RWTH Aachen

Department of Computer Science
Technical Report

Complexity Analysis for Term Rewriting by Integer Transition Systems

M. Naaf, F. Frohn, M. Brockschmidt, C. Fuhs, and J. Giesl

ISSN 0935–3232 · Aachener Informatik-Berichte · AIB-2017-05

RWTH Aachen · Department of Computer Science · June 2017

Complexity Analysis for Term Rewriting by Integer Transition Systems*

M. Naaf[1], F. Frohn[1], M. Brockschmidt[2], C. Fuhs[3], and J. Giesl[1]

[1] LuFG Informatik 2, RWTH Aachen University, Germany
[2] Microsoft Research, Cambridge, UK
[3] Birkbeck, University of London, UK

Abstract. We present a new method to infer upper bounds on the innermost runtime complexity of term rewrite systems (TRSs), which benefits from recent advances on complexity analysis of integer transition systems (ITSs). To this end, we develop a transformation from TRSs to a generalized notion of ITSs with (possibly non-tail) recursion. To analyze their complexity, we introduce a modular technique which allows us to use existing tools for standard ITSs in order to infer complexity bounds for our generalized ITSs. The key idea of our technique is a summarization method that allows us to analyze components of the transition system independently. We implemented our contributions in the tool AProVE, and our experiments show that one can now infer bounds for significantly more TRSs than with previous state-of-the-art tools for term rewriting.

1 Introduction

There are many techniques for automatic complexity analysis of programs with integer (or natural) numbers, e.g., [1,2,4,11,13,14,16–18,23,26–28,34]. On the other hand, several techniques analyze complexity of *term rewrite systems* (TRSs), e.g., [7, 8, 12, 19, 20, 24, 29, 32, 36]. TRSs are a classical model for equational reasoning and evaluation with user-defined data structures and recursion [9].

Although the approaches for complexity analysis of term rewriting support modularity, they usually cannot completely remove rules from the TRS after having analyzed them. In contrast, approaches for integer programs may regard small program parts independently and combine the results for these parts to obtain a result for the overall program. In this work, we show how to obtain such a form of modularity also for complexity analysis of TRSs.

After recapitulating TRSs and their complexity in Sect. 2, in Sect. 3 we introduce a transformation from TRSs into a variant of integer transition systems (ITSs) called *recursive natural transition systems* (RNTSs). In contrast to standard ITSs, RNTSs allow arbitrary recursion, and the variables only range over the natural numbers. We show that the innermost runtime complexity of the original TRS is bounded by the complexity of the resulting RNTS, i.e., one can now use any complexity tool for RNTSs to infer complexity bounds for TRSs.

* Supported by DFG grant GI 274/6-1 and the Air Force Research Laboratory (AFRL).

Unfortunately, many existing techniques and tools for standard ITSs do not support the non-tail recursive calls that can occur in RNTSs. Therefore, in Sect. 4 we develop an approach to infer complexity bounds for RNTSs which can use arbitrary complexity tools for standard ITSs as a back-end. The approach from Sect. 4 is completely modular, as it repeatedly finds bounds for parts of the RNTS and combines them. In this way, our technique benefits from all advances of any ITS tools, irrespective of whether they support non-tail recursion (e.g., CoFloCo [16,17]) or not (e.g., KoAT [13]). As demonstrated by our implementation in AProVE [22], our contributions allow us to derive complexity bounds for many TRSs where state-of-the-art tools fail, cf. Sect. 5. App. A presents improvements to increase the precision when abstracting TRSs to RNTSs. All proofs can be found in App. B.

2 Complexity of Term Rewriting

We assume basic knowledge of term rewriting [9] and recapitulate innermost (relative) term rewriting and its runtime complexity.

Definition 1 (Term Rewriting [8, 9]). *We denote the set of* terms *over a finite signature* Σ *and the variables* $\mathcal{V}$ *by* $\mathcal{T}(\Sigma, \mathcal{V})$*. The* size $|t|$ *of a term* t *is defined as* $|x| = 1$ *if* $x \in \mathcal{V}$ *and* $|f(t_1, \dots, t_k)| = 1 + \sum_{i=1}^{k} |t_i|$*. A* TRS $\mathcal{R}$ *is a set of rules* $\{\ell_1 \to r_1, \dots, \ell_n \to r_n\}$ *with* $\ell_i, r_i \in \mathcal{T}(\Sigma, \mathcal{V})$*,* $\ell_i \notin \mathcal{V}$*, and* $\mathcal{V}(r_i) \subseteq \mathcal{V}(\ell_i)$ *for all* $1 \leq i \leq n$*. The* rewrite relation *is defined as* $s \to_{\mathcal{R}} t$ *iff there is a rule* $\ell \to r \in \mathcal{R}$*, a position* $\pi \in \mathcal{P}os(s)$*, and a substitution* σ *such that* $s|_\pi = \ell\sigma$ *and* $t = s[r\sigma]_\pi$*. Here,* $\ell\sigma$ *is called the* redex *of the rewrite step.*

For two TRSs $\mathcal{R}$ *and* $\mathcal{S}$*,* $\mathcal{R}/\mathcal{S}$ *is a* relative TRS*, and its* rewrite relation $\to_{\mathcal{R}/\mathcal{S}}$ *is* $\to_{\mathcal{S}}^{*} \circ \to_{\mathcal{R}} \circ \to_{\mathcal{S}}^{*}$*, i.e., it allows rewriting with* $\mathcal{S}$ *before and after each* $\mathcal{R}$*-step. We define the* innermost *rewrite relation as* $s \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}} t$ *iff* $s \to_{\mathcal{S}}^{*} s' \to_{\mathcal{R}} s'' \to_{\mathcal{S}}^{*} t$ *for some terms* s', s''*, where the proper subterms of the redexes of each step with* $\to_{\mathcal{S}}$ *or* $\to_{\mathcal{R}}$ *are in normal form w.r.t.* $\mathcal{R} \cup \mathcal{S}$*. We write* $\xrightarrow{\mathrm{i}}_{\mathcal{R}}$ *instead of* $\xrightarrow{\mathrm{i}}_{\mathcal{R}/\varnothing}$*.*

$\Sigma_d^{\mathcal{R}\cup\mathcal{S}} = \{\mathrm{root}(\ell) \mid \ell \to r \in \mathcal{R}\cup\mathcal{S}\}$ *and* $\Sigma_c^{\mathcal{R}\cup\mathcal{S}} = \Sigma \setminus \Sigma_d^{\mathcal{R}\cup\mathcal{S}}$ *are the* defined *(resp.* constructor*) symbols of* $\mathcal{R}/\mathcal{S}$*. A term* $f(t_1, \dots, t_k)$ *is* basic *iff* $f \in \Sigma_d^{\mathcal{R}\cup\mathcal{S}}$ *and* $t_1, \dots, t_k \in \mathcal{T}(\Sigma_c^{\mathcal{R}\cup\mathcal{S}}, \mathcal{V})$*.* $\mathcal{R}/\mathcal{S}$ *is a* constructor system *iff* ℓ *is basic for all* $\ell \to r \in \mathcal{R}\cup\mathcal{S}$*.*

In this paper, we will restrict ourselves to the analysis of constructor systems.

Example 2. The following rules implement the insertion sort *algorithm.*

$$
\begin{array}{rlr}
\mathsf{isort}(\mathsf{nil}, ys) \to ys & & (1)\\
\mathsf{isort}(\mathsf{cons}(x, xs), ys) \to \mathsf{isort}(xs, \mathsf{ins}(x, ys)) & & (2)\\
\mathsf{ins}(x, \mathsf{nil}) \to \mathsf{cons}(x, \mathsf{nil}) & & (3)\\
\mathsf{ins}(x, \mathsf{cons}(y, ys)) \to \mathsf{if}(\mathsf{gt}(x, y), x, \mathsf{cons}(y, ys)) & & (4)\\
\mathsf{if}(\mathsf{true}, x, \mathsf{cons}(y, ys)) \to \mathsf{cons}(y, \mathsf{ins}(x, ys)) & & (5)\\
\mathsf{if}(\mathsf{false}, x, \mathsf{cons}(y, ys)) \to \mathsf{cons}(x, \mathsf{cons}(y, ys)) & & (6)
\end{array}
\qquad
\begin{array}{rr}
\mathsf{gt}(0, y) \to \mathsf{false} & (7)\\
\mathsf{gt}(\mathsf{s}(x), 0) \to \mathsf{true} & (8)\\
\mathsf{gt}(\mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{gt}(x, y) & (9)
\end{array}
$$

Relative rules are useful to model built-in operations in programming languages since applications of these rules are disregarded for the complexity of a

TRS. For example, the translation from RAML *programs [27] to term rewriting in [8] uses relative rules to model the semantics of comparisons and similar operations on* RAML*'s primitive data types. Thus, we decompose the rules above into a relative TRS $\mathcal{R}/\mathcal{S}$ with $\mathcal{R} = \{(1), \ldots, (6)\}$ and $\mathcal{S} = \{(7), (8), (9)\}$.*[4]

In our example, we have $\Sigma_d^{\mathcal{R}\cup\mathcal{S}} = \{\mathsf{isort}, \mathsf{ins}, \mathsf{if}, \mathsf{gt}\}$ and $\Sigma_c^{\mathcal{R}\cup\mathcal{S}} = \{\mathsf{cons}, \mathsf{nil}, \mathsf{s}, \mathsf{0}, \mathsf{true}, \mathsf{false}\}$. Since all left-hand sides are basic, $\mathcal{R}/\mathcal{S}$ is a constructor system. An example rewrite sequence to sort the list $[2, 0]$ is

$$\begin{array}{lll} t = \mathsf{isort}(\mathsf{cons}(\mathsf{s}(\mathsf{s}(\mathsf{0})), \mathsf{cons}(\mathsf{0}, \mathsf{nil})), \mathsf{nil}) & \xrightarrow{\mathsf{i}}_{\mathcal{R}} \mathsf{isort}(\mathsf{cons}(\mathsf{0}, \mathsf{nil}), \mathsf{ins}(\mathsf{s}(\mathsf{s}(\mathsf{0})), \mathsf{nil})) & \xrightarrow{\mathsf{i}}_{\mathcal{R}} \\ \quad \mathsf{isort}(\mathsf{cons}(\mathsf{0}, \mathsf{nil}), \mathsf{cons}(\mathsf{s}(\mathsf{s}(\mathsf{0})), \mathsf{nil})) & \xrightarrow{\mathsf{i}}_{\mathcal{R}} \mathsf{isort}(\mathsf{nil}, \mathsf{ins}(\mathsf{0}, \mathsf{cons}(\mathsf{s}(\mathsf{s}(\mathsf{0})), \mathsf{nil}))) & \xrightarrow{\mathsf{i}}_{\mathcal{R}} \\ \quad \mathsf{isort}(\mathsf{nil}, \mathsf{if}(\mathsf{gt}(\mathsf{0}, \mathsf{s}(\mathsf{s}(\mathsf{0}))), \ldots, \ldots)) & \xrightarrow{\mathsf{i}}_{\mathcal{S}} \mathsf{isort}(\mathsf{nil}, \mathsf{if}(\mathsf{false}, \ldots, \ldots)) & \xrightarrow{\mathsf{i}}_{\mathcal{R}} \\ \quad \mathsf{isort}(\mathsf{nil}, \mathsf{cons}(\mathsf{0}, \mathsf{cons}(\mathsf{s}(\mathsf{s}(\mathsf{0})), \mathsf{nil}))) & \xrightarrow{\mathsf{i}}_{\mathcal{R}} \mathsf{cons}(\mathsf{0}, \mathsf{cons}(\mathsf{s}(\mathsf{s}(\mathsf{0})), \mathsf{nil})) & \end{array}$$

Note that ordinary TRSs are a special case of relative TRSs (where $\mathcal{S} = \varnothing$). We usually just write "TRSs" to denote "relative TRSs". We now define the *runtime complexity* of a TRS $\mathcal{R}/\mathcal{S}$. In Def. 3, ω is the smallest infinite ordinal, i.e., $\omega > e$ holds for all $e \in \mathbb{N}$, and for any $M \subseteq \mathbb{N} \cup \{\omega\}$, $\sup M$ is the least upper bound of M, where $\sup \varnothing = 0$.

Definition 3 (Innermost Runtime Complexity [24, 25, 32, 36]). *The* derivation height *of a term t w.r.t. a relation $\to$ is the length of the longest sequence of $\to$-steps starting with t, i.e., $\mathrm{dh}(t, \to) = \sup\{e \mid \exists t' \in \mathcal{T}(\Sigma, \mathcal{V}).\ t \to^e t'\}$. If t starts an infinite $\to$-sequence, this yields $\mathrm{dh}(t, \to) = \omega$. The* innermost runtime complexity *function $\mathrm{irc}_{\mathcal{R}/\mathcal{S}}$ maps any $n \in \mathbb{N}$ to the length of the longest sequence of $\xrightarrow{\mathsf{i}}_{\mathcal{R}/\mathcal{S}}$-steps starting with a basic term whose size is at most n, i.e., $\mathrm{irc}_{\mathcal{R}/\mathcal{S}}(n) = \sup\{\mathrm{dh}(t, \xrightarrow{\mathsf{i}}_{\mathcal{R}/\mathcal{S}}) \mid t \text{ is basic}, |t| \leq n\}$.*

Example 4. The rewrite sequence for t in Ex. 2 is maximal, and thus, $\mathrm{dh}(t, \xrightarrow{\mathsf{i}}_{\mathcal{R}/\mathcal{S}}) = 6$. So the $\xrightarrow{\mathsf{i}}_{\mathcal{S}}$-step does not contribute to t's derivation height. As $|t| = 9$, this implies $\mathrm{irc}_{\mathcal{R}/\mathcal{S}}(9) \geq 6$. We will show how our new approach proves $\mathrm{irc}_{\mathcal{R}/\mathcal{S}}(n) \in \mathcal{O}(n^2)$ automatically.

3 From TRSs to Recursive Natural Transition Systems

We now reduce complexity analysis of TRSs to complexity analysis of *recursive natural transition systems* (RNTSs). In contrast to term rewriting, RNTSs offer built-in support for arithmetic, but disallow pattern matching. To analyze TRSs, it suffices to regard RNTSs where all variables range over $\mathbb{N}$. We use the signature $\Sigma_{\mathsf{exp}} = \{+, \cdot\} \cup \mathbb{N}$ for arithmetic expressions and $\Sigma_{\mathsf{fml}} = \Sigma_{\mathsf{exp}} \cup \{\mathsf{true}, \mathsf{false}, <, \wedge\}$ for arithmetic formulas ("constraints"). We will also use relations like $=$ in constraints, but these are just syntactic sugar. To extend the rewrite relation with semantics for these symbols, let $[\![.]\!]$ evaluate all arithmetic and Boolean

[4] In this way, the complexity of gt is 0, whereas comparisons have complexity 1 with the slightly more complicated encoding from [8]. Since this difference does not affect the asymptotic complexity of Ex. 2, we use the simpler encoding for the sake of readability.

expressions in a term. So for example, $[\![\mathsf{gt}(1+2, 5+y)]\!] = \mathsf{gt}(3, 5+y)$ and $[\![3 > 5 \wedge \mathsf{true}]\!] = \mathsf{false}$. We allow substitutions with infinite domains and call σ a *natural substitution* iff $\sigma(x) \in \mathbb{N}$ for all $x \in \mathcal{V}$.

Definition 5 (Recursive Natural Transition System). *An* RNTS *over a finite signature Σ with $\Sigma \cap \Sigma_{\mathsf{fml}} = \varnothing$ is a set of rules $\mathcal{P} = \{\ell_1 \xrightarrow{w_1} r_1\ [\varphi_1], \ldots, \ell_n \xrightarrow{w_n} r_n\ [\varphi_n]\}$ with $\ell_i = f(x_1, \ldots, x_k)$ for $f \in \Sigma$ and pairwise different variables $x_1, \ldots, x_k$, $r_i \in \mathcal{T}(\Sigma \uplus \Sigma_{\mathsf{exp}}, \mathcal{V})$, constraints $\varphi_i \in \mathcal{T}(\Sigma_{\mathsf{fml}}, \mathcal{V})$, and weights $w_i \in \mathcal{T}(\Sigma_{\mathsf{exp}}, \mathcal{V})$. An RNTS $\mathcal{P}$ induces a rewrite relation $\xrightarrow{m}_{\mathcal{P}}$ on ground terms from $\mathcal{T}(\Sigma \uplus \Sigma_{\mathsf{exp}}, \varnothing)$, where $s \xrightarrow{m}_{\mathcal{P}} t$ iff there are $\ell \xrightarrow{w} r\ [\varphi] \in \mathcal{P}$, $\pi \in \mathcal{P}os(s)$, and a natural substitution σ such that $s|_\pi = \ell\sigma$, $[\![\varphi\sigma]\!] = \mathsf{true}$, $m = [\![w\sigma]\!] \in \mathbb{N}$, and $t = [\![s[r\sigma]_\pi]\!]$. We sometimes just write $s \to_{\mathcal{P}} t$ instead of $s \xrightarrow{m}_{\mathcal{P}} t$. Again, let $\Sigma_d^{\mathcal{P}} = \{\mathrm{root}(\ell) \mid \ell \xrightarrow{w} r\, [\varphi] \in \mathcal{P}\}$ and $\Sigma_c^{\mathcal{P}} = \Sigma \setminus \Sigma_d^{\mathcal{P}}$.*

A term $f(n_1, \ldots, n_k)$ with $f \in \Sigma$ and $n_1, \ldots, n_k \in \mathbb{N}$ is nat-basic*, and its* size *is $\|f(n_1, \ldots, n_k)\| = 1 + n_1 + \ldots + n_k$. To consider weights for derivation heights, we define $\mathrm{dhw}(t, \to_{\mathcal{P}})$ to be the maximum weight of any $\to_{\mathcal{P}}$-sequence starting with t, i.e., $\mathrm{dhw}(t_0, \to_{\mathcal{P}}) = \sup\{\sum_{i=1}^{e} m_i \mid \exists t_1, \ldots, t_e \in \mathcal{T}(\Sigma \uplus \Sigma_{\mathsf{exp}}, \varnothing).\ t_0 \xrightarrow{m_1}_{\mathcal{P}} \ldots \xrightarrow{m_e}_{\mathcal{P}} t_e\}$. Then $\mathrm{irc}_{\mathcal{P}}$ maps $n \in \mathbb{N}$ to the maximum weight of any $\to_{\mathcal{P}}$-sequence starting with a nat-basic term whose size is at most n, i.e., $\mathrm{irc}_{\mathcal{P}}(n) = \sup\{\mathrm{dhw}(t, \to_{\mathcal{P}}) \mid t$ is nat-basic, $\|t\| \leq n\}$.*

Note that the rewrite relation for RNTSs is "innermost" by construction, as rules do not contain symbols from Σ below the root in left-hand sides, and they are only applicable if all variables are instantiated by numbers.

The crucial idea of our approach is to model the behavior of a TRS by a corresponding RNTS which results from abstracting constructor terms to their size. Thus, we use the following transformation $\wr \cdot \int$ from TRSs to RNTSs.

Definition 6 (Abstraction $\wr \cdot \int$ from TRSs to RNTSs). *For a TRS $\mathcal{R}/\mathcal{S}$, the* size abstraction *$\wr t \int$ of a term $t \in \mathcal{T}(\Sigma, \mathcal{V})$ is defined as follows:*

$$\begin{aligned} \wr x \int &= x && \text{for } x \in \mathcal{V} \\ \wr f(t_1, \ldots, t_k) \int &= 1 + \wr t_1 \int + \ldots + \wr t_k \int && \text{if } f \in \Sigma_c^{\mathcal{R} \cup \mathcal{S}} \\ \wr f(t_1, \ldots, t_k) \int &= f(\wr t_1 \int, \ldots, \wr t_k \int) && \text{if } f \in \Sigma_d^{\mathcal{R} \cup \mathcal{S}} \end{aligned}$$

We lift $\wr \cdot \int$ to rules with basic left-hand sides. For $\ell = f(t_1, \ldots, t_k)$ with $t_1, \ldots, t_k \in \mathcal{T}(\Sigma_c^{\mathcal{R} \cup \mathcal{S}}, \mathcal{V})$ and $w \in \mathcal{T}(\Sigma_{\mathsf{exp}}, \mathcal{V})$, we define

$$\wr \ell \to r \int_w \quad = \quad f(x_1, \ldots, x_k) \xrightarrow{w} \wr r \int \quad \left[\bigwedge\nolimits_{i=1}^{k} x_i = \wr t_i \int \ \wedge \ \bigwedge\nolimits_{x \in \mathcal{V}(\ell)} x \geq 1 \right]$$

for pairwise different fresh variables $x_1, \ldots, x_k$. For a constructor system $\mathcal{R}/\mathcal{S}$, we define the RNTS $\wr \mathcal{R}/\mathcal{S} \int = \{\wr \ell \to r \int_1 \mid \ell \to r \in \mathcal{R}\} \cup \{\wr \ell \to r \int_0 \mid \ell \to r \in \mathcal{S}\}$.

Example 7. For the TRS $\mathcal{R}/\mathcal{S}$ from Ex. 2, $\wr \mathcal{R}/\mathcal{S} \int$ corresponds to the following RNTS.

$$
\begin{array}{rll}
\mathsf{isort}(xs, ys) \xrightarrow{1} ys & [xs \;= 1 \wedge \ldots] & (1') \\
\mathsf{isort}(xs', ys) \xrightarrow{1} \mathsf{isort}(xs, \mathsf{ins}(x, ys)) & [xs' = 1 + x + xs \wedge \ldots] & (2') \\
\mathsf{ins}(x, ys) \xrightarrow{1} 2 + x & [ys \;= 1 \wedge \ldots] & (3') \\
\mathsf{ins}(x, ys') \xrightarrow{1} \mathsf{if}(\mathsf{gt}(x, y), x, ys') & [ys' = 1 + y + ys \wedge \ldots] & (4') \\
\mathsf{if}(b, x, ys') \xrightarrow{1} 1 + y + \mathsf{ins}(x, ys) & [b = 1 \wedge ys' = 1 + y + ys \wedge \ldots] & (5') \\
\mathsf{if}(b, x, ys') \xrightarrow{1} 1 + x + ys' & [b = 1 \wedge ys' = 1 + y + ys \wedge \ldots] & (6') \\
\mathsf{gt}(x, y) \xrightarrow{0} 1 & [x \;= 1 \wedge \ldots] & (7') \\
\mathsf{gt}(x', y) \xrightarrow{0} 1 & [x' = 1 + x \wedge y = 1 \wedge \ldots] & (8') \\
\mathsf{gt}(x', y') \xrightarrow{0} \mathsf{gt}(x, y) & [x' = 1 + x \wedge y' = 1 + y \wedge \ldots] & (9')
\end{array}
$$

In these rules, "$\wedge \ldots$" stands for the constraint that all variables have to be instantiated with values ≥ 1. Note that we make use of fresh variables like x and xs on the right-hand side of $(2')$ to simulate matching of constructor terms. Using this RNTS, the rewrite steps in Ex. 2 can be simulated as follows.

$$
\begin{array}{rlll}
t' & = \mathsf{isort}(7,1) & \xrightarrow{1} \mathsf{isort}(3, \mathsf{ins}(3,1)) & \xrightarrow{1} \mathsf{isort}(3,5) \\
& \xrightarrow{1} \mathsf{isort}(1, \mathsf{ins}(1,5)) & \xrightarrow{1} \mathsf{isort}(1, \mathsf{if}(\mathsf{gt}(1,3),1,5)) & \xrightarrow{0} \mathsf{isort}(1, \mathsf{if}(1,1,5)) \\
& \xrightarrow{1} \mathsf{isort}(1,7) & \xrightarrow{1} 7 &
\end{array}
$$

For the nat-basic term t', we have $\|t'\| = 1 + 7 + 1 = 9$. So the above sequence proves $\mathrm{dhw}(t', \to_{\mathcal{P}}) \geq 6$ *and hence,* $\mathrm{irc}_{\mathcal{P}}(9) \geq 6$. *Note that unlike Ex. 2, here rewriting nat-basic terms is non-deterministic as, e.g., we also have* $\mathsf{isort}(7,1) \xrightarrow{1} \mathsf{isort}(2, \mathsf{ins}(4,1))$. *The reason is that $\wr \cdot \wr$ is a* blind abstraction *[10], which abstracts several different terms to the same number.*

$[\![\wr \cdot \wr]\!]$ maps basic ground terms to nat-basic terms, e.g., $[\![\wr \mathsf{ins}(\mathsf{s}(\mathsf{0}), \mathsf{nil}) \wr]\!] = [\![\mathsf{ins}(1{+}1, 1)]\!] = \mathsf{ins}(2,1)$. We now show that under certain conditions, $\mathrm{dh}(t, \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}}) \leq \mathrm{dhw}([\![\wr t \wr]\!], \to_{\wr \mathcal{R}/\mathcal{S} \wr})$ holds for all ground terms t, i.e., rewrite sequences of a TRS $\mathcal{R}/\mathcal{S}$ can be simulated in the RNTS $\wr \mathcal{R}/\mathcal{S} \wr$ resulting from its transformation. We would like to conclude that in these cases, we also have $\mathrm{irc}_{\mathcal{R}/\mathcal{S}}(n) \leq \mathrm{irc}_{\wr \mathcal{R}/\mathcal{S} \wr}(n)$. However, irc considers arbitrary (basic) terms, but the above connection between the derivation heights of t and $[\![\wr t \wr]\!]$ only holds for *ground* terms t. For *full* rewriting, we clearly have $\mathrm{dh}(t, \to_{\mathcal{R}}) \leq \mathrm{dh}(t\sigma, \to_{\mathcal{R}})$ for any substitution σ. However, this does not hold for *innermost* rewriting. For example, $\mathsf{f}(\mathsf{g}(x))$ has an infinite innermost reduction with the TRS $\{\mathsf{f}(\mathsf{g}(x)) \to \mathsf{f}(\mathsf{g}(x)), \mathsf{g}(\mathsf{a}) \to \mathsf{a}\}$, but $\mathsf{f}(\mathsf{g}(\mathsf{a}))$ is innermost terminating. Nevertheless, we show in Thm. 9 that for *constructor systems* $\mathcal{R}$, $\mathrm{dh}(t, \xrightarrow{\mathrm{i}}_{\mathcal{R}}) \leq \mathrm{dh}(t\sigma, \xrightarrow{\mathrm{i}}_{\mathcal{R}})$ holds for any ground substitution σ.

However, for *relative* rewriting with constructor systems $\mathcal{R}$ and $\mathcal{S}$, $\mathrm{dh}(t, \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}}) \leq \mathrm{dh}(t\sigma, \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}})$ does not necessarily hold if $\mathcal{S}$ is not innermost terminating. To see this, consider $\mathcal{R} = \{\mathsf{f}(x) \to \mathsf{f}(x)\}$ and $\mathcal{S} = \{\mathsf{g}(\mathsf{a}) \to \mathsf{g}(\mathsf{a})\}$. Now $\mathsf{f}(\mathsf{g}(x))$ has an infinite reduction w.r.t. $\xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}}$ since $\mathsf{g}(x)$ is a normal form w.r.t. $\mathcal{R} \cup \mathcal{S}$. However, its instance $\mathsf{f}(\mathsf{g}(\mathsf{a}))$ has the derivation height 0 w.r.t. $\xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}}$, as $\mathsf{g}(\mathsf{a})$ is not innermost terminating w.r.t. $\mathcal{S}$ and no rule of $\mathcal{R}$ can ever be applied. To solve this problem, we extend the TRS $\mathcal{S}$ by a *terminating variant* $\mathcal{N}$.

Definition 8 (Terminating Variant). *A TRS $\mathcal{N}$ is a* terminating variant *of $\mathcal{S}$ iff $\xrightarrow{\mathrm{i}}_{\mathcal{N}}$ terminates and every $\mathcal{N}$-normal form is also an $\mathcal{S}$-normal form.*

So if one can prove innermost termination of $\mathcal{S}$, then one can use $\mathcal{S}$ as a terminating variant of itself. For instance in Ex. 2, termination of $\mathcal{S} = \{(7), (8), (9)\}$ can easily be shown automatically by standard tools like AProVE [22]. Otherwise, one can for instance use a terminating variant $\{f(x_1, \ldots, x_k) \to t_f \mid f \in \Sigma_d^{\mathcal{S}}\}$ where for each f, we pick some constructor ground term $t_f \in \mathcal{T}(\Sigma_c^{\mathcal{R}\cup\mathcal{S}}, \varnothing)$. Now one can prove that for innermost (relative) rewriting, the derivation height of a term does not decrease when it is instantiated by a ground substitution.

Theorem 9 (Soundness of Instantiation and Terminating Variants). *Let $\mathcal{R}$, $\mathcal{S}$ be constructor systems and $\mathcal{N}$ be a terminating variant of $\mathcal{S}$. Then $\mathrm{dh}(t, \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}}) \leq \mathrm{dh}(t\sigma, \xrightarrow{\mathrm{i}}_{\mathcal{R}/(\mathcal{S}\cup\mathcal{N})})$ holds for any term t where $t\sigma$ is ground.*

However, the restriction to ground terms t still does not ensure $\mathrm{dh}(t, \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}}) \leq \mathrm{dhw}([\![\wr t \wr]\!], \to_{\wr\mathcal{R}/\mathcal{S}\wr})$. The problem is that $\xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}}$ can rewrite a term t at position π also if there is a defined symbol below $t|_\pi$ as long as no rule can be applied to that subterm. So for Ex. 2, we have $\mathsf{isort}(\mathsf{nil}, \mathsf{if}(\mathsf{true}, \mathsf{0}, \mathsf{nil})) \xrightarrow{\mathrm{i}}_{\mathcal{R}} \mathsf{if}(\mathsf{true}, \mathsf{0}, \mathsf{nil})$, but $\wr\mathcal{R}/\mathcal{S}\wr$ cannot rewrite $[\![\wr \mathsf{isort}(\mathsf{nil}, \mathsf{if}(\mathsf{true}, \mathsf{0}, \mathsf{nil}))\wr]\!] = \mathsf{isort}(1, \mathsf{if}(1, 1, 1))$ since the if-rules of $\wr\mathcal{R}/\mathcal{S}\wr$ may be applied only if the third argument is ≥ 3, and the variables in the isort-rule may be instantiated only by numbers (not by normal forms like $\mathsf{if}(1, 1, 1)$). This problem can be solved by requiring that $\mathcal{R}/\mathcal{S}$ is *completely defined*, i.e., that $\mathcal{R} \cup \mathcal{S}$ can rewrite every basic ground term. However, this is too restrictive as we, e.g., would like $\mathsf{gt}(\mathsf{true}, \mathsf{false})$ to be in normal form. Fortunately, (innermost) runtime complexity is *persistent* w.r.t. type introduction [6]. Thus, we only need to ensure that every *well-typed* basic ground term can be rewritten.

Definition 10 (Typed TRSs (cf. e.g. [21, 37])). *In a many-sorted (first-order monomorphic) signature Σ over the set of types Ty, every symbol $f \in \Sigma$ has a type of the form $\tau_1 \times \ldots \times \tau_k \to \tau$ with $\tau_1, \ldots, \tau_k, \tau \in Ty$. Moreover, every variable has a type from Ty, and we assume that $\mathcal{V}$ contains infinitely many variables of every type in Ty. We call $t \in \mathcal{T}(\Sigma, \mathcal{V})$ a* well-typed *term of type τ iff either $t \in \mathcal{V}$ is a variable of type τ or $t = f(t_1, \ldots, t_k)$ where f has the type $\tau_1 \times \ldots \times \tau_k \to \tau$ and each t_i is a well-typed term of type τ_i.*

A rewrite rule $\ell \to r$ is well typed iff ℓ and r are well-typed terms of the same type. A TRS $\mathcal{R}/\mathcal{S}$ is well typed iff all rules of $\mathcal{R} \cup \mathcal{S}$ are well typed. (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.)

Example 11. For any TRS $\mathcal{R}/\mathcal{S}$, standard algorithms can compute a type assignment to make $\mathcal{R}/\mathcal{S}$ well typed (and to decompose the terms into as many types as possible). For the TRS from Ex. 2 we obtain the following type assignment. Note that for this type assignment the TRS is not completely defined *since* $\mathsf{if}(\mathsf{true}, \mathsf{0}, \mathsf{nil})$ *is a well-typed basic ground term in normal form w.r.t. $\mathcal{R} \cup \mathcal{S}$.*

$\mathsf{isort} :: \mathbf{List} \times \mathbf{List} \to \mathbf{List}$	$\mathsf{0} :: \mathbf{Nat}$	$\mathsf{gt} :: \mathbf{Nat} \times \mathbf{Nat} \to \mathbf{Bool}$
$\mathsf{ins} :: \mathbf{Nat} \times \mathbf{List} \to \mathbf{List}$	$\mathsf{s} :: \mathbf{Nat} \to \mathbf{Nat}$	$\mathsf{true}, \mathsf{false} :: \mathbf{Bool}$
$\mathsf{if} :: \mathbf{Bool} \times \mathbf{Nat} \times \mathbf{List} \to \mathbf{List}$	$\mathsf{nil} :: \mathbf{List}$	$\mathsf{cons} :: \mathbf{Nat} \times \mathbf{List} \to \mathbf{List}$

Definition 12 (Completely Defined). *A well-typed TRS $\mathcal{R}/\mathcal{S}$ over a many-sorted signature with types Ty is* completely defined *iff there is at least one constant for each $\tau \in Ty$ and no well-typed basic ground term in $\mathcal{R} \cup \mathcal{S}$-normal form.*

For completely defined TRSs, the transformation from TRSs to RNTSs is sound.

Theorem 13 (Soundness of Abstraction $\wr\cdot\wr$). *Let $\mathcal{R}/\mathcal{S}$ be a well-typed, completely defined constructor system. Then $\mathrm{dh}(t, \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}}) \leq \mathrm{dhw}([\![\wr t \wr]\!], \to_{\wr\mathcal{R}/\mathcal{S}\wr})$ holds for all well-typed ground terms t. Let $\mathcal{N}$ be a terminating variant of $\mathcal{S}$ such that $\mathcal{R}/(\mathcal{S} \cup \mathcal{N})$ is also well typed. If $\mathcal{R}/(\mathcal{S} \cup \mathcal{N})$ is completely defined, then we have $\mathrm{irc}_{\mathcal{R}/\mathcal{S}}(n) \leq \mathrm{irc}_{\wr\mathcal{R}/(\mathcal{S}\cup\mathcal{N})\wr}(n)$ for all $n \in \mathbb{N}$.*

As every TRS $\mathcal{R}/\mathcal{S}$ is well typed w.r.t. *some* type assignment (e.g., the one with just a single type), the only additional restriction in Thm. 13 is that the TRS has to be completely defined. This can always be achieved by extending $\mathcal{S}$ by a suitable terminating variant $\mathcal{N}$ of $\mathcal{S}$ automatically. Based on standard algorithms to detect well-typed basic ground terms $f(\ldots)$ in $(\mathcal{R} \cup \mathcal{S})$-normal form [30, 31], we add the rules $f(x_1, \ldots, x_k) \to t_f$ to $\mathcal{N}$, where again for each f, we choose some constructor ground term $t_f \in \mathcal{T}(\Sigma_c^{\mathcal{R}\cup\mathcal{S}}, \varnothing)$. As shown by Thm. 9, we have $\mathrm{dh}(t, \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}}) \leq \mathrm{dh}(t\sigma, \xrightarrow{\mathrm{i}}_{\mathcal{R}/(\mathcal{S}\cup\mathcal{N})})$ for *any* terminating variant $\mathcal{N}$, i.e., adding such rules never decreases the derivation height. So even if $\mathcal{R}/\mathcal{S}$ is not completely defined and just $\mathcal{R}/(\mathcal{S} \cup \mathcal{N})$ is completely defined, we still have $\mathrm{irc}_{\mathcal{R}/\mathcal{S}}(n) \leq \mathrm{irc}_{\mathcal{R}/(\mathcal{S}\cup\mathcal{N})}(n) \leq \mathrm{irc}_{\wr\mathcal{R}/(\mathcal{S}\cup\mathcal{N})\wr}(n)$.

Example 14. *To make the TRS of Ex. 2 completely defined, we add rules for all defined symbols in basic ground normal forms. In this example, the only such symbol is* if. *Hence, for instance we add* $\mathsf{if}(b, x, xs) \to \mathsf{nil}$ *to* $\mathcal{S}$. *The resulting TRS* $\mathcal{S} \cup \{\mathsf{if}(b, x, xs) \to \mathsf{nil}\}$ *is clearly a terminating variant of* $\mathcal{S}$. *Hence, to analyze complexity of the insertion sort TRS, we now extend the RNTS of Ex. 7 by*

$$\wr \mathsf{if}(b, x, xs) \to \mathsf{nil} \wr_0 \quad = \quad \mathsf{if}(b, x, xs) \xrightarrow{0} 1 \;\; [b \geq 1 \land x \geq 1 \land xs \geq 1] \qquad (10)$$

4 Analyzing the Complexity of RNTSs

Thm. 13 allows us to reduce complexity analysis of term rewriting to the analysis of RNTSs. Our RNTSs are related to *integer transition systems* (ITSs), a formalism often used to abstract programs. The main difference is that RNTSs can model procedure calls by nested function symbols $f(\ldots g(\ldots) \ldots)$ on the right-hand side of rules, whereas ITSs may allow right-hand sides like $f(\ldots) + g(\ldots)$, but no nesting of $f, g \in \Sigma$. So ITSs cannot pass the result of one function as a parameter to another function. Note that in contrast to the usual definition of ITSs, in our setting reductions can begin with any (nat-basic) terms instead of dedicated start terms, and it suffices to regard natural instead of integer numbers. (An extension to recursive transition systems on integers would be possible by measuring the size of integers by their absolute value, as in [13].)

Definition 15 (ITS). *An RNTS $\mathcal{P}$ over the signature Σ is an* ITS *iff symbols from Σ occur only at parallel positions in right-hand sides of $\mathcal{P}$. Here, π and π' are* parallel *iff π is not a prefix of π' and π' is not a prefix of π.*

Upper runtime complexity bounds for an ITS $\mathcal{P}$ can, for example, be inferred by generating ranking functions which decrease with each application of a rule from $\mathcal{P}$. Then, the ranking functions are multiplied with the weight of the rules.

However, many analysis techniques for ITSs (e.g., [1,4,13,34]) cannot handle the RNTSs generated from standard TRSs. Thus, we now introduce a new modular approach that allows us to apply existing tools for ITSs to analyze RNTSs. Our approach builds upon the idea of alternating between *runtime* and *size* analysis [13]. The key insight is to *summarize* procedures by approximating their runtime and the size of their result, and then to eliminate them from the program. In this way, our analysis decomposes the "call graph" of the RNTS into "blocks" of mutually recursive functions and exports each block of mutually recursive functions into a separate ITS. Thus, in each analysis step it suffices to analyze just an ITS instead of an RNTS. We use weakly monotonic runtime and size bounds from $\mathcal{T}(\Sigma_{\mathsf{exp}}, \mathcal{V})$ to compose them easily when analyzing nested terms.

Definition 16 (Runtime and Size Bounds). *For any terms $t_1, \ldots, t_k$, let $\{x_1/t_1, \ldots, x_k/t_k\}$ be the substitution σ with $x_i\sigma = t_i$ for $1 \leq i \leq k$ and $y\sigma = y$ for $y \in \mathcal{V} \setminus \{x_1, \ldots, x_k\}$. Then $\mathsf{rt} : \Sigma \to \mathcal{T}(\Sigma_{\mathsf{exp}}, \mathcal{V}) \cup \{\omega\}$ is a* runtime bound *for an RNTS $\mathcal{P}$ iff we have $\mathrm{dhw}(f(n_1, \ldots, n_k), \to_{\mathcal{P}}) \leq [\![\mathsf{rt}(f)\,\{x_1/n_1, \ldots, x_k/n_k\}]\!]$ for all $n_1, \ldots, n_k \in \mathbb{N}$ and all $f \in \Sigma$. Similarly, $\mathsf{sz} : \Sigma \to \mathcal{T}(\Sigma_{\mathsf{exp}}, \mathcal{V}) \cup \{\omega\}$ is a* size bound *for $\mathcal{P}$ iff $n \leq [\![\mathsf{sz}(f)\,\{x_1/n_1, \ldots, x_k/n_k\}]\!]$ for all $n_1, \ldots, n_k \in \mathbb{N}$, all $f \in \Sigma$, and all $n \in \mathbb{N}$ with $f(n_1, \ldots, n_k) \to_{\mathcal{P}}^* n$.*

Example 17. *For the RNTS $\{(1'), \ldots, (9'), (10)\}$ from Ex. 14, any function rt with $\mathsf{rt}(\mathsf{isort}) \geq \lfloor \frac{x_1 - 1}{2} \rfloor \cdot x_2 + 1$, $\mathsf{rt}(\mathsf{ins}) \geq x_2$, $\mathsf{rt}(\mathsf{if}) \geq x_3 - 1$, and $\mathsf{rt}(\mathsf{gt}) \geq 0$ is a runtime bound (recall that the gt-rules have weight 0). Similarly, any sz with $\mathsf{sz}(\mathsf{isort}) \geq x_1 + x_2 - 1$, $\mathsf{sz}(\mathsf{ins}) \geq x_1 + x_2 + 1$, $\mathsf{sz}(\mathsf{if}) \geq x_2 + x_3 + 1$, $\mathsf{sz}(\mathsf{gt}) \geq 1$ is a size bound.*

A runtime bound clearly gives rise to an upper bound on the runtime complexity.

Theorem 18 (rt and irc). *Let rt be a runtime bound for an RNTS $\mathcal{P}$. Then for all $n \in \mathbb{N}$, we have $\mathrm{irc}_{\mathcal{P}}(n) \leq \sup\{[\![\mathsf{rt}(f)\,\{x_1/n_1, \ldots, x_k/n_k\}]\!] \mid f \in \Sigma, n_1, ..., n_k \in \mathbb{N}, \sum_{i=1}^{k} n_i < n\}$. So in particular, $\mathrm{irc}_{\mathcal{P}}(n) \in \mathcal{O}(\sum_{f \in \Sigma} [\![\mathsf{rt}(f)\,\{x_1/n, \ldots, x_k/n\}]\!])$.*

Thus, a suitable runtime bound rt for the RNTS $\{(1'), \ldots, (9'), (10)\}$ yields $\mathrm{irc}(n) \in \mathcal{O}(n^2)$, cf. Ex. 17. In Sect. 4.2 we present a new technique to infer runtime and size bounds rt and sz automatically with existing complexity tools for ITSs. As these tools usually return only runtime bounds, Sect. 4.1 shows how they can also be used to generate size bounds.

4.1 Size Bounds as Runtime Bounds

We first present a transformation for a large class of ITSs that lets us obtain size bounds from any method that can infer runtime bounds. The transformation extends each function symbol from Σ by an additional accumulator argument. Then terms that are multiplied with the result of a function are collected in the accumulator. Terms that are added to the result are moved to the weight of the rule.

Theorem 19 (ITS Size Bounds). *Let $\mathcal{P}$ be an ITS whose rules are of the form $\ell \xrightarrow{w} u + v \cdot r\,[\varphi]$ or $\ell \xrightarrow{w} u\,[\varphi]$ with $u, v \in \mathcal{T}(\Sigma_{\mathsf{exp}}, \mathcal{V})$ and $\mathrm{root}(r) \in \Sigma$. Let $\mathcal{P}_{\mathsf{size}} =$*

$$\{f'(x_1,...,x_k,z) \xrightarrow{u\cdot z} g'(t_1,...,t_n, v\cdot z)\,[\varphi] \mid f(x_1,...,x_k) \xrightarrow{w} u + v \cdot g(t_1,...,t_n)\,[\varphi] \in \mathcal{P}\}$$
$$\cup \quad \{f'(x_1,...,x_k,z) \xrightarrow{u\cdot z} 0\ [\varphi] \mid f(x_1,...,x_k) \xrightarrow{w} u\ [\varphi] \in \mathcal{P}\}$$

for a fresh variable $z \in \mathcal{V}$. Let rt be a runtime bound for $\mathcal{P}_{\mathsf{size}}$. Then sz with $\mathsf{sz}(f) = \mathsf{rt}(f')\{x_{k+1}/1\}$ for any $f \in \Sigma$ is a size bound for $\mathcal{P}$.

Thm. 19 can be generalized to right-hand sides like $f(x){+}2{\cdot}g(y)$ with $f, g \in \Sigma$, cf. App. B. However, it is not applicable if the results of function calls are multiplied on right-hand sides (e.g., $f(x) \cdot g(y)$) and our technique fails in such cases.

Example 20. To get a size bound for $\mathcal{P}^{\mathsf{gt}} = \{(7'), (8'), (9')\}$, we construct $\mathcal{P}^{\mathsf{gt}}_{\mathsf{size}}$:

$$\mathsf{gt}'(x,y,z) \xrightarrow{z} 0\ [x = 1 \land \ldots] \qquad \mathsf{gt}'(x',y,z) \xrightarrow{z} 0\ [x' = 1 + x \land y = 1 \land \ldots]$$
$$\mathsf{gt}'(x',y',z) \xrightarrow{0} \mathsf{gt}'(x,y,z)\,[x' = 1 + x \land y' = 1 + y \land \ldots]$$

Existing ITS tools can compute a runtime bound like $\mathsf{rt}(\mathsf{gt}') = x_3$ for $\mathcal{P}^{\mathsf{gt}}_{\mathsf{size}}$. Hence, by Thm. 19 we obtain the size bound sz for $\mathcal{P}^{\mathsf{gt}}$ with $\mathsf{sz}(\mathsf{gt}) = \mathsf{rt}(\mathsf{gt}')\{x_3/1\} = 1$.

4.2 Complexity Bounds for Recursive Programs

Now we show how complexity tools for ITSs can be used to infer runtime and size bounds for RNTSs. We first define a *call-graph* relation $\sqsupset$ to determine in which order we analyze symbols of Σ. Essentially, $f \sqsupset g$ holds iff $f(\ldots)$ rewrites to a term containing g.

Definition 21 ($\sqsupset$). *For an RNTS $\mathcal{P}$, the* call-graph relation $\sqsupset$ *is the transitive closure of $\{(\mathrm{root}(\ell), g) \mid \ell \xrightarrow{w} r[\varphi] \in \mathcal{P}, g \in \Sigma$ occurs in $r\}$. An RNTS has* nested recursion *iff it has a rule $\ell \xrightarrow{w} r\,[\varphi]$ with $\mathrm{root}(r|_\pi) \sqsupset \mathrm{root}(\ell)$ and $\mathrm{root}(r|_{\pi'}) \sqsupset \mathrm{root}(\ell)$ for positions $\pi < \pi'$. As usual, $\pi < \pi'$ means that π is a proper prefix of π' (i.e., that π' is* strictly below *π). A symbol $f \in \Sigma^{\mathcal{P}}_d$ is a* bottom symbol *iff $f \sqsupset g$ implies $g \sqsupset f$ for all $g \in \Sigma^{\mathcal{P}}_d$. The sub-RNTS of $\mathcal{P}$* induced *by f is $\mathcal{P}^f = \{\ell \xrightarrow{w} r[\varphi] \in \mathcal{P} \mid f \sqsupseteq \mathrm{root}(\ell)\}$, where $\sqsupseteq$ is the reflexive closure of $\sqsupset$.*

Example 22. For the RNTS $\mathcal{P}$ from Ex. 14 and 17, we have $\mathsf{isort} \sqsupset \mathsf{ins} \sqsupset \mathsf{if} \sqsupset \mathsf{ins} \sqsupset \mathsf{gt}$. The only bottom symbol is gt. It induces the sub-RNTS $\mathcal{P}^{\mathsf{gt}} = \{(7'), (8'), (9')\}$, ins induces $\{(3'), \ldots, (9'), (10)\}$, and isort induces the full RNTS of Ex. 14.

Our approach cannot handle programs like $f(\ldots) \to f(\ldots f(\ldots)\ldots)$ with nested recursion, but such programs rarely occur in practice. To compute bounds for an RNTS $\mathcal{P}$ without nested recursion, we start with the trivial bounds $\mathsf{rt}(f) = \mathsf{sz}(f) = \omega$ for all $f \in \Sigma^{\mathcal{P}}_d$. In each step, we analyze the sub-RNTS $\mathcal{P}^f$ induced by a bottom symbol f and refine rt and sz for all defined symbols of $\mathcal{P}^f$. Afterwards we remove the rules $\mathcal{P}^f$ from $\mathcal{P}$ and continue with the next bottom symbol. By this removal of rules, the former defined symbol f becomes a constructor, and former non-bottom symbols are turned into bottom symbols.

Algorithm 1 Computing Runtime and Size Bounds for RNTSs

1 Let $\mathsf{rt}(f) := \mathsf{sz}(f) := \omega$ for each $f \in \Sigma_d^{\mathcal{P}}$ and $\mathsf{rt}(f) := \mathsf{sz}(f) := 0$ for each $f \in \Sigma_c^{\mathcal{P}}$.
2 If $\mathcal{P}$ has nested recursion, then return rt and sz.
3 While $\mathcal{P}$ is not empty:
  3.1 Choose a bottom symbol f of $\mathcal{P}$ and let $\mathcal{P}^f$ be the sub-RNTS induced by f.
  3.2 Construct $\mathcal{P}^f_{\mathsf{sz}}$ according to Thm. 27 and $(\mathcal{P}^f_{\mathsf{sz}})_{\mathsf{size}}$ according to Thm. 19 (resp. its generalization) if possible, otherwise return rt and sz.
  3.3 Compute a runtime bound for $(\mathcal{P}^f_{\mathsf{sz}})_{\mathsf{size}}$ using existing ITS tools and let sz_f be this bound (cf. Thm. 19).
  3.4 For each $g \in \Sigma_d^{\mathcal{P}^f}$, let $\mathsf{sz}(g) := \mathsf{sz}_f(g)$.
  3.5 Construct $\mathcal{P}^f_{\mathsf{rt},\mathsf{sz}}$ according to Thm. 27.
  3.6 Compute a runtime bound rt_f for $\mathcal{P}^f_{\mathsf{rt},\mathsf{sz}}$ using existing ITS tools.
  3.7 For each $g \in \Sigma_d^{\mathcal{P}^f}$, let $\mathsf{rt}(g) := \mathsf{rt}_f(g)$.
  3.8 Let $\mathcal{P} := \mathcal{P} \setminus \mathcal{P}^f$.
4 Return rt and sz.

To analyze the RNTS $\mathcal{P}^f$, Thm. 27 will transform $\mathcal{P}^f$ into two ITSs $\mathcal{P}^f_{\mathsf{sz}}$ and $\mathcal{P}^f_{\mathsf{rt},\mathsf{sz}}$ by abstracting away calls to functions which we already analyzed. Then existing tools for ITSs can be used to compute a size resp. runtime bound for $\mathcal{P}^f_{\mathsf{sz}}$ resp. $\mathcal{P}^f_{\mathsf{rt},\mathsf{sz}}$. Our overall algorithm to infer bounds for RNTSs is summarized in Alg. 1. It clearly terminates, as every loop iteration eliminates a defined symbol (since Step 3.8 removes all rules for the currently analyzed symbol f).

When computing bounds for a bottom symbol $f \in \Sigma_d^{\mathcal{P}}$, we already know (weakly monotonic) size and runtime bounds for all constructors $g \in \Sigma_c^{\mathcal{P}}$. Hence to transform RNTSs into ITSs, *outer* calls of constructors g in terms $g(\dots f(\dots)\dots)$ can be replaced by $\mathsf{sz}(g)$. In Def. 23, while $\mathsf{sz}(t)$ replaces *all* calls to procedures $g \in \Sigma$ in t by their size bound, the *outer abstraction* $\mathfrak{a}^{\circ}_{\mathsf{sz}}(t)$ only replaces constructors $g \in \Sigma_c^{\mathcal{P}}$ by their size bound $\mathsf{sz}(g)$, provided that they do not occur below defined symbols $f \in \Sigma_d^{\mathcal{P}}$.

Definition 23 (Outer Abstraction). *Let $\mathcal{P}$ be an RNTS with the size bound* sz. *We lift* sz *to terms by defining* $\mathsf{sz}(x) = x$ *for* $x \in \mathcal{V}$ *and*

$$\mathsf{sz}(g(s_1,\dots,s_n)) = \begin{cases} \mathsf{sz}(g)\,\{x_j/\mathsf{sz}(s_j) \mid 1 \le j \le n\} & \text{if } g \in \Sigma \\ g(\mathsf{sz}(s_1),\dots,\mathsf{sz}(s_n)) & \text{if } g \in \Sigma_{\mathit{exp}} \end{cases}$$

The outer abstraction *of a term is defined as* $\mathfrak{a}^{\circ}_{\mathsf{sz}}(x) = x$ *for* $x \in \mathcal{V}$ *and*

$$\mathfrak{a}^{\circ}_{\mathsf{sz}}(g(s_1,\dots,s_n)) = \begin{cases} \mathsf{sz}(g)\,\{x_j/\mathfrak{a}^{\circ}_{\mathsf{sz}}(s_j) \mid 1 \le j \le n\} & \text{if } g \in \Sigma_c^{\mathcal{P}} \\ g(\mathfrak{a}^{\circ}_{\mathsf{sz}}(s_1),\dots,\mathfrak{a}^{\circ}_{\mathsf{sz}}(s_n)) & \text{if } g \in \Sigma_{\mathit{exp}} \\ g(s_1,\dots,s_n) & \text{if } g \in \Sigma_d^{\mathcal{P}} \end{cases}$$

Example 24. Consider the following variant $\mathcal{R}^{\times}$ *of* `AG01/#3.16.xml` *from the TPDB*[5] *and its RNTS-counterpart* $\wr\mathcal{R}^{\times}\wr$*:*

[5] *Termination Problems Data Base*, the collection of examples used at the annual *Termination and Complexity Competition*, see `http://termination-portal.org`.

$\mathcal{R}^{\times}$:	$\wr\mathcal{R}^{\times}\int$:			
$\mathsf{f}_+(\mathsf{0}, y) \quad \to y$	$\mathsf{f}_+(x, y) \xrightarrow{1} y$		$[x \ = 1 \wedge \ldots]$	(11)
$\mathsf{f}_+(\mathsf{s}(x), y) \to \mathsf{s}(\mathsf{f}_+(x, y))$	$\mathsf{f}_+(x', y) \xrightarrow{1} 1 + \mathsf{f}_+(x, y)$		$[x' = x + 1 \wedge \ldots]$	(12)
$\mathsf{f}_\times(\mathsf{0}, y) \quad \to \mathsf{0}$	$\mathsf{f}_\times(x, y) \xrightarrow{1} 1$		$[x \ = 1 \wedge \ldots]$	(13)
$\mathsf{f}_\times(\mathsf{s}(x), y) \to \mathsf{f}_+(\mathsf{f}_\times(x, y), y)$	$\mathsf{f}_\times(x', y) \xrightarrow{1} \mathsf{f}_+(\mathsf{f}_\times(x, y), y)$		$[x' = x + 1 \wedge \ldots]$	(14)

Assume that we already analyzed its only bottom symbol f_+ and obtained $\mathsf{sz}(\mathsf{f}_+) = x_1 + x_2$ and $\mathsf{rt}(\mathsf{f}_+) = x_1$. Afterwards, (11) and (12) were removed. Now Def. 23 is used to transform the sub-RNTS $\{(13), (14)\}$ induced by $\mathsf{f}_\times$ into an ITS. The only rule of $\wr\mathcal{R}^{\times}\int$ that violates the restriction of ITSs is (14). Thus, let $(14')$ result from (14) by replacing its right-hand side by $\mathfrak{a}^{\mathsf{o}}_{\mathsf{sz}}(\mathsf{f}_+(\mathsf{f}_\times(x, y), y)) = \mathsf{sz}(\mathsf{f}_+)\,\{x_1/\mathsf{f}_\times(x, y), x_2/y\} = \mathsf{f}_\times(x, y){+}y$. Now $\{(13), (14')\}$ is an ITS, and together with Thm. 19, existing ITS tools can generate a size bound like $\mathsf{sz}(\mathsf{f}_\times) = x_1 \cdot x_2$.

To finish the transformation of RNTSs to ITSs, we now handle terms like $f(\ldots g(\ldots) \ldots)$ where $f \in \Sigma_d^{\mathcal{P}}$ is the bottom symbol we are analyzing and we have an *inner* call of a constructor $g \in \Sigma_c^{\mathcal{P}}$. We would like to replace g by $\mathsf{sz}(g)$ again. However, f might behave non-monotonically (i.e., f might need *less* runtime on *greater* arguments). Therefore, we replace all inner calls $g(\ldots)$ of constructors by fresh variables x. The size bound of the replaced call $g(\ldots)$ is an upper bound for the value of x, but x can also take smaller values.

Definition 25 (Inner Abstraction). *Let $\mathcal{P}$ be an RNTS with size bound sz, t be a term, and $\mathcal{P}os_c^{top} \subseteq \mathcal{P}os(t)$ be the topmost positions of $\Sigma_c^{\mathcal{P}}$-symbols below $\Sigma_d^{\mathcal{P}}$-symbols in t. Thus, $\mu \in \mathcal{P}os_c^{top}$ iff $\mathrm{root}(t|_\mu) \in \Sigma_c^{\mathcal{P}}$, there exists a $\pi < \mu$ with $\mathrm{root}(t|_\pi) \in \Sigma_d^{\mathcal{P}}$, and $\mathrm{root}(t|_{\pi'}) \in \Sigma_{\mathsf{exp}}$ for all $\pi < \pi' < \mu$. For $\mathcal{P}os_c^{top} = \{\mu_1, \ldots, \mu_k\}$, t's* inner abstraction *is $\mathfrak{a}^{\mathsf{i}}(t) = t[x_1]_{\mu_1} \ldots [x_k]_{\mu_k}$ where $x_1, \ldots, x_k$ are pairwise different fresh variables, and its* condition *is $\psi^{\mathsf{i}}_{\mathsf{sz}}(t) = \bigwedge_{1 \le i \le k} x_i \le \mathsf{sz}(t|_{\mu_i})$.*

Example 26. For the RNTS of Ex. 14 and 17, we start with analyzing $\mathcal{P}^{\mathsf{gt}}$ which yields $\mathsf{sz}(\mathsf{gt}) = 1$ and $\mathsf{rt}(\mathsf{gt}) = 0$, cf. Ex. 20. After removing the gt-rules, the new bottom symbols are ins and if. The right-hand side of Rule $(4')$ contains a call of gt below *the symbol if. With the size bound $\mathsf{sz}(\mathsf{gt}) = 1$, the inner abstraction of this right-hand side is $\mathfrak{a}^{\mathsf{i}}(\mathsf{if}(\mathsf{gt}(x, y), x, ys')) = \mathsf{if}(x_1, x, ys')$, and the corresponding condition $\psi^{\mathsf{i}}_{\mathsf{sz}}(\mathsf{if}(\mathsf{gt}(x, y), x, ys'))$ is $x_1 \le 1$, since $\mathsf{sz}(\mathsf{gt}(x, y)) = 1$.*

Thm. 27 states how to transform RNTSs into ITSs in order to compute runtime and size bounds. Suppose that we have already analyzed the function symbols $g_1, \ldots, g_m$, that f becomes a new bottom symbol if the rules for $g_1, \ldots, g_m$ are removed, that $\mathcal{Q}$ is the sub-RNTS induced by f, and that $\mathcal{P}$ results from $\mathcal{Q}$ by deleting the rules for $g_1, \ldots, g_m$. Thus, if g_i occurs in $\mathcal{P}$, then $g_i \in \Sigma_c^{\mathcal{P}}$.

So in our leading example, we have $g_1 = \mathsf{gt}$ (i.e., all gt-rules were analyzed and removed). Thus, ins is a new bottom symbol. If we want to analyze it by Thm. 27, then $\mathcal{Q}$ contains all ins-, if-, and gt-rules and $\mathcal{P}$ just contains all ins- and if-rules.

Since we restricted ourselves to RNTSs $\mathcal{Q}$ without nested recursion, $\mathcal{P}$ has no nested defined symbols. To infer a *size* bound for the bottom symbol f of $\mathcal{P}$, we abstract away inner occurrences of g_i by $\mathfrak{a}^{\mathsf{i}}$ (e.g., gt on the right-hand side of Rule ($4'$) in our example), and we abstract away outer occurrences of g_i by $\mathfrak{a}^{\mathsf{o}}_{\mathsf{sz}}$. So every right-hand side r is replaced by $\mathfrak{a}^{\mathsf{o}}_{\mathsf{sz}}(\mathfrak{a}^{\mathsf{i}}(r))$ and we add the condition $\psi^{\mathsf{i}}_{\mathsf{sz}}(r)$ which restricts the values of the fresh variables introduced by $\mathfrak{a}^{\mathsf{i}}$.

To infer *runtime* bounds, inner occurrences of g_i are also abstracted by $\mathfrak{a}^{\mathsf{i}}$, and outer occurrences of g_i are simply removed. So every right-hand side r is replaced by $\sum_{\pi \in \mathcal{P}os_d(r)} \mathfrak{a}^{\mathsf{i}}(r|_\pi)$, where $\mathcal{P}os_d(r) = \{\pi \in \mathcal{P}os(r) \mid \mathrm{root}(r|_\pi) \in \Sigma^{\mathcal{P}}_d\}$. However, we have to take into account how many computation steps would be required in the procedures g_i that were called in r. Therefore, we compute the *cost* of all calls of g_i in a rule's right-hand side and add it to the weight of the rule. To estimate the cost of a call $g_i(s_1, \ldots, s_n)$, we "apply" $\mathsf{rt}(g_i)$ to the size bounds of $s_1, \ldots, s_n$ and add the costs for evaluating $s_1, \ldots, s_n$.

Theorem 27 (Transformation of RNTSs to ITSs). *Let $\mathcal{Q}$ be an RNTS with size and runtime bounds sz and rt and let $\mathcal{P} = \mathcal{Q} \setminus (\mathcal{Q}^{g_1} \cup \ldots \cup \mathcal{Q}^{g_m})$, where $g_1, \ldots, g_m \in \Sigma$ and $\mathcal{Q}^{g_i}$ is the sub-RNTS of $\mathcal{Q}$ induced by g_i. We define*

$$\mathcal{P}_{\mathsf{sz}} \;=\; \{\, \ell \xrightarrow{w} \mathfrak{a}^{\mathsf{o}}_{\mathsf{sz}}(\mathfrak{a}^{\mathsf{i}}(r)) \;\big[\varphi \wedge \psi^{\mathsf{i}}_{\mathsf{sz}}(r)\big] \;\mid\; \ell \xrightarrow{w} r\,[\varphi] \in \mathcal{P} \,\}$$

Let sz' be a size bound for $\mathcal{P}_{\mathsf{sz}}$ where $\mathsf{sz}'(f) = \mathsf{sz}(f)$ for all $f \in \Sigma \setminus \Sigma^{\mathcal{P}}_d$. If $\mathcal{P}$ does not have nested defined symbols, then sz' is a size bound for $\mathcal{Q}$.

To obtain a runtime bound for $\mathcal{Q}$, we define an RNTS $\mathcal{P}_{\mathsf{rt},\mathsf{sz}'}$. To this end, we define the cost *of a term as $\mathfrak{c}_{\mathsf{rt},\mathsf{sz}'}(x) = 0$ for $x \in \mathcal{V}$ and*

$$\mathfrak{c}_{\mathsf{rt},\mathsf{sz}'}(g(s_1, \ldots, s_n)) = \begin{cases} \sum_{1 \le j \le n} \mathfrak{c}_{\mathsf{rt},\mathsf{sz}'}(s_j) + \mathsf{rt}(g)\,\{x_j/\mathsf{sz}'(s_j) \mid 1 \le j \le n\} & \text{if } g \in \Sigma^{\mathcal{P}}_c \\ \sum_{1 \le j \le n} \mathfrak{c}_{\mathsf{rt},\mathsf{sz}'}(s_j) & \text{otherwise} \end{cases}$$

Now $\mathcal{P}_{\mathsf{rt},\mathsf{sz}'} = \{\ell \xrightarrow{w + \mathfrak{c}_{\mathsf{rt},\mathsf{sz}'}(r)} \sum_{\pi \in \mathcal{P}os_d(r)} \mathfrak{a}^{\mathsf{i}}(r|_\pi) \;\big[\varphi \wedge \psi^{\mathsf{i}}_{\mathsf{sz}'}(r)\big] \;\mid\; \ell \xrightarrow{w} r\,[\varphi] \in \mathcal{P}\}$. Then every runtime bound rt' for $\mathcal{P}_{\mathsf{rt},\mathsf{sz}'}$ with $\mathsf{rt}'(f) = \mathsf{rt}(f)$ for all $f \in \Sigma \setminus \Sigma^{\mathcal{P}}_d$ is a runtime bound for $\mathcal{Q}$. Here, all occurrences of ω in $\mathcal{P}_{\mathsf{sz}}$ or $\mathcal{P}_{\mathsf{rt},\mathsf{sz}'}$ are replaced by pairwise different fresh variables.

If $\mathcal{P}$ does not have nested defined symbols, then $\mathcal{P}_{\mathsf{sz}}$ and $\mathcal{P}_{\mathsf{rt},\mathsf{sz}'}$ are ITSs and thus, they can be analyzed by existing ITS tools.

Example 28. We now finish analyzing the RNTS $\wr\mathcal{R}^{\times}\wr$ after updating sz as in Ex. 24. The cost of the right-hand side of (14) is $\mathfrak{c}_{\mathsf{rt},\mathsf{sz}}(\mathsf{f}_+(\mathsf{f}_\times(x,y),y)) = \mathsf{rt}(\mathsf{f}_+)\,\{x_1/x \cdot y, x_2/y\} = x \cdot y$. So for the sub-RNTS $\mathcal{P} = \{(13),(14)\}$ induced by $\mathsf{f}_\times$, $\mathcal{P}_{\mathsf{rt},\mathsf{sz}}$ is

$$\mathsf{f}_\times(x,y) \xrightarrow{1} 0 \;\;[x = 1 \wedge \ldots] \qquad\qquad \mathsf{f}_\times(x',y) \xrightarrow{1+x \cdot y} \mathsf{f}_\times(x,y) \;\;[x' = x + 1 \wedge \ldots]$$

Hence, existing ITS tools like CoFloCo [16,17] or KoAT [13] yield a bound like $\mathsf{rt}(\mathsf{f}_\times) = x_1^2 \cdot x_2$. So by Thm. 13 and 18 we get $\mathrm{irc}_{\mathcal{R}^\times}(n) \le \mathrm{irc}_{\wr\mathcal{R}^\times\wr}(n) \in \mathcal{O}(n^3)$.

Example 29. To finish the analysis of the RNTS from Ex. 14, we continue Ex. 26. After we removed $\mathcal{P}^{\mathsf{gt}}$, the new bottom symbols ins and if both induce $\mathcal{P}^{\mathsf{ins}} = \{(3'), \ldots, (6'), (10)\}$. Constructing $\mathcal{P}^{\mathsf{ins}}_{\mathsf{sz}}$ yields the rules $(3'), (5'), (6'), (10)$, and

$$\mathsf{ins}(x, ys') \xrightarrow{1} \mathsf{if}(x_1, x, ys') \;\;[\,ys' = 1 + y + ys \,\wedge\, \ldots \,\wedge\, x_1 \le 1\,] \qquad (4'')$$

Existing tools like CoFloCo or KoAT compute size bounds like $1 + x_1 + x_2$ *for* ins *and* $1 + x_2 + x_3$ *for* if *using Thm. 19. After updating* sz*, we construct* $\mathcal{P}^{\mathsf{ins}}_{\mathsf{rt},\mathsf{sz}}$ *which consists of* $(4'')$ *and variants of* $(3'), (5'), (6'), (10)$ *with unchanged weights (as* $\mathfrak{c}_{\mathsf{rt},\mathsf{sz}}(\mathsf{gt}(x, y)) = \mathsf{rt}(\mathsf{gt}) = 0$*). ITS tools now infer runtime bounds like* $2 \cdot x_2$ *for* ins *and* $2 \cdot x_3$ *for* if*. After removing* ins *and* if*, we analyze the remaining RNTS* $\mathcal{P}^{\mathsf{isort}} = \{(1'), (2')\}$*. Since the right-hand side of* $(2')$ *contains an inner occurrence of* ins *below* isort*,* $(2')$ *is replaced by*

$$\mathsf{isort}(xs', ys) \xrightarrow{w} \mathsf{isort}(xs, ys') \quad [xs' = 1 + x + xs \land ys' \leq 1 + x + ys \land \ldots]$$

where $w = 1$ *in* $\mathcal{P}^{\mathsf{isort}}_{\mathsf{sz}}$ *and* $w = 1 + \mathsf{rt}(\mathsf{ins})\{x_1/x, x_2/ys\} = 1 + 2 \cdot ys$ *in* $\mathcal{P}^{\mathsf{isort}}_{\mathsf{rt},\mathsf{sz}}$*. Using Thm. 19, one can now infer bounds like* $\mathsf{sz}(\mathsf{isort}) = x_1 + x_2$ *and* $\mathsf{rt}(\mathsf{isort}) = x_1^2 + 2 \cdot x_1 \cdot x_2$*. Hence, by Thm. 18 one can deduce* $\mathrm{irc}(n) \in \mathcal{O}(n^2)$*.*

Based on Thm. 27, we can now show the correctness of our overall analysis.

Theorem 30 (Alg. 1 is Sound). *Let* $\mathcal{P}$ *be an RNTS and let* rt *and* sz *be the result of Alg. 1 for* $\mathcal{P}$*. Then* rt *is a runtime bound and* sz *is a size bound for* $\mathcal{P}$*.*

5 Related Work, Experiments, and Conclusion

To make techniques for complexity analysis of integer programs also applicable to TRSs, we presented two main contributions: First, we showed in Sect. 3 how TRSs can be abstracted to a variant of integer transition systems (called RNTSs) and presented conditions for the soundness of this abstraction. While abstractions from term-shaped data to numbers are common in program analysis (e.g., for proving termination), soundness of our abstraction for *complexity* of TRSs is not trivial. In [3] a related abstraction technique from first-order functional programs to a formalism corresponding to RNTSs is presented. However, there are important differences between such functional programs and term rewriting: In TRSs, one can also rewrite non-ground terms, whereas functional programming only evaluates ground expressions. Moreover, overlapping rules in TRSs may lead to non-determinism. The most challenging part in Sect. 3 is Thm. 9, i.e., showing that the step from innermost term rewriting to ground innermost rewriting is complexity preserving, even for relative rewriting. Mappings from terms to numbers were also used for complexity analysis of logic programs [15]. However, [15] operates on the logic program level, i.e., it does not translate programs to ITSs and it does not allow the application of ITS-techniques and tools.

Our second contribution (Sect. 4) is an approach to lift any technique for runtime complexity of ITSs to handle (non-nested, but otherwise *arbitrary*) recursion as well. This approach is useful for the analysis of recursive arithmetic programs in general. In particular, by combining our two main contributions we obtain a completely modular approach for the analysis of TRSs. To infer runtime bounds, we also compute size bounds, which may be useful on their own as well.

There exist several approaches that also analyze complexity by inferring both runtime and size bounds. Wegbreit [35] tries to generate closed forms for the exact runtime and size of the result of each analyzed function, whereas we estimate runtime and size by upper bounds. Hence, [35] fails whenever finding such exact closed forms automatically is infeasible. Serrano et al. [33] also compute runtime and size bounds, but in contrast to us they work on logic programs, and their approach is based on abstract interpretation. Our technique in Sect. 4 was inspired by our work on the tool KoAT [13], which composes results of alternating size and runtime complexity analyses for ITSs. In [13] we developed a "bottom-up" technique that corresponds to the approach of Sect. 4.2 when restricting it to ordinary ITSs *without (non-tail) recursion*. But in contrast to Sect. 4.2, KoAT's support for recursion is very limited, as it disregards the return values of "inner" calls. Moreover, [13] does not contain an approach like Thm. 19 in Sect. 4.1 which allows us to obtain size bounds from techniques that compute runtime bounds.

RAML [26–28] reduces the inference of resource annotated types (and hence complexity bounds) for ML programs to linear optimization. Like other techniques for functional programs, it is not directly applicable to TRSs due to the differences between ML and term rewriting.[6] Moreover, RAML has two theoretical boundaries w.r.t. modularity [26]: (A) The number of linear constraints arising from type inference grows exponentially in the size of the program. (B) To achieve context-sensitivity, functions are typed differently for different invocations. In our setting, a blow-up similar to (A) may occur within the used ITS tool, but as the program is analyzed one function at a time, this blow-up is exponential in the size of a single function instead of the whole program. To avoid (B), we analyze each function only once. However, RAML takes amortization effects into account and obtains impressive results in practice. Further leading tools for complexity analysis of programs on integers (resp. naturals) are, e.g., ABC [11], C^{4}B [14], CoFloCo [16,17], LoAT [18], Loopus [34], PUBS [1,2], Rank [4], and SPEED [23].

Finally, there are numerous techniques for automated complexity analysis of TRSs, e.g., [7, 8, 24, 32, 36]. While they also allow forms of modularity, the modularity of our approach differs substantially due to two reasons:

(1) Most previous complexity analysis techniques for TRSs are *top-down* approaches which estimate how often a rule $g(\ldots) \to \ldots$ is applied in reductions that start with terms of a certain size. So the complexity of a rule depends on the context of the whole TRS. This restricts the modularity of these approaches, since one cannot analyze g's complexity without taking the rest of the TRS into account. In contrast, we propose a *bottom-up* approach which analyzes how the complexity of any function g depends on g's inputs. Hence, one can analyze g without taking into account how g is called by other functions f.

(2) In our technique, if a function g has been analyzed, we can replace it by its size bound and do not have to regard g's rules anymore when analyzing a function f that calls g. This is possible because we use a fixed abstraction from terms to numbers. In contrast, existing approaches for TRSs cannot remove rules from the original TRS after having oriented them (with a strict order $\succ$), except for special cases. When other parts of the TRS are analyzed afterwards, these previous rules still have to be oriented weakly (with $\succsim$), since existing TRS

[6] See [29] for an adaption of an amortized analysis as in [27] to term rewriting. However, [29] is not automated, and it is restricted to *ground* rewriting with orthogonal rules.

approaches do not have any dedicated size analysis. This makes the existing approaches for TRSs less modular, but also more flexible (since they do not use a fixed abstraction from terms to numbers). In future work, we will try to improve our approach by integrating ideas from [3] which could allow us to infer and to apply multiple norms when abstracting functional programs to RNTSs.

We implemented our contributions in the tool AProVE [22] and evaluated its power on all 922 examples of the category "Runtime Complexity - Innermost Rewriting" of the *Termination and Complexity Competition 2016*.[7] Here, we excluded the 100 examples where AProVE shows $\mathrm{irc}(n) = \omega$.

In our experiments, we consider the previous version of AProVE (AProVE '16), a version using only the techniques from this paper (AProVE RNTS), and AProVE '17 which integrates the techniques from this paper into AProVE's previous approach to analyze irc. In all these versions, AProVE pre-processes the TRS to remove rules with non-basic left-hand sides that are unreachable from basic terms, cf. [19]. AProVE RNTS uses the external tools CoFloCo, KoAT, and PUBS to compute runtime bounds for the ITSs resulting from the technique in Sect. 4. While we restricted ourselves to polynomial arithmetic for simplicity in this paper, KoAT's ability to prove exponential bounds for ITSs also enables AProVE to infer exponential upper bounds for some TRSs. Thus, the capabilities of the back-end ITS tool determine which kinds of bounds can be derived by AProVE. We also compare with TcT 3.1.0 [7], since AProVE and TcT were the most powerful complexity tools for TRSs at the *Termination and Complexity Competition 2016*.

Note that while the approach of Sect. 4 allows us to use *any* existing (or future) ITS tools for complexity analysis of RNTSs, CoFloCo can also infer complexity bounds for recursive ITSs directly, i.e., it does not require the technique in Sect. 4. To this end, CoFloCo analyzes program parts independently and uses linear invariants to compose the results. So CoFloCo's approach differs significantly from Sect. 4, which can also infer non-linear size bounds. Thus, the approach of Sect. 4 is especially suitable for examples where non-linear growth of data causes non-linear runtime. For instance, in Ex. 28 the quadratic size bound for $\mathsf{f}_\times$ is crucial to prove a (tight) cubic runtime bound with the technique of Sect. 4. Consequently, CoFloCo's linear invariants are not sufficient and hence it fails for this RNTS. See [5] for a list of 17 examples with non-linear runtime where Sect. 4 was superior to all other considered techniques in our experiments. However, CoFloCo's amortized analysis often results in very precise bounds, i.e., both approaches are orthogonal. Therefore, as an alternative to Sect. 4, AProVE RNTS also uses CoFloCo to analyze the RNTSs obtained from the transformation in Sect. 3 directly.

The table on the right shows the results of our experiments. As suggested in [8], we used a timeout of 300 seconds per

$\mathrm{irc}_{\mathcal{R}}(n)$	TcT	AProVE RNTS	AProVE '16	AProVE & TcT	AProVE '17
$\mathcal{O}(1)$	47	43	48	53	53
$\leq \mathcal{O}(n)$	276	254	320	354	379
$\leq \mathcal{O}(n^2)$	362	366	425	463	506
$\leq \mathcal{O}(n^3)$	386	402	439	485	541
$\leq \mathcal{O}(n^{>3})$	393	412	439	491	548
$\leq EXP$	393	422	439	491	553

[7] See http://termination-portal.org/wiki/Termination_Competition/

example (on an Intel Xeon with 4 cores at 2.33 GHz each and 16 GB of RAM). AProVE & TcT represents the former state of the art, i.e., for each example here we took the best bound found by AProVE '16 or TcT. A row "$\leq \mathcal{O}(n^k)$" means that the corresponding tools proved a bound $\leq \mathcal{O}(n^k)$ (e.g., TcT proved constant or linear upper bounds in 276 cases). Clearly, AProVE '17 is the most powerful tool, i.e., the contributions of this paper significantly improve the state of the art for complexity analysis of TRSs. This also shows that the new technique of this paper is orthogonal to the existing ones. In fact, AProVE RNTS infers better bounds than AProVE & TcT in 127 cases. In 102 of them, AProVE & TcT fails to prove any bound at all. The main reasons for this orthogonality are that on the one hand, our approaches loses precision when abstracting terms to numbers. But on the other hand, our approach allows us to apply arbitrary tools for complexity analysis of ITSs in the back-end and to benefit from their respective strengths. Moreover as mentioned above, the approach of Sect. 4 succeeds on many examples where non-linear growth of data leads to non-linear runtime, which are challenging for existing techniques.

For further details on our experiments including a detailed comparison of AProVE RNTS and prior techniques for TRSs, and to access AProVE '17 via a web interface, we refer to [5].

Acknowledgments. We thank A. Flores-Montoya for his help with CoFloCo and the anonymous reviewers for their suggestions and comments.

References

1. Albert, E., Arenas, P., Genaim, S., Puebla, G.: Closed-form upper bounds in static cost analysis. Journal of Automated Reasoning 46(2), 161–203 (2011)
2. Albert, E., Arenas, P., Genaim, S., Puebla, G., Zanardini, D.: Cost analysis of object-oriented bytecode programs. Theoretical Comp. Sc. 413(1), 142–159 (2012)
3. Albert, E., Genaim, S., Gutiérrez, R.: A transformational approach to resource analysis with typed-norms. In: LOPSTR '13. pp. 38–53
4. Alias, C., Darte, A., Feautrier, P., Gonnord, L.: Multi-dimensional rankings, program termination, and complexity bounds of flowchart programs. In: SAS '10. pp. 117–133
5. AProVE: `https://aprove-developers.github.io/trs_complexity_via_its/`.
6. Avanzini, M., Felgenhauer, B.: Type introduction for runtime complexity analysis. In: WST '14. pp. 1–5, available from `http://www.easychair.org/smart-program/VSL2014/WST-proceedings.pdf`
7. Avanzini, M., Moser, G., Schaper, M.: TcT: Tyrolean complexity tool. In: TACAS '16. pp. 407–423
8. Avanzini, M., Moser, G.: A combination framework for complexity. Information and Computation 248, 22–55 (2016)
9. Baader, F., Nipkow, T.: Term Rewriting and All That. Cambridge U. Press (1998)
10. Baillot, P., Dal Lago, U., Moyen, J.Y.: On quasi-interpretations, blind abstractions and implicit complexity. Math. Structures in Comp. Sc. 22(4), 549–580 (2012)
11. Blanc, R., Henzinger, T.A., Hottelier, T., Kovács, L.: ABC: Algebraic bound computation for loops. In: LPAR (Dakar) '10. pp. 103–118

12. Bonfante, G., Cichon, A., Marion, J.Y., Touzet, H.: Algorithms with polynomial interpretation termination proof. J. Functional Programming 11(1), 33–53 (2001)
13. Brockschmidt, M., Emmes, F., Falke, S., Fuhs, C., Giesl, J.: Analyzing runtime and size complexity of integer programs. ACM TOPLAS 38(4) (2016)
14. Carbonneaux, Q., Hoffmann, J., Shao, Z.: Compositional certified resource bounds. In: PLDI '15. pp. 467–478
15. Debray, S., Lin, N.: Cost analysis of logic programs. TOPLAS 15(5), 826–875 (1993)
16. Flores-Montoya, A., Hähnle, R.: Resource analysis of complex programs with cost equations. In: APLAS '14. pp. 275–295
17. Flores-Montoya, A.: Upper and lower amortized cost bounds of programs expressed as cost relations. In: FM '16. pp. 254–273
18. Frohn, F., Naaf, M., Hensel, J., Brockschmidt, M., Giesl, J.: Lower runtime bounds for integer programs. In: IJCAR '16. pp. 550–567
19. Frohn, F., Giesl, J.: Analyzing runtime complexity via innermost runtime complexity. In: LPAR '17. pp. 249–268
20. Frohn, F., Giesl, J., Hensel, J., Aschermann, C., Ströder, T.: Lower bounds for runtime complexity of term rewriting. J. Aut. Reasoning 59(1), 121–163 (2017)
21. Fuhs, C., Giesl, J., Parting, M., Schneider-Kamp, P., Swiderski, S.: Proving termination by dep. pairs and inductive theorem proving. JAR 47(2), 133–160 (2011)
22. Giesl, J., Aschermann, C., Brockschmidt, M., Emmes, F., Frohn, F., Fuhs, C., Hensel, J., Otto, C., Plücker, M., Schneider-Kamp, P., Ströder, T., Swiderski, S., Thiemann, R.: Analyzing program termination and complexity automatically with AProVE. Journal of Automated Reasoning 58, 3–31 (2017)
23. Gulwani, S.: SPEED: Symbolic complexity bound analysis. In: CAV '09. pp. 51–62
24. Hirokawa, N., Moser, G.: Automated complexity analysis based on the dependency pair method. In: IJCAR '08. pp. 364–379
25. Hofbauer, D., Lautemann, C.: Termination proofs and the length of derivations. In: RTA '89. pp. 167–177
26. Hoffmann, J.: Types with Potential: Polynomial Resource Bounds via Automatic Amortized Analysis. Ph.D. thesis, Ludwig-Maximilians-University Munich (2011)
27. Hoffmann, J., Aehlig, K., Hofmann, M.: Multivariate amortized resource analysis. ACM Transactions on Programming Languages and Systems 34(3) (2012)
28. Hoffmann, J., Das, A., Weng, S.C.: Towards automatic resource bound analysis for OCaml. In: POPL '17. pp. 359–373
29. Hofmann, M., Moser, G.: Multivariate amortised resource analysis for term rewrite systems. In: TLCA '15. pp. 241–256
30. Kapur, D., Narendran, P., Zhang, H.: On sufficient completeness and related properties of term rewriting systems. Acta Informatica 24, 395–415 (1987)
31. Kounalis, E.: Completeness in data type specifications. EUROCAL '85. pp. 348–362
32. Noschinski, L., Emmes, F., Giesl, J.: Analyzing innermost runtime complexity of term rewriting by dependency pairs. Journal of Aut. Reasoning 51(1), 27–56 (2013)
33. Serrano, A., López-García, P., Hermenegildo, M.: Resource usage analysis of logic programs via abstract interpretation using sized types. Theory and Practice of Logic Programming 14(4-5), 739–754 (2014)
34. Sinn, M., Zuleger, F., Veith, H.: Complexity and resource bound analysis of imperative programs using difference constraints. J. Aut. Reasoning 59(1), 3–45 (2017)
35. Wegbreit, B.: Mechanical program analysis. Comm. ACM 18, 528–539 (1975)
36. Zankl, H., Korp, M.: Modular complexity analysis for term rewriting. Logical Methods in Computer Science 10(1) (2014)
37. Zantema, H.: Termination of term rewriting: Interpretation and type elimination. Journal of Symbolic Computation 17(1), 23–50 (1994)

A Improving the Precision of the Size Abstraction $\wr\cdot\wr$

In this section we present improvements to increase the precision when abstracting TRSs to RNTSs. In App. A.1 we adapt the abstraction $\wr\cdot\wr$ to distinguish different constant constructors, and in App. A.2 we improve the precision of our analysis by *narrowing*.

A.1 Improved Size Abstraction

Example 31. The TRS $\mathcal{R}^{\mathsf{loop}}$ *(*`Beerendonk_07/1.xml`*, on which all tools failed in the* Termination and Complexity Competition 2016*) contains the* gt*-rules (7), (8), (9), and*

$$\mathsf{loop}(\mathsf{true}, x, y) \to \mathsf{loop}(\mathsf{gt}(x,y), \mathsf{p}(x), y) \tag{15}$$
$$\mathsf{loop}(\mathsf{false}, x, y) \to \mathsf{0} \tag{16}$$
$$\mathsf{p}(\mathsf{0}) \to \mathsf{0} \tag{17}$$
$$\mathsf{p}(\mathsf{s}(x)) \to x \tag{18}$$

$\mathcal{R}^{\mathsf{loop}}$ *is completely defined for the obvious type assignment, and* $\wr\mathcal{R}^{\mathsf{loop}}\wr$ *contains the rules (7′), (8′), (9′) for* gt *from Ex. 7 and the following rules.*

$$\mathsf{loop}(b, x, y) \xrightarrow{1} \mathsf{loop}(\mathsf{gt}(x,y), \mathsf{p}(x), y) \qquad [b\ = 1 \land \ldots] \tag{15′}$$
$$\mathsf{loop}(b, x, y) \xrightarrow{1} 1 \qquad [b\ = 1 \land \ldots] \tag{16′}$$
$$\mathsf{p}(x) \xrightarrow{1} 1 \qquad [x\ = 1] \tag{17′}$$
$$\mathsf{p}(x') \xrightarrow{1} x \qquad [x' = 1 + x \land \ldots] \tag{18′}$$

$\mathcal{R}^{\mathsf{loop}}$ *terminates with the runtime complexity* $\mathrm{irc}_{\mathcal{R}^{\mathsf{loop}}}(n) \in \mathcal{O}(n^2)$*, but* $\wr\mathcal{R}^{\mathsf{loop}}\wr$ *does not terminate, as* $\mathsf{gt}(n_1, n_2) \to^*_{\wr\mathcal{R}^{\mathsf{loop}}\wr} 1$ *holds for all* $n_1, n_2 \in \mathbb{N} \setminus \{0\}$ *and thus, we have* $\mathsf{loop}(1,1,1) \to^+_{\wr\mathcal{R}^{\mathsf{loop}}\wr} \mathsf{loop}(1,1,1)$.

Ex. 31 shows that abstracting all constants to 1 loses critical information. To distinguish different constants after applying $\wr\cdot\wr$, we now improve the abstraction.

Definition 32 (Abstraction $\wr\cdot\wr_{\mathrm{con}}$ from TRSs to RNTSs). *Let* $\mathcal{R}/\mathcal{S}$ *be a TRS with the constants* $\Sigma^0_c = \{f \in \Sigma^{\mathcal{R}\cup\mathcal{S}}_c \mid \mathrm{arity}(f) = 0\}$*. For a mapping* $\mathrm{con} : \Sigma^0_c \to \mathbb{N}$ *from constants to numbers, the* improved size abstraction $\wr t\wr_{\mathrm{con}}$ *of a term* t *is defined as follows:*

$$\begin{array}{rcll} \wr x\wr_{\mathrm{con}} &=& x & \textit{for } x \in \mathcal{V} \\ \wr f\wr_{\mathrm{con}} &=& \mathrm{con}(f) & \textit{if } f \in \Sigma^0_c \\ \wr f(t_1,\ldots,t_k)\wr_{\mathrm{con}} &=& 1 + \wr t_1\wr_{\mathrm{con}} + \ldots + \wr t_k\wr_{\mathrm{con}} & \textit{if } f \in \Sigma^{\mathcal{R}\cup\mathcal{S}}_c \setminus \Sigma^0_c \\ \wr f(t_1,\ldots,t_k)\wr_{\mathrm{con}} &=& f(\wr t_1\wr_{\mathrm{con}},\ldots,\wr t_k\wr_{\mathrm{con}}) & \textit{if } f \in \Sigma^{\mathcal{R}\cup\mathcal{S}}_d \end{array}$$

We lift $\wr\cdot\wr_{\mathrm{con}}$ *to rules and TRSs analogous to* $\wr\cdot\wr$ *(cf. Def. 6), where instead of the condition* $\bigwedge_{x\in\mathcal{V}(\ell)} x \geq 1$ *we now use* $\bigwedge_{x\in\mathcal{V}(\ell)}\ x \geq \min\{\mathrm{con}(f) \mid f \in \Sigma^0_c\}$.

The following adaption of Thm. 13 shows how $\wr\cdot\wr_{\mathrm{con}}$ is used in order to infer bounds on the runtime complexity of TRSs via the transformation to RNTSs.

Theorem 33 (Soundness of Abstraction $\wr\cdot\wr_{\mathrm{con}}$). *Let $\mathcal{R}/\mathcal{S}$ be a well-typed constructor system and let* $\mathrm{con}_{\max} = \max(\{1\} \cup \{\mathrm{con}(f) \mid f \in \Sigma_c^0\})$. *Let $\mathcal{N}$ be a terminating variant of $\mathcal{S}$ such that $\mathcal{R}/(\mathcal{S} \cup \mathcal{N})$ is well typed and completely defined. Then we have* $\mathrm{irc}_{\mathcal{R}/\mathcal{S}}(n) \leq \mathrm{irc}_{\wr\mathcal{R}/(\mathcal{S}\cup\mathcal{N})\wr_{\mathrm{con}}}(\mathrm{con}_{\max} \cdot n)$ *for all $n \in \mathbb{N}$.*

Example 34. Using $\mathrm{con}(\mathsf{0}) = \mathrm{con}(\mathsf{false}) = 0$ *and* $\mathrm{con}(\mathsf{true}) = 1$, *the TRS $\mathcal{R}^{\mathsf{loop}}$ from Ex. 31 is transformed to the RNTS $\wr\mathcal{R}^{\mathsf{loop}}\wr_{\mathrm{con}}$ which consists of*

$$\begin{aligned} \mathsf{gt}(x,y) &\xrightarrow{1} 0 && [x\ = 0] && (7'') \\ \mathsf{gt}(x',y) &\xrightarrow{1} 1 && [x' = 1 + x \wedge y = 0] && (8'') \\ \mathsf{loop}(b,x,y) &\xrightarrow{1} 0 && [b = 0] && (16'') \\ \mathsf{p}(x) &\xrightarrow{1} 0 && [x = 0] && (17'') \end{aligned}$$

and rules corresponding to $(9'), (15'), (18')$. *Now the information on the control flow is kept and we have* $\mathrm{irc}_{\wr\mathcal{R}^{\mathsf{loop}}\wr_{\mathrm{con}}}(n) \in \mathcal{O}(n^2)$, *which implies* $\mathrm{irc}_{\mathcal{R}^{\mathsf{loop}}}(n) \in \mathcal{O}(n^2)$ *by Thm. 33.*

A.2 Pre-Processing TRSs by Narrowing

Although $\wr\mathcal{R}^{\mathsf{loop}}\wr_{\mathrm{con}}$ from Ex. 34 terminates, our approach of Sect. 4 still fails.

Example 35. For the two bottom symbols gt *and* p *of* $\wr\mathcal{R}^{\mathsf{loop}}\wr_{\mathrm{con}}$, *we can easily infer the size bounds* $\mathsf{sz}(\mathsf{gt}) = 1$ *and* $\mathsf{sz}(\mathsf{p}) = x_1$. *However, when abstracting the inner calls of* gt *and* p *in Rule* $(15')$ *using these bounds, we obtain the non-terminating rule*

$$\mathsf{loop}(b,x,y) \to \mathsf{loop}(b',x',y) \qquad [b = 1 \wedge b' \leq 1 \wedge x' \leq x]$$

The problem is that our size bounds are too imprecise to distinguish whether p *returns* x_1 *or* $x_1 - 1$ *and whether* gt *returns* 1 *or* 0 *(resp.* true *or* false*).*

To make such case analyses explicit, we apply *narrowing* to the original TRS in a pre-processing step. Related applications of narrowing for complexity analysis of TRSs were proposed in [32], for example. If the TRS is a completely defined constructor system, then a rule $\ell \to r$ with $r|_\pi = f(\ldots g(\ldots)\ldots)$ for $f, g \in \Sigma_d$ can be replaced by those rules that result from performing all possible narrowing steps on $g(\ldots)$. The reason is that in any reduction of ground terms, the inner subterm $g(\ldots)$ must be reduced to normal form before a rule can be applied to the outer symbol f.

Theorem 36 (Narrowing for Complexity). *Let $\mathcal{R}/\mathcal{S}$ be a completely defined constructor system, $\ell \to r \in \mathcal{R} \cup \mathcal{S}$, $r|_\pi = f(\ldots)$ for some $f \in \Sigma_d^{\mathcal{R}\cup\mathcal{S}}$, and let $r|_\mu$ be a basic term for some $\mu > \pi$. Let $\ell_1 \to r_1, \ldots, \ell_m \to r_m \in \mathcal{R} \cup \mathcal{S}$ be all* *(variable-renamed) rules where ℓ_i unifies with $r|_\mu$ and let $\sigma_i = mgu(r|_\mu, \ell_i)$ for $1 \leq i \leq m$. Let*

$$\mathcal{R}' = (\mathcal{R} \setminus \{\ell \to r\}) \cup \{\ell\sigma_i \to r[r_i]_\mu \sigma_i \mid 1 \leq i \leq m\} \quad \textit{and} \quad \mathcal{S}' = \mathcal{S} \setminus \{\ell \to r\}.$$

Then we have $\mathrm{dh}(t, \xrightarrow{\mathsf{i}}_{\mathcal{R}/\mathcal{S}}) \leq 2 \cdot \mathrm{dh}(t, \xrightarrow{\mathsf{i}}_{\mathcal{R}'/\mathcal{S}'})$ *for all ground terms t.*

If both $\ell \to r$ and $\ell_i \to r_i$ are just rules of $\mathcal{S}$, then one can slightly improve the above construction by adding $\ell\sigma_i \to r[r_i]_\mu \sigma_i$ to $\mathcal{S}'$ instead of $\mathcal{R}'$. Together with Thm. 9, Thm. 36 implies that even if $\mathcal{R}/\mathcal{S}$ is not completely defined, if $\mathcal{N}$ is a terminating variant of $\mathcal{S}$ where $\mathcal{R}/(\mathcal{S} \cup \mathcal{N})$ is completely defined, then pre-processing $\mathcal{R}/(\mathcal{S} \cup \mathcal{N})$ by k narrowing steps to a TRS $\mathcal{R}'/\mathcal{M}'$ does not change the asymptotic complexity, i.e., $\mathrm{irc}_{\mathcal{R}/\mathcal{S}}(n) \leq \mathrm{irc}_{\mathcal{R}/(\mathcal{S} \cup \mathcal{N})}(n) \leq 2^k \cdot \mathrm{irc}_{\mathcal{R}'/\mathcal{M}'}(n)$.

Ex. 37 shows that narrowing is indeed useful as a pre-processing step. On the other hand, it usually increases the number of rules. So as a heuristic, our implementation applies just one narrowing step to all basic terms below defined symbols on right-hand sides.

Example 37. In the TRS of Ex. 31, there are two ways to narrow the basic subterm $\mathsf{p}(x)$ *below the symbol* loop *in Rule* (15). *Hence,* (15) *is replaced by*

$$\begin{array}{rll} \mathsf{loop}(\mathsf{true}, \mathsf{0}, y) \to \mathsf{loop}(\mathsf{gt}(\mathsf{0}, y), \mathsf{0}, y) & \textit{using } (17), \ \textit{mgu: } \{x/\mathsf{0}\} & (15_1) \\ \mathsf{loop}(\mathsf{true}, \mathsf{s}(x'), y) \to \mathsf{loop}(\mathsf{gt}(\mathsf{s}(x'), y), x', y) & \textit{using } (18), \ \textit{mgu: } \{x/\mathsf{s}(x')\} & (15_2) \end{array}$$

A next narrowing step simplifies gt *in both rules and replaces* (15_1) *and* (15_2) *by*

$$\begin{array}{rl} \mathsf{loop}(\mathsf{true}, \mathsf{0}, y') \to \mathsf{loop}(\mathsf{false}, \mathsf{0}, y') & \textit{by narrowing } (15_1) \textit{ with } (7) \\ \mathsf{loop}(\mathsf{true}, \mathsf{s}(x''), \mathsf{0}) \to \mathsf{loop}(\mathsf{true}, x'', \mathsf{0}) & \textit{by narrowing } (15_2) \textit{ with } (8) \\ \mathsf{loop}(\mathsf{true}, \mathsf{s}(x''), \mathsf{s}(y')) \to \mathsf{loop}(\mathsf{gt}(x'', y'), x'', \mathsf{s}(y')) & \textit{by narrowing } (15_2) \textit{ with } (9) \end{array}$$

When we replace (15) *by these three rules and use the improved size abstraction, our technique now automatically infers the complexity* $\mathcal{O}(n^2)$ *for Ex. 31.*

B Proofs

To ease the formulation, we use the following notion. For two TRSs $\mathcal{R}$ and $\mathcal{Q}$, $\xrightarrow{\mathcal{Q}}_{\mathcal{R}}$ is the *$\mathcal{Q}$-restricted rewrite relation*, where $s \xrightarrow{\mathcal{Q}}_{\mathcal{R}} t$ iff all proper subterms of the redex of the rewrite step are in normal form w.r.t. $\mathcal{Q}$ (i.e., no left-hand side of $\mathcal{Q}$ matches a proper subterm of the redex). So $\xrightarrow{\mathcal{R}}_{\mathcal{R}} = \xrightarrow{\mathrm{i}}_{\mathcal{R}}$ is the innermost rewrite relation of $\mathcal{R}$ and $\xrightarrow{\varnothing}_{\mathcal{R}}$ is the ordinary full rewrite relation. So then the *innermost rewrite relation* $\xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}}$ of a relative TRS $\mathcal{R}/\mathcal{S}$ is $\xrightarrow{\mathcal{R}\cup\mathcal{S}}{}^{*}_{\mathcal{S}} \circ \xrightarrow{\mathcal{R}\cup\mathcal{S}}_{\mathcal{R}} \circ \xrightarrow{\mathcal{R}\cup\mathcal{S}}{}^{*}_{\mathcal{S}}$.

B.1 Proofs for Sect. 3

To prove Thm. 9 for relative rewriting, we need the following crucial properties of terminating variants.

Lemma 38 (Properties of Terminating Variants). *Let $\mathcal{R}$ and $\mathcal{S}$ be TRSs and let $\mathcal{N}$ be a terminating variant of $\mathcal{S}$. Then we have the following:*

(a) *If a term t has an infinite reduction w.r.t. $\xrightarrow{\mathrm{i}}_{\mathcal{R}\cup\mathcal{N}}$ then it also has an infinite reduction w.r.t. $\xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{N}}$.*
(b) *For any terms s and t, $s \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{N}} t$ implies $s \xrightarrow{\mathrm{i}}_{\mathcal{R}/(\mathcal{S}\cup\mathcal{N})} t$.*

Proof. **(a)** Suppose that t starts an infinite reduction w.r.t. $\xrightarrow{\mathrm{i}}_{\mathcal{R}\cup\mathcal{N}}$. Since $\mathcal{N}$ is innermost terminating, $\xrightarrow{\mathcal{R}\cup\mathcal{N}}_{\mathcal{N}}$ is also innermost terminating and thus, after a finite number of steps with $\xrightarrow{\mathcal{R}\cup\mathcal{N}}_{\mathcal{N}}$, there must be a step with $\xrightarrow{\mathcal{R}\cup\mathcal{N}}_{\mathcal{R}}$, etc. So the reduction has the form $t = t_0 \xrightarrow{\mathcal{R}\cup\mathcal{N}}{}^{*}_{\mathcal{N}} \overline{t_0} \xrightarrow{\mathcal{R}\cup\mathcal{N}}_{\mathcal{R}} t_1 \xrightarrow{\mathcal{R}\cup\mathcal{N}}{}^{*}_{\mathcal{N}} \overline{t_1} \xrightarrow{\mathcal{R}\cup\mathcal{N}}_{\mathcal{R}} t_2 \xrightarrow{\mathcal{R}\cup\mathcal{N}}{}^{*}_{\mathcal{N}} \dots$ This implies $t = t_0 \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{N}} t_1 \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{N}} t_2 \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{N}} \dots$
(b) Recall that $s \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{N}} t$ means $s \xrightarrow{\mathcal{R}\cup\mathcal{N}}{}^{*}_{\mathcal{N}} \overline{s} \xrightarrow{\mathcal{R}\cup\mathcal{N}}_{\mathcal{R}} \overline{t} \xrightarrow{\mathcal{R}\cup\mathcal{N}}{}^{*}_{\mathcal{N}} t$. So all proper subterms of all redexes are in normal form w.r.t. $\mathcal{R}\cup\mathcal{N}$. Hence, they are also in normal form w.r.t. $\mathcal{S}$ and we have $s \xrightarrow{\mathcal{R}\cup\mathcal{S}\cup\mathcal{N}}{}^{*}_{\mathcal{N}} \overline{s} \xrightarrow{\mathcal{R}\cup\mathcal{S}\cup\mathcal{N}}_{\mathcal{R}} \overline{t} \xrightarrow{\mathcal{R}\cup\mathcal{S}\cup\mathcal{N}}{}^{*}_{\mathcal{N}} t$. This implies $s \xrightarrow{\mathrm{i}}_{\mathcal{R}/(\mathcal{S}\cup\mathcal{N})} t$. □

Now for any constructor system $\mathcal{Q}$, we define the notion of a *saturated ground system* $\mathcal{X}$ of $\mathcal{Q}$. The idea is that $\mathcal{X}$ consists of (possibly infinitely many) instantiations of rules from $\mathcal{Q}$, where all variables are instantiated by ground normal forms. Moreover, the conditions on $\mathcal{X}$ ensure that $\mathcal{X}$ can reduce any ground redex of $\mathcal{Q}$ and that it is confluent.

Definition 39 (Saturated Ground System). *Let $\mathcal{Q}$ be a constructor system. Then a possibly infinite TRS $\mathcal{X}$ is called a* saturated ground system *of $\mathcal{Q}$ iff the following conditions hold:*

- *For all $\ell \to r \in \mathcal{Q}$ and all substitutions σ where $x\sigma$ is a ground normal form w.r.t. $\mathcal{Q}$ for all $x \in \mathcal{V}(\ell)$, there is a rule in $\mathcal{X}$ whose left-hand side is $\ell\sigma$.*
- *For all $\ell' \to r' \in \mathcal{X}$ there exists a rule $\ell \to r \in \mathcal{Q}$ such that $\ell' \to r' = \ell\sigma \to r\sigma$ for a substitution σ where $x\sigma$ is a ground normal form w.r.t. $\mathcal{Q}$ for all $x \in \mathcal{V}(\ell)$.*

- $\mathcal{X}$ *does not contain two rules with the same left-hand side.*

So a saturated ground system can be obtained by first taking the infinite set $\{\ell\sigma \to r\sigma \mid \ell \to r \in \mathcal{Q}, \sigma$ instantiates all variables by ground normal forms$\}$. Afterwards, whenever two rules with the same left-hand side are contained in this set, we delete one of them. The following observations about saturated ground systems are obvious.

Lemma 40 (Properties of Saturated Ground Systems). *Let $\mathcal{X}$ be a saturated ground system for a constructor system $\mathcal{Q}$. Then we have the following properties:*

(a) *$\mathcal{X}$ is confluent.*
(b) *Every ground term that is innermost terminating w.r.t. $\mathcal{Q}$ is also terminating w.r.t. $\mathcal{X}$.*
(c) *Every ground normal form w.r.t. $\mathcal{X}$ is also a normal form w.r.t. $\mathcal{Q}$.*

Proof. For (a), note that the rules in $\mathcal{X}$ do not contain any variables (so in particular, $\mathcal{X}$ is left-linear). Moreover, $\mathcal{X}$ is non-overlapping as all its rules have distinct left-hand sides of the form $\ell\sigma$ for a basic term ℓ and a substitution σ that instantiates all variables of ℓ by ground normal forms. Every orthogonal (i.e., left-linear non-overlapping) TRS is confluent.

Claim (b) follows from the fact that every rewrite step with $\mathcal{X}$ is an innermost rewrite step with $\mathcal{Q}$. The reason is that $t \to_{\mathcal{X}} s$ means that there is a $\pi \in \mathcal{P}os(t)$ with $t|_\pi = \ell'$ and $s = t[r']_\pi$ for a (ground) rule $\ell' \to r' \in \mathcal{X}$, i.e., $t|_\pi = \ell\sigma$ and $s = t[r\sigma]_\pi$ for a rule $\ell \to r \in \mathcal{Q}$ and a substitution σ that instantiates all variables of ℓ by ground normal forms. Since ℓ is a basic term, this implies that $\ell\sigma$ does not have any redex below the root.

For (c), note that every ground term that is not in $\mathcal{Q}$-normal form contains an innermost $\mathcal{Q}$-redex $\ell\sigma$, which is also a redex w.r.t. $\mathcal{X}$. □

For any substitution μ that instantiates all variables by ground terms, any saturated ground system $\mathcal{X}$ (for some constructor system $\mathcal{Q}$), and any constructor system $\mathcal{U}$, we now define a relation $\twoheadrightarrow_{\mathcal{X},\mu,\mathcal{U}}$ on terms. Here, $t \twoheadrightarrow_{\mathcal{X},\mu,\mathcal{U}} s$ holds iff s results from $t\mu$ by rewriting $q\mu$ w.r.t. $\mathcal{X}$ for some subterms q of t that are in $\mathcal{U}$-normal form.

Definition 41 ($\twoheadrightarrow_{\mathcal{X},\mu,\mathcal{U}}$). *Let $\mathcal{U}$ be a constructor systems, $\mathcal{X}$ be a TRS, and μ be a substitution such that $x\mu$ is a ground term for all $x \in \mathcal{V}$. Then for any terms t and s, we have $t \twoheadrightarrow_{\mathcal{X},\mu,\mathcal{U}} s$ iff*

- *$t = f(t_1, \dots, t_k)$, $s = f(s_1, \dots, s_k)$, and $t_i \twoheadrightarrow_{\mathcal{X},\mu,\mathcal{U}} s_i$ for all $1 \leq i \leq k$ or*
- *t is in $\mathcal{U}$-normal form and $t\mu \to_{\mathcal{X}}^* s$*

Now we show some properties of the $\twoheadrightarrow_{\mathcal{X},\mu,\mathcal{U}}$-replacement that will be needed afterwards. If $\mathcal{X}$, μ, and $\mathcal{U}$ are clear from the context, we write $\twoheadrightarrow$ instead of $\twoheadrightarrow_{\mathcal{X},\mu,\mathcal{U}}$. As usual, we use ε to denote the empty position and $\pi.\pi'$ is the concatenation of the positions π and π'.

Lemma 42 (Properties of $\rightarrowtriangle_{\mathcal{X},\mu,\mathcal{U}}$). *Let $\mathcal{U}$, $\mathcal{X}$, and μ be as in Def. 41. Then we have the following:*

(a) *For all terms $t \in \mathcal{T}(\Sigma, \mathcal{V})$, we have $t \rightarrowtriangle t\mu$.*
(b) *For all terms $t, s \in \mathcal{T}(\Sigma, \mathcal{V})$, $t \rightarrowtriangle s$ implies $t\mu \to_{\mathcal{X}}^{*} s$.*
(c) *Let $t \in \mathcal{T}(\Sigma, \mathcal{V})$ and let σ, σ' be substitutions such that $x\sigma \rightarrowtriangle x\sigma'$ for all $x \in \mathcal{V}(t)$. Then we have $t\sigma \rightarrowtriangle t\sigma'$.*
(d) *Let $t, s, r, q \in \mathcal{T}(\Sigma, \mathcal{V})$ where $t|_\pi$ is not in normal form w.r.t. $\mathcal{U}$ for some $\pi \in \mathcal{P}os(t)$. Then $t \rightarrowtriangle s$ and $r \rightarrowtriangle q$ imply $t[r]_\pi \rightarrowtriangle s[q]_\pi$.*

Proof. **(a)** We use induction on t. Every variable x is in $\mathcal{U}$-normal form, which implies $x \rightarrowtriangle x\mu$. If $t = f(t_1, \ldots, t_k)$, then we have $t_i \rightarrowtriangle t_i\mu$ for all $1 \leq i \leq k$ by the induction hypothesis, which implies $t \rightarrowtriangle t\mu$.

(b) We again use induction on t. If $t \rightarrowtriangle s$ and $t\mu \not\to_{\mathcal{X}}^{*} s$, then we have $t = f(t_1, \ldots, t_k)$, $s = f(s_1, \ldots, s_k)$, and $t_i \rightarrowtriangle s_i$ for all $1 \leq i \leq k$. Hence, we have $t_i\mu \to_{\mathcal{X}}^{*} s_i$ by the induction hypothesis, which implies $t\mu \to_{\mathcal{X}}^{*} s$.

(c) We use induction on t. For variables, the desired property holds by the prerequisites on σ and σ'. If $t = f(t_1, \ldots, t_k)$, then we have $t_i\sigma \rightarrowtriangle t_i\sigma'$ by the induction hypothesis. Hence, by the definition of $\rightarrowtriangle$, we also have $t\sigma \rightarrowtriangle t\sigma'$.

(d) We use induction on π. If $\pi = \varepsilon$, then the claim is trivial. Thus, we now consider $\pi = j.\pi'$. Hence, $t = f(t_1, \ldots, t_k)$ with $\pi' \in \mathcal{P}os(t_j)$. As t is not in $\mathcal{U}$-normal form, we have $s = f(s_1, \ldots, s_k)$ with $s_i \rightarrowtriangle t_i$ for all $1 \leq i \leq k$. Since $t_j|_{\pi'}$ is not in normal form, the induction hypothesis implies $t_j[r]_{\pi'} \rightarrowtriangle s_j[q]_{\pi'}$. Hence, $t[r]_\pi = f(t_1, \ldots, t_j[r]_{\pi'}, \ldots, t_k) \rightarrowtriangle f(s_1, \ldots, s_j[q]_{\pi'}, \ldots, s_k) = s[q]_\pi$.

□

The following lemma is the crucial observation needed to show that for any ground substitution μ, the derivation height of any term t is at most as large as the derivation height of $t\mu$. To prove this, we will show in Thm. 9 that any (finite or infinite) rewrite sequence $t = t_0 \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}} t_1 \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}} t_2 \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}} \ldots$ can be transformed into a rewrite sequence $t\mu = s_0 \xrightarrow{\mathrm{i}}_{\mathcal{R}/(\mathcal{S}\cup\mathcal{N})}{}^{+} s_1 \xrightarrow{\mathrm{i}}_{\mathcal{R}/(\mathcal{S}\cup\mathcal{N})}{}^{+} s_2 \xrightarrow{\mathrm{i}}_{\mathcal{R}/(\mathcal{S}\cup\mathcal{N})}{}^{+} \ldots$ of at least the same length. For this proof, the essential idea is to show that if $t_i \rightarrowtriangle_{\mathcal{X},\mu,\mathcal{R}\cup\mathcal{S}} s_i$ and $t_i \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}} t_{i+1}$, then there exists an s_{i+1} with $t_{i+1} \rightarrowtriangle_{\mathcal{X},\mu,\mathcal{R}\cup\mathcal{S}} s_{i+1}$ and $s_i \xrightarrow{\mathrm{i}}_{\mathcal{R}/(\mathcal{S}\cup\mathcal{N})}{}^{+} s_{i+1}$, where $\mathcal{X}$ is a saturated ground system for $\mathcal{R} \cup \mathcal{N}$. This step is provided by the following lemma when setting $\mathcal{Q} = \mathcal{R} \cup \mathcal{N}$ and $\mathcal{U} = \mathcal{R} \cup \mathcal{S}$, and when setting $\mathcal{P} = \mathcal{R}$ resp. $\mathcal{P} = \mathcal{S}$ in order to simulate steps with $\mathcal{R}$ or $\mathcal{S}$.

Lemma 43 (Simulating Innermost Rewriting by Ground Innermost Rewriting). *Let $\mathcal{Q}$, $\mathcal{U}$, $\mathcal{P}$ be constructor systems such that every $\mathcal{Q}$-normal form is also an $\mathcal{U}$-normal form and every $\mathcal{U}$-normal form is also a $\mathcal{P}$-normal form, let t, t' be terms with $t \xrightarrow{\mathcal{U}}_{\mathcal{P}} t'$, and let μ instantiate all variables of t by ground terms. Then for any saturated ground system $\mathcal{X}$ for $\mathcal{Q}$ and for any term s with $t \rightarrowtriangle_{\mathcal{X},\mu,\mathcal{U}} s$ where s is innermost terminating w.r.t. $\mathcal{Q}$, there exists a term s' with $t' \rightarrowtriangle_{\mathcal{X},\mu,\mathcal{U}} s'$ and $s \xrightarrow{\mathcal{Q}}_{\mathcal{Q}}^{*} \circ \xrightarrow{\mathcal{Q}}_{\mathcal{P}} s'$.*

Proof. Since $t \xrightarrow{\mathcal{U}}_{\mathcal{P}} t'$, there is a rule $\ell \to r \in \mathcal{P}$, a position $\pi \in \mathcal{P}os(t)$, and a substitution σ such that $t|_\pi = \ell\sigma$ and $t' = t[r\sigma]_\pi$. Thus, $\ell = f(\ell_1, \dots, \ell_k)$ and $t|_\pi = f(t_1, \dots, t_k)$ for a defined symbol $f \in \Sigma_d$, and $\ell_1, \dots, \ell_k$ are constructor terms with $\ell_i\sigma = t_i$ for all $1 \le i \le k$. Let $t \mathrel{-\!\!\triangleright} s$ where s is innermost terminating w.r.t. $\mathcal{Q}$. Since $t|_\pi$ is a $\mathcal{P}$-redex, no subterm of t at a position on or above π is in $\mathcal{P}$-normal form. As every $\mathcal{U}$-normal form is also a $\mathcal{P}$-normal form, we have $\pi \in \mathcal{P}os(s)$ and $s|_\pi = f(s_1, \dots, s_k)$ with $t_i \mathrel{-\!\!\triangleright} s_i$ for all $1 \le i \le k$. For all $x \in \mathcal{V}(\ell)$, let $\kappa^x_1, \dots, \kappa^x_{n_x}$ be all positions of ℓ where x occurs. As $f(t_1, \dots, t_k) = \ell\sigma$, this implies $f(t_1, \dots, t_k)|_{\kappa^x_j} = x\sigma$ for all $1 \le j \le n_x$. Note that on positions that are below the root but above the positions κ^x_j, $f(t_1, \dots, t_k)$ has the same symbol as ℓ. Thus, $f(t_1, \dots, t_k)$ only contains constructors on these positions since ℓ is basic. As $t_i \mathrel{-\!\!\triangleright} s_i$ for all $1 \le i \le k$, this also means that $f(s_1, \dots, s_k)$ and $f(t_1, \dots, t_k)$ have the same symbols on all positions above the positions κ^x_j. Moreover, since $t_i \mathrel{-\!\!\triangleright} s_i$ implies $t_i\mu \to^*_{\mathcal{X}} s_i$ by Lemma 42 (b), we therefore have $x\sigma\mu \to^*_{\mathcal{X}} f(s_1, \dots, s_k)|_{\kappa^x_j}$ for all $1 \le j \le n_x$. Recall that $\mathcal{X}$ is confluent by Lemma 40 (a) and s is innermost terminating w.r.t. $\mathcal{Q}$ (and by Lemma 40 (b) therefore also w.r.t. $\mathcal{X}$). Thus, $f(s_1, \dots, s_k)$ is also terminating w.r.t. $\mathcal{X}$ and hence, there exists a unique term s_x in normal form such that $f(s_1, \dots, s_k)|_{\kappa^x_j} \to^*_{\mathcal{X}} s_x$ for all $1 \le j \le n_x$. For all variables $x \in \mathcal{V}(\ell)$, we define $x\sigma' = s_x$. As $x\sigma$ is in $\mathcal{U}$-normal form (since $\ell\sigma$ contains no $\mathcal{U}$-redex below the root due to the step $\ell\sigma \xrightarrow{\mathcal{U}}_{\mathcal{P}} r\sigma$), $x\sigma\mu \to^*_{\mathcal{X}} s_x = x\sigma'$ implies $x\sigma \mathrel{-\!\!\triangleright} x\sigma'$. Note that since s_x is in normal form w.r.t. $\mathcal{X}$ it is also in normal form w.r.t. $\mathcal{Q}$ by Lemma 40 (c).

Then we have $f(s_1, \dots, s_k) \to^*_{\mathcal{X}} \ell\sigma'$ since for $\mathcal{V}(\ell) = \{x_1, \dots, x_m\}$, we have:

$$
\begin{array}{lll}
 & f(s_1, \dots, s_k) & \\
= & f(t_1, \dots, t_k) & [f(s_1, \dots, s_k)|_{\kappa^{x_1}_1}]_{\kappa^{x_1}_1} \;\dots\; [f(s_1, \dots, s_k)|_{\kappa^{x_1}_{n_1}}]_{\kappa^{x_1}_{n_1}} \\
 & & \qquad\qquad\qquad\vdots \\
 & & [f(s_1, \dots, s_k)|_{\kappa^{x_m}_1}]_{\kappa^{x_m}_1} \;\dots\; [f(s_1, \dots, s_k)|_{\kappa^{x_m}_{n_m}}]_{\kappa^{x_m}_{n_m}} \\
= & \ell & [f(s_1, \dots, s_k)|_{\kappa^{x_1}_1}]_{\kappa^{x_1}_1} \;\dots\; [f(s_1, \dots, s_k)|_{\kappa^{x_1}_{n_1}}]_{\kappa^{x_1}_{n_1}} \\
 & & \qquad\qquad\qquad\vdots \\
 & & [f(s_1, \dots, s_k)|_{\kappa^{x_m}_1}]_{\kappa^{x_m}_1} \;\dots\; [f(s_1, \dots, s_k)|_{\kappa^{x_m}_{n_m}}]_{\kappa^{x_m}_{n_m}} \\
\to^*_{\mathcal{X}} & \ell & [s_{x_1}]_{\kappa^{x_1}_1} \qquad\qquad\quad \dots\; [s_{x_1}]_{\kappa^{x_1}_{n_1}} \\
 & & \qquad\qquad\qquad\vdots \\
 & & [s_{x_m}]_{\kappa^{x_m}_1} \qquad\qquad\quad \dots\; [s_{x_m}]_{\kappa^{x_m}_{n_m}} \\
= & \ell\sigma' &
\end{array}
$$

As every rewrite step with $\mathcal{X}$ is obviously an innermost step with $\mathcal{Q}$, this implies $s = s[f(s_1, \dots, s_k)]_\pi \xrightarrow{\mathcal{Q}}{}^*_{\mathcal{Q}} s[\ell\sigma']_\pi$. Note that σ' instantiates all variables $x \in \mathcal{V}(\ell)$ by the $\mathcal{Q}$-normal form s_x. Thus, as ℓ is a basic term, $\ell\sigma'$ does not have any $\mathcal{Q}$-redex below the root. This implies that we can continue the above reduction by $s \xrightarrow{\mathcal{Q}}{}^*_{\mathcal{Q}} s[\ell\sigma']_\pi \xrightarrow{\mathcal{Q}}_{\mathcal{P}} s[r\sigma']_\pi$.

It remains to show that $t' = t[r\sigma]_\pi \mathrel{-\!\!\triangleright} s[r\sigma']_\pi$ holds. As $x\sigma \mathrel{-\!\!\triangleright} x\sigma'$ for all $x \in \mathcal{V}(r)$, Lemma 42 (c) implies $r\sigma \mathrel{-\!\!\triangleright} r\sigma'$. As $t|_\pi$ is not in normal form w.r.t.

$\mathcal{P}$ and therefore also not in normal form w.r.t. $\mathcal{U}$, we can use Lemma 42 (d) to conclude $t' = t[r\sigma]_\pi \twoheadrightarrow s[r\sigma']_\pi$ which proves the current lemma. □

Now we can show Thm. 9.

Theorem 9 (Soundness of Instantiation and Terminating Variants). *Let $\mathcal{R}$, $\mathcal{S}$ be constructor systems and $\mathcal{N}$ be a terminating variant of $\mathcal{S}$. Then* $\mathrm{dh}(t, \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}}) \leq \mathrm{dh}(t\sigma, \xrightarrow{\mathrm{i}}_{\mathcal{R}/(\mathcal{S}\cup\mathcal{N})})$ *holds for any term t where $t\sigma$ is ground.*

Proof. If $t\mu$ is not terminating w.r.t. $\xrightarrow{\mathrm{i}}_{\mathcal{R}/(\mathcal{S}\cup\mathcal{N})}$, then we get $\mathrm{dh}(t\mu, \xrightarrow{\mathrm{i}}_{\mathcal{R}/(\mathcal{S}\cup\mathcal{N})}) = \omega$ and the theorem obviously holds. Otherwise, note that termination of $t\mu$ w.r.t. $\xrightarrow{\mathrm{i}}_{\mathcal{R}/(\mathcal{S}\cup\mathcal{N})}$ implies innermost termination of $t\mu$ w.r.t. $\mathcal{R}\cup\mathcal{N}$. To see this, assume that $t\mu$ has an infinite innermost reduction w.r.t. $\mathcal{R}\cup\mathcal{N}$. By Lemma 38 (a), then $t\mu$ also has an infinite reduction w.r.t. $\xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{N}}$. Then Lemma 38 (b) implies that $t\mu$ also has an infinite reduction w.r.t. $\xrightarrow{\mathrm{i}}_{\mathcal{R}/(\mathcal{S}\cup\mathcal{N})}$.

Let $\mathcal{Q} = \mathcal{R}\cup\mathcal{N}$ and let $\mathcal{U} = \mathcal{R}\cup\mathcal{S}$. Then every $\mathcal{Q}$-normal form is also an $\mathcal{U}$-normal form by the definition of terminating variants. Moreover, every $\mathcal{U}$-normal form is also an $\mathcal{S}$-normal form and an $\mathcal{R}$-normal form. Let $\mathcal{X}$ be a saturated ground system for $\mathcal{Q}$ and consider a finite or infinite sequence $t \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}} t_1 \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}} t_2 \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}} \dots$ Thus, we have

$$t = t_0 \xrightarrow{\mathcal{U}}{}^{*}_{\mathcal{S}} \overline{t_0} \xrightarrow{\mathcal{U}}_{\mathcal{R}} t_1 \xrightarrow{\mathcal{U}}{}^{*}_{\mathcal{S}} \overline{t_1} \xrightarrow{\mathcal{U}}_{\mathcal{R}} \dots$$

We always have $t \twoheadrightarrow t\mu$ by Lemma 42 (a) (where we again write $\twoheadrightarrow$ instead of $\twoheadrightarrow_{\mathcal{X},\mu,\mathcal{U}}$). Hence, by using Lemma 43 for $\mathcal{P} = \mathcal{S}$, we have $t\mu \xrightarrow{\mathcal{Q}}{}^{*}_{\mathcal{Q}\cup\mathcal{S}} \overline{s_0}$ for a term $\overline{s_0}$ with $\overline{t_0} \twoheadrightarrow \overline{s_0}$. Clearly termination of $t\mu$ w.r.t. $\xrightarrow{\mathrm{i}}_{\mathcal{R}/(\mathcal{S}\cup\mathcal{N})}$ implies that any term that is reachable from $t\mu$ by $\xrightarrow{\mathrm{i}}_{\mathcal{R}\cup\mathcal{S}\cup\mathcal{N}}$ is also terminating w.r.t. $\xrightarrow{\mathrm{i}}_{\mathcal{R}/(\mathcal{S}\cup\mathcal{N})}$. Recall that $t\mu \xrightarrow{\mathcal{Q}}{}^{*}_{\mathcal{Q}\cup\mathcal{S}} \overline{s_0}$ means $t\mu \xrightarrow{\mathcal{R}\cup\mathcal{N}}{}^{*}_{\mathcal{R}\cup\mathcal{S}\cup\mathcal{N}} \overline{s_0}$ and as every $\mathcal{N}$-normal form is also an $\mathcal{S}$-normal form, we also have $t\mu \xrightarrow{\mathcal{R}\cup\mathcal{S}\cup\mathcal{N}}{}^{*}_{\mathcal{R}\cup\mathcal{S}\cup\mathcal{N}} \overline{s_0}$, i.e., $t\mu \xrightarrow{\mathrm{i}}_{\mathcal{R}\cup\mathcal{S}\cup\mathcal{N}}{}^{*} \overline{s_0}$. Thus, termination of $t\mu$ w.r.t. $\xrightarrow{\mathrm{i}}_{\mathcal{R}/(\mathcal{S}\cup\mathcal{N})}$ implies that $\overline{s_0}$ is also terminating w.r.t. $\xrightarrow{\mathrm{i}}_{\mathcal{R}/(\mathcal{S}\cup\mathcal{N})}$. With the same argumentation as in the beginning of the proof, Lemma 38 implies that $\overline{s_0}$ is innermost terminating w.r.t. $\mathcal{R}\cup\mathcal{N}$ and hence, we can apply Lemma 43 for $\mathcal{P} = \mathcal{R}$ to obtain $t\mu \xrightarrow{\mathcal{Q}}{}^{*}_{\mathcal{Q}\cup\mathcal{S}} \overline{s_0} \xrightarrow{\mathcal{Q}}{}^{*}_{\mathcal{Q}} \circ \xrightarrow{\mathcal{Q}}_{\mathcal{R}} s_1$ for a term s_1 with $t_1 \twoheadrightarrow s_1$. Note that since $\mathcal{Q} = \mathcal{R}\cup\mathcal{N}$, this means $t\mu \xrightarrow{\mathcal{R}\cup\mathcal{N}}{}^{*}_{\mathcal{R}\cup\mathcal{S}\cup\mathcal{N}} \circ \xrightarrow{\mathcal{R}\cup\mathcal{N}}_{\mathcal{R}} s_1$. As every $\mathcal{N}$-normal form is also an $\mathcal{S}$-normal form, we have $t\mu \xrightarrow{\mathcal{R}\cup\mathcal{S}\cup\mathcal{N}}{}^{*}_{\mathcal{R}\cup\mathcal{S}\cup\mathcal{N}} \circ \xrightarrow{\mathcal{R}\cup\mathcal{S}\cup\mathcal{N}}_{\mathcal{R}} s_1$, i.e., $t\mu \xrightarrow{\mathrm{i}}_{\mathcal{R}/(\mathcal{S}\cup\mathcal{N})}{}^{+} s_1$.

By repeating this construction, we obtain a rewrite sequence $t\mu \xrightarrow{\mathrm{i}}_{\mathcal{R}/(\mathcal{S}\cup\mathcal{N})}{}^{+} s_1 \xrightarrow{\mathrm{i}}_{\mathcal{R}/(\mathcal{S}\cup\mathcal{N})}{}^{+} s_2 \xrightarrow{\mathrm{i}}_{\mathcal{R}/(\mathcal{S}\cup\mathcal{N})}{}^{+} \dots$ of at least the same length as the original rewrite sequence for t. □

To prove Thm. 13, we need an auxiliary lemma which shows that every rewrite step on ground terms with a constructor system $\mathcal{R}$ can also be simulated by its abstraction $\wr\mathcal{R}\wr$.

Note that $[\![.]\!] : \mathcal{T}(\Sigma \uplus \Sigma_{\mathsf{fml}}, \mathcal{V}) \cup \{\omega\} \to \mathcal{T}(\Sigma \uplus \Sigma_{\mathsf{fml}}, \mathcal{V}) \cup \{\omega\}$ is defined as $[\![x]\!] = x$ for $x \in \mathcal{V} \cup \{\omega\}$. For $\circ \in \{+, \cdot, <\}$, let $[\![\circ(t_1, t_2)]\!] = \bullet([\![t_1]\!], [\![t_2]\!])$ if $[\![t_1]\!], [\![t_2]\!] \in \mathbb{N}$, where $\bullet$ is the arithmetic function associated with the symbol $\circ$. Similarly,

$[\![\wedge(t_1, t_2)]\!] = \bullet([\![t_1]\!], [\![t_2]\!])$ if $[\![t_1]\!], [\![t_2]\!] \in \{\mathsf{true}, \mathsf{false}\}$, where $\bullet$ is the Boolean conjunction. In all other cases, we define $[\![f(t_1, \ldots, t_k)]\!] = f([\![t_1]\!], \ldots, [\![t_k]\!])$.

Lemma 44 (Size Abstraction Does Not Decrease dh**).** *Let $\mathcal{R}, \mathcal{Q}$ be well-typed TRSs where $\mathcal{R}$ is a constructor system and $\mathcal{Q}$ is completely defined w.r.t. a many-sorted signature Σ. Let $m \in \mathbb{N}$ and $\mathcal{P} = \{\wr \ell \to r \wr_m \mid \ell \to r \in \mathcal{R}\}$. Let $s, t \in \mathcal{T}(\Sigma, \varnothing)$ be (well-typed) ground terms. Then $s \xrightarrow{\mathcal{Q}}_{\mathcal{R}} t$ implies $[\![\wr s \wr]\!] \xrightarrow{m}_{\mathcal{P}} [\![\wr t \wr]\!]$.*

Proof. Since $s \xrightarrow{\mathcal{Q}}_{\mathcal{R}} t$, there is a rule $\ell \to r \in \mathcal{R}$ with $s|_\pi = \ell\sigma$ and $t = s[r\sigma]_\pi$ for some substitution σ and some position π. Moreover, no proper subterm of $\ell\sigma$ is a $\mathcal{Q}$-redex. As $\mathcal{Q}$ is completely defined, this means that $\ell\sigma$ is a basic ground term and σ instantiates every $x \in \mathcal{V}$ by a constructor ground term. So if $\ell = f(t_1, \ldots, t_k)$, then $t_1\sigma, \ldots, t_k\sigma$ are constructor ground terms. Hence for all $1 \le i \le k$, we have $\wr t_i\sigma \wr \in \mathcal{T}(\Sigma_{\mathsf{exp}}, \varnothing)$ and $n_i = [\![\wr t_i\sigma \wr]\!] \in \mathbb{N}$. Let δ be a natural substitution with $x_i\delta = n_i$ for fresh variables $x_1, \ldots, x_k$ and $x\delta = [\![\wr x\sigma \wr]\!]$ for all other variables $x \in \mathcal{V}$.

Note that $\mathcal{P}$ contains the rule $\wr \ell \to r \wr_m$, which is $f(x_1, \ldots, x_k) \xrightarrow{m} \wr r \wr\ [c]$ where c is the constraint $\bigwedge_{i=1}^{k} x_i = \wr t_i \wr \ \wedge\ \bigwedge_{x \in \mathcal{V}(\ell)} x \ge 1$. Clearly $[\![c\delta]\!] = \mathsf{true}$, because $[\![x_i\delta]\!] = [\![n_i]\!] = n_i = [\![\wr t_i\sigma \wr]\!] = [\![\wr t_i \wr \sigma]\!] = [\![\wr t_i \wr \delta]\!]$ and $[\![x\delta]\!] = [\![\wr x\sigma \wr]\!] \ge 1$, as σ instantiates every $x \in \mathcal{V}$ by a constructor ground term.

We now prove that $[\![\wr s \wr]\!] \xrightarrow{m}_{\mathcal{P}} [\![\wr t \wr]\!]$ holds by induction on the position π. In the induction base, we have $\pi = \varepsilon$. Thus, $s = \ell\sigma$ and $t = r\sigma$. Hence, we obtain

$$\begin{array}{lll} & [\![\wr s \wr]\!] & \\ = & [\![\wr \ell\sigma \wr]\!] & \\ = & f(n_1, \ldots, n_k) & \\ \xrightarrow{m}_{\mathcal{P}} & [\![\wr r \wr \delta]\!] & \quad \text{as } [\![c\delta]\!] = \mathsf{true} \\ = & [\![\wr r\sigma \wr]\!] & \\ = & [\![\wr t \wr]\!] & \end{array}$$

In the induction step, we have $\pi = i.\pi'$ for some $1 \le i \le k$. So there is some $g \in \Sigma$ with $s = g(s_1, \ldots, s_i, \ldots, s_d)$, $s_i|_{\pi'} = \ell\sigma$, and $t = g(s_1, \ldots, s_i[r\sigma]_{\pi'}, \ldots, s_d)$. The induction hypothesis states that $[\![\wr s_i \wr]\!] \xrightarrow{m}_{\mathcal{P}} [\![\wr s_i[r\sigma]_{\pi'} \wr]\!]$. Now we have

$$\begin{array}{ll} & [\![\wr s \wr]\!] \\ = & [\![\wr g(s_1, \ldots, s_i, \ldots, s_d) \wr]\!] \\ = & g([\![\wr s_1 \wr]\!], \ldots, [\![\wr s_i \wr]\!], \ldots, [\![\wr s_d \wr]\!]) \\ \xrightarrow{m}_{\mathcal{P}} & [\![g([\![\wr s_1 \wr]\!], \ldots, [\![\wr s_i[r\sigma]_{\pi'} \wr]\!], \ldots, [\![\wr s_d \wr]\!])]\!] \\ = & [\![g(\wr s_1 \wr, \ldots, \wr s_i[r\sigma]_{\pi'} \wr, \ldots, \wr s_d \wr)]\!] \\ = & [\![\wr g(s_1, \ldots, s_i[r\sigma]_{\pi'}, \ldots, s_d) \wr]\!] \\ = & [\![\wr t \wr]\!] \end{array}$$

□

Theorem 13 (Soundness of Abstraction $\wr \cdot \wr$). *Let $\mathcal{R}/\mathcal{S}$ be a well-typed, completely defined constructor system. Then* $\mathrm{dh}(t, \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}}) \le \mathrm{dhw}([\![\wr t \wr]\!], \to_{\wr \mathcal{R}/\mathcal{S} \wr})$

holds for all well-typed ground terms t. Let $\mathcal{N}$ be a terminating variant of $\mathcal{S}$ such that $\mathcal{R}/(\mathcal{S} \cup \mathcal{N})$ is also well typed. If $\mathcal{R}/(\mathcal{S} \cup \mathcal{N})$ is completely defined, then we have $\mathrm{irc}_{\mathcal{R}/\mathcal{S}}(n) \leq \mathrm{irc}_{\wr \mathcal{R}/(\mathcal{S}\cup\mathcal{N}) \wr}(n)$ *for all* $n \in \mathbb{N}$.

Proof. By Lemma 44, for any well-typed ground term s, $s \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}} t$ implies $[\![\wr s \wr]\!] \to_{\wr \mathcal{R}/\mathcal{S} \wr}^{+} [\![\wr t \wr]\!]$, where the sum of the weights of the rewrite steps is 1. Therefore,

$$\mathrm{dh}(t, \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}}) \leq \mathrm{dhw}([\![\wr t \wr]\!], \to_{\wr \mathcal{R}/\mathcal{S} \wr}) \quad \text{holds for all well-typed ground terms } t. \tag{19}$$

By the requirement on completely defined TRSs, for every type τ there is a constant c_τ of type τ. Let μ instantiate every variable of type τ by c_τ. Thus, we obtain:

$$\begin{array}{lll}
& \mathrm{irc}_{\mathcal{R}/\mathcal{S}}(n) & \\
= & \sup\{\mathrm{dh}(t, \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}}) \mid t \text{ basic}, |t| \leq n\} & \\
= & \sup\{\mathrm{dh}(t, \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}}) \mid t \text{ well typed and basic}, |t| \leq n\} & \text{by persistence of irc [6]} \\
\leq & \sup\{\mathrm{dh}(t\mu, \xrightarrow{\mathrm{i}}_{\mathcal{R}/(\mathcal{S}\cup\mathcal{N})}) \mid t \text{ well typed and basic}, |t| \leq n\} & \\
& & \text{by Thm. 9} \\
\leq & \sup\{\mathrm{dh}(s, \xrightarrow{\mathrm{i}}_{\mathcal{R}/(\mathcal{S}\cup\mathcal{N})}) \mid s \text{ well typed, basic, and ground}, |s| \leq n\} & \\
& & \text{since } |t| = |t\mu| \text{ for all terms } t \\
\leq & \sup\{\mathrm{dhw}([\![\wr s \wr]\!], \xrightarrow{\mathrm{i}}_{\wr \mathcal{R}/(\mathcal{S}\cup\mathcal{N}) \wr}) \mid s \text{ well typed, basic, and ground}, |s| \leq n\} & \\
& & \text{by (19)} \\
\leq & \sup\{\mathrm{dhw}(q, \to_{\wr \mathcal{R}/(\mathcal{S}\cup\mathcal{N}) \wr}) \mid q \text{ nat-basic}, \|q\| \leq n\} & \text{since } |s| = \|[\![\wr s \wr]\!]\| \text{ for all} \\
& & \text{basic ground terms } s \\
= & \mathrm{irc}_{\wr \mathcal{R}/(\mathcal{S}\cup\mathcal{N}) \wr}(n) &
\end{array}$$

□

B.2 Proofs for Sect. 4

Theorem 18 (rt and irc). *Let* rt *be a runtime bound for an RNTS* $\mathcal{P}$. *Then for all* $n \in \mathbb{N}$, *we have* $\mathrm{irc}_{\mathcal{P}}(n) \leq \sup\{[\![\mathsf{rt}(f)\,\{x_1/n_1, \ldots, x_k/n_k\}]\!] \mid f \in \Sigma, n_1, ..., n_k \in \mathbb{N}, \sum_{i=1}^{k} n_i < n\}$. *So in particular,* $\mathrm{irc}_{\mathcal{P}}(n) \in \mathcal{O}(\sum_{f\in\Sigma} [\![\mathsf{rt}(f)\,\{x_1/n, \ldots, x_k/n\}]\!])$.

Proof. For any $n \in \mathbb{N}$ we have

$$\begin{array}{ll}
\mathrm{irc}_{\mathcal{P}}(n) & \\
= \sup\{\mathrm{dhw}(t, \to_{\mathcal{P}}) \mid t \text{ is nat-basic}, \|t\| \leq n\} & \text{Def. 5} \\
= \sup\{\mathrm{dhw}(f(n_1, \ldots, n_k), \to_{\mathcal{P}}) \mid f \in \Sigma, n_1, \ldots, n_k \in \mathbb{N} \wedge 1 + \sum_{i=1}^{k} n_i \leq n\} & \text{Def. 5} \\
\leq \sup\{[\![\mathsf{rt}(f)\{x_1/n_1, \ldots, x_k/n_k\}]\!] \mid f \in \Sigma, n_1, \ldots, n_k \in \mathbb{N} \wedge 1 + \sum_{i=1}^{k} n_i \leq n\} & \text{Def. 16} \\
= \sup\{[\![\mathsf{rt}(f)\{x_1/n_1, \ldots, x_k/n_k\}]\!] \mid f \in \Sigma, n_1, \ldots, n_k \in \mathbb{N} \wedge \sum_{i=1}^{k} n_i < n\} &
\end{array}$$

The second statement of the theorem follows by weak monotonicity of $\mathsf{rt}(f)$ for all $f \in \Sigma$, i.e., by

$$[\![\mathsf{rt}(f)\{x_1/n_1, \ldots, x_i/n_i, \ldots, x_k/n_k\}]\!] \leq [\![\mathsf{rt}(f)\{x_1/n_1, \ldots, x_i/n_i + 1, \ldots, x_k/n_k\}]\!]$$

for all $n_1, \ldots, n_k \in \mathbb{N}$ and all $1 \leq i \leq k$. The reason is due to the construction of $\mathsf{rt}(f)$: constant functions and variables are monotonic w.r.t. $\leq$, combinations of monotonic functions by $+$ or $\cdot$ are again monotonic, and the special value ω is monotonic as well. $\square$

Instead of Thm. 19, we prove the following generalization to an arbitrary number of defined symbols on the right-hand side. In the following, instead of $t_0 \xrightarrow{m_1}_{\mathcal{P}} \ldots \xrightarrow{m_n}_{\mathcal{P}} t_n$, we often write $t_0 \xrightarrow{m}{}^*_{\mathcal{P}} t_n$, where $m = m_1 + \ldots + m_n$.

Theorem 45 (ITS Size Bounds (Generalized)). *Let $\mathcal{P}$ be an ITS whose rules are of the form $\ell \xrightarrow{w} u\,[\varphi]$ or $\ell \xrightarrow{w} u + \sum_{1 \leq i \leq m} v_i \cdot r_i\,[\varphi]$ for $u, v_i \in \mathcal{T}(\Sigma_{\mathsf{exp}}, \mathcal{V})$ and $\mathrm{root}(r_i) \in \Sigma$. Let*

$$\mathcal{P}_{\mathsf{size}} = \left\{ \begin{array}{r} f'(x_1, \ldots, x_k, z) \xrightarrow{u \cdot z} \sum_{1 \leq i \leq m} g_i'(t_{i,1}, \ldots, t_{i,k_i}, v_i \cdot z)\,[\varphi] \qquad\qquad\qquad \\ \mid f(x_1, \ldots, x_k) \xrightarrow{w} u + \sum_{1 \leq i \leq m} v_i \cdot g_i(t_{i,1}, \ldots, t_{i,k_i})\,[\varphi] \in \mathcal{P} \end{array} \right\}$$
$$\cup \{ f'(x_1, \ldots, x_k, z) \xrightarrow{u \cdot z} 0\ [\varphi] \mid f(x_1, \ldots, x_k) \xrightarrow{w} u\ [\varphi] \in \mathcal{P} \}$$

for a fresh variable $z \in \mathcal{V}$. Let rt be a runtime bound for $\mathcal{P}_{\mathsf{size}}$. Then sz with $\mathsf{sz}(f) = \mathsf{rt}(f')\{x_{k+1}/1\}$ for any $f \in \Sigma$ is a size bound for $\mathcal{P}$.

Proof. To be able to use the runtime bound rt for $\mathcal{P}_{\mathsf{size}}$ as a size bound for $\mathcal{P}$, we prove that if $f(n_1, \ldots, n_k) \to^*_{\mathcal{P}} n$, then we have $f'(n_1, \ldots, n_k, 1) \xrightarrow{e}{}^*_{\mathcal{P}_{\mathsf{size}}} 0$ with $e \geq n$ for all $f \in \Sigma$ and $n, n_1, \ldots, n_k \in \mathbb{N}$. Instead, we prove the following generalized statement for all numbers $d \geq 0$:

$$f(n_1, \ldots, n_k) \to^*_{\mathcal{P}} n \text{ implies } f'(n_1, \ldots, n_k, d) \xrightarrow{e}{}^*_{\mathcal{P}_{\mathsf{size}}} 0 \quad \text{with } e \geq n \cdot d. \tag{20}$$

From this, the claim of Thm. 45 follows.

We prove (20) by induction on the length of the derivation. In the base case, we have $f(n_1, \ldots, n_k) \to_{\mathcal{P}} n$ with some rule $f(x_1, \ldots, x_k) \to u\,[\varphi] \in \mathcal{P}$ and some substitution σ with $[\![u\sigma]\!] = n$, and thus by construction $f'(n_1, \ldots, n_k, d) \xrightarrow{n \cdot d}_{\mathcal{P}_{\mathsf{size}}} 0$, from which (20) trivially follows.

In the induction step, we have

$$f(n_1, \ldots, n_k) \to_{\mathcal{P}} \tilde{u} + \sum_{1 \leq i \leq m} \tilde{v}_i \cdot g_i(\tilde{t}_{i,1}, \ldots, \tilde{t}_{i,k_i}) \to^*_{\mathcal{P}} n$$

for ground terms $\tilde{u}, \tilde{v}_i \in \mathcal{T}(\Sigma_{\mathsf{exp}}, \varnothing)$. Thus, by construction

$$f'(n_1, \ldots, n_k, d) \xrightarrow{\tilde{u} \cdot d}_{\mathcal{P}_{\mathsf{size}}} \sum_{1 \leq i \leq m} \tilde{v}_i \cdot g_i'(\tilde{t}_{i,1}, \ldots, \tilde{t}_{i,k_i}, \tilde{v}_i \cdot d).$$

Let $\tilde{n}_i \in \mathbb{N}$ be the $\mathcal{P}$-normal forms of $g_i(\tilde{t}_{i,1}, \ldots, \tilde{t}_{i,k_i})$ such that $n = \tilde{u} + \sum_{1 \leq i \leq m} \tilde{v}_i \cdot \tilde{n}_i$.

From the induction hypothesis (20), we obtain $g_i'(\tilde{t}_{i,1}, \ldots, \tilde{t}_{i,k_i}, \tilde{v}_i \cdot d) \xrightarrow{e_i}{}^*_{\mathcal{P}_{\mathsf{size}}} 0$ with $e_i \geq \tilde{n}_i \cdot \tilde{v}_i \cdot d$. Hence, the total weight of the $\mathcal{P}_{\mathsf{size}}$-derivation $f'(n_1, \ldots, n_k, d) \to^*_{\mathcal{P}_{\mathsf{size}}} 0$ is $\tilde{u} \cdot d + \sum_{1 \leq i \leq m} e_i \geq \tilde{u} \cdot d + \sum_{1 \leq i \leq m} \tilde{n}_i \cdot \tilde{v}_i \cdot d = n \cdot d$. Thus, (20) follows. $\square$

To ease notation, we now introduce abbreviations for the RNTSs that result from applying the inner and the outer abstraction, respectively ($\mathcal{P}^i_{\mathsf{rt},\mathsf{sz}}$ resp. $\mathcal{P}^o_{\mathsf{sz}}$). We also introduce a third abstraction $\mathfrak{a}^\mathsf{s}$, which eliminates outer constructors as in the construction of $\mathcal{P}_{\mathsf{rt},\mathsf{sz}}$. The result of this new abstraction is $\mathcal{P}^c$. Moreover, we introduce $\mathfrak{c}^\mathsf{i}$ and $\mathfrak{c}^\mathsf{o}$, which correspond to the costs of applying $\mathfrak{a}^\mathsf{i}$ resp. $\mathfrak{a}^\mathsf{s}$. We also introduce an RNTS $\widehat{\mathcal{P}}_{\mathsf{rt},\mathsf{sz}}$ which rewrites every nat-basic term to any results bounded by the size bound sz. Finally, we introduce $\mathcal{P}'_{\mathsf{rt},\mathsf{sz}}$, which results from first applying $\mathfrak{a}^\mathsf{i}$ and then applying $\mathfrak{a}^\mathsf{s}$ to all rules of a RNTS (and adding the corresponding costs and conditions to the resulting rules). While $\mathcal{P}'_{\mathsf{rt},\mathsf{sz}}$ and $\mathcal{P}_{\mathsf{rt},\mathsf{sz}}$ are very similar, their rules have slightly different costs. However, later we will see that every runtime bound for $\mathcal{P}_{\mathsf{rt},\mathsf{sz}}$ is also a runtime bound for $\mathcal{P}'_{\mathsf{rt},\mathsf{sz}}$ (cf. Corollary 52).

To ease readability, in the following we write Σ_d and Σ_c instead of $\Sigma_d^{\mathcal{P}}$ and $\Sigma_c^{\mathcal{P}}$.

Definition 46 (Abstraction and Resulting Costs and RNTSs). *Let $\mathcal{P}$ be an RNTS with size and runtime bounds sz and rt. We define:*

$$
\begin{aligned}
\mathfrak{c}^\mathsf{i}_{\mathsf{rt},\mathsf{sz}}(t) &= \begin{cases} 0 & \text{if } t \in \mathcal{V} \\ \sum_{1 \le i \le k} \mathfrak{c}^\mathsf{i}_{\mathsf{rt},\mathsf{sz}}(t_i) & \text{if } t = f(t_1,\dots,t_k), f \notin \Sigma_d \\ \mathfrak{c}_{\mathsf{rt},\mathsf{sz}}(t) & \text{if } t = f(t_1,\dots,t_k), f \in \Sigma_d \end{cases} \\
\mathcal{P}^i_{\mathsf{rt},\mathsf{sz}} &= \left\{ \ell \xrightarrow{w+\mathfrak{c}^\mathsf{i}_{\mathsf{rt},\mathsf{sz}}(r)} \mathfrak{a}^\mathsf{i}(r)\ \left[\varphi \wedge \psi^\mathsf{i}_{\mathsf{sz}}(r)\right] \;\middle|\; \ell \xrightarrow{w} r\,[\varphi] \in \mathcal{P} \right\} \\
\mathfrak{c}^\mathsf{o}_{\mathsf{rt},\mathsf{sz}}(t) &= \begin{cases} 0 & \text{if } t \in \mathcal{V} \text{ or } \mathrm{root}(t) \in \Sigma_d \\ \sum_{1 \le i \le k} \mathfrak{c}^\mathsf{o}_{\mathsf{rt},\mathsf{sz}}(t_i) & \text{if } t = f(t_1,\dots,t_k), f \in \Sigma_{exp} \\ \mathsf{rt}(f)(\mathsf{sz}(t_1),\dots,\mathsf{sz}(t_k)) + \sum_{1 \le i \le k} \mathfrak{c}^\mathsf{o}_{\mathsf{rt},\mathsf{sz}}(t_i) & \text{if } t = f(t_1,\dots,t_k), f \in \Sigma_c \end{cases} \\
\mathcal{P}^o_{\mathsf{sz}} &= \left\{ \ell \xrightarrow{w} \mathfrak{a}^\mathsf{o}_{\mathsf{sz}}(r)\ [\varphi] \;\middle|\; \ell \xrightarrow{w} r\,[\varphi] \in \mathcal{P} \right\} \\
\mathfrak{a}^\mathsf{s}(t) &= \textstyle\sum_{\pi \in \mathcal{P}os_d(t)} t|_\pi \\
\mathcal{P}^c_{\mathsf{rt},\mathsf{sz}} &= \left\{ \ell \xrightarrow{w+\mathfrak{c}^\mathsf{o}_{\mathsf{rt},\mathsf{sz}}(r)} \mathfrak{a}^\mathsf{s}(r)\ [\varphi] \;\middle|\; \ell \xrightarrow{w} r\,[\varphi] \in \mathcal{P} \right\} \\
\mathcal{P}'_{\mathsf{rt},\mathsf{sz}} &= \left\{ \ell \xrightarrow{w+\mathfrak{c}^\mathsf{i}_{\mathsf{rt},\mathsf{sz}}(r)+\mathfrak{c}^\mathsf{o}_{\mathsf{rt},\mathsf{sz}}(\mathfrak{a}^\mathsf{i}(r))} \mathfrak{a}^\mathsf{s}(\mathfrak{a}^\mathsf{i}(r))\ \left[\varphi \wedge \psi^\mathsf{i}_{\mathsf{sz}}(r)\right] \;\middle|\; \ell \xrightarrow{w} r\,[\varphi] \in \mathcal{P} \right\} \\
\widehat{\mathcal{P}}_{\mathsf{rt},\mathsf{sz}} &= \left\{ f(x_1,\dots,x_k) \xrightarrow{\mathsf{rt}(f)} x\ [x \le \mathsf{sz}(f)] \;\middle|\; f \in \Sigma_c \right\}
\end{aligned}
$$

For ψ^i, $\mathfrak{a}^\mathsf{o}$, $\mathfrak{c}$, $\mathfrak{c}^\mathsf{i}$, $\mathcal{P}^i$, $\mathfrak{c}^\mathsf{o}$, $\mathcal{P}^o$, $\mathcal{P}^c$, and $\widehat{\mathcal{P}}$, from now on we often omit the indices indicating the used runtime and size bound unless they differ from rt resp. sz. Moreover, we assume that all occurrences of ω in $\mathcal{P}^i$, $\mathcal{P}^o$, $\mathcal{P}^c$, $\mathcal{P}'$, and $\widehat{\mathcal{P}}$ are replaced by pairwise different fresh variables.

The following corollary shows that our notations $\mathcal{P}^i$, $\mathcal{P}^o$, and $\mathcal{P}^c$ from Def. 46 can be used to express $\mathcal{P}_{\mathsf{sz}}$ from Thm. 27 and $\mathcal{P}'_{\mathsf{rt},\mathsf{sz}}$.

Corollary 47 (Expressing $\mathcal{P}_{\mathsf{sz}}$ and $\mathcal{P}'_{\mathsf{rt},\mathsf{sz}}$ with $\mathcal{P}^i$ and $\mathcal{P}^o$). *Let $\mathcal{P}$ be an RNTS with size and runtime bounds sz and rt. Then we have $(\mathcal{P}^i)^o = \mathcal{P}_{\mathsf{sz}}$ up to the weights of the rules in $\mathcal{P}_{\mathsf{sz}}$ (which do not matter, since $\mathcal{P}_{\mathsf{sz}}$ is only used to compute size bounds, but no runtime bounds). Moreover, we have $(\mathcal{P}^i)^c = \mathcal{P}'_{\mathsf{rt},\mathsf{sz}}$.*

Proof. We have:

$$
\begin{aligned}
&(\mathcal{P}^i)^o\\
&= \left\{\ell \xrightarrow{w^i} \mathfrak{a}^{\mathsf{o}}(r^i)\;[\varphi^i] \;\middle|\; \ell \xrightarrow{w^i} r^i\;[\varphi^i] \in \mathcal{P}^i\right\}\\
&= \left\{\begin{array}{l|l} \ell \xrightarrow{w^i} \mathfrak{a}^{\mathsf{o}}(r^i)\;[\varphi^i] & \ell \xrightarrow{w^i} r^i\;[\varphi^i] \in \\ & \qquad \left\{\ell \xrightarrow{w+\mathfrak{c}^{\mathsf{i}}(r)} \mathfrak{a}^{\mathsf{i}}(r)\;[\varphi \wedge \psi^{\mathsf{i}}(r)] \;\middle|\; \ell \xrightarrow{w} r\;[\varphi] \in \mathcal{P}\right\}\end{array}\right\}\\
&= \left\{\ell \xrightarrow{w+\mathfrak{c}^{\mathsf{i}}(r)} \mathfrak{a}^{\mathsf{o}}(\mathfrak{a}^{\mathsf{i}}(r))\;[\varphi \wedge \psi^{\mathsf{i}}(r)] \;\middle|\; \ell \xrightarrow{w} r\;[\varphi] \in \mathcal{P}\right\}\\
&= \mathcal{P}_{\mathsf{sz}}
\end{aligned}
$$

and

$$
\begin{aligned}
&(\mathcal{P}^i)^c\\
&= \left\{\ell \xrightarrow{w^i+\mathfrak{c}^{\mathsf{o}}(r^i)} \mathfrak{a}^{\mathsf{s}}(r^i)\;\;[\varphi^i] \;\middle|\; \ell \xrightarrow{w^i} r^i\;[\varphi^i] \in \mathcal{P}^i\right\}\\
&= \left\{\begin{array}{l|l} \ell \xrightarrow{w^i+\mathfrak{c}^{\mathsf{o}}(r^i)} \mathfrak{a}^{\mathsf{s}}(r^i)\;\;[\varphi^i] & \ell \xrightarrow{w^i} r^i\;[\varphi^i] \in \\ & \qquad \left\{\ell \xrightarrow{w+\mathfrak{c}^{\mathsf{i}}(r)} \mathfrak{a}^{\mathsf{i}}(r)\;\;[\varphi \wedge \psi^{\mathsf{i}}(r)] \;\middle|\; \ell \xrightarrow{w} r\;[\varphi] \in \mathcal{P}\right\}\end{array}\right\}\\
&= \left\{\ell \xrightarrow{w+\mathfrak{c}^{\mathsf{i}}(r)+\mathfrak{c}^{\mathsf{o}}(\mathfrak{a}^{\mathsf{i}}(r))} \mathfrak{a}^{\mathsf{s}}(\mathfrak{a}^{\mathsf{i}}(r))\;\;[\varphi \wedge \psi^{\mathsf{i}}(r)] \;\middle|\; \ell \xrightarrow{w} r\;[\varphi] \in \mathcal{P}\right\}\\
&= \mathcal{P}'_{\mathsf{rt},\mathsf{sz}}
\end{aligned}
$$

□

The following straightforward lemma states that bounds for $\mathcal{P}$ are also valid for $\widehat{\mathcal{P}}$ and, in some cases, for certain supersets of $\widehat{\mathcal{P}}$.

Lemma 48 (Runtime and Size Bounds of $\mathcal{P}$ and $\widehat{\mathcal{P}}$). *Let $\mathcal{P}$ be an RNTS with size and runtime bounds sz and rt. Then sz and rt are also size and runtime bounds for $\widehat{\mathcal{P}}$. Moreover, if $\mathsf{sz}(f) = \omega$ resp. $\mathsf{rt}(f) = \omega$ for all $f \in \Sigma_d$, then sz resp. rt is also a size resp. runtime bound for $\mathcal{P} \cup \widehat{\mathcal{P}}$ and for $\mathcal{P}^i \cup \widehat{\mathcal{P}}$.*

Proof. Let $s = f(n_1, \dots, n_k)$ be a nat-basic term such that $s \xrightarrow{m}^+_{\widehat{\mathcal{P}}} q$ (resp. $s \xrightarrow{m}^+_{\mathcal{P}\cup\widehat{\mathcal{P}}} q$ or $s \xrightarrow{m}^+_{\mathcal{P}^i\cup\widehat{\mathcal{P}}} q$) for some term q (the case where the rewrite sequence is empty is trivial).

If $f \in \Sigma_c^{\mathcal{P}}$, then s can neither be reduced by $\mathcal{P}$ nor by $\mathcal{P}^i$, but just by $\widehat{\mathcal{P}}$. Hence, we have $s \xrightarrow{m}_{\widehat{\mathcal{P}}} q \in \mathbb{N}$ where the rule used for this rewrite step is $f(x_1, \dots, x_k) \xrightarrow{\mathsf{rt}(f)} x\,[x \le \mathsf{sz}(f)]$ by definition of $\widehat{\mathcal{P}}$. Thus, we get $m = [\![\mathsf{rt}(f)\sigma]\!]$, and $q \le [\![\mathsf{sz}(f)\sigma]\!]$ where $\sigma = \{x_1/n_1, \dots, x_k/n_k\}$.

If $f \in \Sigma_d^{\mathcal{P}}$, then s is in $\widehat{\mathcal{P}}$-normal form and hence we just have to consider the cases $s \xrightarrow{m}^+_{\mathcal{P}\cup\widehat{\mathcal{P}}} q$ and $s \xrightarrow{m}^+_{\mathcal{P}^i\cup\widehat{\mathcal{P}}} q$. If $\mathsf{rt}(f) = \omega$, we clearly have $m \le [\![\mathsf{rt}(f)\sigma]\!]$ for each natural substitution σ. Moreover, if $\mathsf{sz}(f) = \omega$ and $q \in \mathbb{N}$, we clearly have $q \le [\![\mathsf{sz}(f)\sigma]\!]$ for each natural substitution σ. □

B.2.1 Properties of sz, $\mathfrak{c}$, $\mathfrak{c}^{\mathsf{i}}$, and $\mathfrak{c}^{\mathsf{o}}$

The following auxiliary lemma shows that our lifting of sz from function symbols to terms is sound. From now on, we sometimes write $\mathsf{rt}(f)(t_1,\dots,t_k)$ instead of $\mathsf{rt}(f)\{x_1/t_1,\dots,x_k/t_k\}$ and $\mathsf{sz}(f)(t_1,\dots,t_k)$ instead of $\mathsf{sz}(f)\{x_1/t_1,\dots,x_k/t_k\}$ to ease readability.

Lemma 49 (Soundness of sz on Terms). *Let $\mathcal{P}$ be an RNTS with size bound sz. If $n \in \mathbb{N}$ is a normal form of $s \in \mathcal{T}(\Sigma \cup \Sigma_{\mathsf{exp}}, \varnothing)$ w.r.t. $\mathcal{P}$, then $[\![\mathsf{sz}(s)]\!] \geq n$.*

Proof. We use structural induction on s. If $s \in \mathcal{T}(\Sigma_{\mathsf{exp}}, \varnothing)$, then $s = n$ as s is already in normal form w.r.t. $\mathcal{P}$ and by Def. 23 we have $[\![\mathsf{sz}(s)]\!] = s$. If $s = g(s_1,\dots,s_m)$ and $g \in \Sigma$, then

$$\mathsf{sz}(s) = \mathsf{sz}(g)(\mathsf{sz}(s_1),\dots,\mathsf{sz}(s_m)). \tag{21}$$

For each s_j, let $n_j \in \mathbb{N}$ be the normal form obtained for s_j in the rewrite sequence $s \to_{\mathcal{P}}^* n$, i.e., we have $g(s_1,\dots,s_m) \to_{\mathcal{P}}^* g(n_1,\dots,n_m) \to_{\mathcal{P}}^* n$ and hence

$$[\![\mathsf{sz}(g(n_1,\dots,n_m))]\!] \geq n \tag{22}$$

by Def. 16, as sz is a size bound for $\mathcal{P}$. By the induction hypothesis, we have

$$[\![\mathsf{sz}(s_j)]\!] \geq n_j \text{ for each } 1 \leq j \leq m. \tag{23}$$

Hence, we get:

$$\begin{array}{ll}
\ [\![\mathsf{sz}(s)]\!] & \\
= [\![\mathsf{sz}(g)(\mathsf{sz}(s_1),\dots,\mathsf{sz}(s_m))]\!] & \text{by (21)} \\
\geq [\![\mathsf{sz}(g)(n_1,\dots,n_m)]\!] & \text{by weak monotonicity of } \mathsf{sz}(g) \text{ and (23)} \\
= [\![\mathsf{sz}(g(n_1,\dots,n_m))]\!] & \text{by def. of } \mathsf{sz} \\
\geq n & \text{by (22)}
\end{array}$$

□

According to the following lemma, our lifting of sz to terms is also weakly *monotonic*, i.e., replacing a subterm of a term q with a "smaller" term results in a term whose evaluation is smaller or equal to q. In the following, we often write $\mathcal{T}$ instead of $\mathcal{T}(\Sigma, \mathcal{V})$.

Lemma 50 (Lifting of sz is Monotonic). *Let $t, q \in \mathcal{T}$ and $\pi \in \mathcal{P}os(q)$ such that $[\![\mathsf{sz}(q|_\pi)]\!] \geq [\![\mathsf{sz}(t)]\!]$. Then $[\![\mathsf{sz}(q)]\!] \geq [\![\mathsf{sz}(q[t]_\pi)]\!]$.*

Proof. Let t be an arbitrary term. We use induction on π. If $\pi = \varepsilon$ we get:

$$\begin{array}{ll}
\ [\![\mathsf{sz}(q)]\!] \geq [\![\mathsf{sz}(q[t]_\pi)]\!] & \\
\iff [\![\mathsf{sz}(q)]\!] \geq [\![\mathsf{sz}(t)]\!] & \\
\iff \mathsf{true} & \text{by assumption } [\![\mathsf{sz}(q|_\pi)]\!] = [\![\mathsf{sz}(q)]\!] \geq [\![\mathsf{sz}(t)]\!]
\end{array}$$

If $\pi = i.\pi'$, then $q = f(q_1, \dots, q_k)$ and the induction hypothesis implies

$$[\![\mathsf{sz}(q_i)]\!] \geq [\![\mathsf{sz}(q_i[t]_{\pi'})]\!]. \tag{24}$$

If $f \in \Sigma$, then we get:

$$\begin{array}{ll} & [\![\mathsf{sz}(q)]\!] \\ = & [\![\mathsf{sz}(f)(\mathsf{sz}(q_1), \dots, \mathsf{sz}(q_k))]\!] \quad \text{by def. of } \mathsf{sz} \\ \geq & [\![\mathsf{sz}(f)(\mathsf{sz}(q_1), \dots, \mathsf{sz}(q_i[t]_{\pi'}), \dots, \mathsf{sz}(q_k))]\!] \quad \text{by (24) and monotonicity of } \mathsf{sz}(f) \\ = & [\![\mathsf{sz}(q[t]_\pi)]\!] \quad \text{by def. of } \mathsf{sz} \end{array}$$

If $f \in \Sigma_{\mathsf{exp}}$, then the proof is analogous, since Σ_{exp} only contains the non-constant functions $+$ and $\cdot$ which are weakly monotonic. $\square$

The following lemma clarifies the relation between the costs imposed by $\mathcal{P}'_{\mathsf{rt},\mathsf{sz}}$ and $\mathcal{P}_{\mathsf{rt},\mathsf{sz}}$. Hence, it allows us to show that runtime bounds for $\mathcal{P}_{\mathsf{rt},\mathsf{sz}}$ are also runtime bounds for $\mathcal{P}'_{\mathsf{rt},\mathsf{sz}}$.

Lemma 51 (Relation of $\mathfrak{c}$, $\mathfrak{c}^{\mathsf{i}}$, and $\mathfrak{c}^{\circ}$). *Let $\mathcal{P}$ be an RNTS with size and runtime bounds sz and rt. Let t be a term and let θ be a substitution with $[\![\psi^{\mathsf{i}}(t)\theta]\!] =$ true. Then $[\![\mathfrak{c}^{\mathsf{i}}(t) + \mathfrak{c}^{\circ}(\mathfrak{a}^{\mathsf{i}}(t))\theta]\!] \leq [\![\mathfrak{c}(t)]\!]$.*

Proof. Structural induction on t. If $t \in \mathcal{V}$, then we have

$$[\![\mathfrak{c}^{\mathsf{i}}(t) + \mathfrak{c}^{\circ}(\mathfrak{a}^{\mathsf{i}}(t))\theta]\!] = 0 = [\![\mathfrak{c}(t)]\!].$$

Let $t = f(t_1, \dots, t_k)$. By the induction hypothesis, we have

$$[\![\mathfrak{c}^{\mathsf{i}}(t_i) + \mathfrak{c}^{\circ}(\mathfrak{a}^{\mathsf{i}}(t_i))\theta']\!] \leq [\![\mathfrak{c}(t_i)]\!] \tag{25}$$

for all $1 \leq i \leq k$ and all substitutions θ' with $[\![\psi^{\mathsf{i}}(t_i)\theta']\!] = \mathsf{true}$. If $f \in \Sigma_{\mathsf{exp}}$, then we have

$$\begin{array}{ll} & [\![\mathfrak{c}^{\mathsf{i}}(t) + \mathfrak{c}^{\circ}(\mathfrak{a}^{\mathsf{i}}(t))\theta]\!] \\ = & \sum_{1\leq i\leq k}[\![\mathfrak{c}^{\mathsf{i}}(t_i)]\!] + \sum_{1\leq i\leq k}[\![\mathfrak{c}^{\circ}(\mathfrak{a}^{\mathsf{i}}(t)|_i)\theta]\!] \quad \text{by def. of } \mathfrak{c}^{\mathsf{i}} \text{ and } \mathfrak{c}^{\circ} \\ = & \sum_{1\leq i\leq k}[\![\mathfrak{c}^{\mathsf{i}}(t_i)]\!] + \sum_{1\leq i\leq k}[\![\mathfrak{c}^{\circ}(\mathfrak{a}^{\mathsf{i}}(t_i)\mu)\theta]\!] \quad \text{where } \mu \text{ is a variable renaming} \\ & \qquad \text{such that } \mathfrak{a}^{\mathsf{i}}(t_i)\mu = \mathfrak{a}^{\mathsf{i}}(t)|_i \\ = & \sum_{1\leq i\leq k}[\![\mathfrak{c}^{\mathsf{i}}(t_i)]\!] + \sum_{1\leq i\leq k}[\![\mathfrak{c}^{\circ}(\mathfrak{a}^{\mathsf{i}}(t_i))\mu\theta]\!] \\ \leq & \sum_{1\leq i\leq k}[\![\mathfrak{c}(t_i)]\!] \quad \text{by (25) and } (\ddagger) \\ = & [\![\mathfrak{c}(t)]\!] \quad \text{by def. of } \mathfrak{c} \end{array}$$

For the step marked with ($\ddagger$), note that we have

$$[\![\psi^{\mathsf{i}}(t_i)\mu\theta]\!] = \mathsf{true} \tag{26}$$

as we have

$$\begin{array}{ll} & [\![\psi^{\mathsf{i}}(t)\theta]\!] = \mathsf{true} \\ \iff & [\![\psi^{\mathsf{i}}(f(t_1, \dots, t_k))\theta]\!] = \mathsf{true} \\ \iff & [\![\wedge_{1\leq i\leq k}\psi^{\mathsf{i}}(t_i)\mu\theta]\!] = \mathsf{true} \quad \text{as } f \in \Sigma_{\mathsf{exp}} \text{ and as } \mu \text{ is a variable renaming} \\ & \qquad \text{such that } \mathfrak{a}^{\mathsf{i}}(t_i)\mu = \mathfrak{a}^{\mathsf{i}}(t)|_i \end{array}$$

Moreover, note that the variable renaming μ exists as all variables introduced by $\mathfrak{a}^{\mathsf{i}}$ (which in turn occur in ψ^{i}) are fresh.

If $f \in \Sigma_c$, then we have

$$\begin{aligned}
&[\![\mathfrak{c}^{\mathsf{i}}(t) + \mathfrak{c}^{\circ}(\mathfrak{a}^{\mathsf{i}}(t))\theta]\!] \\
= &\textstyle\sum_{1 \le i \le k} [\![\mathfrak{c}^{\mathsf{i}}(t_i)]\!] + \sum_{1 \le i \le k} [\![\mathfrak{c}^{\circ}(\mathfrak{a}^{\mathsf{i}}(t)|_i)\theta]\!] + [\![\mathsf{rt}(f)(\mathsf{sz}(\mathfrak{a}^{\mathsf{i}}(t)|_1), \dots, \mathsf{sz}(\mathfrak{a}^{\mathsf{i}}(t)|_k))\theta]\!] \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{by def. of } \mathfrak{c}^{\mathsf{i}} \text{ and } \mathfrak{c}^{\circ} \\
= &\textstyle\sum_{1 \le i \le k} [\![\mathfrak{c}^{\mathsf{i}}(t_i)]\!] + \sum_{1 \le i \le k} [\![\mathfrak{c}^{\circ}(\mathfrak{a}^{\mathsf{i}}(t_i)\mu)\theta]\!] + [\![\mathsf{rt}(f)(\mathsf{sz}(\mathfrak{a}^{\mathsf{i}}(t)|_1), \dots, \mathsf{sz}(\mathfrak{a}^{\mathsf{i}}(t)|_k))\theta]\!] \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{see below } (\dagger) \\
= &\textstyle\sum_{1 \le i \le k} [\![\mathfrak{c}^{\mathsf{i}}(t_i)]\!] + \sum_{1 \le i \le k} [\![\mathfrak{c}^{\circ}(\mathfrak{a}^{\mathsf{i}}(t_i))\mu\theta]\!] + [\![\mathsf{rt}(f)(\mathsf{sz}(\mathfrak{a}^{\mathsf{i}}(t)|_1), \dots, \mathsf{sz}(\mathfrak{a}^{\mathsf{i}}(t)|_k))\theta]\!] \\
\le &\textstyle\sum_{1 \le i \le k} [\![\mathfrak{c}(t_i)]\!] + [\![\mathsf{rt}(f)(\mathsf{sz}(\mathfrak{a}^{\mathsf{i}}(t)|_1), \dots, \mathsf{sz}(\mathfrak{a}^{\mathsf{i}}(t)|_k))\theta]\!] \quad \text{by (25) and (26)} \\
\le &\textstyle\sum_{1 \le i \le k} [\![\mathfrak{c}(t_i)]\!] + [\![\mathsf{rt}(f)(\mathsf{sz}(t_1), \dots, \mathsf{sz}(t_k))]\!] \qquad\qquad \text{see below } (\dagger\dagger) \\
= &[\![\mathfrak{c}(t)]\!] \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{by def. of } \mathfrak{c}
\end{aligned}$$

In the step marked with ($\dagger$), μ is again a variable renaming such that $\mathfrak{a}^{\mathsf{i}}(t_i)\mu = \mathfrak{a}^{\mathsf{i}}(t)|_i$ for all $1 \le i \le k$. The step marked with ($\dagger\dagger$) holds as $[\![\psi^{\mathsf{i}}(t)\theta]\!] = \mathsf{true}$ and $f \in \Sigma_c$ implies

$$[\![\mathfrak{a}^{\mathsf{i}}(t)|_{i.\pi}\theta]\!] = [\![\mathfrak{a}^{\mathsf{i}}(t)|_i\theta|_\pi]\!] \le [\![\mathsf{sz}(t|_{i.\pi})]\!] \text{ for each } 1 \le i \le k \text{ and } \pi \in \mathcal{P}os_c^{top}(t|_i).$$

As $\mathfrak{a}^{\mathsf{i}}(t)|_{i.\pi} \in \mathcal{V}$, this implies

$$[\![\mathsf{sz}(\mathfrak{a}^{\mathsf{i}}(t)|_i\theta|_\pi)]\!] \le [\![\mathsf{sz}(t|_{i.\pi})]\!].$$

By Lemma 50, we know

$$[\![\mathsf{sz}(s)]\!] \le [\![\mathsf{sz}(q|_\pi)]\!] \text{ implies } [\![\mathsf{sz}(q[s]_\pi)]\!] \le [\![\mathsf{sz}(q)]\!].$$

With $s = \mathfrak{a}^{\mathsf{i}}(t)|_i\theta|_\pi$ and $q = t|_i$, we get

$$[\![\mathsf{sz}(t|_i[\mathfrak{a}^{\mathsf{i}}(t)|_i\theta|_\pi]_\pi)]\!] \le [\![\mathsf{sz}(t|_i)]\!].$$

Hence, we have

$$[\![\mathsf{sz}(t|_i[\mathfrak{a}^{\mathsf{i}}(t)|_i\theta|_{\pi_1}]_{\pi_1} \dots [\mathfrak{a}^{\mathsf{i}}(t)|_i\theta|_{\pi_m}]_{\pi_m})]\!] \le [\![\mathsf{sz}(t|_i)]\!]$$

where $\{\pi_1, \dots, \pi_m\} = \mathcal{P}os_c^{top}(t|_i)$. As $t|_i$ and $\mathfrak{a}^{\mathsf{i}}(t)|_i\theta$ only differ at the positions $\pi_1, \dots, \pi_m$, by the definition of $\mathfrak{a}^{\mathsf{i}}$, this implies

$$[\![\mathsf{sz}(\mathfrak{a}^{\mathsf{i}}(t)|_i\theta)]\!] \le [\![\mathsf{sz}(t|_i)]\!].$$

With monotonicity of rt, this implies ($\dagger\dagger$).

If $f \in \Sigma_d$, then we have

$$[\![\mathfrak{c}^{\mathsf{i}}(t) + \mathfrak{c}^{\circ}(\mathfrak{a}^{\mathsf{i}}(t))\theta]\!] = [\![\mathfrak{c}(t) + 0]\!] = [\![\mathfrak{c}(t)]\!].$$

□

Corollary 52. *Every runtime bound for $\mathcal{P}_{\mathsf{rt},\mathsf{sz}}$ is also a runtime bound for $\mathcal{P}'_{\mathsf{rt},\mathsf{sz}}$.*

Proof. Immediate consequence of Lemma 51. □

The following lemmas introduce alternative equivalent representations of $\mathfrak{c}$ and $\mathfrak{c}^{\mathsf{i}}$ as switching from one representation to the other is handy in later proofs.

Lemma 53 (Alternative Representation of $\mathfrak{c}$). *Let $\mathcal{P}$ be an RNTS with size and runtime bounds* sz *and* rt*. Then for every term $t \in \mathcal{T}$ with $\mathrm{root}(t) \in \Sigma_d$ we have $\mathfrak{c}(t) = \sum_{\pi \in \mathcal{P}os_c^{top}(t)} \mathfrak{c}(t|_\pi)$.*

Proof. Immediate consequence of the definition of $\mathfrak{c}$. □

Lemma 54 (Alternative Representation of $\mathfrak{c}^{\mathsf{i}}$). *Let $\mathcal{P}$ be an RNTS with size and runtime bounds* sz *and* rt*. Then for every term $t \in \mathcal{T}$ we have $\mathfrak{c}^{\mathsf{i}}(t) = \sum_{\pi \in \mathcal{P}os_c^{top}(t)} \mathfrak{c}(t|_\pi)$.*

Proof. Structural induction on t. If $t \in \mathcal{V}$, we have

$$\mathfrak{c}^{\mathsf{i}}(t) = 0 = \sum_{\pi \in \mathcal{P}os_c^{top}(t) = \varnothing} \mathfrak{c}(t|_\pi).$$

Let $t = f(t_1, \dots, t_k)$. If $f \notin \Sigma_d$, then we have

$$\begin{array}{ll} \quad \mathfrak{c}^{\mathsf{i}}(t) & \\ = \sum_{1 \le i \le k} \mathfrak{c}^{\mathsf{i}}(t_i) & \text{by def. of } \mathfrak{c}^{\mathsf{i}} \\ = \sum_{1 \le i \le k} \sum_{\pi \in \mathcal{P}os_c^{top}(t_i)} \mathfrak{c}(t_i|_\pi) & \text{by the induction hypothesis} \\ = \sum_{\pi \in \mathcal{P}os_c^{top}(t)} \mathfrak{c}(t|_\pi) & \text{as } f \notin \Sigma_d \end{array}$$

If $f \in \Sigma_d$, then we have

$$\begin{array}{ll} \quad \mathfrak{c}^{\mathsf{i}}(t) & \\ = \mathfrak{c}(t) & \text{by def. of } \mathfrak{c}^{\mathsf{i}} \\ = \sum_{\pi \in \mathcal{P}os_c^{top}(t)} \mathfrak{c}(t|_\pi) & \text{by Lemma 53} \end{array}$$

□

Similar to Lemma 50 for sz, the following lemma proves monotonicity of $\mathfrak{c}$. More precisely, replacing a subterm $q|_\pi$ of q with a "smaller" term decreases the cost of q weakly even if we re-add the costs of the replaced subterm $q|_\pi$.

Lemma 55 ($\mathfrak{c}$ is Monotonic). *Let $q \in \mathcal{T}, t \in \mathcal{T}(\Sigma_{\mathsf{exp}}, \varnothing), \pi \in \mathcal{P}os(q)$, and $[\![\mathsf{sz}(q|_\pi)]\!] \ge [\![\mathsf{sz}(t)]\!]$. Then $[\![\mathfrak{c}(q)]\!] \ge [\![\mathfrak{c}(q[t]_\pi)]\!] + [\![\mathfrak{c}(q|_\pi)]\!]$.*

Proof. Let t be an arbitrary ground term. We use induction on π. If $\pi = \varepsilon$, then we have

$$\begin{array}{lll} & [\![\mathfrak{c}(q)]\!] \ge [\![\mathfrak{c}(q[t]_\pi)]\!] + [\![\mathfrak{c}(q|_\pi)]\!] & \\ \iff & [\![\mathfrak{c}(q)]\!] \ge [\![\mathfrak{c}(t)]\!] + [\![\mathfrak{c}(q)]\!] & \text{as } \pi = \varepsilon \\ \iff & 0 \ge [\![\mathfrak{c}(t)]\!] & \\ \iff & \mathsf{true} & \text{as } t \in \mathcal{T}(\Sigma_{\mathsf{exp}}, \varnothing) \end{array}$$

If $\pi = i.\pi'$, then $q = f(q_1, \ldots, q_k)$. By the induction hypothesis, we have

$$[\![\mathfrak{c}(q_i)]\!] \geq [\![\mathfrak{c}(q_i[t]_{\pi'})]\!] + [\![\mathfrak{c}(q_i|_{\pi'})]\!]. \tag{27}$$

By Lemma 50, the assumption $[\![\mathsf{sz}(q|_\pi)]\!] = [\![\mathsf{sz}(q|_{i.\pi'})]\!] \geq [\![\mathsf{sz}(t)]\!]$ implies

$$[\![\mathsf{sz}(q_i)]\!] \geq [\![\mathsf{sz}(q_i[t]_{\pi'})]\!]. \tag{28}$$

If $f \in \Sigma_c$, we get:

$$\begin{array}{ll}
\quad [\![\mathfrak{c}(q)]\!] & \\
= [\![\mathsf{rt}(f)(\mathsf{sz}(q_1), \ldots, \mathsf{sz}(q_k)) + \mathfrak{c}(q_1) + \ldots + \mathfrak{c}(q_k)]\!] & \text{by def. of } \mathfrak{c} \\
= [\![\mathsf{rt}(f)(\mathsf{sz}(q_1), \ldots, \mathsf{sz}(q_k))]\!] + [\![\mathfrak{c}(q_1)]\!] + \ldots + [\![\mathfrak{c}(q_k)]\!] & \\
\geq [\![\mathsf{rt}(f)(\mathsf{sz}(q_1), \ldots, \mathsf{sz}(q_k))]\!] & \\
\quad + [\![\mathfrak{c}(q_1)]\!] + \ldots + [\![\mathfrak{c}(q_i[t]_{\pi'})]\!] + \ldots + [\![\mathfrak{c}(q_k)]\!] + [\![\mathfrak{c}(q_i|_{\pi'})]\!] & \text{by (27)} \\
= [\![\mathsf{rt}(f)(\mathsf{sz}(q_1), \ldots, \mathsf{sz}(q_k)) & \\
\quad + \mathfrak{c}(q_1) + \ldots + \mathfrak{c}(q_i[t]_{\pi'}) + \ldots + \mathfrak{c}(q_k)]\!] + [\![\mathfrak{c}(q_i|_{\pi'})]\!] & \\
\geq [\![\mathsf{rt}(f)(\mathsf{sz}(q_1), \ldots, \mathsf{sz}(q_i[t]_{\pi'}), \ldots \mathsf{sz}(q_k)) & \text{by (28) and} \\
\quad + \mathfrak{c}(q_1) + \ldots + \mathfrak{c}(q_i[t]_{\pi'}) + \ldots + \mathfrak{c}(q_k)]\!] + [\![\mathfrak{c}(q_i|_{\pi'})]\!] & \text{monotonicity of } \mathsf{rt}(f) \\
= [\![\mathsf{rt}(f)(\mathsf{sz}(q_1), \ldots, \mathsf{sz}(q[t]_\pi|_i), \ldots \mathsf{sz}(q_k)) & \text{as } \pi = i.\pi' \\
\quad + \mathfrak{c}(q_1) + \ldots + \mathfrak{c}(q[t]_\pi|_i) + \ldots + \mathfrak{c}(q_k)]\!] + [\![\mathfrak{c}(q_i|_{\pi'})]\!] & \\
= [\![\mathfrak{c}(q[t]_\pi)]\!] + [\![\mathfrak{c}(q_i|_{\pi'})]\!] & \text{by def. of } \mathfrak{c} \\
= [\![\mathfrak{c}(q[t]_\pi)]\!] + [\![\mathfrak{c}(q|_\pi)]\!] & \text{as } \pi = i.\pi'
\end{array}$$

Otherwise, we get:

$$\begin{array}{ll}
\quad [\![\mathfrak{c}(q)]\!] & \\
= [\![\mathfrak{c}(q_1) + \ldots + \mathfrak{c}(q_k)]\!] & \text{by def. of } \mathfrak{c} \\
= [\![\mathfrak{c}(q_1)]\!] + \ldots + [\![\mathfrak{c}(q_k)]\!] & \\
\geq [\![\mathfrak{c}(q_1)]\!] + \ldots + [\![\mathfrak{c}(q_i[t]_{\pi'})]\!] + \ldots + [\![\mathfrak{c}(q_k)]\!] + [\![\mathfrak{c}(q_i|_{\pi'})]\!] & \text{by (27)} \\
= [\![\mathfrak{c}(q_1) + \ldots + \mathfrak{c}(q_i[t]_{\pi'}) + \ldots + \mathfrak{c}(q_k)]\!] + [\![\mathfrak{c}(q_i|_{\pi'})]\!] & \\
= [\![\mathfrak{c}(q_1) + \ldots + \mathfrak{c}(q[t]_\pi|_i) + \ldots + \mathfrak{c}(q_k)]\!] + [\![\mathfrak{c}(q_i|_{\pi'})]\!] & \text{as } \pi = i.\pi' \\
= [\![\mathfrak{c}(q[t]_\pi)]\!] + [\![\mathfrak{c}(q_i|_{\pi'})]\!] & \text{by def. of } \mathfrak{c} \\
= [\![\mathfrak{c}(q[t]_\pi)]\!] + [\![\mathfrak{c}(q|_\pi)]\!] & \text{as } \pi = i.\pi'
\end{array}$$

$\square$

The following lemma proves the soundness of our definition of $\mathfrak{c}$. To this end, we show that the cost of rewriting with $\widehat{\mathcal{P}}$ (whose costs are determined by rt) is indeed bounded by $\mathfrak{c}$.

Lemma 56 (Soundness of $\mathfrak{c}$). *Let $\mathcal{P}$ be an RNTS with size and runtime bounds sz and rt and let q be a normal form of t w.r.t. $\widehat{\mathcal{P}}$, i.e., we have $t \xrightarrow{m}{}^{*}_{\widehat{\mathcal{P}}} q$. Then $[\![\mathfrak{c}(t)]\!] \geq m$.*

Proof. We use induction on the length n of the reduction $t \xrightarrow{m}{}^{n}_{\widehat{\mathcal{P}}} q$. If $n = 0$, we have $m = 0$ and hence the claim is trivial. If $n > 0$, we have $t \xrightarrow{m_1}_{\widehat{\mathcal{P}}} t' \xrightarrow{m_2}{}^{n-1}_{\widehat{\mathcal{P}}} q$ with $m = m_1 + m_2$ and, by the induction hypothesis,

$$[\![\mathfrak{c}(t')]\!] \geq m_2. \tag{29}$$

Let $\ell \to r\,[\varphi]$, σ, and π be the rule, the natural substitution, and the position of the rewrite step $t \xrightarrow{m_1}_{\widehat{\mathcal{P}}} t'$ and let $t|_\pi = f(t_1, \dots, t_k)$. By definition of $\widehat{\mathcal{P}}$, we have $f \in \Sigma_c$, $t_1, \dots, t_k \in \mathbb{N}$, $m_1 = [\![\mathsf{rt}(f)(t_1, \dots, t_k)]\!]$, $[\![r\sigma]\!] = e \in \mathbb{N}$, $t' = [\![t[e]_\pi]\!]$, and $[\![\mathsf{sz}(t|_\pi)]\!] \geq e$. By Lemma 55, we get

$$[\![\mathfrak{c}(t)]\!] \geq [\![\mathfrak{c}(t[e]_\pi)]\!] + [\![\mathfrak{c}(t|_\pi)]\!] = [\![\mathfrak{c}(t')]\!] + [\![\mathfrak{c}(t|_\pi)]\!]. \tag{30}$$

With (29), (30) implies

$$[\![\mathfrak{c}(t)]\!] \geq m_2 + [\![\mathfrak{c}(t|_\pi)]\!]. \tag{31}$$

As $t|_\pi = f(t_1, \dots, t_k)$ with $f \in \Sigma_c$ and $t_1, \dots, t_k \in \mathbb{N}$, we have

$$[\![\mathfrak{c}(t|_\pi)]\!] = [\![\mathsf{rt}(f)(t_1, \dots, t_k)]\!] \geq m_1 \tag{32}$$

by definition of $\mathfrak{c}$ and hence

$$[\![\mathfrak{c}(t)]\!] \geq m_2 + m_1 = m$$

by (31) and (32) $\square$

Lemma 56 shows that $\mathfrak{c}$ measures the costs of constructors correctly if they are evaluated with $\widehat{\mathcal{P}}$. However, Lemma 56 does not deal with evaluating defined symbols. The following lemma shows that $\mathfrak{c}$ remains an upper bound on the cost of possible $\widehat{\mathcal{P}}$-derivations if subterms are normalized w.r.t. $\mathcal{P} \cup \widehat{\mathcal{P}}$. So in particular, it also applies if rules of $\mathcal{P}$ are used to evaluate defined symbols.

Lemma 57 (Normalization Preserves $\mathfrak{c}$). *Let $\mathcal{P}$ be an RNTS with size bound* sz *and runtime bound* rt*, let t be a ground term with $t = [\![t]\!]$, let $\mathcal{P}os_d(t) = \{\pi_1, \dots, \pi_n\}$ be parallel positions, let $q_1, \dots, q_n$ be $\mathcal{P} \cup \widehat{\mathcal{P}}$-normal forms of $t|_{\pi_1}, \dots, t|_{\pi_n}$, respectively, and let s be a $\widehat{\mathcal{P}}$-normal form of $[\![t[q_1]_{\pi_1} \dots [q_n]_{\pi_n}]\!]$, i.e., we have $[\![t[q_1]_{\pi_1} \dots [q_n]_{\pi_n}]\!] \xrightarrow{m}{}^*_{\widehat{\mathcal{P}}} s$. Then $[\![\mathfrak{c}(t)]\!] \geq m$.*

Proof. We use structural induction on t. If $\mathrm{root}(t) \in \Sigma_d$, then $\mathcal{P}os_d(t) = \{\varepsilon\}$. So we have $t \to^*_{\mathcal{P} \cup \widehat{\mathcal{P}}} q \xrightarrow{m}{}^*_{\widehat{\mathcal{P}}} s$, but since q is already a normal form, we have $s = m$ and $m = 0$. Hence, the claim is trivial.

Now we consider $t = f(t_1, \dots, t_k)$ and $f \in \Sigma_{\mathsf{exp}}$. W.l.o.g., we assume that the positions $\pi_1, \dots, \pi_n$ are ordered lexicographically. Then each t_i either contains no defined symbols or there exist $1 \leq a_i \leq b_i \leq k$ such that $\pi_{a_i}, \dots, \pi_{b_i}$ are those positions from $\{\pi_1, \dots, \pi_n\}$ that are in the subterm t_i (i.e., these positions start with i). If t_i does not contain defined symbols, then we have $t_i = [\![t_i]\!] \xrightarrow{m_i}{}^*_{\widehat{\mathcal{P}}} s_i$ and by Lemma 56 we obtain $[\![\mathfrak{c}(t_i)]\!] \geq m_i$. If t_i contains defined symbols, then we have $t_i \to^*_{\mathcal{P} \cup \widehat{\mathcal{P}}} [\![t_i[q_{a_i}]_{\pi_{a_i}} \dots [q_{b_i}]_{\pi_{b_i}}]\!] \xrightarrow{m_i}{}^*_{\widehat{\mathcal{P}}} s_i$ and the induction hypothesis implies $[\![\mathfrak{c}(t_i)]\!] \geq m_i$. Thus, $s = [\![f(s_1, \dots, s_k)]\!]$ and $m = m_1 + \ldots + m_k$. Moreover, we get:

$$\begin{aligned}
&[\![\mathfrak{c}(t)]\!] \\
= {}&[\![\mathfrak{c}(f(t_1, \dots, t_k))]\!] \\
= {}&[\![\mathfrak{c}(t_1)]\!] + \ldots + [\![\mathfrak{c}(t_k)]\!] \text{ by def. of } \mathfrak{c} \\
\geq {}&m_1 + \ldots + m_k \\
= {}&m
\end{aligned}$$

Finally, let $t = f(t_1, \ldots, t_k)$ and $f \in \Sigma_c$. Again each t_i either contains no defined symbols or there exist $1 \leq a_i \leq b_i \leq k$ such that $\pi_{a_i}, \ldots, \pi_{b_i}$ are those positions from $\{\pi_1, \ldots, \pi_n\}$ that are in the subterm t_i. As in the previous case, if t_i does not contain defined symbols, then $t_i = [\![t_i]\!] \xrightarrow{m_i}{}^*_{\widehat{\mathcal{P}}} s_i$ and $[\![\mathfrak{c}(t_i)]\!] \geq m_i$ by Lemma 56. If t_i contains defined symbols, then $t_i \to^*_{\mathcal{P} \cup \widehat{\mathcal{P}}} [\![t_i[q_{a_i}]_{\pi_{a_i}} \ldots [q_{b_i}]_{\pi_{b_i}}]\!] \xrightarrow{m_i}{}^*_{\widehat{\mathcal{P}}} s_i$ and $[\![\mathfrak{c}(t_i)]\!] \geq m_i$ by the induction hypothesis. So our overall reduction has the form

$$t \to^*_{\mathcal{P} \cup \widehat{\mathcal{P}}} [\![t[q_1]_{\pi_1} \ldots [q_n]_{\pi_n}]\!] \xrightarrow{m_1 + \ldots + m_k}{}^*_{\widehat{\mathcal{P}}} f(s_1, \ldots, s_k) \xrightarrow{m'}{}^*_{\widehat{\mathcal{P}}} s,$$

where $m = m_1 + \ldots + m_k + m'$ and $[\![\mathfrak{c}(f(s_1, \ldots, s_k))]\!] \geq m'$ by Lemma 56. Moreover, we get:

$$\begin{array}{ll} & [\![\mathfrak{c}(t)]\!] \\ = & [\![\mathfrak{c}(f(t_1, \ldots, t_k))]\!] \\ = & [\![\mathsf{rt}(f)(\mathsf{sz}(t_1), \ldots, \mathsf{sz}(t_k))]\!] + [\![\mathfrak{c}(t_1)]\!] + \ldots + [\![\mathfrak{c}(t_k)]\!] \text{ by def. of } \mathfrak{c} \\ \geq & [\![\mathsf{rt}(f)(\mathsf{sz}(t_1), \ldots, \mathsf{sz}(t_k))]\!] + m_1 + \ldots + m_k \end{array}$$

If $\{s_1, \ldots, s_k\} \not\subset \mathbb{N}$, then $f(s_1, \ldots, s_k)$ is a $\widehat{\mathcal{P}}$-normal form and thus we have $m = m_1 + \ldots + m_k$. Now assume $\{s_1, \ldots, s_k\} \subset \mathbb{N}$. Then:

$$\begin{array}{lll} & [\![\mathsf{rt}(f)(\mathsf{sz}(t_1), \ldots, \mathsf{sz}(t_k))]\!] + m_1 + \ldots + m_k & \\ \geq & [\![\mathsf{rt}(f)(s_1, \ldots, s_k)]\!] + m_1 + \ldots + m_k & (\dagger) \\ = & [\![\mathfrak{c}(f(s_1, \ldots, s_k))]\!] + m_1 + \ldots + m_k & \\ \geq & m' + m_1 + \ldots + m_k & \\ = & m & \end{array}$$

The step (†) holds because $t_i \to^*_{\mathcal{P} \cup \widehat{\mathcal{P}}} s_i$ and sz is a size bound of $\mathcal{P}$ and hence also of $\mathcal{P} \cup \widehat{\mathcal{P}}$ by Lemma 48, and because of weak monotonicity of $\mathsf{rt}(f)$. ☐

B.2.2 Properties of $\widehat{\mathcal{P}}$

We use $\widehat{\mathcal{P}}$ to "summarize" a sub-RNTS $\mathcal{U}$ of a larger RNTS $\mathcal{Q}$ by the size and runtime bounds of $\mathcal{U}$. Hence, it is important that the RNTS that results from this summarization behaves similar to the original RNTS. The following two lemmas state that our summarization indeed preserves normal forms from $\mathbb{N}$ and over-approximates the costs of the original RNTS.

Lemma 58 (Replacing Rules With $\widehat{\mathcal{P}}$ Preserves Normal Forms from $\mathbb{N}$ and Costs). *Let $\mathcal{Q} = \mathcal{P} \cup \mathcal{U}$ be an RNTS with size and runtime bounds* sz *and* rt. *If $t \in \mathcal{T}(\Sigma_c^{\mathcal{P}} \cup \Sigma_{\mathit{exp}}, \varnothing)$ and $t \xrightarrow{m}{}^*_{\mathcal{Q}} n \in \mathbb{N}$, then $t \xrightarrow{e}{}^*_{\widehat{\mathcal{P}}} n$ for some $e \geq m$.*

Proof. We use structural induction on $t = f(t_1, \ldots, t_k)$. Let $n_1, \ldots, n_k$ be the normal forms obtained for $t_1, \ldots, t_k$ in the rewrite sequence $f(t_1, \ldots, t_k) \xrightarrow{m}{}^*_{\mathcal{Q}} n$

and let $m_1, \dots, m_k$ be the costs of normalizing $t_1, \dots, t_k$. First assume $f \in \Sigma_{\mathsf{exp}}$, i.e., we have

$$f(t_1, \dots, t_k) \xrightarrow{m_1+\ldots+m_k}{}^{*}_{\mathcal{Q}} [\![f(n_1, \dots, n_k)]\!] = n \text{ and} \tag{33}$$
$$m_1 + \ldots + m_k = m. \tag{34}$$

By the induction hypothesis, we have

$$f(t_1, \dots, t_k) \xrightarrow{e_1+\ldots+e_k}{}^{*}_{\widehat{\mathcal{P}}} [\![f(n_1, \dots, n_k)]\!] \text{ with } e_i \geq m_i \text{ for all } 1 \leq i \leq k \tag{35}$$

and hence $e_1 + \ldots + e_k \geq m$ by (34).

Now assume $f \in \Sigma_c$, i.e., we have

$$f(t_1, \dots, t_k) \xrightarrow{m_1+\ldots+m_k}{}^{*}_{\mathcal{Q}} [\![f(n_1, \dots, n_k)]\!] \xrightarrow{m'}{}^{*}_{\mathcal{Q}} n \text{ and} \tag{36}$$
$$m_1 + \ldots + m_k + m' = m. \tag{37}$$

By the induction hypothesis, we again have (35). Moreover, we have

$$[\![f(n_1, \dots, n_k)]\!] = f(n_1, \dots, n_k) \qquad \text{and} \tag{38}$$
$$f(x_1, \dots, x_k) \xrightarrow{\mathsf{rt}(f)} x \quad [x \leq \mathsf{sz}(f)] \in \widehat{\mathcal{P}} \tag{39}$$

by definition of $[\![\cdot]\!]$ and $\widehat{\mathcal{P}}$. As sz is a size bound for $\mathcal{Q}$, (36) implies

$$[\![\mathsf{sz}(f)(n_1, \dots, n_k)]\!] \geq n. \tag{40}$$

Hence, (38), (39), and (40) imply

$$[\![f(n_1, \dots, n_k)]\!] \xrightarrow{[\![\mathsf{rt}(f)(n_1,\ldots,n_k)]\!]}_{\widehat{\mathcal{P}}} n. \tag{41}$$

Thus, we have

$$f(t_1, \dots, t_k) \xrightarrow{e_1+\ldots+e_k}{}^{*}_{\widehat{\mathcal{P}}} [\![f(n_1, \dots, n_k)]\!] \xrightarrow{[\![\mathsf{rt}(f)(n_1,\ldots,n_k)]\!]}_{\widehat{\mathcal{P}}} n$$

by (35) and (41). As rt is a runtime bound for $\mathcal{Q}$, (36) implies

$$[\![\mathsf{rt}(f)(n_1, \dots, n_k)]\!] \geq m'. \tag{42}$$

Finally, (37), (35), and (42) imply $e_1 + \ldots + e_k + [\![\mathsf{rt}(f)(n_1, \dots, n_k)]\!] \geq m$. □

Lemma 59 (Replacing Rules With $\widehat{\mathcal{P}}$ Preserves Costs for Arbitrary Normal Forms). *Let $\mathcal{Q} = \mathcal{P} \cup \mathcal{U}$ be an RNTS with size and runtime bounds* sz *and* rt. *If $t \in \mathcal{T}(\Sigma_c^{\mathcal{P}} \cup \Sigma_{\mathsf{exp}}, \varnothing)$, $t \xrightarrow{m}{}^{*}_{\mathcal{Q}} q$, and q is a $\mathcal{Q}$-normal form, then $t \xrightarrow{e}{}^{*}_{\widehat{\mathcal{P}}} n$ for some $n \in \mathbb{N}$ and $e \geq m$.*

Proof. We use structural induction on $t = f(t_1, \dots, t_k)$. Let $q_1, \dots, q_k$ be the $\mathcal{Q}$-normal forms of $t_1, \dots, t_k$ obtained in the rewrite sequence $f(t_1, \dots, t_k) \xrightarrow{m}{}^{*}_{\mathcal{Q}} q$

and let $m_1, \ldots, m_k$ be the costs of reducing t_i to q_i for each $1 \leq i \leq k$. First assume $f \in \Sigma_{\mathsf{exp}}$, i.e., we have

$$f(t_1, \ldots, t_k) \xrightarrow{m_1+\ldots+m_k}{}^{*}_{\mathcal{Q}} [\![f(q_1, \ldots, q_k)]\!] = q \quad \text{and} \tag{43}$$

$$m = m_1 + \ldots + m_k. \tag{44}$$

By the induction hypothesis, we have

$$t_i \xrightarrow{e_i}_{\widehat{\mathcal{P}}} n_i \text{ with } e_i \geq m_i \text{ and } n_i \in \mathbb{N} \text{ for all } 1 \leq i \leq k. \tag{45}$$

With (44), we get $e_1 + \ldots + e_k \geq m$. Furthermore, we have $[\![f(n_1, \ldots, n_k)]\!] \in \mathbb{N}$ by definition of $[\![\cdot]\!]$.

Now assume $f \in \Sigma_c$, i.e., we have

$$f(t_1, \ldots, t_k) \xrightarrow{m_1+\ldots+m_k}{}^{*}_{\mathcal{Q}} f(q_1, \ldots, q_k) \xrightarrow{m'}{}^{*}_{\mathcal{Q}} q \quad \text{and} \tag{46}$$

$$m = m_1 + \ldots + m_k + m'. \tag{47}$$

By the induction hypothesis, we again get (45). Moreover, we have

$$f(x_1, \ldots, x_k) \xrightarrow{\mathsf{rt}(f)} x \quad [x \leq \mathsf{sz}(f)] \in \widehat{\mathcal{P}} \tag{48}$$

by definition of $\widehat{\mathcal{P}}$. If $f(q_1, \ldots, q_k)$ is a $\mathcal{Q}$-normal form, then we have $m' = 0$ and thus $e_1 + \ldots + e_k \geq m$ by (45) and (47). Moreover we have:

$$\begin{array}{lll} & f(t_1, \ldots, t_k) & \\ \xrightarrow{e_1+\ldots+e_k}{}^{*}_{\widehat{\mathcal{P}}} & f(n_1, \ldots, n_k) & \text{by (45)} \\ \xrightarrow{\mathsf{rt}(f)(n_1,\ldots,n_k)}_{\widehat{\mathcal{P}}} & 0 & \text{by (48)} \end{array}$$

If $f(q_1, \ldots, q_k)$ is not a normal form, then we have $q_1, \ldots, q_k \in \mathbb{N}$ by definition of $\to_{\mathcal{Q}}$. Hence, by Lemma 58, we have

$$t_i \xrightarrow{e_i}_{\widehat{\mathcal{P}}} q_i \text{ with } e_i \geq m_i \text{ for each } 1 \leq i \leq k. \tag{49}$$

Thus, we get:

$$\begin{array}{lll} & f(t_1, \ldots, t_k) & \\ \xrightarrow{e_1+\ldots+e_k}{}^{*}_{\widehat{\mathcal{P}}} & f(q_1, \ldots, q_k) & \text{by (49)} \\ \xrightarrow{[\![\mathsf{rt}(f)(q_1,\ldots,q_k)]\!]}_{\widehat{\mathcal{P}}} & 0 & \text{by (48)} \end{array}$$

Moreover, we have $[\![\mathsf{rt}(f)(q_1, \ldots, q_k)]\!] \geq m'$, as rt is a runtime bound for $\mathcal{Q}$. Thus, we have $e_1 + \ldots + e_k + [\![\mathsf{rt}(f)(q_1, \ldots, q_k)]\!] \geq m$ by (47). $\square$

The following lemma allows us to replace subterms where the evaluation "got stuck" by other terms.

Lemma 60 (Replacing Subterms in Reductions). *If $[\![s]\!] \xrightarrow{m}{}^{k}_{\mathcal{P}} [\![t]\!]$ and $[\![s|_\pi]\!] \notin \mathbb{N}$ is a $\mathcal{P}$-normal form, then for all $q \in \mathcal{T}$ we have $[\![s[q]_\pi]\!] \xrightarrow{m}{}^{k}_{\mathcal{P}} [\![t[q]_\pi]\!]$.*

Proof. We use induction on the length of the reduction k. If $k = 0$, then the claim is trivial. Let $k > 0$, i.e., we have $[\![s]\!] \xrightarrow{m_1}_{\mathcal{P}} [\![s']\!] \xrightarrow{m_2}{}^*_{\mathcal{P}} [\![t]\!]$ with $m = m_1 + m_2$. Let $\ell \to r\,[\varphi]$, σ, and κ be the rule, the natural substitution, and the position used for the rewrite step $[\![s]\!] \xrightarrow{w_1}_{\mathcal{P}} [\![s']\!]$. Since $[\![s|_\pi]\!] \notin \mathbb{N}$ is in $\mathcal{P}$-normal form and Σ_{exp}-symbols above π cannot be evaluated, κ and π are parallel. Hence we have:

$$\begin{array}{lll} & [\![s[q]_\pi]\!] & \\ = & [\![s[q]_\pi]\!][\ell\sigma]_\kappa & \\ \xrightarrow{m_1}_{\mathcal{P}} & [\![[\![s[q]_\pi]\!][r\sigma]_\kappa]\!] & \\ = & [\![[\![s[r\sigma]_\kappa]\!][q]_\pi]\!] & \text{as } \pi \text{ and } \kappa \text{ are parallel and } (\dagger) \\ = & [\![s'[q]_\pi]\!] & \\ \xrightarrow{m_2}{}^{k-1}_{\mathcal{P}} & [\![t[q]_\pi]\!] & \text{by the induction hypothesis} \end{array}$$

For the step marked with $(\dagger)$, note that Σ_{exp}-symbols above π resp. κ cannot be evaluated by $[\![\cdot]\!]$ as $[\![s|_\pi]\!] \notin \mathbb{N}$ resp. $[\![s|_\kappa]\!] \notin \mathbb{N}$. $\square$

Building upon the previous two lemmas, the following theorem shows that replacing a larger RNTS $\mathcal{Q}$ by a subsystem $\mathcal{P}$ and the rules $\widehat{\mathcal{P}}$ which summarize the function symbols whose rules have been removed is indeed sound for size as well as runtime bounds. In the following, for any RNTS $\mathcal{U}$ let $\Sigma^{\mathcal{U}}$ consist of all function symbols occurring $\mathcal{U}$ except the ones from Σ_{fml}.

Theorem 61 (Approximating Removed Rules by $\widehat{\mathcal{P}}$). *Let $\mathcal{Q} = \mathcal{P} \cup \mathcal{U}$ be an RNTS with size and runtime bounds* sz *and* rt *such that $\Sigma_d^{\mathcal{P}} \cap \Sigma^{\mathcal{U}} = \varnothing$. Then every size bound* sz' *and every runtime bound* rt' *for $\mathcal{P} \cup \widehat{\mathcal{P}}$ is also a size resp. runtime bound for $\mathcal{Q}$. Here, we assume $\mathsf{sz}'(f) = \mathsf{rt}'(f) = \omega$ for all symbols f from Σ that do not occur in $\mathcal{P}$.*

Proof. To prove that every size bound sz' for $\mathcal{P} \cup \widehat{\mathcal{P}}$ is also a size bound for $\mathcal{Q}$, let t be a nat-basic term such that $t \xrightarrow{m}{}^*_{\mathcal{Q}} n \in \mathbb{N}$ where $\mathrm{root}(t)$ occurs in $\mathcal{P}$. We assume a rewrite strategy where $\mathcal{U}$-rules are applied with a higher priority than $\mathcal{P}$-rules (i.e., $\mathcal{P}$-rules are only applied to $\mathcal{U}$-normal forms). This assumption can be made without loss of generality, because $\mathcal{P}$ and $\mathcal{U}$ share no defined symbols and the variables in rules of RNTSs can only be instantiated by numbers (i.e., every $\mathcal{U}$ redex is in $\mathcal{P}$-normal form and vice versa). Hence, the rewrite sequence $t \xrightarrow{m}{}^*_{\mathcal{Q}} n$ has the form

$$t = t_0 \xrightarrow{e_1}{}^*_{\mathcal{U}} s_1 \xrightarrow{m_1}{}^*_{\mathcal{P}} t_1 \xrightarrow{e_2}{}^*_{\mathcal{U}} \ldots \xrightarrow{e_k}{}^*_{\mathcal{U}} s_k \xrightarrow{m_k}{}^*_{\mathcal{P}} t_k = n \tag{50}$$

where each s_i is in $\mathcal{U}$-normal form. Note that we have

$$t_0, s_1, t_1, \ldots, s_k, t_k \in \mathcal{T}(\Sigma_d^{\mathcal{P}} \cup \left(\Sigma_c^{\mathcal{P}} \cap \Sigma_d^{\mathcal{U}}\right) \cup \Sigma_{\mathsf{exp}}, \varnothing). \tag{51}$$

To see why symbols from $\Sigma_c^{\mathcal{U}} \cup \left(\Sigma_c^{\mathcal{P}} \setminus \Sigma_d^{\mathcal{U}}\right)$ cannot occur, recall that we have

$$\Sigma_d^{\mathcal{P}} \cap \Sigma^{\mathcal{U}} = \varnothing \tag{52}$$

$$\Sigma_d^{\mathcal{Q}} = \Sigma_d^{\mathcal{P}} \cup \Sigma_d^{\mathcal{U}} \tag{53}$$

$$\Sigma_d^{\mathcal{U}} \cap \Sigma_c^{\mathcal{U}} = \varnothing \text{ and } \Sigma_d^{\mathcal{P}} \cap \Sigma_c^{\mathcal{P}} = \varnothing \tag{54}$$

and hence

$$
\begin{aligned}
& \Sigma_d^{\mathcal{Q}} \cap \Sigma_c^{\mathcal{U}} \\
={}& \left(\Sigma_d^{\mathcal{P}} \cup \Sigma_d^{\mathcal{U}}\right) \cap \Sigma_c^{\mathcal{U}} && \text{by (53)} \\
={}& \left(\Sigma_d^{\mathcal{P}} \cap \Sigma_c^{\mathcal{U}}\right) \cup \left(\Sigma_d^{\mathcal{U}} \cap \Sigma_c^{\mathcal{U}}\right) \\
={}& \Sigma_d^{\mathcal{U}} \cap \Sigma_c^{\mathcal{U}} && \text{by (52), as } \Sigma_c^{\mathcal{U}} \subseteq \Sigma^{\mathcal{U}} \\
={}& \varnothing && \text{by (54)}
\end{aligned} \tag{55}
$$

Moreover, we have

$$
\begin{aligned}
& \Sigma_d^{\mathcal{Q}} \cap (\Sigma_c^{\mathcal{P}} \setminus \Sigma_d^{\mathcal{U}}) \\
={}& (\Sigma_d^{\mathcal{Q}} \cap \Sigma_c^{\mathcal{P}}) \setminus \Sigma_d^{\mathcal{U}} \\
={}& \left((\Sigma_d^{\mathcal{P}} \cup \Sigma_d^{\mathcal{U}}) \cap \Sigma_c^{\mathcal{P}}\right) \setminus \Sigma_d^{\mathcal{U}} && \text{by (53)} \\
={}& \left(\left(\Sigma_d^{\mathcal{P}} \cap \Sigma_c^{\mathcal{P}}\right) \cup \left(\Sigma_d^{\mathcal{U}} \cap \Sigma_c^{\mathcal{P}}\right)\right) \setminus \Sigma_d^{\mathcal{U}} \\
={}& \left(\Sigma_d^{\mathcal{U}} \cap \Sigma_c^{\mathcal{P}}\right) \setminus \Sigma_d^{\mathcal{U}} && \text{by (54)} \\
={}& \varnothing
\end{aligned} \tag{56}
$$

Hence, by (55) and (56), symbols from $\Sigma_c^{\mathcal{U}} \cup (\Sigma_c^{\mathcal{P}} \setminus \Sigma_d^{\mathcal{U}})$ cannot be reduced using $\mathcal{Q}$ and thus every term containing symbols from $\Sigma_c^{\mathcal{U}} \cup (\Sigma_c^{\mathcal{P}} \setminus \Sigma_d^{\mathcal{U}})$ cannot be reduced to a natural number. With (50), this implies (51).

For any $0 \leq i \leq k-1$, if the length of the $\mathcal{U}$-reduction of t_i is not 0, there exist positions π such that

(A) $\mathrm{root}(t_i|_\pi) \in \Sigma_d^{\mathcal{U}}$,
(B) $t_i|_\pi \in \mathcal{T}(\Sigma_d^{\mathcal{U}} \cup \Sigma_{\mathsf{exp}}, \varnothing)$ and hence $t_i|_\pi \in \mathcal{T}(\Sigma_c^{\mathcal{P}} \cup \Sigma_{\mathsf{exp}}, \varnothing)$ due to (51), and
(C) there is no proper prefix κ of π that satisfies (A) and (B)

Let $n' \in \mathbb{N}$ be the normal form obtained for $t_i|_\pi$ in the rewrite sequence $t \to_{\mathcal{Q}}^* n$. Note that we have $n' \in \mathbb{N}$ as otherwise t_i and hence t could not be reduced to a natural number. With (B) and (52), we get $t_i|_\pi \to_{\mathcal{U}}^* n'$. Hence, we can apply Lemma 58 to $t_i|_\pi$ for each position π satisfying (A), (B), and (C). Thus, we get $t_0 \xrightarrow{e'_1}{}^*_{\widehat{\mathcal{P}}} s_1 \xrightarrow{m_1}{}^*_{\mathcal{P}} t_1 \xrightarrow{e'_2}{}^*_{\widehat{\mathcal{P}}} \dots \xrightarrow{e'_k}{}^*_{\widehat{\mathcal{P}}} s_k \xrightarrow{m_k}{}^*_{\mathcal{P}} t_k = n$ where $e'_i \geq e_i$ for all $0 \leq i \leq k-1$. Hence, if sz' is a size bound for $\mathcal{P} \cup \widehat{\mathcal{P}}$, then it is also a size bound for $\mathcal{Q}$.

To prove that every runtime bound rt' for $\mathcal{P} \cup \widehat{\mathcal{P}}$ is also a runtime bound for $\mathcal{Q}$, we additionally have to consider the case that we obtain a normal form $q \notin \mathbb{N}$. This time, let $t \in \mathcal{T}(\Sigma_d^{\mathcal{P}} \cup \left(\Sigma_c^{\mathcal{P}} \cap \Sigma_d^{\mathcal{U}}\right) \cup \{\square\} \cup \Sigma_{\mathsf{exp}}, \varnothing)$ be an arbitrary ground term, i.e., t does not have to be nat-basic. Here, $\square$ is a fresh constant. (This is needed for the generalized statement that we need to prove the claim.) Again, the rewrite sequence $t \xrightarrow{m}{}^*_{\mathcal{Q}} q$ has the form

$$t = t_0 \xrightarrow{e_1}{}^*_{\mathcal{U}} s_1 \xrightarrow{m_1}{}^*_{\mathcal{P}} \dots \xrightarrow{e_k}{}^*_{\mathcal{U}} s_k \xrightarrow{m_k}{}^*_{\mathcal{P}} t_k = q$$

where each s_i is in $\mathcal{U}$-normal form. We prove the following statement by induction on k. This suffices to show that every runtime bound rt' for $\mathcal{P} \cup \widehat{\mathcal{P}}$ is also a runtime bound for $\mathcal{Q}$.

$$\begin{array}{l} \text{For all } t_0, s_1, t_1, \ldots, t_k, s_k \in \mathcal{T}(\Sigma_d^{\mathcal{P}} \cup (\Sigma_c^{\mathcal{P}} \cap \Sigma_d^{\mathcal{U}}) \cup \{\square\} \cup \Sigma_{\mathsf{exp}}, \varnothing) \\ \text{where } t_0 = [\![t_0]\!] \text{ and each } s_i \text{ is in } \mathcal{U}\text{-normal form} \\ t_0 \xrightarrow{e_1}{}^*_{\mathcal{U}} s_1 \xrightarrow{m_1}{}^*_{\mathcal{P}} \ldots \xrightarrow{e_k}{}^*_{\mathcal{U}} s_k \xrightarrow{m_k}{}^*_{\mathcal{P}} t_k \text{ implies} \\ t_0 \xrightarrow{e}{}^*_{\mathcal{P} \cup \widehat{\mathcal{P}}} q' \text{ for some } q' \in \mathcal{T}, e \geq e_1 + m_1 + \ldots + e_k + m_k \end{array} \tag{57}$$

If $k = 0$, the claim is trivial. Let $k > 0$. Then we have

$$\begin{array}{rl} & t_0 \\ \xrightarrow{d_1+\ldots+d_n}{}^*_{\mathcal{U}} & s_1 \\ = & [\![t_0[q_1]_{\pi_1} \ldots [q_n]_{\pi_n}]\!] \end{array}$$

where $\pi_1, \ldots, \pi_n$ are the positions satisfying (A), (B), and (C) for the term t_0, q_i is the $\mathcal{U}$-normal form of $t_0|_{\pi_i}$ obtained in the rewrite sequence $t_0 \to^*_{\mathcal{U}} s_1$, d_i is the cost of reducing $t_0|_{\pi_i}$ to q_i, and $d_1 + \ldots + d_n = e_1$.

W.l.o.g, assume $q_1, \ldots, q_c \in \mathbb{N}$ and $q_{c+1}, \ldots, q_n \notin \mathbb{N}$ for some $1 \leq c \leq n$. Then, due to (B) and (52), each $\mathcal{U}$-normal form $q_{c+1}, \ldots, q_n$ contains at least one $\Sigma^{\mathcal{U}}$-symbol and no $\Sigma_d^{\mathcal{P}}$-symbol. Hence, we have $s_1|_{\pi_i} = t_1|_{\pi_i} = \ldots = s_k|_{\pi_i} = t_k|_{\pi_i} = q_i$ for each $c < i \leq n$ as Σ_{exp}-symbols above π_i can never be evaluated by $[\![\cdot]\!]$ and no rules are applicable above π_i (as q_i contains $\Sigma^{\mathcal{U}}$-symbols). Hence, we have

$$\begin{array}{rl} & t_0 \\ \xrightarrow{e_1}{}^*_{\mathcal{U}} & s_1[q_{c+1}]_{\pi_{c+1}} \ldots [q_n]_{\pi_n} \\ \xrightarrow{m_1}{}^*_{\mathcal{P}} & t_1[q_{c+1}]_{\pi_{c+1}} \ldots [q_n]_{\pi_n} \\ \xrightarrow{e_2}{}^*_{\mathcal{U}} & \ldots \\ \xrightarrow{e_k}{}^*_{\mathcal{U}} & s_k[q_{c+1}]_{\pi_{c+1}} \ldots [q_n]_{\pi_n} \\ \xrightarrow{m_k}{}^*_{\mathcal{P}} & t_k[q_{c+1}]_{\pi_{c+1}} \ldots [q_n]_{\pi_n} \end{array}$$

With the definition of $\to_{\mathcal{U}}$ and $\to_{\mathcal{P}}$ and $t_0 = [\![t_0]\!]$ this implies

$$\begin{array}{rl} & t_0 \\ \xrightarrow{e_1}{}^*_{\mathcal{U}} & [\![s_1[q_{c+1}]_{\pi_{c+1}} \ldots [q_n]_{\pi_n}]\!] \\ \xrightarrow{m_1}{}^*_{\mathcal{P}} & [\![t_1[q_{c+1}]_{\pi_{c+1}} \ldots [q_n]_{\pi_n}]\!] \\ \xrightarrow{e_2}{}^*_{\mathcal{U}} & \ldots \\ \xrightarrow{e_k}{}^*_{\mathcal{U}} & [\![s_k[q_{c+1}]_{\pi_{c+1}} \ldots [q_n]_{\pi_n}]\!] \\ \xrightarrow{m_k}{}^*_{\mathcal{P}} & [\![t_k[q_{c+1}]_{\pi_{c+1}} \ldots [q_n]_{\pi_n}]\!] \end{array}$$

By applying Lemma 60, we obtain[8]

$$\begin{array}{rl} & [\![t_1[\square]_{\pi_{c+1}} \ldots [\square]_{\pi_n}]\!] \\ \xrightarrow{e_2}{}^*_{\mathcal{U}} & \ldots \\ \xrightarrow{e_k}{}^*_{\mathcal{U}} & [\![s_k[\square]_{\pi_{c+1}} \ldots [\square]_{\pi_n}]\!] \\ \xrightarrow{m_k}{}^*_{\mathcal{P}} & [\![t_k[\square]_{\pi_{c+1}} \ldots [\square]_{\pi_n}]\!] \end{array} \tag{58}$$

[8] See Footnote 9 for an explanation why this replacement by $\square$ is needed.

By the induction hypothesis, (58) implies

$$[\![t_1[\Box]_{\pi_{c+1}} \dots [\Box]_{\pi_n}]\!] \xrightarrow{e}{}^{*}_{\mathcal{P}\cup\widehat{\mathcal{P}}} q' \text{with } e \geq e_2 + m_2 + \dots + e_k + m_k \tag{59}$$

By applying Lemma 58 to $t_0|_{\pi_i}$ for $1 \leq i \leq c$ (which is applicable due to (B)), we get

$$t_0|_{\pi_i} \xrightarrow{d'_i}{}_{\widehat{\mathcal{P}}} q_i \text{ with } d'_i \geq d_i. \tag{60}$$

By applying Lemma 59 to $t_0|_{\pi_i}$ for $c < i \leq n$ (which is applicable due to (B)), we get

$$t_0|_{\pi_i} \xrightarrow{d'_i}{}_{\widehat{\mathcal{P}}} b_i \text{ with } d'_i \geq d_i \text{ and } b_i \in \mathbb{N} \tag{61}$$

Hence, we have

$$\begin{array}{ll} & t_0 \\ \xrightarrow{d'_1+\dots+d'_k}{}_{\widehat{\mathcal{P}}} & [\![t_0[q_1]_{\pi_1} \dots [q_c]_{\pi_c}[b_{c+1}]_{\pi_{c+1}} \dots [b_n]_{\pi_n}]\!] \qquad \text{by (60) and (61)} \\ = & [\![t_0[q_1]_{\pi_1} \dots [q_n]_{\pi_n}[b_{c+1}]_{\pi_{c+1}} \dots [b_n]_{\pi_n}]\!] \\ = & [\![[\![t_0[q_1]_{\pi_1} \dots [q_n]_{\pi_n}]\!][b_{c+1}]_{\pi_{c+1}} \dots [b_n]_{\pi_n}]\!] \ (\dagger) \\ = & [\![s_1[b_{c+1}]_{\pi_{c+1}} \dots [b_n]_{\pi_n}]\!] \end{array} \tag{62}$$

The step marked with (†) holds as $q_{c+1}, \dots, q_n$ contain $\Sigma^{\mathcal{U}}$-symbols and hence Σ_{exp}-symbols above $\pi_{c+1}, \dots, \pi_n$ cannot be evaluated by $[\![\cdot]\!]$.

By applying Lemma 60 to the rewrite step

$$[\![s_1[q_{c+1}]_{\pi_{c+1}} \dots [q_n]_{\pi_n}]\!] \xrightarrow{m_1}{}^{*}_{\mathcal{P}} [\![t_1[q_{c+1}]_{\pi_{c+1}} \dots [q_n]_{\pi_n}]\!]$$

we get

$$[\![s_1[b_{c+1}]_{\pi_{c+1}} \dots [b_n]_{\pi_n}]\!] \xrightarrow{m_1}{}^{*}_{\mathcal{P}} [\![t_1[b_{c+1}]_{\pi_{c+1}} \dots [b_n]_{\pi_n}]\!] \tag{63}$$

Moreover, by applying Lemma 60 to (59), we obtain[9]

$$[\![t_1[b_{c+1}]_{\pi_{c+1}} \dots [b_n]_{\pi_n}]\!] \xrightarrow{e}{}^{*}_{\mathcal{P}\cup\widehat{\mathcal{P}}} [\![q'[b_{c+1}]_{\pi_{c+1}} \dots [b_n]_{\pi_n}]\!] \tag{64}$$

Thus, we have

$$\begin{array}{ll} & t_0 \\ \xrightarrow{d'_1+\dots+d'_k}{}_{\widehat{\mathcal{P}}} & [\![s_1[b_{c+1}]_{\pi_{c+1}} \dots [b_n]_{\pi_n}]\!] \text{ by (62)} \\ \xrightarrow{m_1}{}^{*}_{\mathcal{P}} & [\![t_1[b_{c+1}]_{\pi_{c+1}} \dots [b_n]_{\pi_n}]\!] \text{ by (63)} \\ \xrightarrow{e}{}^{*}_{\mathcal{P}\cup\widehat{\mathcal{P}}} & [\![q'[b_{c+1}]_{\pi_{c+1}} \dots [b_n]_{\pi_n}]\!] \text{ by (64)} \end{array}$$

I remains to show $d'_1 + \dots + d'_k + m_1 + e \geq e_1 + m_1 + \dots + e_k + m_k$. By (60) and (61), we have $d'_1 + \dots + d'_k \geq d_1 + \dots d_k = e_1$. Hence, it suffices to show $e \geq e_2 + m_2 + \dots + e_k + m_k$, which follows by (59). □

[9] Here, one can see why we need the construction with □. Otherwise, Lemma 60 would not apply here, as $q_{c+1}, \dots, q_n$ are in $\mathcal{U}$-normal form, but not necessarily in $\mathcal{P} \cup \widehat{\mathcal{P}}$-normal form.

The outer abstraction eliminates constructors above defined symbols. This elimination should be over-approximating the *size* of the result in the sense that reducing the resulting term does not yield smaller results than reducing the original term. The following lemma shows that $\mathfrak{a}^{\circ}$ indeed has this property.

Lemma 62 (Soundness of Outer Abstraction for Size). *Let $\mathcal{P}$ be an RNTS with size bound sz. If $[\![s]\!] \to_{\widehat{\mathcal{P}}}^{*} n \in \mathbb{N}$, then $[\![\mathfrak{a}^{\circ}(s)]\!] \to_{\widehat{\mathcal{P}}}^{*} n'$ with $n' \geq n$.*

Proof. We use structural induction on s. If $s \in \mathcal{T}(\Sigma_{\mathsf{exp}}, \varnothing)$ then the claim is trivial. Otherwise, we have $s = f(s_1, \ldots, s_k)$ and $[\![s]\!] = f([\![s_1]\!], \ldots, [\![s_k]\!])$. Let $n_1, \ldots, n_k$ be the normal forms of $[\![s_1]\!], \ldots, [\![s_k]\!]$ obtained in the rewrite sequence $f(s_1, \ldots, s_k) \to_{\widehat{\mathcal{P}}}^{*} n$, i.e., we have

$$[\![s]\!] = f([\![s_1]\!], \ldots, [\![s_k]\!]) \to_{\widehat{\mathcal{P}}}^{*} [\![f(n_1, \ldots, n_k)]\!] \to_{\widehat{\mathcal{P}}}^{*} n. \tag{65}$$

Clearly, $n \in \mathbb{N}$ implies $n_1, \ldots, n_k \in \mathbb{N}$ by definition of $\to_{\widehat{\mathcal{P}}}$. By the induction hypothesis, we have

$$[\![\mathfrak{a}^{\circ}(s_i)]\!] \to_{\widehat{\mathcal{P}}}^{*} n_i' \text{ with } n_i' \geq n_i \text{ for each } 1 \leq i \leq k. \tag{66}$$

If $f \in \Sigma_{\mathsf{exp}}$, then we have $[\![f(n_1, \ldots, n_k)]\!] = n$ and hence $[\![f(n_1', \ldots, n_k')]\!] \geq n$ by (66) and monotonicity of $f \in \Sigma_{\mathsf{exp}}$.

If $f \in \Sigma_c$, then we have $f(n_1', \ldots, n_k') = [\![f(n_1', \ldots, n_k')]\!]$ and

$$f(x_1, \ldots, x_k) \xrightarrow{w} x\, [x \leq \mathsf{sz}(f)] \in \widehat{\mathcal{P}}$$

for some w, by definition of $\widehat{\mathcal{P}}$. Hence, we get

$$[\![f(n_1', \ldots, n_k')]\!] \to_{\widehat{\mathcal{P}}} \mathsf{sz}(f)(n_1', \ldots, n_k').$$

As $[\![f(n_1, \ldots, n_k)]\!] = f(n_1, \ldots, n_k) \to_{\widehat{\mathcal{P}}}^{*} n$ by (65) and sz is a size bound for $\mathcal{P}$ and thus also for $\widehat{\mathcal{P}}$ by Lemma 48, we have $\mathsf{sz}(f)(n_1, \ldots, n_k) \geq n$. By monotonicity of sz and (66), this implies $\mathsf{sz}(f)(n_1', \ldots, n_k') \geq n$.

Note that $f \in \Sigma_d$ is not possible, since a term containing defined symbols cannot be reduced to a natural number using $\widehat{\mathcal{P}}$. □

B.2.3 Properties of $\mathcal{P}^i$

The following lemma shows the soundness of our inner abstraction from Def. 25 for ground terms whose only defined symbol is at the root position.

Lemma 63 (Soundness of Inner Abstraction). *Let $\mathcal{P}$ be an RNTS with size bound sz, let t be a ground term whose only defined symbol is at position ε, and let q be a normal form of t w.r.t. $\widehat{\mathcal{P}}$. Then there is a substitution θ such that $[\![\mathfrak{a}^{\mathsf{i}}(t)\theta]\!] = [\![q]\!]$ and $[\![\psi^{\mathsf{i}}(t)\theta]\!] = \mathsf{true}$.*

Proof. If t is already in normal form w.r.t. $\widehat{\mathcal{P}}$, then the claim is trivial since $\mathfrak{a}^{\mathsf{i}}(t) = t$ and $\psi^{\mathsf{i}}(t) = \mathsf{true}$. Otherwise, let $\mathcal{P}os_c^{top}(t) = \{\mu_1, \ldots, \mu_k\}$, let $g_i = \mathrm{root}(t|_{\mu_i})$ for each $1 \le i \le k$, and let $n_i \in \mathbb{N}$ be the $\widehat{\mathcal{P}}$-normal form obtained for $t|_{\mu_i}$ in the rewrite sequence $t \to_{\widehat{\mathcal{P}}}^* q$ for each $1 \le i \le k$, i.e., we have:

$$\begin{array}{ll} & t \\ \to_{\widehat{\mathcal{P}}}^* & [\![t[n_1]_{\mu_1} \ldots [n_k]_{\mu_k}]\!] \\ = & q \qquad\qquad\qquad\quad \text{as } \mathrm{root}(t) \in \Sigma_d \end{array} \tag{67}$$

To see why the normal forms n_i are in $\mathbb{N}$, note that we have $t|_{\mu_i} \in \mathcal{T}(\Sigma_c \cup \Sigma_{\mathsf{exp}}, \varnothing)$, as t is ground and its only defined symbol is at position ε. Moreover, every $\widehat{\mathcal{P}}$-normal form of a term from $\mathcal{T}(\Sigma_c \cup \Sigma_{\mathsf{exp}}, \varnothing)$ is in $\mathbb{N}$ by definition of $\widehat{\mathcal{P}}$. Since sz is a size bound for $\mathcal{P}$ and hence, by Lemma 48, also for $\widehat{\mathcal{P}}$, we get

$$n_i \le [\![\mathsf{sz}(t|_{\mu_i})]\!]. \tag{68}$$

By definition of $\mathfrak{a}^{\mathsf{i}}$, $\mathfrak{a}^{\mathsf{i}}(t)|_{\mu_1} = y_1, \ldots, \mathfrak{a}^{\mathsf{i}}(t)|_{\mu_k} = y_k$ are pairwise different fresh variables and these are the only positions where t and $\mathfrak{a}^{\mathsf{i}}(t)$ differ. Let $\theta = \{y_i/n_i \mid 1 \le i \le k\}$. Then (67) implies

$$[\![\mathfrak{a}^{\mathsf{i}}(t)\theta]\!] = [\![q]\!].$$

Moreover, we get:

$$\begin{array}{lll} & [\![\psi^{\mathsf{i}}(t)\theta]\!] & \\ = & \left[\!\!\left[\left(\bigwedge_{1 \le i \le k} y_i \le \mathsf{sz}(t|_{\mu_i})\right)\theta\right]\!\!\right] & \text{by def. of } \psi^{\mathsf{i}} \\ = & \left[\!\!\left[\left(\bigwedge_{1 \le i \le k} n_i \le \mathsf{sz}(t|_{\mu_i})\right)\right]\!\!\right] & \text{by def. of } \theta \\ = & \mathsf{true} & \text{by (68)} \end{array}$$

□

Lemma 64 shows that rewrite steps with $\mathcal{P}$ on nat-basic terms can be simulated by the RNTS $\mathcal{P}^i$ that results from the inner abstraction.

Lemma 64 (Simulating $\mathcal{P}$-steps with $\mathcal{P}^i$). *Let $\mathcal{P}$ be an RNTS with size and runtime bounds* sz *and* rt*, where $\mathcal{P}$ does not have nested defined symbols, and let s be a nat-basic term such that $s \xrightarrow{m_1}_{\mathcal{P}} t \xrightarrow{m_2}_{\widehat{\mathcal{P}}}^* t'$ where t' is a $\widehat{\mathcal{P}}$-normal form. Then $s \xrightarrow{e}_{\mathcal{P}^i \cup \widehat{\mathcal{P}}}^* t'$ where $e \ge m_1 + m_2$.*

Proof. Let $\mathcal{P}os_d(t) = \{\pi_1, \ldots, \pi_m\}$. Since $\mathcal{P}$ does not contain nested defined symbols, these positions are parallel. Let q_i be the $\widehat{\mathcal{P}}$-normal form of $t|_{\pi_i}$ obtained in the rewrite sequence $t \to_{\widehat{\mathcal{P}}}^* t'$ for each $1 \le i \le m$, i.e., we have

$$t \xrightarrow{m_{2,1}}_{\widehat{\mathcal{P}}}^* [\![t[q_1]_{\pi_1} \ldots [q_m]_{\pi_m}]\!] \xrightarrow{m_{2,2}}_{\widehat{\mathcal{P}}}^* t' \text{ where } m_{2,1} + m_{2,2} = m_2. \tag{69}$$

By definition, $\mathfrak{a}^{\mathsf{i}}$ only modifies subterms below defined symbols. Hence,

$$t \text{ and } \mathfrak{a}^{\mathsf{i}}(t) \text{ only differ below } \pi_1, \ldots, \pi_m. \tag{70}$$

By Lemma 63, for each $1 \leq i \leq m$ there is a substitution θ_i such that

$$[\![\mathfrak{a}^{\mathsf{i}}(t|_{\pi_i})\theta_i]\!] = [\![q_i]\!] \text{ and} \tag{71}$$
$$[\![\psi^{\mathsf{i}}(t|_{\pi_i})\theta_i]\!] = \mathsf{true}. \tag{72}$$

As t does not have nested defined symbols, we have

$$\mathfrak{a}^{\mathsf{i}}(t)|_{\pi_i} = \mathfrak{a}^{\mathsf{i}}(t|_{\pi_i})\mu \text{ and} \tag{73}$$
$$\psi^{\mathsf{i}}(t) = \bigwedge\nolimits_{1 \leq i \leq m} \psi^{\mathsf{i}}(t|_{\pi_i})\mu \tag{74}$$

for each $1 \leq i \leq m$ where μ is a variable renaming such that $\mathfrak{a}^{\mathsf{i}}(t|_{\pi_i})\mu = \mathfrak{a}^{\mathsf{i}}(t)|_{\pi_i}$ for all $i \leq i \leq m$. Note that such a variable renaming exists, as all variables introduced by $\mathfrak{a}^{\mathsf{i}}$ are fresh. Let $\ell \xrightarrow{w} r\,[\varphi]$ and σ be the rule and substitution used for the rewrite step $s \to_{\mathcal{P}} t$. Note that we have

$$t = [\![r\sigma]\!] \tag{75}$$

as s is nat-basic. By definition of $\mathcal{P}^i$ and Lemma 54, we have

$$\ell \xrightarrow{w+u} \mathfrak{a}^{\mathsf{i}}(r)\,\left[\varphi \wedge \psi^{\mathsf{i}}(r)\right] \in \mathcal{P}^i \text{ where } u = \sum\nolimits_{\mu \in \mathcal{P}os_c^{top}(r)} \mathfrak{c}(r|_\mu). \tag{76}$$

Let $\mu^{-1} = \{x/y \mid \mu(y) = x\}$ be the inverse of the variable renaming μ and let θ be the substitution that behaves like $\mu^{-1}\theta_i$ on the fresh variables in $\mathfrak{a}^{\mathsf{i}}(t|_{\pi_i})\mu$ and like σ on the variables of the applied rule $\ell \xrightarrow{w+u} r\,[\varphi]$. Then we have

$$\begin{array}{lll}
& [\![(\varphi \wedge \psi^{\mathsf{i}}(r))\theta]\!] & \\
= & [\![\varphi\sigma \wedge \psi^{\mathsf{i}}(r)\theta]\!] & \text{since } \varphi\theta = \varphi\sigma \\
= & [\![\varphi\sigma \wedge \psi^{\mathsf{i}}(r)\theta_1 \dots \theta_m\sigma]\!] & \text{by def. of } \theta \\
= & [\![\varphi\sigma \wedge \psi^{\mathsf{i}}(r\sigma)\theta_1 \dots \theta_m]\!] & \text{as } x\theta_i = x \text{ for all } x \in \mathcal{V}(r), 1 \leq i \leq m, \\
& & \text{and } \sigma \text{ is nat. subst.} \\
= & [\![\varphi\sigma \wedge \psi^{\mathsf{i}}(t)\theta_1 \dots \theta_m]\!] & \text{by (75)} \\
= & [\![\varphi\sigma \wedge \left(\bigwedge\nolimits_{1 \leq i \leq m} \psi^{\mathsf{i}}(t|_{\pi_i})\mu\right) \theta_1 \dots \theta_m]\!] & \text{by (74)} \\
= & [\![\varphi\sigma \wedge \left(\bigwedge\nolimits_{1 \leq i \leq m} \psi^{\mathsf{i}}(t|_{\pi_i})\mu\mu^{-1}\theta_i\right)]\!] & \text{by def. of } \theta_i \\
= & [\![\varphi\sigma \wedge \left(\bigwedge\nolimits_{1 \leq i \leq m} \psi^{\mathsf{i}}(t|_{\pi_i})\theta_i\right)]\!] & \text{by def. of } \mu^{-1} \\
= & \mathsf{true} & \text{by (72)}
\end{array} \tag{77}$$

and hence

$$
\begin{array}{lll}
 & \ell\theta & \\
\xrightarrow{[\![w\theta+u\theta]\!]}_{\mathcal{P}^i} & [\![\mathfrak{a}^{\mathsf{i}}(r)\theta]\!] & \text{by (76) and (77)} \\
= & [\![\mathfrak{a}^{\mathsf{i}}(r\sigma)\theta]\!] & \text{since } r\theta = r\sigma \text{ and} \\
 & & \sigma \text{ is nat. subst.} \\
= & [\![\mathfrak{a}^{\mathsf{i}}(t)\theta]\!] & \text{by (75)} \\
= & [\![\mathfrak{a}^{\mathsf{i}}(t)[\mathfrak{a}^{\mathsf{i}}(t|_{\pi_1})\mu]_{\pi_1} \dots [\mathfrak{a}^{\mathsf{i}}(t|_{\pi_m})\mu]_{\pi_m}\theta]\!] & \text{by (73)} \\
= & [\![t[\mathfrak{a}^{\mathsf{i}}(t|_{\pi_1})\mu]_{\pi_1} \dots [\mathfrak{a}^{\mathsf{i}}(t|_{\pi_m})\mu]_{\pi_m}\theta]\!] & \text{by (70)} \\
= & [\![t[\mathfrak{a}^{\mathsf{i}}(t|_{\pi_1})\mu\mu^{-1}\theta_1]_{\pi_1} \dots [\mathfrak{a}^{\mathsf{i}}(t|_{\pi_m})\mu\mu^{-1}\theta_m]_{\pi_m}]\!] & \text{by def. of } \theta \\
= & [\![t[\mathfrak{a}^{\mathsf{i}}(t|_{\pi_1})\theta_1]_{\pi_1} \dots [\mathfrak{a}^{\mathsf{i}}(t|_{\pi_m})\theta_m]_{\pi_m}]\!] & \text{by def. of } \mu^{-1} \\
= & [\![t[[\![\mathfrak{a}^{\mathsf{i}}(t|_{\pi_1})\theta_1]\!]]_{\pi_1} \dots [[\![\mathfrak{a}^{\mathsf{i}}(t|_{\pi_m})\theta_m]\!]]_{\pi_m}]\!] & (\dagger) \\
= & [\![t[[\![q_1]\!]]_{\pi_1} \dots [[\![q_m]\!]]_{\pi_m}]\!] & \text{by (71)} \\
= & [\![t[q_1]_{\pi_1} \dots [q_m]_{\pi_m}]\!] & (\dagger) \\
\xrightarrow{m_{2,2}}{}^{*}_{\widehat{\mathcal{P}}} & t'. & \text{by (69)}
\end{array}
$$

The steps marked with (†) hold as we clearly have $[\![t[q]_\pi]\!] = [\![t[[\![q]\!]]_\pi]\!]$ for all $t, q \in \mathcal{T}$ and all positions $\pi \in \mathcal{P}os(t)$.

It remains to show $[\![w\theta + u\theta]\!] + m_{2,2} \geq m_1 + m_2$. We have:

$$
\begin{array}{lll}
 & [\![w\theta + u\theta]\!] + m_{2,2} \geq m_1 + m_2 & \\
\iff & [\![w\theta]\!] + [\![u\theta]\!] + m_{2,2} \geq m_1 + m_{2,1} + m_{2,2} & \text{by (69)} \\
\iff & [\![w\theta]\!] + [\![u\theta]\!] \geq m_1 + m_{2,1} & \\
\iff & [\![w\theta]\!] + [\![u\theta]\!] \geq [\![w\sigma]\!] + m_{2,1} & \\
\iff & [\![w\sigma]\!] + [\![u\sigma]\!] \geq [\![w\sigma]\!] + m_{2,1} & \text{by def. of } \theta \\
\iff & [\![u\sigma]\!] \geq m_{2,1} & \\
\iff & \sum_{\mu \in \mathcal{P}os_c^{top}(r)} [\![\mathfrak{c}(r|_\mu)\sigma]\!] \geq m_{2,1} & \text{by (76)} \\
\iff & \sum_{\mu \in \mathcal{P}os_c^{top}(r)} [\![\mathfrak{c}(r\sigma|_\mu)]\!] \geq m_{2,1} & \sigma \text{ is nat. subst.} \\
\iff & \sum_{\mu \in \mathcal{P}os_c^{top}(t)} [\![\mathfrak{c}(t|_\mu)]\!] \geq m_{2,1} & \text{by (75)} \\
\iff & [\![\mathfrak{c}(t)]\!] \geq m_{2,1} & \text{by Lemma 53} \\
\iff & \mathsf{true} & \text{by Lemma 56}
\end{array}
$$

For the last step, recall that $m_{2,1}$ is the cost of normalizing all terms $t|_\pi$ with $\widehat{\mathcal{P}}$, for all $\pi \in \mathcal{P}os_d(t)$. □

In the special case where the term after the rewrite step is already in normal form w.r.t. $\widehat{\mathcal{P}}$, the corresponding rewrite step can of course also be done with $\mathcal{P}^i$.

Lemma 65 (Simulating $\mathcal{P}$-steps with $\mathcal{P}^i$ for $\widehat{\mathcal{P}}$-normal forms). *Let $\mathcal{P}$ be an RNTS with size and runtime bounds* sz *and* rt*. If $s \xrightarrow{m}_{\mathcal{P}} t$ and t is in $\widehat{\mathcal{P}}$-normal form, then $s \xrightarrow{m}_{\mathcal{P}^i} t$.*

Proof. Let $\ell \xrightarrow{w} r\,[\varphi]$ be the rule used for the rewrite step $s \to_{\mathcal{P}} t$. As t is in $\widehat{\mathcal{P}}$-normal form, r does not have constructors below defined symbols. Hence, we have $\ell \xrightarrow{w} r\,[\varphi] \in \mathcal{P}^i$. □

Now we can show that every size resp. time bound for $\mathcal{P}^i \cup \widehat{\mathcal{P}}$ is also a size resp. time bound for $\mathcal{P} \cup \widehat{\mathcal{P}}$.

Theorem 66 (Bounds for $\mathcal{P} \cup \widehat{\mathcal{P}}$ and $\mathcal{P}^i \cup \widehat{\mathcal{P}}$). *Let $\mathcal{P}$ be an RNTS with size and runtime bounds* sz *and* rt*, where $\mathcal{P}$ does not have nested defined symbols. Then every size bound for $\mathcal{P}^i \cup \widehat{\mathcal{P}}$ is also a size bound for $\mathcal{P} \cup \widehat{\mathcal{P}}$ and every runtime bound for $\mathcal{P}^i \cup \widehat{\mathcal{P}}$ is also a runtime bound for $\mathcal{P} \cup \widehat{\mathcal{P}}$.*

Proof. To prove that every size bound for $\mathcal{P}^i \cup \widehat{\mathcal{P}}$ is also a size bound for $\mathcal{P} \cup \widehat{\mathcal{P}}$, it suffices to show that if a nat-basic term has a normal form $n \in \mathbb{N}$ w.r.t. $\mathcal{P} \cup \widehat{\mathcal{P}}$, then it has the same normal form w.r.t. $\mathcal{P}^i \cup \widehat{\mathcal{P}}$. To this end, we show a slightly generalized claim: If $s \in \mathcal{T}$ is a ground term in $\widehat{\mathcal{P}}$-normal form without nested defined symbols and $s \to^k_{\mathcal{P} \cup \widehat{\mathcal{P}}} n$ for $n \in \mathbb{N}$, then we also have $s \to^*_{\mathcal{P}^i \cup \widehat{\mathcal{P}}} n$.

We use induction on k and assume a reduction strategy that applies rules from $\widehat{\mathcal{P}}$ with a higher preference than rules from $\mathcal{P}$, i.e., $\mathcal{P}$-rules are just applied to $\widehat{\mathcal{P}}$-normal forms. This assumption can be made without loss of generality, because the variables in the rules of RNTSs may only be instantiated by numbers.

In the induction base ($k = 0$) we have $s = n$ and hence the claim is trivial. In the induction step ($k > 0$) there are two cases:

Case 1: We first consider the case $s \to^k_{\mathcal{P}} n$. Note that due to the reduction strategy, all terms in this sequence are in $\widehat{\mathcal{P}}$-normal form. Then we have $s \to^k_{\mathcal{P}^i} n$ due to Lemma 65.

Case 2: Now we consider the case $s \to^a_{\mathcal{P}} s' \to_{\mathcal{P}} s'' \to^b_{\widehat{\mathcal{P}}} \widehat{s} \to^{k-a-b-1}_{\mathcal{P} \cup \widehat{\mathcal{P}}} n$ for $a, b \geq 0$ where $\widehat{s}$ is in $\widehat{\mathcal{P}}$-normal form. Note that in this case there is at least one $\mathcal{P}$-step before the first $\widehat{\mathcal{P}}$-step, as s is in $\widehat{\mathcal{P}}$-normal form. Let $\ell \to r\,[\varphi]$, σ, and π be the rule, the substitution, and the position of the rewrite step $s' \to_{\mathcal{P}} s''$. Then, by Lemma 64, we get $s'|_\pi \to^*_{\mathcal{P}^i \cup \widehat{\mathcal{P}}} q$ where q is the $\widehat{\mathcal{P}}$-normal form of $\llbracket r\sigma \rrbracket$ obtained in the rewrite sequence $s'' \to^b_{\widehat{\mathcal{P}}} \widehat{s}$, i.e., we have

$$s' \to^*_{\mathcal{P}^i \cup \widehat{\mathcal{P}}} \llbracket s'[q]_\pi \rrbracket \to^*_{\widehat{\mathcal{P}}} \widehat{s}.$$

By the induction hypothesis, we know that n is a $\mathcal{P}^i \cup \widehat{\mathcal{P}}$-normal form of $\widehat{s}$. Moreover, we have $s \to^a_{\mathcal{P}^i} s'$ due to Lemma 65 since by the assumption on the reduction strategy, rules from $\mathcal{P}$ are just applied to $\widehat{\mathcal{P}}$-normal forms. Hence, we obtain

$$s \to^a_{\mathcal{P}^i} s' \to^*_{\mathcal{P}^i \cup \widehat{\mathcal{P}}} \llbracket s'[q]_\pi \rrbracket \to^*_{\widehat{\mathcal{P}}} \widehat{s} \to^*_{\mathcal{P}^i \cup \widehat{\mathcal{P}}} n.$$

To prove that every runtime bound for $\mathcal{P}^i \cup \widehat{\mathcal{P}}$ is also a runtime bound for $\mathcal{P} \cup \widehat{\mathcal{P}}$, it suffices to show that if we have $s \xrightarrow{m}{}^*_{\mathcal{P} \cup \widehat{\mathcal{P}}} t$ for a nat-basic term s and a term t in $\widehat{\mathcal{P}}$-normal form, then we have $s \xrightarrow{m'}{}^*_{\mathcal{P}^i \cup \widehat{\mathcal{P}}} t$ for some $m' \geq m$. Note that we can assume that t is in $\widehat{\mathcal{P}}$-normal form, as $\widehat{\mathcal{P}}$ is trivially terminating. Again, we show a slightly generalized claim: If $s \in \mathcal{T}$ is a ground term in $\widehat{\mathcal{P}}$-normal form without nested defined symbols and $s \xrightarrow{m}^{k}_{\mathcal{P}\cup\widehat{\mathcal{P}}} t$ for some term t in $\widehat{\mathcal{P}}$-normal form, then $s \xrightarrow{m'}^{*}_{\mathcal{P}^i\cup\widehat{\mathcal{P}}} t$ for some $m' \geq m$.

We again use induction on k and assume a reduction strategy that applies rules from $\widehat{\mathcal{P}}$ with a higher preference than rules from $\mathcal{P}$.

In the induction base ($k = 0$) we have $s = t$ and $m = 0$ and hence the claim is trivial. In the induction step ($k > 0$) there are two cases:

Case 1: We first consider the case $s \xrightarrow{m}^{k}_{\mathcal{P}} t$. Since again all terms in this sequence are in $\widehat{\mathcal{P}}$-normal form by the reduction strategy, we have $s \xrightarrow{m}^{k}_{\mathcal{P}^i} t$ due to Lemma 65.

Case 2: Now we consider the case $s \xrightarrow{m_1}^{a}_{\mathcal{P}} s' \xrightarrow{m_2}_{\mathcal{P}} s'' \xrightarrow{m_3}^{b}_{\widehat{\mathcal{P}}} \widehat{s} \xrightarrow{m_4}^{k-a-b-1}_{\mathcal{P}\cup\widehat{\mathcal{P}}} t$ for $a, b \geq 0$ where $\widehat{s}$ is in $\widehat{\mathcal{P}}$-normal form. Let $\ell \xrightarrow{w} r\,[\varphi]$, σ, and π be the rule, the substitution, and the position of the rewrite step $s' \xrightarrow{m_2}_{\mathcal{P}} s''$ and let q be the $\widehat{\mathcal{P}}$-normal form of $[\![r\sigma]\!]$ obtained in the rewrite sequence $s'' \xrightarrow{m_3}^{b}_{\widehat{\mathcal{P}}} \widehat{s}$, i.e., we have

$$s' \xrightarrow{m_2}_{\mathcal{P}} \circ \xrightarrow{m_{3,1}}^{*}_{\widehat{\mathcal{P}}} [\![s'[q]_\pi]\!] \xrightarrow{m_{3,2}}^{*}_{\widehat{\mathcal{P}}} \widehat{s} \text{ where } m_{3,1} + m_{3,2} = m_3. \tag{78}$$

Then, by Lemma 64, we get

$$s'|_\pi \xrightarrow{e_{2,3}}^{*}_{\mathcal{P}^i\cup\widehat{\mathcal{P}}} q \text{ where } e_{2,3} \geq m_2 + m_{3,1}. \tag{79}$$

By the induction hypothesis, we know

$$\widehat{s} \xrightarrow{e_4}^{*}_{\mathcal{P}^i\cup\widehat{\mathcal{P}}} t \text{ for some } e_4 \geq m_4. \tag{80}$$

Moreover, we have

$$s \xrightarrow{m_1}^{a}_{\mathcal{P}^i} s' \tag{81}$$

due to Lemma 65 since by the assumption on the reduction strategy, rules from $\mathcal{P}$ are just applied to $\widehat{\mathcal{P}}$-normal forms. Hence, we obtain

$$\begin{array}{lll} & s & \\ \xrightarrow{m_1}^{a}_{\mathcal{P}^i} & s' & \text{by (81)} \\ \xrightarrow{e_{2,3}}^{*}_{\mathcal{P}^i\cup\widehat{\mathcal{P}}} & [\![s'[q]_\pi]\!] & \text{by (79)} \\ \xrightarrow{m_{3,2}}^{*}_{\widehat{\mathcal{P}}} & \widehat{s} & \text{by (78)} \\ \xrightarrow{e_4}^{*}_{\mathcal{P}^i\cup\widehat{\mathcal{P}}} & t & \text{by (80)} \end{array}$$

and we have

$$\begin{array}{ll} \ m_1 + e_{2,3} + m_{3,2} + e_4 & \\ \geq m_1 + m_2 + m_{3,1} + m_{3,2} + e_4 & \text{by (79)} \\ \geq m_1 + m_2 + m_{3,1} + m_{3,2} + m_4 & \text{by (80)} \\ = m_1 + m_2 + m_3 + m_4 & \text{by (78)} \end{array}$$

□

B.2.4 Properties of $\mathcal{P}^o$

Now we want to prove a similar theorem for P^o instead of P^i. Here, however, we are not interested in the runtime of $\mathcal{P}^o$, but just in the size of the results computed by $\mathcal{P}^o$. The reason is that $\mathfrak{a}^\circ$ (which is used to obtain $\mathcal{P}^o$) is just used to construct $\mathcal{P}_{\mathsf{sz}}$, but not for $\mathcal{P}_{\mathsf{rt,sz}}$. In other words, $\mathfrak{a}^\circ$ is just needed to compute size bounds, but not for time bounds.

As a first auxiliary lemma towards this goal, we show how to exchange the order of replacing subterms in a term and of applying $\mathfrak{a}^\circ$.

Lemma 67 (Subterm Replacement and $\mathfrak{a}^\circ$). *Let $\mathcal{P}$ be an RNTS with size bound* sz*, let s be a term, and let $\pi \in \mathcal{P}os(s)$. Then there are positions $\pi_1, \ldots, \pi_m$ with $\mathfrak{a}^\circ(s[t]_\pi) = \mathfrak{a}^\circ(s)[t]_{\pi_1} \ldots [t]_{\pi_m}$ for every ground term t where constructors just occur below defined symbols.*

Proof. Note that we have $\mathfrak{a}^\circ(t) = t$, as constructors just occur below defined symbols in t. We use induction on π. If $\pi = \varepsilon$, let $m = 1$ and $\pi_1 = \varepsilon$. Then we get $\mathfrak{a}^\circ(s[t]_\pi) = \mathfrak{a}^\circ(t) = t = \mathfrak{a}^\circ(s)[t]_{\pi_1}$. In the induction step, let $\pi = i.\pi'$ and $s = g(s_1, \ldots, s_n)$. By the induction hypothesis, there exist positions $\pi'_1, \ldots, \pi'_k$ with

$$\mathfrak{a}^\circ(s_i[t]_{\pi'}) = \mathfrak{a}^\circ(s_i)[t]_{\pi'_1} \ldots [t]_{\pi'_k}. \tag{82}$$

If $g \in \Sigma_{\mathsf{exp}}$, we have

$$\begin{aligned}
&\mathfrak{a}^\circ(g(s_1, \ldots, s_n)[t]_\pi) \\
= &\mathfrak{a}^\circ(g(s_1, \ldots, s_n)[t]_{i.\pi'}) \\
= &\mathfrak{a}^\circ(g(s_1, \ldots, s_i[t]_{\pi'}, \ldots, s_n) \\
= &g(\mathfrak{a}^\circ(s_1), \ldots, \mathfrak{a}^\circ(s_i[t]_{\pi'}), \ldots \mathfrak{a}^\circ(s_n)) && \text{by Def. 23} \\
= &g(\mathfrak{a}^\circ(s_1), \ldots, \mathfrak{a}^\circ(s_i)[t]_{\pi'_1} \ldots [t]_{\pi'_k}, \ldots, \mathfrak{a}^\circ(s_n)) && \text{by (82)} \\
= &g(\mathfrak{a}^\circ(s_1), \ldots, \mathfrak{a}^\circ(s_i), \ldots, \mathfrak{a}^\circ(s_n))[t]_{i.\pi'_1} \ldots [t]_{i.\pi'_k} \\
= &\mathfrak{a}^\circ(g(s_1, \ldots, s_n))[t]_{i.\pi'_1} \ldots [t]_{i.\pi'_k} && \text{by Def. 23}
\end{aligned}$$

Now we consider the case $g \in \Sigma_c$. Let $\kappa_1, \ldots, \kappa_d$ be the positions of x_i in $\mathsf{sz}(g)$ (†). Then we get:

$$\begin{aligned}
&\mathfrak{a}^\circ(g(s_1, \ldots, s_n)[t]_\pi) \\
= &\mathfrak{a}^\circ(g(s_1, \ldots, s_n)[t]_{i.\pi'}) \\
= &\mathfrak{a}^\circ(g(s_1, \ldots, s_i[t]_{\pi'}, \ldots, s_n)) \\
= &\mathsf{sz}(g)(\mathfrak{a}^\circ(s_1), \ldots, \mathfrak{a}^\circ(s_i[t]_{\pi'}), \ldots, \mathfrak{a}^\circ(s_n)) && \text{by Def. 23} \\
= &\mathsf{sz}(g)(\mathfrak{a}^\circ(s_1), \ldots, \mathfrak{a}^\circ(s_i)[t]_{\pi'_1} \ldots [t]_{\pi'_k}, \ldots, \mathfrak{a}^\circ(s_n)) && \text{by (82)} \\
= &\mathsf{sz}(g)(\mathfrak{a}^\circ(s_1), \ldots, \mathfrak{a}^\circ(s_i), \ldots, \mathfrak{a}^\circ(s_n)) \\
&\qquad [t]_{\kappa_1.\pi'_1} \ldots [t]_{\kappa_1.\pi'_k} \ldots [t]_{\kappa_d.\pi'_1} \ldots [t]_{\kappa_d.\pi'_k} && \text{by (†)} \\
= &\mathfrak{a}^\circ(g(s_1, \ldots, s_n))[t]_{\kappa_1.\pi'_1} \ldots [t]_{\kappa_1.\pi'_k} \ldots [t]_{\kappa_d.\pi'_1} \ldots, [t]_{\kappa_d.\pi'_k} && \text{by Def. 23}
\end{aligned}$$

If $g \in \Sigma_d$, then the claim is trivial since $\mathfrak{a}^\circ(s[t]_\pi) = s[t]_\pi$ and $\mathfrak{a}^\circ(s) = s$. □

Now we show that every rewrite step with $\mathcal{P}$ can be simulated with $\mathcal{P}^o$ when applying $\mathfrak{a}^\circ$.

Lemma 68 (Simulating $\mathcal{P}$-steps with $\mathcal{P}^o$). *Let $\mathcal{P}$ an RNTS with size bound* sz *and let s be a term without nested defined symbols. If $s \to_{\mathcal{P}} t$, then*

$$\mathfrak{a}^\circ(s) \to^*_{\mathcal{P}^o} [\![\mathfrak{a}^\circ(t)]\!].$$

Proof. Let $\ell \xrightarrow{w} r\,[\varphi]$, π, and σ be the rule, the position, and the natural substitution used for the rewrite step $s \to_{\mathcal{P}} t$. By Lemma 67 there are positions $\pi_1, \ldots, \pi_m$ such that

$$\mathfrak{a}^\circ(s)[q]_{\pi_1} \ldots [q]_{\pi_m} = \mathfrak{a}^\circ(s[q]_\pi)$$

for every ground term q where constructors just occur below defined symbols. Hence, we have

$$\mathfrak{a}^\circ(s)[s|_\pi]_{\pi_1} \ldots [s|_\pi]_{\pi_m} = \mathfrak{a}^\circ(s) \tag{83}$$

as $\mathrm{root}(s|_\pi) \in \Sigma_d$ and

$$\begin{aligned} &\mathfrak{a}^\circ(s)[[\![\mathfrak{a}^\circ(r\sigma)]\!]]_{\pi_1} \ldots [[\![\mathfrak{a}^\circ(r\sigma)]\!]]_{\pi_m} \\ = \;&\mathfrak{a}^\circ(s[[\![\mathfrak{a}^\circ(r\sigma)]\!]]_\pi) \end{aligned} \tag{84}$$

as, by definition of $\mathfrak{a}^\circ$, constructors just occur below defined symbols in $\mathfrak{a}^\circ(r\sigma)$ and hence also in $[\![\mathfrak{a}^\circ(r\sigma)]\!]$. Since s does not have nested defined symbols and $\mathrm{root}(s|_\pi) \in \Sigma_d$, s does not have defined symbols above the position π. Hence, since $[\![s[r\sigma]_\pi]\!] = t$, we get

$$\mathfrak{a}^\circ(s[[\![\mathfrak{a}^\circ(r\sigma)]\!]]_\pi) = \mathfrak{a}^\circ([\![s[r\sigma]_\pi]\!]) = \mathfrak{a}^\circ([\![t]\!]). \tag{85}$$

By definition, we have $\ell \xrightarrow{w} \mathfrak{a}^\circ(r)\,[\varphi] \in \mathcal{P}^o$. Hence, we get

$$\begin{array}{lll} & \mathfrak{a}^\circ(s) & \\ = & \mathfrak{a}^\circ(s)[s|_\pi]_{\pi_1} \ldots [s|_\pi]_{\pi_m} & \text{by (83)} \\ \to^*_{\mathcal{P}^o} & [\![\mathfrak{a}^\circ(s)[\mathfrak{a}^\circ(r)\sigma]_{\pi_1} \ldots [\mathfrak{a}^\circ(r)\sigma]_{\pi_m}]\!] & \text{as } \ell\sigma = s|_\pi \text{ and } [\![\varphi\sigma]\!] = \mathsf{true} \\ = & [\![\mathfrak{a}^\circ(s)[\mathfrak{a}^\circ(r\sigma)]_{\pi_1} \ldots [\mathfrak{a}^\circ(r\sigma)]_{\pi_m}]\!] & \text{as } \sigma \text{ is a natural substitution} \\ = & [\![\mathfrak{a}^\circ(s[\mathfrak{a}^\circ(r\sigma)]_\pi)]\!] & \text{by (84)} \\ = & [\![\mathfrak{a}^\circ(t)]\!]. & \text{by (85)} \end{array}$$

□

The following lemma shows that rewrite steps with $\mathcal{P}^o$ can still be performed when applying $\mathfrak{a}^\circ$.

Lemma 69 (Simulating $\mathcal{P}^o$-steps When Applying $\mathfrak{a}^\circ$). *Let $\mathcal{P}$ be an RNTS with size bound* sz *and let s be a term without nested defined symbols. If $s \to_{\mathcal{P}^o} t$, then*

$$[\![\mathfrak{a}^\circ(s)]\!] \to^*_{\mathcal{P}^o} [\![\mathfrak{a}^\circ(t)]\!].$$

Proof. Let $\ell \xrightarrow{w} r\,[\varphi]$, π, and σ be the rule, the position, and the natural substitution used for the rewrite step $s \to_{\mathcal{P}} t$. By Lemma 67 there are positions $\pi_1, \ldots, \pi_m$ such that

$$\mathfrak{a}^\circ(s)[q]_{\pi_1} \ldots [q]_{\pi_m} = \mathfrak{a}^\circ(s[q]_\pi)$$

for every ground term q where constructors just occur below defined symbols. Hence, we have

$$\mathfrak{a}^\circ(s)[s|_\pi]_{\pi_1}\dots[s|_\pi]_{\pi_m} = \mathfrak{a}^\circ(s) \tag{86}$$

as $\mathrm{root}(s|_\pi) \in \Sigma_d$ and

$$\mathfrak{a}^\circ(s)[[\![r\sigma]\!]]_{\pi_1}\dots[[\![r\sigma]\!]]_{\pi_m} = \mathfrak{a}^\circ(s[[\![r\sigma]\!]]_\pi),$$

as constructors just occur below defined symbols in r (and hence $r\sigma$) by definition of $\mathcal{P}^o$. As $[\![s[r\sigma]_\pi]\!] = t$, this implies

$$[\![\mathfrak{a}^\circ(s)[r\sigma]_{\pi_1}\dots[r\sigma]_{\pi_m}]\!] = [\![\mathfrak{a}^\circ(t)]\!]. \tag{87}$$

Hence, we get:

$$\begin{array}{lll} & [\![\mathfrak{a}^\circ(s)]\!] & \\ = & [\![\mathfrak{a}^\circ(s)[s|_\pi]_{\pi_1}\dots[s|_\pi]_{\pi_m}]\!] & \text{by (86)} \\ \to^*_{\mathcal{P}^o} & [\![\mathfrak{a}^\circ(s)[r\sigma]_{\pi_1}\dots[r\sigma]_{\pi_m}]\!] & \\ = & [\![\mathfrak{a}^\circ(t)]\!] & \text{by (87)} \end{array}$$

□

Similar to Thm. 66 for $\mathcal{P}^i$, we can now show that every size bound for $\mathcal{P}^o \cup \widehat{\mathcal{P}}$ is also a size bound for $\mathcal{P} \cup \widehat{\mathcal{P}}$.

Theorem 70 (Size Bounds for $\mathcal{P} \cup \widehat{\mathcal{P}}$ and $\mathcal{P}^o \cup \widehat{\mathcal{P}}$). *Let $\mathcal{P}$ be an RNTS with size bound sz, where $\mathcal{P}$ does not have Σ-symbols below defined symbols. Then every size bound for $\mathcal{P}^o \cup \widehat{\mathcal{P}}$ is also a size bound for $\mathcal{P} \cup \widehat{\mathcal{P}}$.*

Proof. Let $t \in \mathcal{T}$ be a nat-basic term and let $n \in \mathbb{N}$ such that $t \to^k_{\mathcal{P}\cup\widehat{\mathcal{P}}} n$. We prove $t \to^k_{\mathcal{P}^o\cup\widehat{\mathcal{P}}} n' \geq n$ by induction on k. If $k = 0$, then the claim is trivial. Let $k > 0$. As t is nat-basic, we have $t \to_{\mathcal{P}} t' \to^{k-1}_{\mathcal{P}\cup\widehat{\mathcal{P}}} n$ and

$$t = \mathfrak{a}^\circ(t). \tag{88}$$

By Lemma 68 we get

$$\mathfrak{a}^\circ(t) \to^*_{\mathcal{P}^o} [\![\mathfrak{a}^\circ(t')]\!]. \tag{89}$$

Let $\mathcal{P}os_d(t') = \{\pi_1, \dots, \pi_m\}$. Since $\mathcal{P}$ does not contain nested defined symbols, these positions are parallel. Let $n_i \in \mathbb{N}$ be the normal form of $t'|_{\pi_i}$ obtained in the rewrite sequence $t \to^{k-1}_{\mathcal{P}\cup\widehat{\mathcal{P}}} n$, i.e., we have

$$t \to_{\mathcal{P}} t' \to^*_{\mathcal{P}\cup\widehat{\mathcal{P}}} [\![t'[n_1]_{\pi_1}\dots[n_m]_{\pi_m}]\!] \to^*_{\widehat{\mathcal{P}}} n. \tag{90}$$

(If $\mathcal{P}os_d(t') = \varnothing$, then we have just t' instead of $[\![t'[n_1]_{\pi_1}\dots[n_m]_{\pi_m}]\!]$, but the proof works analogously.) Note that as $\mathcal{P}$ does not have Σ-symbols below defined symbols, $t'|_{\pi_i} = [\![t'|_{\pi_i}]\!]$ is nat-basic. By the induction hypothesis, we therefore have $t'|_{\pi_i} \to^*_{\mathcal{P}^o\cup\widehat{\mathcal{P}}} n'_i$ with $n'_i \geq n_i$ for each $1 \leq i \leq m$. As $\mathcal{P}$ and hence $\mathcal{P}^o$ and t' do not have Σ-symbols below defined symbols, the reduction has the form

$t'|_{\pi_i} \to^*_{\mathcal{P}^o} t'' \to^*_{\widehat{\mathcal{P}}} n'_i$, i.e., we can first reduce $t'|_{\pi_i}$ to its $\mathcal{P}^o$-normal form t'' and reduce the constructors in t'' afterwards. Hence, we obtain

$$[\![\mathfrak{a}^\circ(t'|_{\pi_i})]\!] \to^*_{\mathcal{P}^o} [\![\mathfrak{a}^\circ(t'')]\!] \to^*_{\widehat{\mathcal{P}}} n''_i \geq n'_i \tag{91}$$

by Lemma 69 and 62. (To see why we can apply Lemma 62, note that t'' results from a rewrite step with an RNTS and thus, $t'' = [\![t'']\!]$.) For each $1 \leq i \leq m$, as $\mathrm{root}(t'|_{\pi_i}) \in \Sigma_d$, by Lemma 67 there are positions $\kappa^i_1, \ldots, \kappa^i_{k_i}$ such that

$$\mathfrak{a}^\circ(t')[q]_{\kappa^i_1} \ldots [q]_{\kappa^i_{k_i}} = \mathfrak{a}^\circ(t'[q]_{\pi_i})$$

for every ground term q where constructors just occur below defined symbols. Hence, we have

$$\mathfrak{a}^\circ(t') = \mathfrak{a}^\circ(t')[t'|_{\pi_i}]_{\kappa^i_1} \ldots [t'|_{\pi_i}]_{\kappa^i_{k_i}}$$

as $\mathrm{root}(t'|_{\pi_i}) \in \Sigma_d$ and hence

$$\mathfrak{a}^\circ(t') = \mathfrak{a}^\circ(t')[\mathfrak{a}^\circ(t'|_{\pi_i})]_{\kappa^i_1} \ldots [\mathfrak{a}^\circ(t'|_{\pi_i})]_{\kappa^i_{k_i}} \tag{92}$$

as $\mathfrak{a}^\circ(q) = q$ for each term q with $\mathrm{root}(q) \in \Sigma_d$. Moreover, we have

$$\mathfrak{a}^\circ(t'[n''_i]_{\pi_i}) = \mathfrak{a}^\circ(t')[n''_i]_{\kappa^i_1} \ldots [n''_i]_{\kappa^i_{k_i}}. \tag{93}$$

Thus, we get

$$\begin{array}{lll}
 & t & \text{by} \\
= & \mathfrak{a}^\circ(t) & \text{by (88)} \\
\to^*_{\mathcal{P}^o} & [\![\mathfrak{a}^\circ(t')]\!] & \text{by (89)} \\
= & [\![\mathfrak{a}^\circ(t')[\mathfrak{a}^\circ(t'|_{\pi_i})]_{\kappa^i_1} \ldots [\mathfrak{a}^\circ(t'|_{\pi_i})]_{\kappa^i_{k_i}}]\!] & \text{by (92)} \\
\to^*_{\mathcal{P}^o \cup \widehat{\mathcal{P}}} & [\![\mathfrak{a}^\circ(t')[n''_i]_{\kappa^i_1} \ldots [n''_i]_{\kappa^i_{k_i}}]\!] & \text{by (91)} \\
= & [\![\mathfrak{a}^\circ(t'[n''_i]_{\pi_i})]\!] & \text{by (93)}
\end{array}$$

for each $1 \leq i \leq m$ and hence:

$$\begin{array}{lll}
 & t & \\
\to^*_{\mathcal{P}^o \cup \widehat{\mathcal{P}}} & [\![\mathfrak{a}^\circ(t'[n''_1]_{\pi_1} \ldots [n''_m]_{\pi_m})]\!] & \\
\to^*_{\widehat{\mathcal{P}}} & [\![\mathsf{sz}(t'[n''_1]_{\pi_1} \ldots [n''_m]_{\pi_m})]\!] & \text{as } t'[n''_1]_{\pi_1} \ldots [n''_m]_{\pi_m} \in \mathcal{T}(\Sigma_c \cup \Sigma_{\mathsf{exp}}, \varnothing) \\
\geq & [\![\mathsf{sz}(t'[n_1]_{\pi_1} \ldots [n_m]_{\pi_m})]\!] & \text{by monotonicity of } \mathsf{sz}
\end{array}$$

By Lemma 48, sz is a size bound for $\widehat{\mathcal{P}}$. By (90), n is a $\widehat{\mathcal{P}}$-normal form of $[\![t'[n_1]_{\pi_1} \ldots [n_m]_{\pi_m}]\!]$. Hence, we have $[\![\mathsf{sz}(t'[n_1]_{\pi_1} \ldots [n_m]_{\pi_m})]\!] \geq n$, which proves our claim. □

B.2.5 Properties of $\mathcal{P}^c$

Finally, we also want to prove a similar theorem for $\mathcal{P}^c$. Here, however, we are just interested in the runtime of $\mathcal{P}^c$. The reason is that $\mathfrak{a}^{\mathsf{s}}$ (which is used to obtain $\mathcal{P}^c$) is just used to construct $\mathcal{P}'_{\mathsf{rt},\mathsf{sz}}$, but not for $\mathcal{P}_{\mathsf{sz}}$. In other words, $\mathfrak{a}^{\mathsf{s}}$ is just needed to compute time bounds, but not for size bounds.

As a first step towards this goal, we show how $\mathcal{P}$-steps can be simulated using $\mathcal{P}^c$.

Lemma 71 (Simulating $\mathcal{P}$-steps With $\mathcal{P}^c$). *Let $\mathcal{P}$ be an RNTS with size bound* sz *and runtime bound* rt *without constructors below defined symbols and let s be a nat-basic term. If $s \xrightarrow{m}_{\mathcal{P}} t$, then*

$$[\![\mathfrak{a}^{\mathsf{s}}(s)]\!] \xrightarrow{[\![m+\mathfrak{c}(t)]\!]}_{\mathcal{P}^c} [\![\mathfrak{a}^{\mathsf{s}}(t)]\!].$$

Proof. Let $\ell \xrightarrow{w} r\,[\varphi]$ and σ be the rule and the natural substitution used for the rewrite step $s \to_{\mathcal{P}} t$. We get

$$\begin{array}{lll} & [\![\mathfrak{a}^{\mathsf{s}}(s)]\!] & \\ = & s & \text{as } s \text{ is nat-basic} \\ \xrightarrow{[\![w\sigma+\mathfrak{c}^{\mathsf{o}}(r)\sigma]\!]}_{\mathcal{P}^c} & [\![\mathfrak{a}^{\mathsf{s}}(r)\sigma]\!] & \text{by def. of } \mathcal{P}^c \\ = & [\![\mathfrak{a}^{\mathsf{s}}(r\sigma)]\!] & \text{as } \sigma \text{ is nat. subst.} \\ = & [\![\mathfrak{a}^{\mathsf{s}}(t)]\!] & \end{array}$$

We have $[\![w\sigma]\!] = m$ and $[\![\mathfrak{c}^{\mathsf{o}}(r)\sigma]\!] = [\![\mathfrak{c}^{\mathsf{o}}(r\sigma)]\!] = [\![\mathfrak{c}^{\mathsf{o}}(t)]\!]$. Moreover, we have $[\![\mathfrak{c}^{\mathsf{o}}(t)]\!] = [\![\mathfrak{c}(t)]\!]$ as $\mathcal{P}$ and thus t does not have constructors below defined symbols. ☐

The next lemma shows that $\mathcal{P}^c$-steps can still be applied if the reduced term is abstracted using $\mathfrak{a}^{\mathsf{s}}$.

Lemma 72 (Simulating $\mathcal{P}^c$-steps When Applying $\mathfrak{a}^{\mathsf{s}}$). *Let $\mathcal{P}$ be an RNTS with size bound* sz *and runtime bound* rt *without Σ-symbols below defined symbols and let s be a nat-basic term. If $s \xrightarrow{m}_{\mathcal{P}^c} t$, then*

$$[\![\mathfrak{a}^{\mathsf{s}}(s)]\!] \xrightarrow{m}_{\mathcal{P}^c} [\![\mathfrak{a}^{\mathsf{s}}(t)]\!].$$

Proof. Let $\ell \xrightarrow{w} r\,[\varphi] \in \mathcal{P}^c$ and σ be the rule and the natural substitution used for the rewrite step $s \to_{\mathcal{P}^c} t$. As $\mathcal{P}$ does not have Σ-symbols below defined symbols and r is a sum of terms with defined root symbols, we have

$$r = \mathfrak{a}^{\mathsf{s}}(r). \tag{94}$$

We get

$$\begin{array}{lll} & [\![\mathfrak{a}^{\mathsf{s}}(s)]\!] & \\ = & s & \text{as } s \text{ is nat-basic} \\ \xrightarrow{[\![w\sigma]\!]}_{\mathcal{P}^c} & [\![r\sigma]\!] & \\ = & [\![\mathfrak{a}^{\mathsf{s}}(r)\sigma]\!] & \text{by (94)} \\ = & [\![\mathfrak{a}^{\mathsf{s}}(r\sigma)]\!] & \sigma \text{ is nat. subst.} \\ = & [\![\mathfrak{a}^{\mathsf{s}}(t)]\!] & \end{array}$$

As we have $[\![w\sigma]\!] = m$, this proves the claim. ☐

Now we can show that time bounds for $\mathcal{P}^c \cup \widehat{\mathcal{P}}$ are indeed also time bounds for $\mathcal{P} \cup \widehat{\mathcal{P}}$.

Theorem 73 (Time Bounds for $\mathcal{P} \cup \widehat{\mathcal{P}}$ and $\mathcal{P}^c \cup \widehat{\mathcal{P}}$). *Let $\mathcal{P}$ be an RNTS with size and runtime bounds sz and rt without Σ-symbols below defined symbols. Then every runtime bound for $\mathcal{P}^c \cup \widehat{\mathcal{P}}$ is also a runtime bound for $\mathcal{P} \cup \widehat{\mathcal{P}}$.*

Proof. Let t be a nat-basic term and let q be a $\mathcal{P} \cup \widehat{\mathcal{P}}$-normal form such that $t \xrightarrow{m}{}^{k}_{\mathcal{P}\cup\widehat{\mathcal{P}}} q$. We prove $t \xrightarrow{e}{}^{*}_{\mathcal{P}^c\cup\widehat{\mathcal{P}}} q'$ for some term q' with $e \geq m$ by induction on k. If $k = 0$, then the claim is trivial. Let $k > 0$. If $\mathrm{root}(t) \in \Sigma_c^{\mathcal{P}}$, as t is nat-basic we obtain $t \xrightarrow{m}_{\widehat{\mathcal{P}}} q \in \mathbb{N}$, i.e., $k = 1$ and the claim is again trivial. If $\mathrm{root}(t) \in \Sigma_d^{\mathcal{P}}$, we have $t \xrightarrow{m_1}_{\mathcal{P}} t' \xrightarrow{m_2}{}^{k-1}_{\mathcal{P}\cup\widehat{\mathcal{P}}} q$,

$$m = m_1 + m_2, \text{ and} \tag{95}$$
$$t = \mathfrak{a}^{\mathsf{s}}(t). \tag{96}$$

By Lemma 71 we get

$$[\![\mathfrak{a}^{\mathsf{s}}(t)]\!] \xrightarrow{[\![m_1 + \mathfrak{c}(t')]\!]}_{\mathcal{P}^c} [\![\mathfrak{a}^{\mathsf{s}}(t')]\!]. \tag{97}$$

Let $\mathcal{P}os_d(t') = \{\pi_1, \ldots, \pi_n\}$. Since $\mathcal{P}$ does not contain nested defined symbols, these positions are parallel. For all $1 \leq i \leq n$, let q_i be the normal form of $t'|_{\pi_i}$ obtained in the rewrite sequence $t' \xrightarrow{m_2}{}^{k-1}_{\mathcal{P}\cup\widehat{\mathcal{P}}} q$ and let d_i be the cost of reducing $t'|_{\pi_i}$ to q_i, i.e., we have

$$t \xrightarrow{m_1}_{\mathcal{P}} t' \xrightarrow{d_1+\ldots+d_n}{}^{*}_{\mathcal{P}\cup\widehat{\mathcal{P}}} [\![t'[q_1]_{\pi_1} \ldots [q_n]_{\pi_n}]\!] \xrightarrow{m_{2,2}}{}^{*}_{\widehat{\mathcal{P}}} q \text{ where} \tag{98}$$
$$d_1 + \ldots + d_n + m_{2,2} = m_2. \tag{99}$$

Note that as $\mathcal{P}$ does not have Σ-symbols below defined symbols, $t'|_{\pi_i}$ is nat-basic. By the induction hypothesis, we therefore have

$$t'|_{\pi_i} \xrightarrow{e_i}{}^{*}_{\mathcal{P}^c\cup\widehat{\mathcal{P}}} q'_i \text{ for some term } q'_i \text{ with } e_i \geq d_i \text{ for each } 1 \leq i \leq n. \tag{100}$$

As $\mathcal{P}$ does not have Σ-symbols below defined symbols, $t'|_{\pi_i}$ and the right-hand sides of $\mathcal{P}^c$ do not contain any symbols from $\Sigma_c^{\mathcal{P}}$. Hence, we get $t'|_{\pi_i} \xrightarrow{e_i}{}^{*}_{\mathcal{P}^c} q'_i$. Thus, we obtain

$$[\![\mathfrak{a}^{\mathsf{s}}(t'|_{\pi_i})]\!] \xrightarrow{e_i}{}^{*}_{\mathcal{P}^c} [\![\mathfrak{a}^{\mathsf{s}}(q'_i)]\!] \tag{101}$$

by Lemma 72. As $\mathcal{P}os_d(t') = \{\pi_1, \ldots, \pi_n\}$ and $t'|_{\pi_i} = \mathfrak{a}^{\mathsf{s}}(t'|_{\pi_i})$ (as t' does not have nested defined symbols and $\mathrm{root}(t'|_{\pi_i})$ is defined), we have

$$\mathfrak{a}^{\mathsf{s}}(t') = \sum\nolimits_{1 \leq i \leq n} t'|_{\pi_i} = \sum\nolimits_{1 \leq i \leq n} \mathfrak{a}^{\mathsf{s}}(t'|_{\pi_i}). \tag{102}$$

Thus, we get

$$\begin{array}{lll}
 & t & \\
= & [\![t]\!] & \text{as } t \text{ is nat-basic} \\
= & [\![\mathfrak{a}^{\mathsf{s}}(t)]\!] & \text{by (96)} \\
\xrightarrow{[\![m_1+\mathfrak{c}(t')]\!]}_{\mathcal{P}^c} & [\![\mathfrak{a}^{\mathsf{s}}(t')]\!] & \text{by (97)} \\
= & \sum_{1\leq i\leq n} [\![\mathfrak{a}^{\mathsf{s}}(t'|_{\pi_i})]\!] & \text{by (102)} \\
\xrightarrow{e_1+\ldots+e_n}{}^{*}_{\mathcal{P}^c} & \sum_{1\leq i\leq n} [\![\mathfrak{a}^{\mathsf{s}}(q_i')]\!] & \text{by (101)}
\end{array}$$

It remains to show $[\![m_1 + \mathfrak{c}(t')]\!] + e_1 + \ldots + e_n \geq m$. We have:

$$\begin{array}{lll}
 & [\![m_1 + \mathfrak{c}(t')]\!] + e_1 + \ldots + e_n \geq m & \\
\iff & m_1 + [\![\mathfrak{c}(t')]\!] + e_1 + \ldots + e_n \geq m_1 + m_2 & \text{by (95)} \\
\iff & [\![\mathfrak{c}(t')]\!] + e_1 + \ldots + e_n \geq m_2 & \\
\iff & [\![\mathfrak{c}(t')]\!] + e_1 + \ldots + e_n \geq d_1 + \ldots + d_n + m_{2,2} & \text{by (99)} \\
\Longleftarrow & [\![\mathfrak{c}(t')]\!] \geq m_{2,2} & \text{by (100)} \\
\iff & \mathsf{true} & (\dagger)
\end{array}$$

The step marked with (†) holds because of Lemma 57. To see this, recall that $m_{2,2}$ is the cost of normalizing $[\![t'[q_1]_{\pi_1} \ldots [q_n]_{\pi_n}]\!]$ with $\widehat{\mathcal{P}}$ and q_i is a $\mathcal{P}\cup\widehat{\mathcal{P}}$-normal form of $t'|_{\pi_i}$ for each $1 \leq i \leq n$. Hence, Lemma 57 applies. □

B.2.6 Properties of $\mathcal{P}_{\mathsf{sz}}$ and $\mathcal{P}_{\mathsf{rt,sz}}$

Now we can show the soundness of our construction of $\mathcal{P}_{\mathsf{sz}}$ when considering size bounds of $\mathcal{P} \cup \widehat{\mathcal{P}}$.

Theorem 74 (Soundness of $\mathcal{P}_{\mathsf{sz}}$ for Size Bounds of $\mathcal{P} \cup \widehat{\mathcal{P}}$). *Let $\mathcal{P}$ be an RNTS with size and runtime bounds* sz *and* rt*, where $\mathcal{P}$ does not have nested defined symbols. Let $\mathcal{P}_{\mathsf{sz}}$ be defined as in Thm. 27. Then every size bound* sz' *for $\mathcal{P}_{\mathsf{sz}}$ (where we assume $\mathsf{sz}'(f) = \omega$ if f does not occur in $\mathcal{P}_{\mathsf{sz}}$) is also a size bound for $\mathcal{P} \cup \widehat{\mathcal{P}}$.*

Proof. The size bound sz' for $\mathcal{P}_{\mathsf{sz}}$ is also a size bound for $\mathcal{P}_{\mathsf{sz}} \cup \widehat{\mathcal{P}}$. The reason is that $\mathcal{P}_{\mathsf{sz}}$ does not contain symbols from $\Sigma_c^{\mathcal{P}}$ and thus, we have $\mathsf{sz}'(f) = \omega$ for all $f \in \Sigma_c^{\mathcal{P}}$. Moreover, the rules from $\widehat{\mathcal{P}}$ cannot be applied in $(\mathcal{P}_{\mathsf{sz}} \cup \widehat{\mathcal{P}})$-rewrite sequences that start with a nat-basic term whose root is from $\Sigma_d^{\mathcal{P}}$.

W.l.o.g., let $\mathsf{sz}(f) = \omega$ for all $f \in \Sigma_d^{\mathcal{P}}$. Then sz is clearly still a size bound for $\mathcal{P}$ and this change of sz does not influence the construction of $\mathcal{P}^i$, $\mathcal{P}^o$, or $\widehat{\mathcal{P}}$ since they only rely on the size bound for constructors (as $\mathcal{P}$ does not have nested defined symbols). Moreover w.l.o.g., let $\mathsf{rt}(f) = \omega$ for all $f \in \Sigma_d^{\mathcal{P}}$. Again, this does not influence the construction of $\mathcal{P}^i$ or $\widehat{\mathcal{P}}$ as they only rely on the runtime bound for constructors. By Lemma 48, then sz is also a size bound for $\mathcal{P}^i \cup \widehat{\mathcal{P}}$ and thus in particular for $\mathcal{P}^i$. Clearly, $\mathcal{P}^i$ does not have Σ-symbols below defined symbols. So by Thm. 70, every size bound for $(\mathcal{P}^i)^o \cup \widehat{\mathcal{P}^i}$ is also a size bound for $\mathcal{P}^i \cup \widehat{\mathcal{P}^i}$. With Corollary 47, we have $(\mathcal{P}^i)^o = \mathcal{P}_{\mathsf{sz}}$, i.e., every size bound for $\mathcal{P}_{\mathsf{sz}} \cup \widehat{\mathcal{P}^i}$ is also a size bound for $\mathcal{P}^i \cup \widehat{\mathcal{P}^i}$. As sz' is a size bound for $\mathcal{P}_{\mathsf{sz}} \cup \widehat{\mathcal{P}}$, it is clearly also a size bound for $\mathcal{P}_{\mathsf{sz}} \cup \widehat{\mathcal{P}^i}$ (as $\widehat{\mathcal{P}^i} \subseteq \widehat{\mathcal{P}}$). Thus, it is also a size bound for $\mathcal{P}^i \cup \widehat{\mathcal{P}^i}$. Clearly, defined symbols from $\mathcal{P}^i \cup \widehat{\mathcal{P}}$ that do not occur in $\mathcal{P}^i \cup \widehat{\mathcal{P}^i}$ do not occur in $\mathcal{P}_{\mathsf{sz}}$. As we assumed $\mathsf{sz}'(f) = \omega$ if f does not occur in $\mathcal{P}_{\mathsf{sz}}$, sz' is also a size bound for $\mathcal{P}^i \cup \widehat{\mathcal{P}}$.

Finally, by Thm. 66, every size bound for $\mathcal{P}^i \cup \widehat{\mathcal{P}}$ is a size bound for $\mathcal{P} \cup \widehat{\mathcal{P}}$. □

Similarly, the following theorem shows the soundness of our construction of $\mathcal{P}_{\mathsf{rt},\mathsf{sz}}$ when considering time bounds of $\mathcal{P} \cup \widehat{\mathcal{P}}$.

Theorem 75 (Soundness of $\mathcal{P}_{\mathsf{rt},\mathsf{sz}}$ for Time Bounds of $\mathcal{P} \cup \widehat{\mathcal{P}}$). *Let $\mathcal{P}$ be an RNTS with size and runtime bounds sz and rt, where $\mathcal{P}$ does not have nested defined symbols and sz is also a size bound for $\mathcal{P}^i$. Then every time bound rt' for $\mathcal{P}_{\mathsf{rt},\mathsf{sz}}$ (where we assume $\mathsf{rt}'(f) = \omega$ if f does not occur in $\mathcal{P}_{\mathsf{rt},\mathsf{sz}}$) is also a time bound for $\mathcal{P} \cup \widehat{\mathcal{P}}$.*

Proof. The time bound rt' for $\mathcal{P}_{\mathsf{rt},\mathsf{sz}}$ is also a time bound for $\mathcal{P}'_{\mathsf{rt},\mathsf{sz}}$ by Corollary 52 and hence it is also a time bound for $\mathcal{P}'_{\mathsf{rt},\mathsf{sz}} \cup \widehat{\mathcal{P}}$. The reason is that $\mathcal{P}_{\mathsf{rt},\mathsf{sz}}$ does not contain symbols from $\Sigma_c^{\mathcal{P}}$ and thus, we have $\mathsf{rt}'(f) = \omega$ for all $f \in \Sigma_c^{\mathcal{P}}$. Moreover, the rules from $\widehat{\mathcal{P}}$ cannot be applied in $(\mathcal{P}'_{\mathsf{rt},\mathsf{sz}} \cup \widehat{\mathcal{P}})$-rewrite sequences that start with a nat-basic term whose root is from $\Sigma_d^{\mathcal{P}}$.

W.l.o.g., let $\mathsf{rt}(f) = \omega$ for all $f \in \Sigma_d^{\mathcal{P}}$. Then rt is clearly still a time bound for $\mathcal{P}$ and this change of rt does not influence the construction of $\mathcal{P}^i$, $\mathcal{P}^c$, or $\widehat{\mathcal{P}}$ since they only rely on the time bound for constructors (as $\mathcal{P}$ does not have nested defined symbols). Hence, by Lemma 48, rt is also a time bound for $\mathcal{P}^i \cup \widehat{\mathcal{P}}$ and thus in particular also for $\mathcal{P}^i$. Moreover, by the prerequisites, sz is a size bound for $\mathcal{P}^i$. Clearly, $\mathcal{P}^i$ does not have Σ-symbols below defined symbols. So by Thm. 73, every time bound for $(\mathcal{P}^i)^c \cup \widehat{\mathcal{P}^i}$ is also a time bound for $\mathcal{P}^i \cup \widehat{\mathcal{P}^i}$.

With Corollary 47, we have $(\mathcal{P}^i)^c = \mathcal{P}'_{\mathsf{rt},\mathsf{sz}}$, i.e., every time bound for $\mathcal{P}'_{\mathsf{rt},\mathsf{sz}} \cup \widehat{\mathcal{P}^i}$ is also a time bound for $\mathcal{P}^i \cup \widehat{\mathcal{P}^i}$. As rt' is a time bound for $\mathcal{P}'_{\mathsf{rt},\mathsf{sz}} \cup \widehat{\mathcal{P}}$, it is clearly also a time bound for $\mathcal{P}'_{\mathsf{rt},\mathsf{sz}} \cup \widehat{\mathcal{P}^i}$ (as $\widehat{\mathcal{P}^i} \subseteq \widehat{\mathcal{P}}$). Thus, it is also a time bound for $\mathcal{P}^i \cup \widehat{\mathcal{P}^i}$. Clearly, defined symbols from $\mathcal{P}^i \cup \widehat{\mathcal{P}}$ that do not occur in $\mathcal{P}^i \cup \widehat{\mathcal{P}^i}$ do not occur in $\mathcal{P}_{\mathsf{rt},\mathsf{sz}}$. As we assumed $\mathsf{rt}'(f) = \omega$ if f does not occur in $\mathcal{P}_{\mathsf{rt},\mathsf{sz}}$, rt' is also a time bound for $\mathcal{P}^i \cup \widehat{\mathcal{P}}$.

Finally, by Thm. 66, every time bound for $\mathcal{P}^i \cup \widehat{\mathcal{P}}$ is a time bound for $\mathcal{P} \cup \widehat{\mathcal{P}}$. □

In contrast to Thm. 74, Thm. 75 requires that sz is also a size bound for $\mathcal{P}^i$. The following auxiliary lemma allows us to ensure this requirement.

Lemma 76 (Size Bounds for $\mathcal{P}$ and $\mathcal{P}^i$). *Let $\mathcal{P}$ be an RNTS with size and runtime bounds sz and rt and without nested defined symbols and let sz' be a size bound for $\mathcal{P}_{\mathsf{sz}}$ such that*

$$[\![\mathsf{sz}'(f)\sigma]\!] \leq [\![\mathsf{sz}(f)\sigma]\!] \text{ for all } f \in \Sigma \text{ and all natural substitutions } \sigma. \tag{103}$$

Then sz' *is a size bound for* $\mathcal{P}^i_{\mathsf{rt},\mathsf{sz}'}$.

Proof. In the following, we write $\mathcal{P}^i_{\mathsf{sz}'}$ instead of $\mathcal{P}^i_{\mathsf{rt},\mathsf{sz}'}$. Let s be an arbitrary term such that $s \to_{\mathcal{P}^i_{\mathsf{sz}'}} t$ and let

$$\ell \to \mathfrak{a}^{\mathsf{i}}(r) \left[\varphi \wedge \psi^{\mathsf{i}}_{\mathsf{sz}'}(r)\right]$$

and σ be the rule and the substitution used for this rewrite step. We omit the weights here, since they are not of interest. By construction, we have $\ell \to \mathfrak{a}^{\mathsf{i}}(r) \left[\varphi \wedge \psi^{\mathsf{i}}_{\mathsf{sz}}(r)\right] \in \mathcal{P}^i_{\mathsf{sz}}$. By definition of ψ^{i}, $[\![(\varphi \wedge \psi^{\mathsf{i}}_{\mathsf{sz}'}(r))\sigma]\!] = \mathsf{true}$ and (103) implies $[\![(\varphi \wedge \psi^{\mathsf{i}}_{\mathsf{sz}}(r))\sigma]\!] = \mathsf{true}$. Hence, we get $s \to_{\mathcal{P}^i_{\mathsf{sz}}} t$. So $s \to_{\mathcal{P}^i_{\mathsf{sz}'}} t$ implies $s \to_{\mathcal{P}^i_{\mathsf{sz}}} t$ and hence

$$\text{every size bound for } \mathcal{P}^i_{\mathsf{sz}} \text{ is also a size bound for } \mathcal{P}^i_{\mathsf{sz}'}. \tag{104}$$

Now we want to apply Thm. 74 to deduce that every size bound for $(\mathcal{P}^i_{\mathsf{sz}})_{\mathsf{sz}}$ is also a size bound for $\mathcal{P}^i_{\mathsf{sz}} \cup \widehat{\mathcal{P}^i_{\mathsf{sz}}}$. To apply Thm. 74, we need to know that $\mathcal{P}^i_{\mathsf{sz}}$ does not have nested defined symbols (which is the case by Def. 46) and that sz is a size bound for $\mathcal{P}^i_{\mathsf{sz}}$. W.l.o.g., assume $\mathsf{sz}(f) = \omega$ for each $f \in \Sigma^{\mathcal{P}}_d = \Sigma^{\mathcal{P}^i_{\mathsf{sz}}}_d$. Clearly, this assumption does not affect the construction of $\mathcal{P}^i_{\mathsf{sz}}$ and $(\mathcal{P}^i_{\mathsf{sz}})_{\mathsf{sz}}$, which just require size bounds for constructors. Then, sz is trivially a size bound for $\mathcal{P}^i_{\mathsf{sz}}$ and hence Thm. 74 applies. By Corollary 47, we have $(\mathcal{P}^i_{\mathsf{sz}})_{\mathsf{sz}} = ((\mathcal{P}^i_{\mathsf{sz}})^i_{\mathsf{sz}})^o_{\mathsf{sz}}$. Moreover, as an immediate consequence of Def. 46, we have $(\mathcal{P}^i_{\mathsf{sz}})^i_{\mathsf{sz}} = \mathcal{P}^i_{\mathsf{sz}}$. Hence, by Thm. 74 every size bound for $(\mathcal{P}^i_{\mathsf{sz}})_{\mathsf{sz}} = ((\mathcal{P}^i_{\mathsf{sz}})^i_{\mathsf{sz}})^o_{\mathsf{sz}} = (\mathcal{P}^i_{\mathsf{sz}})^o_{\mathsf{sz}}$ is also a size bound for $\mathcal{P}^i_{\mathsf{sz}} \cup \widehat{\mathcal{P}^i_{\mathsf{sz}}}$. By Corollary 47, we have $(\mathcal{P}^i_{\mathsf{sz}})^o_{\mathsf{sz}} = \mathcal{P}_{\mathsf{sz}}$ and hence every size bound for $\mathcal{P}_{\mathsf{sz}}$ is also a size bound for $\mathcal{P}^i_{\mathsf{sz}} \cup \widehat{\mathcal{P}^i_{\mathsf{sz}}}$. Since sz' is a size bound for $\mathcal{P}_{\mathsf{sz}}$, this implies that sz' is a size bound for $\mathcal{P}^i_{\mathsf{sz}} \cup \widehat{\mathcal{P}^i_{\mathsf{sz}}}$. As $\mathcal{P}^i_{\mathsf{sz}}$ is a subset of $\mathcal{P}^i_{\mathsf{sz}} \cup \widehat{\mathcal{P}^i_{\mathsf{sz}}}$, sz' is also a size bound for $\mathcal{P}^i_{\mathsf{sz}}$. With (104), this proves the claim.

$\square$

Now we can finally prove the soundness of Thm. 27.

Theorem 27 (Transformation of RNTSs to ITSs). *Let $\mathcal{Q}$ be an RNTS with size and runtime bounds* sz *and* rt *and let $\mathcal{P} = \mathcal{Q} \setminus (\mathcal{Q}^{g_1} \cup \ldots \cup \mathcal{Q}^{g_m})$, where $g_1, \ldots, g_m \in \Sigma$ and $\mathcal{Q}^{g_i}$ is the sub-RNTS of $\mathcal{Q}$ induced by g_i. We define*

$$\mathcal{P}_{\mathsf{sz}} \quad = \quad \{\, \ell \xrightarrow{w} \mathfrak{a}^{\mathsf{o}}_{\mathsf{sz}}(\mathfrak{a}^{\mathsf{i}}(r)) \left[\varphi \wedge \psi^{\mathsf{i}}_{\mathsf{sz}}(r)\right] \quad | \quad \ell \xrightarrow{w} r\,[\varphi] \in \mathcal{P} \,\}$$

Let sz' *be a size bound for* $\mathcal{P}_{\mathsf{sz}}$ *where* $\mathsf{sz}'(f) = \mathsf{sz}(f)$ *for all $f \in \Sigma \setminus \Sigma^{\mathcal{P}}_d$. If $\mathcal{P}$ does not have nested defined symbols, then* sz' *is a size bound for $\mathcal{Q}$.*

To obtain a runtime bound for $\mathcal{Q}$, we define an RNTS $\mathcal{P}_{\mathsf{rt},\mathsf{sz}'}$. To this end, we define the cost *of a term as $\mathfrak{c}_{\mathsf{rt},\mathsf{sz}'}(x) = 0$ for $x \in \mathcal{V}$ and*

$$\mathfrak{c}_{\mathsf{rt},\mathsf{sz}'}(g(s_1,\ldots,s_n)) = \begin{cases} \sum_{1\le j\le n} \mathfrak{c}_{\mathsf{rt},\mathsf{sz}'}(s_j) + \mathsf{rt}(g)\,\{x_j/\mathsf{sz}'(s_j) \mid 1 \le j \le n\} & \text{if } g \in \Sigma^{\mathcal{P}}_c \\ \sum_{1\le j\le n} \mathfrak{c}_{\mathsf{rt},\mathsf{sz}'}(s_j) & \text{otherwise} \end{cases}$$

Now $\mathcal{P}_{\mathsf{rt},\mathsf{sz}'} = \{\ell \xrightarrow{w+\mathsf{c}_{\mathsf{rt},\mathsf{sz}'}(r)} \sum_{\pi \in \mathcal{P}os_d(r)} \mathfrak{a}^{\mathsf{i}}(r|_\pi) \;\; [\varphi \wedge \psi^{\mathsf{i}}_{\mathsf{sz}'}(r)] \;\mid\; \ell \xrightarrow{w} r\,[\varphi] \in \mathcal{P}\}$*. Then every runtime bound* rt' *for* $\mathcal{P}_{\mathsf{rt},\mathsf{sz}'}$ *with* $\mathsf{rt}'(f) = \mathsf{rt}(f)$ *for all* $f \in \Sigma \setminus \Sigma_d^{\mathcal{P}}$ *is a runtime bound for* $\mathcal{Q}$*. Here, all occurrences of* ω *in* $\mathcal{P}_{\mathsf{sz}}$ *or* $\mathcal{P}_{\mathsf{rt},\mathsf{sz}'}$ *are replaced by pairwise different fresh variables.*

Proof. By Thm. 74, every size bound for $\mathcal{P}_{\mathsf{sz}}$ is a size bound for $\mathcal{P} \cup \widehat{\mathcal{P}}$. By Thm. 61, every size bound for $\mathcal{P} \cup \widehat{\mathcal{P}}$ is a size bound for $\mathcal{Q}$, as $\Sigma_d^{\mathcal{P}} \cap \Sigma^{\mathcal{Q}\setminus\mathcal{P}} = \varnothing$ by construction.

Now we want to apply Lemma 76 to establish that sz' is a size bound for $\mathcal{P}^i_{\mathsf{sz}'}$. To this end, w.l.o.g. assume $\mathsf{sz}(f) = \mathsf{rt}(f) = \omega$ for all $f \in \Sigma_d^{\mathcal{P}}$ as in the proof of Thm. 74. Then we clearly have $[\![\mathsf{sz}'(f)\sigma]\!] \leq [\![\mathsf{sz}(f)\sigma]\!]$ for all $f \in \Sigma$ and all natural substitutions σ by definition of sz'. Hence, by Lemma 76, sz' is a size bound for $\mathcal{P}^i_{\mathsf{sz}'}$. With Thm. 75, this implies that every time bound for $\mathcal{P}_{\mathsf{rt},\mathsf{sz}'}$ is a time bound for $\mathcal{P} \cup \widehat{\mathcal{P}}$. By Thm. 61, every time bound for $\mathcal{P} \cup \widehat{\mathcal{P}}$ is a time bound for $\mathcal{Q}$, as $\Sigma_d^{\mathcal{P}} \cap \Sigma^{\mathcal{Q}\setminus\mathcal{P}} = \varnothing$ by construction. ☐

Theorem 30 (Alg. 1 is Sound). *Let* $\mathcal{P}$ *be an RNTS and let* rt *and* sz *be the result of Alg. 1 for* $\mathcal{P}$*. Then* rt *is a runtime bound and* sz *is a size bound for* $\mathcal{P}$*.*

Proof. Let $\mathcal{P}_{in}$ be the analyzed RNTS. We prove that

$$\mathsf{sz} \text{ is a size bound for } \mathcal{P}_{in} \quad \text{and} \tag{105}$$
$$\mathsf{rt} \text{ is a runtime bound for } \mathcal{P}_{in} \tag{106}$$

are loop invariants. Then our claim follows, since the algorithm obviously terminates. Initially, sz and rt are clearly size and runtime bounds for $\mathcal{P}_{in}$ by construction. If $\mathcal{P}_{in}$ has nested recursion, the algorithm returns these initial bounds in Step 2, so from now on we assume that $\mathcal{P}_{in}$ does not have nested recursion. Assume that (105) and (106) hold at the beginning of the loop body. First consider the case that $(\mathcal{P}^f_{\mathsf{sz}})_{\mathsf{size}}$ is not well defined (as $\mathcal{P}^f_{\mathsf{sz}}$ does not satisfy the preconditions of Thm. 45 (i.e., the generalization of Thm. 19)). Then the algorithm soundly returns rt and sz.

Now consider the case that $(\mathcal{P}^f_{\mathsf{sz}})_{\mathsf{size}}$ is well defined. When entering the loop for the first time, we have $\mathcal{P} = \mathcal{P}_{in}$. But in general, we have $\mathcal{P} = \mathcal{P}_{in} \setminus (\mathcal{P}^{g_1}_{in} \cup \ldots \cup \mathcal{P}^{g_n}_{in})$ for some $g_1, \ldots, g_n \in \Sigma$ such that $g_i \not\sqsupset g$ for all $g \in \Sigma_d^{\mathcal{P}}$. For the bottom symbol f of $\mathcal{P}$, we have $\mathcal{P}^f = \mathcal{P}^f_{in} \setminus (\mathcal{P}^{g_1}_{in} \cup \ldots \cup \mathcal{P}^{g_n}_{in})$.

By Thm. 45, sz_f is a size bound for $\mathcal{P}^f_{\mathsf{sz}}$. Let sz' and rt' be the values of sz and rt at the end of the loop body. As sz' is like sz_f for all defined symbols of $\mathcal{P}^f$ and like sz for all other symbols of Σ, we obtain that

$$\mathsf{sz}' \text{ is a size bound for } \mathcal{P}^f_{in}$$

by Thm. 27. As sz is a size bound for $\mathcal{P}_{in}$ and sz and sz' only differ for symbols from $\Sigma_d^{\mathcal{P}^f} \subseteq \Sigma_d^{\mathcal{P}^f_{in}}$, sz' is a size bound for $\mathcal{P}_{in}$, which finishes the proof of (105).

It remains to show that (106) is a loop invariant. Recall that we have $\mathcal{P}^f = \mathcal{P}^f_{in} \setminus (\mathcal{P}^{g_1}_{in} \cup \ldots \cup \mathcal{P}^{g_n}_{in})$. Note that rt_f is a runtime bound for $\mathcal{P}^f_{\mathsf{rt},\mathsf{sz}'}$. As rt' is like

rt_f for all defined symbols of $\mathcal{P}^f$ and like rt for all other symbols of Σ, we obtain that

$$\mathsf{rt}' \text{ is a runtime bound for } \mathcal{P}^f_{in}$$

by Thm. 27. As rt is a runtime bound for $\mathcal{P}_{in}$ and rt and rt' only differ for symbols from $\Sigma_d^{\mathcal{P}^f} \subseteq \Sigma_d^{\mathcal{P}^f_{in}}$, rt' is a runtime bound for $\mathcal{P}_{in}$, which finishes the proof of (106). $\square$

B.3 Proofs for Appendix A

Theorem 33 (Soundness of Abstraction $\wr\cdot\int_{\mathrm{con}}$). *Let $\mathcal{R}/\mathcal{S}$ be a well-typed constructor system and let $\mathrm{con}_{\max} = \max(\{1\} \cup \{\mathrm{con}(f) \mid f \in \Sigma_c^0\})$. Let $\mathcal{N}$ be a terminating variant of $\mathcal{S}$ such that $\mathcal{R}/(\mathcal{S} \cup \mathcal{N})$ is well typed and completely defined. Then we have $\mathrm{irc}_{\mathcal{R}/\mathcal{S}}(n) \leq \mathrm{irc}_{\wr\mathcal{R}/(\mathcal{S}\cup\mathcal{N})\int_{\mathrm{con}}}(\mathrm{con}_{\max} \cdot n)$ for all $n \in \mathbb{N}$.*

Proof. Lemma 44 can easily be adapted to the improved size abstraction $\wr\cdot\int_{\mathrm{con}}$. Thus, for any well-typed ground term s, $s \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}} t$ implies $[\![\wr s\int_{\mathrm{con}}]\!] \to_{\wr\mathcal{R}/\mathcal{S}\int_{\mathrm{con}}}{}^{+} [\![\wr t\int_{\mathrm{con}}]\!]$, where the sum of the weights of the rewrite steps is 1. Therefore, $\mathrm{dh}(t, \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}}) \leq \mathrm{dhw}([\![\wr t\int_{\mathrm{con}}]\!], \to_{\wr\mathcal{R}/\mathcal{S}\int_{\mathrm{con}}})$ holds for all well-typed ground terms t.

Now $\mathrm{irc}_{\mathcal{R}/\mathcal{S}}(n) \leq \mathrm{irc}_{\wr\mathcal{R}/(\mathcal{S}\cup\mathcal{N})\int_{\mathrm{con}}}(\mathrm{con}_{\max} \cdot n)$ can be shown similar as in the proof of Thm. 13. The only main difference is the following step:

$$\begin{aligned} &\sup\{\mathrm{dhw}([\![\wr s\int_{\mathrm{con}}]\!], \xrightarrow{\mathrm{i}}_{\wr\mathcal{R}/(\mathcal{S}\cup\mathcal{N})\int_{\mathrm{con}}}) \mid s \text{ well typed, basic, and ground}, |s| \leq n\} \\ \leq\ &\sup\{\mathrm{dhw}(q, \to_{\wr\mathcal{R}/(\mathcal{S}\cup\mathcal{N})\int_{\mathrm{con}}}) \mid q \text{ nat-basic}, \|q\| \leq \mathrm{con}_{\max} \cdot n\} \end{aligned}$$

To see why this step is correct, note that $[\![\wr s\int_{\mathrm{con}}]\!] \leq \mathrm{con}_{\max} \cdot |s|$ holds for all constructor ground terms s. Thus, we have $\|[\![\wr s\int_{\mathrm{con}}]\!]\| \leq \mathrm{con}_{\max} \cdot |s|$ for all basic ground terms s. $\square$

Theorem 36 (Narrowing for Complexity). *Let $\mathcal{R}/\mathcal{S}$ be a completely defined constructor system, $\ell \to r \in \mathcal{R} \cup \mathcal{S}$, $r|_\pi = f(\ldots)$ for some $f \in \Sigma_d^{\mathcal{R}\cup\mathcal{S}}$, and let $r|_\mu$ be a basic term for some $\mu > \pi$. Let $\ell_1 \to r_1, \ldots, \ell_m \to r_m \in \mathcal{R} \cup \mathcal{S}$ be all (variable-renamed) rules where ℓ_i unifies with $r|_\mu$ and let $\sigma_i = mgu(r|_\mu, \ell_i)$ for $1 \leq i \leq m$. Let*

$$\mathcal{R}' = (\mathcal{R} \setminus \{\ell \to r\}) \cup \{\ell\sigma_i \to r[r_i]_\mu\sigma_i \mid 1 \leq i \leq m\} \quad \textit{and} \quad \mathcal{S}' = \mathcal{S} \setminus \{\ell \to r\}.$$

Then we have $\mathrm{dh}(t, \xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}}) \leq 2 \cdot \mathrm{dh}(t, \xrightarrow{\mathrm{i}}_{\mathcal{R}'/\mathcal{S}'})$ for all ground terms t.

Proof. A maximal (finite or infinite) $\xrightarrow{\mathrm{i}}_{\mathcal{R}/\mathcal{S}}$-derivation of t has the form

$$t = t_0 \xrightarrow{\mathcal{R}\cup\mathcal{S}}_{\mathcal{P}_0} t_1 \xrightarrow{\mathcal{R}\cup\mathcal{S}}_{\mathcal{P}_1} t_2 \xrightarrow{\mathcal{R}\cup\mathcal{S}}_{\mathcal{P}_2} \ldots$$

where $\mathcal{P}_j \in \{\mathcal{R}, \mathcal{S}\}$ for all j. Let i be the first step where the reduction $t_i \xrightarrow{\mathcal{R}\cup\mathcal{S}}_{\mathcal{P}_i} t_{i+1}$ is performed via the rule $\ell \to r$. Thus, $t_i = t_i[\ell\rho]_\kappa \xrightarrow{\mathcal{R}\cup\mathcal{S}}_{\mathcal{P}_i} t_i[r\rho]_\kappa = t_{i+1}$ for some substitution ρ and some position $\kappa \in \mathcal{P}os(t_i)$. Note that ρ instantiates all variables of ℓ by constructor ground terms. The reason is that all other ground terms are not in normal form w.r.t. $\mathcal{R} \cup \mathcal{S}$, as $\mathcal{R}/\mathcal{S}$ is completely defined.

Since $r|_\pi$ is a basic term, $t_{i+1}|_{\kappa\pi} = r\rho|_\pi = r|_\pi\rho$ is a basic ground term as ρ instantiates r's variables by constructor ground terms. Since $\mathcal{R}/\mathcal{S}$ is completely defined, $t_{i+1}|_{\kappa\pi}$ is not in normal form w.r.t. $\mathcal{R} \cup \mathcal{S}$.

As we consider innermost rewriting and the derivation is maximal, w.l.o.g. we can assume that the next rewriting step is performed below κ (otherwise one can rearrange the derivation steps without changing the length of the reduction). Then we have

$$t_i = t_i[\ell\rho]_\kappa \xrightarrow{\mathcal{R}\cup\mathcal{S}}_{\mathcal{P}_i} t_i[r\rho]_\kappa = t_i[r\rho[\ell'\delta]_\pi]_\kappa \xrightarrow{\mathcal{R}\cup\mathcal{S}}_{\mathcal{P}_{i+1}} \to t_i[r\rho[r'\delta]_\pi]_\kappa = t_{i+2}$$

for some rule $\ell' \to r' \in \mathcal{R} \cup \mathcal{S}$ and some substitution δ.

W.l.o.g. we can assume that $\ell \to r$ and $\ell' \to r'$ have distinct variables. We know that $r|_\pi\rho = \ell'\delta$. Instead of using δ, we can extend ρ to the variables of ℓ' and obtain $r|_\pi\rho = \ell'\rho$. Thus, ρ unifies $r|_\pi$ and ℓ'. Hence, there exists an mgu σ of $r|_\pi$ and ℓ' and we have $\rho = \sigma\rho'$ for some substitution ρ'.

So we have $t_i|_\kappa = \ell\rho = (\ell\sigma)\rho'$, i.e., the narrowed rule $\ell\sigma \to r[r']_\pi\sigma$ is applicable. Moreover, its application is again an innermost step w.r.t. $\mathcal{R} \cup \mathcal{S}$ since $\ell\sigma\rho' = \ell\rho$ has no redex as a proper subterm. Applying this narrowed rule to t_i results in

$$\begin{aligned} t_i = t_i[(\ell\sigma)\rho']_\kappa &\xrightarrow{\mathcal{R}\cup\mathcal{S}}_{\mathcal{R}'} t_i[(r[r']_\pi\sigma)\rho']_\kappa \\ &= t_i[r[r']_\pi\rho]_\kappa = t_i[r\rho[r'\rho]_\pi]_\kappa = t_i[r\rho[r'\delta]_\pi]_\kappa = t_{i+2} \end{aligned}$$

Thus, we have $t_i \xrightarrow{\mathcal{R}\cup\mathcal{S}}_{\mathcal{R}'} t_{i+2}$. We proceed in the same way for all such steps i. Note that by the construction of the narrowed rules, a ground term is a normal form w.r.t. $\mathcal{R} \cup \mathcal{S}$ iff it is a normal form w.r.t. $\mathcal{R}' \cup \mathcal{S}'$. Hence, every rewrite step with $\xrightarrow{\mathcal{R}\cup\mathcal{S}}_{\mathcal{P}}$ is also a rewrite step with $\xrightarrow{\mathcal{R}'\cup\mathcal{S}'}_{\mathcal{P}}$. Therefore, our construction yields a derivation w.r.t. $\xrightarrow{\mathsf{i}}_{\mathcal{R}'/\mathcal{S}'}$ which is at least half as long as the original $\xrightarrow{\mathsf{i}}_{\mathcal{R}/\mathcal{S}}$-derivation, i.e., we obtain $\mathrm{dh}(t, \xrightarrow{\mathsf{i}}_{\mathcal{R}/\mathcal{S}}) \leq 2 \cdot \mathrm{dh}(t, \xrightarrow{\mathsf{i}}_{\mathcal{R}'/\mathcal{S}'})$. $\square$

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`

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
2015-01 * Fachgruppe Informatik: Annual Report 2015
2015-02 Dominik Franke: Testing Life Cycle-related Properties of Mobile Applications
2015-05 Florian Frohn, Jürgen Giesl, Jera Hensel, Cornelius Aschermann, and Thomas Ströder: Inferring Lower Bounds for Runtime Complexity

2015-06 Thomas Ströder and Wolfgang Thomas (Editors): Proceedings of the Young Researchers' Conference "Frontiers of Formal Methods"
2015-07 Hilal Diab: Experimental Validation and Mathematical Analysis of Cooperative Vehicles in a Platoon
2015-08 Mathias Pelka, Jó Agila Bitsch, Horst Hellbrück, and Klaus Wehrle (Editors): Proceedings of the 1st KuVS Expert Talk on Localization
2015-09 Xin Chen: Reachability Analysis of Non-Linear Hybrid Systems Using Taylor Models
2015-11 Stefan Wüller, Marián Kühnel, and Ulrike Meyer: Information Hiding in the Public RSA Modulus
2015-12 Christoph Matheja, Christina Jansen, and Thomas Noll: Tree-like Grammars and Separation Logic
2015-13 Andreas Polzer: Ansatz zur variantenreichen und modellbasierten Entwicklung von eingebetteten Systemen unter Berücksichtigung regelungs- und softwaretechnischer Anforderungen
2015-14 Niloofar Safiran and Uwe Naumann: Symbolic vs. Algorithmic Differentiation of GSL Integration Routines
2016-01 * Fachgruppe Informatik: Annual Report 2016
2016-02 Ibtissem Ben Makhlouf: Comparative Evaluation and Improvement of Computational Approaches to Reachability Analysis of Linear Hybrid Systems
2016-03 Florian Frohn, Matthias Naaf, Jera Hensel, Marc Brockschmidt, and Jürgen Giesl: Lower Runtime Bounds for Integer Programs
2016-04 Jera Hensel, Jürgen Giesl, Florian Frohn, and Thomas Ströder: Proving Termination of Programs with Bitvector Arithmetic by Symbolic Execution
2016-05 Mathias Pelka, Grigori Goronzy, Jó Agila Bitsch, Horst Hellbrück, and Klaus Wehrle (Editors): Proceedings of the 2nd KuVS Expert Talk on Localization
2016-06 Martin Henze, René Hummen, Roman Matzutt, Klaus Wehrle: The SensorCloud Protocol: Securely Outsourcing Sensor Data to the Cloud
2016-07 Sebastian Biallas : Verification of Programmable Logic Controller Code using Model Checking and Static Analysis
2016-09 Thomas Ströder, Jürgen Giesl, Marc Brockschmidt, Florian Frohn, Carsten Fuhs, Jera Hensel, Peter Schneider-Kamp, and Cornelius Aschermann: Automatically Proving Termination and Memory Safety for Programs with Pointer Arithmetic
2016-10 Stefan Wüller, Ulrike Meyer, and Susanne Wetzel: Towards Privacy-Preserving Multi-Party Bartering
2017-01 * Fachgruppe Informatik: Annual Report 2017
2017-02 Florian Frohn and Jürgen Giesl: Analyzing Runtime Complexity via Innermost Runtime Complexity

* These reports are only available as a printed version.

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