrite lemma $s' \xrightarrow{\mathrm{i}}{}^{rt'(\overline{n}')} t'$ that was used in the proof of $s \xrightarrow{\mathrm{i}}{}^{rt(\overline{n})} t$, we assume that rt' is known. Thus, rt' can be used to infer the number of rewrite steps represented by that previous lemma. To avoid treating rules and rewrite lemmas separately, in Def. 11 we regard each rule $s \to t \in \mathcal{R}$ as a rewrite lemma $s \xrightarrow{\mathrm{i}}{}^{1} t$.

▶ **Definition 11** (ih, ib, is). Let $s \xrightarrow{\mathrm{i}}{}^{rt(\overline{n})} t$ be a rewrite lemma with an induction proof as in Thm. 9. More precisely, let $u_1 \xrightarrow{\mathrm{i}}_{\mathcal{R}} \dots \xrightarrow{\mathrm{i}}_{\mathcal{R}} u_{b+1}$ be the rewrite sequence $s[n/0] \xrightarrow{\mathrm{i}}{}^{*}_{\mathcal{R}} t[n/0]$ for the induction base and let $v_1 \xrightarrow{\mathrm{i}}_{(\mathcal{R},\mathrm{IH})} \dots \xrightarrow{\mathrm{i}}_{(\mathcal{R},\mathrm{IH})} v_{k+1}$ be the rewrite sequence $s[n/n+1] \xrightarrow{\mathrm{i}}{}^{*}_{(\mathcal{R},\mathrm{IH})} t[n/n+1]$ for the induction step, where IH: $s \to t$ is applied ih times. For $j \in \{1, \dots, b\}$, let $\ell_j \xrightarrow{\mathrm{i}}{}^{rt_j(\overline{y}_j)} r_j$ and σ_j be the rewrite lemma and substitution used to reduce u_j to u_{j+1}. Similarly for $j \in \{1, \dots, k\}$, let $p_j \to^{rt'_j(\overline{z}_j)} q_j$ and θ_j be the lemma and substitution used to reduce v_j to v_{j+1}. Then we define:

$$\mathit{ib}(\widetilde{n}) \;=\; \sum\nolimits_{j \in \{1,\dots,b\}} \mathit{rt}_j(\overline{y}_j \sigma_j) \qquad \text{and} \qquad \mathit{is}(\overline{n}) \;=\; \sum\nolimits_{j \in \{1,\dots,k\},\; p_j \to q_j \neq \mathrm{IH}} \mathit{rt}'_j(\overline{z}_j \theta_j)$$

By solving the recurrence equations (9), we can now compute rt explicitly.

▶ **Theorem 12** (Explicit Runtime of Rewrite Lemmas)**.** *Let $s \xrightarrow{\mathsf{i}}{}^{\mathit{rt}(\overline{n})} t$ be a rewrite lemma, where ih, ib, and is are as in Def. 11. Then we obtain $\mathit{rt}(\overline{n}) = \mathit{ih}^n \cdot \mathit{ib}(\widetilde{n}) + \sum_{i=0}^{n-1} \mathit{ih}^{n-1-i} \cdot \mathit{is}(\overline{n}[n/i])$.*

▶ **Example 13** (Computing rt)**.** Reconsider $\mathsf{qs}(\gamma_{\mathbf{List}}(n)) \xrightarrow{\mathsf{i}}{}^{\mathit{rt}(n)} \gamma_{\mathbf{List}}(n)$ from Ex. 10. The proof of the induction base is $\mathsf{qs}(\gamma_{\mathbf{List}}(0)) \equiv_{\mathcal{G}} \mathsf{qs}(\mathsf{nil}) \xrightarrow{\mathsf{i}}_{\mathcal{R}_{\mathsf{qs}}} \mathsf{nil} \equiv_{\mathcal{G}} \gamma_{\mathbf{List}}(0)$. Hence, $\mathit{ib} = \mathit{rt}_1 = 1$. The proof of the induction step is as follows. Here, we use that the runtime of both previously proved lemmas (5) and (6) is $3n+1$. Note that the non-overlap condition required by the relation $\xrightarrow{\mathsf{i}}_{\mathcal{R}_{\mathsf{qs}}}$ is clearly satisfied in all steps with $\xrightarrow{\mathsf{i}}_{\mathcal{R}_{\mathsf{qs}}}$ in the proof.

$$\begin{array}{rcl|l}
\mathsf{qs}(\gamma_{\mathbf{List}}(n+1)) \quad \equiv_{\mathcal{G}} & \mathsf{qs}(\mathsf{cons}(\gamma_{\mathbf{Nats}}(0), \gamma_{\mathbf{List}}(n))) & \xrightarrow{\mathsf{i}}_{\mathcal{R}_{\mathsf{qs}}} & \mathit{rt}'_1 = 1 \\
\mathsf{qs}(\mathsf{low}(\gamma_{\mathbf{Nats}}(0), \gamma_{\mathbf{List}}(n))) \;\mathsf{++}\; & \mathsf{cons}(\gamma_{\mathbf{Nats}}(0), \mathsf{qs}(\mathsf{high}(\dots))) & \to_{\mathcal{L}} & \mathit{rt}'_2(n) = 3n+1 \\
\mathsf{qs}(\gamma_{\mathbf{List}}(0)) \;\mathsf{++}\; & \mathsf{cons}(\gamma_{\mathbf{Nats}}(0), \mathsf{qs}(\mathsf{high}(\gamma_{\mathbf{Nats}}(0), \gamma_{\mathbf{List}}(n)))) & \to_{\mathcal{L}} & \mathit{rt}'_3(n) = 3n+1 \\
 & \mathsf{qs}(\gamma_{\mathbf{List}}(0)) \;\mathsf{++}\; \mathsf{cons}(\gamma_{\mathbf{Nats}}(0), \mathsf{qs}(\gamma_{\mathbf{List}}(n))) & \equiv_{\mathcal{G}} & \\
 & \mathsf{qs}(\mathsf{nil}) \;\mathsf{++}\; \mathsf{cons}(\mathsf{zero}, \mathsf{qs}(\gamma_{\mathbf{List}}(n))) & \xrightarrow{\mathsf{i}}_{\mathcal{R}_{\mathsf{qs}}} & \mathit{rt}'_4 = 1 \\
 & \mathsf{nil} \;\mathsf{++}\; \mathsf{cons}(\mathsf{zero}, \mathsf{qs}(\gamma_{\mathbf{List}}(n))) & \mapsto_{\mathrm{IH}} & \mathit{rt}'_5(n) = \mathit{rt}(n) \\
 & \mathsf{nil} \;\mathsf{++}\; \mathsf{cons}(\mathsf{zero}, \gamma_{\mathbf{List}}(n)) & \xrightarrow{\mathsf{i}}_{\mathcal{R}_{\mathsf{qs}}} & \mathit{rt}'_6 = 1 \\
 & \mathsf{cons}(\mathsf{zero}, \gamma_{\mathbf{List}}(n)) \quad \equiv_{\mathcal{G}} \quad \gamma_{\mathbf{List}}(n+1) & &
\end{array}$$

$$\begin{array}{rclcl}
\text{Hence, } \mathit{is}(n) & = & \sum\nolimits_{j \in \{1,\dots,6\},\; p_j \to q_j \neq \mathrm{IH}} \mathit{rt}'_j(\overline{z}_j \theta_j) & = & \mathit{rt}'_1 + \mathit{rt}'_2(n) + \mathit{rt}'_3(n) + \mathit{rt}'_4 + \mathit{rt}'_6 \\
 & = & 1 + (3n+1) + (3n+1) + 1 + 1 & = & 6n + 5.
\end{array}$$

In our example, we have $\mathit{ih} = 1$. So Thm. 12 implies $\mathit{rt}(n) = \mathit{ib} + \sum_{i=0}^{n-1} \mathit{is}(i) = 1 + \sum_{i=0}^{n-1} (6i+5)$ $= 3n^2 + 2n + 1$. Thus, we get the rewrite lemma (4): $\mathsf{qs}(\gamma_{\mathbf{List}}(n)) \xrightarrow{\mathsf{i}}{}^{3n^2+2n+1} \gamma_{\mathbf{List}}(n)$.

To compute asymptotic bounds for the complexity of a TRS afterwards, we have to infer asymptotic bounds for the runtime of rewrite lemmas. Based on Thm. 12, such bounds can be automatically obtained from the induction proofs of the lemmas. To ease the formulation of bounds for $\mathit{rt} : \mathbb{N}^m \to \mathbb{N}$, we define the unary function $\mathit{rt}_{\mathbb{N}} : \mathbb{N} \to \mathbb{N}$ as $\mathit{rt}_{\mathbb{N}}(n) = \mathit{rt}(n, \dots, n)$.

If the induction hypothesis was not used in the proof of a rewrite lemma (i.e., $\mathit{ih} = 0$), then we have $\mathit{rt}(\overline{n}[n/0]) = \mathit{ib}(\widetilde{n})$ and $\mathit{rt}(\overline{n}[n/n+1]) = \mathit{is}(\overline{n})$. Thus, if ib and is are polynomials of degree d_{ib} and d_{is}, respectively, then we obtain $\mathit{rt}_{\mathbb{N}}(n) \in \Omega(n^{\max\{d_{\mathit{ib}}, d_{\mathit{is}}\}})$.

If $\mathit{ih} = 1$, then Thm. 12 implies $\mathit{rt}(\overline{n}) = \mathit{ib}(\widetilde{n}) + \sum_{i=0}^{n-1} \mathit{is}(\overline{n}[n/i])$. Again, let ib and is be polynomials of degree d_{ib} and d_{is}, respectively. Then $\mathit{is}(\overline{n}) = t_0 + t_1 n + t_2 n^2 + \ldots + t_{d_{\mathit{is}}} n^{d_{\mathit{is}}}$, where the t_j are polynomials of degree at most $d_{\mathit{is}} - j$ containing variables from $\widetilde{n}$. Hence, $\mathit{rt}(\overline{n}) =$

$$\mathit{ib}(\widetilde{n}) + \sum_{i=0}^{n-1} (t_0 + t_1 i + t_2 i^2 + \ldots + t_{d_{\mathit{is}}} i^{d_{\mathit{is}}}) = \mathit{ib}(\widetilde{n}) + t_0 \cdot \sum_{i=0}^{n-1} i^0 + t_1 \cdot \sum_{i=0}^{n-1} i^1 + t_2 \cdot \sum_{i=0}^{n-1} i^2 + \ldots + t_{d_{\mathit{is}}} \cdot \sum_{i=0}^{n-1} i^{d_{\mathit{is}}}.$$

By Faulhaber's formula [14], for any $e \in \mathbb{N}$, $\sum_{i=0}^{n-1} i^e$ is a polynomial over the variable n of degree $e+1$. For example if $e = 1$, then $\sum_{i=0}^{n-1} i^1 = \frac{n \cdot (n-1)}{2}$ has degree 2. By taking also the degree d_{ib} of ib into account, rt has degree $\max\{d_{\mathit{ib}}, d_{\mathit{is}} + 1\}$, i.e., $\mathit{rt}_{\mathbb{N}}(n) \in \Omega(n^{\max\{d_{\mathit{ib}}, d_{\mathit{is}}+1\}})$.

Finally we consider the case where the induction hypothesis was used several times, i.e., $\mathit{ih} > 1$. By construction we always have $\mathit{is}(\overline{n}) \geq 1$ (since the induction step cannot only consist of applying the induction hypothesis). Thus, Thm. 12 implies $\mathit{rt}(\overline{n}) \geq \sum_{i=0}^{n-1} \mathit{ih}^{n-1-i} = \sum_{j=0}^{n-1} \mathit{ih}^j = \frac{\mathit{ih}^n - 1}{\mathit{ih} - 1}$. So $\mathit{rt}_{\mathbb{N}}(n) \in \Omega(\mathit{ih}^n)$, i.e., the runtime of the rewrite lemma is exponential.

▶ **Theorem 14** (Asymptotic Runtime of Rewrite Lemmas)**.** *Let $s \xrightarrow{\mathsf{i}}{}^{\mathit{rt}(\overline{n})} t$ be a rewrite lemma with ih, ib, and is as in Def. 11. Moreover, let ib and is be polynomials of degree d_{ib} and d_{is}.*

- *If* $\mathit{ih} = 0$, *then* $\mathit{rt}_{\mathbb{N}}(n) \in \Omega(n^{\max\{d_{\mathit{ib}}, d_{\mathit{is}}\}})$.
- *If* $\mathit{ih} = 1$, *then* $\mathit{rt}_{\mathbb{N}}(n) \in \Omega(n^{\max\{d_{\mathit{ib}}, d_{\mathit{is}}+1\}})$.
- *If* $\mathit{ih} > 1$, *then* $\mathit{rt}_{\mathbb{N}}(n) \in \Omega(\mathit{ih}^n)$.

▶ **Example 15** (Exponential Runtime). Consider the TRS $\mathcal{R}_{exp}$ with the rules $\mathsf{f}(\mathsf{succ}(x), \mathsf{succ}(x)) \to \mathsf{f}(\mathsf{f}(x,x), \mathsf{f}(x,x))$ and $\mathsf{f}(\mathsf{zero}, \mathsf{zero}) \to \mathsf{zero}$. Our approach speculates and proves the rewrite lemma $\mathsf{f}(\gamma_{\mathbf{Nats}}(n), \gamma_{\mathbf{Nats}}(n)) \xrightarrow{\mathrm{i}}{}^{\mathit{rt}(n)} \mathsf{zero}$. For the induction base, we have $\mathsf{f}(\gamma_{\mathbf{Nats}}(0), \gamma_{\mathbf{Nats}}(0)) \equiv_{\mathcal{G}} \mathsf{f}(\mathsf{zero}, \mathsf{zero}) \xrightarrow{\mathrm{i}}_{\mathcal{R}_{exp}} \mathsf{zero}$ and thus $\mathit{ib} = 1$. The induction step is proved as follows:

$$
\begin{array}{rcll}
\mathsf{f}(\gamma_{\mathbf{Nats}}(n+1), \gamma_{\mathbf{Nats}}(n+1)) & \equiv_{\mathcal{G}} & \mathsf{f}(\mathsf{succ}(\gamma_{\mathbf{Nats}}(n)), \mathsf{succ}(\gamma_{\mathbf{Nats}}(n))) & \xrightarrow{\mathrm{i}}_{\mathcal{R}_{exp}} \quad \mathit{rt}'_1 = 1 \\
 & & \mathsf{f}(\mathsf{f}(\gamma_{\mathbf{Nats}}(n), \gamma_{\mathbf{Nats}}(n)), \mathsf{f}(\gamma_{\mathbf{Nats}}(n), \gamma_{\mathbf{Nats}}(n))) & \mapsto^2_{\mathrm{IH}} \\
 & & \mathsf{f}(\mathsf{zero}, \mathsf{zero}) & \xrightarrow{\mathrm{i}}_{\mathcal{R}_{exp}} \quad \mathit{rt}'_4 = 1 \\
 & & \mathsf{zero} &
\end{array}
$$

Thus, $\mathit{ih} = 2$ and $\mathit{is}(n)$ is the constant 2 for all $n \in \mathbb{N}$. Hence, by Thm. 14 we have $\mathit{rt}(n) \in \Omega(2^n)$. Indeed, Thm. 12 implies $\mathit{rt}(n) = 2^n + \sum_{i=0}^{n-1} 2^{n-1-i} \cdot 2 = 2^{n+1} + 2^n - 2$.

5 Inferring Bounds for TRSs

We now use rewrite lemmas to infer lower bounds for the *innermost runtime complexity* $\mathrm{irc}_{\mathcal{R}}$ of a TRS $\mathcal{R}$. To define $\mathrm{irc}_{\mathcal{R}}$, the *derivation height* of a term t w.r.t. a relation $\to$ is the length of the longest $\to$-sequence starting with t, i.e., $\mathrm{dh}(t, \to) = \sup\{ m \mid \exists t' \in \mathcal{T}(\Sigma, \mathcal{V}),\, t \to^m t' \}$, cf. [12]. Here, for any $M \subseteq \mathbb{N} \cup \{\omega\}$, $\sup M$ is the least upper bound of M and $\sup \varnothing = 0$. Since we only regard finite TRSs, $\mathrm{dh}(t, \xrightarrow{\mathrm{i}}_{\mathcal{R}}) = \omega$ iff t starts an infinite sequence of $\xrightarrow{\mathrm{i}}_{\mathcal{R}}$-steps. So as in [16], dh treats terminating and non-terminating terms in a uniform way.

When analyzing the complexity of *programs*, one is interested in evaluations of *basic terms* $f(t_1, \ldots, t_k)$ where a defined symbol $f \in \Sigma_{def}(\mathcal{R})$ is applied to data objects $t_1, \ldots, t_k \in \mathcal{T}(\Sigma_{con}(\mathcal{R}), \mathcal{V})$. The *innermost runtime complexity function* $\mathrm{irc}_{\mathcal{R}}$ corresponds to the usual notion of "complexity" for programs. It maps any $n \in \mathbb{N}$ to the length of the longest sequence of $\xrightarrow{\mathrm{i}}_{\mathcal{R}}$-steps starting with a basic term t with $|t| \leq n$. Here, the *size* of a term is $|x| = 1$ for $x \in \mathcal{V}$ and $|f(t_1, \ldots, t_k)| = 1 + |t_1| + \ldots + |t_k|$, and $\mathcal{T}_B$ is the set of all basic terms.

▶ **Definition 16** (Innermost Runtime Complexity $\mathrm{irc}_{\mathcal{R}}$ [11]). For a TRS $\mathcal{R}$, its *innermost runtime complexity function* $\mathrm{irc}_{\mathcal{R}} : \mathbb{N} \to \mathbb{N} \cup \{\omega\}$ is $\mathrm{irc}_{\mathcal{R}}(n) = \sup\{ \mathrm{dh}(t, \xrightarrow{\mathrm{i}}_{\mathcal{R}}) \mid t \in \mathcal{T}_B, |t| \leq n \}$.

In Sect. 4 we computed the length $\mathit{rt}(\overline{n})$ of the rewrite sequences represented by a rewrite lemma $s \xrightarrow{\mathrm{i}}{}^{\mathit{rt}(\overline{n})} t$, where $\mathcal{V}(s) = \overline{n}$. However, $\mathrm{irc}_{\mathcal{R}}$ is defined w.r.t. the size of the start term of a rewrite sequence. Thus, to obtain a lower bound for $\mathrm{irc}_{\mathcal{R}}$ from $\mathit{rt}(\overline{n})$, for any $\sigma : \mathcal{V}(s) \to \mathbb{N}$ one has to take the relation between $\overline{n}\sigma$ and the size of the start term $s\sigma{\downarrow_{\mathcal{G}/\mathcal{A}}}$ into account. Note that our approach in Sect. 2 only speculates lemmas where s has the form $f(\gamma_{\tau_1}(s_1), \ldots, \gamma_{\tau_k}(s_k))$. Here, $f \in \Sigma_{def}(\mathcal{R})$, $s_1, \ldots, s_k$ are polynomials over $\overline{n}$, and $\tau_1, \ldots, \tau_k$ are simply structured types. For any τ_i, let $d_{\tau_i} : \rho_1 \times \cdots \times \rho_b \to \tau$ be τ_i's recursive constructor. Then for any $n \in \mathbb{N}$, Def. 3 implies $|\gamma_{\tau_i}(n){\downarrow_{\mathcal{G}/\mathcal{A}}}| = \mathit{sz}_{\tau_i}(n)$ for $\mathit{sz}_{\tau_i} : \mathbb{N} \to \mathbb{N}$ with

$$\mathit{sz}_{\tau_i}(n) = |\gamma_{\tau_i}(0){\downarrow_{\mathcal{G}/\mathcal{A}}}| + n \cdot \big(1 + |\gamma_{\rho_1}(0){\downarrow_{\mathcal{G}/\mathcal{A}}}| + \cdots + |\gamma_{\rho_b}(0){\downarrow_{\mathcal{G}/\mathcal{A}}}| - |\gamma_{\tau_i}(0){\downarrow_{\mathcal{G}/\mathcal{A}}}|\big).$$

The reason is that $\gamma_{\tau_i}(n){\downarrow_{\mathcal{G}/\mathcal{A}}}$ contains n occurrences of d_{τ_i} and of each $\gamma_{\rho_1}(0){\downarrow_{\mathcal{G}/\mathcal{A}}}, \ldots, \gamma_{\rho_b}(0){\downarrow_{\mathcal{G}/\mathcal{A}}}$ except $\gamma_{\tau_i}(0){\downarrow_{\mathcal{G}/\mathcal{A}}}$, and just one occurrence of $\gamma_{\tau_i}(0){\downarrow_{\mathcal{G}/\mathcal{A}}}$. For instance, $|\gamma_{\mathbf{Nats}}(n){\downarrow_{\mathcal{G}/\mathcal{A}}}|$ is $\mathit{sz}_{\mathbf{Nats}}(n) = |\gamma_{\mathbf{Nats}}(0){\downarrow_{\mathcal{G}/\mathcal{A}}}| + n \cdot (1 + |\gamma_{\mathbf{Nats}}(0){\downarrow_{\mathcal{G}/\mathcal{A}}}| - |\gamma_{\mathbf{Nats}}(0){\downarrow_{\mathcal{G}/\mathcal{A}}}|) = |\mathsf{zero}| + n = 1 + n$ and $|\gamma_{\mathbf{List}}(n){\downarrow_{\mathcal{G}/\mathcal{A}}}|$ is $\mathit{sz}_{\mathbf{List}}(n) = |\gamma_{\mathbf{List}}(0){\downarrow_{\mathcal{G}/\mathcal{A}}}| + n \cdot (1 + |\gamma_{\mathbf{Nats}}(0){\downarrow_{\mathcal{G}/\mathcal{A}}}|) = |\mathsf{nil}| + n \cdot (1 + |\mathsf{zero}|) = 1 + n \cdot 2$. Consequently, the size of $s{\downarrow_{\mathcal{G}/\mathcal{A}}} = f(\gamma_{\tau_1}(s_1), \ldots, \gamma_{\tau_k}(s_k)){\downarrow_{\mathcal{G}/\mathcal{A}}}$ with $\mathcal{V}(s) = \overline{n}$ is given by the following function $\mathit{sz} : \mathbb{N}^m \to \mathbb{N}$:

$$sz(\overline{n}) = 1 + sz_{\tau_1}(s_1) + \cdots + sz_{\tau_k}(s_k)$$

For instance, the term $\mathsf{qs}(\gamma_{\mathbf{List}}(n))\downarrow_{\mathcal{G}/\mathcal{A}} = \mathsf{qs}(\mathsf{cons}^n(\mathsf{zero},\mathsf{nil}))$ has the size $sz(n) = 1 + sz_{\mathbf{List}}(n) = 2n+2$. Since $|\gamma_\tau(0)\downarrow_{\mathcal{G}/\mathcal{A}}|$ is a constant for each type τ, sz is a polynomial whose degree is given by the maximal degree of the polynomials $s_1,\dots,s_k$.

So the rewrite lemma (4) for qs states that there are terms of size $sz(n) = 2n+2$ with reductions of length $rt(n) = 3n^2+2n+1$. To determine a lower bound for $\mathrm{irc}_{\mathcal{R}_{\mathsf{qs}}}$, we construct an inverse function sz^{-1} with $(sz \circ sz^{-1})(n) = n$. In our example where $sz(n) = 2n+2$, we have $sz^{-1}(n) = \frac{n-2}{2}$ if n is even. So there are terms of size $sz(sz^{-1}(n)) = n$ with reductions of length $rt(sz^{-1}(n)) = rt(\frac{n-2}{2}) = \frac{3}{4}n^2 - 2n + 2$. Since multivariate polynomials $sz(n_1,\dots,n_m)$ cannot be inverted, we invert the unary function $sz_{\mathbb{N}} : \mathbb{N} \to \mathbb{N}$ with $sz_{\mathbb{N}}(n) = sz(n,\dots,n)$ instead.

Of course, inverting $sz_{\mathbb{N}}$ fails if $sz_{\mathbb{N}}$ is not injective. However, the conjectures speculated in Sect. 2 only contain polynomials with natural coefficients. Then, $sz_{\mathbb{N}}$ is always strictly monotonically increasing. So we only proceed if there is a $sz_{\mathbb{N}}^{-1} : \mathrm{img}(sz_{\mathbb{N}}) \to \mathbb{N}$ where $(sz_{\mathbb{N}} \circ sz_{\mathbb{N}}^{-1})(n) = n$ holds for all $n \in \mathrm{img}(sz_{\mathbb{N}}) = \{n \in \mathbb{N} \mid \exists v \in \mathbb{N}.\ sz_{\mathbb{N}}(v) = n\}$. To extend $sz_{\mathbb{N}}^{-1}$ to a function on $\mathbb{N}$, for any (total) function $h : M \to \mathbb{N}$ with $M \subseteq \mathbb{N}$, we define $\lfloor h \rfloor(n) : \mathbb{N} \to \mathbb{N}$ by:

$$\lfloor h \rfloor(n) = h(\max\{n' \mid n' \in M, n' \le n\}),\ \text{if } n \ge \min(M) \quad \text{and} \quad \lfloor h \rfloor(n) = 0,\ \text{otherwise}$$

Using this notation, the following theorem states how we can derive lower bounds for $\mathrm{irc}_{\mathcal{R}}$.

▶ **Theorem 17** (Explicit Lower Bounds for $\mathrm{irc}_{\mathcal{R}}$)**.** *Let $s \xrightarrow{\mathrm{i}}{}^{rt(n_1,\dots,n_m)} t$ be a rewrite lemma for $\mathcal{R}$, let $sz : \mathbb{N}^m \to \mathbb{N}$ be a function such that $sz(b_1,\dots,b_m)$ is the size of $s[n_1/b_1,\dots,n_m/b_m]\downarrow_{\mathcal{G}/\mathcal{A}}$ for all $b_1,\dots,b_m \in \mathbb{N}$, and let $sz_{\mathbb{N}}$'s inverse function $sz_{\mathbb{N}}^{-1}$ exist. Then $rt_{\mathbb{N}} \circ \lfloor sz_{\mathbb{N}}^{-1} \rfloor$ is a lower bound for $\mathrm{irc}_{\mathcal{R}}$, i.e., $(rt_{\mathbb{N}} \circ \lfloor sz_{\mathbb{N}}^{-1} \rfloor)(n) \le \mathrm{irc}_{\mathcal{R}}(n)$ holds for all $n \in \mathbb{N}$ with $n \ge \min(\mathrm{img}(sz_{\mathbb{N}}))$.*

So for the rewrite lemma (4) for qs where $sz_{\mathbb{N}}(n) = 2n+2$, we have $\lfloor sz_{\mathbb{N}}^{-1} \rfloor(n) = \lfloor \frac{n-2}{2} \rfloor \ge \frac{n-3}{2}$ and $\mathrm{irc}_{\mathcal{R}_{\mathsf{qs}}}(n) \ge rt(\lfloor sz_{\mathbb{N}}^{-1} \rfloor(n)) \ge rt(\frac{n-3}{2}) = \frac{3}{4}n^2 - \frac{7}{2}n + \frac{19}{4}$ for all $n \ge 2$.

However, even if $sz_{\mathbb{N}}^{-1}$ exists, finding resp. approximating $sz_{\mathbb{N}}^{-1}$ automatically can be non-trivial in general. Therefore, we now show how to obtain an asymptotic lower bound for $\mathrm{irc}_{\mathcal{R}}$ directly from a rewrite lemma $f(\gamma_{\tau_1}(s_1),\dots,\gamma_{\tau_k}(s_k)) \xrightarrow{\mathrm{i}}{}^{rt(\overline{n})} t$ without constructing $sz_{\mathbb{N}}^{-1}$. As mentioned, if e is the maximal degree of the polynomials $s_1,\dots,s_k$, then sz is also a polynomial of degree e and thus, $sz_{\mathbb{N}}(n) \in \mathcal{O}(n^e)$. Moreover, from the induction proof of the rewrite lemma we obtain an asymptotic lower bound for $rt_{\mathbb{N}}$, cf. Thm. 14. Using these bounds, Lemma 18 can be used to infer an asymptotic lower bound for $\mathrm{irc}_{\mathcal{R}}$ directly.

▶ **Lemma 18** (Asymptotic Bounds for Function Composition)**.** *Let $rt_{\mathbb{N}}, sz_{\mathbb{N}} : \mathbb{N} \to \mathbb{N}$ where $sz_{\mathbb{N}} \in \mathcal{O}(n^e)$ for some $e \ge 1$ and where $sz_{\mathbb{N}}$ is strictly monotonically increasing.*

- *If $rt_{\mathbb{N}}(n) \in \Omega(n^d)$ with $d \ge 0$, then $(rt_{\mathbb{N}} \circ \lfloor sz_{\mathbb{N}}^{-1} \rfloor)(n) \in \Omega(n^{\frac{d}{e}})$.*
- *If $rt_{\mathbb{N}}(n) \in \Omega(b^n)$ with $b \ge 1$, then $(rt_{\mathbb{N}} \circ \lfloor sz_{\mathbb{N}}^{-1} \rfloor)(n) \in \Omega(b^{\sqrt[e]{n}})$.*

So for the rewrite lemma $\mathsf{qs}(\gamma_{\mathbf{List}}(n)) \xrightarrow{\mathrm{i}}{}^{rt(n)} \gamma_{\mathbf{List}}(n)$ where $rt_{\mathbb{N}} = rt$ and $sz_{\mathbb{N}} = sz$, we only need the asymptotic bounds $sz(n) \in \mathcal{O}(n)$ and $rt(n) \in \Omega(n^2)$, to conclude $\mathrm{irc}_{\mathcal{R}_{\mathsf{qs}}}(n) \in \Omega(n^{\frac{2}{1}}) = \Omega(n^2)$, i.e., to prove that the quicksort TRS has at least quadratic complexity.

So while Thm. 17 explains how to find concrete lower bounds for $\mathrm{irc}_{\mathcal{R}}$ (if $sz_{\mathbb{N}}$ can be inverted), the following theorem summarizes our results on asymptotic lower bounds for $\mathrm{irc}_{\mathcal{R}}$. To this end, we combine Thm. 14 on the inference of asymptotic bounds for rt with Lemma 18.

▶ **Theorem 19** (Asymptotic Lower Bounds for $\mathrm{irc}_{\mathcal{R}}$)**.** *Let $s \xrightarrow{\mathrm{i}}{}^{rt(\overline{n})} t$ be a rewrite lemma for $\mathcal{R}$ and let $sz : \mathbb{N}^m \to \mathbb{N}$ be a function such that $sz(b_1,\dots,b_m)$ is the size of $s[n_1/b_1,\dots,n_m/b_m]\downarrow_{\mathcal{G}/\mathcal{A}}$ for all $b_1,\dots,b_m \in \mathbb{N}$, where $sz_{\mathbb{N}}(n) \in \mathcal{O}(n^e)$ for some $e \ge 1$ and $sz_{\mathbb{N}}$ is strictly monotonically increasing. Furthermore, let ih, ib, and is be defined as in Def. 11.*

1. *If $ih = 0$ and ib and is are polynomials of degree d_{ib} and d_{is}, then $\mathrm{irc}_{\mathcal{R}}(n) \in \Omega(n^{\frac{\max\{d_{ib},d_{is}\}}{e}})$.*

2. *If* $\mathit{ih} = 1$ *and* ib *and* is *are polynomials of degree* d_{ib} *and* d_{is}*, then* $\mathrm{irc}_{\mathcal{R}}(n) \in \Omega(n^{\frac{\max\{d_{\mathit{ib}}, d_{\mathit{is}}+1\}}{e}})$.
3. *If* $\mathit{ih} > 1$*, then* $\mathrm{irc}_{\mathcal{R}}(n) \in \Omega(\mathit{ih}^{\sqrt[e]{n}})$.

6 Preprocessing TRSs by Argument Filtering

A drawback of our approach is that generator functions only represent homogeneous data objects (e.g., lists or trees where all elements have the same value zero). To prove lower complexity bounds also in cases where one needs other forms of rewrite lemmas, we use *argument filtering* [2] to remove certain argument positions of function symbols.

▶ **Example 20** (Argument Filtering)**.** Consider the following TRS $\mathcal{R}_{\mathsf{intlist}}$:

$$\mathsf{intlist}(\mathsf{zero}) \to \mathsf{nil} \qquad\qquad \mathsf{intlist}(\mathsf{succ}(x)) \to \mathsf{cons}(x, \mathsf{intlist}(x))$$

We have $\mathsf{intlist}(\mathsf{succ}^n(\mathsf{zero})) \xrightarrow{\mathrm{i}}{}^{n+1} \mathsf{cons}(\mathsf{succ}^{n-1}(\mathsf{zero}), \ldots \mathsf{cons}(\mathsf{succ}(\mathsf{zero}), \mathsf{cons}(\mathsf{zero}, \mathsf{nil})))$ for all $n \in \mathbb{N}$. However, the inhomogeneous list on the right-hand side cannot be expressed in a rewrite lemma. Filtering away the first argument of cons yields the TRS $(\mathcal{R}_{\mathsf{intlist}})_{\setminus(\mathsf{cons},1)}$:

$$\mathsf{intlist}(\mathsf{zero}) \to \mathsf{nil} \qquad\qquad \mathsf{intlist}(\mathsf{succ}(x)) \to \mathsf{cons}(\mathsf{intlist}(x))$$

For this TRS, our approach can speculate and prove the rewrite lemma $\mathsf{intlist}(\gamma_{\mathbf{Nats}}(n)) \xrightarrow{\mathrm{i}}{}^{n+1} \gamma_{\mathbf{List}}(n)$, i.e., $\mathsf{intlist}(\mathsf{succ}^n(\mathsf{zero})) \xrightarrow{\mathrm{i}}{}^{n+1} \mathsf{cons}^n(\mathsf{nil})$. From this rewrite lemma, one can infer $n - 1 \le \mathrm{irc}_{(\mathcal{R}_{\mathsf{intlist}})_{\setminus(\mathsf{cons},1)}}(n)$ for all $n \ge 2$ resp. $\mathrm{irc}_{(\mathcal{R}_{\mathsf{intlist}})_{\setminus(\mathsf{cons},1)}}(n) \in \Omega(n)$.

Def. 21 introduces the concept of argument filtering for terms and TRSs formally.

▶ **Definition 21** (Argument Filtering)**.** Let Σ be a signature with $f \in \Sigma$, $\mathrm{ar}_\Sigma(f) = k$, and let $i \in \{1, \ldots, k\}$. Let $\Sigma_{\setminus(f,i)}$ be like Σ, but with $\mathrm{ar}_{\Sigma_{\setminus(f,i)}}(f) = k - 1$. For any term $t \in \mathcal{T}(\Sigma, \mathcal{V})$, we define the term $t_{\setminus(f,i)} \in \mathcal{T}(\Sigma_{\setminus(f,i)}, \mathcal{V})$ resulting from *filtering* away the i-th argument of f:

$$t_{\setminus(f,i)} = \begin{cases} t, & \text{if } t \text{ is a variable} \\ f(\,(t_1)_{\setminus(f,i)}, \ldots, (t_{i-1})_{\setminus(f,i)}, (t_{i+1})_{\setminus(f,i)}, \ldots, (t_k)_{\setminus(f,i)}\,), & \text{if } t = f(t_1, \ldots, t_k) \\ g(\,(t_1)_{\setminus(f,i)}, \ldots, (t_b)_{\setminus(f,i)}\,), & \text{if } t = g(t_1, \ldots, t_b) \text{ for } g \ne f \end{cases}$$

Let $\mathcal{R}$ be a TRS over Σ. Then we define $\mathcal{R}_{\setminus(f,i)} = \{\ell_{\setminus(f,i)} \to r_{\setminus(f,i)} \mid \ell \to r \in \mathcal{R}\}$.

However, a lower bound for the runtime of $\mathcal{R}_{\setminus(f,i)}$ does not imply a lower bound for $\mathcal{R}$ if the argument that is filtered away influences the control flow of the evaluation. Thus, several conditions have to be imposed to ensure that argument filtering is sound for lower bounds:

(a) Argument filtering must not remove function symbols on left-hand sides of rules.
An argument may not be filtered away if it is used for non-trivial pattern matching (i.e., if there is a left-hand side of a rule where the i-th argument of f is not a variable). As an example, consider $\mathcal{R} = \{\mathsf{f}(\mathsf{cons}(\mathsf{true}, \mathit{xs})) \to \mathsf{f}(\mathsf{cons}(\mathsf{false}, \mathit{xs}))\}$ where $\mathrm{irc}_{\mathcal{R}}(n) \le 1$ for all n. But if one filters away the first argument of cons, then one obtains the non-terminating rule $\mathsf{f}(\mathsf{cons}(\mathit{xs})) \to \mathsf{f}(\mathsf{cons}(\mathit{xs}))$, i.e., $\mathrm{irc}_{\mathcal{R}_{\setminus(\mathsf{cons},1)}}(n) = \omega$ for $n \ge 3$.

(b) The TRS must be left-linear.
To illustrate this, consider $\mathcal{R} = \{\mathsf{f}(\mathit{xs}, \mathit{xs}) \to \mathsf{f}(\mathsf{cons}(\mathsf{true}, \mathit{xs}), \mathsf{cons}(\mathsf{false}, \mathit{xs}))\}$, where again $\mathrm{irc}_{\mathcal{R}}(n) \le 1$. But filtering away the first argument of cons yields the non-terminating rule $\mathsf{f}(\mathit{xs}, \mathit{xs}) \to \mathsf{f}(\mathsf{cons}(\mathit{xs}), \mathsf{cons}(\mathit{xs}))$, i.e., $\mathrm{irc}_{\mathcal{R}_{\setminus(\mathsf{cons},1)}}(n) = \omega$ for $n \ge 3$.

(c) Argument filtering must not result in free variables on right-hand sides of rules.
The reason is that otherwise, argument filtering might again turn terminating TRSs into non-terminating ones. For instance, consider $\mathcal{R} = \{\mathsf{f}(\mathsf{cons}(x, \mathit{xs})) \to \mathsf{f}(\mathit{xs})\}$ where $\mathrm{irc}_{\mathcal{R}}(n) = \lfloor \frac{n}{2} \rfloor - 1$. But if one filters away the second argument of cons, then one gets the rule $\mathsf{f}(\mathsf{cons}(x)) \to \mathsf{f}(\mathit{xs})$ whose runtime is unbounded, i.e., $\mathrm{irc}_{\mathcal{R}_{\setminus(\mathsf{cons},2)}}(n) = \omega$ for $n \ge 3$.

Thm. 22 states that (a) - (c) are indeed sufficient for the soundness of argument filtering. To infer a lower bound for $\mathrm{irc}_{\mathcal{R}}$ from a bound for $\mathrm{irc}_{\mathcal{R}_{\setminus(f,i)}}$, we have to take into account that filtering changes the size of terms. As an example, consider $\mathcal{R} = \{\mathsf{f}(x) \to \mathsf{a}\}$. Here, we have $\mathrm{irc}_{\mathcal{R}_{\setminus(\mathsf{f},1)}}(1) = 1$ due to the rewrite sequence $\mathsf{f} \xrightarrow{\mathrm{i}}_{\mathcal{R}_{\setminus(\mathsf{f},1)}} \mathsf{a}$. The corresponding rewrite sequence in the original TRS $\mathcal{R}$ is $\mathsf{f}(x) \xrightarrow{\mathrm{i}}_{\mathcal{R}} \mathsf{a}$. Thus, $\mathrm{irc}_{\mathcal{R}}(2) = 1$, but all terms of size 1 are normal forms of $\mathcal{R}$, i.e., $\mathrm{irc}_{\mathcal{R}}(1) = 0$. So $\mathrm{irc}_{\mathcal{R}_{\setminus(f,i)}}(n) \leq \mathrm{irc}_{\mathcal{R}}(n)$ does not hold in general. Nevertheless, for any rewrite sequence of $\mathcal{R}_{\setminus(f,i)}$ starting with a term t, there is a corresponding rewrite sequence of $\mathcal{R}$ starting with a term[7] s where $|s| \leq 2 \cdot |t|$. Thus, if we have derived a lower bound $p(n)$ for $\mathrm{irc}_{\mathcal{R}_{\setminus(f,i)}}(n)$, we can use $p(\frac{n}{2})$ as a lower bound for $\mathrm{irc}_{\mathcal{R}}(n)$. Hence, in Ex. 20, we obtain $\frac{n}{2} - 1 \leq \mathrm{irc}_{\mathcal{R}_{\mathsf{intlist}}}(n)$ for all $n \geq 4$ resp. $\mathrm{irc}_{\mathcal{R}_{\mathsf{intlist}}}(n) \in \Omega(n)$.

▶ **Theorem 22** (Soundness of Argument Filtering). *Let $f \in \Sigma$, $\mathrm{ar}_\Sigma(f) = k$, and $i \in \{1, \ldots, k\}$. Moreover, let $\mathcal{R}$ be a TRS over Σ where the following conditions hold for all rules $\ell \to r \in \mathcal{R}$:*
(a) *If $f(t_1, \ldots, t_k)$ is a subterm of ℓ, then $t_i \in \mathcal{V}$.*
(b) *For any $x \in \mathcal{V}$, there is at most one position $\pi \in \mathrm{pos}(\ell)$ such that $\ell|_\pi = x$.*
(c) *$\mathcal{V}(r_{\setminus(f,i)}) \subseteq \mathcal{V}(\ell_{\setminus(f,i)})$.*
Then for all $n \in \mathbb{N}$, we have $\mathrm{irc}_{\mathcal{R}_{\setminus(f,i)}}(\frac{n}{2}) \leq \mathrm{irc}_{\mathcal{R}}(n)$.

In our implementation, as a heuristic we always perform argument filtering if it is permitted by Thm. 22, except for cases where filtering removes defined function symbols on right-hand sides of rules. As an example, consider $\mathcal{R} = \{\mathsf{a} \to \mathsf{f}(\mathsf{a}, \mathsf{b})\}$ where $\mathrm{irc}_{\mathcal{R}}(n) = \omega$ for $n \geq 1$. If one filters away f's first argument, then one obtains $\mathsf{a} \to \mathsf{f}(\mathsf{b})$ and thus, $\mathrm{irc}_{\mathcal{R}_{\setminus(\mathsf{f},1)}}(n) = 1$ for $n \geq 1$. So here, argument filtering is sound, but it results in significantly worse lower bounds.

7 Indefinite Rewrite Lemmas

Our technique often fails if the analyzed TRS is not completely defined, i.e., if there are normal forms containing defined symbols. As an example, the runtime complexity of $\mathcal{R}_{in} = \{\mathsf{f}(\mathsf{succ}(x)) \to \mathsf{succ}(\mathsf{f}(x))\}$ is linear due to the rewrite sequences $\mathsf{f}(\mathsf{succ}^n(\mathsf{zero})) \xrightarrow{\mathrm{i}}^n \mathsf{succ}^n(\mathsf{f}(\mathsf{zero}))$. However, the term $\mathsf{succ}^n(\mathsf{f}(\mathsf{zero}))$ on the right-hand side contains f and thus, it cannot be represented in a rewrite lemma. Therefore, we now also allow *indefinite* conjectures and rewrite lemmas with unknown right-hand sides. Then for our example, we could speculate the indefinite conjecture $\mathsf{f}(\gamma_{\mathbb{N}}(n)) \xrightarrow{\mathrm{i}}^* \star$, which gives rise to the indefinite rewrite lemma $\mathsf{f}(\gamma_{\mathbb{N}}(n)) \xrightarrow{\mathrm{i}}^n \star$, where $\star$ represents an arbitrary unknown term. To distinguish indefinite conjectures and rewrite lemmas from ordinary ones, we call the latter *definite*.

Recall that when speculating conjectures in Sect. 2, we built a narrowing tree for a term $s = f(\ldots)$ and obtained a sample point (t, σ, d) whenever we reached a normal form t. When speculating indefinite conjectures, we do not narrow in order to reach normal forms, but we create a sample point (σ, d) after each application of a recursive f-rule. Here, σ is again the substitution and d is the recursion depth of the path. Note that while previously proven lemmas $\mathcal{L}$ may be used during narrowing, we do not use previous indefinite rewrite lemmas, since they do not yield any information on the terms resulting from rewriting.

▶ **Example 23** (Speculating Indefinite Conjectures). For $\mathcal{R}_{in}$, we narrow the term $s = \mathsf{f}(\gamma_{\mathbf{Nats}}(x))$. We get $\mathsf{f}(\gamma_{\mathbf{Nats}}(x)) \rightsquigarrow \mathsf{succ}(\mathsf{f}(\gamma_{\mathbf{Nats}}(x')))$ with the substitution $\sigma_1 = [x/x' + 1]$. Since we applied a recursive f-rule once, we construct the sample point $(\sigma_1, 1)$. We continue narrowing and get $\mathsf{succ}(\mathsf{f}(\gamma_{\mathbf{Nats}}(x'))) \rightsquigarrow \mathsf{succ}(\mathsf{succ}(\mathsf{f}(\gamma_{\mathbf{Nats}}(x''))))$ with the substitution $\sigma_2 =$

[7] The term s can be obtained from t by adding a variable as the i-th argument for any f occurring in t.

$[x'/x''+1]$ and recursion depth 2. Since $\sigma_2 \circ \sigma_1$ corresponds to $[x/x''+2]$, this yields the sample point $([x/x''+2], 2)$. Another narrowing step results in the sample point $([x/x'''+3], 3)$.

These sample points represent the sample conjectures $\mathsf{f}(\gamma_{\mathbf{Nats}}(x'+1)) \xrightarrow{\mathrm{i}}^* \star$, $\mathsf{f}(\gamma_{\mathbf{Nats}}(x''+2)) \xrightarrow{\mathrm{i}}^* \star$, $\mathsf{f}(\gamma_{\mathbf{Nats}}(x'''+3)) \xrightarrow{\mathrm{i}}^* \star$ that are identical up to the occurring numbers and variable names. Thus, they are suitable for generalization. As in Sect. 2, we replace the numbers in the sample conjectures by a polynomial *pol* in one variable n that stands for the recursion depth. This leads to $\mathsf{f}(\gamma_{\mathbf{Nats}}(x + pol)) \xrightarrow{\mathrm{i}}^* \star$ and the constraints $pol(1) = 1$, $pol(2) = 2$, $pol(3) = 3$. A solution is $pol = n$ and thus, we speculate the indefinite conjecture $\mathsf{f}(\gamma_{\mathbf{Nats}}(x+n)) \xrightarrow{\mathrm{i}}^* \star$.

Every indefinite conjecture gives rise to an indefinite rewrite lemma.

▶ **Definition 24** (Indefinite Rewrite Lemmas)**.** Let $\mathcal{R}$, s, rt be as in Def. 8. Then $s \xrightarrow{\mathrm{i}}^{rt(\overline{n})} \star$ is an *indefinite rewrite lemma* for $\mathcal{R}$ iff for all $\sigma : \mathcal{V}(s) \to \mathbb{N}$ there is a term t such that $s\sigma\downarrow_{\mathcal{G}/\mathcal{A}} \xrightarrow{\mathrm{i}}_{\mathcal{R}}^{rt(\overline{n}\sigma)} t$, i.e., $s\sigma\downarrow_{\mathcal{G}/\mathcal{A}}$ starts an innermost $\mathcal{R}$-reduction of at least $rt(n_1\sigma, \ldots, n_m\sigma)$ steps.

In principle, proving indefinite conjectures $s \xrightarrow{\mathrm{i}}^* \star$ is not necessary, since $s \xrightarrow{\mathrm{i}}^0 \star$ is always a valid indefinite rewrite lemma. However, to derive useful lower complexity bounds, we need rewrite lemmas $s \xrightarrow{\mathrm{i}}^{rt(\overline{n})} \star$ for non-trivial functions rt. Thm. 25 shows that the approaches for proving lemmas from Sect. 3 and for deriving bounds from these proofs in Sect. 4 can also be used for indefinite rewrite lemmas. The only adaption needed is that the relation $\xrightarrow{\mathrm{i}}_{\mathcal{R}}$ may not reduce redexes that contain the symbol $\star$. This restriction is needed due to the innermost evaluation strategy, because $\star$ represents arbitrary terms that are not necessarily in normal form. In this way, all previously proven (definite or indefinite) rewrite lemmas $\mathcal{L}$ can be used in the proof of new (definite or indefinite) rewrite lemmas.

▶ **Theorem 25** (Bounds for Indefinite Rewrite Lemmas)**.** *Let $\xrightarrow{\mathrm{i}}_{\mathcal{R}}$ and $\xrightarrow{\mathrm{i}}_{(\mathcal{R},\mathrm{IH})}$ be restricted such that redexes may not contain the symbol $\star$ and let ih, ib, and is be defined as in Def. 11. Here, for an indefinite rewrite lemma $s \xrightarrow{\mathrm{i}}^{rt(\overline{n})} \star$ with $n \in \mathcal{V}(s)$, we say that any rewrite sequence $s[n/0] = u_1 \xrightarrow{\mathrm{i}}_{\mathcal{R}} u_2 \xrightarrow{\mathrm{i}}_{\mathcal{R}} \ldots \xrightarrow{\mathrm{i}}_{\mathcal{R}} u_{b+1}$ "proves" the induction base and any rewrite sequence $s[n/n+1] = v_1 \xrightarrow{\mathrm{i}}_{(\mathcal{R},\mathrm{IH})} v_2 \xrightarrow{\mathrm{i}}_{(\mathcal{R},\mathrm{IH})} \ldots \xrightarrow{\mathrm{i}}_{(\mathcal{R},\mathrm{IH})} v_{k+1}$ "proves" the induction step, where IH is the rule $s \to \star$. Then Thm. 12 and Thm. 14 on explicit and asymptotic runtimes hold for any definite or indefinite rewrite lemma.*

▶ **Example 26** (Complexity of Indefinite Rewrite Lemmas)**.** To continue with Ex. 23, we now infer the runtime for the rewrite lemma $\mathsf{f}(\gamma_{\mathbf{Nats}}(x+n)) \xrightarrow{\mathrm{i}}^{rt(x,n)} \star$. Since $\mathsf{f}(\gamma_{\mathbf{Nats}}(x+0))$ is already in normal form w.r.t. $\xrightarrow{\mathrm{i}}_{\mathcal{R}}$, the length of the rewrite sequence in the induction base is $ib(x) = 0$. In the induction step, we obtain $\mathsf{f}(\gamma_{\mathbf{Nats}}(x+n+1)) \xrightarrow{\mathrm{i}}_{\mathcal{R}} \mathsf{succ}(\mathsf{f}(\gamma_{\mathbf{Nats}}(x+n))) \mapsto_{\mathrm{IH}} \mathsf{succ}(\star)$. Thus, the induction hypothesis is applied $ih = 1$ time and the number of remaining rewrite steps is $is(x,n) = 1$. According to Thm. 12, we have $rt(x,n) = ih^n \cdot ib(x) + \sum_{i=0}^{n-1} ih^{n-1-i} \cdot is(x,i) = 1^n \cdot 0 + \sum_{i=0}^{n-1} 1^{n-1-i} \cdot 1 = n$. Similarly, since $ih = 1$ and both $ib(x) = 0$ and $is(x,n) = 1$ are polynomials of degree 0, Thm. 14 implies $rt_{\mathbb{N}}(n) \in \Omega(n^{\max\{0,0+1\}}) = \Omega(n)$.

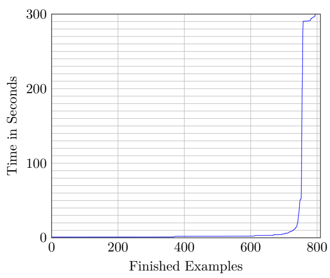
8 Experiments and Conclusion

We presented the first approach to infer *lower* bounds for the innermost runtime complexity of TRSs automatically. It is based on speculating rewrite lemmas by narrowing, proving them by induction, and determining the length of the corresponding rewrite sequences from this proof. By taking the size of the start term of the rewrite lemma into account, this yields a lower bound for $\mathrm{irc}_{\mathcal{R}}$. Our approach can be improved by argument filtering and by allowing rewrite lemmas with unknown right-hand sides. In this way the rewrite lemmas do not have to represent rewrite sequences of the original TRS precisely. Future work will be concerned with considering more general forms of induction proofs and rewrite lemmas.

We implemented our approach in AProVE [10], which uses Z3 [6] to solve arithmetic constraints. While our technique can also infer concrete bounds, currently AProVE only computes asymptotic bounds and provides the lemma that leads to the reported runtime as a witness.

There exist a few results on lower bounds for *derivational complexity* (e.g., [15, 19]) and in the *Termination Competitions*[8] 2009 - 2011, Matchbox [18] proved lower bounds for full derivational complexity where arbitrary rewrite sequences are considered.[9] However, there are no other tools that infer lower bounds for innermost runtime complexity. Hence, we compared our results with the asymptotic *upper* bounds computed by AProVE and TcT [4], the winners in the category *"Runtime Complexity – Innermost Rewriting"* at the *Termination Competition* 2014. We tested with 808 TRSs from this category of the *Termination Problem Data Base* (TPDB 9.0.2) which was used for the Termination Competition 2014. We omitted 118 non-standard TRSs with extra variables on right-hand sides or relative rules. We also disregarded 51 TRSs where AProVE or TcT proved $\mathrm{irc}_{\mathcal{R}}(n) \in \mathcal{O}(1)$ and 87 examples with $\mathrm{irc}_{\mathcal{R}}(n) \in \Omega(\omega)$ (gray cells in the table below). To identify the latter, we adapted existing innermost non-termination techniques to only consider sequences starting with basic terms. Each tool had a time limit of 300 s for each example. The following table compares the lower bound found by our implementation with the minimum upper bound computed by AProVE or TcT.

$\mathrm{irc}_{\mathcal{R}}(n)$	$\Omega(1)$	$\Omega(n)$	$\Omega(n^2)$	$\Omega(n^3)$	$\Omega(n^{>3})$	$\Omega(2^n)$	$\Omega(3^n)$	$\Omega(\omega)$
$\mathcal{O}(1)$	(51)	–	–	–	–	–	–	–
$\mathcal{O}(n)$	65	201	–	–	–	–	–	–
$\mathcal{O}(n^2)$	5	57	17	–	–	–	–	–
$\mathcal{O}(n^3)$	–	10	3	8	–	–	–	–
$\mathcal{O}(n^{>3})$	3	3	1	–	–	–	–	–
$\mathcal{O}(2^n)$	–	–	–	–	–	–	–	–
$\mathcal{O}(3^n)$	–	–	–	–	–	–	–	–
$\mathcal{O}(\omega)$	78	293	47	6	–	10	1	(87)



The average runtime of our implementation was 22.5 s, but according to the chart above, it was usually much faster. In 694 cases, the analysis finished in 5 seconds. AProVE inferred lower bounds for 657 (81%) of the 808 TRSs. Upper bounds were only obtained for 373 (46%) TRSs, although such bounds exist for at least all 670 TRSs where AProVE shows innermost termination. So although this is the first technique for lower $\mathrm{irc}_{\mathcal{R}}$-bounds, its applicability exceeds the applicability of the techniques for upper bounds which were developed for years. Tight bounds (where the lower and upper bounds are equal) were proven for the 226 TRSs on the diagonal of the table. There are just 74 TRSs where different non-trivial lower and upper bounds were inferred and for 60 of these cases, they just differ by the factor n.

Our approach is particularly powerful for TRSs that implement realistic algorithms, e.g., it shows $\mathrm{irc}_{\mathcal{R}}(n) \in \Omega(n^2)$ for many implementations of classical sorting algorithms like *quicksort*, *maxsort*, *minsort*, and *selection-sort* from the TPDB where neither AProVE nor TcT prove $\mathrm{irc}_{\mathcal{R}}(n) \in \mathcal{O}(n^2)$. Detailed experimental results and a web interface for our implementation are available at [1].

Acknowledgments We thank Fabian Emmes for important initial ideas for this paper.

8 See `http://termination-portal.org/wiki/Termination_Competition`.

9 For derivational complexity, every non-empty TRS has a trivial linear lower bound. In contrast, proving linear lower bounds for runtime complexity is not trivial. Thus, lower bounds for derivational complexity are in general unsound for runtime complexity. Therefore, an experimental comparison with tools for derivational complexity is not meaningful.

References

1 AProVE: http://aprove.informatik.rwth-aachen.de/eval/lowerbounds/.
2 T. Arts and J. Giesl. Termination of term rewriting using dependency pairs. *Theoretical Computer Science*, 236:133–178, 2000.
3 M. Avanzini and G. Moser. A combination framework for complexity. In *Proc. RTA '13*, LIPIcs 21, pages 55–70, 2013.
4 M. Avanzini and G. Moser. Tyrolean Complexity Tool: Features and usage. In *Proc. RTA '13*, LIPIcs 21, pages 71–80, 2013.
5 F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.
6 L. de Moura and N. Bjørner. Z3: An efficient SMT solver. In *Proc. TACAS '08*, LNCS 4963, pages 337–340, 2008.
7 F. Emmes, T. Enger, and J. Giesl. Proving non-looping non-termination automatically. In *Proc. IJCAR '12*, LNAI 7364, pages 225–240, 2012.
8 C. Fuhs, J. Giesl, M. Parting, P. Schneider-Kamp, S. Swiderski. Proving termination by dependency pairs and inductive theorem proving. *Journal of Automated Reasoning*, 47(2):133–160, 2011.
9 J. Giesl, M. Raffelsieper, P. Schneider-Kamp, S. Swiderski, and R. Thiemann. Automated termination proofs for Haskell by term rewriting. *ACM Transactions on Programming Languages and Systems*, 33(2), 2011.
10 J. Giesl, M. Brockschmidt, F. Emmes, F. Frohn, C. Fuhs, C. Otto, M. Plücker, P. Schneider-Kamp, T. Ströder, S. Swiderski, and R. Thiemann. Proving termination of programs automatically with AProVE. In *Proc. IJCAR '14*, LNAI 8562, pages 184–191, 2014.
11 N. Hirokawa and G. Moser. Automated complexity analysis based on the dependency pair method. In *Proc. IJCAR '08*, LNAI 5195, pages 364–379, 2008.
12 D. Hofbauer and C. Lautemann. Termination proofs and the length of derivations. In *Proc. RTA '89*, LNCS 355, pages 167–177, 1989.
13 M. Hofmann and G. Moser. Amortised resource analysis and typed polynomial interpretations. In *Proc. RTA-TLCA '14*, LNCS 8560, pages 272–286, 2014.
14 D. Knuth. Johann Faulhaber and sums of powers. *Mathematics of Computation*, 61(203):277–294, 1993.
15 G. Moser and A. Schnabl. The derivational complexity induced by the dependency pair method. *Logical Methods in Computer Science*, 7(3), 2011.
16 L. Noschinski, F. Emmes, and J. Giesl. Analyzing innermost runtime complexity of term rewriting by dependency pairs. *Journal of Automated Reasoning*, 51(1):27–56, 2013.
17 C. Otto, M. Brockschmidt, C. von Essen, and J. Giesl. Automated termination analysis of Java Bytecode by term rewriting. In *Proc. RTA '10*, LIPIcs 6, pages 259–276, 2010.
18 J. Waldmann. Matchbox: A tool for match-bounded string rewriting. In *Proc. RTA '04*, LNCS 3091, pages 85–94, 2004.
19 J. Waldmann. Automatic termination. In *Proc. RTA '09*, LNCS 5595, pages 1–16, 2009.
20 H. Zankl and M. Korp. Modular complexity analysis for term rewriting. *Logical Methods in Computer Science*, 10(1), 2014.
21 H. Zantema. Termination of term rewriting: Interpretation and type elimination. *Journal of Symbolic Computation*, 17(1):23–50, 1994.

A Proofs

To prove Thm. 9, we need some auxiliary lemmas. Lemma 27 implies that $\stackrel{\mathrm{i}}{\rightharpoonup}_{\mathcal{R}}$ is closed under instantiations of variables with natural numbers. Note that this only holds due the non-overlap condition in the definition of $\stackrel{\mathrm{i}}{\rightharpoonup}_{\mathcal{R}}$.

▶ **Lemma 27** (Stability of $\stackrel{\mathrm{i}}{\rightharpoonup}_{\mathcal{R}}$ and $\stackrel{\mathrm{i}}{\rightharpoonup}_{(\mathcal{R},\ell\to r)}$)**.** *Let $\mathcal{R}$ be a TRS, let ℓ, r, s, t be terms where s only contains variables of type $\mathbb{N}$, and let $\mu : \mathcal{V}(s) \to \mathbb{N}$ be a substitution with natural numbers. Then we have the following:*

- *$s \stackrel{\mathrm{i}}{\rightharpoonup}_{\mathcal{R}} t$ implies $s\mu \stackrel{\mathrm{i}}{\rightharpoonup}_{\mathcal{R}} t\mu$*
- *$s \stackrel{\mathrm{i}}{\rightharpoonup}_{(\mathcal{R},\ell\to r)} t$ implies $s\mu \stackrel{\mathrm{i}}{\rightharpoonup}_{(\mathcal{R},\ell\mu\to r\mu)} t\mu$*

Proof. We first consider the case where $s \stackrel{\mathrm{i}}{\rightharpoonup}_{\mathcal{R}} t$ holds. By the definition of $\stackrel{\mathrm{i}}{\rightharpoonup}_{\mathcal{R}}$, $s \stackrel{\mathrm{i}}{\rightharpoonup}_{\mathcal{R}} t$ implies that there is a term s', a substitution σ, a $\pi \in \mathrm{pos}(s')$, and a rule $\ell \to r \in \mathcal{R} \cup \mathcal{L}$ such that $s \equiv_{\mathcal{G}\cup\mathcal{A}} s'$, $s'|_\pi = \ell\sigma$, and $s'[r\sigma]_\pi \equiv_{\mathcal{G}\cup\mathcal{A}} t$. Moreover if $\ell \to r \in \mathcal{R}$, then there is no proper non-variable subterm q of $\ell\sigma$ that unifies modulo $\mathcal{G} \cup \mathcal{A}$ with a variable-renamed left-hand side of a rule from $\mathcal{R}$.

We clearly have $s\mu \equiv_{\mathcal{G}\cup\mathcal{A}} s'\mu$, $s'\mu|_\pi = \ell\sigma\mu$, and $s'\mu[r\sigma\mu]_\pi = (s'[r\sigma]_\pi)\mu \equiv_{\mathcal{G}\cup\mathcal{A}} t\mu$. If $\ell \to r \in \mathcal{L}$, then this implies $s\mu \stackrel{\mathrm{i}}{\rightharpoonup}_{\mathcal{R}} t\mu$, as desired.

Otherwise, we have $\ell \to r \in \mathcal{R}$. Assume that there exists a proper non-variable subterm q' of $\ell\sigma\mu$ that unifies modulo $\mathcal{G} \cup \mathcal{A}$ with a variable-renamed left-hand side of a rule from $\mathcal{R}$. Since the root symbol of q' must be from $\Sigma_{def}(\mathcal{R})$ and the range of μ does not include any defined symbols, we must have $q' = q\mu$ for some term q that is a proper non-variable subterm of $\ell\sigma$. But then q would already unify modulo $\mathcal{G} \cup \mathcal{A}$ with a variable-renamed left-hand side of a rule from $\mathcal{R}$, which is a contradiction to $s \stackrel{\mathrm{i}}{\rightharpoonup}_{\mathcal{R}} t$ above. Thus, in this case we can also conclude $s\mu \stackrel{\mathrm{i}}{\rightharpoonup}_{\mathcal{R}} t\mu$.

Now we consider the case where $s \stackrel{\mathrm{i}}{\rightharpoonup}_{(\mathcal{R},\ell\to r)} t$ holds. If we also have $s \stackrel{\mathrm{i}}{\rightharpoonup}_{\mathcal{R}} t$, then the claim follows from the observation above. Otherwise, we have $s \mapsto_{\ell\to r} t$. Now $s\mu \mapsto_{\ell\mu\to r\mu} t\mu$ is an immediate consequence of the definition of $\mapsto$. ◀

The next lemma shows how to infer information on innermost rewrite sequences with $\stackrel{\mathrm{i}}{\to}_{\mathcal{R}}$ from the relation $\stackrel{\mathrm{i}}{\rightharpoonup}_{\mathcal{R}}$. Here, $\twoheadrightarrow$ is like $\mapsto$, but without using the underlying equations $\mathcal{G} \cup \mathcal{A}$. So we define $s \twoheadrightarrow_{\ell\to r} t$ iff there exists a $\pi \in \mathrm{pos}(s)$ such that $s|_\pi = \ell$ and $t = s[r]_\pi$.

▶ **Lemma 28** (From $\stackrel{\mathrm{i}}{\rightharpoonup}_{\mathcal{R}}$ to $\stackrel{\mathrm{i}}{\to}_{\mathcal{R}}$)**.** *Let $\mathcal{R}$ be a TRS, let ℓ, r be terms with $\mathrm{root}(\ell) \in \Sigma_{def}(\mathcal{R})$, and let s, t be ground terms. Moreover, let $\mathcal{R}$, $\{\ell \to r\}$, and s be well typed w.r.t. Σ' and $\mathcal{V}'$, where s does not have the type $\mathbb{N}$.*

(a) *If $s \stackrel{\mathrm{i}}{\rightharpoonup}_{\mathcal{R}} t$ and this reduction is done using a rule $\ell \to r \in \mathcal{R}$, then we have $s{\downarrow_{\mathcal{G}/\mathcal{A}}} \stackrel{\mathrm{i}}{\to}_{\mathcal{R}} t{\downarrow_{\mathcal{G}/\mathcal{A}}}$.*

(b) *If $s \stackrel{\mathrm{i}}{\rightharpoonup}_{\mathcal{R}} t$ and this reduction is done using the rewrite lemma $\ell \stackrel{\mathrm{i}}{\to}^{rt(\overline{n})} r$ and the substitution σ, then we have $s{\downarrow_{\mathcal{G}/\mathcal{A}}} \stackrel{\mathrm{i}}{\to}_{\mathcal{R}}^{rt(\overline{n}\sigma)} t{\downarrow_{\mathcal{G}/\mathcal{A}}}$.*

(c) *If $s \mapsto_{\ell\to r} t$, then we have $s{\downarrow_{\mathcal{G}/\mathcal{A}}} \twoheadrightarrow_{(\ell\downarrow_{\mathcal{G}/\mathcal{A}} \to r\downarrow_{\mathcal{G}/\mathcal{A}})} t{\downarrow_{\mathcal{G}/\mathcal{A}}}$.*

Proof. In cases (a) and (b), we have $s \stackrel{\mathrm{i}}{\rightharpoonup}_{\mathcal{R}} t$. By the definition of $\stackrel{\mathrm{i}}{\rightharpoonup}_{\mathcal{R}}$, $s \stackrel{\mathrm{i}}{\rightharpoonup}_{\mathcal{R}} t$ implies that there is a term s', a substitution σ, a $\pi \in \mathrm{pos}(s')$, and a rule $\ell \to r \in \mathcal{R} \cup \mathcal{L}$ such that $s \equiv_{\mathcal{G}\cup\mathcal{A}} s'$, $s'|_\pi = \ell\sigma$, and $s'[r\sigma]_\pi \equiv_{\mathcal{G}\cup\mathcal{A}} t$. Moreover if $\ell \to r \in \mathcal{R}$, then there is no proper non-variable subterm q of $\ell\sigma$ that unifies modulo $\mathcal{G} \cup \mathcal{A}$ with a variable-renamed left-hand side of a rule from $\mathcal{R}$.

Note that $s \equiv_{\mathcal{G}\cup\mathcal{A}} s'$ implies $s{\downarrow_{\mathcal{G}/\mathcal{A}}} \equiv_{\mathcal{A}} s'{\downarrow_{\mathcal{G}/\mathcal{A}}}$ as $\to_{\mathcal{G}/\mathcal{A}}$ is terminating and confluent modulo $\mathcal{A}$. Here, $\to_{\mathcal{G}/\mathcal{A}}$ denotes the relation resulting from regarding $\mathcal{G}$ as rewrite rules oriented from left to right where rewriting is performed modulo $\mathcal{A}$ (i.e., modulo arithmetic). Since s is a ground term that does not have the type $\mathbb{N}$, $s{\downarrow_{\mathcal{G}/\mathcal{A}}}$ does not contain subterms of type $\mathbb{N}$ and therefore, $s{\downarrow_{\mathcal{G}/\mathcal{A}}} \equiv_{\mathcal{A}} s'{\downarrow_{\mathcal{G}/\mathcal{A}}}$ implies $s{\downarrow_{\mathcal{G}/\mathcal{A}}} = s'{\downarrow_{\mathcal{G}/\mathcal{A}}}$.

Since ℓ matches $s'|_\pi$, s' has a defined symbol from $\Sigma_{def}(\mathcal{R})$ at position π. Hence, there are no generator symbols and no subterms of type $\mathbb{N}$ in s' on or above the position π. Therefore, $\mathcal{G}$ and $\mathcal{A}$ cannot be applied on or above the position π. This implies $s{\downarrow_{\mathcal{G}/\mathcal{A}}} = s'{\downarrow_{\mathcal{G}/\mathcal{A}}} = (s'[s'|_\pi]_\pi){\downarrow_{\mathcal{G}/\mathcal{A}}} = s'{\downarrow_{\mathcal{G}/\mathcal{A}}}\,[(s'|_\pi){\downarrow_{\mathcal{G}/\mathcal{A}}}]_\pi = s'{\downarrow_{\mathcal{G}/\mathcal{A}}}\,[\ell\sigma{\downarrow_{\mathcal{G}/\mathcal{A}}}]_\pi$.

Let us first regard case (b) where $\ell \to r \in \mathcal{L}$. Thus, $\ell \to r$ corresponds to a rewrite lemma $\ell \xrightarrow{\text{i}}{}^{rt(\overline{n})}\, r$ and we have $\ell\sigma{\downarrow_{\mathcal{G}/\mathcal{A}}} \xrightarrow{\text{i}}{}_{\mathcal{R}}^{rt(\overline{n}\sigma)}\, r\sigma{\downarrow_{\mathcal{G}/\mathcal{A}}}$. Thus, we obtain $s{\downarrow_{\mathcal{G}/\mathcal{A}}} = s'{\downarrow_{\mathcal{G}/\mathcal{A}}}\,[\ell\sigma{\downarrow_{\mathcal{G}/\mathcal{A}}}]_\pi \xrightarrow{\text{i}}{}_{\mathcal{R}}^{rt(\overline{n}\sigma)}\, s'{\downarrow_{\mathcal{G}/\mathcal{A}}}\,[r\sigma{\downarrow_{\mathcal{G}/\mathcal{A}}}]_\pi = (s'[r\sigma]_\pi){\downarrow_{\mathcal{G}/\mathcal{A}}} \equiv_{\mathcal{A}} t{\downarrow_{\mathcal{G}/\mathcal{A}}}$. As $t{\downarrow_{\mathcal{G}/\mathcal{A}}}$ does not contain subterms of type $\mathbb{N}$, we have $(s'[r\sigma]_\pi){\downarrow_{\mathcal{G}/\mathcal{A}}} = t{\downarrow_{\mathcal{G}/\mathcal{A}}}$, which finishes the proof of (b).

Now we regard the case where $\ell \to r \in \mathcal{R}$. Since ℓ does not contain generator function symbols or subterms of type $\mathbb{N}$, we have $s{\downarrow_{\mathcal{G}/\mathcal{A}}} = s'{\downarrow_{\mathcal{G}/\mathcal{A}}}\,[\ell\sigma{\downarrow_{\mathcal{G}/\mathcal{A}}}]_\pi = s'{\downarrow_{\mathcal{G}/\mathcal{A}}}\,[\ell\sigma']_\pi$ for the substitution where $\sigma'(x) = \sigma(x){\downarrow_{\mathcal{G}/\mathcal{A}}}$ for all $x \in \mathcal{V}$. Let $t' = s'{\downarrow_{\mathcal{G}/\mathcal{A}}}\,[r\sigma']_\pi$. To prove (a), now it suffices to show that $s'{\downarrow_{\mathcal{G}/\mathcal{A}}}\,[\ell\sigma']_\pi \xrightarrow{\text{i}}_{\mathcal{R}} t'$. The reason is that then we have $s{\downarrow_{\mathcal{G}/\mathcal{A}}} = s'{\downarrow_{\mathcal{G}/\mathcal{A}}}\,[\ell\sigma']_\pi \xrightarrow{\text{i}}_{\mathcal{R}} t' = s'{\downarrow_{\mathcal{G}/\mathcal{A}}}\,[r\sigma']_\pi = s'{\downarrow_{\mathcal{G}/\mathcal{A}}}\,[r\sigma{\downarrow_{\mathcal{G}/\mathcal{A}}}]_\pi = (s'[r\sigma]_\pi){\downarrow_{\mathcal{G}/\mathcal{A}}} = t{\downarrow_{\mathcal{G}/\mathcal{A}}}$.

To prove that $s'{\downarrow_{\mathcal{G}/\mathcal{A}}}\,[\ell\sigma']_\pi \xrightarrow{\text{i}}_{\mathcal{R}} t' = s'{\downarrow_{\mathcal{G}/\mathcal{A}}}\,[r\sigma']_\pi$ holds, one has to show that this rewrite step respects the innermost evaluation strategy. Assume that there is a proper subterm q' of $\ell\sigma' = \ell\sigma{\downarrow_{\mathcal{G}/\mathcal{A}}}$ and a left-hand side of a rule from $\mathcal{R}$ that matches q'. Since the root of q' must be a defined symbol, there exists a non-variable proper subterm q of $\ell\sigma$ such that $q' = q{\downarrow_{\mathcal{G}/\mathcal{A}}}$. But then q unifies modulo $\mathcal{G} \cup \mathcal{A}$ with a variable-renamed left-hand side of a rule from $\mathcal{R}$, which is a contradiction. Thus, (a) is proved.

Finally, we have to prove (c). By definition, $s \mapsto_{\ell\to r} t$ implies that there is a term s' and a $\pi \in \text{pos}(s')$ such that $s \equiv_{\mathcal{G}\cup\mathcal{A}} s'$, $s'|_\pi = \ell$, and $s'[r]_\pi \equiv_{\mathcal{G}\cup\mathcal{A}} t$. As in case (a) and (b), since s is a ground term that does not have the type $\mathbb{N}$, we can conclude $s{\downarrow_{\mathcal{G}/\mathcal{A}}} = s'{\downarrow_{\mathcal{G}/\mathcal{A}}}$. Moreover, s' again has a defined symbol at position π, since $\text{root}(\ell) \in \Sigma_{def}(\mathcal{R})$. As in case (a) and (b), this implies $s{\downarrow_{\mathcal{G}/\mathcal{A}}} = s'{\downarrow_{\mathcal{G}/\mathcal{A}}} = s'{\downarrow_{\mathcal{G}/\mathcal{A}}}\,[\ell{\downarrow_{\mathcal{G}/\mathcal{A}}}]_\pi$. Thus, we obtain $s{\downarrow_{\mathcal{G}/\mathcal{A}}} = s'{\downarrow_{\mathcal{G}/\mathcal{A}}}\,[\ell{\downarrow_{\mathcal{G}/\mathcal{A}}}]_\pi \rightarrow_{(\ell{\downarrow_{\mathcal{G}/\mathcal{A}}} \to r{\downarrow_{\mathcal{G}/\mathcal{A}}})} s'{\downarrow_{\mathcal{G}/\mathcal{A}}}\,[r{\downarrow_{\mathcal{G}/\mathcal{A}}}]_\pi = s'[r]_\pi{\downarrow_{\mathcal{G}/\mathcal{A}}} \equiv_{\mathcal{A}} t{\downarrow_{\mathcal{G}/\mathcal{A}}}$. Since $\ell \to r$ is well typed and ℓ does not have type $\mathbb{N}$, $t{\downarrow_{\mathcal{G}/\mathcal{A}}}$ does not contain subterms of type $\mathbb{N}$. Hence, we have $s'[r]_\pi{\downarrow_{\mathcal{G}/\mathcal{A}}} = t{\downarrow_{\mathcal{G}/\mathcal{A}}}$, which proves (c). ◀

Now we prove Thm. 9 and Thm. 12 together.

▶ **Theorem 9** (Proving Rewrite Lemmas). *Let $\mathcal{R}$, s, t be as in Def. 8, $n \in \mathcal{V}(s) = \{n_1, \ldots, n_m\}$, and $\overline{n} = (n_1, \ldots, n_m)$. If $s[n/0] \xrightarrow{\text{i}}{}_{\mathcal{R}}^{*}\, t[n/0]$ and $s[n/n+1] \xrightarrow{\text{i}}{}_{(\mathcal{R},\text{IH})}^{*}\, t[n/n+1]$, where IH is the rule $s \to t$, then there is an $rt : \mathbb{N}^m \to \mathbb{N}$ such that $s \xrightarrow{\text{i}}{}^{rt(\overline{n})}\, t$ is a rewrite lemma for $\mathcal{R}$.*

▶ **Theorem 12** (Explicit Runtime of Rewrite Lemmas). *Let $s \xrightarrow{\text{i}}{}^{rt(\overline{n})}\, t$ be a rewrite lemma, where ih, ib, and is are as in Def. 11. Then we obtain $rt(\overline{n}) = ih^n \cdot ib(\widetilde{n}) + \sum_{i=0}^{n-1} ih^{n-1-i} \cdot is(\overline{n}[n/i])$.*

Proof. Let rt be defined by the recurrence equations (9). We show that $s \xrightarrow{\text{i}}{}^{rt(\overline{n})}\, t$ is a rewrite lemma. More precisely, for any $\mu : \mathcal{V}(s) \to \mathbb{N}$ which instantiates all variables of s by natural numbers we show that $s\mu{\downarrow_{\mathcal{G}/\mathcal{A}}} \xrightarrow{\text{i}}{}_{\mathcal{R}}^{rt(\overline{n}\mu)}\, t\mu{\downarrow_{\mathcal{G}/\mathcal{A}}}$ holds. To this end, we use induction on $n\mu$.

In the induction base case, we have $n\mu = 0$. We first regard the case where the reduction $s[n/0] \xrightarrow{\text{i}}{}_{\mathcal{R}}^{*}\, t[n/0]$ has length 0. Note that $s[n/0] \xrightarrow{\text{i}}{}_{\mathcal{R}}^{0}\, t[n/0]$ means $s[n/0] \equiv_{\mathcal{G}/\mathcal{A}} t[n/0]$. This implies $s\mu = s[n/0]\mu \equiv_{\mathcal{G}/\mathcal{A}} t[n/0]\mu = t\mu$. Since $\to_{\mathcal{G}/\mathcal{A}}$ is terminating and confluent modulo $\mathcal{A}$, $s\mu \equiv_{\mathcal{G}/\mathcal{A}} t\mu$ implies $s\mu{\downarrow_{\mathcal{G}/\mathcal{A}}} \equiv_{\mathcal{A}} t\mu{\downarrow_{\mathcal{G}/\mathcal{A}}}$. Since $s\mu$ and $t\mu$ are ground terms that do not have the type $\mathbb{N}$, $s\mu{\downarrow_{\mathcal{G}/\mathcal{A}}}$ and $t\mu{\downarrow_{\mathcal{G}/\mathcal{A}}}$ do not contain any subterms of type $\mathbb{N}$.

Hence, $s\mu{\downarrow_{\mathcal{G}/\mathcal{A}}} \equiv_{\mathcal{A}} t\mu{\downarrow_{\mathcal{G}/\mathcal{A}}}$ implies $s\mu{\downarrow_{\mathcal{G}/\mathcal{A}}} = t\mu{\downarrow_{\mathcal{G}/\mathcal{A}}}$, which proves the desired claim, since $\mathit{ib}(\widetilde{n}\mu) = 0$ and thus also $\mathit{rt}(\overline{n}\mu) = 0$.

Now we regard the case $s[n/0] = u_1 \stackrel{\mathrm{i}}{\rightharpoonup}_{\mathcal{R}} \ldots \stackrel{\mathrm{i}}{\rightharpoonup}_{\mathcal{R}} u_{b+1} = t[n/0]$ for $b \geq 1$. By Lemma 27, $\stackrel{\mathrm{i}}{\rightharpoonup}_{\mathcal{R}}$ is stable and thus, we obtain $s\mu = s[n/0]\mu = u_1\mu \stackrel{\mathrm{i}}{\rightharpoonup}_{\mathcal{R}} \ldots \stackrel{\mathrm{i}}{\rightharpoonup}_{\mathcal{R}} u_{b+1}\mu = t[n/0]\mu = t\mu$. When regarding rewrite rules also as rewrite lemmas as in Def. 11, Lemma 28 (a) and (b) imply $s\mu{\downarrow_{\mathcal{G}/\mathcal{A}}} = u_1\mu{\downarrow_{\mathcal{G}/\mathcal{A}}} \stackrel{\mathrm{i}}{\to}_{\mathcal{R}}^{\mathit{rt}(\overline{y}_1\sigma_1\mu)} \ldots \stackrel{\mathrm{i}}{\to}_{\mathcal{R}}^{\mathit{rt}(\overline{y}_b\sigma_b\mu)} u_{b+1}\mu{\downarrow_{\mathcal{G}/\mathcal{A}}} = t\mu{\downarrow_{\mathcal{G}/\mathcal{A}}}$. This means that $s\mu{\downarrow_{\mathcal{G}/\mathcal{A}}} \stackrel{\mathrm{i}}{\to}_{\mathcal{R}}^{\mathit{ib}(\widetilde{n}\mu)} t\mu{\downarrow_{\mathcal{G}/\mathcal{A}}}$ or in other words, $s\mu{\downarrow_{\mathcal{G}/\mathcal{A}}} \stackrel{\mathrm{i}}{\to}_{\mathcal{R}}^{\mathit{rt}(\overline{n}\mu)} t\mu{\downarrow_{\mathcal{G}/\mathcal{A}}}$.

In the induction step case, we have $n\mu > 0$. Let $\mu' : \mathcal{V}(s) \to \mathbb{N}$ where μ' is like μ for all $\mathcal{V}(s) \setminus \{n\}$ and $n\mu' = n\mu - 1$. In the case where the reduction $s[n/n+1] \stackrel{\mathrm{i}}{\rightharpoonup}_{\mathcal{R}}^{*} t[n/n+1]$ has length 0, we have $s\mu \equiv_{\mathcal{A}} s[n/n+1]\mu' \equiv_{\mathcal{G}/\mathcal{A}} t[n/n+1]\mu' \equiv_{\mathcal{A}} t\mu$. Thus, we again have $s\mu{\downarrow_{\mathcal{G}/\mathcal{A}}} \equiv_{\mathcal{A}} t\mu{\downarrow_{\mathcal{G}/\mathcal{A}}}$ which implies $s\mu{\downarrow_{\mathcal{G}/\mathcal{A}}} = t\mu{\downarrow_{\mathcal{G}/\mathcal{A}}}$. This again proves the desired claim, since $\mathit{ih} = 0$ and $\mathit{is}(\overline{n}\mu') = 0$ and thus also $\mathit{rt}(\overline{n}\mu) = \mathit{rt}(\overline{n}[n/n+1]\mu') = 0$.

Now we regard the case $s[n/n+1] = v_1 \stackrel{\mathrm{i}}{\rightharpoonup}_{(\mathcal{R},\mathrm{IH})} \ldots \stackrel{\mathrm{i}}{\rightharpoonup}_{(\mathcal{R},\mathrm{IH})} v_{k+1} = t[n/n+1]$ for $k \geq 1$. By Lemma 27, $\stackrel{\mathrm{i}}{\rightharpoonup}_{\mathcal{R}}$ is stable and thus, we obtain $s[n/n+1]\mu' = v_1\mu' \stackrel{\mathrm{i}}{\rightharpoonup}_{(\mathcal{R},\mathrm{IH}\mu')} \ldots \stackrel{\mathrm{i}}{\rightharpoonup}_{(\mathcal{R},\mathrm{IH}\mu')} v_{k+1}\mu' = t[n/n+1]\mu'$.

If $v_j\mu' \stackrel{\mathrm{i}}{\rightharpoonup}_{\mathcal{R}} v_{j+1}\mu'$, then Lemma 28 (a) and (b) imply $v_j\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}} \stackrel{\mathrm{i}}{\to}_{\mathcal{R}}^{\mathit{rt}'_j(\overline{z}_j\theta_j\mu')} v_{j+1}\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}$. Otherwise, if $v_j\mu' \mapsto_{\mathrm{IH}\mu'} v_{j+1}\mu'$, then by Lemma 28 (c) we have $v_j\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}} \twoheadrightarrow_{(s\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}} \to t\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}})} v_{j+1}\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}$. Note that the induction hypothesis implies $s\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}} \stackrel{\mathrm{i}}{\to}_{\mathcal{R}}^{\mathit{rt}(\overline{n}\mu')} t\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}$. This means that $v_j\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}} \twoheadrightarrow_{(s\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}} \to t\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}})} v_{j+1}\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}$ implies $v_j\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}} \stackrel{\mathrm{i}}{\to}_{\mathcal{R}}^{\mathit{rt}(\overline{n}\mu')} v_{j+1}\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}$. Since there are ih many of these steps, we finally get $s[n/n+1]\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}} \stackrel{\mathrm{i}}{\to}_{\mathcal{R}}^{\mathit{ih}\cdot\mathit{rt}(\overline{n}\mu')+\mathit{is}(\overline{n}\mu')} t[n/n+1]\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}$ or in other words, $s[n/n+1]\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}} \stackrel{\mathrm{i}}{\to}_{\mathcal{R}}^{\mathit{rt}(\overline{n}[n/n+1]\mu')} t[n/n+1]\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}$. This proves the desired claim, since $s\mu{\downarrow_{\mathcal{G}/\mathcal{A}}} = s[n/n+1]\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}$, $t\mu{\downarrow_{\mathcal{G}/\mathcal{A}}} = t[n/n+1]\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}$, and $\mathit{rt}(\overline{n}\mu) = \mathit{rt}(\overline{n}[n/n+1]\mu')$.

Finally we show by induction on n that the closed form for $\mathit{rt}(\overline{n})$ given in Thm. 12 indeed satisfies the recurrence equations (9). We obtain $\mathit{rt}(\overline{n}[n/0]) = \mathit{ih}^0 \cdot \mathit{ib}(\widetilde{n}) = \mathit{ib}(\widetilde{n})$, as required in (9). Similarly, we have $\mathit{rt}(\overline{n}[n/n+1]) = \mathit{ih}^{n+1} \cdot \mathit{ib}(\widetilde{n}) \;+\; \sum_{i=0}^{n} \mathit{ih}^{n-i} \cdot \mathit{is}(\overline{n}[n/i]) = \mathit{ih} \cdot (\, \mathit{ih}^n \cdot \mathit{ib}(\widetilde{n}) \;+\; \sum_{i=0}^{n-1} \mathit{ih}^{n-1-i} \cdot \mathit{is}(\overline{n}[n/i])\,) + \mathit{is}(\overline{n}) = \mathit{ih} \cdot \mathit{rt}(\overline{n}) + \mathit{is}(\overline{n})$, as in (9). ◀

▶ **Theorem 14** (Asymptotic Runtime of Rewrite Lemmas). *Let $s \stackrel{\mathrm{i}}{\to}^{\mathit{rt}(\overline{n})} t$ be a rewrite lemma with ih, ib, and is as in Def. 11. Moreover, let ib and is be polynomials of degree d_{ib} and d_{is}.*

- *If $\mathit{ih} = 0$, then $\mathit{rt}_{\mathbb{N}}(n) \in \Omega(n^{\max\{d_{\mathit{ib}}, d_{\mathit{is}}\}})$.*
- *If $\mathit{ih} = 1$, then $\mathit{rt}_{\mathbb{N}}(n) \in \Omega(n^{\max\{d_{\mathit{ib}}, d_{\mathit{is}}+1\}})$.*
- *If $\mathit{ih} > 1$, then $\mathit{rt}_{\mathbb{N}}(n) \in \Omega(\mathit{ih}^n)$.*

Proof. If $\mathit{ih} = 0$, then the theorem immediately follows from the fact that $\mathit{rt}(\overline{n}[n/0]) = \mathit{ib}(\widetilde{n})$ and $\mathit{rt}(\overline{n}[n/n+1]) = \mathit{is}(\overline{n})$, otherwise.

If $\mathit{ih} = 1$, then Thm. 12 implies $\mathit{rt}(\overline{n}) = \mathit{ib}(\widetilde{n}) \;+\; \sum_{i=0}^{n-1} \mathit{is}(\overline{n}[n/i])$. Since is is a polynomial of degree d_{is}, we have $\mathit{is}(\overline{n}) = t_0 + t_1 n + t_2 n^2 + \ldots + t_{d_{\mathit{is}}} n^{d_{\mathit{is}}}$. Here, $t_1, \ldots, t_{d_{\mathit{is}}}$ are polynomials containing variables from $\widetilde{n}$. Since $\mathrm{degree}(\mathit{is}) = \max\{\mathrm{degree}(t_k n^k) \mid 0 \leq k \leq d_{\mathit{is}}\}$ and $\mathrm{degree}(t_k n^k) = \mathrm{degree}(t_k) + k$, there must be a $j \in \{0, \ldots, d_{\mathit{is}}\}$ such that $\mathrm{degree}(t_j) = d_{\mathit{is}} - j$ and for all $k \in \{0, \ldots, d_{\mathit{is}}\}$ with $k \neq j$ we have $\mathrm{degree}(t_k) \leq d_{\mathit{is}} - k$. Hence,

$$
\begin{aligned}
\mathit{rt}(\overline{n}) \;&=\; \mathit{ib}(\widetilde{n}) \;+\; \sum_{i=0}^{n-1} (t_0 + t_1 i + t_2 i^2 + \ldots + t_{d_{\mathit{is}}} i^{d_{\mathit{is}}}) \\
&=\; \mathit{ib}(\widetilde{n}) \;+\; t_0 \cdot \sum_{i=0}^{n-1} i^0 + t_1 \cdot \sum_{i=0}^{n-1} i^1 + t_2 \cdot \sum_{i=0}^{n-1} i^2 + \ldots + t_{d_{\mathit{is}}} \cdot \sum_{i=0}^{n-1} i^{d_{\mathit{is}}}. \qquad (10)
\end{aligned}
$$

To prove $rt_{\mathbb{N}}(n) \in \Omega(n^{\max\{d_{ib}, d_{is}+1\}})$, we now show that rt is a polynomial of degree $\max\{d_{ib}, d_{is} + 1\}$. To this end, note that by Faulhaber's formula [14], for any $e \in \mathbb{N}$, $\sum_{i=0}^{n-1} i^e$ is a polynomial over the variable n of degree $e + 1$. More precisely, for $e = 0$ we have $\sum_{i=0}^{n-1} i^e = n$ and for $e \geq 1$ we have $\sum_{i=0}^{n-1} i^e = \frac{1}{e+1} \left(\sum_{j=0}^{e} \binom{e+1}{j} \cdot B_j \cdot n^{e+1-j}\right)$. Here, B_j is the j-th Bernoulli number, where we use the *first Bernoulli numbers* which are defined as $B_0 = 1$ and $B_m = -\frac{1}{m+1} \cdot \sum_{k=0}^{m-1} \binom{m+1}{k} B_k$ for $m > 0$. So $B_1 = -\frac{1}{2}$. Hence, we obtain

$$\begin{array}{rcll} \text{degree}(rt) & = & \max\{d_{ib}, \text{degree}(t_k \cdot \sum_{i=0}^{n-1} i^k) \mid k \in \{0, \dots, d_{is}\}\} & \text{by (10)} \\ & = & \max\{d_{ib}, \text{degree}(t_k) + k + 1 \mid k \in \{0, \dots, d_{is}\}\} & \text{by Faulhaber's formula} \\ & = & \max\{d_{ib}, d_{is} + 1\} & \text{as degree}(t_j) = d_{is} - j \text{ and} \\ & & & \text{degree}(t_k) \leq d_{is} - k \text{ for} \\ & & & \text{all } k \neq j \end{array}$$

Finally we regard the case where $ih > 1$. Now Thm. 12 implies

$$\begin{array}{rcll} rt(\overline{n}) & = & ih^n \cdot ib(\widetilde{n}) + \sum_{i=0}^{n-1} ih^{n-1-i} \cdot is(\overline{n}[n/i]) & \\ & \geq & \sum_{i=0}^{n-1} ih^{n-1-i} & \text{as } is(\overline{n}) \geq 1 \text{ for all } \overline{n} \\ & = & \sum_{j=0}^{n-1} ih^j & \\ & = & \frac{ih^n - 1}{ih - 1}. & \end{array}$$

Hence, we obtain $rt_{\mathbb{N}}(n) \in \Omega(ih^n)$. ◀

▶ **Theorem 17** (Explicit Lower Bounds for $\text{irc}_{\mathcal{R}}$)**.** *Let $s \xrightarrow{\mathrm{i}}{}^{rt(n_1,\dots,n_m)} t$ be a rewrite lemma for $\mathcal{R}$, let $sz : \mathbb{N}^m \to \mathbb{N}$ be a function such that $sz(b_1, \dots, b_m)$ is the size of $s[n_1/b_1, \dots, n_m/b_m]{\downarrow_{\mathcal{G}/\mathcal{A}}}$ for all $b_1, \dots, b_m \in \mathbb{N}$, and let $sz_{\mathbb{N}}$'s inverse function $sz_{\mathbb{N}}^{-1}$ exist. Then $rt_{\mathbb{N}} \circ \lfloor sz_{\mathbb{N}}^{-1} \rfloor$ is a lower bound for $\text{irc}_{\mathcal{R}}$, i.e., $(rt_{\mathbb{N}} \circ \lfloor sz_{\mathbb{N}}^{-1} \rfloor)(n) \leq \text{irc}_{\mathcal{R}}(n)$ holds for all $n \in \mathbb{N}$ with $n \geq \min(\text{img}(sz_{\mathbb{N}}))$.*

Proof. If $n \geq \min(\text{img}(sz_{\mathbb{N}}))$, then there is a maximal $n' \leq n$ such that $n' \in \text{img}(sz_{\mathbb{N}})$. Thus, $\lfloor sz_{\mathbb{N}}^{-1} \rfloor(n) = sz_{\mathbb{N}}^{-1}(n')$. Note that due to the rewrite lemma $s \xrightarrow{\mathrm{i}}{}^{rt(n_1,\dots,n_m)} t$, the term $s[n_1/sz_{\mathbb{N}}^{-1}(n'), \dots, n_m/sz_{\mathbb{N}}^{-1}(n')]{\downarrow_{\mathcal{G}/\mathcal{A}}}$ has an innermost evaluation of length $rt(sz_{\mathbb{N}}^{-1}(n'), \dots, sz_{\mathbb{N}}^{-1}(n')) = rt(\lfloor sz_{\mathbb{N}}^{-1} \rfloor(n), \dots, \lfloor sz_{\mathbb{N}}^{-1} \rfloor(n)) = (rt_{\mathbb{N}} \circ \lfloor sz_{\mathbb{N}}^{-1} \rfloor)(n)$. The size of the start term $s[n_1/sz_{\mathbb{N}}^{-1}(n'), \dots, n_m/sz_{\mathbb{N}}^{-1}(n')] {\downarrow_{\mathcal{G}/\mathcal{A}}}$ is $sz(sz_{\mathbb{N}}^{-1}(n'), \dots, sz_{\mathbb{N}}^{-1}(n')) = sz_{\mathbb{N}}(sz_{\mathbb{N}}^{-1}(n')) = n'$. Since $\text{root}(s) \in \Sigma_{def}(\mathcal{R})$ and s has no defined symbol below the root, $s[n_1/sz_{\mathbb{N}}^{-1}(n'), \dots, n_m/sz_{\mathbb{N}}^{-1}(n')]{\downarrow_{\mathcal{G}/\mathcal{A}}}$ is a basic term. As this basic term has the size $n' \leq n$ and its evaluation has the length $(rt_{\mathbb{N}} \circ \lfloor sz_{\mathbb{N}}^{-1} \rfloor)(n)$, this implies $(rt_{\mathbb{N}} \circ \lfloor sz_{\mathbb{N}}^{-1} \rfloor)(n) \leq \text{irc}_{\mathcal{R}}(n)$. ◀

To prove Lemma 18, we need the following auxiliary lemma. Here, similar to $\lfloor h \rfloor$, for any (total) function $h : M \to \mathbb{N}$ where M is an infinite subset of $\mathbb{N}$, let $\lceil h \rceil(n) : \mathbb{N} \to \mathbb{N}$, be defined by $\lceil h \rceil(n) = h(\min\{n' \mid n' \in M, n' \geq n\})$. We only define $\lfloor h \rfloor$ if h's domain M is infinite, because this is needed to ensure that there is always an $n' \in M$ with $n' \geq n$.

▶ **Lemma 29** (Connection Between $\lfloor h \rfloor$ and $\lceil h \rceil$)**.** *Let M be an infinite subset of $\mathbb{N}$ and let $h : M \to \mathbb{N}$ be strictly monotonically increasing and surjective. Then we have $\lfloor h \rfloor(n) \in \{\lceil h \rceil(n), \lceil h \rceil(n) - 1\}$ for all $n \in \mathbb{N}$.*

Proof. Let $n \in \mathbb{N}$. If $n \in M$, then $\lfloor h \rfloor(n) = \lceil h \rceil(n)$.

If $n \notin M$ and $n < \min(M)$, then $\lfloor h \rfloor(n) = 0$. Moreover, since h is strictly monotonically increasing and surjective, we also have $\lceil h \rceil(n) = 0$.

If $n \notin M$ and $n > \min(M)$, let $n' = \max\{n' \mid n' \in M, n' < n\}$ and let $n'' = \min\{n'' \mid n'' \in M, n'' > n\}$. Thus, $n' < n < n''$. By strict monotonicity of h, we have $h(n') < h(n'')$.

Assume that $h(n'') - h(n') > 1$. Then, by surjectivity of h, there is an $\tilde{n} \in M$ such that $h(\tilde{n}) = h(n') + 1$ and thus $h(n') < h(\tilde{n}) < h(n'')$. By strict monotonicity of h, we obtain $n' < \tilde{n} < n''$. Thus, we either have $\tilde{n} < n$ which contradicts the definition of $n' = \max\{n' \mid n' \in M, n' < n\}$ or $\tilde{n} > n$ which contradicts $n'' = \min\{n'' \mid n'' \in M, n'' > n\}$. Hence, $\lfloor h \rfloor(n) = h(n') = h(n'') - 1 = \lceil h \rceil(n) - 1$. ◀

▶ **Lemma 18** (Asymptotic Bounds for Function Composition). *Let $rt_{\mathbb{N}}, sz_{\mathbb{N}} : \mathbb{N} \to \mathbb{N}$ where $sz_{\mathbb{N}} \in \mathcal{O}(n^e)$ for some $e \geq 1$ and where $sz_{\mathbb{N}}$ is strictly monotonically increasing.*

- *If $rt_{\mathbb{N}}(n) \in \Omega(n^d)$ with $d \geq 0$, then $(rt_{\mathbb{N}} \circ \lfloor sz_{\mathbb{N}}^{-1} \rfloor)(n) \in \Omega(n^{\frac{d}{e}})$.*
- *If $rt_{\mathbb{N}}(n) \in \Omega(b^n)$ with $b \geq 1$, then $(rt_{\mathbb{N}} \circ \lfloor sz_{\mathbb{N}}^{-1} \rfloor)(n) \in \Omega(b^{\sqrt[e]{n}})$.*

Proof. We first consider the case where $rt_{\mathbb{N}}(n) \in \Omega(n^d)$. By definition of $\mathcal{O}$, $sz_{\mathbb{N}}(n) \in \mathcal{O}(n^e)$ implies

$$\exists n_0, c > 0.\ \forall n \in \mathbb{N}, n > n_0. \quad c \cdot n^e \geq sz_{\mathbb{N}}(n).$$

By instantiating n with $sz_{\mathbb{N}}^{-1}(n)$, we obtain

$$\exists n_0, c > 0.\ \forall n \in \operatorname{img}(sz_{\mathbb{N}}), sz_{\mathbb{N}}^{-1}(n) > n_0. \quad c \cdot (sz_{\mathbb{N}}^{-1}(n))^e \geq sz_{\mathbb{N}}(sz_{\mathbb{N}}^{-1}(n)).$$

Since $sz_{\mathbb{N}}$ is strictly monotonically increasing, $sz_{\mathbb{N}}^{-1}$ is also strictly monotonically increasing. Thus, there is an n_1 such that for all $n > n_1$ with $n \in \operatorname{img}(sz_{\mathbb{N}})$ we have $sz_{\mathbb{N}}^{-1}(n) > n_0$. Hence, we get

$$\exists n_1, c > 0.\ \forall n \in \operatorname{img}(sz_{\mathbb{N}}), n > n_1. \quad c \cdot (sz_{\mathbb{N}}^{-1}(n))^e \geq sz_{\mathbb{N}}(sz_{\mathbb{N}}^{-1}(n)).$$

This simplifies to

$$\exists n_1, c > 0.\ \forall n \in \operatorname{img}(sz_{\mathbb{N}}), n > n_1. \quad c \cdot (sz_{\mathbb{N}}^{-1}(n))^e \geq n.$$

When dividing by c and building the e-th root on both sides, we get

$$\exists n_1, c > 0.\ \forall n \in \operatorname{img}(sz_{\mathbb{N}}), n > n_1. \quad sz_{\mathbb{N}}^{-1}(n) \geq \sqrt[e]{\frac{n}{c}}.$$

By the monotonicity of $\sqrt[e]{\frac{n}{c}}$, this implies

$$\exists n_1, c > 0.\ \forall n \in \mathbb{N}, n > n_1. \quad \lceil sz_{\mathbb{N}}^{-1} \rceil(n) \geq \sqrt[e]{\frac{n}{c}}.$$

Note that $sz_{\mathbb{N}}$ is total and hence, $sz_{\mathbb{N}}^{-1}$ is surjective. Moreover, $\operatorname{img}(sz_{\mathbb{N}})$ is infinite as $sz_{\mathbb{N}}$ is strictly monotonically increasing. Hence, with Lemma 29 we get $\lfloor sz_{\mathbb{N}}^{-1} \rfloor(n) + 1 \geq \lceil sz_{\mathbb{N}}^{-1} \rceil(n)$ for all $n \in \mathbb{N}$ and thus

$$\exists n_1, c > 0.\ \forall n \in \mathbb{N}, n > n_1. \quad \lfloor sz_{\mathbb{N}}^{-1} \rfloor(n) + 1 \geq \sqrt[e]{\frac{n}{c}}.$$

Hence

$$\exists n_1, c > 0.\ \forall n \in \mathbb{N}, n > n_1. \quad \lfloor sz_{\mathbb{N}}^{-1} \rfloor(n) \geq \sqrt[e]{\frac{n}{c}} - 1. \tag{11}$$

By definition of Ω, $rt_{\mathbb{N}}(n) \in \Omega(n^d)$ implies

$$\exists n_0, c' > 0.\ \forall n \in \mathbb{N}, n > n_0. \quad c' \cdot n^d \leq rt_{\mathbb{N}}(n).$$

By instantiating n with $sz_{\mathbb{N}}^{-1}(n)$ again, we obtain

$$\exists n_0, c' > 0.\ \forall n \in \mathbb{N}, \lfloor sz_{\mathbb{N}}^{-1} \rfloor(n) > n_0. \quad c' \cdot (\lfloor sz_{\mathbb{N}}^{-1} \rfloor(n))^d \le rt_{\mathbb{N}}(\lfloor sz_{\mathbb{N}}^{-1} \rfloor(n)).$$

Since $sz_{\mathbb{N}}^{-1}$ is strictly monotonically increasing, $\lfloor sz_{\mathbb{N}}^{-1} \rfloor$ is weakly monotonically increasing by construction. As $\lfloor sz_{\mathbb{N}}^{-1} \rfloor$ is surjective, there is an n_2 such that for all $n > n_2$ we have $\lfloor sz_{\mathbb{N}}^{-1} \rfloor(n) > n_0$. Thus, we obtain

$$\exists n_2, c' > 0.\ \forall n \in \mathbb{N}, n > n_2. \quad c' \cdot (\lfloor sz_{\mathbb{N}}^{-1} \rfloor(n))^d \le rt_{\mathbb{N}}(\lfloor sz_{\mathbb{N}}^{-1} \rfloor(n)).$$

With (11) and weak monotonicity of $c' \cdot n^d$, we get the following by choosing $n_3 = \max\{n_1, n_2\}$.

$$\exists n_3, c, c' > 0.\ \forall n \in \mathbb{N}, n > n_3. \quad c' \cdot (\sqrt[e]{\frac{n}{c}} - 1)^d \le rt_{\mathbb{N}}(\lfloor sz_{\mathbb{N}}^{-1} \rfloor(n))$$

Thus, we have

$$\exists c > 0. \quad (rt_{\mathbb{N}} \circ \lfloor sz_{\mathbb{N}}^{-1} \rfloor)(n) \in \Omega\left(\left(\sqrt[e]{\frac{n}{c}} - 1\right)^d\right)$$

and, as a result, $(rt_{\mathbb{N}} \circ \lfloor sz_{\mathbb{N}}^{-1} \rfloor)(n) \in \Omega\left(n^{\frac{d}{e}}\right)$.

Now we consider the case where $rt_{\mathbb{N}}(n) \in \Omega(b^n)$. As in the previous case, (11) holds. By definition of Ω, $rt_{\mathbb{N}}(n) \in \Omega(b^n)$ implies

$$\exists n_0, c' > 0.\ \forall n \in \mathbb{N}, n > n_0. \quad c' \cdot b^n \le rt_{\mathbb{N}}(n).$$

By instantiating n with $sz_{\mathbb{N}}^{-1}(n)$, we obtain

$$\exists n_0, c' > 0.\ \forall n \in \mathbb{N}, \lfloor sz_{\mathbb{N}}^{-1} \rfloor(n) > n_0. \quad c' \cdot b^{\lfloor sz_{\mathbb{N}}^{-1} \rfloor(n)} \le rt_{\mathbb{N}}(\lfloor sz_{\mathbb{N}}^{-1} \rfloor(n)).$$

Since $sz_{\mathbb{N}}^{-1}$ is strictly monotonically increasing, $\lfloor sz_{\mathbb{N}}^{-1} \rfloor$ is weakly monotonically increasing by construction. As $\lfloor sz_{\mathbb{N}}^{-1} \rfloor$ is surjective, there is an n_2 such that for all $n > n_2$ we have $\lfloor sz_{\mathbb{N}}^{-1} \rfloor(n) > n_0$. Thus, we obtain

$$\exists n_2, c' > 0.\ \forall n \in \mathbb{N}, n > n_2. \quad c' \cdot b^{\lfloor sz_{\mathbb{N}}^{-1} \rfloor(n)} \le rt_{\mathbb{N}}(\lfloor sz_{\mathbb{N}}^{-1} \rfloor(n)).$$

With (11) and weak monotonicity of b^n, we get the following by choosing $n_3 = \max\{n_1, n_2\}$.

$$\exists n_3, c, c' > 0.\ \forall n \in \mathbb{N}, n > n_3. \quad c' \cdot b^{\sqrt[e]{\frac{n}{c}} - 1} \le rt_{\mathbb{N}}(\lfloor sz_{\mathbb{N}}^{-1} \rfloor(n)).$$

Thus, we have

$$\exists c > 0. \quad (rt_{\mathbb{N}} \circ \lfloor sz_{\mathbb{N}}^{-1} \rfloor)(n) \in \Omega\left(b^{\sqrt[e]{\frac{n}{c}} - 1}\right)$$

and, as a result, $(rt_{\mathbb{N}} \circ \lfloor sz_{\mathbb{N}}^{-1} \rfloor)(n) \in \Omega\left(b^{\sqrt[e]{n}}\right)$. ◀

▶ **Theorem 19** (Asymptotic Lower Bounds for $\mathrm{irc}_{\mathcal{R}}$). *Let $s \xrightarrow{\mathrm{i}}{}^{rt(\overline{n})} t$ be a rewrite lemma for $\mathcal{R}$ and let $sz : \mathbb{N}^m \to \mathbb{N}$ be a function such that $sz(b_1, \ldots, b_m)$ is the size of $s[n_1/b_1, \ldots, n_m/b_m]\downarrow_{\mathcal{G}/\mathcal{A}}$ for all $b_1, \ldots, b_m \in \mathbb{N}$, where $sz_{\mathbb{N}}(n) \in \mathcal{O}(n^e)$ for some $e \ge 1$ and $sz_{\mathbb{N}}$ is strictly monotonically increasing. Furthermore, let ih, ib, and is be defined as in Def. 11.*

1. *If $\mathit{ih} = 0$ and ib and is are polynomials of degree d_{ib} and d_{is}, then $\mathrm{irc}_{\mathcal{R}}(n) \in \Omega(n^{\frac{\max\{d_{\mathit{ib}}, d_{\mathit{is}}\}}{e}})$.*

2. *If* $\mathit{ih} = 1$ *and* ib *and* is *are polynomials of degree* d_{ib} *and* d_{is}*, then* $\mathrm{irc}_{\mathcal{R}}(n) \in \Omega(n^{\frac{\max\{d_{\mathit{ib}}, d_{\mathit{is}}+1\}}{e}})$.
3. *If* $\mathit{ih} > 1$*, then* $\mathrm{irc}_{\mathcal{R}}(n) \in \Omega(\mathit{ih}^{\sqrt[e]{n}})$.

Proof.
1. In this case, Thm. 14 implies $\mathit{rt}_{\mathbb{N}}(n) \in \Omega(n^{\max\{d_{\mathit{ib}}, d_{\mathit{is}}\}})$. With Lemma 18, we get $(\mathit{rt}_{\mathbb{N}} \circ \lfloor \mathit{sz}_{\mathbb{N}}^{-1} \rfloor)(n) \in \Omega(n^{\frac{\max\{d_{\mathit{ib}}, d_{\mathit{is}}\}}{e}})$. Moreover, Thm. 17 states that $(\mathit{rt}_{\mathbb{N}} \circ \lfloor \mathit{sz}_{\mathbb{N}}^{-1} \rfloor)(n) \leq \mathrm{irc}_{\mathcal{R}}(n)$ holds for all $n \in \mathbb{N}$. Thus, we obtain $\mathrm{irc}_{\mathcal{R}}(n) \in \Omega(n^{\frac{\max\{d_{\mathit{ib}}, d_{\mathit{is}}\}}{e}})$.
2. Now Thm. 14 implies $\mathit{rt}_{\mathbb{N}}(n) \in \Omega(n^{\max\{d_{\mathit{ib}}, d_{\mathit{is}}+1\}})$. Thus, with Lemma 18, we result in $(\mathit{rt}_{\mathbb{N}} \circ \lfloor \mathit{sz}_{\mathbb{N}}^{-1} \rfloor)(n) \in \Omega(n^{\frac{\max\{d_{\mathit{ib}}, d_{\mathit{is}}+1\}}{e}})$. Similar to the previous case, this implies $\mathrm{irc}_{\mathcal{R}}(n) \in \Omega(n^{\frac{\max\{d_{\mathit{ib}}, d_{\mathit{is}}+1\}}{e}})$.
3. In this case, Thm. 14 states $\mathit{rt}_{\mathbb{N}}(n) \in \Omega(\mathit{ih}^n)$. With Lemma 18, we get $(\mathit{rt}_{\mathbb{N}} \circ \lfloor \mathit{sz}_{\mathbb{N}}^{-1} \rfloor)(n) \in \Omega(\mathit{ih}^{\sqrt[e]{n}})$. Similar to the previous cases, we obtain $\mathrm{irc}_{\mathcal{R}}(n) \in \Omega(\mathit{ih}^{\sqrt[e]{n}})$.

◀

To prove Thm. 22, we need an auxiliary definition and some lemmas. In the following, let $\mathcal{R}$ be a TRS over a signature Σ, let $f \in \Sigma$ with $\mathrm{ar}_{\Sigma}(f) = k$, and let $i \in \{1, \ldots, k\}$. We now introduce an operation which lifts a position of a filtered term to the corresponding term where the filtered argument is still present. Similarly, we also introduce a converse operation which transforms positions of the original term into positions of the filtered term.

► **Definition 30** (π_t^+ **and** π_s^-)**.** Given a term $t \in \mathcal{T}(\Sigma_{\setminus(f,i)}, \mathcal{V})$, we define π_t^+ as follows for all positions $\pi \in \mathrm{pos}(t)$:
- If $\pi = \varepsilon$, then $\pi_t^+ = \varepsilon$.
- If $\pi = j.\pi'$ for a $j \in \mathbb{N}$, then
  - if $\mathrm{root}(t) = f'$ and $j \geq i$, then $\pi_t^+ = (j+1) \,.\, (\pi')_{t|_j}^+$,
  - otherwise $\pi_t^+ = j \,.\, (\pi')_{t|_j}^+$

Similarly, given a term $s \in \mathcal{T}(\Sigma, \mathcal{V})$, we define π_s^- as follows for all positions $\pi \in \mathrm{pos}(s)$ that are not below the i-th argument of an occurrence of f in s:
- If $\pi = \varepsilon$, then $\pi_s^- = \varepsilon$.
- If $\pi = j.\pi'$ for a $j \in \mathbb{N}$, then
  - if $\mathrm{root}(s) = f$ and $j > i$, then $\pi_s^- = (j-1) \,.\, (\pi')_{s|_j}^-$,
  - otherwise $\pi_s^- = j \,.\, (\pi')_{s|_j}^-$

As an example, consider the terms $s = \mathsf{intlist}(\mathsf{cons}(x, \mathsf{nil}))$ and $t = \mathsf{intlist}(\mathsf{cons}(\mathsf{nil}))$. We have $s_{\setminus(\mathsf{cons},1)} = t$, i.e., here the symbol f is cons and the position that is filtered away is $i = 1$. The purpose of the above definition is to transform a position of t to the corresponding position in the term s and vice versa. So we have $(1.1)_t^+ = 1.(1)_{\mathsf{cons}(\mathsf{nil})}^+ = 1.2.(\varepsilon)_{\mathsf{nil}}^+ = 1.2$. Similarly, we have $(1.2)_s^- = 1.1$.

The following lemma shows that any position π in a filtered term t corresponds to the position π_t^+ in the original non-filtered term.

► **Lemma 31** (Lifting of Positions From Filtered Terms)**.** *Let* $t \in \mathcal{T}(\Sigma_{\setminus(f,i)}, \mathcal{V})$ *and* $s \in \mathcal{T}(\Sigma, \mathcal{V})$ *with* $t = s_{\setminus(f,i)}$*. Then for all* $\pi \in \mathrm{pos}(t)$*, we have* $t|_\pi = (s|_{\pi_t^+})_{\setminus(f,i)}$ *and* $(\pi_t^+)_s^- = \pi$*.*

Proof. We use induction on π. If $\pi = \varepsilon$, then the lemma trivially holds, since $t|_\varepsilon = t = s_{\setminus(f,i)} = (s|_{\varepsilon_t^+})_{\setminus(f,i)}$ and $(\varepsilon_t^+)_s^- = \varepsilon$.

Now let $\pi = j.\pi'$ and $\mathrm{root}(t) = f$. Then we have $s = f(s_1, \ldots, s_k)$ for some terms $s_1, \ldots, s_k$ and $t = f(t_1, \ldots, t_{k-1})$, where $t_m = (s_m)_{\setminus(f,i)}$ for $1 \leq m < i$ and $t_m = (s_{m+1})_{\setminus(f,i)}$ for $i \leq m \leq k-1$. We obtain

$$t|_\pi = f(t_1, \ldots, t_{k-1})|_{j.\pi'} = t_j|_{\pi'}.$$

If $j \geq i$, then

$$(s|_{\pi_t^+})_{\setminus(f,i)} = (f(s_1,\ldots,s_k)|_{(j+1)\,.\,(\pi')^+_{t_j}})_{\setminus(f,i)} = (s_{j+1}|_{(\pi')^+_{t_j}})_{\setminus(f,i)}$$

and the lemma follows from the induction hypothesis which states $t_j|_{\pi'} = (s_{j+1}|_{(\pi')^+_{t_j}})_{\setminus(f,i)}$. In a similar way, we obtain $(\pi_t^+)^-_s = ((j.\pi')^+_t)^-_s = ((j+1)\,.\,(\pi')^+_t)^-_s = j\,.\,((\pi')^+_t)^-_s = j.\pi' = \pi$ by the induction hypothesis.

If $j < i$, then

$$(s|_{\pi_t^+})_{\setminus(f,i)} = (f(s_1,\ldots,s_k)|_{j\,.\,(\pi')^+_{t_j}})_{\setminus(f,i)} = (s_j|_{(\pi')^+_{t_j}})_{\setminus(f,i)}$$

and the lemma again follows from the induction hypothesis which states that $t_j|_{\pi'} = (s_j|_{(\pi')^+_{t_j}})_{\setminus(f,i)}$. In a similar way, we get $(\pi_t^+)^-_s = ((j.\pi')^+_t)^-_s = (j\,.\,(\pi')^+_t)^-_s = j\,.\,((\pi')^+_t)^-_s = j.\pi' = \pi$ by the induction hypothesis.

Finally, we regard the case where $\pi = j.\pi'$ and $\mathrm{root}(t) = g \neq f$. Then we have $s = g(s_1,\ldots,s_b)$ for some terms $s_1,\ldots,s_b$ and $t = g(t_1,\ldots,t_b)$, where $t_m = (s_m)_{\setminus(f,i)}$ for $1 \leq m \leq b$. We have

$$t|_\pi = g(t_1,\ldots,t_b)|_{j.\pi'} = t_j|_{\pi'}$$

and

$$(s|_{\pi_t^+})_{\setminus(f,i)} = (g(s_1,\ldots,s_b)|_{j\,.\,(\pi')^+_{t_j}})_{\setminus(f,i)} = (s_j|_{(\pi')^+_{t_j}})_{\setminus(f,i)}.$$

So the lemma follows from the induction hypothesis which states that $t_j|_{\pi'} = (s_j|_{(\pi')^+_{t_j}})_{\setminus(f,i)}$. In a similar way, we can also prove $(\pi_t^+)^-_s = \pi$. ◀

The next lemma shows that π_t^+ and π_s^- are also inverse operations if π_s^- is applied first, provided that t is $s_{\setminus(f,i)}$ (up to the positions of variables in $s_{\setminus(f,i)}$).

▶ **Lemma 32** (π_t^+ and π_s^- are Inverse Operations). *Let $t \in \mathcal{T}(\Sigma_{\setminus(f,i)},\mathcal{V})$ and $s \in \mathcal{T}(\Sigma,\mathcal{V})$ such that $\mathrm{root}(t|_\pi) = \mathrm{root}((s_{\setminus(f,i)})|_\pi)$ for all $\pi \in \mathrm{pos}(s_{\setminus(f,i)})$ where $(s_{\setminus(f,i)})|_\pi \notin \mathcal{V}$. Then for all positions $\pi \in \mathrm{pos}(s)$ that are not below the i-th argument of an occurrence of f in s, we have $(\pi_s^-)^+_t = \pi$.*

Proof. Again, we use induction on π. For $\pi = \varepsilon$ the lemma trivially holds.

Now let $\pi = j.\pi'$ and let $q = s_{\setminus(f,i)}$. We first regard the case where $\mathrm{root}(s) = f$ and $j > i$. Then we have $\mathrm{root}(q) = \mathrm{root}(t) = f$ and obtain

$$(\pi_s^-)^+_t = ((j-1)\,.\,(\pi')^-_{s|_j})^+_t = j\,.\,((\pi')^-_{s|_j})^+_{t|_{j-1}}.$$

By Lemma 31, we have $q|_{j-1} = (s|_{(j-1)^+_q})_{\setminus(f,i)} = (s|_j)_{\setminus(f,i)}$. Moreover, since $\mathrm{root}(t|_\pi) = \mathrm{root}(q|_\pi)$ for all $\pi \in \mathrm{pos}(q)$ where $q|_\pi \notin \mathcal{V}$, we have $\mathrm{root}((t|_{j-1})|_\pi) = \mathrm{root}((q|_{j-1})|_\pi)$ for all $\pi \in \mathrm{pos}(q|_{j-1})$ where $(q|_{j-1})|_\pi \notin \mathcal{V}$. Since π' is not below the i-th argument of an occurrence of f in $s|_j$, $j\,((\pi')^-_{s|_j})^+_{t|_{j-1}} = j\,\pi' = \pi$ follows from the induction hypothesis.

Otherwise, we get

$$(\pi_s^-)^+_t = (j\,.\,(\pi')^-_{s|_j})^+_t = j\,.\,((\pi')^-_{s|_j})^+_{t|_j}.$$

By Lemma 31, we have $q|_j = (s|_{j^+_q})_{\setminus(f,i)} = (s|_j)_{\setminus(f,i)}$. Moreover, since $\mathrm{root}(t|_\pi) = \mathrm{root}(q|_\pi)$ for all $\pi \in \mathrm{pos}(q)$ where $q|_\pi \notin \mathcal{V}$, we have $\mathrm{root}((t|_j)|_\pi) = \mathrm{root}((q|_j)|_\pi)$ for all $\pi \in \mathrm{pos}(q|_j)$ where $(q|_{j-1})|_\pi \notin \mathcal{V}$. So similar to the previous case, $j\,.\,((\pi')^-_{s|_j})^+_{t|_j} = j.\pi' = \pi$ follows from the induction hypothesis. ◀

The last auxiliary lemma shows that variables in a filtered term correspond to the same variables in the original non-filtered term.

▶ **Lemma 33** (Variables in Filtered Terms). *Let $t \in \mathcal{T}(\Sigma_{\setminus(f,i)}, \mathcal{V})$ and $s \in \mathcal{T}(\Sigma, \mathcal{V})$ with $t = s_{\setminus(f,i)}$. For any $\pi \in \text{pos}(t)$ where $t|_\pi \in \mathcal{V}$, we have $t|_\pi = s|_{\pi_t^+}$.*

Proof. By Lemma 31, we have $t|_\pi = (s|_{\pi_t^+})_{\setminus(f,i)} \in \mathcal{V}$. Since the result of filtering can only be a variable if the original term was also the same variable, we therefore have $(s|_{\pi_t^+})_{\setminus(f,i)} = s|_{\pi_t^+}$. ◀

Finally, we can prove Thm. 22.

▶ **Theorem 22** (Soundness of Argument Filtering). *Let $f \in \Sigma$, $\text{ar}_\Sigma(f) = k$, and $i \in \{1, \ldots, k\}$. Moreover, let $\mathcal{R}$ be a TRS over Σ where the following conditions hold for all rules $\ell \to r \in \mathcal{R}$:*
(a) *If $f(t_1, \ldots, t_k)$ is a subterm of ℓ, then $t_i \in \mathcal{V}$.*
(b) *For any $x \in \mathcal{V}$, there is at most one position $\pi \in \text{pos}(\ell)$ such that $\ell|_\pi = x$.*
(c) *$\mathcal{V}(r_{\setminus(f,i)}) \subseteq \mathcal{V}(\ell_{\setminus(f,i)})$.*
Then for all $n \in \mathbb{N}$, we have $\text{irc}_{\mathcal{R}_{\setminus(f,i)}}(\frac{n}{2}) \leq \text{irc}_\mathcal{R}(n)$.

Proof. To prove the theorem we show that whenever there is an innermost rewrite sequence w.r.t. $\mathcal{R}_{\setminus(f,i)}$ that starts with a basic term of size at most $\frac{n}{2}$, then there is also an innermost rewrite sequence w.r.t. $\mathcal{R}$ of at least the same length that starts with a basic term of size at most n. So let $t_1 \in \mathcal{T}(\Sigma_{\setminus(f,i)}, \mathcal{V})$ be a basic term w.r.t. $\mathcal{R}_{\setminus(f,i)}$ that starts a rewrite sequence

$$t_1 \xrightarrow{\text{i}}_{\mathcal{R}_{\setminus(f,i)}} \ldots \xrightarrow{\text{i}}_{\mathcal{R}_{\setminus(f,i)}} t_m.$$

Let s_1 result from t_1 by adding a fresh variable as i-th argument for each occurrence of f. Note that we have $|s_1| \leq 2 \cdot |t_1|$ since t_1 contains at most $|t_1|$ occurrences of f and s_1 contains an additional variable for each occurrence of f.

Thus, to prove the theorem it suffices to show that s_1 has an innermost rewrite sequence w.r.t. $\mathcal{R}$ that has at least length m. If s_1 starts an infinite innermost $\mathcal{R}$-reduction, then the claim obviously holds. Thus, we now regard the case where s_1 is innermost terminating w.r.t. $\mathcal{R}$ and inductively construct an innermost rewrite sequence

$$s_1 \xrightarrow{\text{i}}^+_{\mathcal{R}} \ldots \xrightarrow{\text{i}}^+_{\mathcal{R}} s_m$$

such that for all $j \in \{1, \ldots, m\}$, we have $s_j \in \mathcal{T}(\Sigma, \mathcal{V})$ and $(s_j)_{\setminus(f,i)} = t_j$. Moreover, s_j does not contain any redex below the i-th argument of any occurring f.

In the induction base, by the construction of s_1 we clearly have $(s_1)_{\setminus(f,i)} = t_1$. Moreover, s_1 only has variables on the i-th arguments of f.

In the induction step, we assume that we have already constructed $s_1, \ldots, s_j$ and now our goal is to construct s_{j+1}. Let $\ell \to r \in \mathcal{R}_{\setminus(f,i)}$ be a rule that reduces t_j to t_{j+1} at position π by an innermost rewrite step, and let $\overline{\ell} \to \overline{r} \in \mathcal{R}$ be a rule such that $\overline{\ell}_{\setminus(f,i)} = \ell$ and $\overline{r}_{\setminus(f,i)} = r$. Let $\overline{\pi} = \pi_{t_j}^+$, i.e., this is the position where the rule $\overline{\ell} \to \overline{r}$ should be applied to s_j. We now show the following observation:

$$\text{For all } \tau \in \text{pos}(\overline{\ell}) \text{ where } \overline{\ell}|_\tau \notin \mathcal{V}, \text{ we have } \overline{\pi}.\tau \in \text{pos}(s_j) \text{ and } \text{root}(\overline{\ell}|_\tau) = \text{root}(s_j|_{\overline{\pi}.\tau}). \quad (12)$$

To show (12), we prove that

$$\overline{\pi}.\tau \in \text{pos}(s_j) \text{ and } \text{root}((\overline{\ell}|_\tau)_{\setminus(f,i)}) = \text{root}((s_j|_{\overline{\pi}.\tau})_{\setminus(f,i)}). \quad (13)$$

Note that (13) indeed implies (12): If $\text{root}(\overline{\ell}|_\tau) \in \Sigma \setminus \{f\}$, then $\text{root}(\overline{\ell}|_\tau) = \text{root}((\overline{\ell}|_\tau)_{\setminus(f,i)})$. By (13), we have $\text{root}((\overline{\ell}|_\tau)_{\setminus(f,i)}) = \text{root}((s_j|_{\overline{\pi}.\tau})_{\setminus(f,i)}) = \text{root}(s_j|_{\overline{\pi}.\tau})$. Otherwise if $\text{root}(\overline{\ell}|_\tau)$

is the symbol f of arity k, then $\mathrm{root}((\overline{\ell}|_\tau)_{\setminus(f,i)})$ is the symbol f of arity $k-1$. By (13), $\mathrm{root}((s_j|_{\overline{\pi}.\tau})_{\setminus(f,i)})$ is also the symbol f of arity $k-1$, which implies that $\mathrm{root}(s_j|_{\overline{\pi}.\tau})$ is f with arity k.

To prove (13), note that by Requirement (a), the position $\tau \in \mathrm{pos}(\overline{\ell})$ with $\overline{\ell}|_\tau \notin \mathcal{V}$ cannot be below the i-th argument of the symbol f in $\overline{\ell}$. Moreover, we have $\ell = \overline{\ell}_{\setminus(f,i)}$. Hence, we can apply Lemma 32 and obtain

$$\begin{array}{rcll}
\mathrm{root}((\overline{\ell}|_\tau)_{\setminus(f,i)}) & = & \mathrm{root}((\overline{\ell}|_{(\tau^-_{\overline{\ell}})^+_\ell})_{\setminus(f,i)}) & \text{by Lemma 32} \\
& = & \mathrm{root}(\ell|_{\tau^-_{\overline{\ell}}}) & \text{by Lemma 31} \\
& = & \mathrm{root}(t_j|_{\pi\,.\,\tau^-_{\overline{\ell}}}) & \text{as } \ell \text{ matches } t_j|_\pi \\
& = & \mathrm{root}((s_j|_{(\pi\,.\,\tau^-_{\overline{\ell}})^+_{t_j}})_{\setminus(f,i)}) & \text{by Lemma 31, as } t_j = (s_j)_{\setminus(f,i)} \\
& = & \mathrm{root}((s_j|_{\overline{\pi}\,.\,(\tau^-_{\overline{\ell}})^+_{t_j|_\pi}})_{\setminus(f,i)}) & \\
& = & \mathrm{root}((s_j|_{\overline{\pi}\,.\,\tau})_{\setminus(f,i)}) & \text{by Lemma 32}
\end{array}$$

In the last step, Lemma 32 can be applied since $\mathrm{root}(t_j|_{\pi.\delta}) = \mathrm{root}(\ell|_\delta) = \mathrm{root}((\overline{\ell}_{\setminus(f,i)})|_\delta)$ holds for all $\delta \in \mathrm{pos}(\ell)$ where $\ell|_\delta \notin \mathcal{V}$.

Note that (12) implies that for those $\tau \in \mathrm{pos}(\overline{\ell})$ where $\overline{\ell}|_\tau \in \mathcal{V}$, we can also conclude $\overline{\pi}.\tau \in \mathrm{pos}(s_j)$, because we must have $\tau = \tau'.m$ for some position τ' and $m \in \mathbb{N}$. By (12), the function symbol at the position τ' in $\overline{\ell}$ is the same as the function symbol at the position $\pi.\tau'$ in s_j. Hence, we can define the substitution

$$\sigma = [\,(\overline{\ell}|_\tau)/(s_j|_{\overline{\pi}.\tau}) \mid \tau \in \mathrm{pos}(\overline{\ell}),\ \overline{\ell}|_\tau \in \mathcal{V}\,]. \tag{14}$$

By Requirement (b), $\overline{\ell}$ is linear and thus, σ is well defined. Now (12) implies

$$\overline{\ell}\sigma = s_j|_{\overline{\pi}}, \tag{15}$$

i.e., the left-hand side of the rule $\overline{\ell} \to \overline{r}$ matches the subterm of s_j at position $\overline{\pi}$ using the matcher σ.

After having shown (15), we now investigate the connection between σ and the substitution θ that was used for the rewrite step $t_j \xrightarrow{\mathrm{i}}_{\mathcal{R}_{\setminus(f,i)}} t_{j+1}$. Since this rewrite step was performed with the rule $\ell \to r$ at position π, we have $\ell\theta = t_j|_\pi$ and $t_j[r\theta]_\pi = t_{j+1}$. By Requirement (c), we get

$$\theta = [\,(\ell|_\tau)/(t_j|_{\pi.\tau}) \mid \tau \in \mathrm{pos}(\ell),\ \ell|_\tau \in \mathcal{V}\,]. \tag{16}$$

By Lemma 33, $\ell|_\tau \in \mathcal{V}$ implies $\overline{\ell}|_{\tau^+_\ell} = \ell|_\tau \in \mathcal{V}$. So if $x \in \mathrm{dom}(\theta)$, we also have $x \in \mathrm{dom}(\sigma)$. Moreover, then we have $\theta(x) = (\sigma(x))_{\setminus(f,i)}$. To see this, consider an $x \in \mathrm{dom}(\theta)$. So we have $x = \ell|_\tau = \overline{\ell}|_{\tau^+_\ell}$ for some $\tau \in \mathrm{pos}(\ell)$. Then we obtain

$$\begin{array}{rcll}
(\sigma(x))_{\setminus(f,i)} & = & (\sigma(\overline{\ell}|_{\tau^+_\ell}))_{\setminus(f,i)} & \\
& = & (s_j|_{\overline{\pi}\,.\,\tau^+_\ell})_{\setminus(f,i)} & \\
& = & (s_j|_{\overline{\pi}\,.\,\tau^+_{t_j|_\pi}})_{\setminus(f,i)} & \text{as } \tau \in \mathrm{pos}(\ell) \text{ and } \ell \text{ matches } t_j|_\pi \\
& = & (s_j|_{(\pi.\tau)^+_{t_j}})_{\setminus(f,i)} & \\
& = & t_j|_{\pi.\tau} & \text{by Lemma 31 as } t_j = (s_j)_{\setminus(f,i)} \\
& = & \theta(\ell|_\tau) & \\
& = & \theta(x) &
\end{array}$$

So we have

$$\theta = [x/t_{\setminus(f,i)} \mid x \in \mathcal{V}(\ell),\ x/t \in \sigma]. \tag{17}$$

Let s_{j+1} result from $s_j[\overline{r}\sigma]_{\overline{\pi}}$ by innermost reducing all subterms on i-th arguments of f to normal form. This is possible since s_1 is innermost terminating. Then by (15) we have $s_j = s_j[\overline{\ell}\sigma]_{\overline{\pi}} \to_{\mathcal{R}} s_j[\overline{r}\sigma]_{\overline{\pi}} \xrightarrow{\text{i}}{}^{*}_{\mathcal{R}} s_{j+1}$ and s_{j+1} does not contain any redex below the i-th argument of any occurring f. Moreover, we have

$$\begin{array}{rcll} (s_{j+1})_{\setminus(f,i)} & = & (s_j[\overline{r}\sigma]_{\overline{\pi}})_{\setminus(f,i)} & \text{as } s_{j+1} \text{ only differs from } s_j[\overline{r}\sigma]_{\overline{\pi}} \text{ on } f\text{'s } i\text{-th arguments} \\ & = & (s_j)_{\setminus(f,i)}[(\overline{r}\sigma)_{\setminus(f,i)}]_{\pi} & \\ & = & t_j[(\overline{r}\sigma)_{\setminus(f,i)}]_{\pi} & \\ & = & t_j[\overline{r}_{\setminus(f,i)}\theta]_{\pi} & \text{due to (17)} \\ & = & t_j[r\theta]_{\pi} & \\ & = & t_{j+1}. & \end{array}$$

It remains to show that $s_j = s_j[\overline{\ell}\sigma]_{\overline{\pi}} \to_{\mathcal{R}} s_j[\overline{r}\sigma]_{\overline{\pi}}$ is an innermost rewrite step. To this end, assume that there is a $\delta \in \text{pos}(s_j)$ such that $\overline{\pi}$ is a proper prefix of δ and $s_j|_\delta$ is a redex. Then we have

$$(s_j|_\delta)_{\setminus(f,i)} = t_j|_{\delta^-_{s_j}}. \tag{18}$$

To prove (18), note that s_j does not contain any redex below the i-th argument of f. Hence, δ is not below any i-th argument of f in s_j and thus, Lemma 32 implies $(\delta^-_{s_j})^+_{t_j} = \delta$. Hence, we have

$$\begin{array}{rll} & (s_j|_\delta)_{\setminus(f,i)} & \\ = & (s_j|_{(\delta^-_{s_j})^+_{t_j}})_{\setminus(f,i)} & \text{by Lemma 32} \\ = & t_j|_{\delta^-_{s_j}} & \text{by Lemma 31} \end{array}$$

which proves (18).

If $s_j|_\delta$ is a redex, then there exists a rule $u \to v \in \mathcal{R}$ and a substitution μ such that $u\mu = s_j|_\delta$. This implies $(u\mu)_{\setminus(f,i)} = (s_j|_\delta)_{\setminus(f,i)}$. Note that $(u\mu)_{\setminus(f,i)} = u_{\setminus(f,i)}\mu'$ for the substitution μ' with $\mu'(x) = (\mu(x))_{\setminus(f,i)}$ for all $x \in \mathcal{V}$. So the left-hand side $u_{\setminus(f,i)}$ of a rule from $\mathcal{R}_{\setminus(f,i)}$ matches $(s_j|_\delta)_{\setminus(f,i)}$. By (18), this means that t_j contains a redex at the position $\delta^-_{s_j}$.

As $\overline{\pi}$ is a proper prefix of δ, we have $\delta = \overline{\pi}.\delta'$ for some $\delta' \neq \varepsilon$. This implies $\delta^-_{s_j} = (\overline{\pi}.\delta')^-_{s_j} = \overline{\pi}^-_{s_j} . (\delta')^-_{s_j} = \pi . (\delta')^-_{s_j}$ by Lemma 31, where $(\delta')^-_{s_j} \neq \varepsilon$. Thus, π is a proper prefix of $\delta^-_{s_j}$ and t_j contains a redex at the position $\delta^-_{s_j}$. This is a contradiction to the fact that the reduction from t_j to t_{j+1} at position π was an innermost step. ◀

▶ **Theorem 25** (Bounds for Indefinite Rewrite Lemmas). *Let $\xrightarrow{\text{i}}_{\mathcal{R}}$ and $\xrightarrow{\text{i}}_{(\mathcal{R},\text{IH})}$ be restricted such that redexes may not contain the symbol $\star$ and let* ih, ib, *and* is *be defined as in Def. 11. Here, for an indefinite rewrite lemma $s \xrightarrow{\text{i}}{}^{rt(\overline{n})} \star$ with $n \in \mathcal{V}(s)$, we say that any rewrite sequence $s[n/0] = u_1 \xrightarrow{\text{i}}_{\mathcal{R}} u_2 \xrightarrow{\text{i}}_{\mathcal{R}} \dots \xrightarrow{\text{i}}_{\mathcal{R}} u_{b+1}$ "proves" the induction base and any rewrite sequence $s[n/n+1] = v_1 \xrightarrow{\text{i}}_{(\mathcal{R},\text{IH})} v_2 \xrightarrow{\text{i}}_{(\mathcal{R},\text{IH})} \cdots \xrightarrow{\text{i}}_{(\mathcal{R},\text{IH})} v_{k+1}$ "proves" the induction step, where* IH *is the rule $s \to \star$. Then Thm. 12 and Thm. 14 on explicit and asymptotic runtimes hold for any definite or indefinite rewrite lemma.*

Proof. It is easy to show that Lemma 27 also holds for the modified variant of $\xrightarrow{\text{i}}_{\mathcal{R}}$, i.e., $\xrightarrow{\text{i}}_{\mathcal{R}}$ is still closed under instantiations of variables with natural numbers. The reason is that if $\ell\sigma$ does not contain any occurrence of $\star$, then this also holds for $\ell\sigma\mu$ if $\mu : \mathcal{V}(s) \to \mathbb{N}$.

Lemma 28 has to be adapted to take the new symbol $\star$ into account. Let $\mathcal{R}$, ℓ, r, s and t be as in Lemma 28, where ℓ does not contain $\star$. Moreover, s and r may only contain $\star$ at positions that do not have the type $\mathbb{N}$. Moreover, let $\widehat{s}$ be a well-typed ground term from $\mathcal{T}(\Sigma')$ which results from $s{\downarrow_{\mathcal{G}/\mathcal{A}}}$ by replacing all occurrences of $\star$ by arbitrary (possibly different) terms. Then one can show the following claims:

(a) If $s \xrightarrow{\mathrm{i}}_{\mathcal{R}} t$ and this reduction is done using a rule $\ell \to r \in \mathcal{R}$, then there exists a well-typed ground term $\widehat{t} \in \mathcal{T}(\Sigma')$ that results from $t{\downarrow_{\mathcal{G}/\mathcal{A}}}$ by replacing all occurrences of $\star$ by suitable (possibly different) terms such that $\widehat{s} \xrightarrow{\mathrm{i}}_{\mathcal{R}} \widehat{t}$.

(b) If $s \xrightarrow{\mathrm{i}}_{\mathcal{R}} t$ and this reduction is done using the rewrite lemma $\ell \xrightarrow{\mathrm{i}}^{rt(\overline{n})} r$ and the substitution σ, then there exists a well-typed ground term $\widehat{t} \in \mathcal{T}(\Sigma')$ that results from $t{\downarrow_{\mathcal{G}/\mathcal{A}}}$ by replacing all occurrences of $\star$ by suitable (possibly different) terms such that $\widehat{s} \xrightarrow{\mathrm{i}}_{\mathcal{R}}^{rt(\overline{n}\sigma)} \widehat{t}$.

(c) Let $\widehat{r}$ be a well-typed ground term from $\mathcal{T}(\Sigma')$ which results from $r{\downarrow_{\mathcal{G}/\mathcal{A}}}$ by replacing all occurrences of $\star$ by arbitrary (possibly different) terms. If $s \mapsto_{\ell\to r} t$, then there exists a well-typed ground term $\widehat{t} \in \mathcal{T}(\Sigma')$ that results from $t{\downarrow_{\mathcal{G}/\mathcal{A}}}$ by replacing all occurrences of $\star$ by suitable (possibly different) terms such that $\widehat{s} \twoheadrightarrow_{(\ell{\downarrow_{\mathcal{G}/\mathcal{A}}} \to \widehat{r})} \widehat{t}$.

These claims are proved similar to the proof of Lemma 28. Let us first regard Claim (b). As in the proof of Lemma 28, we have $s{\downarrow_{\mathcal{G}/\mathcal{A}}} = s'{\downarrow_{\mathcal{G}/\mathcal{A}}} = (s'[s'|_\pi]_\pi){\downarrow_{\mathcal{G}/\mathcal{A}}} = s'{\downarrow_{\mathcal{G}/\mathcal{A}}} [(s'|_\pi){\downarrow_{\mathcal{G}/\mathcal{A}}}]_\pi = s'{\downarrow_{\mathcal{G}/\mathcal{A}}} [\ell\sigma{\downarrow_{\mathcal{G}/\mathcal{A}}}]_\pi$. Since $\star$ only occurs in s on positions whose type is not $\mathbb{N}$, $\mathcal{G}$ and $\mathcal{A}$ cannot be applied on or above these positions. As $\widehat{s}$ results from replacing the occurrences of $\star$ in $s{\downarrow_{\mathcal{G}/\mathcal{A}}}$ by arbitrary terms, there is a term $\widehat{s}'$ that results from $s'{\downarrow_{\mathcal{G}/\mathcal{A}}}$ by replacing $\star$ by suitable terms such that $\widehat{s} = \widehat{s}'[\ell\sigma{\downarrow_{\mathcal{G}/\mathcal{A}}}]_\pi$.

Since $\ell \to r \in \mathcal{L}$, $\ell \to r$ corresponds to a rewrite lemma $\ell \xrightarrow{\mathrm{i}}^{rt(\overline{n})} r$. Thus, there is a term $\widehat{r}$ resulting from replacing $\star$ in $r\sigma{\downarrow_{\mathcal{G}/\mathcal{A}}}$ by suitable terms such that $\ell\sigma{\downarrow_{\mathcal{G}/\mathcal{A}}} \xrightarrow{\mathrm{i}}_{\mathcal{R}}^{rt(\overline{n}\sigma)} \widehat{r}$. Hence, we obtain $\widehat{s} = \widehat{s}'[\ell\sigma{\downarrow_{\mathcal{G}/\mathcal{A}}}]_\pi \xrightarrow{\mathrm{i}}_{\mathcal{R}}^{rt(\overline{n}\sigma)} \widehat{s}'[\widehat{r}]_\pi = (\widehat{s'[r\sigma]_\pi}){\downarrow_{\mathcal{G}/\mathcal{A}}}$, where $(\widehat{s'[r\sigma]_\pi})$ results from $s'[r\sigma]_\pi$ by replacing all occurrences of $\star$ by suitable terms. Let $\widehat{t} = (\widehat{s'[r\sigma]_\pi}){\downarrow_{\mathcal{G}/\mathcal{A}}}$. Note that this term can be obtained by replacing all occurrences of $\star$ in $(s'[r\sigma]_\pi){\downarrow_{\mathcal{G}/\mathcal{A}}}$ by suitable terms. Since $(s'[r\sigma]_\pi){\downarrow_{\mathcal{G}/\mathcal{A}}} = t{\downarrow_{\mathcal{G}/\mathcal{A}}}$ as in the proof of Lemma 28, this finishes the proof of (b).

The adaption of the proof of (a) is analogous to the adaption of (b). Hence, it remains to show the claim (c). As in the proof of Lemma 28, we have $s{\downarrow_{\mathcal{G}/\mathcal{A}}} = s'{\downarrow_{\mathcal{G}/\mathcal{A}}} = s'{\downarrow_{\mathcal{G}/\mathcal{A}}} [\ell{\downarrow_{\mathcal{G}/\mathcal{A}}}]_\pi$. So as in the adaption of (b), there is a term $\widehat{s}'$ that results from $s'{\downarrow_{\mathcal{G}/\mathcal{A}}}$ by replacing $\star$ by suitable terms such that $\widehat{s} = \widehat{s}'[\ell{\downarrow_{\mathcal{G}/\mathcal{A}}}]_\pi$.

Thus, we obtain $\widehat{s} = \widehat{s}'[\ell{\downarrow_{\mathcal{G}/\mathcal{A}}}]_\pi \twoheadrightarrow_{(\ell{\downarrow_{\mathcal{G}/\mathcal{A}}} \to \widehat{r})} \widehat{s}'[\widehat{r}]_\pi = (\widehat{s'[r]_\pi}){\downarrow_{\mathcal{G}/\mathcal{A}}}$, where $(\widehat{s'[r]_\pi})$ results from $s'[r\sigma]_\pi$ by replacing all occurrences of $\star$ by suitable terms. Let $\widehat{t} = (\widehat{s'[r]_\pi}){\downarrow_{\mathcal{G}/\mathcal{A}}}$. Note that this term can be obtained by replacing all occurrences of $\star$ in $(s'[r]_\pi){\downarrow_{\mathcal{G}/\mathcal{A}}}$ by suitable terms. Since $(s'[r]_\pi){\downarrow_{\mathcal{G}/\mathcal{A}}} = t{\downarrow_{\mathcal{G}/\mathcal{A}}}$ as in the proof of Lemma 28, this finishes the proof of (c).

Now one can now show that the recurrence equations (9) still hold. Let $s \xrightarrow{\mathrm{i}}^{rt(\overline{n})} t$ be a (definite or indefinite) rewrite lemma. For any $\mu : \mathcal{V}(s) \to \mathbb{N}$ which instantiates all variables of s by natural numbers one has to prove that there exists a well-typed ground term $\widehat{t} \in \mathcal{T}(\Sigma')$ that results from $t\mu{\downarrow_{\mathcal{G}/\mathcal{A}}}$ by replacing all occurrences of $\star$ by suitable terms such that $s\mu{\downarrow_{\mathcal{G}/\mathcal{A}}} \xrightarrow{\mathrm{i}}_{\mathcal{R}}^{rt(\overline{n}\mu)} \widehat{t}$ holds. This proof works analogously to the proof of Thm. 12. So we again perform induction on $n\mu$.

In the induction base case, we have $n\mu = 0$. If the reduction $s[n/0] \xrightarrow{\mathrm{i}}_{\mathcal{R}}^{*} t[n/0]$ has length 0, then we cannot have $t = \star$ (since $\star$ does not occur in the left-hand side of rewrite lemmas). Thus, the proof does not have to be adapted in that case. If $s[n/0] = u_1 \xrightarrow{\mathrm{i}}_{\mathcal{R}} \ldots \xrightarrow{\mathrm{i}}_{\mathcal{R}} u_{b+1} = t[n/0]$ for $b \geq 1$, then by Lemma 27, we again obtain $s\mu = s[n/0]\mu = u_1\mu \xrightarrow{\mathrm{i}}_{\mathcal{R}} \ldots \xrightarrow{\mathrm{i}}_{\mathcal{R}} u_{b+1}\mu = t[n/0]\mu = t\mu$. The adapted variant of Lemma 28 (a) and (b) implies $s\mu{\downarrow_{\mathcal{G}/\mathcal{A}}} = u_1\mu{\downarrow_{\mathcal{G}/\mathcal{A}}} \overset{\mathsf{i}}{\to}_{\mathcal{R}}^{\mathit{rt}(\overline{y}_1\sigma_1\mu)} \widehat{u_2\mu{\downarrow_{\mathcal{G}/\mathcal{A}}}} \overset{\mathsf{i}}{\to}_{\mathcal{R}}^{\mathit{rt}(\overline{y}_2\sigma_2\mu)} \dots \overset{\mathsf{i}}{\to}_{\mathcal{R}}^{\mathit{rt}(\overline{y}_b\sigma_b\mu)} \widehat{u_{b+1}\mu{\downarrow_{\mathcal{G}/\mathcal{A}}}}$, where each $\widehat{u_{j+1}\mu{\downarrow_{\mathcal{G}/\mathcal{A}}}}$ results from $u_{j+1}\mu{\downarrow_{\mathcal{G}/\mathcal{A}}}$ by replacing all occurrences of $\star$ by suitable terms. Let $\widehat{t} = \widehat{u_{b+1}\mu{\downarrow_{\mathcal{G}/\mathcal{A}}}}$. Since $u_{b+1}\mu = t\mu$, $\widehat{t}$ results from $t\mu{\downarrow_{\mathcal{G}/\mathcal{A}}}$ by replacing all occurrences of $\star$ by suitable terms. Thus, $s\mu{\downarrow_{\mathcal{G}/\mathcal{A}}} \overset{\mathsf{i}}{\to}_{\mathcal{R}}^{\mathit{ib}(\widetilde{n\mu})} \widehat{t}$ or in other words, $s\mu{\downarrow_{\mathcal{G}/\mathcal{A}}} \overset{\mathsf{i}}{\to}_{\mathcal{R}}^{\mathit{rt}(\overline{n}\mu)} \widehat{t}$.

In the induction step case, we have $n\mu > 0$. In the case where the reduction $s[n/n{+}1] \overset{\mathsf{i}}{\rightharpoonup}_{\mathcal{R}}^{*} t[n/n+1]$ has length 0, we again know that $t \neq \star$ and thus, the proof does not have to be adapted. Thus, we now regard the case $s[n/n{+}1] = v_1 \overset{\mathsf{i}}{\rightharpoonup}_{(\mathcal{R},\mathrm{IH})} \dots \overset{\mathsf{i}}{\rightharpoonup}_{(\mathcal{R},\mathrm{IH})} v_{k+1} = t[n/n{+}1]$ for $k \geq 1$. Let μ' be like μ for all $\mathcal{V}(s) \setminus \{n\}$ and $n\mu' = n\mu - 1$. By Lemma 27, $\overset{\mathsf{i}}{\rightharpoonup}_{\mathcal{R}}$ is stable and thus, we obtain $s[n/n+1]\mu' = v_1\mu' \overset{\mathsf{i}}{\rightharpoonup}_{(\mathcal{R},\mathrm{IH}\mu')} \dots \overset{\mathsf{i}}{\rightharpoonup}_{(\mathcal{R},\mathrm{IH}\mu')} v_{k+1}\mu' = t[n/n+1]\mu'$.

If $v_j\mu' \overset{\mathsf{i}}{\rightharpoonup}_{\mathcal{R}} v_{j+1}\mu'$, then the adapted variant of Lemma 28 (a) and (b) implies the following: If $\widehat{v_j\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}}$ results from $v_j\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}$ by replacing all occurrences of $\star$ by arbitrary terms, then there exists a term $\widehat{v_{j+1}\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}}$ which results from $v_{j+1}\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}$ by replacing all occurrences of $\star$ by suitable terms, such that $\widehat{v_j\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}} \overset{\mathsf{i}}{\to}_{\mathcal{R}}^{\mathit{rt}'_j(\overline{z}_j\theta_j\mu')} \widehat{v_{j+1}\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}}$. Otherwise, if $v_j\mu' \mapsto_{\mathrm{IH}\mu'} v_{j+1}\mu'$, then by the adaption of Lemma 28 (c) we have the following: If $\widehat{v_j\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}}$ results from $v_j\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}$ and $\widehat{t\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}}$ results from $t\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}$ by replacing all occurrences of $\star$ by arbitrary terms, then there exists a term $\widehat{v_{j+1}\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}}$ which results from $v_{j+1}\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}$ by replacing all occurrences of $\star$ by suitable terms, such that $\widehat{v_j\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}} \twoheadrightarrow_{(s\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}} \to \widehat{t\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}})} \widehat{v_{j+1}\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}}$. Note that the induction hypothesis implies $s\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}} \overset{\mathsf{i}}{\to}_{\mathcal{R}}^{\mathit{rt}(\overline{n}\mu')} \widehat{t\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}}$, if $\widehat{t\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}}$ results from $t\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}$ by replacing all occurrences of $\star$ by suitable terms. This means that $\widehat{v_j\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}} \twoheadrightarrow_{(s\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}} \to \widehat{t\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}})} \widehat{v_{j+1}\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}}$ implies $\widehat{v_j\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}} \overset{\mathsf{i}}{\to}_{\mathcal{R}}^{\mathit{rt}(\overline{n}\mu')} \widehat{v_{j+1}\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}}$. Since there are ih many of these steps, we finally get $s[n/n+1]\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}} \overset{\mathsf{i}}{\to}_{\mathcal{R}}^{\mathit{ih}\cdot\mathit{rt}(\overline{n}\mu')+\mathit{is}(\overline{n}\mu')} \widehat{v_{k+1}\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}}$. Let $\widehat{t} = \widehat{v_{k+1}\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}}$. Since $v_{k+1}\mu' = t[n/n+1]\mu'$, t' results from $t\mu{\downarrow_{\mathcal{G}/\mathcal{A}}} = t[n/n+1]\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}$ by replacing all occurrences of $\star$ by suitable terms. Hence, $s\mu{\downarrow_{\mathcal{G}/\mathcal{A}}} = s[n/n+1]\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}} \overset{\mathsf{i}}{\to}_{\mathcal{R}}^{\mathit{rt}(\overline{n}[n/n+1]\mu')} \widehat{v_{k+1}\mu'{\downarrow_{\mathcal{G}/\mathcal{A}}}} = \widehat{t}$. This proves the desired claim, since $\mathit{rt}(\overline{n}\mu) = \mathit{rt}(\overline{n}[n/n+1]\mu')$.

So the recurrence equations (9) for $\mathit{rt}(\overline{n})$ also hold when regarding indefinite rewrite lemmas. Hence, the closed form for $\mathit{rt}(\overline{n})$ given in Thm. 12 satisfies this recurrence equation and the asymptotic bounds of Thm. 14 hold for $\mathit{rt}_{\mathbb{N}}$. ◀

Aachener Informatik-Berichte

This list contains all technical reports published during the past three years. A complete list of reports dating back to 1987 is available from:

http://aib.informatik.rwth-aachen.de/

To obtain copies please consult the above URL or send your request to:

Informatik-Bibliothek, RWTH Aachen, Ahornstr. 55, 52056 Aachen, Email: `biblio@informatik.rwth-aachen.de`

2011-01 * Fachgruppe Informatik: Jahresbericht 2011
2011-02 Marc Brockschmidt, Carsten Otto, Jürgen Giesl: Modular Termination Proofs of Recursive Java Bytecode Programs by Term Rewriting
2011-03 Lars Noschinski, Fabian Emmes, Jürgen Giesl: A Dependency Pair Framework for Innermost Complexity Analysis of Term Rewrite Systems
2011-04 Christina Jansen, Jonathan Heinen, Joost-Pieter Katoen, Thomas Noll: A Local Greibach Normal Form for Hyperedge Replacement Grammars
2011-06 Johannes Lotz, Klaus Leppkes, and Uwe Naumann: dco/c++ - Derivative Code by Overloading in C++
2011-07 Shahar Maoz, Jan Oliver Ringert, Bernhard Rumpe: An Operational Semantics for Activity Diagrams using SMV
2011-08 Thomas Ströder, Fabian Emmes, Peter Schneider-Kamp, Jürgen Giesl, Carsten Fuhs: A Linear Operational Semantics for Termination and Complexity Analysis of ISO Prolog
2011-09 Markus Beckers, Johannes Lotz, Viktor Mosenkis, Uwe Naumann (Editors): Fifth SIAM Workshop on Combinatorial Scientific Computing
2011-10 Markus Beckers, Viktor Mosenkis, Michael Maier, Uwe Naumann: Adjoint Subgradient Calculation for McCormick Relaxations
2011-11 Nils Jansen, Erika Ábrahám, Jens Katelaan, Ralf Wimmer, Joost-Pieter Katoen, Bernd Becker: Hierarchical Counterexamples for Discrete-Time Markov Chains
2011-12 Ingo Felscher, Wolfgang Thomas: On Compositional Failure Detection in Structured Transition Systems
2011-13 Michael Förster, Uwe Naumann, Jean Utke: Toward Adjoint OpenMP
2011-14 Daniel Neider, Roman Rabinovich, Martin Zimmermann: Solving Muller Games via Safety Games
2011-16 Niloofar Safiran, Uwe Naumann: Toward Adjoint OpenFOAM
2011-17 Carsten Fuhs: SAT Encodings: From Constraint-Based Termination Analysis to Circuit Synthesis
2011-18 Kamal Barakat: Introducing Timers to pi-Calculus
2011-19 Marc Brockschmidt, Thomas Ströder, Carsten Otto, Jürgen Giesl: Automated Detection of Non-Termination and NullPointerExceptions for Java Bytecode

2011-24 Callum Corbett, Uwe Naumann, Alexander Mitsos: Demonstration of a Branch-and-Bound Algorithm for Global Optimization using McCormick Relaxations

2011-25 Callum Corbett, Michael Maier, Markus Beckers, Uwe Naumann, Amin Ghobeity, Alexander Mitsos: Compiler-Generated Subgradient Code for McCormick Relaxations

2011-26 Hongfei Fu: The Complexity of Deciding a Behavioural Pseudometric on Probabilistic Automata

2012-01 Fachgruppe Informatik: Annual Report 2012

2012-02 Thomas Heer: Controlling Development Processes

2012-03 Arne Haber, Jan Oliver Ringert, Bernhard Rumpe: MontiArc - Architectural Modeling of Interactive Distributed and Cyber-Physical Systems

2012-04 Marcus Gelderie: Strategy Machines and their Complexity

2012-05 Thomas Ströder, Fabian Emmes, Jürgen Giesl, Peter Schneider-Kamp, and Carsten Fuhs: Automated Complexity Analysis for Prolog by Term Rewriting

2012-06 Marc Brockschmidt, Richard Musiol, Carsten Otto, Jürgen Giesl: Automated Termination Proofs for Java Programs with Cyclic Data

2012-07 André Egners, Björn Marschollek, and Ulrike Meyer: Hackers in Your Pocket: A Survey of Smartphone Security Across Platforms

2012-08 Hongfei Fu: Computing Game Metrics on Markov Decision Processes

2012-09 Dennis Guck, Tingting Han, Joost-Pieter Katoen, and Martin R. Neuhäußer: Quantitative Timed Analysis of Interactive Markov Chains

2012-10 Uwe Naumann and Johannes Lotz: Algorithmic Differentiation of Numerical Methods: Tangent-Linear and Adjoint Direct Solvers for Systems of Linear Equations

2012-12 Jürgen Giesl, Thomas Ströder, Peter Schneider-Kamp, Fabian Emmes, and Carsten Fuhs: Symbolic Evaluation Graphs and Term Rewriting — A General Methodology for Analyzing Logic Programs

2012-15 Uwe Naumann, Johannes Lotz, Klaus Leppkes, and Markus Towara: Algorithmic Differentiation of Numerical Methods: Tangent-Linear and Adjoint Solvers for Systems of Nonlinear Equations

2012-16 Georg Neugebauer and Ulrike Meyer: SMC-MuSe: A Framework for Secure Multi-Party Computation on MultiSets

2012-17 Viet Yen Nguyen: Trustworthy Spacecraft Design Using Formal Methods

2013-01 * Fachgruppe Informatik: Annual Report 2013

2013-02 Michael Reke: Modellbasierte Entwicklung automobiler Steuerungssysteme in Klein- und mittelständischen Unternehmen

2013-03 Markus Towara and Uwe Naumann: A Discrete Adjoint Model for OpenFOAM

2013-04 Max Sagebaum, Nicolas R. Gauger, Uwe Naumann, Johannes Lotz, and Klaus Leppkes: Algorithmic Differentiation of a Complex C++ Code with Underlying Libraries

2013-05 Andreas Rausch and Marc Sihling: Software & Systems Engineering Essentials 2013

2013-06 Marc Brockschmidt, Byron Cook, and Carsten Fuhs: Better termination proving through cooperation
2013-07 André Stollenwerk: Ein modellbasiertes Sicherheitskonzept für die extrakorporale Lungenunterstützung
2013-08 Sebastian Junges, Ulrich Loup, Florian Corzilius and Erika Ábrahám: On Gröbner Bases in the Context of Satisfiability-Modulo-Theories Solving over the Real Numbers
2013-10 Joost-Pieter Katoen, Thomas Noll, Thomas Santen, Dirk Seifert, and Hao Wu: Performance Analysis of Computing Servers using Stochastic Petri Nets and Markov Automata
2013-12 Marc Brockschmidt, Fabian Emmes, Stephan Falke, Carsten Fuhs, and Jürgen Giesl: Alternating Runtime and Size Complexity Analysis of Integer Programs
2013-13 Michael Eggert, Roger Häußling, Martin Henze, Lars Hermerschmidt, René Hummen, Daniel Kerpen, Antonio Navarro Pérez, Bernhard Rumpe, Dirk Thißen, and Klaus Wehrle: SensorCloud: Towards the Interdisciplinary Development of a Trustworthy Platform for Globally Interconnected Sensors and Actuators
2013-14 Jörg Brauer: Automatic Abstraction for Bit-Vectors using Decision Procedures
2013-19 Florian Schmidt, David Orlea, and Klaus Wehrle: Support for error tolerance in the Real-Time Transport Protocol
2013-20 Jacob Palczynski: Time-Continuous Behaviour Comparison Based on Abstract Models
2014-01 * Fachgruppe Informatik: Annual Report 2014
2014-02 Daniel Merschen: Integration und Analyse von Artefakten in der modellbasierten Entwicklung eingebetteter Software
2014-03 Uwe Naumann, Klaus Leppkes, and Johannes Lotz: dco/c++ User Guide
2014-04 Namit Chaturvedi: Languages of Infinite Traces and Deterministic Asynchronous Automata
2014-05 Thomas Ströder, Jürgen Giesl, Marc Brockschmidt, Florian Frohn, Carsten Fuhs, Jera Hensel, and Peter Schneider-Kamp: Automated Termination Analysis for Programs with Pointer Arithmetic
2014-06 Esther Horbert, Germán Martín García, Simone Frintrop, and Bastian Leibe: Sequence Level Salient Object Proposals for Generic Object Detection in Video
2014-07 Niloofar Safiran, Johannes Lotz, and Uwe Naumann: Algorithmic Differentiation of Numerical Methods: Second-Order Tangent and Adjoint Solvers for Systems of Parametrized Nonlinear Equations
2014-08 Christina Jansen, Florian Göbe, and Thomas Noll: Generating Inductive Predicates for Symbolic Execution of Pointer-Manipulating Programs
2014-09 Thomas Ströder and Terrance Swift (Editors): Proceedings of the International Joint Workshop on Implementation of Constraint and Logic Programming Systems and Logic-based Methods in Programming Environments 2014

2014-14 Florian Schmidt, Matteo Ceriotti, Niklas Hauser, and Klaus Wehrle: HotBox: Testing Temperature Effects in Sensor Networks

2014-15 Dominique Gückel: Synthesis of State Space Generators for Model Checking Microcontroller Code

2014-16 Hongfei Fu: Verifying Probabilistic Systems: New Algorithms and Complexity Results

* These reports are only available as a printed version.

Please contact `biblio@informatik.rwth-aachen.de` to obtain copies.