

FROM INNERMOST TO FULL PROBABILISTIC TERM REWRITING: ALMOST-SURE TERMINATION, COMPLEXITY, AND MODULARITY

JAN-CHRISTOPH KASSING  AND JÜRGEN GIESL 

RWTH Aachen University, Aachen, Germany
e-mail address: {kassing,giesl}@cs.rwth-aachen.de

ABSTRACT. There are many evaluation strategies for term rewrite systems, but automatically proving termination or analyzing complexity is usually easiest for innermost rewriting. Several syntactic criteria exist when innermost termination implies (full) termination or when runtime complexity and innermost runtime complexity coincide. We adapt these criteria to the probabilistic setting, e.g., we show when it suffices to analyze almost-sure termination w.r.t. innermost rewriting in order to prove (full) almost-sure termination of probabilistic term rewrite systems. These criteria can be applied for both termination and complexity analysis in the probabilistic setting. We implemented and evaluated our new contributions in the tool AProVE. Moreover, we also use our new results to investigate the modularity of probabilistic termination properties.

1. INTRODUCTION

Termination and complexity analysis are among the main tasks in program verification, and techniques and tools to analyze termination or complexity of term rewrite systems (TRSs) automatically have been studied for decades. While a direct application of classical reduction orderings is often too weak, these orderings can be used successfully within the *dependency pair* (DP) framework for termination [AG00, GTSKF06] and for innermost runtime complexity [NEG13]. Moreover, the framework of [AM16] uses *weak* dependency pairs in order to analyze derivational and runtime complexity. These frameworks allow for modular termination and complexity proofs by decomposing the original problem into sub-problems which can then be analyzed independently using different techniques. Thus, DPs are used in essentially all current termination and complexity tools for TRSs (e.g., AProVE [GAB⁺17], MuTerm [GL20], NaTT [YKS14], TcT [AMS16], T_TT₂ [KSZM09]). To allow certification of proofs with DPs, they have been formalized in several proof assistants (e.g., in Rocq (formerly Coq) [CCF⁺07, BK11], Isabelle [TS09], and recently together with the size-change principle [LJB01, TG05, MV06] in PVS [AAR20, MAM⁺23]), and there exist several corresponding certification tools for termination and complexity proofs with DPs (e.g., CeTA [TS09]).

Key words and phrases: Probabilistic Programming, Term Rewriting, Evaluation Strategies, Modularity.

* This is a revised and extended journal version of our earlier conference paper [KFG24].

Funded by the DFG Research Training Group 2236 UnRAVeL.

Probabilistic programs are used to describe randomized algorithms and probability distributions, with applications in many areas, see, e.g., [GHNR14]. To use TRSs also for such programs, *probabilistic term rewrite systems* (PTRSs) were introduced in [BK02, BG05, ADLY20]. In the probabilistic setting, there are several notions of “termination”. For example, a program is *almost-surely terminating* (AST) if the probability for termination is 1. Another interesting property is *positive almost-sure termination* (PAST) [Sah78, BG05], which means that the expected length of every evaluation is finite. Finally, *strong* or *bounded almost-sure termination* (SAST) [FC19, ADLY20] requires that, for every configuration t (e.g., for every term), there is a finite bound on the expected lengths of all evaluations starting in t . Thus, if there is a start configuration t which non-deterministically leads to evaluations of arbitrary finite length, then the program can be PAST, but not SAST. In general, SAST implies PAST, and PAST implies AST, but the converse directions do not hold.

We recently developed an adaption of the DP framework for AST [KG24] and an adaption for innermost AST [KG23a, KDG24] (i.e., AST restricted to rewrite sequences where one only evaluates at innermost positions), which allows us to benefit from a similar modularity as in the non-probabilistic setting. However, the DP framework for innermost AST is substantially more powerful than the one for AST. Indeed, also in the non-probabilistic setting, innermost termination is usually substantially easier to prove than (full) termination, i.e., termination w.r.t. any possible evaluation strategy, see, e.g., [AG00, GTSKF06]. The same holds for non-probabilistic complexity analysis, where the DP framework of [NEG13] is restricted to innermost rewriting and the framework of [AM16] is considerably more powerful for innermost than for full rewriting. To lift innermost termination and complexity proofs to full rewriting, in the non-probabilistic setting there exist several sufficient criteria which ensure that innermost termination implies full termination [Gra95] and that innermost runtime complexity coincides with full runtime complexity [FG17].

Up to now no such results were known in the probabilistic setting. Our paper presents the first sufficient criteria for PTRSs which ensure that, e.g., AST coincides for full and innermost rewriting, and we also show similar results for other rewrite strategies like *leftmost-innermost* rewriting. We focus on criteria that can be checked automatically, so we can combine our results with the DP framework for proving innermost AST of PTRSs [KG23a, KDG24]. In this way, we obtain a technique that (if applicable) can prove AST for *full* rewriting automatically and is significantly more powerful than the corresponding DP framework for full AST [KG24].

Our criteria also hold for PAST, SAST, and expected complexity. Similar to AST, very recently we developed an adaption of the DP framework to prove innermost SAST and upper bounds on the innermost expected runtime complexity [KSG25]. Again, we can use our results on the relation between innermost and full rewriting to apply this adaption of the DP framework for arbitrary evaluation strategies. A corresponding evaluation can be found in [KSG25]. Except for applying polynomial or matrix orderings directly on the whole PTRS [ADLY20], currently there do not exist any other automatic techniques for PAST, SAST, or expected complexity of PTRSs.

As a corollary of our results in the probabilistic setting, we also develop the first results relating derivational and innermost derivational complexity for ordinary non-probabilistic TRSs. The difference between derivational and runtime complexity is that runtime complexity only considers start terms where a defined function symbol (i.e., an “algorithm”) is applied to arguments built with constructor symbols (i.e., to “data”), while derivational complexity allows arbitrary start terms.

There exist numerous results on the modularity of termination, confluence, and completeness of TRSs in the non-probabilistic setting, see, e.g., [Gra95, Gra12, Toy87a, Toy87b]. Based on our novel criteria, we develop the first modularity results for probabilistic termination w.r.t. different evaluation strategies, i.e., we investigate whether **AST**, **PAST**, or **SAST** are preserved for unions of PTRSs. Additionally, we also study preservation of **AST**, **PAST**, or **SAST** under signature extensions, which can be seen as a special case of modularity, i.e., a specific union of PTRSs, where the second PTRS only contains trivially terminating rewrite rules over the new signature. We show that while **AST** and **SAST** are preserved under signature extensions, this does not hold for **PAST**, implying that any sound and complete proof technique for **PAST** of PTRSs has to take the specific signature into account. Related to these results, we show that **PAST** and **SAST** coincide for a broad class of PTRSs. For example, if the signature contains at least one function symbol of arity greater than 1, then there is no difference between **PAST** and **SAST** for finite PTRSs.

Structure. We start with preliminaries on term rewriting in Section 2. Then we recapitulate PTRSs based on [BG05, DCM18, ADLY20, Fag22, KG23a] in Section 3. In Section 4 we show that the properties of [Gra95] that ensure equivalence of innermost and full termination do not suffice in the probabilistic setting and extend them accordingly. In particular, we show that innermost and full **AST** coincide for PTRSs that are non-overlapping and linear. This result also holds for **PAST**, **SAST**, and the expected runtime complexity, as well as for strategies like leftmost-innermost evaluation. In Section 5 we try to weaken the linearity requirement in order to prove full **AST** for larger classes of PTRSs. The implementation of our criteria in the tool **AProVE** is evaluated in Section 6. Afterwards, in Section 7 we analyze the modularity of all these (full and innermost) termination properties for PTRSs. We discuss related work on the verification of probabilistic programs in Section 8. Finally, we conclude in Section 9 and refer to App. A for all missing proofs.¹

Novel Contributions of the Paper. The current paper extends our earlier conference paper [KFG24] by:

- All results concerning **SAST** and the novel relations between **SAST** and **PAST** (the whole Section 3.3 as well as all results on **SAST** from Sections 4 and 5).
- The new result that **PAST** is not closed under signature extensions (Thm. 3.16), while **AST** and **SAST** are (Thm. 7.15).
- The novel definitions of expected derivational and runtime complexity (Def. 3.11 and 3.13) and all corresponding results (Thm. 4.6, 4.13, and 5.12).
- The corollaries for non-probabilistic derivational complexity (Cor. 4.8 and 4.14).
- The whole Section 7 concerning the analysis of modularity.
- Numerous additional explanations, examples, and remarks.
- More details on the proofs of the main theorems (including central lemmas like Lemma 4.3, 4.12, and 5.10), in addition to the full proofs in App. A.
- A new counterexample (Counterex. 5.5) showing a mistake in our earlier conference paper [KFG24].
- An improved implementation and evaluation which combines the contributions of the current paper with the DP framework for full **AST** from [KG24] (which had not yet been developed at the time of our conference paper [KFG24]), see Section 6.

¹To ease readability, for those proofs which require larger technical constructions, we only give proof sketches in the main part of the paper and present the corresponding full technical proofs in App. A.

2. PRELIMINARIES

For any relation $\rightarrow \subseteq A \times A$ on some set A and $n \in \mathbb{N}$, we define \rightarrow^n as $\rightarrow^0 = \{(a, a) \mid a \in A\}$ and $\rightarrow^{n+1} = \rightarrow^n \circ \rightarrow$, where “ \circ ” denotes composition of relations, and define $\rightarrow^* = \bigcup_{n \in \mathbb{N}} \rightarrow^n$, i.e., \rightarrow^* is the *reflexive and transitive closure* of \rightarrow . Let $\mathbf{NF}_{\rightarrow}$ denote the set of all elements that are in *normal form* w.r.t. \rightarrow , i.e., for all $a \in \mathbf{NF}_{\rightarrow}$ there is no $b \in A$ with $a \rightarrow b$.

We assume familiarity with term rewriting [BN98], but recapitulate the notions that are needed for this work. We write $\mathcal{T}(\Sigma, \mathcal{V})$ for the set of all *terms* over a (possibly infinite) countable set of *function symbols* $\Sigma = \biguplus_{k \in \mathbb{N}} \Sigma_k$ and a (possibly infinite) countable set of *variables* \mathcal{V} , and \mathcal{T} if the specific sets Σ and \mathcal{V} are irrelevant or clear from the context. To be precise, $\mathcal{T}(\Sigma, \mathcal{V})$ is the smallest set with $\mathcal{V} \subseteq \mathcal{T}(\Sigma, \mathcal{V})$, and if $f \in \Sigma_k$ and $t_1, \dots, t_k \in \mathcal{T}(\Sigma, \mathcal{V})$ then $f(t_1, \dots, t_k) \in \mathcal{T}(\Sigma, \mathcal{V})$. We say that a function symbol has *arity* $k \in \mathbb{N}$ if $f \in \Sigma_k$, i.e., the arity denotes how many arguments a function symbol takes. A *substitution* is a function $\sigma : \mathcal{V} \rightarrow \mathcal{T}$ with $\sigma(x) = x$ for all but finitely many $x \in \mathcal{V}$, and we often write $x\sigma$ instead of $\sigma(x)$. Substitutions homomorphically extend to terms: If $t = f(t_1, \dots, t_k) \in \mathcal{T}$ then $t\sigma = f(t_1\sigma, \dots, t_k\sigma)$. For a term $t \in \mathcal{T}$, the set of *positions* $\text{Pos}(t)$ is the smallest subset of \mathbb{N}^* satisfying $\varepsilon \in \text{Pos}(t)$, and if $t = f(t_1, \dots, t_k)$ then for all $1 \leq i \leq k$ and all $\pi \in \text{Pos}(t_i)$ we have $i.\pi \in \text{Pos}(t)$. If $\pi \in \text{Pos}(t)$ then $t|_{\pi}$ denotes the subterm starting at position π and $t[r]_{\pi}$ denotes the term that results from replacing the subterm $t|_{\pi}$ at position π with the term $r \in \mathcal{T}$. A *context* C is a term containing exactly one occurrence of a special constant symbol \square , denoting a hole. The constant \square is used to indicate where one can insert a term, i.e., if \square is at position π of C , then $C[t]_{\pi}$ is a shorthand notation for $C[t]_{\pi}$.

A *rewrite rule* $\ell \rightarrow r$ is a pair of terms $\ell, r \in \mathcal{T}$ such that $\mathcal{V}(r) \subseteq \mathcal{V}(\ell)$ and $\ell \notin \mathcal{V}$, where $\mathcal{V}(t)$ denotes the set of all variables occurring in $t \in \mathcal{T}$. A *term rewrite system* (TRS) is a (possibly infinite) countable set of rewrite rules. As an example, consider the TRS \mathcal{R}_d that doubles a natural number (represented by the terms s and 0) with the rewrite rules $d(s(x)) \rightarrow s(d(x))$ and $d(0) \rightarrow 0$. A TRS \mathcal{R} induces a *rewrite relation* $\xrightarrow{\mathcal{R}} \subseteq \mathcal{T} \times \mathcal{T}$ on terms where $s \xrightarrow{\mathcal{R}} t$ holds if there is a position $\pi \in \text{Pos}(s)$, a rule $\ell \rightarrow r \in \mathcal{R}$, and a substitution σ such that $s|_{\pi} = \ell\sigma$ and $t = s[r\sigma]_{\pi}$. Here, \mathbf{f} stands for “full rewriting”² as we did not fix any specific strategy yet. A rewrite step $s \xrightarrow{\mathcal{R}} t$ is an *innermost* rewrite step (denoted $s \xrightarrow{i} t$) if all proper subterms of the used *redex* $\ell\sigma$ are in normal form w.r.t. \mathcal{R} , i.e., the proper subterms of $\ell\sigma$ do not contain redexes themselves and thus, they cannot be reduced with $\xrightarrow{\mathcal{R}}$. For example, we have $d(s(d(s(0)))) \xrightarrow{i} d(s(s(d(0))))$. Let $\mathbf{NF}_{\mathcal{R}}$ denote the set of all terms that are in normal form w.r.t. $\xrightarrow{\mathcal{R}}$.

Let $<$ be the *prefix ordering* on positions and let \leq be its reflexive closure. Two positions τ and π are *parallel* if both $\tau \not\leq \pi$ and $\pi \not\leq \tau$ hold. For two parallel positions τ and π we define $\tau < \pi$ if we have $i < j$ for the unique i, j such that $\chi.i \leq \tau$ and $\chi.j \leq \pi$, where χ is the longest common prefix of τ and π . An innermost rewrite step $s \xrightarrow{i} t$ at position π is *leftmost* (denoted $s \xrightarrow{li} t$) if s does not contain any redex at a position τ with $\tau < \pi$.

In this paper, we will consider innermost rewriting (**i**), leftmost-innermost rewriting (**li**), and full rewriting (**f**). Since every leftmost-innermost rewrite step is an innermost rewrite step, and every innermost rewrite step is a step w.r.t. full rewriting, one directly obtains $\xrightarrow{li} \subseteq \xrightarrow{i} \subseteq \xrightarrow{f}$ for every TRS \mathcal{R} . Let $\mathbb{S} = \{\mathbf{f}, \mathbf{i}, \mathbf{li}\}$ be the set of these three rewrite strategies.

²In the literature, one usually simply writes $\rightarrow_{\mathcal{R}}$ instead. We use $\xrightarrow{\mathcal{R}}$ here to clearly indicate the corresponding rewrite strategy.

2.1. Termination. Let $\rightarrow \subseteq \mathcal{T} \times \mathcal{T}$ be a relation on terms. We call \rightarrow *strongly normalizing* or SN_{\rightarrow} for short if \rightarrow is well founded. A TRS \mathcal{R} is *terminating* if we have $\text{SN}_{\rightarrow_{\mathcal{R}}}$, and \mathcal{R} is *innermost terminating* or *leftmost-innermost terminating* if we have $\text{SN}_{\rightarrow_{\mathcal{R}}}^i$ or $\text{SN}_{\rightarrow_{\mathcal{R}}}^l$, respectively. If every term $t \in \mathcal{T}$ has a normal form w.r.t. \rightarrow (i.e., we have $t \rightarrow^* t'$ where $t' \in \text{NF}_{\rightarrow}$) then we call \rightarrow *weakly normalizing* (denoted by WN_{\rightarrow}).

Two terms $s, t \in \mathcal{T}$ are *joinable* via \mathcal{R} if there exists a $w \in \mathcal{T}$ such that $s \xrightarrow{\mathcal{R}}^* w \xleftarrow{\mathcal{R}}^* t$. Two rules $\ell_1 \rightarrow r_1, \ell_2 \rightarrow r_2 \in \mathcal{R}$ with renamed variables such that $\mathcal{V}(\ell_1) \cap \mathcal{V}(\ell_2) = \emptyset$ are *overlapping* if there exists a non-variable position π of ℓ_1 such that $\ell_1|_{\pi}$ and ℓ_2 are unifiable, i.e., there exists a substitution σ such that $\ell_1|_{\pi}\sigma = \ell_2\sigma$. If $(\ell_1 \rightarrow r_1) = (\ell_2 \rightarrow r_2)$, then we require that $\pi \neq \varepsilon$. \mathcal{R} is *non-overlapping* (NO) if it has no overlapping rules. As an example, the TRS \mathcal{R}_d is non-overlapping. A TRS is *left-linear* (LL) (*right-linear*, RL) if every variable occurs at most once in the left-hand side (right-hand side) of a rule. A TRS is *linear* if it is both left- and right-linear. A TRS is *non-erasing* (NE) if in every rule, all variables of the left-hand side also occur in the right-hand side.

Next, we recapitulate the relations between $\text{SN}_{\rightarrow_{\mathcal{R}}}$, $\text{SN}_{\rightarrow_{\mathcal{R}}}^i$, $\text{SN}_{\rightarrow_{\mathcal{R}}}^l$, and $\text{WN}_{\rightarrow_{\mathcal{R}}}$ in the non-probabilistic setting. Obviously, the stronger notion always implies the weaker one, e.g., $\text{SN}_{\rightarrow_{\mathcal{R}}}$ implies $\text{SN}_{\rightarrow_{\mathcal{R}}}^i$ and $\text{WN}_{\rightarrow_{\mathcal{R}}}^i$ implies $\text{WN}_{\rightarrow_{\mathcal{R}}}$ since $\rightarrow_{\mathcal{R}}^i \subseteq \rightarrow_{\mathcal{R}}$. The interesting question is for which classes of TRSs the weaker and the stronger notion are equivalent.³ We start with the relation between $\text{SN}_{\rightarrow_{\mathcal{R}}}$ and $\text{SN}_{\rightarrow_{\mathcal{R}}}^i$.

Counterexample 2.1 (Toyama's Counterexample [Toy87a]). For the TRS \mathcal{R}_1 with the rules $f(a, b, x) \rightarrow f(x, x, x)$, $g \rightarrow a$, and $g \rightarrow b$, we do not have $\text{SN}_{\rightarrow_{\mathcal{R}_1}}$ due to the infinite rewrite sequence $f(a, b, g) \xrightarrow{\mathcal{R}_1} f(g, g, g) \xrightarrow{\mathcal{R}_1} f(a, g, g) \xrightarrow{\mathcal{R}_1} f(a, b, g) \xrightarrow{\mathcal{R}_1} \dots$. But the only innermost rewrite sequences starting with $f(a, b, g)$ are $f(a, b, g) \xrightarrow{\mathcal{R}_1} f(a, b, a) \xrightarrow{\mathcal{R}_1} f(a, a, a)$ and $f(a, b, g) \xrightarrow{\mathcal{R}_1} f(a, b, b) \xrightarrow{\mathcal{R}_1} f(b, b, b)$, i.e., both of them reach normal forms in the end. Thus, $\text{SN}_{\rightarrow_{\mathcal{R}_1}}^i$ holds as we have to rewrite the inner g before we can use the f -rule.

The first property known to ensure equivalence of $\text{SN}_{\rightarrow_{\mathcal{R}}}$ and $\text{SN}_{\rightarrow_{\mathcal{R}}}^i$ is orthogonality, which was already shown in [O'D77]. A TRS is *orthogonal* (OR) if it is non-overlapping and left-linear.

Theorem 2.2 (From $\text{SN}_{\rightarrow_{\mathcal{R}}}^i$ to $\text{SN}_{\rightarrow_{\mathcal{R}}}$ (1), [O'D77]). *If a TRS \mathcal{R} is OR, then:*

$$\text{SN}_{\rightarrow_{\mathcal{R}}} \iff \text{SN}_{\rightarrow_{\mathcal{R}}}^i$$

Then, in [Gra95], it was shown that one can remove the requirement of left-linearity.

Theorem 2.3 (From $\text{SN}_{\rightarrow_{\mathcal{R}}}^i$ to $\text{SN}_{\rightarrow_{\mathcal{R}}}$ (2), [Gra95]). *If a TRS \mathcal{R} is NO, then:*

$$\text{SN}_{\rightarrow_{\mathcal{R}}} \iff \text{SN}_{\rightarrow_{\mathcal{R}}}^i$$

Moreover, [Gra95] refined Thm. 2.3 further. A TRS \mathcal{R} is an *overlay system* (OS) if its rules may only overlap at the root position, i.e., $\pi = \varepsilon$. For instance, \mathcal{R}_1 from Counterex. 2.1 is an overlay system. Furthermore, a TRS is *locally confluent* (or *weakly Church-Rosser*, abbreviated WCR) if for all terms $s, t_1, t_2 \in \mathcal{T}$ such that $t_1 \xleftarrow{\mathcal{R}} s \xrightarrow{\mathcal{R}} t_2$ the terms t_1 and t_2 are joinable. \mathcal{R}_1 is *not* WCR, as we have $a \xleftarrow{\mathcal{R}_1} g \xrightarrow{\mathcal{R}_1} b$, but a and b are not joinable. If a TRS has both of these properties, then $\text{SN}_{\rightarrow_{\mathcal{R}}}^i$ and $\text{SN}_{\rightarrow_{\mathcal{R}}}$ are again equivalent.

³Note that to this end, we do not have to consider $\text{WN}_{\rightarrow_{\mathcal{R}}}^l$ and $\text{WN}_{\rightarrow_{\mathcal{R}}}^i$. The reason is that when analyzing under which conditions $\text{WN}_{\rightarrow_{\mathcal{R}}}$ implies $\text{SN}_{\rightarrow_{\mathcal{R}}}$, we also know that under these conditions we have $\text{WN}_{\rightarrow_{\mathcal{R}}}^s \implies \text{SN}_{\rightarrow_{\mathcal{R}}}^s$ for all $s \in \mathbb{S}$, since $\text{WN}_{\rightarrow_{\mathcal{R}}}^s \implies \text{WN}_{\rightarrow_{\mathcal{R}}}$ and $\text{SN}_{\rightarrow_{\mathcal{R}}} \implies \text{SN}_{\rightarrow_{\mathcal{R}}}^s$ hold. Moreover, we have $\text{SN}_{\rightarrow_{\mathcal{R}}}^s \implies \text{WN}_{\rightarrow_{\mathcal{R}}}^s$ for all $s \in \mathbb{S}$.

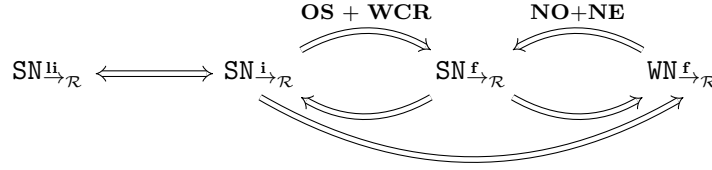


Figure 1: Relations between the different termination properties for TRSs

Theorem 2.4 (From $\text{SN}^i_{\to \mathcal{R}}$ to $\text{SN}^f_{\to \mathcal{R}}$ (3), [Gra95]). *If a TRS \mathcal{R} is OS and WCR, then:*

$$\text{SN}^f_{\to \mathcal{R}} \iff \text{SN}^i_{\to \mathcal{R}}$$

Thm. 2.4 is stronger than Thm. 2.3 as every non-overlapping TRS is a locally confluent overlay system [KB70].

Next, we recapitulate the results on the relation between $\text{WN}^f_{\to \mathcal{R}}$ and $\text{SN}^f_{\to \mathcal{R}}$.

Counterexample 2.5. Consider the TRS \mathcal{R}_2 with the rules $f(x) \rightarrow b$ and $a \rightarrow f(a)$. We do not have $\text{SN}^f_{\to \mathcal{R}_2}$ since we can always rewrite the inner a to get $a \xrightarrow{f}_{\mathcal{R}_2} f(a) \xrightarrow{f}_{\mathcal{R}_2} f(f(a)) \xrightarrow{f}_{\mathcal{R}_2} \dots$. On the other hand, $\text{WN}^f_{\to \mathcal{R}_2}$ holds since we can also rewrite the outer $f(\dots)$ before we use the a -rule twice, resulting in the term b , which is a normal form. For the TRS \mathcal{R}_3 with the rules $f(a) \rightarrow b$ and $a \rightarrow f(a)$, the situation is similar.

The TRS \mathcal{R}_2 from Counterex. 2.5 is erasing and \mathcal{R}_3 is overlapping. For TRSs with neither of those two properties, $\text{SN}^f_{\to \mathcal{R}}$ and $\text{WN}^f_{\to \mathcal{R}}$ are equivalent.

Theorem 2.6 (From $\text{WN}^f_{\to \mathcal{R}}$ to $\text{SN}^f_{\to \mathcal{R}}$, [Gra95]). *If a TRS \mathcal{R} is NO and NE, then:*

$$\text{SN}^f_{\to \mathcal{R}} \iff \text{WN}^f_{\to \mathcal{R}}$$

Finally, we look at the relation between rewrite strategies that use an ordering for parallel redexes like leftmost-innermost rewriting compared to innermost rewriting. It turns out that such an ordering does not interfere with termination at all.

Theorem 2.7 (From $\text{SN}^u_{\to \mathcal{R}}$ to $\text{SN}^i_{\to \mathcal{R}}$, [Kri00]). *For all TRSs \mathcal{R} we have:*

$$\text{SN}^i_{\to \mathcal{R}} \iff \text{SN}^u_{\to \mathcal{R}}$$

The relations between the different termination properties for non-probabilistic TRSs (given in Thm. 2.4, 2.6, and 2.7) are summarized in Fig. 1.

2.2. Complexity. Next, we recapitulate known results regarding the complexity of TRSs under different rewrite strategies. There are two standard notions of complexity used in term rewriting: *derivational* and *runtime complexity* [HL89, HM08]. For any $M \subseteq \mathbb{N} \cup \{\omega\}$, $\sup M$ denotes the least upper bound of M , where $\sup \emptyset = 0$. For a relation $\rightarrow \subseteq \mathcal{T} \times \mathcal{T}$, the *derivation height* $\text{dh}_{\rightarrow} : \mathcal{T} \rightarrow \mathbb{N} \cup \{\omega\}$ of a term $t \in \mathcal{T}$ is $\text{dh}_{\rightarrow}(t) = \sup\{m \mid \exists t' \in \mathcal{T} : t \rightarrow^m t'\}$, i.e., the length of the longest \rightarrow -rewrite sequence starting with t . Then, the *derivational complexity* $\text{dc}_{\rightarrow} : \mathbb{N} \rightarrow \mathbb{N} \cup \{\omega\}$ of \rightarrow is defined as $\text{dc}_{\rightarrow}(n) = \sup\{\text{dh}_{\rightarrow}(t) \mid t \in \mathcal{T}, |t| \leq n\}$. Here, the *size* $|t|$ of a term t is the number of occurrences of function symbols and variables in t , i.e., we have $|x| = 1$ for $x \in \mathcal{V}$ and $|f(t_1, \dots, t_k)| = 1 + \sum_{i=1}^k |t_i|$. So $\text{dc}_{\rightarrow}(n)$ denotes the length of the longest \rightarrow -rewrite sequence starting with an arbitrary term of at most size n . For a TRS \mathcal{R} and a strategy $s \in \mathbb{S}$, the *derivational complexity* of \mathcal{R} w.r.t. s is $\text{dc}_{s \rightarrow \mathcal{R}}$.

In contrast, for *runtime complexity* one restricts the start terms to be *basic*. For a TRS \mathcal{R} , we decompose its signature $\Sigma = \Sigma_C \uplus \Sigma_D$ such that $f \in \Sigma_D$ if $f = \text{root}(\ell)$ for some rule $\ell \rightarrow r \in \mathcal{R}$. The symbols in Σ_C and Σ_D are called *constructors* and *defined symbols*, respectively. A term $t \in \mathcal{T}$ is *basic* if $t = f(t_1, \dots, t_k)$ such that $f \in \Sigma_D$ and $t_i \in \mathcal{T}(\Sigma_C, \mathcal{V})$ for all $1 \leq i \leq k$, and the set of all basic terms is denoted by $\mathcal{T}_{\mathcal{B}}$. The *runtime complexity* of a relation $\rightarrow \subseteq \mathcal{T} \times \mathcal{T}$ w.r.t. a TRS \mathcal{R} maps any $n \in \mathbb{N}$ to the length of the longest \rightarrow -rewrite sequence starting with a basic term $t \in \mathcal{T}_{\mathcal{B}}$ with $|t| \leq n$. So $\text{rc}_{\rightarrow, \mathcal{R}} : \mathbb{N} \rightarrow \mathbb{N} \cup \{\omega\}$ is defined as $\text{rc}_{\rightarrow, \mathcal{R}}(n) = \sup\{\text{dh}_{\rightarrow}(t) \mid t \in \mathcal{T}_{\mathcal{B}}, |t| \leq n\}$, where $\mathcal{T}_{\mathcal{B}}$ are the basic terms w.r.t. \mathcal{R} . For a TRS \mathcal{R} and a strategy $s \in \mathbb{S}$, the *runtime complexity* of \mathcal{R} w.r.t. s is $\text{rc}_{s \rightarrow \mathcal{R}, \mathcal{R}}$, where we often omit the additional index \mathcal{R} for readability, i.e., $\text{rc}_{s \rightarrow \mathcal{R}, \mathcal{R}} = \text{rc}_{s \rightarrow \mathcal{R}}$.

We are not aware of any non-trivial classes of TRSs where innermost and full derivational complexity coincide, i.e., where we have $\text{dc}_{f \rightarrow \mathcal{R}}(n) = \text{dc}_{i \rightarrow \mathcal{R}}(n)$ for all $n \in \mathbb{N}$ and all TRSs \mathcal{R} from that class. However, for runtime complexity, there exist sufficient criteria for TRSs \mathcal{R} which imply $\text{rc}_{f \rightarrow \mathcal{R}} = \text{rc}_{i \rightarrow \mathcal{R}}$ [FG17]. To relate innermost and full runtime complexity, it does not suffice to require that the rules are non-overlapping, but we also have to make sure that we cannot duplicate redexes during an evaluation, as shown by the following counterexample from [FG17].

Counterexample 2.8. Let $s^n(0)$ abbreviate $\underbrace{s(\dots s(0) \dots)}_{n \text{ times}}$ and let \mathcal{R}_4 consist of the rules:

$$\begin{array}{ll} f(0, y) \rightarrow y & g(x) \rightarrow f(x, a) \\ f(s(x), y) \rightarrow f(x, \text{node}(y, y)) & a \rightarrow b \end{array}$$

Note that \mathcal{R}_4 is non-overlapping. For the basic term $g(s^n(0))$ of size $n + 2$ we have $g(s^n(0)) \xrightarrow{f \rightarrow \mathcal{R}_4} f(s^n(0), a)$. Next, one could apply the second f -rule repeatedly and obtain a term corresponding to a full binary tree of height n whose exponentially many leaves all correspond to the symbol a . Finally, these leaves can all be reduced to b in 2^n steps, hence $\text{rc}_{f \rightarrow \mathcal{R}_4}^{\mathcal{R}_4}(n) \in \Theta(2^n)$. On the other hand, any basic term of size n only leads to innermost rewrite sequences of length $\mathcal{O}(n)$, as for example in $f(s^n(0), a)$ we have to rewrite the inner a before we are able to duplicate it. Thus, we obtain $\text{rc}_{i \rightarrow \mathcal{R}_4}^{\mathcal{R}_4}(n) \in \Theta(n)$.

The property that no redexes are duplicated during a rewrite sequence that starts with a basic term is called *spareness* [FG17]. For a TRS \mathcal{R} , a rewrite step using the rule $\ell \rightarrow r \in \mathcal{R}$ and the substitution σ is *sparse* if $\sigma(x) \in \text{NF}_{\mathcal{R}}$ for every $x \in \mathcal{V}$ that occurs more than once in r . A $\xrightarrow{f \rightarrow \mathcal{R}}$ -rewrite sequence is *sparse* if each of its $\xrightarrow{f \rightarrow \mathcal{R}}$ -steps is sparse. \mathcal{R} is *sparse* (SP) if each $\xrightarrow{f \rightarrow \mathcal{R}}$ -rewrite sequence that starts with a basic term $t \in \mathcal{T}_{\mathcal{B}}$ is sparse. In general, it is undecidable whether a TRS is sparse. However, there exist computable sufficient conditions for spareness, see [FG17].

Similar to the results regarding termination, for the equivalence of $\text{rc}_{f \rightarrow \mathcal{R}}$ and $\text{rc}_{i \rightarrow \mathcal{R}}$ it again suffices to require overlay systems instead of non-overlapping TRSs. However, while we have to require spareness, in contrast to Thm. 2.4 we do not need local confluence.

Theorem 2.9 (From $\text{rc}_{i \rightarrow \mathcal{R}}$ to $\text{rc}_{f \rightarrow \mathcal{R}}$, [FG17]). *If a TRS \mathcal{R} is OS and SP, then:*

$$\text{rc}_{f \rightarrow \mathcal{R}} = \text{rc}_{i \rightarrow \mathcal{R}}$$

Counterex. 2.8 also works for derivational complexity, showing that duplication of redexes needs to be prohibited for equivalence of $\text{dc}_{f \rightarrow \mathcal{R}}$ and $\text{dc}_{i \rightarrow \mathcal{R}}$ as well. However, note that *spareness* requires basic start terms, so an analogous theorem for derivational complexity would have to require a stronger property, e.g., right-linearity of \mathcal{R} . We will present such a

result as a corollary of our analysis of probabilistic rewriting in Section 4 (Cor. 4.8). Finally, the relation between the runtime complexities for rewrite strategies that use an ordering for parallel redexes has not been studied so far. Here, our probabilistic analysis will also yield a corresponding corollary (Cor. 4.14).

3. PROBABILISTIC TERM REWRITING

In Section 3.1, we define *probabilistic TRSs* [BG05, ADLY20, KG23a]. In Section 3.2, we recapitulate notions of termination in the probabilistic setting, like almost-sure termination (AST), positive almost-sure termination (PAST), strong almost-sure termination (SAST), and we introduce a novel definition of *expected derivational/runtime complexity*. Then in Section 3.3, we present new results on the relation between PAST and SAST.

3.1. Probabilistic Term Rewriting. In contrast to TRSs, a PTRS has finite⁴ multi-distributions on the right-hand sides of its rewrite rules.⁵ A finite *multi-distribution* μ on a set $A \neq \emptyset$ is a finite multiset of pairs $(p : a)$, where $0 < p \leq 1$ is a probability and $a \in A$, such that $\sum_{(p:a) \in \mu} p = 1$. $\text{FDist}(A)$ is the set of all finite multi-distributions on A . For $\mu \in \text{FDist}(A)$, its *support* is the multiset $\text{Supp}(\mu) = \{a \mid (p : a) \in \mu \text{ for some } p\}$. A *probabilistic rewrite rule* is a pair $(\ell \rightarrow \mu) \in \mathcal{T} \times \text{FDist}(\mathcal{T})$ such that $\ell \notin \mathcal{V}$ and $\mathcal{V}(r) \subseteq \mathcal{V}(\ell)$ for every $r \in \text{Supp}(\mu)$. A *probabilistic TRS* (PTRS) is a (possibly infinite) countable set \mathcal{P} of probabilistic rewrite rules. Similar to TRSs, the PTRS \mathcal{P} induces a *rewrite relation* $\xrightarrow{\mathcal{P}} \subseteq \mathcal{T} \times \text{FDist}(\mathcal{T})$ where $s \xrightarrow{\mathcal{P}} \{p_1 : t_1, \dots, p_k : t_k\}$ if there is a position π , a rule $\ell \rightarrow \{p_1 : r_1, \dots, p_k : r_k\} \in \mathcal{P}$, and a substitution σ such that $s|_{\pi} = \ell\sigma$ and $t_j = s[r_j\sigma]_{\pi}$ for all $1 \leq j \leq k$. We call $s \xrightarrow{\mathcal{P}} \mu$ an *innermost* rewrite step (denoted $s \xrightarrow{i\mathcal{P}} \mu$) if all proper subterms of the used redex $\ell\sigma$ are in normal form w.r.t. \mathcal{P} . We have $s \xrightarrow{l\mathcal{P}} \mu$ if the rewrite step $s \xrightarrow{i\mathcal{P}} \mu$ at position π is leftmost (i.e., there is no redex at a position τ with $\tau \prec \pi$). For example, the PTRS \mathcal{P}_{rw} with the only rule $g \rightarrow \{1/2 : c(g, g), 1/2 : 0\}$ corresponds to a symmetric random walk on the number of g -symbols in a term.

Many properties of TRSs from Section 2 can be lifted to PTRSs in a straightforward way: A PTRS \mathcal{P} is right-linear (non-erasing) iff the TRS $\{\ell \rightarrow r \mid \ell \rightarrow \mu \in \mathcal{P}, r \in \text{Supp}(\mu)\}$ has the respective property. Moreover, all properties that just consider the left-hand sides, e.g., normal forms, left-linearity, being non-overlapping, orthogonality, and being an overlay system, can be lifted to PTRSs directly as well, since their rules again only have a single left-hand side.

3.2. Probabilistic Notions of Termination. Next, we introduce the different notions of probabilistic termination. In this section, we regard an arbitrary *probabilistic (term) relation* $\rightarrow \subseteq \mathcal{T} \times \text{FDist}(\mathcal{T})$ and let NF_{\rightarrow} again be the set of all normal forms for \rightarrow . As in [DCM18, ADLY20, Fag22, KG23a], we *lift* $\rightarrow \subseteq \mathcal{T} \times \text{FDist}(\mathcal{T})$ to a rewrite relation $\Rightarrow \subseteq \text{FDist}(\mathcal{T}) \times \text{FDist}(\mathcal{T})$ between multi-distributions in order to track all probabilistic rewrite sequences (up to non-determinism) at once. For any $0 < p \leq 1$ and any $\mu \in \text{FDist}(\mathcal{T})$, let $p \cdot \mu = \{(p \cdot q : a) \mid (q : a) \in \mu\}$.

⁴The restriction to *finite* multi-distributions allows us to simplify the handling of PTRSs in the proofs. However, we conjecture that most of our results also hold for PTRSs with infinite countable multi-distributions.

⁵A different form of probabilistic rewrite rules was proposed in PMAude [AMS06], where numerical extra variables in right-hand sides of rules are instantiated according to a probability distribution.

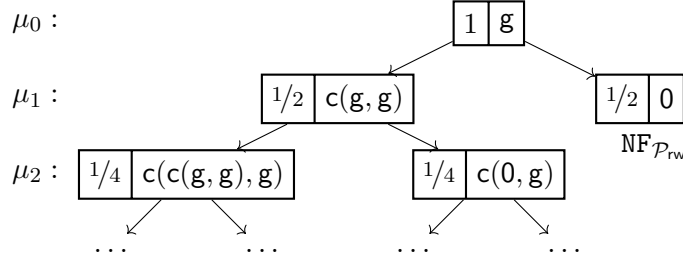


Figure 2: Corresponding RST for the $\xrightarrow{f}_{\mathcal{P}_{rw}}$ -rewrite sequence in Ex. 3.2.

Definition 3.1 (Lifting). The *lifting* $\Rightarrow \subseteq \text{FDist}(\mathcal{T}) \times \text{FDist}(\mathcal{T})$ of a relation $\rightarrow \subseteq \mathcal{T} \times \text{FDist}(\mathcal{T})$ is the smallest relation with:

- If $t \in \mathcal{T}$ is in normal form w.r.t. \rightarrow , then $\{1 : t\} \Rightarrow \{1 : t\}$.
- If $t \rightarrow \mu$, then $\{1 : t\} \Rightarrow \mu$.
- If for all $1 \leq j \leq k$ there are $\mu_j, \nu_j \in \text{FDist}(\mathcal{T})$ with $\mu_j \Rightarrow \nu_j$ and $0 < p_j \leq 1$ with $\sum_{1 \leq j \leq k} p_j = 1$, then $\bigcup_{1 \leq j \leq k} p_j \cdot \mu_j \Rightarrow \bigcup_{1 \leq j \leq k} p_j \cdot \nu_j$.

For a PTRS \mathcal{P} , we write $\xrightarrow{f}_{\mathcal{P}}$, $\xrightarrow{i}_{\mathcal{P}}$, and $\xrightarrow{li}_{\mathcal{P}}$ for the liftings of $\xrightarrow{f}_{\mathcal{P}}$, $\xrightarrow{i}_{\mathcal{P}}$, and $\xrightarrow{li}_{\mathcal{P}}$, respectively.

Example 3.2. We obtain the following $\xrightarrow{f}_{\mathcal{P}_{rw}}$ -rewrite sequence (which is also a $\xrightarrow{i}_{\mathcal{P}_{rw}}$ -rewrite sequence, but not a $\xrightarrow{li}_{\mathcal{P}_{rw}}$ -rewrite sequence).

$$\begin{aligned}
 & \{1 : g\} \\
 \xrightarrow{f}_{\mathcal{P}_{rw}} & \{1/2 : c(g, g), 1/2 : 0\} \\
 \xrightarrow{f}_{\mathcal{P}_{rw}} & \{1/4 : c(c(g, g), g), 1/4 : c(0, g), 1/2 : 0\} \\
 \xrightarrow{f}_{\mathcal{P}_{rw}} & \{1/8 : c(c(g, g), c(g, g)), 1/8 : c(c(g, g), 0), 1/8 : c(0, c(g, g)), 1/8 : c(0, 0), 1/2 : 0\}
 \end{aligned}$$

Another way to track all possible rewrite sequences with their corresponding probabilities is to lift \rightarrow to *rewrite sequence trees* (RSTs) [KDG24]. The nodes v of an \rightarrow -RST are labeled by pairs $(p_v : t_v)$ of a probability p_v and a term t_v , where the root is always labeled with the probability 1. For each node v with the successors w_1, \dots, w_k , the edge relation represents a step with the relation \rightarrow , i.e., $t_v \rightarrow \{ \frac{p_{w_1}}{p_v} : t_{w_1}, \dots, \frac{p_{w_k}}{p_v} : t_{w_k} \}$. For an \rightarrow -RST \mathfrak{T} , let $N^{\mathfrak{T}}$ denote the set of nodes and $\text{Leaf}^{\mathfrak{T}}$ denote the set of leaves. We say that \mathfrak{T} is *fully evaluated* if for every $v \in \text{Leaf}^{\mathfrak{T}}$ the corresponding term t_v is a normal form w.r.t. \rightarrow , i.e., $t_v \in \text{NF}_{\rightarrow}$. In Fig. 2 one can see the $\xrightarrow{f}_{\mathcal{P}_{rw}}$ -RST for the $\xrightarrow{f}_{\mathcal{P}_{rw}}$ -rewrite sequence from Ex. 3.2. Note that the normal forms remain in each multi-distribution of a $\xrightarrow{f}_{\mathcal{P}_{rw}}$ -rewrite sequence, but they are leaves of the corresponding $\xrightarrow{f}_{\mathcal{P}_{rw}}$ -RST.

To express the concept of almost-sure termination, one has to determine the probability for normal forms in a multi-distribution.

Definition 3.3 ($|\mu|_{\rightarrow}$, $|\mu|_{\mathcal{P}}$). Let $\mu \in \text{FDist}(\mathcal{T})$. For a relation $\rightarrow \subseteq \mathcal{T} \times \text{FDist}(\mathcal{T})$, let $|\mu|_{\rightarrow} = \sum_{(p:t) \in \mu, t \in \text{NF}_{\rightarrow}} p$, and for a PTRS \mathcal{P} , let $|\mu|_{\mathcal{P}} = \sum_{(p:t) \in \mu, t \in \text{NF}_{\mathcal{P}}} p$.

Example 3.4. Consider $\{1/8 : c(c(g, g), c(g, g)), 1/8 : c(c(g, g), 0), 1/8 : c(0, c(g, g)), 1/8 : c(0, 0), 1/2 : 0\} = \mu$ from Ex. 3.2. Then $|\mu|_{\mathcal{P}_{rw}} = 1/8 + 1/2 = 5/8$, since $c(0, 0)$ and 0 are both normal forms w.r.t. \mathcal{P}_{rw} .

Definition 3.5 (Almost-Sure Termination [ADLY20]). Let $\rightarrow \subseteq \mathcal{T} \times \text{FDist}(\mathcal{T})$ and $\vec{\mu} = (\mu_n)_{n \in \mathbb{N}}$ be an infinite \Rightarrow -rewrite sequence, i.e., $\mu_n \Rightarrow \mu_{n+1}$ for all $n \in \mathbb{N}$. We say that $\vec{\mu}$

converges with probability $\lim_{n \rightarrow \infty} |\mu_n|_{\rightarrow}$. The relation \rightarrow is *almost-surely terminating* (denoted AST_{\rightarrow}) if $\lim_{n \rightarrow \infty} |\mu_n|_{\rightarrow} = 1$ holds for every infinite \Rightarrow -rewrite sequence $(\mu_n)_{n \in \mathbb{N}}$. We say that \rightarrow is *weakly AST* (denoted $\text{wAST}_{\rightarrow}$) if for every term t there exists an infinite \Rightarrow -rewrite sequence $(\mu_n)_{n \in \mathbb{N}}$ with $\lim_{n \rightarrow \infty} |\mu_n|_{\rightarrow} = 1$ and $\mu_0 = \{1 : t\}$.

For the definition of wAST recall that by Def. 3.1 every term (even normal forms) can start infinite \Rightarrow -rewrite sequences, as we keep normal forms in \Rightarrow -steps.

Equivalently, one can also define AST_{\rightarrow} (and $\text{wAST}_{\rightarrow}$) via \rightarrow -RSTs. For any \rightarrow -RST \mathfrak{T} we define its *convergence probability* $|\mathfrak{T}| = \sum_{v \in \text{Leaf}^{\mathfrak{T}}} p_v$. Then AST_{\rightarrow} holds iff for all \rightarrow -RSTs \mathfrak{T} we have $|\mathfrak{T}| = 1$. Moreover, $\text{wAST}_{\rightarrow}$ holds iff for every term t there exists a fully evaluated \rightarrow -RST \mathfrak{T} whose root is labeled with $(1 : t)$ such that $|\mathfrak{T}| = 1$.

Example 3.6. For every infinite extension $(\mu_n)_{n \in \mathbb{N}}$ of the $\xrightarrow{f}_{\mathcal{P}_{\text{rw}}}$ -rewrite sequence in Ex. 3.2, we have $\lim_{n \rightarrow \infty} |\mu_n|_{\mathcal{P}_{\text{rw}}} = 1$. Indeed, we have $\text{AST}_{\xrightarrow{f}_{\mathcal{P}_{\text{rw}}}}$ and thus also $\text{AST}_{\xrightarrow{i}_{\mathcal{P}_{\text{rw}}}}$, $\text{AST}_{\xrightarrow{h}_{\mathcal{P}_{\text{rw}}}}$, and $\text{wAST}_{\xrightarrow{f}_{\mathcal{P}_{\text{rw}}}}$.

Next, we define *positive almost-sure termination*, which considers the *expected derivation length* $\text{edl}(\vec{\mu})$ of a rewrite sequence $\vec{\mu}$, i.e., the expected number of steps until one reaches a normal form. For positive almost-sure termination, we require that the expected derivation length of every possible rewrite sequence is finite. In the following definition, $(1 - |\mu_n|_{\rightarrow})$ is the probability of terms that are *not* in normal form w.r.t. \rightarrow after the n -th step.

Definition 3.7 (Positive Almost-Sure Termination, edl [ADLY20]). Let $\rightarrow \subseteq \mathcal{T} \times \text{FDist}(\mathcal{T})$ and $\vec{\mu} = (\mu_n)_{n \in \mathbb{N}}$ be an infinite \Rightarrow -rewrite sequence. By $\text{edl}(\vec{\mu}) = \sum_{n=0}^{\infty} (1 - |\mu_n|_{\rightarrow})$ we denote the *expected derivation length* of $\vec{\mu}$. The relation \rightarrow is *positively almost-surely terminating* (denoted $\text{PAST}_{\rightarrow}$) if $\text{edl}(\vec{\mu})$ is finite for every infinite \Rightarrow -rewrite sequence $\vec{\mu} = (\mu_n)_{n \in \mathbb{N}}$ starting with a single term, i.e., $\mu_0 = \{1 : t\}$ with $t \in \mathcal{T}$. We say that \rightarrow is *weakly PAST* (denoted $\text{wPAST}_{\rightarrow}$) if for every term t there exists an infinite \Rightarrow -rewrite sequence $\vec{\mu} = (\mu_n)_{n \in \mathbb{N}}$ such that $\text{edl}(\vec{\mu})$ is finite and $\mu_0 = \{1 : t\}$.

In terms of \rightarrow -RSTs, we define the *expected derivation length* of an \rightarrow -RST \mathfrak{T} to be $\text{edl}(\mathfrak{T}) = \sum_{v \in N^{\mathfrak{T}} \setminus \text{Leaf}^{\mathfrak{T}}} p_v$. Then, we have $\text{PAST}_{\rightarrow}$ iff $\text{edl}(\mathfrak{T})$ is finite for every \rightarrow -RST \mathfrak{T} . Similarly, we have $\text{wPAST}_{\rightarrow}$ iff for every term t there exists a fully evaluated \rightarrow -RST \mathfrak{T} whose root is labeled with $(1 : t)$ such that $\text{edl}(\mathfrak{T})$ is finite.

Remark 3.8. For every \Rightarrow -rewrite sequence $\vec{\mu}$ that converges with probability 1, we also have $\text{edl}(\vec{\mu}) = \sum_{n=0}^{\infty} (1 - |\mu_n|_{\rightarrow}) = \sum_{n=1}^{\infty} n \cdot (|\mu_n|_{\rightarrow} - |\mu_{n-1}|_{\rightarrow})$, where $|\mu_n|_{\rightarrow} - |\mu_{n-1}|_{\rightarrow}$ denotes the probability that we reach a normal form in the n -th step. This is due to the correspondence between a probability mass function $f : \mathbb{N} \rightarrow [0, 1]$ and its distribution function $F : \mathbb{N} \rightarrow [0, 1]$ w.r.t. the expected value, namely $\mathbb{E}(f) = \sum_{n=1}^{\infty} n \cdot f(n) = \sum_{n=1}^{\infty} (1 - F(n))$. With $f_{\vec{\mu}}(0) = |\mu_0|_{\rightarrow}$ and $f_{\vec{\mu}}(n) = |\mu_n|_{\rightarrow} - |\mu_{n-1}|_{\rightarrow}$ for all $n > 0$, we get $F_{\vec{\mu}}(n) = |\mu_n|_{\rightarrow}$, and hence we obtain the above equation for $\text{edl}(\vec{\mu})$. Note that $f_{\vec{\mu}}$ is only a probability mass function if $\lim_{n \rightarrow \infty} |\mu_n|_{\rightarrow} = 1$, because then $\sum_{n=0}^{\infty} f_{\vec{\mu}}(n) = |\mu_0|_{\rightarrow} + \sum_{n=1}^{\infty} (|\mu_n|_{\rightarrow} - |\mu_{n-1}|_{\rightarrow}) = \lim_{n \rightarrow \infty} |\mu_n|_{\rightarrow} = 1$.

It is well known that $\text{PAST}_{\rightarrow}$ implies AST_{\rightarrow} , but not vice versa.

Example 3.9. For every infinite extension $\vec{\mu} = (\mu_n)_{n \in \mathbb{N}}$ of the $\xrightarrow{f}_{\mathcal{P}_{\text{rw}}}$ -rewrite sequence in Ex. 3.2, the expected derivation length $\text{edl}(\vec{\mu})$ is infinite, hence $\text{wPAST}_{\xrightarrow{f}_{\mathcal{P}_{\text{rw}}}}$ does not hold, and $\text{PAST}_{\xrightarrow{f}_{\mathcal{P}_{\text{rw}}}}$, $\text{PAST}_{\xrightarrow{i}_{\mathcal{P}_{\text{rw}}}}$, or $\text{PAST}_{\xrightarrow{h}_{\mathcal{P}_{\text{rw}}}}$ do not hold either.

Next, we define *strong almost-sure termination* [FC19, ADLY20], which is even stricter than **PAST** in case of non-determinism. It requires a finite bound on the expected derivation lengths of all rewrite sequences with the same start term. For a term $t \in \mathcal{T}$, the *expected derivation height* $\text{edh}_{\rightarrow}(t)$ considers all \Rightarrow -rewrite sequences that start with $\{1 : t\}$.

Definition 3.10 (Strong Almost-Sure Termination, edh [ADLY20]). For every relation $\rightarrow \subseteq \mathcal{T} \times \text{FDist}(\mathcal{T})$ we define the *expected derivation height* of a term $t \in \mathcal{T}$ by $\text{edh}_{\rightarrow}(t) = \sup\{\text{edl}(\vec{\mu}) \mid \vec{\mu} = (\mu_n)_{n \in \mathbb{N}}$ is a \Rightarrow -rewrite sequence with $\mu_0 = \{1 : t\}\}$. We say that \rightarrow is *strongly almost-surely terminating* (**SAST** $_{\rightarrow}$) if $\text{edh}_{\rightarrow}(t)$ is finite for all $t \in \mathcal{T}$.

In terms of \rightarrow -RSTs, we have **SAST** $_{\rightarrow}$ iff $\sup\{\text{edl}(\mathfrak{T}) \mid \mathfrak{T} \text{ is an } \rightarrow\text{-RST whose root is labeled with } (1 : t)\}$ is finite for all $t \in \mathcal{T}$.

Note that in contrast to **wAST** and **wPAST**, we did not define any notion of *weak SAST*. The reason is that the definition of weak forms of termination always only requires the *existence* of some suitable rewrite sequence. But the definition of strong almost-sure termination imposes a requirement on *all* rewrite sequences. Thus, it is not clear how to obtain a useful definition for “weak **SAST**” that differs from **wPAST**.

PAST $_{\rightarrow}$ and **SAST** $_{\rightarrow}$ are already defined in terms of the expected derivation length and height, but they consider arbitrary start terms of arbitrary size. As in the non-probabilistic setting, one can also regard a function that maps the size of the start term to the expected derivation length, leading to the novel notion of the *expected derivational complexity*.

Definition 3.11 (Expected Derivational Complexity, edc_{\rightarrow}). For a relation $\rightarrow \subseteq \mathcal{T} \times \text{FDist}(\mathcal{T})$, we define its *expected derivational complexity* $\text{edc}_{\rightarrow} : \mathbb{N} \rightarrow \mathbb{N} \cup \{\omega\}$ as $\text{edc}_{\rightarrow}(n) = \sup\{\text{edh}_{\rightarrow}(t) \mid t \in \mathcal{T}, |t| \leq n\}$.

Note that as in the non-probabilistic setting, this definition uses the expected derivation height of a term, and not the expected derivation length of a rewrite sequence. Hence, the expected derivational complexity edc_{\rightarrow} relates to **SAST** $_{\rightarrow}$ more than to **PAST** $_{\rightarrow}$. Indeed, we may have both **PAST** $_{\rightarrow}$ and $\text{edc}_{\rightarrow} \in \Theta(\omega)$, see Counterex. 3.14 in the next section. Here, $\text{edc}_{\rightarrow} \in \Theta(\omega)$ means that there is some $n \in \mathbb{N}$ such that $\text{edc}_{\rightarrow}(n) = \omega$, and hence, $\text{edc}_{\rightarrow}(n') = \omega$ for all $n' \geq n$. Furthermore, we have the following easy observation regarding edc_{\rightarrow} and **SAST** $_{\rightarrow}$, where $\text{edc}_{\rightarrow} \in o(\omega)$ means that for all $n \in \mathbb{N}$ we have $\text{edc}_{\rightarrow}(n) < \omega$.

Lemma 3.12 (Relation between edc and **SAST**). *Let $\rightarrow \subseteq \mathcal{T} \times \text{FDist}(\mathcal{T})$ where $\mathcal{T} = \mathcal{T}(\Sigma, \mathcal{V})$ for a finite signature Σ . Then $\text{edc}_{\rightarrow} \in o(\omega)$ iff **SAST** $_{\rightarrow}$.*

Proof. If we do not have **SAST** $_{\rightarrow}$, then $\text{edc}_{\rightarrow} \in \Theta(\omega)$ follows directly from the definition. On the other hand, if Σ is finite, then $\text{edc}_{\rightarrow}(n) = \sup\{\text{edh}_{\rightarrow}(t) \mid t \in \mathcal{T}, |t| \leq n\}$ where $\{t \in \mathcal{T} \mid |t| \leq n\}$ is finite (up to variable renamings), so that the supremum is equal to $\text{edh}_{\rightarrow}(t)$ for a term $t \in \mathcal{T}$ with $|t| \leq n$. Hence, if $\text{edc}_{\rightarrow} \in \Theta(\omega)$, then there exists some $n \in \mathbb{N}$ and a term $t \in \mathcal{T}$ such that $|t| \leq n$ and $\text{edh}_{\rightarrow}(t) = \omega$, such that **SAST** $_{\rightarrow}$ does not hold. \square

Finally, for expected runtime complexity we additionally require basic start terms again.

Definition 3.13 (Expected Runtime Complexity, $\text{erc}_{\rightarrow, \mathcal{P}}$). For a relation $\rightarrow \subseteq \mathcal{T} \times \text{FDist}(\mathcal{T})$, the *expected runtime complexity* $\text{erc}_{\rightarrow, \mathcal{P}} : \mathbb{N} \rightarrow \mathbb{N} \cup \{\omega\}$ w.r.t. a PTRS \mathcal{P} is $\text{erc}_{\rightarrow, \mathcal{P}}(n) = \sup\{\text{edh}_{\rightarrow}(t) \mid t \in \mathcal{T}_{\mathcal{B}}, |t| \leq n\}$, where $\mathcal{T}_{\mathcal{B}}$ denotes the basic terms w.r.t. \mathcal{P} .

When considering the expected runtime complexity for a rewrite relation like $\xrightarrow{\mathcal{P}}$, we again omit the additional index \mathcal{P} , i.e., $\text{erc}_{\xrightarrow{\mathcal{P}}, \mathcal{P}} = \text{erc}_{\xrightarrow{\mathcal{P}}}$.

3.3. Relating Positive and Strong Almost-Sure Termination. We now present novel results on the relation between $\text{PAST}_{\rightarrow \mathcal{P}}$ and $\text{SAST}_{\rightarrow \mathcal{P}}$. Similar to the implication from $\text{PAST}_{\rightarrow}$ to AST_{\rightarrow} , it is well known that $\text{SAST}_{\rightarrow}$ implies $\text{PAST}_{\rightarrow}$, and this implication is again strict [ADLY20]. While the counterexample for $\text{SAST}_{\rightarrow \mathcal{P}} = \text{PAST}_{\rightarrow \mathcal{P}}$ in [ADLY20] uses infinitely many rules, the following new example shows that even for PTRSs with *finitely* many rules, $\text{SAST}_{\rightarrow \mathcal{P}}$ and $\text{PAST}_{\rightarrow \mathcal{P}}$ are not equivalent.

Counterexample 3.14. Consider the PTRS $\mathcal{P}_{\text{unary}}$ with the rules

$$\begin{array}{ll} f(x) \rightarrow \{1/2 : f(s(x)), 1/2 : b\} & h(s(x)) \rightarrow \{1 : q(h(x))\} \\ f(x) \rightarrow \{1 : g(x)\} & h(0) \rightarrow \{1 : a(a(a(0))))\} \\ g(0) \rightarrow \{1 : 0\} & q(a(x)) \rightarrow \{1 : q_1(x)\} \\ g(s(x)) \rightarrow \{1 : g_1(s(x))\} & q_1(x) \rightarrow \{1 : q_2(x)\} \\ g_1(s(x)) \rightarrow \{1 : h(s(x))\} & q_2(x) \rightarrow \{1 : a(a(a(q(x))))\} \end{array}$$

and the following rewrite sequence:

$$\begin{array}{llll} g(s^n(0)) & \xrightarrow{2}_{\mathcal{P}_{\text{unary}}} & h(s^n(0)) & \xrightarrow{n}_{\mathcal{P}_{\text{unary}}} q^n(h(0)) \\ & \rightarrow_{\mathcal{P}_{\text{unary}}} & q^n(a^4(0)) & \xrightarrow{3*4}_{\mathcal{P}_{\text{unary}}} q^{n-1}(a^{4^2}(q(0))) \\ & \xrightarrow{3*4^2}_{\mathcal{P}_{\text{unary}}} & q^{n-2}(a^{4^3}(q^2(0))) & \xrightarrow{3*4^3}_{\mathcal{P}_{\text{unary}}} \dots \\ & \xrightarrow{3*4^{n-1}}_{\mathcal{P}_{\text{unary}}} & q(a^{4^n}(q^{n-1}(0))) & \xrightarrow{3*4^n}_{\mathcal{P}_{\text{unary}}} a^{4^{n+1}}(q^n(0)) \end{array}$$

To ease readability, here we wrote $\rightarrow_{\mathcal{P}_{\text{unary}}}$ instead of $\xrightarrow{\mathbf{f}}_{\mathcal{P}_{\text{unary}}}$ and we omitted the multi-distributions, as all the used rules have trivial probabilities, i.e., they are of the form $\ell \rightarrow \{1 : r\}$ for some $\ell, r \in \mathcal{T}$. Thus, for every $n \in \mathbb{N}$ with $n > 0$, the term $g(s^n(0))$ has an expected derivation height of $2 + n + 1 + 3 * \sum_{i=1}^n 4^i = n + 3 * \sum_{i=0}^n 4^i = n + 3 * \frac{4^{n+1}-1}{3} = 4^{n+1} - 1 + n$.

Next, consider the possible derivations of $f(0)$. If we never use the rule from f to g , then the expected derivation length is $\frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \dots = \sum_{i=1}^{\infty} \frac{i}{2^i} = 2$ and if we never use the probabilistic f -rule, then the derivation length is 2 as well. Otherwise, if we use the probabilistic f -rule in the first $k \in \mathbb{N}$ steps with $k > 0$ and the rule from f to g in the $(k+1)$ -th step, then the expected derivation length is $(\sum_{i=1}^k \frac{i}{2^i}) + \frac{1}{2^k} + \frac{4^{k+1}-1+k}{2^k} = (\sum_{i=1}^k \frac{i}{2^i}) + 2^{k+2} + \frac{k}{2^k}$. Thus, for every rewrite sequence starting with $f(0)$, the expected derivation length is finite. However, since k can be any number, for any $k > 0$, there is a rewrite sequence starting with $f(0)$ whose expected derivation length is $(\sum_{i=1}^k \frac{i}{2^i}) + 2^{k+2} + \frac{k}{2^k} \geq 2^{k+2}$. In other words, although every rewrite sequence starting with $f(0)$ has finite expected derivation length, the supremum over the lengths for all these rewrite sequences is infinite. Thus, we do not have $\text{SAST}_{\rightarrow \mathcal{P}_{\text{unary}}}$, as $f(0)$ has infinite expected derivation height, but $\text{PAST}_{\rightarrow \mathcal{P}_{\text{unary}}}$ holds, as every rewrite sequence starting with $f(0)$ has finite expected derivation length and a similar argument holds for every other start term.

The PTRS in Counterex. 3.14 is of a very specific form: Its signature contains only unary symbols and constants, and the probabilistic rule $f(x) \rightarrow \{1/2 : f(s(x)), 1/2 : b\}$ is erasing. If we remove one of these properties, then $\text{PAST}_{\rightarrow \mathcal{P}_{\text{unary}}}$ does not hold either.

Example 3.15. Reconsider $\mathcal{P}_{\text{unary}}$ from Counterex. 3.14. If we extend the signature by a binary symbol c , then $\text{PAST}_{\rightarrow \mathcal{P}_{\text{unary}}}$ does not hold anymore, as we can start with the term $c(f(0), f(0))$ and consider the following rewrite sequence.

$$\begin{array}{l}
\{1 : c(f(0), f(0))\} \\
\begin{array}{l} \xrightarrow{f} \mathcal{P}_{\text{unary}} \\ \xrightarrow{f} \mathcal{P}_{\text{unary}} \\ \xrightarrow{f} \mathcal{P}_{\text{unary}} \end{array}
\begin{array}{l}
\{1/2 : \underline{c(b, f(0))}, 1/2 : c(f(s(0)), f(0))\} \\
\{ \dots, 1/4 : \underline{c(b, f(0))}, 1/4 : c(f(s^2(0)), f(0))\} \\
\{ \dots, \dots, 1/8 : \underline{c(b, f(0))}, 1/8 : c(f(s^3(0)), f(0))\}
\end{array}
\end{array}$$

Here, we use the term $f(0)$ in c 's first argument to create infinitely many copies of the term $f(0)$ in c 's second argument using the probabilistic f -rule. The underlined terms are not yet normal forms, as they contain the subterm $f(0)$, and will be evaluated further. As in Counterex. 3.14, for each $k > 0$, there is a rewrite sequence starting with $f(0)$ whose expected derivation length is at least 2^{k+2} . We can choose $k = 1$ for the first underlined term, $k = 2$ for the second, etc., leading to a final expected derivation length of at least $\sum_{i=1}^{\infty} \frac{1}{2^i} \cdot 2^{i+2} = \sum_{i=1}^{\infty} 4$, which diverges to infinity. Hence, $\text{PAST}_{\xrightarrow{f} \mathcal{P}_{\text{unary}}}$ does not hold over the extended signature. In particular, this shows that in contrast to ordinary termination for non-probabilistic TRSs, PAST is not preserved under extensions of the signature.

Similarly, if we consider $\mathcal{P}'_{\text{unary}}$ with the non-erasing rule $f(x) \rightarrow \{1/2 : f(s(x)), 1/2 : b(x)\}$ using a unary function symbol b instead of $f(x) \rightarrow \{1/2 : f(s(x)), 1/2 : b\}$, then we obtain the following rewrite sequence:

$$\begin{array}{l}
\{1 : f(f(0))\} \\
\begin{array}{l} \xrightarrow{f} \mathcal{P}_{\text{unary}} \\ \xrightarrow{f} \mathcal{P}_{\text{unary}} \\ \xrightarrow{f} \mathcal{P}_{\text{unary}} \end{array}
\begin{array}{l}
\{1/2 : \underline{b(f(0))}, 1/2 : f(s(f(0)))\} \\
\{ \dots, 1/4 : \underline{b(s(f(0)))}, 1/4 : f(s^2(f(0)))\} \\
\{ \dots, \dots, 1/8 : \underline{b(s^2(f(0)))}, 1/8 : f(s^3(f(0)))\}
\end{array}
\end{array}$$

Again, we can extend this sequence to an infinite rewrite sequence with an infinite expected derivation length. Hence, $\text{PAST}_{\xrightarrow{f} \mathcal{P}_{\text{unary}}}$ does not hold either.

Counterex. 3.14 and Ex. 3.15 show that $\text{PAST}_{\xrightarrow{s} \mathcal{P}}$ is not preserved under signature extensions for any strategy $s \in \mathbb{S}$, as the first rewrite sequence given in Ex. 3.15 is in fact a leftmost-innermost rewrite sequence.

Theorem 3.16 (Signature Extensions for $\text{PAST}_{\xrightarrow{s} \mathcal{P}}$). *Let $s \in \mathbb{S}$. There exists a PTRS \mathcal{P} and signatures Σ, Σ' with $\Sigma \subset \Sigma'$ such that $\text{PAST}_{\xrightarrow{s} \mathcal{P}}$ holds over the signature Σ , but not over Σ' .*

In contrast, when analyzing modularity of $\text{AST}_{\xrightarrow{s} \mathcal{P}}$ and $\text{SAST}_{\xrightarrow{s} \mathcal{P}}$ in Section 7, we will show that they are closed under signature extensions (see Thm. 7.15).

The core idea of both examples in Ex. 3.15 is that in the limit we can reach an infinite multi-distribution whose support contains an infinite number of terms like $f(0)$ with unbounded expected derivation height. A PTRS that allows such sequences is said to *admit infinite splits*.

Definition 3.17 (Infinite Splits). A PTRS \mathcal{P} *admits infinite splits* if for every term $t \in \mathcal{T}$ there exists a term $t' \in \mathcal{T}$ and an $\xrightarrow{f} \mathcal{P}$ -RST whose root is labeled with $(1 : t')$ such that there are infinitely many leaves labeled with terms that have t as a subterm.

Example 3.18. $\mathcal{P}_{\text{unary}}$ over the signature containing a function symbol of arity ≥ 2 and the non-erasing PTRS $\mathcal{P}'_{\text{unary}}$ both admit infinite splits. However, $\mathcal{P}_{\text{unary}}$ over the signature $\{f, b, s, 0, g, g_1, h, a, q, q_1, q_2\}$ does not admit infinite splits.

As shown by the following theorem, admitting infinite splits implies that $\text{PAST}_{\xrightarrow{f} \mathcal{P}}$ is the same as $\text{SAST}_{\xrightarrow{f} \mathcal{P}}$.

Theorem 3.19 (Equivalence of $\text{PAST}_{\mathcal{P}}$ and $\text{SAST}_{\mathcal{P}}$ via Infinite Splits). *If a PTRS \mathcal{P} admits infinite splits, then:*

$$\text{PAST}_{\mathcal{P}} \iff \text{SAST}_{\mathcal{P}}$$

Proof. We only have to prove “ \implies ”, and do this via contraposition. Assume that $\text{SAST}_{\mathcal{P}}$ does not hold. Then there exists a term $t \in \mathcal{T}$ such that for every $n \in \mathbb{N}$ there exists an \mathcal{P} -RST \mathfrak{T}_n^t whose root is labeled with $(1 : t)$ such that $\text{edl}(\mathfrak{T}_n^t) \geq n$. We now construct a single \mathcal{P} -RST \mathfrak{T}^∞ with $\text{edl}(\mathfrak{T}^\infty) = \infty$, which implies that $\text{PAST}_{\mathcal{P}}$ does not hold either. Since \mathcal{P} admits infinite splits, there exists a term $t' \in \mathcal{T}$ and an \mathcal{P} -RST \mathfrak{T} whose root is labeled with $(1 : t')$ such that there are infinitely many leaves whose corresponding terms have t as a subterm. Let v be a leaf in \mathfrak{T} with $t_v^\mathfrak{T} = C[t]$ for some context C and let $n_v \in \mathbb{N}$ be such that $p_v^\mathfrak{T} \geq \frac{1}{2^{n_v}}$. Then we can replace the leaf by the \mathcal{P} -RST $\mathfrak{T}_{2^{n_v}}^{C[t]}$. Here, $\mathfrak{T}_{2^{n_v}}^{C[t]}$ is the same tree as $\mathfrak{T}_{2^{n_v}}^t$ where in addition we have the context C around every term of every node. Hence, $\text{edl}(\mathfrak{T}_{2^{n_v}}^{C[t]}) = \text{edl}(\mathfrak{T}_{2^{n_v}}^t) \geq 2^{n_v}$. Let \mathfrak{T}^∞ be the \mathcal{P} -RST that results from performing this replacement for every leaf v in \mathfrak{T} that contains t as a subterm. Then, for \mathfrak{T}^∞ we have

$$\begin{aligned} \text{edl}(\mathfrak{T}^\infty) &\geq \sum_{v \in \text{Leaf}^\mathfrak{T} \wedge t_v^\mathfrak{T} = C[t] \text{ for some context } C} p_v^\mathfrak{T} \cdot \text{edl}(\mathfrak{T}_{2^{n_v}}^{C[t]}) \\ &\geq \sum_{v \in \text{Leaf}^\mathfrak{T} \wedge t_v^\mathfrak{T} = C[t]} \frac{1}{2^{n_v}} \cdot 2^{n_v} = \sum_{v \in \text{Leaf}^\mathfrak{T} \wedge t_v^\mathfrak{T} = C[t]} 1 = \infty \quad \square \end{aligned}$$

Remark 3.20. Thm. 3.19 can also be adapted to strategies like innermost or leftmost-innermost rewriting, but then one has to ensure that in the infinitely many leaves of the RST in Def. 3.17, the subterm t can be used as the next redex according to the respective strategy.

As an application of Thm. 3.19 we give two syntactical criteria that ensure that $\text{PAST}_{\mathcal{P}}$ is equivalent to $\text{SAST}_{\mathcal{P}}$ for a given PTRS \mathcal{P} , where both criteria are very easy to check automatically. The first one (illustrated by the first PTRS in Ex. 3.15) states that if we consider only PTRSs with finitely many rules, then the existence of a function symbol of arity at least 2 suffices for equivalence of $\text{PAST}_{\mathcal{P}}$ and $\text{SAST}_{\mathcal{P}}$. Thus, this novel observation shows that for almost all finite PTRSs in practice, there is no difference between $\text{PAST}_{\mathcal{P}}$ and $\text{SAST}_{\mathcal{P}}$.

Theorem 3.21 (Equivalence of $\text{PAST}_{\mathcal{P}}$ and $\text{SAST}_{\mathcal{P}}$ (1)). *If a PTRS \mathcal{P} has only finitely many rules and the corresponding signature contains a function symbol of at least arity 2, then:*

$$\text{PAST}_{\mathcal{P}} \iff \text{SAST}_{\mathcal{P}}$$

Proof. Let \mathcal{P} contain only finitely many rules and assume for a contradiction that we have $\text{PAST}_{\mathcal{P}}$ but not $\text{SAST}_{\mathcal{P}}$. Then there exists a term t' and an \mathcal{P} -RST whose root is labeled with $(1 : t')$ which has infinitely many leaves. (If there were only finitely many leaves for every term, then we would either have $\text{SAST}_{\mathcal{P}}$ if there exists no infinite path, or we would not have $\text{PAST}_{\mathcal{P}}$ if there exists an infinite path.) Let $t \in \mathcal{T}$ be an arbitrary term, and c be a function symbol of arity ≥ 2 . We can now construct an \mathcal{P} -RST that starts with $(1 : c(t', t, \dots, t))$ such that there are infinitely many leaves labeled with terms that have t as a subterm. To do so, one simply restricts rewriting to the first argument of c . Note that this construction is only possible because we assume that there is a function symbol of at least arity 2, such that both t and t' can be subterms of the same term $c(t', t, \dots, t)$. This

shows that \mathcal{P} admits infinite splits, and thus, we have $\mathbf{SAST}_{\mathcal{P}}$ by Thm. 3.19, which is our desired contradiction. \square

The second sufficient criterion for equivalence of $\mathbf{PAST}_{\mathcal{P}}$ and $\mathbf{SAST}_{\mathcal{P}}$ (illustrated by the second PTRS in Ex. 3.15) requires specific forms of “non-erasing loops”.

Theorem 3.22 (Equivalence of $\mathbf{PAST}_{\mathcal{P}}$ and $\mathbf{SAST}_{\mathcal{P}}$ (2)). *If there exists a probabilistic rule $\ell \rightarrow \{p_1 : C[\ell\sigma], p_2 : s, \dots\}$ such that there is a variable $x \in \mathcal{V}(\ell)$ with $x \in \mathcal{V}(x\sigma) \cap \mathcal{V}(s)$, then:*

$$\mathbf{PAST}_{\mathcal{P}} \iff \mathbf{SAST}_{\mathcal{P}}$$

Proof. Let $t \in \mathcal{T}$ be an arbitrary term, and let δ be a substitution such that $x\delta = t$ for a variable $x \in \mathcal{V}(x\sigma) \cap \mathcal{V}(s)$. We can now construct an \mathcal{P} -RST that starts with $(1 : \ell\delta)$ such that there are infinitely many leaves labeled with terms that have t as a subterm. To do so, one rewrites the redex $\ell\delta$ with the rule $\ell \rightarrow \{p_1 : C[\ell\sigma], p_2 : s, \dots\}$, leading to a new leaf labeled with the term $s\delta$ (containing the subterm t since $x \in \mathcal{V}(s)$) and a new node labeled with the term $C[\ell\sigma]\delta = C\delta[\ell\sigma\delta]$, where we can rewrite the redex $\ell\sigma\delta$ again. This in turn will lead to a leaf labeled with $s\sigma\delta$ (which again contains t since $x \in \mathcal{V}(x\sigma) \cap \mathcal{V}(s)$ implies $x \in \mathcal{V}(s\sigma)$) and a new node labeled with a term containing $\ell\sigma^2\delta$, etc. Hence, \mathcal{P} admits infinite splits and the theorem is implied by Thm. 3.19. \square

Thm. 3.22 can also be extended so that the loop does not consist of a single rewrite step, but we can have arbitrary many steps during the loop.

4. RELATING VARIANTS OF PROBABILISTIC TERMINATION AND EXPECTED COMPLEXITY

Our goal is to relate the different probabilistic termination properties (\mathbf{AST} , \mathbf{PAST} , and \mathbf{SAST}) and the expected complexity of full rewriting to the respective properties of innermost rewriting (Section 4.1), weak termination (Section 4.2), and leftmost-innermost rewriting (Section 4.3). More precisely, we want to find properties of a PTRS \mathcal{P} which are suitable for automated checking and which guarantee that, e.g., $\mathbf{AST}_{\mathcal{P}} \iff \mathbf{AST}'_{\mathcal{P}}$ for $s, s' \in \mathbb{S}$. Then, for example, we can use existing tools that analyze $\mathbf{AST}_{\mathcal{P}}$ in order to prove $\mathbf{AST}'_{\mathcal{P}}$, if we have successfully checked the properties that guarantee equivalence of $\mathbf{AST}_{\mathcal{P}}$ and $\mathbf{AST}'_{\mathcal{P}}$. Let $\mathbf{PSN} \in \{\mathbf{AST}, \mathbf{PAST}, \mathbf{SAST}\}$. As most of our results hold for all these three termination properties, we use \mathbf{PSN} (*probabilistic strong normalization*) to refer to all of them. Similarly, we use $\mathbf{wPSN} \in \{\mathbf{wAST}, \mathbf{wPAST}\}$. Clearly, we have to require at least the same properties as in the non-probabilistic setting, as every TRS \mathcal{R} can be transformed into a PTRS \mathcal{P} by replacing every rule $\ell \rightarrow r$ with $\ell \rightarrow \{1 : r\}$. Then for every $s \in \mathbb{S}$, we have $\mathbf{SN}_{\mathcal{R}} \iff \mathbf{AST}_{\mathcal{P}}$ and $\mathbf{WN}_{\mathcal{R}} \iff \mathbf{wAST}_{\mathcal{P}}$.

The following subsections are all structured as follows: We first give examples to show why the criteria from the non-probabilistic setting do not carry over to the probabilistic setting. Then, we explain the criteria needed in the probabilistic setting, state the corresponding theorem, and usually give a lemma to explain the main proof idea. We also use these lemmas to reason about expected complexity.

4.1. From $\text{PSN}_{\rightarrow \mathcal{P}}$ to $\text{PSN}_{\rightarrow \mathcal{P}}$. We start by analyzing the relation between innermost and full rewriting. The following example shows that Thm. 2.2 does not carry over to the probabilistic setting, i.e., orthogonality is not sufficient to ensure that $\text{PSN}_{\rightarrow \mathcal{P}}$ implies $\text{PSN}_{\rightarrow \mathcal{P}}$.

Counterexample 4.1 (Orthogonality Does Not Suffice). Consider the orthogonal PTRS \mathcal{P}_1 with the two rules:

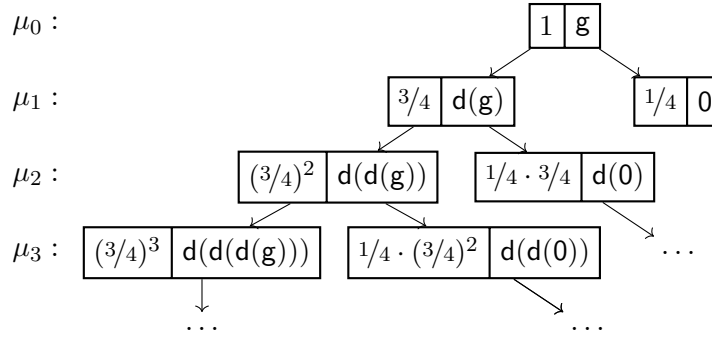
$$g \rightarrow \{3/4 : d(g), 1/4 : 0\} \qquad d(x) \rightarrow \{1 : c(x, x)\}$$

We do not have $\text{AST}_{\rightarrow \mathcal{P}_1}$ (hence also neither $\text{PAST}_{\rightarrow \mathcal{P}_1}$ nor $\text{SAST}_{\rightarrow \mathcal{P}_1}$), because $\{1 : g\} \xrightarrow{\mathbf{f}_2}_{\mathcal{P}_1} \{3/4 : c(g, g), 1/4 : 0\}$, which corresponds to a random walk biased towards non-termination (since $\frac{3}{4} > \frac{1}{4}$).

However, the d -rule can only duplicate normal forms in innermost evaluations. To see that we have $\text{SAST}_{\rightarrow \mathcal{P}_1}$ (hence $\text{PAST}_{\rightarrow \mathcal{P}_1}$ and $\text{AST}_{\rightarrow \mathcal{P}_1}$), consider the only possible innermost rewrite sequence $\vec{\mu}$ starting with $\{1 : g\}$:

$$\{1 : g\} \xrightarrow{\mathbf{i}}_{\mathcal{P}_1} \{3/4 : d(g), 1/4 : 0\} \xrightarrow{\mathbf{i}}_{\mathcal{P}_1} \{(3/4)^2 : d(d(g)), 1/4 \cdot 3/4 : d(0), 1/4 : 0\} \xrightarrow{\mathbf{i}}_{\mathcal{P}_1} \dots$$

We can also view this rewrite sequence as an $\xrightarrow{\mathbf{i}}_{\mathcal{P}_1}$ -RST:



The branch to the right that starts with 0 stops after 0 innermost steps, the branch that starts with $d(0)$ stops after 1 innermost step, the branch that starts with $d(d(0))$ stops after 2 innermost steps, and so on. So if we start with the term $d^n(0)$, then we reach a normal form after n steps, and we reach $d^n(0)$ after $n + 1$ steps from the initial term g . For every $k \in \mathbb{N}$ we have $|\mu_{2 \cdot k+1}|_{\mathcal{P}_1} = |\mu_{2 \cdot k+2}|_{\mathcal{P}_1} = \sum_{n=0}^k 1/4 \cdot (3/4)^n$ and thus

$$\begin{aligned} \text{edl}(\vec{\mu}) &= \sum_{n=0}^{\infty} (1 - |\mu_n|_{\mathcal{P}_1}) &= 1 + 2 \cdot \sum_{k \in \mathbb{N}} (1 - |\mu_{2 \cdot k+1}|_{\mathcal{P}_1}) \\ &= 1 + 2 \cdot \sum_{k \in \mathbb{N}} (1 - \sum_{n=0}^k 1/4 \cdot (3/4)^n) &= 1 + 2 \cdot \sum_{k \in \mathbb{N}} (3/4)^{k+1} \\ &= (2 \cdot \sum_{k \in \mathbb{N}} (3/4)^k) - 1 &= 7 \end{aligned}$$

Due to innermost rewriting, there is no non-determinism in this sequence, i.e., when starting with $\{1 : g\}$, there is no rewrite sequence with higher expected derivation length. Thus, we also have $\text{edh}_{\rightarrow \mathcal{P}_1}(g) = 7$. Analogously, in all other innermost rewrite sequences, the d -rule can also only duplicate normal forms. Thus, all terms have finite expected derivation height w.r.t. innermost rewriting. Therefore, $\text{SAST}_{\rightarrow \mathcal{P}_1}$ and thus, also $\text{PAST}_{\rightarrow \mathcal{P}_1}$ and $\text{AST}_{\rightarrow \mathcal{P}_1}$ hold. The latter can also be proved automatically by our implementation of the probabilistic DP framework for $\text{AST}_{\rightarrow \mathcal{P}}$ [KG23a, KDG24] in AProVE.

To construct a counterexample, we exploited the fact that \mathcal{P}_1 is not right-linear, which allows us to duplicate the redex g repeatedly during the rewrite sequence. Similar to the complexity analysis in the non-probabilistic setting (see Section 2.2), we need to prohibit

the duplication of redexes. Indeed, requiring right-linearity prevents this kind of duplication and yields our desired result.

Theorem 4.2 (From $\text{PSN}_{\rightarrow \mathcal{P}}^i$ to $\text{PSN}_{\rightarrow \mathcal{P}}^f$). *If a PTRS \mathcal{P} is OR and RL (i.e., NO and linear), then:*

$$\text{PSN}_{\rightarrow \mathcal{P}}^f \iff \text{PSN}_{\rightarrow \mathcal{P}}^i$$

For the proof of Thm. 4.2, we prove the following lemma which directly implies Thm. 4.2.

Lemma 4.3 (From Innermost to Full Rewriting). *If a PTRS \mathcal{P} is OR and RL (i.e., NO and linear), then for every infinite $\xrightarrow{f}_{\mathcal{P}}$ -rewrite sequence $\vec{\mu} = (\mu_n)_{n \in \mathbb{N}}$ there exists an infinite $\xrightarrow{i}_{\mathcal{P}}$ -rewrite sequence $\vec{\nu} = (\nu_n)_{n \in \mathbb{N}}$ such that*

- (i) $\lim_{n \rightarrow \infty} |\mu_n|_{\mathcal{P}} \geq \lim_{n \rightarrow \infty} |\nu_n|_{\mathcal{P}}$
- (ii) $\text{edl}(\vec{\mu}) \leq \text{edl}(\vec{\nu})$

As mentioned, all missing complete proofs can be found in App. A.

Proof Sketch. The proofs for all lemmas in this section follow a similar structure. We always iteratively replace rewrite steps by ones that use the desired strategy and ensure that this neither increases the probability of convergence nor decreases the expected derivation length. For this replacement, we lift the corresponding construction from the non-probabilistic to the probabilistic setting. However, this cannot be done directly but instead, we have to regard the limit of a sequence of transformation steps.

Let \mathcal{P} be a PTRS that is non-overlapping, linear, and there exists an infinite $\xrightarrow{f}_{\mathcal{P}}$ -rewrite sequence $\vec{\mu} = (\mu_n)_{n \in \mathbb{N}}$ such that $\lim_{n \rightarrow \infty} |\mu_n|_{\mathcal{P}} = c$ for some $c \in \mathbb{R}$ with $0 \leq c < 1$. Our goal is to transform this sequence into an innermost sequence that converges with at most probability c . If the sequence is not yet an innermost one, then in $(\mu_n)_{n \in \mathbb{N}}$ at least one rewrite step is performed with a redex that is not an innermost redex. Since \mathcal{P} is non-overlapping, we can replace a first such non-innermost rewrite step with an innermost rewrite step using a similar construction as in the non-probabilistic setting. In this way, we result in a rewrite sequence $\vec{\mu}^{(1)} = (\mu_n^{(1)})_{n \in \mathbb{N}}$ with $\lim_{n \rightarrow \infty} |\mu_n^{(1)}|_{\mathcal{P}} = \lim_{n \rightarrow \infty} |\mu_n|_{\mathcal{P}} = c$. Here, linearity is needed to ensure that the probability of convergence does not increase during this replacement. We can then repeat this replacement for every non-innermost rewrite step, i.e., we again replace a first non-innermost rewrite step in $(\mu_n^{(1)})_{n \in \mathbb{N}}$ to obtain $(\mu_n^{(2)})_{n \in \mathbb{N}}$ with the same convergence probability, etc. In the end, the limit of all these rewrite sequences $\vec{\mu}^{(\infty)} = \lim_{i \rightarrow \infty} (\mu_n^{(i)})_{n \in \mathbb{N}}$ is an innermost rewrite sequence that converges with probability at most c .

Regarding the expected derivation length, we can use exactly the same construction, as this also guarantees that in each step, $\vec{\mu}^{(1)}$ does not only converge with the same probability as $\vec{\mu}$, but we also have $\text{edl}(\vec{\mu}^{(1)}) \geq \text{edl}(\vec{\mu})$ and $\text{edl}(\vec{\mu}^{(i+1)}) \geq \text{edl}(\vec{\mu}^{(i)})$ for all $i > 0$. So in the end, the limit of all these rewrite sequences $\vec{\mu}^{(\infty)}$ is an innermost rewrite sequence with $\text{edl}(\vec{\mu}^{(\infty)}) \geq \text{edl}(\vec{\mu})$. \square

Proof of Thm. 4.2. We only need to prove the “ \Leftarrow ” direction. Let us first consider $\text{AST}_{\rightarrow \mathcal{P}}^f$. Assume that \mathcal{P} is orthogonal, right-linear, and that $\text{AST}_{\rightarrow \mathcal{P}}^f$ does not hold. Then, there exists an infinite $\xrightarrow{f}_{\mathcal{P}}$ -rewrite sequence $\vec{\mu} = (\mu_n)_{n \in \mathbb{N}}$ such that $\lim_{n \rightarrow \infty} |\mu_n|_{\mathcal{P}} < 1$. By Lemma 4.3 we obtain an infinite $\xrightarrow{i}_{\mathcal{P}}$ -rewrite sequence $\vec{\nu} = (\nu_n)_{n \in \mathbb{N}}$ such that $\lim_{n \rightarrow \infty} |\nu_n|_{\mathcal{P}} \leq \lim_{n \rightarrow \infty} |\mu_n|_{\mathcal{P}} < 1$,

and thus, $\text{AST}_{\rightarrow \mathcal{P}}$ does not hold either. The argument is completely analogous for $\text{PAST}_{\rightarrow \mathcal{P}}$, just reasoning about the expected derivation length.

For $\text{SAST}_{\rightarrow \mathcal{P}}$ we have two cases: If there exists a single $\xrightarrow{\mathbf{f}}_{\mathcal{P}}$ -rewrite sequence $\vec{\mu} = (\mu_n)_{n \in \mathbb{N}}$ such that $\text{edl}(\vec{\mu}) = \omega$, then we proceed as for $\text{PAST}_{\rightarrow \mathcal{P}}$. Otherwise, there exists an infinite set $\{\vec{\mu}^{(i)} \mid i \in \mathbb{N}\}$ of $\xrightarrow{\mathbf{f}}_{\mathcal{P}}$ -rewrite sequences $\vec{\mu}^{(i)}$ with the same initial multi-distribution such that $\sup\{\text{edl}(\vec{\mu}^{(i)}) \mid i \in \mathbb{N}\} = \omega$. For each of these rewrite sequences we apply Lemma 4.3 as before such that we obtain *innermost* rewrite sequences $\vec{\nu}^{(i)}$ with $\text{edl}(\vec{\nu}^{(i)}) \geq \text{edl}(\vec{\mu}^{(i)})$ for all $i \in \mathbb{N}$. Thus, $\{\vec{\nu}^{(i)} \mid i \in \mathbb{N}\}$ is an infinite set of innermost rewrite sequences with the same initial multi-distribution such that $\sup\{\text{edl}(\vec{\nu}^{(i)}) \mid i \in \mathbb{N}\} = \omega$, which proves that $\text{SAST}_{\rightarrow \mathcal{P}}$ does not hold either. \square

One may wonder whether we can remove the left-linearity requirement from Thm. 4.2, as in the non-probabilistic setting. It turns out that this is not possible.

Counterexample 4.4 (Left-Linearity Cannot be Removed). Consider the PTRS \mathcal{P}_2 with the rules:

$$f(x, x) \rightarrow \{1 : f(a, a)\} \qquad a \rightarrow \{1/2 : b, 1/2 : c\}$$

We do not have $\text{AST}_{\rightarrow \mathcal{P}_2}$ (hence also neither $\text{PAST}_{\rightarrow \mathcal{P}_2}$ nor $\text{SAST}_{\rightarrow \mathcal{P}_2}$), since $\{1 : f(a, a)\} \xrightarrow{\mathbf{f}}_{\mathcal{P}_2} \{1 : f(a, a)\} \xrightarrow{\mathbf{f}}_{\mathcal{P}_2} \dots$ is an infinite rewrite sequence that converges with probability 0. However, we have $\text{SAST}_{\rightarrow \mathcal{P}_2}$ (and hence, $\text{PAST}_{\rightarrow \mathcal{P}_2}$ and $\text{AST}_{\rightarrow \mathcal{P}_2}$) since the corresponding innermost sequence has the form $\{1 : f(a, a)\} \xrightarrow{\mathbf{i}}_{\mathcal{P}_2} \{\frac{1}{2} : f(b, a), \frac{1}{2} : f(c, a)\} \xrightarrow{\mathbf{i}}_{\mathcal{P}_2} \{\frac{1}{4} : f(b, b), \frac{1}{4} : f(b, c), \frac{1}{4} : f(c, b), \frac{1}{4} : f(c, c)\} \xrightarrow{\mathbf{i}}_{\mathcal{P}_2} \dots$. Here, the last distribution contains two normal forms $f(b, c)$ and $f(c, b)$ that did not occur in the previous rewrite sequence, so that the expected derivation length of this $\xrightarrow{\mathbf{i}}_{\mathcal{P}_2}$ -rewrite sequence is $2 + 3 \cdot \sum_{i=1}^{\infty} (1/2)^i = 5$. Since all innermost rewrite sequences keep on adding such normal forms after a constant number of steps for each start term, $\text{edh}_{\rightarrow \mathcal{P}_2}(t)$ is finite for each $t \in \mathcal{T}$ (again, $\text{AST}_{\rightarrow \mathcal{P}_2}$ can be shown automatically by AProVE). Note that adding the requirement of being non-erasing would not help to get rid of the left-linearity requirement, as shown by the PTRS \mathcal{P}_3 which results from \mathcal{P}_2 by replacing the f -rule with $f(x, x) \rightarrow \{1 : d(f(a, a), x)\}$.

The problem here is that although we rewrite both occurrences of a with the same rewrite rule, the two a -symbols are replaced by two different terms (each with a probability > 0). This would be impossible in the non-probabilistic setting.

Next, one could try to adapt Thm. 2.4 to the probabilistic setting (when requiring linearity in addition). So one could investigate whether $\text{PSN}_{\rightarrow \mathcal{P}}$ implies $\text{PSN}_{\rightarrow \mathcal{P}}$ for PTRSs that are linear locally confluent overlay systems. A PTRS \mathcal{P} is *locally confluent* if for all multi-distributions μ, μ_1, μ_2 such that $\mu_1 \xrightarrow{\mathbf{f}}_{\mathcal{P}} \mu \xrightarrow{\mathbf{f}}_{\mathcal{P}} \mu_2$, there exists a multi-distribution μ' such that $\mu_1 \xrightarrow{\mathbf{f}}_{\mathcal{P}} \mu' \xrightarrow{\mathbf{f}}_{\mathcal{P}} \mu_2$, see [DCM18]. Note that in contrast to the probabilistic setting, there are non-overlapping PTRSs that are not locally confluent (e.g., the variant \mathcal{P}'_2 of \mathcal{P}_2 that consists of the rules $f(x, x) \rightarrow \{1 : d\}$ and $a \rightarrow \{1/2 : b, 1/2 : c\}$, since we have $\{1 : d\} \xrightarrow{\mathbf{f}}_{\mathcal{P}'_2} \{1 : f(a, a)\} \xrightarrow{\mathbf{f}}_{\mathcal{P}'_2} \{1/2 : f(b, a), 1/2 : f(c, a)\}$ and the two resulting multi-distributions are not joinable). Whether every linear, non-overlapping PTRS is locally confluent is an open problem, thus, it is open whether an adaption of Thm. 2.4 would subsume Thm. 4.2 as in the non-probabilistic setting.

In contrast to the proof of Thm. 2.2, the proof of Thm. 2.4 relies on a minimality requirement for the used redex. In the non-probabilistic setting, whenever a term t starts an infinite rewrite sequence, then there exists a *minimal* infinite rewrite sequence beginning

with t , where one only reduces redexes whose proper subterms are terminating. However, such minimal infinite sequences do not always exist in the probabilistic setting.

Example 4.5 (No Minimal Infinite Rewrite Sequence for $\text{AST}_{\mathcal{F}, \mathcal{P}}$ and $\text{PAST}_{\mathcal{F}, \mathcal{P}}$). Reconsider the PTRS \mathcal{P}_1 from Counterex. 4.1, where $\text{AST}_{\mathcal{F}, \mathcal{P}_1}$ does not hold. However, there is no minimal rewrite sequence with convergence probability < 1 . If we always rewrite the proper subterm \mathbf{g} of the redex $\mathbf{d}(\mathbf{g})$, then this yields a rewrite sequence that converges with probability 1, like the $\xrightarrow{i}_{\mathcal{P}_1}$ -rewrite sequence in Counterex. 4.1. Hence, a rewrite sequence $\vec{\mu}$ with convergence probability < 1 would have to eventually use the \mathbf{d} -rule on a term of the form $\mathbf{d}(t)$ where t contains \mathbf{g} . But then $\vec{\mu}$ is not minimal since \mathbf{g} itself starts a rewrite sequence with convergence probability < 1 .

To simplify the argumentation for $\text{PAST}_{\mathcal{F}, \mathcal{P}_1}$, let us replace the rule $\mathbf{d}(x) \rightarrow \{1 : \mathbf{c}(x, x)\}$ by $\mathbf{d}(\mathbf{g}) \rightarrow \{1 : \mathbf{c}(\mathbf{g}, \mathbf{g})\}$ such that we can only duplicate \mathbf{g} and no other symbols. Then the same argumentation as for $\text{AST}_{\mathcal{F}, \mathcal{P}_1}$ above shows that there is also no minimal non- $\text{PAST}_{\mathcal{F}, \mathcal{P}_1}$ sequence, i.e., no minimal rewrite sequence $\vec{\mu}$ with $\text{edl}(\vec{\mu}) = \infty$. Again, if we always rewrite the proper subterm \mathbf{g} if possible, then the expected derivation length of this sequence would be finite. Thus, we have to rewrite \mathbf{d} eventually, although its argument contains \mathbf{g} . However, then the rewrite sequence is not minimal.

It remains open whether one can also adapt Thm. 2.4 to the probabilistic setting (e.g., if one can replace non-overlappingness in Thm. 4.2 by the requirement of locally confluent overlay systems). There are two main difficulties when trying to adapt the proof of Thm. 2.4 to PTRSs. First, the minimality requirement cannot be imposed in the probabilistic setting, as discussed above. In the non-probabilistic setting, this requirement is needed to ensure that any subterm below a position that was reduced in the original (minimal) infinite rewrite sequence is terminating. Second, the original proof of Thm. 2.4 uses Newman's Lemma [New42] which states that local confluence implies confluence for strongly normalizing terms t , and thus it implies that t has a unique normal form. Local confluence and adaptations of the unique normal form property for the probabilistic setting have been studied in [DCM18, Fag22], who concluded that obtaining an analogous statement to Newman's Lemma for PTRSs that are AST would be very difficult. The reason is that one cannot use well-founded induction on the length of a rewrite sequence of a PTRS that is AST , since these rewrite sequences may be infinite.

We can also use Lemma 4.3 for expected complexity, leading to the following result.

Theorem 4.6 (From Innermost to Full Expected Complexity). *If a PTRS \mathcal{P} is OR and RL (i.e., NO and linear), then:*

$$\text{edc}_{\mathcal{F}, \mathcal{P}} = \text{edc}_{\mathcal{I}, \mathcal{P}} \quad \text{and} \quad \text{erc}_{\mathcal{F}, \mathcal{P}} = \text{erc}_{\mathcal{I}, \mathcal{P}}$$

Proof. Similar to the proof of Thm. 4.2, this is a direct consequence of Lemma 4.3. □

While it is open whether an adaption to locally confluent overlay systems would work for $\text{PSN}_{\mathcal{I}, \mathcal{P}}$, we can give a counterexample to show that one cannot weaken the requirement of NO to overlay systems for expected complexity, i.e., Thm. 4.6 does not hold for linear overlay systems in general. In contrast, in the non-probabilistic setting we have $\text{rc}_{\mathcal{F}, \mathcal{R}} = \text{rc}_{\mathcal{I}, \mathcal{R}}$ for all right-linear overlay systems (since right-linearity implies sparseness), see Thm. 2.9.

Example 4.7. Consider the PTRS \mathcal{P}_4 with the six rules:

$$\begin{array}{ll} f(x) \rightarrow \{1/2 : g(x), 1/2 : h(x)\} & d \rightarrow \{1 : f(a)\} \\ g(b) \rightarrow \{1 : f(a)\} & a \rightarrow \{1 : b\} \\ h(c) \rightarrow \{1 : f(a)\} & a \rightarrow \{1 : c\} \end{array}$$

\mathcal{P}_4 is a linear overlay system. Moreover, $\text{erc}_{\mathbf{f}_{\rightarrow \mathcal{P}}} \in \Theta(\omega)$ due to the infinite $\mathbf{f}_{\rightarrow \mathcal{P}_4}$ -rewrite sequence $\{1 : d\} \xrightarrow{\mathbf{f}_{\rightarrow \mathcal{P}_4}} \{1 : f(a)\} \xrightarrow{\mathbf{f}_{\rightarrow \mathcal{P}_4}} \{1/2 : g(a), 1/2 : h(a)\} \xrightarrow{\mathbf{f}_{\rightarrow \mathcal{P}_4}} \{1/2 : g(b), 1/2 : h(c)\} \xrightarrow{\mathbf{f}_{\rightarrow \mathcal{P}_4}} \{1/2 : f(a), 1/2 : f(a)\}$ that converges with probability 0. But for $\mathbf{i}_{\rightarrow \mathcal{P}_4}$ -rewrite sequences we have to rewrite the argument a of $f(a)$ first, leading to a normal form with a chance of $1/2$ after two steps. Hence, $\text{erc}_{\mathbf{i}_{\rightarrow \mathcal{P}}} \in \mathcal{O}(1)$. For expected derivational complexity, we get $\text{edc}_{\mathbf{f}_{\rightarrow \mathcal{P}}} \in \Theta(\omega)$ and $\text{edc}_{\mathbf{i}_{\rightarrow \mathcal{P}}} \in \mathcal{O}(n)$ via similar arguments. Note that we have $\text{edc}_{\mathbf{i}_{\rightarrow \mathcal{P}}} \in \mathcal{O}(n)$ and not $\text{edc}_{\mathbf{i}_{\rightarrow \mathcal{P}}} \in \mathcal{O}(1)$, since for derivational complexity we allow start terms like $f^n(a)$ where n steps are needed to reach a multi-distribution that also contains normal forms.

Clearly \mathcal{P}_4 is not locally confluent. Similar as for $\text{PSN}_{\mathbf{i}_{\rightarrow \mathcal{P}}}$, it remains open whether there is an adaption of Thm. 4.6 for locally confluent overlay systems.

In Section 5, we will show that for left-linear, non-overlapping, and *sparse* PTRSs \mathcal{P} we still have $\text{erc}_{\mathbf{f}_{\rightarrow \mathcal{P}}} = \text{erc}_{\mathbf{i}_{\rightarrow \mathcal{P}}}$, see Thm. 5.12. In other words, RL can be weakened to SP.

Note that Thm. 4.6 of course also holds for PTRSs with trivial probabilities, i.e., if all rules have the form $\ell \rightarrow \{1 : r\}$. Thus, we can use Thm. 4.6 to obtain the first result on the relation between full and innermost derivational complexity in the non-probabilistic setting.

Corollary 4.8 (From Innermost to Full Complexity). *If a TRS \mathcal{R} is OR and RL (i.e., NO and linear), then:*

$$\text{dc}_{\mathbf{f}_{\rightarrow \mathcal{R}}} = \text{dc}_{\mathbf{i}_{\rightarrow \mathcal{R}}}$$

4.2. From $\mathbf{wPSN}_{\mathbf{f}_{\rightarrow \mathcal{P}}}$ to $\mathbf{PSN}_{\mathbf{f}_{\rightarrow \mathcal{P}}}$. Next, we investigate $\mathbf{wPSN}_{\mathbf{f}_{\rightarrow \mathcal{P}}}$. Since $\mathbf{PSN}_{\mathbf{i}_{\rightarrow \mathcal{P}}}$ implies $\mathbf{wPSN}_{\mathbf{f}_{\rightarrow \mathcal{P}}}$, we essentially have the same problems as for $\mathbf{PSN}_{\mathbf{i}_{\rightarrow \mathcal{P}}}$, i.e., in addition to non-overlappingness, we need linearity. This can be seen in Counterex. 4.1 and 4.4, as for $i \in \{1, 3\}$ we have $\mathbf{PSN}_{\mathbf{i}_{\rightarrow \mathcal{P}_i}}$ (and hence $\mathbf{wPSN}_{\mathbf{f}_{\rightarrow \mathcal{P}_i}}$) but not $\mathbf{PSN}_{\mathbf{f}_{\rightarrow \mathcal{P}_i}}$, while \mathcal{P}_1 and \mathcal{P}_3 are non-overlapping and non-erasing, but not linear. Furthermore, we need non-erasingness as we did in the non-probabilistic setting for the same reasons, see Counterex. 2.5.

Theorem 4.9 (From $\mathbf{wPSN}_{\mathbf{f}_{\rightarrow \mathcal{P}}}$ to $\mathbf{PSN}_{\mathbf{f}_{\rightarrow \mathcal{P}}}$). *If a PTRS \mathcal{P} is NO, linear, and NE, then*

$$\mathbf{PSN}_{\mathbf{f}_{\rightarrow \mathcal{P}}} \iff \mathbf{wPSN}_{\mathbf{f}_{\rightarrow \mathcal{P}}}$$

4.3. From $\mathbf{PSN}_{\mathbf{i}_{\rightarrow \mathcal{P}}}$ to $\mathbf{PSN}_{\mathbf{f}_{\rightarrow \mathcal{P}}}$. Finally, we look at leftmost-innermost rewriting as an example for a rewrite strategy that uses an ordering for parallel redexes. In contrast to the non-probabilistic setting, it turns out that $\mathbf{PSN}_{\mathbf{i}_{\rightarrow \mathcal{P}}}$ and $\mathbf{PSN}_{\mathbf{f}_{\rightarrow \mathcal{P}}}$ are not equivalent in general. The following counterexample is similar to Counterex. 4.4, which illustrated that $\mathbf{PSN}_{\mathbf{f}_{\rightarrow \mathcal{P}}}$ and $\mathbf{PSN}_{\mathbf{i}_{\rightarrow \mathcal{P}}}$ are not equivalent without left-linearity.

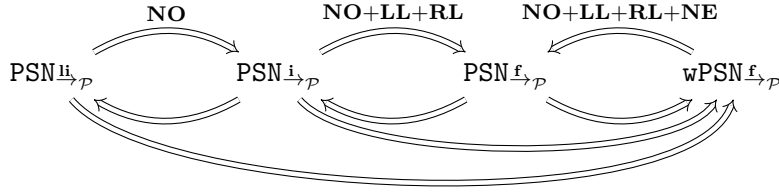


Figure 3: Relations between the different termination properties for PTRSs

Counterexample 4.10. Consider the PTRS \mathcal{P}_5 with the five rules:

$$\begin{array}{ll}
 a \rightarrow \{1 : c_1\} & b \rightarrow \{1/2 : d_1, 1/2 : d_2\} \\
 a \rightarrow \{1 : c_2\} & f(c_1, d_1) \rightarrow \{1 : f(a, b)\} \\
 & f(c_2, d_2) \rightarrow \{1 : f(a, b)\}
 \end{array}$$

We do not have $\text{AST}_{\vec{i} \rightarrow \mathcal{P}_5}$ (hence also neither $\text{PAST}_{\vec{i} \rightarrow \mathcal{P}_5}$ nor $\text{SAST}_{\vec{i} \rightarrow \mathcal{P}_5}$), since there exists the infinite rewrite sequence $\{1 : f(a, b)\} \xrightarrow{\vec{i}}_{\mathcal{P}_5} \{1/2 : f(a, d_1), 1/2 : f(a, d_2)\} \xrightarrow{\vec{i}}_{\mathcal{P}_5} \{1/2 : f(c_1, d_1), 1/2 : f(c_2, d_2)\} \xrightarrow{\vec{i}}_{\mathcal{P}_5} \{1/2 : f(a, b), 1/2 : f(a, b)\} \xrightarrow{\vec{i}}_{\mathcal{P}_5} \dots$, which converges with probability 0. It first “splits” the term $f(a, b)$ with the b -rule, and then applies one of the two different a -rules to each of the resulting terms. In contrast, when applying a leftmost-innermost rewrite strategy, we have to decide which a -rule to use before we split the term with the b -rule. For example, we have $\{1 : f(a, b)\} \xrightarrow{\vec{i}}_{\mathcal{P}_5} \{1 : f(c_1, b)\} \xrightarrow{\vec{i}}_{\mathcal{P}_5} \{1/2 : f(c_1, d_1), 1/2 : f(c_1, d_2)\}$. Here, the second term $f(c_1, d_2)$ is a normal form. Since all leftmost-innermost rewrite sequences keep on adding such normal forms after a certain number of steps for each start term, we have $\text{SAST}_{\vec{i} \rightarrow \mathcal{P}_5}$ (and hence, $\text{PAST}_{\vec{i} \rightarrow \mathcal{P}_5}$ and $\text{AST}_{\vec{i} \rightarrow \mathcal{P}_5}$).

The counterexample above can easily be adapted to variants of innermost rewriting that impose different orders on parallel redexes like, e.g., *rightmost*-innermost rewriting.

However, $\text{PSN}_{\vec{i} \rightarrow \mathcal{P}}$ and $\text{PSN}_{\vec{i} \rightarrow \mathcal{P}}$ are again equivalent for non-overlapping PTRSs \mathcal{P} . For such PTRSs, at most one rule can be used to rewrite at a given position, which prevents the problem illustrated in Counterex. 4.10.

Theorem 4.11 (From $\text{PSN}_{\vec{i} \rightarrow \mathcal{P}}$ to $\text{PSN}_{\vec{i} \rightarrow \mathcal{P}}$). *If a PTRS \mathcal{P} is NO, then*

$$\text{PSN}_{\vec{i} \rightarrow \mathcal{P}} \iff \text{PSN}_{\vec{i} \rightarrow \mathcal{P}}$$

For the proof of Thm. 4.11, we use the following lemma which immediately implies Thm. 4.11.

Lemma 4.12 (From Leftmost-Innermost to Innermost Rewriting). *If a PTRS \mathcal{P} is NO, then for every infinite $\vec{i} \rightarrow_{\mathcal{P}}$ -rewrite sequence $\vec{\mu} = (\mu_n)_{n \in \mathbb{N}}$ there exists an infinite $\vec{i} \rightarrow_{\mathcal{P}}$ -rewrite sequence $\vec{\nu} = (\nu_n)_{n \in \mathbb{N}}$ such that*

- (i) $\lim_{n \rightarrow \infty} |\mu_n|_{\mathcal{P}} \geq \lim_{n \rightarrow \infty} |\nu_n|_{\mathcal{P}}$
- (ii) $\text{edl}(\vec{\mu}) \leq \text{edl}(\vec{\nu})$

The relations between the different notions of AST , PAST , and SAST of PTRSs for different rewrite strategies (given in Thm. 4.2, 4.9, and 4.11) are summarized in Fig. 3.

For expected complexity, we obtain the following result from Lemma 4.12.

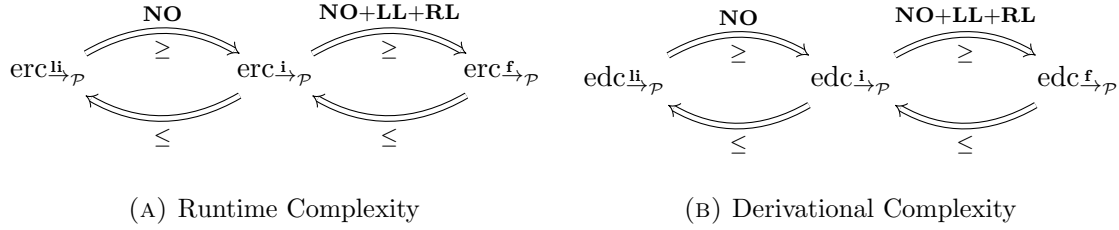


Figure 4: Relations for expected complexity

Theorem 4.13 (From Leftmost-Innermost to Innermost Expected Complexity). *If a PTRS \mathcal{P} is NO, then:*

$$\text{edc}_{\text{li}_\mathcal{P}} = \text{edc}_{\text{i}_\mathcal{P}} \quad \text{and} \quad \text{erc}_{\text{li}_\mathcal{P}} = \text{erc}_{\text{i}_\mathcal{P}}$$

As shown by \mathcal{P}_5 from Counterex. 4.10, again we cannot change the requirement of non-overlapping PTRSs to overlay systems, since \mathcal{P}_5 is an overlay system where $\text{PSN}_{\text{li}_\mathcal{P}_5}$ holds, but $\text{PSN}_{\text{i}_\mathcal{P}_5}$ does not. The relations between the different expected complexities of PTRSs (given in Thm. 4.6 and 4.13) are summarized in Fig. 4. Here, arrows “ \implies ” stand for “ \geq ”.

For the non-probabilistic setting, Thm. 4.13 implies the following corollary.

Corollary 4.14 (From Leftmost-Innermost to Innermost Complexity). *If a TRS \mathcal{R} is NO, then:*

$$\text{dc}_{\text{li}_\mathcal{R}} = \text{dc}_{\text{i}_\mathcal{R}} \quad \text{and} \quad \text{rc}_{\text{li}_\mathcal{R}} = \text{rc}_{\text{i}_\mathcal{R}}$$

5. IMPROVING APPLICABILITY

In this section, we present several ideas to improve the applicability of Thm. 4.2 which relates $\text{PSN}_{\text{f}_\mathcal{P}}$ and $\text{PSN}_{\text{i}_\mathcal{P}}$ — the most interesting theorem in practice. Recall that according to Thm. 4.2, $\text{PSN}_{\text{i}_\mathcal{P}}$ implies $\text{PSN}_{\text{f}_\mathcal{P}}$ for PTRSs \mathcal{P} that are NO, LL, and RL. In Section 5.1, we first discuss attempts in order to remove the requirement of left-linearity by modifying the rewrite relation to *simultaneous rewriting*. Here, we correct an inaccuracy in our previous conference paper [KFG24], where we claimed that simultaneous rewriting allows us to remove the left-linearity requirement. However, we now give a counterexample which shows that this improvement does not work in general. Then in Section 5.2 we target right-linearity, and show that it can be weakened to sparseness if one only considers rewrite sequences that start with basic terms, as in the definition of runtime complexity.

5.1. Attempting to Remove Left-Linearity by Simultaneous Rewriting. To try to remove the left-linearity requirement, one might consider the simultaneous reduction of several copies of identical redexes. For a PTRS \mathcal{P} , this results in the notion of *simultaneous rewriting*, denoted $\succ_{\text{f}_\mathcal{P}}$. The following example illustrates the idea of handling non-left-linear PTRSs by applying the same rewrite rule at parallel positions simultaneously.

Example 5.1 (Simultaneous Rewriting). Reconsider the PTRS \mathcal{P}_2 from Counterex. 4.4 with the rules $\text{f}(x, x) \rightarrow \{1 : \text{f}(\text{a}, \text{a})\}$ and $\text{a} \rightarrow \{1/2 : \text{b}, 1/2 : \text{c}\}$, where we have $\text{PSN}_{\text{i}_\mathcal{P}_2}$, but

not $\text{PSN}_{\rightarrow_{\mathcal{P}_2}}^f$. Our new rewrite relation $\rightarrow_{\mathcal{P}_2}$ allows us to reduce several copies of the same redex simultaneously, so that we get

$$\{1 : f(a, a)\} \xrightarrow{\frac{1}{2}}_{\mathcal{P}_2} \{\frac{1}{2} : f(b, b), \frac{1}{2} : f(c, c)\} \xrightarrow{\frac{1}{2}}_{\mathcal{P}_2} \{1/2 : f(a, a), 1/2 : f(a, a)\} \xrightarrow{\frac{1}{2}}_{\mathcal{P}_2} \dots,$$

i.e., this $\xrightarrow{\frac{1}{2}}_{\mathcal{P}_2}$ -rewrite sequence converges with probability 0 and thus, we do *not* have $\text{AST}_{\rightarrow_{\mathcal{P}_2}}^i$ (hence, also neither $\text{PAST}_{\rightarrow_{\mathcal{P}_2}}^i$ nor $\text{SAST}_{\rightarrow_{\mathcal{P}_2}}^i$). Note that we simultaneously reduced both occurrences of a in the first step.

Definition 5.2 (Simultaneous Rewriting). Let \mathcal{P} be a PTRS. A term s rewrites *simultaneously* to a multi-distribution $\mu = \{p_1 : t_1, \dots, p_k : t_k\}$ (denoted $s \xrightarrow{f}_{\mathcal{P}} \mu$) if there is a non-empty set of parallel positions $\Pi = \{\pi_1, \dots, \pi_m\} \subseteq \text{Pos}(s)$, a rule $\ell \rightarrow \{p_1 : r_1, \dots, p_k : r_k\} \in \mathcal{P}$, and a substitution σ such that $s|_{\pi_i} = \ell\sigma$ for all $1 \leq i \leq m$ and $t_j = s[r_j\sigma]_{\pi_1} \dots [r_j\sigma]_{\pi_m}$ for all $1 \leq j \leq k$. The simultaneous rewrite step is *innermost* (denoted $s \xrightarrow{i}_{\mathcal{P}} \mu$) if all proper subterms of the redex $\ell\sigma$ are in normal form w.r.t. \mathcal{P} .

Clearly, if the set of positions Π in Def. 5.2 is a singleton, then the resulting simultaneous rewrite step is an “ordinary” probabilistic rewrite step, i.e., $\xrightarrow{f}_{\mathcal{P}} \subseteq \xrightarrow{f}_{\mathcal{P}}$ and $\xrightarrow{i}_{\mathcal{P}} \subseteq \xrightarrow{i}_{\mathcal{P}}$.

Corollary 5.3 (From $\rightarrow_{\mathcal{P}}$ to $\rightarrow_{\mathcal{P}}$). *Let \mathcal{P} be a PTRS. Then, $\text{PSN}_{\rightarrow_{\mathcal{P}}}^s$ implies $\text{PSN}_{\rightarrow_{\mathcal{P}}}^s$ for $s \in \{\mathbf{f}, \mathbf{i}\}$.*

However, the converse of Cor. 5.3 does not hold. Ex. 5.1 shows that $\text{PSN}_{\rightarrow_{\mathcal{P}}}^i$ does not imply $\text{PSN}_{\rightarrow_{\mathcal{P}}}^i$, and the following example shows the same for $\text{PSN}_{\rightarrow_{\mathcal{P}}}^f$ and $\text{PSN}_{\rightarrow_{\mathcal{P}}}^f$.

Example 5.4. Consider the PTRS $\overline{\mathcal{P}}_2$ with the three rules:

$$\begin{aligned} f(b, b) &\rightarrow \{1 : f(a, a)\} & a &\rightarrow \{1/2 : b, 1/2 : c\} \\ f(c, c) &\rightarrow \{1 : f(a, a)\} \end{aligned}$$

We have $\text{SAST}_{\rightarrow_{\overline{\mathcal{P}}_2}}^f$ (hence, also $\text{PAST}_{\rightarrow_{\overline{\mathcal{P}}_2}}^f$ and $\text{AST}_{\rightarrow_{\overline{\mathcal{P}}_2}}^f$). But as in Ex. 5.1, we obtain $\{1 : f(a, a)\} \xrightarrow{\frac{1}{2}}_{\overline{\mathcal{P}}_2} \{\frac{1}{2} : f(b, b), \frac{1}{2} : f(c, c)\} \xrightarrow{\frac{1}{2}}_{\overline{\mathcal{P}}_2} \{1/2 : f(a, a), 1/2 : f(a, a)\}$, i.e., there are rewrite sequences with $\xrightarrow{\frac{1}{2}}_{\overline{\mathcal{P}}_2}$, and thus, also with $\xrightarrow{f}_{\overline{\mathcal{P}}_2}$, that converge with probability 0. Hence, $\text{AST}_{\rightarrow_{\overline{\mathcal{P}}_2}}^i$ does not hold, and therefore, $\text{AST}_{\rightarrow_{\overline{\mathcal{P}}_2}}^f$, $\text{PAST}_{\rightarrow_{\overline{\mathcal{P}}_2}}^f$, and $\text{SAST}_{\rightarrow_{\overline{\mathcal{P}}_2}}^f$ do not hold either.

Note that this kind of simultaneous rewriting is different from the “ordinary” parallelism used for non-probabilistic rewriting, which is typically denoted by $\xrightarrow{f}_{\parallel}$. There, one may reduce multiple parallel redexes in a single rewrite step. To be precise, a term s rewrites *parallel* to a multi-distribution μ (denoted $s \xrightarrow{f}_{\parallel} \mu$) w.r.t. a PTRS \mathcal{P} if there is a non-empty set of parallel redex positions $\Pi = \{\pi_1, \dots, \pi_m\} \subseteq \text{Pos}(s)$, such that $\{1 : s\} \xrightarrow{f}_{\mathcal{P}, \pi_1} \dots \xrightarrow{f}_{\mathcal{P}, \pi_m} \mu$. Here, $\xrightarrow{f}_{\mathcal{P}, \pi}$ denotes the relation $\xrightarrow{f}_{\mathcal{P}}$, where we rewrite at position π in each term in the multi-distribution. This is always possible, since the positions are parallel positions of the redexes in the start term s .

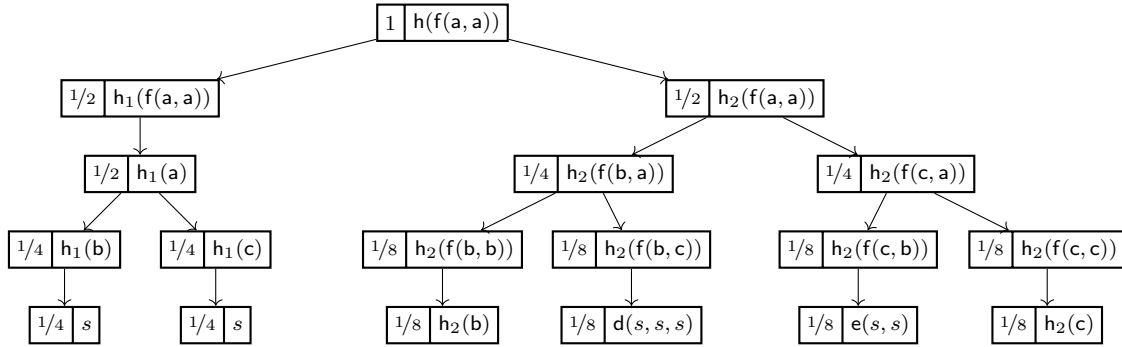
The two differences between simultaneous rewriting and parallel rewriting are that while both of them allow the reduction of multiple redexes, simultaneous rewriting “merges” the corresponding terms in the multi-distributions that result from rewriting the several redexes. Because of this merging, we only allow the simultaneous reduction of *equal* redexes, whereas “ordinary” parallel rewriting allows the simultaneous reduction of arbitrary parallel redexes, which is the second difference. For example, for \mathcal{P}_2 from Counterex. 4.4 we have $\{1 : f(a, a)\} \xrightarrow{\frac{1}{2}}_{\mathcal{P}_2} \{\frac{1}{2} : f(b, b), \frac{1}{2} : f(c, c)\}$, whereas with ordinary parallel rewriting we would obtain $\{1 : f(a, a)\} \xrightarrow{\frac{1}{4}}_{\parallel \mathcal{P}_2} \{\frac{1}{4} : f(b, b), \frac{1}{4} : f(b, c), \frac{1}{4} : f(c, b), \frac{1}{4} : f(c, c)\}$.

Even though simultaneous rewriting solves the issue of Counterex. 4.4, the following (more involved) counterexample shows that even with this refined rewrite relation, we cannot remove left-linearity from Thm. 4.2. In other words, there exist PTRSs \mathcal{P} that are non-overlapping and right-linear, but where $\text{PSN}_{\rightarrow_{\mathcal{P}}}$ does not imply $\text{PSN}_{\rightarrow_{\mathcal{P}}}$.

Counterexample 5.5. Consider the PTRS \mathcal{P}_6 with the following seven rules, where $s = h(f(a, a))$:

$$\begin{aligned} h(x) &\rightarrow \{1/2 : h_1(x), 1/2 : h_2(x)\} & h_1(b) &\rightarrow \{1 : s\} \\ a &\rightarrow \{1/2 : b, 1/2 : c\} & h_1(c) &\rightarrow \{1 : s\} \\ f(x, x) &\rightarrow \{1 : x\} & h_2(f(b, c)) &\rightarrow \{1 : d(s, s, s)\} \\ & & h_2(f(c, b)) &\rightarrow \{1 : e(s, s)\} \end{aligned}$$

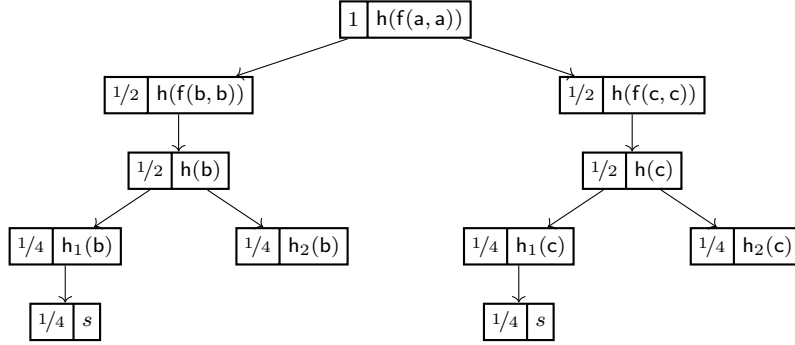
Let us take a closer look at the following $\rightarrow_{\mathcal{P}_6}$ -RST that starts with the term s .



We do not have $\text{AST}_{\rightarrow_{\mathcal{P}_6}}$ (hence, also neither $\text{PAST}_{\rightarrow_{\mathcal{P}_6}}$ nor $\text{SAST}_{\rightarrow_{\mathcal{P}_6}}$). Starting with s , we can either obtain s again with a chance of $1/2$, end in a normal form with a chance of $1/4$, rewrite to two occurrences of s with a chance of $1/8$, or rewrite to three occurrences of s with a chance of $1/8$. Overall, this results in a random walk biased towards non-termination, that is not AST (i.e., this RST can be extended to an RST whose convergence probability is < 1).

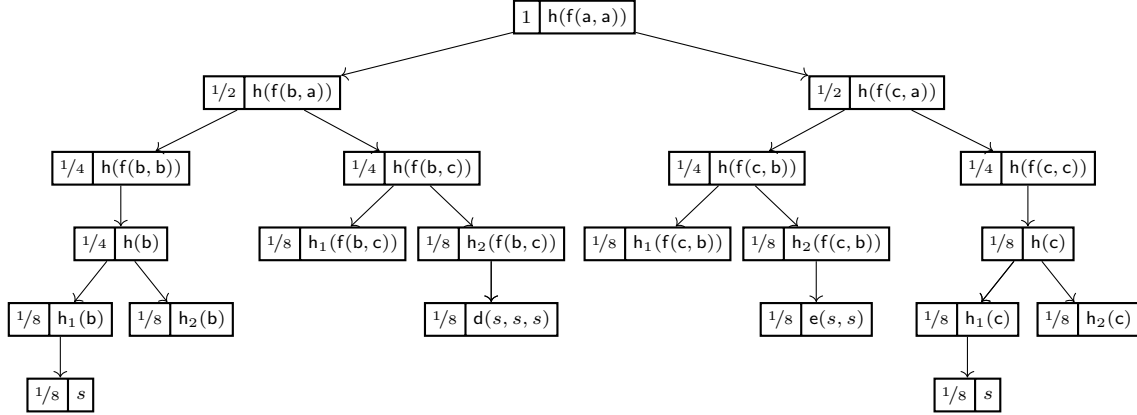
The crucial observation is that the order of rule application differs between the subtrees: In the left subtree starting at $h_1(f(a, a))$, we want to use the non-left-linear f -rule first. In the right subtree starting at $h_2(f(a, a))$, we want to use the a -rules first to create the terms $f(b, c)$ and $f(c, b)$. So essentially, we have to decide on the rewrite strategy after rewriting the top h -symbol. This is impossible in innermost evaluations and cannot be solved by simultaneous rewriting either. For innermost simultaneous rewriting, we would have to decide on whether we want to rewrite both occurrences of a in the first step or just one. In both cases, we obtain an RST with finite expected derivation length. Hence, we have $\text{SAST}_{\rightarrow_{\mathcal{P}_6}}$ (which implies $\text{PAST}_{\rightarrow_{\mathcal{P}_6}}$ and $\text{AST}_{\rightarrow_{\mathcal{P}_6}}$).

More precisely, if we decide to rewrite both a -symbols simultaneously at the start, we obtain the following $\rightarrow_{\mathcal{P}_6}$ -RST.



Overall, this results in a random walk biased towards termination, because we reach a normal form with a chance of $1/2$ and the term s with a chance of $1/2$, but we cannot increase the number of occurrences of s .

If we rewrite both a -symbols individually, then we obtain the following $\rightarrow_{\mathcal{P}_6}^i$ -RST.



Again, this results in a random walk biased towards termination, because starting with s , we can either obtain s again with a chance of $1/4$, end in a normal form with a chance of $1/2$, rewrite to two occurrences of s with a chance of $1/8$, or rewrite to three occurrences of s with a chance of $1/8$. So every $\rightarrow_{\mathcal{P}_6}^i$ -RST has a finite expected derivation length and we obtain $\text{SAST}_{\rightarrow_{\mathcal{P}_6}^i}$ (which implies $\text{PAST}_{\rightarrow_{\mathcal{P}_6}^i}$ and $\text{AST}_{\rightarrow_{\mathcal{P}_6}^i}$).

So this counterexample refutes [KFG24, Thm. 27] where it was claimed that $\text{PSN}_{\rightarrow_{\mathcal{P}}^i}$ implies $\text{PSN}_{\rightarrow_{\mathcal{P}}}$ for $\text{PSN} \in \{\text{AST}, \text{PAST}\}$. The problem is that the proof of [KFG24, Thm. 27] was limited to terms with at most two nested defined symbols. For example, every term in the rewrite sequence of Counterex. 4.4 has at most two nested defined symbols, e.g., in the term $f(a, a)$ the defined symbol f is above the defined symbol a , but there is no further defined symbol above f . However, Counterex. 5.5 uses a term $h(f(a, a))$ with three nested defined symbols. As mentioned in Counterex. 5.5, we have to decide on the rewrite strategy after rewriting the symbol h at the root position, and cannot fix an innermost evaluation strategy. Due to the non-left-linear f -rule, depending on whether one first rewrites f or a , one obtains completely different distributions even though the PTRS is non-overlapping.

Note that for wPSN , the direction of the implication in Cor. 5.3 is reversed, since wPSN requires that for each start term, there *exists* an infinite rewrite sequence satisfying a certain property, whereas PSN requires that *all* infinite rewrite sequences satisfy a certain property. Thus, if there exists an infinite $\Rightarrow_{\mathcal{P}}$ -rewrite sequence that, e.g., converges with probability 1

(showing that $\mathbf{wAST}_{\mathbf{f} \rightarrow \mathcal{P}}$ holds), then this is also a valid $\Rightarrow_{\mathcal{P}}$ -rewrite sequence that converges with probability 1 (showing that $\mathbf{wAST}_{\mathbf{f} \rightarrow \mathcal{P}}$ holds).

Corollary 5.6 (From $\mathbf{wPSN}_{\mathbf{f} \rightarrow \mathcal{P}}$ to $\mathbf{wPSN}_{\mathbf{f} \rightarrow \mathcal{P}}$). *Let \mathcal{P} be a PTRS. Then, $\mathbf{wPSN}_{\mathbf{f} \rightarrow \mathcal{P}}$ implies $\mathbf{wPSN}_{\mathbf{f} \rightarrow \mathcal{P}}$.*

For \mathbf{wPSN} , Counterex. 5.5 shows that simultaneous rewriting cannot be used to remove the requirement of left-linearity from Thm. 4.9 either. The following Counterex. 5.7 shows that we can even give a counterexample that only considers terms where the number of nested defined symbols is at most 2.

Counterexample 5.7. Consider the non-left-linear PTRS \mathcal{P}_7 with the two rules:

$$\mathbf{g} \rightarrow \{3/4 : \mathbf{d}(\mathbf{g}, \mathbf{g}), 1/4 : 0\} \qquad \mathbf{d}(x, x) \rightarrow \{1 : x\}$$

We do not have $\mathbf{AST}_{\mathbf{f} \rightarrow \mathcal{P}_7}$ (hence, not $\mathbf{PAST}_{\mathbf{f} \rightarrow \mathcal{P}_7}$ either), as we have $\{1 : \mathbf{g}\} \xrightarrow{\mathbf{f}}_{\mathcal{P}_7} \{3/4 : \mathbf{d}(\mathbf{g}, \mathbf{g}), 1/4 : 0\}$, which corresponds to a random walk biased towards non-termination if we never use the \mathbf{d} -rule (since $\frac{3}{4} > \frac{1}{4}$). However, if we always use the \mathbf{d} -rule directly after the \mathbf{g} -rule, then we essentially end up with a PTRS whose only rule is $\mathbf{g} \rightarrow \{3/4 : \mathbf{g}, 1/4 : 0\}$, which corresponds to flipping a biased coin until heads comes up. This proves $\mathbf{wPAST}_{\mathbf{f} \rightarrow \mathcal{P}_7}$ and hence, also $\mathbf{wAST}_{\mathbf{f} \rightarrow \mathcal{P}_7}$. As \mathcal{P}_7 is non-overlapping, right-linear, and non-erasing, this shows that a variant of Thm. 4.9 without the requirement of left-linearity would need more than just moving to simultaneous rewriting.

5.2. Weakening Right-Linearity to Spareness. In the non-probabilistic setting, runtime complexity is easier to analyze than derivational complexity because of the restriction to basic start terms. In particular, this restriction also allows us to use notions like spareness such that full runtime complexity can be analyzed via innermost runtime complexity, see Thm. 2.9. Similarly, in the probabilistic setting we can also require spareness instead of right-linearity, if we only consider basic start terms. To adapt spareness to PTRSs \mathcal{P} , a rewrite step using the rule $\ell \rightarrow \mu \in \mathcal{P}$ and the substitution σ is called *spare* if $\sigma(x) \in \mathbf{NF}_{\mathcal{P}}$ for every $x \in \mathcal{V}$ that occurs more than once in some $r \in \text{Supp}(\mu)$. A $\xrightarrow{\mathbf{f}}_{\mathcal{P}}$ -rewrite sequence is spare if each of its $\xrightarrow{\mathbf{f}}_{\mathcal{P}}$ -steps is spare. \mathcal{P} is spare if each $\xrightarrow{\mathbf{f}}_{\mathcal{P}}$ -rewrite sequence that starts with $\{1 : t\}$ for a basic term $t \in \mathcal{T}_{\mathcal{B}}$ is spare.

Example 5.8. Consider the PTRS \mathcal{P}_8 with the two rules:

$$\mathbf{g} \rightarrow \{3/4 : \mathbf{d}(0), 1/4 : \mathbf{g}\} \qquad \mathbf{d}(x) \rightarrow \{1 : \mathbf{c}(x, x)\}$$

It is similar to the PTRS \mathcal{P}_1 from Counterex. 4.1, but we exchanged the symbols \mathbf{g} and 0 in the right-hand side of the \mathbf{g} -rule. This PTRS is orthogonal but duplicating due to the \mathbf{d} -rule. However, in any rewrite sequence that starts with $\{1 : t\}$ for a basic term t we can only duplicate constructor symbols but no terms with defined symbols. Hence, \mathcal{P}_8 is spare.

If a PTRS \mathcal{P} is spare, and we start with a basic term, then we will only duplicate normal forms with our duplicating rules. This means that the duplicating rules do not influence the (expected) runtime and, more importantly for \mathbf{AST} , the probability of convergence. This leads to the following theorem, which weakens the requirement of RL to SP in Thm. 4.2, where “*starting in $\mathcal{T}_{\mathcal{B}}$* ” means that one only considers rewrite sequences that start with $\{1 : t\}$ for a term $t \in \mathcal{T}_{\mathcal{B}}$, where $\mathcal{T}_{\mathcal{B}}$ is again the set of all basic terms w.r.t. \mathcal{P} .

Theorem 5.9 (From $\text{PSN}_{\dot{\rightarrow}_{\mathcal{P}}}$ Starting in $\mathcal{T}_{\mathcal{B}}$ to $\text{PSN}_{\dot{\rightarrow}_{\mathcal{P}}}$ Starting in $\mathcal{T}_{\mathcal{B}}$). *If a PTRS \mathcal{P} is OR and SP, then:*

$$\text{PSN}_{\dot{\rightarrow}_{\mathcal{P}}} \text{ starting in } \mathcal{T}_{\mathcal{B}} \iff \text{PSN}_{\dot{\rightarrow}_{\mathcal{P}}} \text{ starting in } \mathcal{T}_{\mathcal{B}}$$

For the proof of Thm. 5.9 we use the following lemma.

Lemma 5.10 (From Innermost to Full Rewriting Starting in $\mathcal{T}_{\mathcal{B}}$). *If a PTRS \mathcal{P} is OR and SP, then for every infinite $\dot{\rightarrow}_{\mathcal{P}}$ -rewrite sequence $\vec{\mu} = (\mu_n)_{n \in \mathbb{N}}$ that starts with a basic term there exists an infinite $\dot{\rightarrow}_{\mathcal{P}}$ -rewrite sequence $\vec{\nu} = (\nu_n)_{n \in \mathbb{N}}$ that starts with a basic term such that*

- (i) $\lim_{n \rightarrow \infty} |\mu_n|_{\mathcal{P}} \geq \lim_{n \rightarrow \infty} |\nu_n|_{\mathcal{P}}$
- (ii) $\text{edl}(\vec{\mu}) \leq \text{edl}(\vec{\nu})$

The requirement of basic start terms is a real restriction for orthogonal PTRSs, i.e., there exist very simple orthogonal PTRSs \mathcal{P} , where $\text{PSN}_{\dot{\rightarrow}_{\mathcal{P}}}$ starting in $\mathcal{T}_{\mathcal{B}}$ holds, but not $\text{PSN}_{\dot{\rightarrow}_{\mathcal{P}}}$ in general. One can see this already in the non-probabilistic setting for the TRS \mathcal{R} consisting of the two rules $\mathbf{f}(\mathbf{g}(\mathbf{a})) \rightarrow \mathbf{f}(\mathbf{g}(\mathbf{a}))$ and $\mathbf{g}(\mathbf{b}) \rightarrow \mathbf{b}$, where we do not have $\text{SN}_{\dot{\rightarrow}_{\mathcal{R}}}$, but $\text{SN}_{\dot{\rightarrow}_{\mathcal{R}}}$ starting in $\mathcal{T}_{\mathcal{B}}$ for every $s \in \mathbb{S}$.

The following example shows that Thm. 5.9 does not hold for arbitrary start terms.

Counterexample 5.11. Consider the PTRS \mathcal{P}_9 with the two rules:

$$\mathbf{g} \rightarrow \{3/4 : \mathbf{s}(\mathbf{g}), 1/4 : \mathbf{0}\} \qquad \mathbf{f}(\mathbf{s}(x)) \rightarrow \{1 : \mathbf{c}(\mathbf{f}(x), \mathbf{f}(x))\}$$

This PTRS behaves similarly to \mathcal{P}_1 (see Counterex. 4.1). We do not have $\text{AST}_{\dot{\rightarrow}_{\mathcal{P}_9}}$ (and hence, also neither $\text{PAST}_{\dot{\rightarrow}_{\mathcal{P}_9}}$ nor $\text{SAST}_{\dot{\rightarrow}_{\mathcal{P}_9}}$), as we have $\{1 : \mathbf{f}(\mathbf{g})\} \rightrightarrows_{\mathcal{P}_9}^2 \{3/4 : \mathbf{c}(\mathbf{f}(\mathbf{g}), \mathbf{f}(\mathbf{g})), 1/4 : \mathbf{f}(\mathbf{0})\}$, which corresponds to a random walk biased towards non-termination (since $\frac{3}{4} > \frac{1}{4}$).

However, the only basic terms for this PTRS are \mathbf{g} and $\mathbf{f}(t)$ for terms t that do not contain \mathbf{g} or \mathbf{f} . A sequence starting with \mathbf{g} corresponds to flipping a biased coin and a sequence starting with $\mathbf{f}(t)$ will clearly terminate. Hence, we have $\text{SAST}_{\dot{\rightarrow}_{\mathcal{P}_9}}$ (and thus, also $\text{PAST}_{\dot{\rightarrow}_{\mathcal{P}_9}}$ and $\text{AST}_{\dot{\rightarrow}_{\mathcal{P}_9}}$) starting in $\mathcal{T}_{\mathcal{B}}$. Furthermore, note that we have $\text{SAST}_{\dot{\rightarrow}_{\mathcal{P}_9}}$ (and thus, also $\text{PAST}_{\dot{\rightarrow}_{\mathcal{P}_9}}$ and $\text{AST}_{\dot{\rightarrow}_{\mathcal{P}_9}}$) for arbitrary start terms, analogous to \mathcal{P}_1 . Since \mathcal{P}_9 is OR and SP, this shows that Thm. 5.9 cannot be extended to $\text{PSN}_{\dot{\rightarrow}_{\mathcal{P}}}$ in general.

For expected complexity, we obtain a result that is analogous to Thm. 5.9. Since we require basic start terms, the following theorem only holds for expected runtime complexity and not for expected derivation complexity.

Theorem 5.12 (From Innermost to Full Expected Runtime Complexity). *If a PTRS \mathcal{P} is OR and SP, then:*

$$\text{erc}_{\dot{\rightarrow}_{\mathcal{P}}} = \text{erc}_{\dot{\rightarrow}_{\mathcal{P}}}$$

Thus, this theorem weakens the requirement of RL in Thm. 4.6 to SP (recall that right-linearity implies sparseness). A corresponding corollary for the non-probabilistic setting would be subsumed by Thm. 2.9, which ensures $\text{rc}_{\dot{\rightarrow}_{\mathcal{R}}} = \text{rc}_{\dot{\rightarrow}_{\mathcal{R}}}$ for all spare overlay systems \mathcal{R} . As shown by Ex. 4.7, OS is not enough in the probabilistic setting, but one needs OR.

One may wonder whether Thm. 5.9 can nevertheless be used in order to prove $\text{PSN}_{\dot{\rightarrow}_{\mathcal{P}}}$ for a PTRS \mathcal{P} on *all* terms (instead of just basic start terms) by using a suitable transformation from \mathcal{P} to another PTRS \mathcal{P}' such that $\text{PSN}_{\dot{\rightarrow}_{\mathcal{P}}}$ holds for all terms iff $\text{PSN}_{\dot{\rightarrow}_{\mathcal{P}'}}$ holds when starting with basic terms. In [Fuh19], a transformation was presented that extends any

(non-probabilistic) TRS \mathcal{R} by generator rules $\mathcal{G}(\mathcal{R})$ such that the derivational complexity of \mathcal{R} is the same as the runtime complexity of $\mathcal{R} \cup \mathcal{G}(\mathcal{R})$, where $\mathcal{G}(\mathcal{R})$ are considered to be *relative* rules whose rewrite steps do not “count” for the complexity. This transformation can indeed be reused to move from $\text{PSN}_{\mathcal{F}_{\mathcal{P}}}$ starting in $\mathcal{T}_{\mathcal{B}}$ to $\text{PSN}_{\mathcal{F}_{\mathcal{P}}}$ on arbitrary terms.

The idea of the transformation is to introduce a new constructor symbol cons_f for every defined symbol $f \in \Sigma_D$, and to introduce a new defined symbol enc_f for every function symbol $f \in \Sigma$. As an example for \mathcal{P}_8 from Ex. 5.8, then instead of starting with the non-basic term $c(g, f(g))$, we start with the basic term $\text{enc}_c(\text{cons}_g, \text{cons}_f(\text{cons}_g))$, its *basic variant*. The new defined symbol enc_c is used to first build the term $c(g, f(g))$ at the beginning of the rewrite sequence, i.e., it converts all occurrences of cons_f for $f \in \Sigma_D$ back into the defined symbol f , and then we can proceed as if we started with the term $c(g, f(g))$ directly. For this conversion, we need another new defined symbol argenc that iterates through the term and replaces all new constructors cons_f by the original defined symbol f . Thus, we define the generator rules as in [Fuh19] (just with trivial probabilities in the right-hand sides $\ell \rightarrow \{1 : r\}$), since we do not need any probabilities during this initial construction of the original start term.

Definition 5.13 (Generator Rules $\mathcal{G}(\mathcal{P})$). Let \mathcal{P} be a PTRS over the signature Σ . Its *generator rules* $\mathcal{G}(\mathcal{P})$ are the following set of rules

$$\begin{aligned} & \{\text{enc}_f(x_1, \dots, x_k) \rightarrow \{1 : f(\text{argenc}(x_1), \dots, \text{argenc}(x_k))\} \mid f \in \Sigma\} \\ & \cup \{\text{argenc}(\text{cons}_f(x_1, \dots, x_k)) \rightarrow \{1 : f(\text{argenc}(x_1), \dots, \text{argenc}(x_k))\} \mid f \in \Sigma_D\} \\ & \cup \{\text{argenc}(f(x_1, \dots, x_k)) \rightarrow \{1 : f(\text{argenc}(x_1), \dots, \text{argenc}(x_k))\} \mid f \in \Sigma_C\}, \end{aligned}$$

where x_1, \dots, x_k are pairwise different variables and where the function symbols argenc , cons_f , and enc_f are fresh (i.e., they do not occur in \mathcal{P}). Moreover, we define $\Sigma_{\mathcal{G}(\mathcal{P})} = \{\text{enc}_f \mid f \in \Sigma\} \cup \{\text{argenc}\} \cup \{\text{cons}_f \mid f \in \Sigma_D\}$.

Example 5.14. For the PTRS \mathcal{P}_9 from Counterex. 5.11, we obtain the following generator rules $\mathcal{G}(\mathcal{P}_9)$:

$$\begin{aligned} \text{enc}_g & \rightarrow \{1 : g\} \\ \text{enc}_f(x_1) & \rightarrow \{1 : f(\text{argenc}(x_1))\} \\ \text{enc}_c(x_1, x_2) & \rightarrow \{1 : c(\text{argenc}(x_1), \text{argenc}(x_2))\} \\ \text{enc}_s(x_1) & \rightarrow \{1 : s(\text{argenc}(x_1))\} \\ \text{enc}_0 & \rightarrow \{1 : 0\} \\ \text{argenc}(\text{cons}_g) & \rightarrow \{1 : g\} \\ \text{argenc}(\text{cons}_f(x_1)) & \rightarrow \{1 : f(\text{argenc}(x_1))\} \\ \text{argenc}(c(x_1, x_2)) & \rightarrow \{1 : c(\text{argenc}(x_1), \text{argenc}(x_2))\} \\ \text{argenc}(s(x_1)) & \rightarrow \{1 : s(\text{argenc}(x_1))\} \\ \text{argenc}(0) & \rightarrow \{1 : 0\} \end{aligned}$$

As mentioned, using the symbols cons_f and enc_f , as in [Fuh19] every term over Σ can be transformed into a basic term over $\Sigma \cup \Sigma_{\mathcal{G}(\mathcal{P})}$.

The following lemma shows that by adding the generator rules, one can indeed reduce the problem of proving AST on all terms to AST starting in $\mathcal{T}_{\mathcal{B}}$.

Lemma 5.15 (From all Terms to Basic Terms (1)). *For any PTRS \mathcal{P} we have:*

$$\text{AST}_{\mathcal{F}_{\mathcal{P}}} \iff \text{AST}_{\mathcal{F}_{\mathcal{P} \cup \mathcal{G}(\mathcal{P})}} \text{ starting in } \mathcal{T}_{\mathcal{B}}.$$

If one extends the definition of **PAST** by relative rules, then similar results should also be possible for $\mathbf{PAST}_{\rightarrow \mathcal{P}}$, $\mathbf{SAST}_{\rightarrow \mathcal{P}}$, and both expected derivational and runtime complexity. When considering the additional generator rules as ordinary instead of relative rewrite rules, we obtain an upper bound on the expected complexity and sufficient conditions for $\mathbf{PAST}_{\rightarrow \mathcal{P}}$ and $\mathbf{SAST}_{\rightarrow \mathcal{P}}$.

Lemma 5.16 (From all Terms to Basic Terms (2)). *For any PTRS \mathcal{P} we have:*

$$\begin{aligned} \mathbf{PAST}_{\rightarrow \mathcal{P}} &\Leftarrow \mathbf{PAST}_{\rightarrow \mathcal{P} \cup \mathcal{G}(\mathcal{P})} \text{ starting in } \mathcal{T}_{\mathcal{B}} \\ \mathbf{SAST}_{\rightarrow \mathcal{P}} &\Leftarrow \mathbf{SAST}_{\rightarrow \mathcal{P} \cup \mathcal{G}(\mathcal{P})} \text{ starting in } \mathcal{T}_{\mathcal{B}} \\ \text{edc}_{\rightarrow \mathcal{P}} &\leq \text{erc}_{\rightarrow \mathcal{P} \cup \mathcal{G}(\mathcal{P})} \end{aligned}$$

However, even if \mathcal{P} is spare, the PTRS $\mathcal{P} \cup \mathcal{G}(\mathcal{P})$ is not guaranteed to be spare, although the generator rules themselves are right-linear. The problem is that the generator rules include a rule like $\text{enc}_f(x_1) \rightarrow \{1 : f(\text{argenc}(x_1))\}$ where a defined symbol **argenc** occurs below the duplicating symbol **f** on the right-hand side. Indeed, while \mathcal{P}_9 is spare, $\mathcal{P}_9 \cup \mathcal{G}(\mathcal{P}_9)$ is not. For example, when starting with the basic term $\text{enc}_f(s(\text{cons}_g))$, we have

$$\begin{aligned} \{1 : \text{enc}_f(s(\text{cons}_g))\} &\xrightarrow{\mathcal{G}(\mathcal{P}_9)}^2 \{1 : f(s(\text{argenc}(\text{cons}_g)))\} \\ &\xrightarrow{\mathcal{P}_9} \{1 : c(f(\text{argenc}(\text{cons}_g)), f(\text{argenc}(\text{cons}_g)))\} \end{aligned}$$

where the last step is not spare. In general, $\mathcal{P} \cup \mathcal{G}(\mathcal{P})$ is guaranteed to be spare if \mathcal{P} is right-linear. So we could modify Thm. 5.9 into a theorem which states that $\mathbf{AST}_{\rightarrow \mathcal{P}}$ holds for all terms iff $\mathbf{AST}_{\rightarrow \mathcal{P} \cup \mathcal{G}(\mathcal{P})}$ holds when starting in $\mathcal{T}_{\mathcal{B}}$ (and thus, for all terms) for orthogonal and right-linear PTRSs \mathcal{P} . However, this theorem would be subsumed by Thm. 4.2, where we already showed the equivalence of $\mathbf{AST}_{\rightarrow \mathcal{P}}$ and $\mathbf{AST}_{\rightarrow \mathcal{P}}$ if \mathcal{P} is orthogonal and right-linear. Indeed, our goal in Thm. 5.9 was to find a weaker requirement than right-linearity. Hence, such a transformational approach to move from $\mathbf{AST}_{\rightarrow \mathcal{P}}$ on all start terms to $\mathbf{AST}_{\rightarrow \mathcal{P}}$ starting in $\mathcal{T}_{\mathcal{B}}$ does not seem viable for Thm. 5.9.

6. IMPLEMENTATION AND EVALUATION

We implemented our new criteria for the equivalence of $\mathbf{AST}_{\rightarrow \mathcal{P}}$ and $\mathbf{AST}_{\rightarrow \mathcal{P}}$ in our termination prover **AProVE** [GAB⁺17]. For every PTRS, one can indicate whether one wants to analyze its termination behavior for all start terms or only for basic start terms. **AProVE**'s main technique for termination analysis of PTRSs \mathcal{P} is the probabilistic DP framework from [KG23a, KDG24] to prove $\mathbf{AST}_{\rightarrow \mathcal{P}}$, and its adaption for $\mathbf{AST}_{\rightarrow \mathcal{P}}$ from [KG24], which is however strictly less powerful than the framework for $\mathbf{AST}_{\rightarrow \mathcal{P}}$. The general idea of the DP framework is a *divide-and-conquer* approach where *processors* are used to replace a termination problem by several new sub-problems that are easier to analyze than the original problem. These processors are applied repeatedly until all sub-problems are solved. Since different techniques can be used to analyze the different sub-problems, this results in a *modular* approach for almost-sure termination analysis.

For our evaluation, we developed a “new” version of **AProVE**, where we consider Thm. 4.2 and 5.9, and afterwards apply the DP framework as described above. More precisely, if one wants to analyze $\mathbf{AST}_{\rightarrow \mathcal{P}}$ for a PTRS \mathcal{P} , the “new” version of **AProVE** first tries to prove that the conditions of Thm. 4.2 are satisfied if one regards arbitrary start terms or that the conditions of Thm. 5.9 are satisfied if one only wants to consider basic start terms. If this succeeds, then it uses the probabilistic DP framework for $\mathbf{AST}_{\rightarrow \mathcal{P}}$, which then implies

	old	old + Thm. 4.2	new
Arbitrary Terms	52	57	57
Basic Terms	59	65	70

Table 1: Results

$\text{AST}_{\rightarrow\mathcal{P}}$.⁶ If these theorems cannot be applied, then AProVE tries to prove $\text{AST}_{\rightarrow\mathcal{P}}$ via the DP framework for $\text{AST}_{\rightarrow\mathcal{P}}$ [KG24], or using a direct application of polynomial orderings [KG23a]. So in contrast to the experiments in our conference paper [KFG24], where we only used the direct application of polynomial orderings to prove $\text{AST}_{\rightarrow\mathcal{P}}$, in this case, “new” now also uses the DP framework for $\text{AST}_{\rightarrow\mathcal{P}}$ from [KG24].

We compare “new” with the “old” version of AProVE that does not apply any of the theorems of the current paper. Thus, it directly uses the DP framework for $\text{AST}_{\rightarrow\mathcal{P}}$ from [KG24] or the direct application of polynomial orderings [KG23a].

Note that for $\text{AST}_{\rightarrow\mathcal{P}}$ w.r.t. basic start terms, Thm. 5.9 generalizes Thm. 4.2, since right-linearity implies sparseness. To show the advantage of Thm. 5.9 compared to Thm. 4.2, we also experimented with the variant “old + Thm. 4.2” where we always used Thm. 4.2 instead of Thm. 5.9, even when only proving $\text{AST}_{\rightarrow\mathcal{P}}$ for basic start terms.

We used the benchmark set of 130 PTRSs from [KG24], where AProVE can prove $\text{AST}_{\rightarrow\mathcal{P}}$ for 109 of them. Table 1 shows for how many of these 130 PTRSs the respective strategy allows AProVE to conclude $\text{AST}_{\rightarrow\mathcal{P}}$.

The increase in the number of solved examples from “old” to “new” shows that even with dedicated strategies and techniques like the DP framework for $\text{AST}_{\rightarrow\mathcal{P}}$ [KG24], it is still beneficial to use the techniques from the current paper to move from $\text{AST}_{\rightarrow\mathcal{P}}$ to $\text{AST}_{\rightarrow\mathcal{P}}$ whenever possible, since $\text{AST}_{\rightarrow\mathcal{P}}$ is significantly easier to prove than $\text{AST}_{\rightarrow\mathcal{P}}$. Moreover, the table shows that the weakening of right-linearity to sparseness (i.e., the generalization of Thm. 4.2 to Thm. 5.9) indeed increases power when considering basic start terms. As shown in the experiments of [KSG25], our Thm. 4.2 and 5.9 are also very useful in order to move from $\text{SAST}_{\rightarrow\mathcal{P}}$ vs. $\text{SAST}_{\rightarrow\mathcal{P}}$, i.e., they often allow us to use the DP framework for $\text{SAST}_{\rightarrow\mathcal{P}}$ and expected innermost complexity analysis from [KSG25] in order to prove $\text{SAST}_{\rightarrow\mathcal{P}}$ and to analyze expected runtime complexity for full rewriting.

For details on our experiments, our collection of examples, and for instructions on how to run our implementation in AProVE via its *web interface* or locally, we refer to:

<https://aprove-developers.github.io/InnermostToFullAST/>

7. MODULARITY

In this section, we investigate the modularity of probabilistic notions of termination. We will see that as in the non-probabilistic setting [Gra95], in contrast to innermost probabilistic rewriting, full probabilistic rewriting is not modular in general. However, our results on the relation between innermost and full probabilistic rewriting from the previous sections will allow us to obtain modularity results for full probabilistic rewriting as well. A property

⁶Currently, we only use the switch from full to innermost rewriting as a preprocessing step before applying the DP framework. As shown in [KG24], a corresponding modular processor *within* the DP framework (which requires the criteria of our theorems not for the whole PTRS, but just for specific sub-problems within the termination proof) would be unsound.

is called *modular* if it is preserved for certain unions of PTRSs. Modularity is not only interesting from a theoretical point of view, but in practice it is also very important to know whether one can split a huge PTRS into smaller parts such that the property of interest is preserved. Then, one can analyze these parts independently of each other. As remarked earlier, this is also the main idea of the DP framework, and in fact, we already benefit from similar modularity results within the DP framework itself. The fewer conditions are required for modularity of a notion of termination, the stronger the corresponding DP framework is expected to be.

We will study two different forms of unions, namely *disjoint unions* (Section 7.1), where both PTRSs do not share any function symbols, and *shared constructor unions* of PTRSs (Section 7.2), which may have common constructor symbols, but whose defined symbols are disjoint.⁷ In future work, one may extend this analysis to *hierarchical unions*, which are unions of PTRSs where the first PTRS may contain defined symbols of the second one, but not vice versa. Moreover, in this section we only consider the strategies $s \in \{\mathbf{f}, \mathbf{i}\}$ (but it is not difficult to show that our modularity results for innermost rewriting also hold for leftmost-innermost rewriting).

In Section 7.3, we will also investigate signature extensions. We have already seen in Thm. 3.16 that $\text{PAST}_{s \rightarrow \mathcal{P}}$ is not closed under signature extensions for any $s \in \mathbb{S}$. Based on our modularity results, we will now show that both $\text{AST}_{s \rightarrow \mathcal{P}}$ and $\text{SAST}_{s \rightarrow \mathcal{P}}$ are closed under signature extensions.

To distinguish the function symbols of different PTRSs \mathcal{P} , in the following we write $\Sigma_D^{\mathcal{P}}$, $\Sigma_C^{\mathcal{P}}$, and $\Sigma^{\mathcal{P}}$ for the defined symbols, constructor symbols, and all function symbols occurring in the rules of \mathcal{P} , respectively.

7.1. Disjoint Unions. We first consider unions of systems that do not share any function symbols, i.e., we consider two PTRSs $\mathcal{P}^{(1)}$ and $\mathcal{P}^{(2)}$ such that $\Sigma^{\mathcal{P}^{(1)}} \cap \Sigma^{\mathcal{P}^{(2)}} = \emptyset$. In the non-probabilistic setting, [Gra95] showed that innermost termination is modular for disjoint unions. This result can be lifted to $\text{AST}_{\mathbf{i} \rightarrow \mathcal{P}}$ and $\text{SAST}_{\mathbf{i} \rightarrow \mathcal{P}}$. We first investigate $\text{AST}_{\mathbf{i} \rightarrow \mathcal{P}}$, as illustrated by the following example.

Example 7.1. Consider the PTRS $\mathcal{P}_{10} = \mathcal{P}_{10}^{(1)} \cup \mathcal{P}_{10}^{(2)}$ given by

$$\mathcal{P}_{10}^{(1)} : f(x) \rightarrow \{1/2 : f(x), 1/2 : a\} \quad \mathcal{P}_{10}^{(2)} : g(x) \rightarrow \{1/2 : g(x), 1/2 : b\}$$

$\mathcal{P}_{10}^{(1)}$ and $\mathcal{P}_{10}^{(2)}$ both correspond to a fair coin flip, where one terminates when obtaining heads. Hence, for both systems we have $\text{AST}_{\mathbf{f} \rightarrow \mathcal{P}_{10}^{(1)}}$ and $\text{AST}_{\mathbf{f} \rightarrow \mathcal{P}_{10}^{(2)}}$, and thus also $\text{AST}_{\mathbf{i} \rightarrow \mathcal{P}_{10}^{(1)}}$

and $\text{AST}_{\mathbf{i} \rightarrow \mathcal{P}_{10}^{(2)}}$. Furthermore, $\Sigma^{\mathcal{P}_{10}^{(1)}} \cap \Sigma^{\mathcal{P}_{10}^{(2)}} = \emptyset$, i.e., \mathcal{P}_{10} is a disjoint union. When reducing a term like $f(g(x))$ which contains symbols from both systems, then we first reduce the innermost redex $g(x)$ until we reach a normal form. Due to the innermost strategy, we cannot rewrite at the position of f beforehand. This reduction only uses one of the two systems, namely $\mathcal{P}_{10}^{(2)}$, hence it converges with probability 1. Then, we use the next innermost redex, which will be $f(b)$, using only rules of $\mathcal{P}_{10}^{(1)}$ until we reach a normal form, where symbols from

⁷The dependency pair framework for $\text{AST}_{\mathbf{i} \rightarrow \mathcal{P}}$ in AProVE is already capable of splitting disjoint unions and shared constructor unions into sub-problems that can be analyzed independently. However, in this section our goal is to analyze the modularity of different probabilistic notions of termination in general. Thus, this allows the use of these modularity results also outside the DP framework.

$\Sigma^{\mathcal{P}_{10}^{(2)}}$ do not influence the reduction. The reason is that all subterms below or at positions of symbols from $\Sigma^{\mathcal{P}_{10}^{(2)}}$ are in normal form, and there is no symbol from $\Sigma^{\mathcal{P}_{10}^{(2)}}$ above the f at the root position. Again, this reduction converges with probability 1. Thus, in the end, our reduction starting with $f(g(x))$ also converges with probability 1, and the same holds for arbitrary start terms, which implies $\text{AST} \xrightarrow{f}_{\mathcal{P}_{10}}$.

In Ex. 7.1, we considered the term $f(g(x))$ where we swap once between a symbol f from $\Sigma^{\mathcal{P}_{10}^{(1)}}$ and a symbol g from $\Sigma^{\mathcal{P}_{10}^{(2)}}$ on the path from the root to the “leaf” of the term. In the proof of Thm. 7.2 we lift the argumentation of Ex. 7.1 to arbitrary terms via induction. This proof idea was also used by [Gra95] in the non-probabilistic setting to show the modularity of innermost termination for disjoint unions.

Theorem 7.2 (Modularity of $\text{AST} \xrightarrow{i}_{\mathcal{P}}$ for Disjoint Unions). *Let $\mathcal{P}^{(1)}$ and $\mathcal{P}^{(2)}$ be PTRSs with $\Sigma^{\mathcal{P}^{(1)}} \cap \Sigma^{\mathcal{P}^{(2)}} = \emptyset$. Then we have:*

$$\text{AST} \xrightarrow{i}_{\mathcal{P}^{(1)} \cup \mathcal{P}^{(2)}} \iff \text{AST} \xrightarrow{i}_{\mathcal{P}^{(1)}} \text{ and } \text{AST} \xrightarrow{i}_{\mathcal{P}^{(2)}}$$

Proof Sketch. The direction “ \implies ” is trivial and thus, we only prove “ \impliedby ”. Assume that we have $\text{AST} \xrightarrow{i}_{\mathcal{P}^{(1)}}$ and $\text{AST} \xrightarrow{i}_{\mathcal{P}^{(2)}}$.

For $\text{AST} \xrightarrow{i}_{\mathcal{P}^{(1)} \cup \mathcal{P}^{(2)}}$, it suffices to regard only rewrite sequences that start with multi-distributions of the form $\{1 : t\}$ (see Lemma A.3 in App. A for a proof). Thus, we show by structural induction on the term structure that for every $t \in \mathcal{T}(\Sigma^{\mathcal{P}^{(1)}} \cup \Sigma^{\mathcal{P}^{(2)}}, \mathcal{V})$, all $\xrightarrow{i}_{\mathcal{P}^{(1)} \cup \mathcal{P}^{(2)}}$ -rewrite sequences starting with $\{1 : t\}$ converge with probability 1.

If $t \in \mathcal{V}$, then t is in normal form. If t is a constant, then w.l.o.g. let $t \in \mathcal{P}^{(1)}$. Since we have $\text{AST} \xrightarrow{i}_{\mathcal{P}^{(1)}}$, t cannot start an infinite $\xrightarrow{i}_{\mathcal{P}^{(1)} \cup \mathcal{P}^{(2)}}$ -rewrite sequence that converges with probability < 1 .

In the induction step we have $t = f(q_1, \dots, q_k)$. Due to the innermost evaluation strategy, we can only rewrite at the root position if every proper subterm is in normal form. Thus, we first only consider rewrite steps below the root. By the induction hypothesis, every infinite $\xrightarrow{i}_{\mathcal{P}^{(1)} \cup \mathcal{P}^{(2)}}$ -rewrite sequence that starts with some $\{1 : q_i\}$ converges with probability 1, and hence, every infinite $\xrightarrow{i}_{\mathcal{P}^{(1)} \cup \mathcal{P}^{(2)}}$ -rewrite sequence that starts with $\{1 : t\}$ converges with probability 1 as well if it does not perform rewrite steps at the root position. However, in the probabilistic setting, this observation requires a quite complex approximation of the convergence probability. The reason is that if we rewrite a term q_i to $\{p_1 : q_{i,1}, \dots, p_m : q_{i,m}\}$, then we obtain a distribution $\{p_1 : f(q_1, \dots, q_{i,1}, \dots, q_k), \dots, p_m : f(q_1, \dots, q_{i,m}, \dots, q_k)\}$. Now, the terms q_j with $j \neq i$ occur multiple times in this distribution, and we may use different rules to rewrite them. Hence, the order in which we rewrite the different q_i matters and cannot be chosen arbitrarily (as seen in Counterex. 4.10). This duplication of the terms q_j is also the reason why $\text{PAST} \xrightarrow{i}_{\mathcal{P}}$ is not modular for disjoint systems, see Ex. 3.15.

In the second step, we also allow rewrite steps at the root position. W.l.o.g., let the root symbol f of t be from $\Sigma^{\mathcal{P}^{(1)}}$. Before performing a rewrite step at the root, we can replace all maximal (i.e., topmost) subterms of t with root symbols from $\Sigma^{\mathcal{P}^{(2)}}$ by fresh variables (using the same variable for the same subterm).⁸ Since f is the root symbol (i.e., there is no symbol at a position above f), and all proper subterms are in normal form, this replacement does not change the convergence probability. After the replacement, we only have function

⁸Note that this replacement is only performed in the induction step at the root, and not for every swap between symbols from $\Sigma^{\mathcal{P}^{(1)}}$ and $\Sigma^{\mathcal{P}^{(2)}}$ on the path from the root to the leaves of t .

symbols from $\Sigma^{\mathcal{P}^{(1)}}$, and thus the convergence probability of the resulting rewrite sequence is 1 since $\text{AST}_{\rightarrow \mathcal{P}^{(1)}}$ holds. \square

In contrast to $\text{AST}_{\rightarrow \mathcal{P}}$, $\text{PAST}_{\rightarrow \mathcal{P}}$ cannot be modular due to the potential extension of the signature (recall that by Thm. 3.16, $\text{PAST}_{\rightarrow \mathcal{P}}$ is not closed under signature extensions).

Counterexample 7.3. Consider the PTRS $\mathcal{P}_{\text{unary}}$ from Counterex. 3.14, and a PTRS \mathcal{P}_{11} containing a binary function symbol c such that $\text{PAST}_{\rightarrow \mathcal{P}_{11}}$ and $\Sigma^{\mathcal{P}_{\text{unary}}} \cap \Sigma^{\mathcal{P}_{11}} = \emptyset$ hold. For example, \mathcal{P}_{11} could consist of the only rule $c(d, d) \rightarrow \{1 : c(e, e)\}$. As explained in Ex. 3.15, due to the signature extension by a binary function symbol, we do not have $\text{PAST}_{\rightarrow \mathcal{P}_{\text{unary}} \cup \mathcal{P}_{11}}$, while both $\text{PAST}_{\rightarrow \mathcal{P}_{\text{unary}}}$ and $\text{PAST}_{\rightarrow \mathcal{P}_{11}}$ hold.

Finally, we consider $\text{SAST}_{\rightarrow \mathcal{P}}$. To prove that $\text{SAST}_{\rightarrow \mathcal{P}}$ is modular for disjoint unions, we have to show that the expected derivation height of any term t is finite. However, after rewriting t 's proper subterms to normal forms, as we did in Ex. 7.1 and in the induction proof of Thm. 7.2, we may end up with infinitely many different terms. All their expected derivation heights have to be considered in order to compute the expected derivation height of t .

Example 7.4. Consider the PTRSs $\mathcal{P}_{12}^{(1)}$ and $\mathcal{P}_{12}^{(2)}$ with

$$\begin{aligned} \mathcal{P}_{12}^{(1)} : & \quad f(s(x), y) \rightarrow \{1 : f(x, y)\} \\ & \quad f(x, s(y)) \rightarrow \{1 : f(x, y)\} \\ & \quad a \rightarrow \{1/2 : 0, 1/2 : s(a)\} \\ \mathcal{P}_{12}^{(2)} : & \quad g(x) \rightarrow \{1 : x\} \end{aligned}$$

Clearly, we have both $\text{SAST}_{\rightarrow \mathcal{P}_{12}^{(1)}}$ and $\text{SAST}_{\rightarrow \mathcal{P}_{12}^{(2)}}$. Now consider the term $t = f(g(a), g(a))$.

Due to the innermost strategy, we have to rewrite its proper subterms first. When proceeding in a similar way as in the induction proof of Thm. 7.2, then one would first construct bounds on the expected derivation heights of the proper subterms, and then use them to obtain a bound on the expected derivation height of the whole term t . However, reducing t 's proper subterms can create infinitely many different terms, i.e., all terms of the form $f(s^n(0), s^m(0))$ for any $n, m \in \mathbb{N}$ can be reached with a certain probability. Since there is no finite supremum on the derivation height of $f(s^n(0), s^m(0))$ for all $n, m \in \mathbb{N}$, one would have to take the individual probabilities for reaching the terms $f(s^n(0), s^m(0))$ into account in order to prove that the expected derivation height of t is indeed finite.

Instead, we use an easier argument to show that any term like t has finite expected derivation height. Recall that in the induction proof of Thm. 7.2, in the induction step we first rewrite below the root position until every proper subterm is in normal form. Afterwards, if the root symbol of t is from $\Sigma^{\mathcal{P}^{(1)}}$, then we replace all maximal subterms of t with root symbols from $\Sigma^{\mathcal{P}^{(2)}}$ by fresh variables. This results in a term t' over $\Sigma^{\mathcal{P}^{(1)}}$ which is considered for the remaining derivation. As shown in Ex. 7.4, there may be infinitely many such terms t' , e.g., in Ex. 7.4, t' can be any term of the form $f(s^n(0), s^m(0))$. However, this infinite set of terms can be over-approximated using the following finite abstraction.

For the root symbol $f \in \Sigma^{\mathcal{P}^{(1)}}$ of $t = f(g(a), g(a))$, the normal forms reachable from $g(a)$ can be over-approximated by considering the normal forms reachable from the argument a of $g \in \Sigma^{\mathcal{P}^{(2)}}$ (because function symbols like g may have “collapsing rules” which return their arguments) or by fresh variables (which represent possible normal forms that start with symbols from $\Sigma^{\mathcal{P}^{(2)}}$). Thus, instead of considering the rewrite steps at symbols

from $\Sigma^{\mathcal{P}^{(1)}}$ in t , instead we can consider all rewrite steps for the terms from the multiset $\{f(a, a), f(x, a), f(a, y), f(x, y)\}$. This multiset is called the *disjoint union abstraction* of t for $\Sigma^{\mathcal{P}^{(1)}}$. Note that all terms in this disjoint union abstraction are indeed from $\mathcal{T}(\Sigma^{\mathcal{P}^{(1)}}, \mathcal{V})$. To also capture the possibility that the two occurrences of $g(a)$ in t might reach the same normal form that starts with a symbol from $\Sigma^{\mathcal{P}^{(2)}}$, the disjoint union abstraction of t also contains $f(x, x)$ where we identify the variables x and y .

Similarly, instead of considering the rewrite steps at symbols from $\Sigma^{\mathcal{P}^{(2)}}$ in t , we consider the two arguments of f (i.e., the two occurrences of $g(a)$) with roots from $\Sigma^{\mathcal{P}^{(2)}}$ where each occurrence of the subterm $a \in \Sigma^{\mathcal{P}^{(1)}}$ is replaced by a fresh variable (to represent possible normal forms that start with symbols from $\Sigma^{\mathcal{P}^{(1)}}$). Thus, the disjoint union abstraction of t for $\Sigma^{\mathcal{P}^{(2)}}$ contains $g(x)$ and $g(y)$.

So instead of an induction proof as for the modularity of $\mathbf{AST}_{\rightarrow \mathcal{P}}$,⁹ for the modularity proof of $\mathbf{SAST}_{\rightarrow \mathcal{P}}$, we replace the start term t by all terms in the disjoint union abstractions for both $\Sigma^{\mathcal{P}^{(1)}}$ and $\Sigma^{\mathcal{P}^{(2)}}$. This is a finite multiset of K terms for some $K \in \mathbb{N}$. Since every term in this abstraction is either from $\mathcal{T}(\Sigma^{\mathcal{P}^{(1)}}, \mathcal{V})$ or from $\mathcal{T}(\Sigma^{\mathcal{P}^{(2)}}, \mathcal{V})$, they all have a finite expected derivation height. Hence, if $C_{\max} \in \mathbb{N}$ is the maximal expected derivation height of all these terms, then $K \cdot C_{\max}$ is a (finite) bound on the expected derivation height of t .

The following definition introduces the *disjoint union abstraction* formally. Here, $A_1(t)$ is the multiset where all topmost subterms of t with root from $\Sigma^{\mathcal{P}^{(2)}}$ are replaced by fresh variables or by the abstractions of their subterms. The multiset $\text{Abs}_1(t)$ results from $A_1(t)$ by identifying any possible combination of the variables in the terms of $A_1(t)$.

Definition 7.5 (Disjoint Union Abstraction). Let $\mathcal{P}^{(1)}, \mathcal{P}^{(2)}$ be PTRSs with $\Sigma^{\mathcal{P}^{(1)}} \cap \Sigma^{\mathcal{P}^{(2)}} = \emptyset$. For any $d \in \{1, 2\}$ and any $t \in \mathcal{T}(\Sigma^{\mathcal{P}^{(1)}} \cup \Sigma^{\mathcal{P}^{(2)}}, \mathcal{V})$, $A_d(t)$ and $\text{Abs}_d(t)$ are multisets of terms from $\mathcal{T}(\Sigma^{\mathcal{P}^{(d)}}, \mathcal{V})$, which are defined as follows.

$$\begin{aligned} A_d(y) &= \{x\}, \text{ if } y \in \mathcal{V}, \text{ where } x \text{ is always a new fresh variable} \\ A_d(f(t_1, \dots, t_k)) &= \{f(q_1, \dots, q_k) \mid q_1 \in A_d(t_1), \dots, q_k \in A_d(t_k)\}, \text{ if } f \in \Sigma^{\mathcal{P}^{(d)}} \\ A_d(f(t_1, \dots, t_k)) &= \{x\} \cup A_d(t_1) \cup \dots \cup A_d(t_k), \text{ otherwise, where } x \text{ is always a new fresh variable} \end{aligned}$$

So $A_d(t)$ is always a linear term, i.e., it never contains multiple occurrences of the same variable.

For any function $\varphi : X \rightarrow X$ with $X \subseteq \mathcal{V}$, let σ_φ be the substitution that replaces every variable $x \in X$ by $\varphi(x)$ and leaves all other variables unchanged, i.e., $\sigma_\varphi(x) = \varphi(x)$ if $x \in X$ and $\sigma_X(x) = x$ otherwise. Then we define

$$\text{Abs}_d(t) = \{\sigma_\varphi(q) \mid q \in A_d(t), \varphi : \mathcal{V}(q) \rightarrow \mathcal{V}(q)\}$$

The *disjoint union abstraction* of t is the multiset $\text{Abs}_1(t) \cup \text{Abs}_2(t)$.

⁹The proof idea that we use for the modularity of $\mathbf{SAST}_{\rightarrow \mathcal{P}}$ for disjoint unions could also have been used to prove the modularity of $\mathbf{AST}_{\rightarrow \mathcal{P}}$ for disjoint unions (Thm. 7.2). However, for Thm. 7.2 we used an induction proof instead to have it similar to the original proof of [Gra95] in the non-probabilistic setting. Moreover, such an induction proof will also be required when showing the modularity of $\mathbf{AST}_{\rightarrow \mathcal{P}}$ for shared constructor unions (Thm. 7.13).

Example 7.6. Reconsider $\mathcal{P}_{12}^{(1)}$ and $\mathcal{P}_{12}^{(2)}$ from Ex. 7.4. Here, we obtain

$$\begin{aligned} A_1(f(g(a), g(a))) &= \{f(a, a), f(x, a), f(a, y), f(x, y)\} \\ \text{Abs}_1(f(g(a), g(a))) &= \{f(a, a), f(x, a), f(a, y), f(x, y), f(y, x), f(x, x), f(y, y)\} \\ A_2(f(g(a), g(a))) &= \{x', g(x), g(y)\} \\ \text{Abs}_2(f(g(a), g(a))) &= \{x', g(x), g(y)\} \end{aligned}$$

The following lemma states the two most important properties regarding the disjoint union abstraction.

Lemma 7.7 (Properties of Abs_d). *Let $\mathcal{P}^{(1)}, \mathcal{P}^{(2)}$ be PTRSs with $\Sigma^{\mathcal{P}^{(1)}} \cap \Sigma^{\mathcal{P}^{(2)}} = \emptyset$, $t \in \mathcal{T}(\Sigma^{\mathcal{P}^{(1)}} \cup \Sigma^{\mathcal{P}^{(2)}}, \mathcal{V})$, and $d \in \{1, 2\}$. Then:*

- (i) $\text{Abs}_d(t)$ is finite
- (ii) $\text{Abs}_d(t) \subseteq \mathcal{T}(\Sigma^{\mathcal{P}^{(d)}}, \mathcal{V})$

Proof. Simple proof by induction on the term structure. □

With the disjoint union abstraction, we can prove the modularity of $\text{SAST}_{\dot{\rightarrow}_{\mathcal{P}}}$ for disjoint unions.

Theorem 7.8 (Modularity of $\text{SAST}_{\dot{\rightarrow}_{\mathcal{P}}}$ for Disjoint Unions). *Let $\mathcal{P}^{(1)}$ and $\mathcal{P}^{(2)}$ be PTRSs with $\Sigma^{\mathcal{P}^{(1)}} \cap \Sigma^{\mathcal{P}^{(2)}} = \emptyset$. Then we have:*

$$\text{SAST}_{\dot{\rightarrow}_{\mathcal{P}^{(1)} \cup \mathcal{P}^{(2)}}} \iff \text{SAST}_{\dot{\rightarrow}_{\mathcal{P}^{(1)}}} \text{ and } \text{SAST}_{\dot{\rightarrow}_{\mathcal{P}^{(2)}}}$$

Proof Sketch. The direction “ \implies ” is trivial and thus, we only prove “ \impliedby ”. So let $\mathcal{P} = \mathcal{P}^{(1)} \cup \mathcal{P}^{(2)}$ where $\text{SAST}_{\dot{\rightarrow}_{\mathcal{P}^{(1)}}}$ and $\text{SAST}_{\dot{\rightarrow}_{\mathcal{P}^{(2)}}}$. Let \mathfrak{T} be an arbitrary $\dot{\rightarrow}_{\mathcal{P}}$ -RST that starts with $(1 : t)$. We show that $\text{edl}(\mathfrak{T})$ is bounded by some constant which does not depend on \mathfrak{T} but just on t . This proves that $\text{SAST}_{\dot{\rightarrow}_{\mathcal{P}}}$ holds.

Since we have $\text{SAST}_{\dot{\rightarrow}_{\mathcal{P}^{(d)}}}$, the expected derivation length of all $\dot{\rightarrow}_{\mathcal{P}^{(d)}}$ -RSTs with $d \in \{1, 2\}$ that start with a term $q \in \mathcal{T}(\Sigma^{\mathcal{P}^{(d)}}, \mathcal{V})$ is bounded by some constant $C_q < \omega$. Thus, since $|\text{Abs}_1(t) \cup \text{Abs}_2(t)| = K \in \mathbb{N}$ is finite, there is a $C_{\max} < \omega$ such that for all $q \in \text{Abs}_1(t) \cup \text{Abs}_2(t)$ we have $\text{edl}(\mathfrak{T}') \leq C_{\max}$ for every $\dot{\rightarrow}_{\mathcal{P}^{(d)}}$ -RST \mathfrak{T}' that starts with $(1 : q)$. Hence, we obtain $\text{edl}(\mathfrak{T}) \leq K \cdot C_{\max}$. □

Example 7.9. Let us illustrate the notation in the previous proof sketch by applying it to the PTRS from Ex. 7.4. If we reconsider the start term $f(g(a), g(a))$, then $\text{Abs}_1(f(g(a), g(a))) = \{f(a, a), f(x, a), f(a, y), f(x, y), f(y, x), f(x, x), f(y, y)\}$ and $\text{Abs}_2(f(g(a), g(a))) = \{x', g(x), g(y)\}$, as in Ex. 7.6. Moreover, we obtain $C_{\max} = C_{f(a, a)} = 2 + 2 + 2 \cdot \sum_{n=1}^{\infty} (1/2)^{n+1} \cdot n = 2 + 2 + 2 \cdot 1 = 6$, where $C_{f(a, a)}$ is the bound on the expected derivation height of the term $f(a, a)$. (The reason is that reaching a normal form from the subterm a needs 2 steps in expectation, and then for each generated s we have one additional step, where each a generates n s -symbols with probability $(1/2)^{n+1}$.) Hence, the overall bound on the expected derivation height for $f(g(a), g(a))$ is $K \cdot C_{\max} = 10 \cdot 6 = 60$. In fact, the actual expected derivation height for $f(g(a), g(a))$ is $6 + 2 = 8$ (2 steps for the two g -symbols, and in expectation 6 steps for the term $f(a, a)$), but for the proof any finite bound suffices.

For full rewriting, it is well known that termination is already not modular in the non-probabilistic setting.

Counterexample 7.10. Reconsider the TRS \mathcal{R}_1 from Counterex. 2.1. This TRS is the disjoint union of $\mathcal{R}_1^{(1)} = \{f(a, b, x) \rightarrow f(x, x, x)\}$ and $\mathcal{R}_1^{(2)} = \{g \rightarrow a, g \rightarrow b\}$. Both $\mathcal{R}_1^{(1)}$ and $\mathcal{R}_1^{(2)}$ are terminating, but the disjoint union \mathcal{R}_1 is not.

However, one can reuse our results from Section 4 to obtain the following corollary.

Corollary 7.11 (Modularity of $\text{PSN}_{\rightarrow \mathcal{P}}$ for Disjoint Unions). *Let $\mathcal{P}^{(1)}$ and $\mathcal{P}^{(2)}$ be PTRSs with $\Sigma^{\mathcal{P}^{(1)}} \cap \Sigma^{\mathcal{P}^{(2)}} = \emptyset$ that are NO and linear. Then we have:*

$$\begin{aligned} \text{AST}_{\rightarrow \mathcal{P}^{(1)} \cup \mathcal{P}^{(2)}} &\iff \text{AST}_{\rightarrow \mathcal{P}^{(1)}} \text{ and } \text{AST}_{\rightarrow \mathcal{P}^{(2)}} \\ \text{SAST}_{\rightarrow \mathcal{P}^{(1)} \cup \mathcal{P}^{(2)}} &\iff \text{SAST}_{\rightarrow \mathcal{P}^{(1)}} \text{ and } \text{SAST}_{\rightarrow \mathcal{P}^{(2)}} \end{aligned}$$

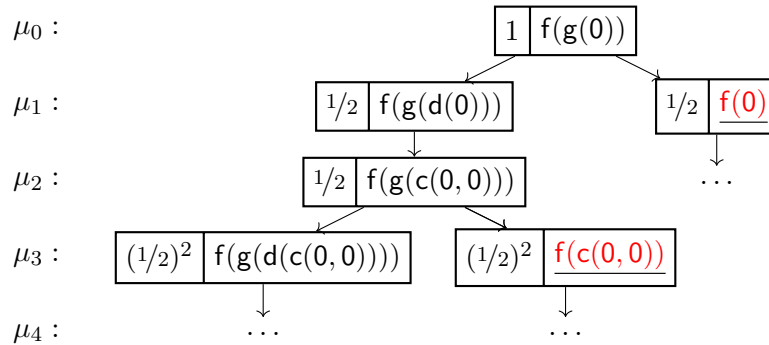
7.2. Shared Constructor Unions. Now we consider unions of PTRSs that may share constructor symbols, i.e., we consider two PTRSs $\mathcal{P}^{(1)}$ and $\mathcal{P}^{(2)}$ such that $\Sigma_D^{\mathcal{P}^{(1)}} \cap \Sigma_D^{\mathcal{P}^{(2)}} = \emptyset$, called *shared constructor unions*. Again, we study innermost rewriting first.

In the non-probabilistic setting, innermost termination is also modular for shared constructor unions [Gra95]. However, $\text{PAST}_{\rightarrow \mathcal{P}}$ was already not modular w.r.t. disjoint unions, so this also holds for shared constructor unions. Moreover, $\text{SAST}_{\rightarrow \mathcal{P}}$ also turns out to be not modular anymore for shared constructor unions.

Counterexample 7.12. Consider the PTRS $\mathcal{P}_{13} = \mathcal{P}_{13}^{(1)} \cup \mathcal{P}_{13}^{(2)}$ with the rules

$$\begin{aligned} \mathcal{P}_{13}^{(1)} : f(c(x, y)) &\rightarrow \{1 : c(f(x), f(y))\} & \mathcal{P}_{13}^{(2)} : g(x) &\rightarrow \{1/2 : g(d(x)), 1/2 : x\} \\ f(0) &\rightarrow \{1 : 0\} & d(x) &\rightarrow \{1 : c(x, x)\} \end{aligned}$$

While $\mathcal{P}_{13}^{(1)}$ and $\mathcal{P}_{13}^{(2)}$ do not have any common defined symbols, they share the constructor c . We do not have $\text{PAST}_{\rightarrow \mathcal{P}_{13}}$ (and thus, not $\text{SAST}_{\rightarrow \mathcal{P}_{13}}$), as the infinite $\xrightarrow{1}_{\mathcal{P}_{13}}$ -rewrite sequence $(\mu_n)_{n \in \mathbb{N}}$ depicted in the following $\xrightarrow{1}_{\mathcal{P}_{13}}$ -RST has an infinite expected derivation length.



For any $n \in \mathbb{N}$, each underlined term $f(c^n(0, 0))$ in the tree above can start a reduction of at least length 2^n , where $c^n(0, 0)$ corresponds to the full binary tree of height n with c in inner nodes and 0 in the leaves. Hence, the term $f(g(0))$ has an expected derivation height of at least $\sum_{n=0}^{\infty} \frac{1}{2^{n+1}} \cdot 2^n = \sum_{n=0}^{\infty} \frac{1}{2}$, which diverges to infinity.

On the other hand, we have $\text{SAST}_{\rightarrow \mathcal{P}_{13}^{(1)}}$, as $\mathcal{P}_{13}^{(1)}$ is a PTRS with only trivial probabilities that corresponds to a terminating TRS. Moreover, $\text{SAST}_{\rightarrow \mathcal{P}_{13}^{(2)}}$ holds as well, as the d -rule can

increase the number of c -symbols in a term exponentially, but those c -symbols will never be used. Thus, $\text{SAST}_{\dot{\rightarrow}_{\mathcal{P}}}$ is not modular for shared constructor unions.

In contrast to the proof of Thm. 7.8, we cannot use the disjoint union abstraction anymore to obtain a bound on the expected derivation height, since symbols from $\Sigma^{\mathcal{P}^{(2)}}$ can now “generate” constructor symbols of $\Sigma^{\mathcal{P}^{(1)}}$. However, for $\text{AST}_{\dot{\rightarrow}_{\mathcal{P}}}$ we can reuse the idea of the previous proof for Thm. 7.2 to obtain a similar result for shared constructor unions.

Theorem 7.13 (Modularity of $\text{AST}_{\dot{\rightarrow}_{\mathcal{P}}}$ for Shared Constructor Unions). *Let $\mathcal{P}^{(1)}$ and $\mathcal{P}^{(2)}$ be PTRSs with $\Sigma_D^{\mathcal{P}^{(1)}} \cap \Sigma_D^{\mathcal{P}^{(2)}} = \emptyset$. Then we have:*

$$\text{AST}_{\dot{\rightarrow}_{\mathcal{P}^{(1)} \cup \mathcal{P}^{(2)}}} \iff \text{AST}_{\dot{\rightarrow}_{\mathcal{P}^{(1)}}} \text{ and } \text{AST}_{\dot{\rightarrow}_{\mathcal{P}^{(2)}}}$$

Proof. The proof for “ \Leftarrow ” is again via structural induction on the term t in the initial multi-distribution $\{1 : t\}$ and very similar to the proof of Thm. 7.2. The only difference is that in the induction step, after performing rewrite steps below the root until all proper subterms are in normal form, if the root is from $\Sigma^{\mathcal{P}^{(1)}}$, then we do not replace all maximal subterms with roots from $\Sigma^{\mathcal{P}^{(2)}}$ by fresh variables but just maximal subterms with root symbols from $\Sigma_D^{\mathcal{P}^{(2)}}$, as the constructor symbols may be used by rules of $\mathcal{P}^{(1)}$. This, however, does not interfere with the proof idea. \square

Again, we can reuse our results from Section 4 on the relation between full and innermost rewriting to obtain the following corollary for full rewriting. Due to Counterex. 7.10, this corollary does not hold for general PTRSs.

Corollary 7.14 (Modularity of $\text{AST}_{\rightarrow_{\mathcal{P}}}$ for Shared Constructor Unions). *Let $\mathcal{P}^{(1)}$ and $\mathcal{P}^{(2)}$ be PTRSs with $\Sigma_D^{\mathcal{P}^{(1)}} \cap \Sigma_D^{\mathcal{P}^{(2)}} = \emptyset$ that are NO and linear. Then we have:*

$$\text{AST}_{\rightarrow_{\mathcal{P}^{(1)} \cup \mathcal{P}^{(2)}}} \iff \text{AST}_{\rightarrow_{\mathcal{P}^{(1)}}} \text{ and } \text{AST}_{\rightarrow_{\mathcal{P}^{(2)}}}$$

7.3. Signature Extensions. Finally, we study signature extensions of PTRSs. While $\text{PAST}_{\dot{\rightarrow}_{\mathcal{P}}}$ is not closed under signature extensions by Thm. 3.16, we now consider $\text{AST}_{\dot{\rightarrow}_{\mathcal{P}}}$ and $\text{SAST}_{\dot{\rightarrow}_{\mathcal{P}}}$. Signature extensions can be seen as special cases of disjoint unions, where the second PTRS $\mathcal{P}^{(2)}$ contains only trivially terminating rules over the new signature that we want to add to $\Sigma^{\mathcal{P}^{(1)}}$. Hence, for innermost rewriting, Thm. 7.2 and 7.8 already imply that $\text{AST}_{\dot{\rightarrow}_{\mathcal{P}}}$ and $\text{SAST}_{\dot{\rightarrow}_{\mathcal{P}}}$ are closed under signature extensions.

For full rewriting, Cor. 7.11 implies that $\text{AST}_{\rightarrow_{\mathcal{P}}}$ and $\text{SAST}_{\rightarrow_{\mathcal{P}}}$ are closed under signature extensions for non-overlapping and linear PTRSs. We now show that this also holds for arbitrary PTRSs. So let \mathcal{P} be an arbitrary PTRS over the signature $\Sigma^{\mathcal{P}}$ for the rest of this section. We consider two cases.

First, let $\Sigma^{\mathcal{P}}$ contain only constants and unary symbols, e.g., $\Sigma^{\mathcal{P}} = \{f, a, b\}$ where f is unary and a, b are constants. If we extend $\Sigma^{\mathcal{P}}$ by a signature Σ' that may also contain symbols of other arities, e.g., a symbol c of arity 2, and consider terms from $\mathcal{T}(\Sigma^{\mathcal{P}} \cup \Sigma', \mathcal{V})$ like $f(c(f(a), f(b)))$, then the fresh symbol c “completely separates” the function symbols from $\Sigma^{\mathcal{P}}$ occurring below and above c . More precisely, instead of $f(c(f(a), f(b)))$, it suffices to analyze the start terms $f(a)$, $f(b)$, and $f(x_c)$ (where the c -subterm is replaced by a fresh variable x_c), which are all from $\mathcal{T}(\Sigma^{\mathcal{P}}, \mathcal{V})$. The reason is that rewriting above c does not interfere with the terms below c and vice versa. Hence, if every rewrite sequence that starts with $f(a)$, $f(b)$, or $f(x_c)$ converges with probability 1, then so does the term $f(c(f(a), f(b)))$. In other words,

if $\text{AST}_{\mathcal{F}_{\mathcal{P}}}$ holds over the signature $\Sigma^{\mathcal{P}}$, then $\text{AST}_{\mathcal{F}_{\mathcal{P}}}$ also holds over the signature $\Sigma^{\mathcal{P}} \cup \Sigma'$. Furthermore, the expected derivation height of $f(c(f(a), f(b)))$ is bounded by $3 \cdot C_{\max}$, where C_{\max} is the maximum expected derivation height of the terms $f(a)$, $f(b)$, and $f(x_c)$, similar as in the proof of Thm. 7.8. Thus, if we have $\text{SAST}_{\mathcal{F}_{\mathcal{P}}}$ over the signature $\Sigma^{\mathcal{P}}$, then we also have $\text{SAST}_{\mathcal{F}_{\mathcal{P}}}$ over the signature $\Sigma^{\mathcal{P}} \cup \Sigma'$.

Second, we consider the case where $\Sigma^{\mathcal{P}}$ itself already contains a function symbol g that has at least arity 2. Again, the fresh symbols c of Σ' separate the function symbols of $\Sigma^{\mathcal{P}}$ occurring above and below them. However, now the expected derivation height of a term like $f(c(f(a), f(b), f(y)))$ from $\mathcal{T}(\Sigma^{\mathcal{P}} \cup \Sigma', \mathcal{V})$ is not bounded by simply adding the expected derivation heights of the corresponding terms $f(a)$, $f(b)$, $f(y)$, $f(x_c)$ from $\mathcal{T}(\Sigma^{\mathcal{P}}, \mathcal{V})$ anymore. The reason is that \mathcal{P} might now duplicate subterms (e.g., there could be a rule like $f(x) \rightarrow \{1 : g(x, x)\}$). However, for any term $t \in \mathcal{T}(\Sigma^{\mathcal{P}} \cup \Sigma', \mathcal{V})$, we can now construct a term $t' \in \mathcal{T}(\Sigma^{\mathcal{P}}, \mathcal{V})$ over the original signature $\Sigma^{\mathcal{P}}$ that has (at least) the same expected derivation height and (at most) the same convergence probability.

The construction works as follows: Let $g \in \Sigma^{\mathcal{P}}$ be a symbol of arity 2 (if its arity is greater than two, then we use the term $g(-, -, x, \dots, x)$ instead, where $x \in \mathcal{V}$). For example, if we extend the signature $\Sigma^{\mathcal{P}}$ by a symbol $c \in \Sigma'$ of arity 3, then instead of a term like $f(c(f(a), f(b), f(y)))$ from $\mathcal{T}(\Sigma^{\mathcal{P}} \cup \Sigma', \mathcal{V})$, we can consider the term $f(g(f(a), g(f(b), g(f(y), x_c))))$ from $\mathcal{T}(\Sigma^{\mathcal{P}}, \mathcal{V})$ without the symbol c , where we do not rewrite the newly added g symbols. So here, we replaced $c(-, -, -)$ by $g(-, g(-, g(-, x_c)))$. Note that this construction works for full but not for innermost rewriting, since it may create new redexes with the symbol g that we may have to rewrite when using the innermost strategy. However, for innermost rewriting we have already proven closedness under signature extensions by Thm. 7.2 and 7.8, as explained above.

To summarize, this leads to the following theorem.

Theorem 7.15 (Signature Extensions for $\text{AST}_{\mathcal{F}_{\mathcal{P}}}$ and $\text{SAST}_{\mathcal{F}_{\mathcal{P}}}$). *Let \mathcal{P} be a PTRS, $s \in \{\mathbf{f}, \mathbf{i}\}$, and let Σ' be some signature. Then we have:*

$$\begin{aligned} \text{AST}_{\mathcal{F}_{\mathcal{P}}} \text{ over } \Sigma^{\mathcal{P}} &\iff \text{AST}_{\mathcal{F}_{\mathcal{P}}} \text{ over } \Sigma^{\mathcal{P}} \cup \Sigma' \\ \text{SAST}_{\mathcal{F}_{\mathcal{P}}} \text{ over } \Sigma^{\mathcal{P}} &\iff \text{SAST}_{\mathcal{F}_{\mathcal{P}}} \text{ over } \Sigma^{\mathcal{P}} \cup \Sigma' \end{aligned}$$

8. RELATED WORK ON VERIFICATION OF PROBABILISTIC PROGRAMS

In the previous sections, we already discussed the connection to related work in term rewriting. However, verification of probabilistic programs has also been studied extensively for imperative programs on numbers, and for different recursive programming languages like the lambda calculus. Thus, in this section we discuss existing work on probabilistic termination analysis outside term rewriting.

Its *hardness* has been investigated in [KKM19, MS24], showing that analyzing almost-sure termination is even more difficult than ordinary termination when considering the halting problem for given inputs. In our paper, SN and AST refer to universal termination, as we consider rewrite sequences starting with arbitrary terms. While for non-probabilistic programs, universal termination is “harder” than the halting problem for given inputs, these problems are equally hard for probabilistic programs [KKM19].

There exist numerous approaches and proof rules mainly based on martingales for different properties of probabilistic programs. For example, there are techniques for

proving **AST** and **PAST** [FFH15, CFG16, ACN17, CFNH18, HFC18, MMKK18, HFCG19, CFN20, AGR21, MBKK21, CGN⁺23, MS25], for proving bounds on the termination probability [CNZ17, KUH19, CGMZ22, MSBK22, FCS⁺23], and for upper and lower bounds on expected runtimes and costs [KKMO18, NCH18, FC19, GGH19, WFG⁺19, AMS20, HKGK20, KKM20, MHG21, DWH23, LMG24]. Many of these approaches can be automated directly or by extending them with automatic invariant synthesis techniques, e.g., [KMMM10, BEFH16, BTP⁺22, KMS⁺22, BCJ⁺23]. Moreover, one can also use such proof rules within a quantitative program verification infrastructure like **Caesar** [SBK⁺23]. Some of these proof rules have already been adapted to term rewriting, e.g., the proof rule of [MMKK18] can be adapted to prove **AST** of PTRSs via polynomial and matrix interpretations [KG23a] and it is also used within the probabilistic DP framework [KG23a, KDG24, KG24]. Further proof rules regarding **PAST** and **SAST** have been adapted in [ADLY20] and again integrated in the corresponding DP framework [KSG25].

In particular, there also exist several *tools* to analyze **AST**, **PAST**, and expected costs for imperative probabilistic programs, e.g., **Amber** [MBKK21], **KoAT** [MHG21, LMG24], **Eco-lmp** [AMS20], **Absynth** [NCH18], and **Pastry** [NYC⁺25]. Moreover, higher moments for a loop's variables are analyzed automatically with the tool **Polar** [MSBK22]. These tools mainly consider imperative programs with an innermost evaluation strategy. Hence, our results on the relation between the different rewrite strategies cannot be directly used for these tools, while our results on modularity may in principle be of interest. However, our results concern the functional recursive nature of term rewriting, where one does not have a fixed control flow as in an imperative program. Our comparison between **PAST** and **SAST** in Thm. 3.21 should also hold for imperative programs which allow for multiple executions in parallel that are considered simultaneously.

Compared to these tools, our implementation in **AProVE** currently can prove **AST**, **PAST**, **SAST**, and analyze the expected complexity of PTRSs. For algorithms whose termination behavior relies only on numbers, techniques for imperative programs with built-in support for arithmetic are usually more powerful than approaches based on term rewriting. The reason is that for term rewriting, numbers have to be represented via terms, e.g., 2 can be represented by the term $s(s(0))$. On the other hand, term rewriting can handle programs with non-trivial recursive structure and arbitrary user-defined data structures, as these structures can easily be represented as terms. For example, the list $[2, 1]$ can be represented by the term $\text{cons}(s(s(0)), \text{cons}(s(0), \text{nil}))$. Thus, tools based on term rewriting are particularly suitable when analyzing programs whose termination depends on, e.g., lists, trees, or graphs. In the non-probabilistic setting, there are also techniques and tools for term rewriting with integrated built-in numbers, e.g., for *integer term rewrite systems* [FGP⁺09] or *logically constrained term rewrite systems* [KN13]. Lifting these approaches to the probabilistic setting is an interesting direction for future work.

In addition to the related work on probabilistic loop programs, there are also several approaches for probabilistic recursive programs, e.g., to analyze the probabilistic lambda calculus or other higher-order functional languages based on types or martingales [ALG19, LG19, BO21, KO21, LFR21, RBG24, DLP24]. There has also been work on functional languages where **AST** or **PAST** is (partly) decidable, e.g., probabilistic higher-order rewrite schemes [KLG20], probabilistic pushdown automata [BEKK13, BKKV15, WK23], and restricted probabilistic tree-stack automata [LMO22]. Many of the results on recursive languages fix a leftmost-innermost rewrite strategy to avoid non-determinism. Here, our

results may be helpful to extend these techniques to different rewrite strategies, and also our modularity results may be of interest for the different recursive probabilistic languages.

Finally, probabilistic programs that allow for *data structures* are analyzed in [WKH20, LMZ22, BKK⁺23]. While [BKK⁺23] uses pointers to represent data structures like tables and lists, [WKH20, LMZ22] consider a probabilistic programming language with matching similar to term rewriting and develop an automatic amortized resource analysis via fixed template potential functions. However, these latter works are mostly targeted towards specific data structures, and we consider general term rewrite systems that can model arbitrary data structures.

9. CONCLUSION

In this paper, we presented the first results on the relationship between $\text{AST}_{\rightarrow_P}^s$ of a PTRS \mathcal{P} for different rewrite strategies $s \in \mathbb{S}$, including several criteria such that $\text{AST}_{\rightarrow_P}^{\text{I}}$ implies $\text{AST}_{\rightarrow_P}^{\text{F}}$. Our results also hold for $\text{PAST}_{\rightarrow_P}^s$, $\text{SAST}_{\rightarrow_P}^s$, and expected complexity, and all of our criteria are suitable for automation (for sparseness, there exist sufficient conditions that can be checked automatically). We implemented our criteria for the equivalence of $\text{AST}_{\rightarrow_P}^{\text{I}}$ and $\text{AST}_{\rightarrow_P}^{\text{F}}$ in our termination prover AProVE, and demonstrated their practical usefulness in an experimental evaluation. Moreover, we developed the first modularity results for termination of PTRSs under unions and signature extensions.

In the paper, we already mentioned several topics for future work, e.g.:

- Improving our current results (in Cor. 4.8 and 4.14) on the relationship between non-probabilistic innermost and full derivational complexity.
- Extending our modularity results to hierarchical unions or finding more specific classes where $\text{SAST}_{\rightarrow_P}^{\text{I}}$ is modular for shared constructor unions, or even classes where $\text{PAST}_{\rightarrow_P}^{\text{I}}$ becomes modular.
- Tackling the (hard) problem of adapting Newman’s Lemma [New42] to the probabilistic setting.

Acknowledgements. We thank Florian Frohn for his help on the earlier conference paper [KFG24] and Stefan Dollase for pointing us to [Fuh19].

REFERENCES

- [AAR20] Ariane Alves Almeida and Mauricio Ayala-Rincón. Formalizing the dependency pair criterion for innermost termination. *Sci. Comput. Prog.*, 195, 2020. doi:10.1016/j.scico.2020.102474.
- [ACN17] Sheshansh Agrawal, Krishnendu Chatterjee, and Petr Novotný. Lexicographic ranking supermartingales: An efficient approach to termination of probabilistic programs. *Proc. ACM Program. Lang.*, 2(POPL), 2017. doi:10.1145/3158122.
- [ADLY20] Martin Avanzini, Ugo Dal Lago, and Akihisa Yamada. On probabilistic term rewriting. *Sci. Comput. Program.*, 185, 2020. doi:10.1016/j.scico.2019.102338.
- [AG00] Thomas Arts and Jürgen Giesl. Termination of term rewriting using dependency pairs. *Theor. Comput. Sc.*, 236(1-2):133–178, 2000. doi:10.1016/S0304-3975(99)00207-8.

- [AGR21] Alessandro Abate, Mirco Giacobbe, and Diptarko Roy. Learning probabilistic termination proofs. In *Proc. CAV '21*, LNCS 12760, pages 3–26, 2021. doi:10.1007/978-3-030-81688-9_1.
- [ALG19] Martin Avanzini, Ugo Dal Lago, and Alexis Ghyselen. Type-based complexity analysis of probabilistic functional programs. In *Proc. LICS '19*, 2019. doi:10.1109/LICS.2019.8785725.
- [AM16] Martin Avanzini and Georg Moser. A combination framework for complexity. *Inf. Comput.*, 248:22–55, 2016. doi:10.1016/j.ic.2015.12.007.
- [AMS06] Gul Agha, José Meseguer, and Koushik Sen. PMAude: Rewrite-based specification language for probabilistic object systems. In *Proc. QAPL '05*, ENTCS 153, pages 213–239, 2006. doi:10.1016/j.entcs.2005.10.040.
- [AMS16] Martin Avanzini, Georg Moser, and Michael Schaper. TcT: Tyrolean Complexity Tool. In *Proc. TACAS '16*, LNCS 9636, pages 407–423, 2016. doi:10.1007/978-3-662-49674-9_24.
- [AMS20] Martin Avanzini, Georg Moser, and Michael Schaper. A modular cost analysis for probabilistic programs. *Proc. ACM Program. Lang.*, 4(OOPSLA), 2020. doi:10.1145/3428240.
- [BCJ⁺23] Kevin Batz, Mingshuai Chen, Sebastian Junges, Benjamin Lucien Kaminski, Joost-Pieter Katoen, and Christoph Matheja. Probabilistic program verification via inductive synthesis of inductive invariants. In *Proc. TACAS '23*, LNCS 13994, pages 410–429, 2023. doi:10.1007/978-3-031-30820-8_25.
- [BEFH16] Gilles Barthe, Thomas Espitau, Luis María Ferrer Fioriti, and Justin Hsu. Synthesizing probabilistic invariants via Doob’s decomposition. In *Proc. CAV '16*, LNCS 9779, pages 43–61, 2016. doi:10.1007/978-3-319-41528-4_3.
- [BEKK13] Tomáš Brázdil, Javier Esparza, Stefan Kiefer, and Antonín Kučera. Analyzing probabilistic pushdown automata. *Formal Methods in System Design*, 43:124–163, 2013. doi:10.1007/s10703-012-0166-0.
- [BG05] Olivier Bournez and Florent Garnier. Proving positive almost-sure termination. In *Proc. RTA '05*, LNCS 3467, pages 323–337, 2005. doi:10.1007/978-3-540-32033-3_24.
- [BK02] Olivier Bournez and Claude Kirchner. Probabilistic rewrite strategies. Applications to ELAN. In *Proc. RTA '02*, LNCS 2378, pages 252–266, 2002. doi:10.1007/3-540-45610-4_18.
- [BK11] Frédéric Blanqui and Adam Koprowski. CoLoR: a Coq library on well-founded rewrite relations and its application to the automated verification of termination certificates. *Math. Struct. Comput. Sci.*, 21(4):827–859, 2011. doi:10.1017/S0960129511000120.
- [BKK⁺23] Kevin Batz, Benjamin Lucien Kaminski, Joost-Pieter Katoen, Christoph Matheja, and Lena Verscht. A calculus for amortized expected runtimes. *Proc. ACM Program. Lang.*, 7(POPL):1957–1986, 2023. doi:10.1145/3571260.
- [BKKV15] Tomáš Brázdil, Stefan Kiefer, Antonín Kucera, and Ivana Hutarová Vareková. Runtime analysis of probabilistic programs with unbounded recursion. *J. Comput. Syst. Sci.*, 81(1):288–310, 2015. doi:10.1016/J.JCSS.2014.06.005.
- [BN98] Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998. doi:10.1017/CB09781139172752.
- [BO21] Raven Beutner and Luke Ong. On probabilistic termination of functional programs with continuous distributions. In *Proc. PLDI '21*, pages 1312–1326,

2021. doi:10.1145/3453483.3454111.
- [BTP⁺22] Jialu Bao, Nitesh Trivedi, Drashti Pathak, Justin Hsu, and Subhajit Roy. Data-driven invariant learning for probabilistic programs. In *Proc. CAV '22*, LNCS 13371, pages 33–54, 2022. doi:10.1007/978-3-031-13185-1_3.
 - [CCF⁺07] Evelyne Contejean, Pierre Courtieu, Julien Forest, Olivier Pons, and Xavier Urbain. Certification of automated termination proofs. In *Proc. FroCoS '07*, LNCS 4720, pages 148–162, 2007. doi:10.1007/978-3-540-74621-8_10.
 - [CFG16] Krishnendu Chatterjee, Hongfei Fu, and Amir Kafshdar Goharshady. Termination analysis of probabilistic programs through positivstellensatz's. In *Proc. CAV '16*, LNCS 9779, pages 3–22, 2016. doi:10.1007/978-3-319-41528-4_1.
 - [CFN20] Krishnendu Chatterjee, Hongfei Fu, and Petr Novotný. Termination analysis of probabilistic programs with martingales. In Gilles Barthe, Joost-Pieter Katoen, and Alexandra Silva, editors, *Foundations of Probabilistic Programming*, page 221–258. Cambridge University Press, 2020. doi:10.1017/9781108770750.008.
 - [CFNH18] Krishnendu Chatterjee, Hongfei Fu, Petr Novotný, and Rouzbeh Hasheminezhad. Algorithmic analysis of qualitative and quantitative termination problems for affine probabilistic programs. *ACM Trans. Program. Lang. Syst.*, 40(2), 2018. doi:10.1145/3174800.
 - [CGMZ22] Krishnendu Chatterjee, Amir Kafshdar Goharshady, Tobias Meggendorfer, and Dorde Zikelic. Sound and complete certificates for quantitative termination analysis of probabilistic programs. In *Proc. CAV '22*, LNCS 13371, pages 55–78, 2022. doi:10.1007/978-3-031-13185-1_4.
 - [CGN⁺23] Krishnendu Chatterjee, Ehsan Kafshdar Goharshady, Petr Novotný, Jiri Závěručky, and Dorde Zikelic. On lexicographic proof rules for probabilistic termination. *Formal Aspects Comput.*, 35(2), 2023. doi:10.1145/3585391.
 - [CNZ17] Krishnendu Chatterjee, Petr Novotný, and Dorde Zikelic. Stochastic invariants for probabilistic termination. In *Proc. POPL '17*, pages 145–160, 2017. doi:10.1145/3009837.3009873.
 - [DCM18] Alejandro Díaz-Caro and Guido Martínez. Confluence in probabilistic rewriting. In *Proc. LSFA '17*, ENTCS 338, pages 115–131, 2018. doi:10.1016/j.entcs.2018.10.008.
 - [DLP24] Ugo Dal Lago and Luca Padovani. On the almost-sure termination of binary sessions. In *Proc. PPDP '24*, 2024. doi:10.1145/3678232.3678239.
 - [DWH23] Ankush Das, Di Wang, and Jan Hoffmann. Probabilistic resource-aware session types. *Proc. ACM Program. Lang.*, 7(POPL):1925–1956, 2023. doi:10.1145/3571259.
 - [Fag22] Claudia Faggian. Probabilistic rewriting and asymptotic behaviour: On termination and unique normal forms. *Log. Methods in Comput. Sci.*, 18(2), 2022. doi:10.46298/lmcs-18(2:5)2022.
 - [FC19] Hongfei Fu and Krishnendu Chatterjee. Termination of nondeterministic probabilistic programs. In *Proc. VMCAI '19*, LNCS 11388, pages 468–490, 2019. doi:10.1007/978-3-030-11245-5_22.
 - [FCS⁺23] Shenghua Feng, Mingshuai Chen, Han Su, Benjamin Lucien Kaminski, Joost-Pieter Katoen, and Naijun Zhan. Lower bounds for possibly divergent probabilistic programs. *Proc. ACM Program. Lang.*, 7(OOPSLA1):696–726, 2023. doi:10.1145/3586051.

- [FFH15] Luis María Ferrer Fioriti and Holger Hermanns. Probabilistic termination: Soundness, completeness, and compositionality. In *Proc. POPL '15*, pages 489–501, 2015. doi:10.1145/2676726.2677001.
- [FG17] Florian Frohn and Jürgen Giesl. Analyzing runtime complexity via innermost runtime complexity. In *Proc. LPAR '17*, EPIc 46, pages 249–228, 2017. doi:10.29007/1nbh.
- [FGP⁺09] Carsten Fuhs, Jürgen Giesl, Martin Plücker, Peter Schneider-Kamp, and Stephan Falke. Proving termination of integer term rewriting. In *Proc. RTA '09*, LNCS 5595, pages 32–47, 2009. doi:10.1007/978-3-642-02348-4_3.
- [Fuh19] Carsten Fuhs. Transforming derivational complexity of term rewriting to runtime complexity. In *Proc. FroCoS '19*, LNCS 11715, pages 348–364, 2019. doi:10.1007/978-3-030-29007-8_20.
- [GAB⁺17] Jürgen Giesl, Cornelius Aschermann, Marc Brockschmidt, Fabian Emmes, Florian Frohn, Carsten Fuhs, Jera Hensel, Carsten Otto, Martin Plücker, Peter Schneider-Kamp, Thomas Ströder, Stephanie Swiderski, and René Thiemann. Analyzing program termination and complexity automatically with AProVE. *J. Autom. Reason.*, 58(1):3–31, 2017. doi:10.1007/s10817-016-9388-y.
- [GGH19] Jürgen Giesl, Peter Giesl, and Marcel Hark. Computing expected runtimes for constant probability programs. In *Proc. CADE '19*, LNCS 11716, pages 269–286, 2019. doi:10.1007/978-3-030-29436-6_16.
- [GHN14] Andrew D. Gordon, Thomas A. Henzinger, Aditya V. Nori, and Sriram K. Rajamani. Probabilistic programming. In *Proc. FOSE '14*, pages 167–181, 2014. doi:10.1145/2593882.2593900.
- [GL20] Raúl Gutiérrez and Salvador Lucas. MU-TERM: Verify termination properties automatically (system description). In *Proc. IJCAR '20*, LNCS 12167, pages 436–447, 2020. doi:10.1007/978-3-030-51054-1_28.
- [Gra95] Bernhard Gramlich. Abstract relations between restricted termination and confluence properties of rewrite systems. *Fundam. Informaticae*, 24:2–23, 1995. doi:10.3233/FI-1995-24121.
- [Gra12] Bernhard Gramlich. Modularity in term rewriting revisited. *Theor. Comput. Sci.*, 464:3–19, 2012. doi:10.1016/J.TCS.2012.09.008.
- [GTSKF06] Jürgen Giesl, René Thiemann, Peter Schneider-Kamp, and Stephan Falke. Mechanizing and improving dependency pairs. *J. Autom. Reason.*, 37(3):155–203, 2006. doi:10.1007/s10817-006-9057-7.
- [HFC18] Mingzhang Huang, Hongfei Fu, and Krishnendu Chatterjee. New approaches for almost-sure termination of probabilistic programs. In *Proc. APLAS '18*, LNCS 11275, pages 181–201, 2018. doi:10.1007/978-3-030-02768-1_11.
- [HFCG19] Mingzhang Huang, Hongfei Fu, Krishnendu Chatterjee, and Amir Kafshdar Goharshady. Modular verification for almost-sure termination of probabilistic programs. *Proc. ACM Program. Lang.*, 3(OOPSLA), 2019. doi:10.1145/3360555.
- [HKGK20] Marcel Hark, Benjamin Lucien Kaminski, Jürgen Giesl, and Joost-Pieter Katoen. Aiming low is harder: induction for lower bounds in probabilistic program verification. *Proc. ACM Program. Lang.*, 4(POPL), 2020. doi:10.1145/3371105.
- [HL89] Dieter Hofbauer and Clemens Lautemann. Termination proofs and the length of derivations (preliminary version). In *Proc. RTA '89*, LNCS 355, pages 167–177,

1989. doi:10.1007/3-540-51081-8_107.
- [HM08] Nao Hirokawa and Georg Moser. Automated complexity analysis based on the dependency pair method. In *Proc. IJCAR '08*, LNCS 5195, pages 364–379, 2008. doi:10.1007/978-3-540-71070-7_32.
 - [KB70] Donald E. Knuth and Peter B. Bendix. Simple word problems in universal algebras. In *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon, 1970. doi:10.1016/B978-0-08-012975-4.50028-X.
 - [KDG24] Jan-Christoph Kassing, Stefan Dollase, and Jürgen Giesl. A complete dependency pair framework for almost-sure innermost termination of probabilistic term rewriting. In *Proc. FLOPS '24*, LNCS 14659, 2024. doi:10.1007/978-981-97-2300-3_4.
 - [KFG24] Jan-Christoph Kassing, Florian Frohn, and Jürgen Giesl. From innermost to full almost-sure termination of probabilistic term rewriting. In *Proc. FoSSaCS '24*, LNCS 14575, pages 206–228, 2024. doi:10.1007/978-3-031-57231-9_10.
 - [KG23a] Jan-Christoph Kassing and Jürgen Giesl. Proving almost-sure innermost termination of probabilistic term rewriting using dependency pairs. In *Proc. CADE '23*, LNCS 14132, pages 344–364, 2023. doi:10.1007/978-3-031-38499-8_20.
 - [KG23b] Jan-Christoph Kassing and Jürgen Giesl. Proving almost-sure innermost termination of probabilistic term rewriting using dependency pairs. *CoRR*, abs/2305.11741, 2023. Long version of [KG23a]. doi:10.48550/arXiv.2305.11741.
 - [KG24] Jan-Christoph Kassing and Jürgen Giesl. Annotated dependency pairs for full almost-sure termination of probabilistic term rewriting. In *Principles of Verification*, LNCS 15260, pages 339–366, 2024. doi:10.1007/978-3-031-75783-9_14.
 - [KKM19] Benjamin Lucien Kaminski, Joost-Pieter Katoen, and Christoph Matheja. On the hardness of analyzing probabilistic programs. *Acta Informatica*, 56(3):255–285, 2019. doi:10.1007/s00236-018-0321-1.
 - [KKM20] Benjamin Lucien Kaminski, Joost-Pieter Katoen, and Christoph Matheja. Expected runtime analysis by program verification. In Gilles Barthe, Joost-Pieter Katoen, and Alexandra Silva, editors, *Foundations of Probabilistic Programming*, page 185–220. Cambridge University Press, 2020. doi:10.1017/9781108770750.007.
 - [KKMO18] Benjamin Lucien Kaminski, Joost-Pieter Katoen, Christoph Matheja, and Federico Olmedo. Weakest precondition reasoning for expected runtimes of randomized algorithms. *J. ACM*, 65(5), 2018. doi:10.1145/3208102.
 - [KLG20] Naoki Kobayashi, Ugo Dal Lago, and Charles Grellois. On the termination problem for probabilistic higher-order recursive programs. *Log. Methods Comput. Sci.*, 16(4), 2020. doi:10.23638/LMCS-16(4:2)2020.
 - [KMMM10] Joost-Pieter Katoen, Annabelle McIver, Larissa Meinicke, and Carroll C. Morgan. Linear-invariant generation for probabilistic programs: Automated support for proof-based methods. In *Proc. SAS '10*, LNCS 6337, pages 390–406, 2010. doi:10.1007/978-3-642-15769-1_24.
 - [KMS⁺22] Andrey Kofnov, Marcel Moosbrugger, Miroslav Stankovic, Ezio Bartocci, and Efstathia Bura. Moment-based invariants for probabilistic loops with non-polynomial assignments. In *Proc. QEST '22*, LNCS 13479, pages 3–25, 2022.

- doi:10.1007/978-3-031-16336-4_1.
- [KN13] Cynthia Kop and Naoki Nishida. Term rewriting with logical constraints. In *Proc. FroCoS '13*, LNCS 8152, pages 343–358, 2013. doi:10.1007/978-3-642-40885-4_24.
 - [KO21] Andrew Kenyon-Roberts and C.-H. Luke Ong. Supermartingales, ranking functions and probabilistic lambda calculus. In *Proc. LICS '21*, 2021. doi:10.1109/LICS52264.2021.9470550.
 - [Kri00] M. R. K. Krishna Rao. Some characteristics of strong innermost normalization. *Theor. Comput. Sc.*, 239:141–164, 2000. doi:10.1016/S0304-3975(99)00215-7.
 - [KSG25] Jan-Christoph Kassing, Leon Spitzer, and Jürgen Giesl. Dependency pairs for expected innermost runtime complexity and strong almost-sure termination of probabilistic term rewriting. In *Proc. PPDP '25*, 2025. To appear. Long version available at *CoRR*, abs/2507.12918. doi:10.48550/arXiv.2507.12918.
 - [KSZM09] Martin Korp, Christian Sternagel, Harald Zankl, and Aart Middeldorp. Tyrolean Termination Tool 2. In *Proc. RTA '09*, LNCS 5595, pages 295–304, 2009. doi:10.1007/978-3-642-02348-4_21.
 - [KUH19] Satoshi Kura, Natsuki Urabe, and Ichiro Hasuo. Tail probabilities for randomized program runtimes via martingales for higher moments. In *Proc. TACAS '19*, LNCS 11428, pages 135–153, 2019. doi:10.1007/978-3-030-17465-1_8.
 - [LFR21] Ugo Dal Lago, Claudia Faggian, and Simona Ronchi Della Rocca. Intersection types and (positive) almost-sure termination. *Proc. ACM Program. Lang.*, 5(POPL), 2021. doi:10.1145/3434313.
 - [LG19] Ugo Dal Lago and Charles Grellois. Probabilistic termination by monadic affine sized typing. *ACM Trans. Program. Lang. Syst.*, 41(2), 2019. doi:10.1145/3293605.
 - [LJB01] Chin Soon Lee, Neil D. Jones, and Amir M. Ben-Amram. The size-change principle for program termination. In *Proc. POPL '01*, pages 81–92, 2001. doi:10.1145/360204.360210.
 - [LMG24] Nils Lommen, Éléanore Meyer, and Jürgen Giesl. Control-flow refinement for complexity analysis of probabilistic programs in KoAT (short paper). In *Proc. IJCAR '24*, LNCS 14739, pages 233–243, 2024. doi:10.1007/978-3-031-63498-7_14.
 - [LMO22] Guanyan Li, Andrzej S. Murawski, and Luke Ong. Probabilistic verification beyond context-freeness. In *Proc. LICS '22*, 2022. doi:10.1145/3531130.3533351.
 - [LMZ22] Lorenz Leutgeb, Georg Moser, and Florian Zuleger. Automated expected amortised cost analysis of probabilistic data structures. In *Proc. CAV '22*, LNCS 13372, pages 70–91, 2022. doi:10.1007/978-3-031-13188-2_4.
 - [MAM⁺23] César A. Muñoz, Mauricio Ayala-Rincón, Mariano M. Moscato, Aaron Dutle, Anthony J. Narkawicz, Ariane Alves Almeida, Andréia B. Avelar da Silva, and Thiago Mendonça Ferreira Ramos. Formal verification of termination criteria for first-order recursive functions. *J. Autom. Reason.*, 67(4):40, 2023. doi:10.1007/S10817-023-09669-Z.
 - [MBKK21] Marcel Moosbrugger, Ezio Bartocci, Joost-Pieter Katoen, and Laura Kovács. Automated termination analysis of polynomial probabilistic programs. In *Proc. ESOP '21*, LNCS 12648, pages 491–518, 2021. doi:10.1007/

- 978-3-030-72019-3_18.
- [MHG21] Fabian Meyer, Marcel Hark, and Jürgen Giesl. Inferring expected runtimes of probabilistic integer programs using expected sizes. In *Proc. TACAS '21*, LNCS 12651, pages 250–269, 2021. doi:10.1007/978-3-030-72016-2_14.
 - [MMKK18] Annabelle McIver, Carroll Morgan, Benjamin Lucien Kaminski, and Joost-Pieter Katoen. A new proof rule for almost-sure termination. *Proc. ACM Program. Lang.*, 2(POPL), 2018. doi:10.1145/3158121.
 - [MS24] Rupak Majumdar and V. R. Sathiyararayanan. Positive almost-sure termination: Complexity and proof rules. *Proc. ACM Program. Lang.*, 8(POPL):1089–1117, 2024. doi:10.1145/3632879.
 - [MS25] Rupak Majumdar and V. R. Sathiyararayanan. Sound and complete proof rules for probabilistic termination. *Proc. ACM Program. Lang.*, 9(POPL):1871–1902, 2025. doi:10.1145/3704899.
 - [MSBK22] Marcel Moosbrugger, Miroslav Stankovic, Ezio Bartocci, and Laura Kovács. This is the moment for probabilistic loops. *Proc. ACM Program. Lang.*, 6(OOPSLA2):1497–1525, 2022. doi:10.1145/3563341.
 - [MV06] Panagiotis Manolios and Daron Vroon. Termination analysis with calling context graphs. In *Proc. CAV '06*, LNCS 4144, pages 401–414, 2006. doi:10.1007/11817963_36.
 - [NCH18] Van Chan Ngo, Quentin Carbonneaux, and Jan Hoffmann. Bounded expectations: Resource analysis for probabilistic programs. In *Proc. PLDI '18*, pages 496–512, 2018. doi:10.1145/3192366.3192394.
 - [NEG13] Lars Noschinski, Fabian Emmes, and Jürgen Giesl. Analyzing innermost runtime complexity of term rewriting by dependency pairs. *J. Autom. Reason.*, 51:27–56, 2013. doi:10.1007/978-3-642-22438-6_32.
 - [New42] M. H. A. Newman. On theories with a combinatorial definition of equivalence. *Annals of Mathematics*, 43(2):223–242, 1942. URL: <http://www.ens-lyon.fr/LIP/REWRITING/TERMINATION/NEWMAN/Newman.pdf>.
 - [NYC⁺25] Sergei Novozhilov, Mingqi Yang, Mingshuai Chen, Zhiyang Li, and Jianwei Yin. On the almost-sure termination of probabilistic counter programs. In *Proc. CAV '25*, LNCS 15932, pages 82–104, 2025. doi:10.1007/978-3-031-98679-6_4.
 - [O'D77] M. J. O'Donnell. *Computing in Systems Described by Equations*. LNCS 58. 1977. doi:10.1007/3-540-08531-9.
 - [RBG24] Vineet Rajani, Gilles Barthe, and Deepak Garg. A modal type theory of expected cost in higher-order probabilistic programs. *Proc. ACM Program. Lang.*, 8(OOPSLA2):389–414, 2024. doi:10.1145/3689725.
 - [Sah78] Nasser Saheb-Djahromi. Probabilistic LCF. In *Proc. MFCS '78*, LNCS 64, pages 442–451, 1978. doi:10.1007/3-540-08921-7_92.
 - [SBK⁺23] Philipp Schröer, Kevin Batz, Benjamin Lucien Kaminski, Joost-Pieter Katoen, and Christoph Matheja. A deductive verification infrastructure for probabilistic programs. *Proc. ACM Program. Lang.*, 7(OOPSLA2):2052–2082, 2023. doi:10.1145/3622870.
 - [TG05] René Thiemann and Jürgen Giesl. The size-change principle and dependency pairs for termination of term rewriting. *Appl. Algebra Eng. Commun. Comput.*, 16(4):229–270, 2005. doi:10.1007/S00200-005-0179-7.

- [Toy87a] Y. Toyama. Counterexamples to the termination for the direct sum of term rewriting systems. *Inf. Proc. Lett.*, 25:141–143, 1987. doi:10.1016/0020-0190(87)90122-0.
- [Toy87b] Yoshihito Toyama. On the Church-Rosser property for the direct sum of term rewriting systems. *J. ACM*, 34:128–143, January 1987. doi:10.1145/7531.7534.
- [TS09] René Thiemann and Christian Sternagel. Certification of termination proofs using CeTA. In *Proc. TPHOLs '09*, LNCS 5674, pages 452–468, 2009. doi:10.1007/978-3-642-03359-9_31.
- [WFG⁺19] Peixin Wang, Hongfei Fu, Amir Kafshdar Goharshady, Krishnendu Chatterjee, Xudong Qin, and Wenjun Shi. Cost analysis of nondeterministic probabilistic programs. In *Proc. PLDI '19*, pages 204–220, 2019. doi:10.1145/3314221.3314581.
- [WK23] Tobias Winkler and Joost-Pieter Katoen. On certificates, expected runtimes, and termination in probabilistic pushdown automata. In *Proc. LICS '23*, 2023. doi:10.1109/LICS56636.2023.10175714.
- [WKH20] Di Wang, David M. Kahn, and Jan Hoffmann. Raising expectations: Automating expected cost analysis with types. *Proc. ACM Program. Lang.*, 4(ICFP), 2020. doi:10.1145/3408992.
- [YKS14] Akihisa Yamada, Keiichirou Kusakari, and Toshiki Sakabe. Nagoya Termination Tool. In *Proc. RTA-TLCA '14*, LNCS 8560, pages 466–475, 2014. doi:10.1007/978-3-319-08918-8_32.

APPENDIX A. MISSING PROOFS

In this appendix, we present all missing proofs for our new contributions and observations. Most of our proofs use \rightarrow -RSTs instead of \Rightarrow -rewrite sequences. Therefore, in App. A.1 we start with the formal definitions for all required notions via RSTs (where some of them were already mentioned in the main part of the paper). Then, in App. A.2 we give the missing proofs for the theorems and lemmas from Section 4 and Section 5, concerning the relation between different rewrite strategies. In App. A.3, we prove the results regarding modularity and signature extensions from Section 7.

A.1. Characterization via RSTs. We start with the formal definition of RSTs.

Definition A.1 (Rewrite Sequence Tree (RST)). Let $\rightarrow \subseteq \mathcal{T} \times \text{FDist}(\mathcal{T})$. A \rightarrow -rewrite sequence tree (\rightarrow -RST) \mathfrak{T} is a labeled tree $\mathfrak{T} = (N, E, L)$ such that

- (1) $N \neq \emptyset$ is a possibly infinite set of nodes and $E \subseteq N \times N$ is a set of directed edges, such that (N, E) is a (possibly infinite) directed tree where $vE = \{w \mid (v, w) \in E\}$ is finite for every $v \in N$.
- (2) $L : N \rightarrow (0, 1] \times \mathcal{T}$ labels every node v by a probability p_v and a term t_v . For the root $v \in N$ of the tree, we have $p_v = 1$.
- (3) For all $v \in N$: If $vE = \{w_1, \dots, w_k\}$, then $t_v \rightarrow \{\frac{p_{w_1}}{p_v} : t_{w_1}, \dots, \frac{p_{w_k}}{p_v} : t_{w_k}\}$.

Leaf denotes the set of leaves of the RST and for a node $x \in N$, $d(x)$ denotes the depth of node x in \mathfrak{T} . Here, the root has depth 0. We say that \mathfrak{T} is *fully evaluated* if for every $v \in \text{Leaf}$ the corresponding term t_v is a normal form w.r.t. \rightarrow , i.e., $t_v \in \text{NF}_{\rightarrow}$.

Whenever the referenced RST is ambiguous, we will explicitly indicate the corresponding tree. For instance, for the probability p_v of the node $v \in N$ of some RST $\mathfrak{T} = (N, E, L)$, we may also write $p_v^{\mathfrak{T}}$, and $N^{\mathfrak{T}} = N$ is the set of nodes of the tree \mathfrak{T} .

Definition A.2 ($|\mathfrak{T}|$, Convergence Probability). Let $\rightarrow \subseteq \mathcal{T} \times \text{FDist}(\mathcal{T})$. For any \rightarrow -RST \mathfrak{T} we define $|\mathfrak{T}| = \sum_{v \in \text{Leaf}} p_v$ and say that the RST \mathfrak{T} *converges with probability* $|\mathfrak{T}|$.

It is now easy to observe that we have AST_{\rightarrow} (i.e., for all \Rightarrow -rewrite sequences $(\mu_n)_{n \in \mathbb{N}}$ we have $\lim_{n \rightarrow \infty} |\mu_n|_{\rightarrow} = 1$) iff for all \rightarrow -RSTs \mathfrak{T} we have $|\mathfrak{T}| = 1$. To see this, note that every infinite \Rightarrow -rewrite sequence $(\mu_n)_{n \in \mathbb{N}}$ that begins with a single start term (i.e., $\mu_0 = \{1 : t\}$) can be represented by an \rightarrow -RST \mathfrak{T} that is fully evaluated such that $\lim_{n \rightarrow \infty} |\mu_n|_{\rightarrow} = |\mathfrak{T}|$ and vice versa. So AST_{\rightarrow} holds iff all fully evaluated \rightarrow -RSTs converge with probability 1. Furthermore, note that for every \rightarrow -RST \mathfrak{T} , there exists a fully evaluated \rightarrow -RST \mathfrak{T}' such that $|\mathfrak{T}| \geq |\mathfrak{T}'|$. To get from \mathfrak{T} to \mathfrak{T}' we can simply perform arbitrary (possibly infinitely many) rewrite steps at the leaves that are not in normal form to fully evaluate the tree.

It remains to prove that it suffices to only regard \Rightarrow -rewrite sequences that start with a single start term.

Lemma A.3 (Single Start Terms Suffice for AST_{\rightarrow}). *Let $\rightarrow \subseteq \mathcal{T} \times \text{FDist}(\mathcal{T})$. If AST_{\rightarrow} does not hold, then there exists an infinite \Rightarrow -rewrite sequence $(\mu'_n)_{n \in \mathbb{N}}$ with a single start term, i.e., $\mu'_0 = \{1 : t\}$, that converges with probability < 1 .*

Proof. We prove the converse. Assume that all infinite \Rightarrow -rewrite sequences $(\mu'_n)_{n \in \mathbb{N}}$ with a single start term converge with probability 1. We prove that then every infinite \Rightarrow -rewrite sequence converges with probability 1, hence we have AST_{\rightarrow} .

Let $(\mu_n)_{n \in \mathbb{N}}$ be an infinite \Rightarrow -rewrite sequence. Suppose that we have $\mu_0 = \{p_1 : t_1, \dots, p_k : t_k\}$. Let $(\mu_n^{(j)})_{n \in \mathbb{N}}$ with $\mu_0^{(j)} = \{1 : t_j\}$ denote the infinite \Rightarrow -rewrite sequence that uses the same rules as $(\mu_n)_{n \in \mathbb{N}}$ does on the term t_j for every $1 \leq j \leq k$. We obtain

$$\begin{aligned} \lim_{n \rightarrow \infty} |\mu_n|_{\rightarrow} &= \lim_{n \rightarrow \infty} \sum_{j=1}^k p_j \cdot |\mu_n^{(j)}|_{\rightarrow} = \sum_{j=1}^k \lim_{n \rightarrow \infty} p_j \cdot |\mu_n^{(j)}|_{\rightarrow} \\ &= \sum_{j=1}^k p_j \cdot \lim_{n \rightarrow \infty} |\mu_n^{(j)}|_{\rightarrow} = \sum_{j=1}^k p_j \cdot 1 = \sum_{j=1}^k p_j = 1 \end{aligned} \quad \square$$

We now obtain the following corollary.

Corollary A.4 (Characterizing $\mathbf{AST}_{\rightarrow}$ with RSTs). *Let $\rightarrow \subseteq \mathcal{T} \times \text{FDist}(\mathcal{T})$. Then $\mathbf{AST}_{\rightarrow}$ holds iff for all \rightarrow -RSTs \mathfrak{T} we have $|\mathfrak{T}| = 1$. This is equivalent to the requirement that for all fully evaluated \rightarrow -RSTs \mathfrak{T} we have $|\mathfrak{T}| = 1$. Moreover, $\mathbf{wAST}_{\rightarrow}$ holds iff for every term $t \in \mathcal{T}$ there exists a fully evaluated \rightarrow -RST \mathfrak{T} whose root is labeled with $(1 : t)$ such that $|\mathfrak{T}| = 1$.*

Next, we recapitulate how $\mathbf{PAST}_{\rightarrow}$ and $\mathbf{SAST}_{\rightarrow}$ can be formulated in terms of RSTs.

Corollary A.5 (Characterizing $\mathbf{PAST}_{\rightarrow}$ with RSTs). *Let $\rightarrow \subseteq \mathcal{T} \times \text{FDist}(\mathcal{T})$, and \mathfrak{T} be an \rightarrow -RST. By*

$$\text{edl}(\mathfrak{T}) = \sum_{x \in N \setminus \text{Leaf}} p_x = \lim_{n \rightarrow \infty} \sum_{\substack{x \in N \setminus \text{Leaf} \\ d(x) \leq n}} p_x$$

we define the expected derivation length of \mathfrak{T} . We have $\mathbf{PAST}_{\rightarrow}$ iff $\text{edl}(\mathfrak{T})$ is finite for every \rightarrow -RST \mathfrak{T} . Similarly, $\mathbf{wPAST}_{\rightarrow}$ holds iff for every term t there exists a fully evaluated \rightarrow -RST \mathfrak{T} whose root is labeled with $(1 : t)$ such that $\text{edl}(\mathfrak{T})$ is finite.

Corollary A.6 (Characterizing $\mathbf{SAST}_{\rightarrow}$ with RSTs). *Let $\rightarrow \subseteq \mathcal{T} \times \text{FDist}(\mathcal{T})$. We have $\mathbf{SAST}_{\rightarrow}$ iff $\sup\{\text{edl}(\mathfrak{T}) \mid \mathfrak{T} \text{ is an } \rightarrow\text{-RST whose root is labeled with } (1 : t)\}$ is finite for all $t \in \mathcal{T}$.*

Both of these corollaries are again easy to observe, similar to the characterization of $\mathbf{AST}_{\rightarrow}$ with RSTs in Cor. A.4. First note that every infinite \Rightarrow -rewrite sequence $\vec{\mu} = (\mu_n)_{n \in \mathbb{N}}$ that begins with a single start term can be represented by an infinite \rightarrow -RST \mathfrak{T} that is fully evaluated such that $\text{edl}(\vec{\mu}) = \sum_{n=0}^{\infty} (1 - |\mu_n|_{\rightarrow}) = \sum_{x \in N \setminus \text{Leaf}} p_x^{\mathfrak{T}} = \text{edl}(\mathfrak{T})$. The reason is that whenever we reach a normal form after n steps, this is both a normal form in μ_n and a leaf at depth n of the RST with the same probability. Otherwise, if we have a term t in $\text{Supp}(\mu_n)$ that is not in normal form, then there exists a (unique) inner node in \mathfrak{T} at depth n with the same probability. Note that we do not need a version of Lemma A.3 for $\mathbf{PAST}_{\rightarrow}$ and $\mathbf{SAST}_{\rightarrow}$, since both of them already consider single start terms.

A.2. Proofs for Section 4 and Section 5. In this subsection, we present all missing proofs on the relation between different restricted forms of probabilistic termination. To this end, we first recapitulate the notion of a *rewrite sequence subtree* from [KG23b].

Definition A.7 (Rewrite Sequence Subtree). Let $\rightarrow \subseteq \mathcal{T} \times \text{FDist}(\mathcal{T})$, and let $\mathfrak{T} = (N, E, L)$ be an \rightarrow -RST. Let $W \subseteq N$ be non-empty, weakly connected, and for all $x \in W$ we have $xE \cap W = \emptyset$ or $xE \cap W = xE$. Then, we define the \rightarrow -rewrite sequence subtree (or simply subtree) $\mathfrak{T}[W]$ by $\mathfrak{T}[W] = (W, E \cap (W \times W), L^W)$. Let $w \in W$ be the root of $\mathfrak{T}[W]$. To ensure that the root of our subtree has the probability 1 again, we use the labeling $L^W(x) = (\frac{p_x^{\mathfrak{T}}}{p_w^{\mathfrak{T}}} : t_x^{\mathfrak{T}})$ for all nodes $x \in W$.

The property of being non-empty and weakly connected ensures that the resulting graph $(W, E \cap (W \times W))$ is a tree again. The property that we either have $xE \cap W = \emptyset$ or $xE \cap W = xE$ ensures that the sum of probabilities for the successors of a node x is equal to the probability for the node x itself.

To prove the theorems in Section 4 and Section 5 we only have to prove the corresponding lemmas. Then the theorems are a direct consequence, similar to the proof of Thm. 4.2 given in Section 4. As mentioned earlier, we will always prove the lemmas using rewrite sequence trees.

Lemma 4.3 (From Innermost to Full Rewriting). *If a PTRS \mathcal{P} is OR and RL (i.e., NO and linear), then for every infinite $\xrightarrow{\mathcal{P}}$ -rewrite sequence $\vec{\mu} = (\mu_n)_{n \in \mathbb{N}}$ there exists an infinite $\xrightarrow{\mathcal{P}}$ -rewrite sequence $\vec{\nu} = (\nu_n)_{n \in \mathbb{N}}$ such that*

- (i) $\lim_{n \rightarrow \infty} |\mu_n|_{\mathcal{P}} \geq \lim_{n \rightarrow \infty} |\nu_n|_{\mathcal{P}}$
- (ii) $\text{edl}(\vec{\mu}) \leq \text{edl}(\vec{\nu})$

Proof. Let \mathcal{P} be a PTRS that is non-overlapping and linear. Furthermore, let \mathfrak{T} be an $\xrightarrow{\mathcal{P}}$ -RST. We create a new $\xrightarrow{\mathcal{P}}$ -RST $\mathfrak{T}^{(\infty)}$ such that $|\mathfrak{T}^{(\infty)}| \leq |\mathfrak{T}|$ and $\text{edl}(\mathfrak{T}^{(\infty)}) \geq \text{edl}(\mathfrak{T})$. W.l.o.g., at least one rewrite step in \mathfrak{T} is performed at some node x with a redex that is not an innermost redex (otherwise we can use $\mathfrak{T}^{(\infty)} = \mathfrak{T}$). The core steps of the proof are the following:

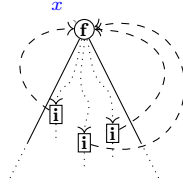
1. We iteratively move innermost rewrite steps to a higher position in the tree using a construction $\Phi(\cdot)$. The limit of this iteration, namely $\mathfrak{T}^{(\infty)}$, is an *innermost* $\xrightarrow{\mathcal{P}}$ -RST with $|\mathfrak{T}^{(\infty)}| \leq |\mathfrak{T}|$. For each step of the iteration:
 - 1.1 We formally define the construction $\Phi(\cdot)$ that replaces a certain subtree \mathfrak{T}_x by a new subtree $\Phi(\mathfrak{T}_x)$, by moving an innermost rewrite step to the root of $\Phi(\mathfrak{T}_x)$.
 - 1.2 We show $|\Phi(\mathfrak{T}_x)| = |\mathfrak{T}_x|$.
 - 1.3 We show that $\Phi(\mathfrak{T}_x)$ is indeed a valid RST.
2. We show that the same construction also guarantees $\text{edl}(\mathfrak{T}^{(\infty)}) \geq \text{edl}(\mathfrak{T})$. For this:
 - 2.1 We prove $\text{edl}(\mathfrak{T}) \leq \text{edl}(\mathfrak{T}^{(1)}) \leq \text{edl}(\mathfrak{T}^{(2)}) \leq \dots$
 - 2.2 We prove $\text{edl}(\mathfrak{T}) \leq \text{edl}(\mathfrak{T}^{(\infty)})$.

1. We iteratively move innermost rewrite steps to a higher position.

In \mathfrak{T} there exists at least one rewrite step performed at some node x , which is not an innermost rewrite step. Furthermore, we can assume that this is the first such rewrite step in the path from the root to the node x and that x is a node of minimum depth¹⁰ with this property. Let \mathfrak{T}_x be the subtree that starts at node x , i.e., $\mathfrak{T}_x = \mathfrak{T}[xE^*]$, where xE^* is the set of all reachable nodes (via the edge relation) from x . We then construct a new tree $\Phi(\mathfrak{T}_x)$ such that $|\Phi(\mathfrak{T}_x)| = |\mathfrak{T}_x|$, where we use an innermost rewrite step at the root node x instead of the old one, i.e., we pushed the first non-innermost rewrite step deeper into the tree. This construction only works because \mathcal{P} is non-overlapping and linear.¹¹ Then, by replacing the subtree \mathfrak{T}_x with the new tree $\Phi(\mathfrak{T}_x)$ in \mathfrak{T} , we obtain an $\xrightarrow{\mathcal{P}}$ -RST $\mathfrak{T}^{(1)}$ with $|\mathfrak{T}^{(1)}| = |\mathfrak{T}|$, where we use an innermost rewrite step at node x instead of the old rewrite step,

¹⁰If we allowed rules with infinite support, then there could be infinitely many nodes at minimum depth. Hence, then one would have to use a more elaborate enumeration of all nodes where a non-innermost step is performed.

¹¹For the construction, non-overlappingness is essential (while a related construction could also be defined without linearity). However, linearity is needed to ensure that the probability of termination in the new tree is not larger than in the original one.

Figure 5: Tree \mathfrak{T}_x in the Construction of $\Phi(-)$

as desired. We can then do such a replacement iteratively for every use of a non-innermost rewrite step, i.e., we again replace the first non-innermost rewrite step in $\mathfrak{T}^{(1)}$ to obtain $\mathfrak{T}^{(2)}$ with $|\mathfrak{T}^{(2)}| = |\mathfrak{T}^{(1)}|$, and so on. In the end, the limit of all these RSTs $\lim_{i \rightarrow \infty} \mathfrak{T}^{(i)}$ is an $\rightarrow_{\mathcal{P}}$ -RST, that we denote by $\mathfrak{T}^{(\infty)}$ such that $|\mathfrak{T}^{(\infty)}| \leq |\mathfrak{T}|$. So while the termination probability remains the same in each step, it can decrease in the limit.¹²

To see that $\mathfrak{T}^{(\infty)}$ is indeed a valid $\rightarrow_{\mathcal{P}}$ -RST, note that in every iteration of the construction we turn a non-innermost rewrite step at minimum depth into an innermost one. Hence, for every depth H of the tree, we eventually turned every non-innermost rewrite step up to depth H into an innermost one. So the construction will not change the tree above depth H anymore,¹³ i.e., there exists an m_H such that $\mathfrak{T}^{(\infty)}$ and $\mathfrak{T}^{(i)}$ are the same trees up to depth H for all $i \geq m_H$. This means that the sequence $\lim_{i \rightarrow \infty} \mathfrak{T}^{(i)}$ really converges into an $\rightarrow_{\mathcal{P}}$ -RST.

Next, we want to prove that we have $|\mathfrak{T}^{(\infty)}| \leq |\mathfrak{T}|$. By induction on n one can prove that $|\mathfrak{T}^{(i)}| = |\mathfrak{T}|$ for all $1 \leq i \leq n$, since we have $|\mathfrak{T}^{(i)}| = |\mathfrak{T}^{(i-1)}|$ for all $i \geq 2$ and $|\mathfrak{T}^{(1)}| = |\mathfrak{T}|$. Assume for a contradiction that $|\mathfrak{T}^{(\infty)}| > |\mathfrak{T}|$. Then there exists a depth $H \in \mathbb{N}$ such that $\sum_{x \in \text{Leaf}^{\mathfrak{T}^{(\infty)}}, d^{\mathfrak{T}^{(\infty)}}(x) \leq H} p_x > |\mathfrak{T}|$. Again, let $m_H \in \mathbb{N}$ such that $\mathfrak{T}^{(\infty)}$ and $\mathfrak{T}^{(m_H)}$ are the same trees up to depth H . But this would mean that $|\mathfrak{T}^{(m_H)}| \geq \sum_{x \in \text{Leaf}^{\mathfrak{T}^{(m_H)}}, d^{\mathfrak{T}^{(m_H)}}(x) \leq H} p_x^{\mathfrak{T}^{(m_H)}} = \sum_{x \in \text{Leaf}^{\mathfrak{T}^{(\infty)}}, d^{\mathfrak{T}^{(\infty)}}(x) \leq H} p_x^{\mathfrak{T}^{(\infty)}} > |\mathfrak{T}|$, which is a contradiction to $|\mathfrak{T}^{(m_H)}| = |\mathfrak{T}|$.

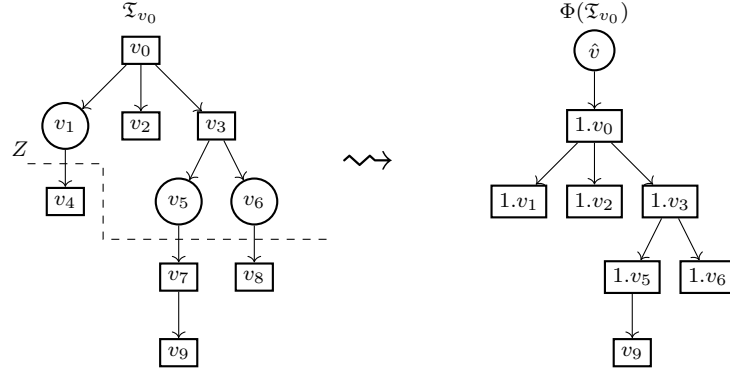
1.1 Construction of $\Phi(-)$

It remains to define the construction $\Phi(-)$ mentioned above. Let \mathfrak{T}_x be an $\rightarrow_{\mathcal{P}}$ -RST that performs a non-innermost rewrite step at the root node x . This step has the form $t_x^{\mathfrak{T}_x} \rightarrow_{\mathcal{P}} \{p_{y_1}^{\mathfrak{T}_x} : t_{y_1}^{\mathfrak{T}_x}, \dots, p_{y_k}^{\mathfrak{T}_x} : t_{y_k}^{\mathfrak{T}_x}\}$ using the rule $\bar{\ell} \rightarrow \{\bar{p}_1 : \bar{r}_1, \dots, \bar{p}_k : \bar{r}_k\}$, the substitution $\bar{\sigma}$, and the position $\bar{\pi}$ such that $t_x^{\mathfrak{T}_x}|_{\bar{\pi}} = \bar{\ell}\bar{\sigma}$. Then we have $t_{y_j}^{\mathfrak{T}_x} = t_x^{\mathfrak{T}_x}[\bar{r}_j\bar{\sigma}]_{\bar{\pi}}$ for all $1 \leq j \leq k$.

Instead of applying a non-innermost rewrite step at the root x we want to directly apply an innermost rewrite step. Let τ be the position of some innermost redex in $t_x^{\mathfrak{T}_x}$ below $\bar{\pi}$. The construction creates a new $\rightarrow_{\mathcal{P}}$ -RST $\Phi(\mathfrak{T}_x) = (N', E', L')$ whose root is labeled with $(1 : t_x^{\mathfrak{T}_x})$ such that $|\Phi(\mathfrak{T}_x)| = |\mathfrak{T}_x|$, and that directly performs the first rewrite step at position τ in the original tree \mathfrak{T}_x (which is an innermost rewrite step) at the root of the tree, by pushing it from the original nodes in the tree \mathfrak{T}_x to the root of the new tree $\Phi(\mathfrak{T}_x)$. (It could also be that this innermost redex was never reduced in \mathfrak{T}_x .) This can be seen in the diagram in Fig. 5, which depicts the tree \mathfrak{T}_x . The boxes represent innermost

¹²As an example, consider a tree \mathfrak{T} which is just a finite path and its path length increases in each iteration by one. Then the limit $\mathfrak{T}^{(\infty)}$ is an infinite path and converges with probability 0, while $\mathfrak{T}, \mathfrak{T}^{(1)}, \dots$ all converge with probability 1.

¹³Again, for rules with infinite support, this would not hold anymore. Nevertheless, with a more elaborate enumeration, one should still be able to obtain a valid $\rightarrow_{\mathcal{P}}$ -RST in the limit.

Figure 6: Illustration of $\Phi(\mathfrak{T})$

rewrite steps at position τ and the dashed lines indicate that we push this rewrite step to the root. This push only results in the same convergence probability due to our restriction that \mathcal{P} is linear. However, since we are allowed to rewrite above τ in the original tree \mathfrak{T}_x , the actual position of the innermost redex that was originally at position τ might change during the application of a rewrite step. Hence, we recursively define the position $\varphi_\tau(v)$ that contains precisely this redex for each node v in \mathfrak{T}_x until we rewrite at this position. Initially, we have $\varphi_\tau(x) = \tau$. Whenever we have defined $\varphi_\tau(v)$ for some node v , and we have $t_v^{\mathfrak{T}_x} \xrightarrow{\mathcal{P}} \{p_{w_1}^{\mathfrak{T}_x} : t_{w_1}^{\mathfrak{T}_x}, \dots, p_{w_m}^{\mathfrak{T}_x} : t_{w_m}^{\mathfrak{T}_x}\}$ for the direct successors $vE = \{w_1, \dots, w_m\}$, using the rule $\ell \rightarrow \{p_1 : r_1, \dots, p_m : r_m\}$, the substitution σ , and the position π , we do the following: If $\varphi_\tau(v) = \pi$, meaning that we rewrite this innermost redex, then we set $\varphi_\tau(w_j) = \perp$ for all $1 \leq j \leq m$ to indicate that we have rewritten the innermost redex. If we have $\varphi_\tau(v) \perp \pi$, meaning that the rewrite step takes place on a position that is parallel to $\varphi_\tau(v)$, then we set $\varphi_\tau(w_j) = \varphi_\tau(v)$ for all $1 \leq j \leq m$, as the position of the innermost redex did not change. Otherwise, we have $\pi < \varphi_\tau(v)$ (since we cannot rewrite below $\varphi_\tau(v)$ as it is an innermost redex), and thus there exists a $\chi \in \mathbb{N}^+$ such that $\pi \cdot \chi = \varphi_\tau(v)$. Since the rules of \mathcal{P} are non-overlapping, the redex must be completely “inside” the used substitution σ , and we can find a position α_q of a variable q in ℓ and another position β such that $\chi = \alpha_q \cdot \beta$. Furthermore, since the rule is linear, q only occurs once in ℓ and at most once in r_j for all $1 \leq j \leq m$. If q occurs in r_j at a position ρ_q^j , then we set $\varphi_\tau(w_j) = \rho_q^j \cdot \beta$. Otherwise, we set $\varphi_\tau(w_j) = \top$ to indicate that the innermost redex was erased during the computation. Finally, if $\varphi_\tau(v) \in \{\perp, \top\}$, then we set $\varphi_\tau(w_j) = \varphi_\tau(v)$ for all $1 \leq j \leq m$ as well. So to summarize, $\varphi_\tau(v)$ is now either the position of the innermost redex in t_v , \top to indicate that the redex was erased, or \perp to indicate that we have rewritten the redex.

In Fig. 6, we illustrate the effect of Φ , where the original tree \mathfrak{T}_x with $x = v_0$ is on the left and $\Phi(\mathfrak{T}_x)$ is on the right. The circled nodes represent the nodes where we perform a rewrite step at position $\varphi_\tau(v)$. We now define the $\xrightarrow{\mathcal{P}}$ -RST $\Phi(\mathfrak{T}_x)$ whose root \hat{v} is labeled with $(1 : t_x^{\mathfrak{T}_x})$ and that directly performs the rewrite step $t_x^{\mathfrak{T}_x} = t_{\hat{v}}^{\Phi(\mathfrak{T}_x)} \xrightarrow{\mathcal{P}, \tau} \{\hat{p}_1 : t_{1.x}^{\Phi(\mathfrak{T}_x)}, \dots, \hat{p}_h : t_{h.x}^{\Phi(\mathfrak{T}_x)}\}$, with the rule $\hat{\ell} \rightarrow \{\hat{p}_1 : \hat{r}_1, \dots, \hat{p}_h : \hat{r}_h\} \in \mathcal{P}$, a substitution $\hat{\sigma}$, and the position τ , at the new root \hat{v} . Here, we wrote “ $\xrightarrow{\mathcal{P}, \tau}$ ” to make the position of the used redex explicit. We have $t_{\hat{v}}^{\Phi(\mathfrak{T}_x)}|_\tau = \hat{\ell}\hat{\sigma}$. Let Z be the set of all nodes z such that $\varphi_\tau(z) \neq \perp$, i.e., at node z we did not yet perform the rewrite step with the innermost redex. In the example we have

$Z = \{v_0, \dots, v_6\} \setminus \{v_4\}$. For each of these nodes $z \in Z$ and each $1 \leq e \leq h$, we create a new node $e.z \in N'$ with edges as in \mathfrak{T}_x for the nodes in Z , e.g., for the node $1.v_3$ we create edges to $1.v_5$ and $1.v_6$. Furthermore, we add the edges from the new root \hat{v} to the nodes $e.x$ for all $1 \leq e \leq h$. Note that x was the root in the tree \mathfrak{T}_x and has to be contained in Z . For example, for the node \hat{v} we create an edge to $1.v_0$. We define the labeling of the nodes in $\Phi(\mathfrak{T}_x)$ as follows for all nodes z in Z :

$$\begin{aligned} \text{(T-1)} \quad t_{e.z}^{\Phi(\mathfrak{T}_x)} &= t_z^{\mathfrak{T}_x}[\hat{r}_e \hat{\sigma}]_{\varphi_\tau(z)} \text{ if } \varphi_\tau(z) \in \mathbb{N}^* \text{ and } t_{e.z}^{\Phi(\mathfrak{T}_x)} = t_z^{\mathfrak{T}_x} \text{ if } \varphi_\tau(z) = \top \\ \text{(T-2)} \quad p_{e.z}^{\Phi(\mathfrak{T}_x)} &= p_z^{\mathfrak{T}_x} \cdot \hat{p}_e \end{aligned}$$

Now, for a leaf $e.z' \in N'$ either $z' \in N$ is also a leaf (e.g., node v_2 in our example) or we rewrite the innermost redex at position $\varphi_\tau(z')$ at node z' in \mathfrak{T}_x (e.g., node v_1 in our example). In the latter case, due to non-overlappingness, the same rule $\hat{\ell} \rightarrow \{\hat{p}_1 : \hat{r}_1, \dots, \hat{p}_h : \hat{r}_h\}$ and the same substitution $\hat{\sigma}$ were used. If we rewrite $t_{z'}^{\mathfrak{T}_x} \xrightarrow{\mathcal{P}, \varphi_\tau(z')} \{\hat{p}_1 : t_{w'_1}^{\mathfrak{T}_x}, \dots, \hat{p}_h : t_{w'_h}^{\mathfrak{T}_x}\}$, then we have $t_{w'_e}^{\mathfrak{T}_x} = t_{z'}^{\mathfrak{T}_x}[\hat{r}_e \hat{\sigma}]_{\varphi_\tau(z')}$ $\stackrel{\text{(T-1)}}{=} t_{e.z'}^{\Phi(\mathfrak{T}_x)}$ and $p_{w'_e}^{\mathfrak{T}_x} \stackrel{\text{(T-2)}}{=} p_{z'}^{\mathfrak{T}_x} \cdot \hat{p}_e = p_{e.z'}^{\Phi(\mathfrak{T}_x)}$. Thus, we can copy the rest of this subtree of \mathfrak{T}_x to our newly generated tree $\Phi(\mathfrak{T}_x)$. In our example, v_1 has the only successor v_4 , hence we can copy the subtree starting at node v_4 , which is only the node itself, to the node $1.v_1$ in $\Phi(\mathfrak{T}_x)$. For v_5 , we have the only successor v_7 , hence we can copy the subtree starting at node v_7 , which is the node itself together with its successor v_9 , to the node $1.v_5$ in $\Phi(\mathfrak{T}_x)$. So essentially, we just had to define how to construct $\Phi(\mathfrak{T}_x)$ for the part of the tree before we reach nodes v with $\varphi_\tau(v) = \perp$ in \mathfrak{T}_x . Now we have to show that $|\Phi(\mathfrak{T}_x)| = |\mathfrak{T}_x|$ and that $\Phi(\mathfrak{T}_x)$ is indeed a valid $\xrightarrow{\mathcal{P}}$ -RST (i.e., that the edges between nodes $e.z$ with $z \in Z$ and its successors correspond to rewrite steps with \mathcal{P}).

1.2 We show $|\Phi(\mathfrak{T}_x)| = |\mathfrak{T}_x|$.

Let v be a leaf in $\Phi(\mathfrak{T}_x)$. If $v = e'.z$ for some node $z \in Z$ that is a leaf in \mathfrak{T}_x (e.g., node $1.v_2$), then also $e.z$ must be a leaf in $\Phi(\mathfrak{T}_x)$ for every $1 \leq e \leq h$. Here, we get $\sum_{1 \leq e \leq h} p_{e.z}^{\Phi(\mathfrak{T}_x)} \stackrel{\text{(T-2)}}{=} \sum_{1 \leq e \leq h} p_z^{\mathfrak{T}_x} \cdot \hat{p}_e = p_z^{\mathfrak{T}_x} \cdot \sum_{1 \leq e \leq h} \hat{p}_e = p_z^{\mathfrak{T}_x} \cdot 1 = p_z^{\mathfrak{T}_x}$, and thus

$$\sum_{\substack{e.z \in \text{Leaf}^{\Phi(\mathfrak{T}_x)} \\ z \in Z, z \in \text{Leaf}^{\mathfrak{T}_x}}} p_{e.z}^{\Phi(\mathfrak{T}_x)} = \sum_{\substack{z \in \text{Leaf}^{\mathfrak{T}_x} \\ z \in Z}} \left(\sum_{1 \leq e \leq h} p_{e.z}^{\Phi(\mathfrak{T}_x)} \right) = \sum_{\substack{z \in \text{Leaf}^{\mathfrak{T}_x} \\ z \in Z}} p_z^{\mathfrak{T}_x}$$

If $v = e.z$ for some node $z \in Z$ that is not a leaf in \mathfrak{T}_x (e.g., node $1.v_1$), then we know by construction that the e -th successor w_e of z in \mathfrak{T}_x is not contained in Z and is a leaf of \mathfrak{T}_x . Let $(zE)_e$ denote the e -th successor of z . Here, we get $p_{e.z}^{\Phi(\mathfrak{T}_x)} \stackrel{\text{(T-2)}}{=} p_z^{\mathfrak{T}_x} \cdot \hat{p}_e = p_{w_e}^{\mathfrak{T}_x}$, and thus

$$\sum_{\substack{e.z \in \text{Leaf}^{\Phi(\mathfrak{T}_x)} \\ z \in Z, z \notin \text{Leaf}^{\mathfrak{T}_x}}} p_{e.z}^{\Phi(\mathfrak{T}_x)} = \sum_{\substack{w_e \in \text{Leaf}^{\mathfrak{T}_x} \\ w_e = (zE)_e, w_e \notin Z, z \in Z}} p_{e.z}^{\Phi(\mathfrak{T}_x)} = \sum_{\substack{w_e \in \text{Leaf}^{\mathfrak{T}_x} \\ w_e = (zE)_e, w_e \notin Z, z \in Z}} p_{w_e}^{\mathfrak{T}_x}$$

Finally, if v does not have the form $v = e.z$, then v is also a leaf in \mathfrak{T}_x with $p_v^{\Phi(\mathfrak{T}_x)} = p_v^{\mathfrak{T}_x}$ and for both v and its predecessor u we have $v, u \notin Z$, and thus

$$\sum_{\substack{v \in \text{Leaf}^{\Phi(\mathfrak{T}_x)} \\ v \in \text{Leaf}^{\mathfrak{T}_x}}} p_v^{\Phi(\mathfrak{T}_x)} = \sum_{\substack{v \in \text{Leaf}^{\mathfrak{T}_x} \\ v \in uE, u \notin Z}} p_v^{\Phi(\mathfrak{T}_x)} = \sum_{\substack{v \in \text{Leaf}^{\mathfrak{T}_x} \\ v \in uE, u \notin Z}} p_v^{\mathfrak{T}_x}$$

Note that these cases cover each leaf of \mathfrak{T}_x exactly once. These three equations imply:

$$\begin{aligned} |\Phi(\mathfrak{T}_x)| &= \sum_{z \in \text{Leaf}^{\Phi(\mathfrak{T}_x)}} p_z^{\Phi(\mathfrak{T}_x)} = \sum_{\substack{e.z \in \text{Leaf}^{\Phi(\mathfrak{T}_x)} \\ z \in Z, z \in \text{Leaf}^{\mathfrak{T}_x}}} p_{e.z}^{\Phi(\mathfrak{T}_x)} + \sum_{\substack{e.z \in \text{Leaf}^{\Phi(\mathfrak{T}_x)} \\ z \in Z, z \notin \text{Leaf}^{\mathfrak{T}_x}}} p_{e.z}^{\Phi(\mathfrak{T}_x)} + \sum_{\substack{v \in \text{Leaf}^{\Phi(\mathfrak{T}_x)} \\ v \in \text{Leaf}^{\mathfrak{T}_x}}} p_v^{\Phi(\mathfrak{T}_x)} \\ &= \sum_{\substack{z \in \text{Leaf}^{\mathfrak{T}_x} \\ z \in Z}} p_z^{\mathfrak{T}_x} + \sum_{\substack{w_e \in \text{Leaf}^{\mathfrak{T}_x} \\ w_e = (zE)_e, w_e \notin Z, z \in Z}} p_{w_e}^{\mathfrak{T}_x} + \sum_{\substack{v \in \text{Leaf}^{\mathfrak{T}_x} \\ v \in uE, u \notin Z}} p_v^{\mathfrak{T}_x} = \sum_{z \in \text{Leaf}^{\mathfrak{T}_x}} p_z^{\mathfrak{T}_x} = |\mathfrak{T}_x| \end{aligned}$$

1.3 We show that $\Phi(\mathfrak{T}_x)$ is indeed a valid RST.

Finally, we prove that $\Phi(\mathfrak{T}_x)$ is a valid $\xrightarrow{\mathcal{P}}$ -RST. Here, we only need to show that $t_{e.z}^{\Phi(\mathfrak{T}_x)} \xrightarrow{\mathcal{P}} \{p_{e.w_1}^{\Phi(\mathfrak{T}_x)} : t_{e.w_1}^{\Phi(\mathfrak{T}_x)}, \dots, p_{e.w_m}^{\Phi(\mathfrak{T}_x)} : t_{e.w_m}^{\Phi(\mathfrak{T}_x)}\}$ for all $z \in Z$ as all the other edges and labelings were already present in \mathfrak{T}_x , which is a valid $\xrightarrow{\mathcal{P}}$ -RST, and we have already seen that we have a valid innermost rewrite step at the new root \hat{v} .

Let $z \in Z$. In the following, we distinguish between two different cases for a rewrite step at a node z of \mathfrak{T}_x :

- (A) We rewrite at a position parallel to $\varphi_\tau(z)$ or we have $\varphi_\tau(z) = \top$.
- (B) We rewrite at a position above $\varphi_\tau(z)$. Here, we need that \mathcal{P} is NO and linear.

Let $t_z^{\mathfrak{T}_x} \xrightarrow{\mathcal{P}} \{p_z^{\mathfrak{T}_x} \cdot p_1 : t_{w_1}^{\mathfrak{T}_x}, \dots, p_z^{\mathfrak{T}_x} \cdot p_m : t_{w_m}^{\mathfrak{T}_x}\}$, with a rule $\ell \rightarrow \{p_1 : r_1, \dots, p_m : r_m\} \in \mathcal{P}$, a substitution σ , and a position π such that $t_z^{\mathfrak{T}_x}|_\pi = \ell\sigma$. We have $t_{w_j}^{\mathfrak{T}_x} = t_z^{\mathfrak{T}_x}[r_j\sigma]_\pi$ for all $1 \leq j \leq m$.

(A) We start with the case where we have $\pi \perp \varphi_\tau(z)$ or $\varphi_\tau(z) = \top$. By (T-1), we get $t_{e.z}^{\Phi(\mathfrak{T}_x)} = t_z^{\mathfrak{T}_x}[\hat{r}_e\hat{\sigma}]_{\varphi_\tau(z)}$ if $\varphi_\tau(z) \in \mathbb{N}^*$ and $t_{e.z}^{\Phi(\mathfrak{T}_x)} = t_z^{\mathfrak{T}_x}$ if $\varphi_\tau(z) = \top$. In both cases, we can rewrite $t_{e.z}^{\Phi(\mathfrak{T}_x)}$ using the same rule, the same substitution, and the same position, as we have $t_{e.z}^{\Phi(\mathfrak{T}_x)}|_\pi = t_z^{\mathfrak{T}_x}[\hat{r}_e\hat{\sigma}]_{\varphi_\tau(z)}|_\pi = t_z^{\mathfrak{T}_x}|_\pi = \ell\sigma$ or directly $t_{e.z}^{\Phi(\mathfrak{T}_x)}|_\pi = t_z^{\mathfrak{T}_x}|_\pi = \ell\sigma$.

It remains to show that $t_{e.w_j}^{\Phi(\mathfrak{T}_x)} = t_{e.z}^{\Phi(\mathfrak{T}_x)}[r_j\sigma]_\pi$ for all $1 \leq j \leq m$, i.e., that the labeling we defined for $\Phi(\mathfrak{T}_x)$ corresponds to this rewrite step. Let $1 \leq j \leq m$. If $\varphi_\tau(z) \in \mathbb{N}^*$, then we have $t_{e.z}^{\Phi(\mathfrak{T}_x)}[r_j\sigma]_\pi = t_z^{\mathfrak{T}_x}[\hat{r}_e\hat{\sigma}]_{\varphi_\tau(z)}[r_j\sigma]_\pi \stackrel{\varphi_\tau(z) \perp \pi}{=} t_z^{\mathfrak{T}_x}[r_j\sigma]_\pi[\hat{r}_e\hat{\sigma}]_{\varphi_\tau(z)} = t_{w_j}^{\mathfrak{T}_x}[\hat{r}_e\hat{\sigma}]_{\varphi_\tau(z)} = t_{e.w_j}^{\Phi(\mathfrak{T}_x)}$. If $\varphi_\tau(z) = \top$, then $t_{e.z}^{\Phi(\mathfrak{T}_x)}[r_j\sigma]_\pi = t_z^{\mathfrak{T}_x}[r_j\sigma]_\pi = t_{w_j}^{\mathfrak{T}_x} = t_{e.w_j}^{\Phi(\mathfrak{T}_x)}$. Finally, note that the probabilities of the labeling are correct, as we are using the same rule with the same probabilities.

(B) If we have $\pi < \varphi_\tau(z)$, then there exists a $\chi \in \mathbb{N}^+$ such that $\pi \cdot \chi = \varphi_\tau(z)$. Since the rules of \mathcal{P} are non-overlapping, the redex must be completely “inside” the used substitution σ , and we can find a position α_q of a variable q in ℓ and another position β such that $\chi = \alpha_q \cdot \beta$. Furthermore, since the rule is linear, q only occurs once in ℓ and at most once in r_j for all $1 \leq j \leq m$. Let ρ_q^j be the position of q in r_j if it exists. By (T-1), we get $t_z^{\mathfrak{T}_x}[\hat{r}_e\hat{\sigma}]_{\varphi_\tau(z)} = t_{e.z}^{\Phi(\mathfrak{T}_x)}$. We can rewrite $t_{e.z}^{\Phi(\mathfrak{T}_x)}$ using the same rule, the substitution σ' with $\sigma'(q) = \sigma(q)[\hat{r}_e\hat{\sigma}]_\beta$ and $\sigma'(q') = \sigma(q')$ for all other variables $q' \neq q$, and the same position, as we have $t_{e.z}^{\Phi(\mathfrak{T}_x)}|_\pi = t_z^{\mathfrak{T}_x}[\hat{r}_e\hat{\sigma}]_{\varphi_\tau(z)}|_\pi = t_z^{\mathfrak{T}_x}|_\pi[\hat{r}_e\hat{\sigma}]_\chi = \ell\sigma'$.

It remains to show that $t_{e.w_j}^{\Phi(\mathfrak{T}_x)} = t_{e.z}^{\Phi(\mathfrak{T}_x)}[r_j\sigma']_\pi$ for all $1 \leq j \leq m$, i.e., that the labeling we defined for $\Phi(\mathfrak{T}_x)$ corresponds to this rewrite step. Let $1 \leq j \leq m$. If ρ_q^j exists, then we have $t_{e.z}^{\Phi(\mathfrak{T}_x)}[r_j\sigma']_\pi = t_z^{\mathfrak{T}_x}[\hat{r}_e\hat{\sigma}]_{\varphi_\tau(z)}[r_j\sigma']_\pi = t_z^{\mathfrak{T}_x}[\hat{r}_e\hat{\sigma}]_{\pi \cdot \alpha_q \cdot \beta}[r_j\sigma']_\pi = t_z^{\mathfrak{T}_x}[r_j\sigma]_\pi[\hat{r}_e\hat{\sigma}]_{\rho_q^j \cdot \beta} = t_{w_j}^{\mathfrak{T}_x}[\hat{r}_e\hat{\sigma}]_{\varphi_\tau(z)} = t_{e.w_j}^{\Phi(\mathfrak{T}_x)}$. Otherwise, we erase the precise redex and obtain $t_{e.z}^{\Phi(\mathfrak{T}_x)}[r_j\sigma]_\pi =$

$t_z^{\mathfrak{T}_x}[\hat{r}_e \hat{\sigma}]_{\varphi_\tau(z)}[r_j \sigma']_\pi = t_z^{\mathfrak{T}_x}[\hat{r}_e \hat{\sigma}]_{\pi.\alpha_q.\beta}[r_j \sigma']_\pi = t_z^{\mathfrak{T}_x}[r_j \sigma]_\pi = t_{w_j}^{\mathfrak{T}_x} = t_{e.w_j}^{\Phi(\mathfrak{T}_x)}$. Again, the probabilities of the labeling are correct, as we are using the same rule with the same probabilities.

2. Analyzing the expected derivation length of $\mathfrak{T}^{(\infty)}$

Our goal is to show that in the construction of the trees $\mathfrak{T}^{(1)}, \mathfrak{T}^{(2)}, \dots$, every leaf v of \mathfrak{T} is turned into leaves of $\mathfrak{T}^{(1)}, \mathfrak{T}^{(2)}, \dots$ whose probabilities sum up to $p_v^{\mathfrak{T}}$, and whose depths are greater or equal than the original depth $d^{\mathfrak{T}}(v)$ of v in \mathfrak{T} . This implies $\text{edl}(\mathfrak{T}) \leq \text{edl}(\mathfrak{T}^{(1)}) \leq \text{edl}(\mathfrak{T}^{(2)}) \leq \dots$. From this observation, we then conclude $\text{edl}(\mathfrak{T}) \leq \text{edl}(\mathfrak{T}^{(\infty)})$.

2.1 We prove $\text{edl}(\mathfrak{T}) \leq \text{edl}(\mathfrak{T}^{(1)}) \leq \text{edl}(\mathfrak{T}^{(2)}) \leq \dots$

We start by considering how the leaves of \mathfrak{T} correspond to the leaves of $\mathfrak{T}^{(1)}$. Let v be a leaf in \mathfrak{T} . If $v \notin xE^*$, then v is also a leaf in $\mathfrak{T}^{(1)}$, labeled with the same probability, and at the same depth. Otherwise, $v \in xE^*$, which means that either $v \in Z$, or $v \notin Z$ but its predecessor is in Z , or neither v nor its predecessor are in Z .

If $v \in Z$ (like the node v_2 in Fig. 6), then also $e.v$ must be a leaf in $\Phi(\mathfrak{T}_x)$ for every $1 \leq e \leq h$. Here, we again have $\sum_{1 \leq e \leq h} p_{e.v}^{\Phi(\mathfrak{T}_x)} = p_v^{\mathfrak{T}_x}$ as before. In addition, we also know that if $e.v$ is at depth m in $\Phi(\mathfrak{T}_x)$, then v is at depth $m - 1$ in \mathfrak{T}_x .

If $v \notin Z$ but v is the e -th successor of a node $z \in Z$ (like the node v_8 in Fig. 6), then $e.z$ is a leaf in $\Phi(\mathfrak{T}_x)$. Here, we again have $p_{e.z}^{\Phi(\mathfrak{T}_x)} = p_v^{\mathfrak{T}_x}$ as before. In addition, we also know that if $e.z$ is at depth m in $\Phi(\mathfrak{T}_x)$, then v is at depth m in \mathfrak{T}_x .

Finally, if neither v nor its predecessor are in Z (like the node v_9 in Fig. 6), then v is also a leaf in $\Phi(\mathfrak{T}_x)$ with $p_v^{\Phi(\mathfrak{T}_x)} = p_v^{\mathfrak{T}_x}$ and at the same depth.

These cases cover each leaf of $\mathfrak{T}^{(1)}$ exactly once and all leaves in $\Phi(\mathfrak{T}_x)$ are at a depth greater or equal than the corresponding leaves in \mathfrak{T}_x , implying that for each leaf u in \mathfrak{T} we can find a set of leaves $\Omega_u^{(1)}$ in $\mathfrak{T}^{(1)}$ such that $d^{\mathfrak{T}}(u) \leq d^{\mathfrak{T}^{(1)}}(w)$ for each $w \in \Omega_u^{(1)}$, $\sum_{w \in \Omega_u^{(1)}} p_w^{\mathfrak{T}^{(1)}} = p_u^{\mathfrak{T}}$, and $\text{Leaf}^{\mathfrak{T}^{(1)}} = \biguplus_{u \in \text{Leaf}^{\mathfrak{T}}} \Omega_u^{(1)}$.

We can now use the same observation for each leaf w in the tree $\mathfrak{T}^{(1)}$ in order to obtain a new set $\Xi_w^{(2)}$ of leaves in $\mathfrak{T}^{(2)}$ and define $\Omega_u^{(2)} = \bigcup_{w \in \Omega_u^{(1)}} \Xi_w^{(2)}$. Again, we get $d^{\mathfrak{T}}(u) \leq d^{\mathfrak{T}^{(2)}}(w)$ for each $w \in \Omega_u^{(2)}$, $\sum_{w \in \Omega_u^{(2)}} p_w^{\mathfrak{T}^{(2)}} = p_u^{\mathfrak{T}}$, and $\text{Leaf}^{\mathfrak{T}^{(2)}} = \biguplus_{u \in \text{Leaf}^{\mathfrak{T}}} \Omega_u^{(2)}$. We now do this for each $i \in \mathbb{N}$ to define the set $\Omega_u^{(i)}$ for each leaf u in the original tree \mathfrak{T} . Overall, this implies $\text{edl}(\mathfrak{T}) \leq \text{edl}(\mathfrak{T}^{(1)}) \leq \text{edl}(\mathfrak{T}^{(2)}) \leq \dots$.

2.2 We prove $\text{edl}(\mathfrak{T}^{(\infty)}) \geq \text{edl}(\mathfrak{T})$.

From the construction of the $\Omega_u^{(i)}$ above, in the end, we obtain

$$\text{Leaf}^{\mathfrak{T}^{(\infty)}} = \biguplus_{u \in \text{Leaf}^{\mathfrak{T}}} \limsup_{i \rightarrow \infty} \Omega_u^{(i)},$$

where $\limsup_{i \rightarrow \infty} \Omega_u^{(i)} = \bigcap_{i' \in \mathbb{N}} \bigcup_{i > i'} \Omega_u^{(i)} = \{w \mid w \text{ is contained in infinitely many } \Omega_u^{(i)}\}$. To see this, let $v \in \text{Leaf}^{\mathfrak{T}^{(\infty)}}$. Remember that for every depth H of the tree, there exists an m_H such that $\mathfrak{T}^{(\infty)}$ and $\mathfrak{T}^{(i)}$ are the same trees up to depth H for all $i \geq m_H$. This means that the node v must be contained in all trees $\mathfrak{T}^{(m)}$ with $m \geq m_{d^{\mathfrak{T}^{(\infty)}}(v)}$, i.e., it is contained in $\biguplus_{u \in \text{Leaf}^{\mathfrak{T}}} \limsup_{i \rightarrow \infty} \Omega_u^{(i)}$. The other direction of the equality follows in the same manner.

Since the probabilities of the leaves in $\mathfrak{T}^{(i)}$ always add up to the probability of the corresponding leaf in \mathfrak{T} , we have $\sum_{v \in \limsup_{i \rightarrow \infty} \Omega_u^{(i)}} p_v^{\mathfrak{T}^{(\infty)}} \leq p_u^{\mathfrak{T}}$ for all $u \in \text{Leaf}^{\mathfrak{T}}$. We now

show $\text{edl}(\mathfrak{T}) \leq \text{edl}(\mathfrak{T}^{(\infty)})$ by considering the two cases where $\sum_{v \in \limsup_{i \rightarrow \infty} \Omega_u^{(i)}} p_v^{\mathfrak{T}^{(\infty)}} < p_u^{\mathfrak{T}}$ for some $u \in \text{Leaf}^{\mathfrak{T}}$ and where $\sum_{v \in \limsup_{i \rightarrow \infty} \Omega_u^{(i)}} p_v^{\mathfrak{T}^{(\infty)}} = p_u^{\mathfrak{T}}$ for all $u \in \text{Leaf}^{\mathfrak{T}}$.

If we have $\sum_{v \in \limsup_{i \rightarrow \infty} \Omega_u^{(i)}} p_v^{\mathfrak{T}^{(\infty)}} < p_u^{\mathfrak{T}}$ for some $u \in \text{Leaf}^{\mathfrak{T}}$, then we obtain

$$\begin{aligned} 1 &\geq |\mathfrak{T}| &= \sum_{u \in \text{Leaf}^{\mathfrak{T}}} p_u^{\mathfrak{T}} \\ &> \sum_{u \in \text{Leaf}^{\mathfrak{T}}} \sum_{v \in \limsup_{i \rightarrow \infty} \Omega_u^{(i)}} p_v^{\mathfrak{T}^{(\infty)}} &= \sum_{v \in \bigsqcup_{u \in \text{Leaf}^{\mathfrak{T}}} \limsup_{i \rightarrow \infty} \Omega_u^{(i)}} p_v^{\mathfrak{T}^{(\infty)}} \\ &= \sum_{v \in \text{Leaf}^{\mathfrak{T}^{(\infty)}}} p_v^{\mathfrak{T}^{(\infty)}} &= |\mathfrak{T}^{(\infty)}| \end{aligned}$$

and from $|\mathfrak{T}^{(\infty)}| < 1$ we directly get $\text{edl}(\mathfrak{T}^{(\infty)}) = \infty$.

Otherwise, we have $\sum_{v \in \limsup_{i \rightarrow \infty} \Omega_u^{(i)}} p_v^{\mathfrak{T}^{(\infty)}} = p_u^{\mathfrak{T}}$ for all $u \in \text{Leaf}^{\mathfrak{T}}$, and thus we obtain

$$\begin{aligned} \text{edl}(\mathfrak{T}) &= \sum_{n=0}^{\infty} \sum_{u \in N^{\mathfrak{T}} \setminus \text{Leaf}^{\mathfrak{T}}} p_u^{\mathfrak{T}} \\ &= \sum_{n=0}^{\infty} (1 - \sum_{\substack{u \in \text{Leaf}^{\mathfrak{T}} \\ d^{\mathfrak{T}}(u) \leq n}} p_u^{\mathfrak{T}}) \\ &= \sum_{n=0}^{\infty} (1 - \sum_{\substack{u \in \text{Leaf}^{\mathfrak{T}} \\ d^{\mathfrak{T}}(u) \leq n}} \sum_{v \in \limsup_{i \rightarrow \infty} \Omega_u^{(i)}} p_v^{\mathfrak{T}^{(\infty)}}) \\ &\leq \sum_{n=0}^{\infty} (1 - \sum_{\substack{u \in \text{Leaf}^{\mathfrak{T}} \\ d^{\mathfrak{T}^{(\infty)}}(v) \leq n}} \sum_{v \in \limsup_{i \rightarrow \infty} \Omega_u^{(i)}} p_v^{\mathfrak{T}^{(\infty)}}) \\ &= \sum_{n=0}^{\infty} (1 - \sum_{\substack{v \in \bigsqcup_{u \in \text{Leaf}^{\mathfrak{T}}} \limsup_{i \rightarrow \infty} \Omega_u^{(i)} \\ d^{\mathfrak{T}^{(\infty)}}(v) \leq n}} p_v^{\mathfrak{T}^{(\infty)}}) \\ &= \sum_{n=0}^{\infty} (1 - \sum_{\substack{v \in \text{Leaf}^{\mathfrak{T}^{(\infty)}} \\ d^{\mathfrak{T}^{(\infty)}}(v) \leq n}} p_v^{\mathfrak{T}^{(\infty)}}) \\ &= \sum_{n=0}^{\infty} \sum_{\substack{u \in N \setminus \text{Leaf}^{\mathfrak{T}^{(\infty)}} \\ d^{\mathfrak{T}^{(\infty)}}(u) = n}} p_u^{\mathfrak{T}^{(\infty)}} \\ &= \text{edl}(\mathfrak{T}^{(\infty)}) \end{aligned}$$

and therefore, $\text{edl}(\mathfrak{T}^{(\infty)}) \geq \text{edl}(\mathfrak{T})$. □

Theorem 4.9 (From $\mathbf{wPSN}_{\mathfrak{f} \rightarrow \mathcal{P}}$ to $\mathbf{PSN}_{\mathfrak{f} \rightarrow \mathcal{P}}$). *If a PTRS \mathcal{P} is NO, linear, and NE, then*

$$\mathbf{PSN}_{\mathfrak{f} \rightarrow \mathcal{P}} \iff \mathbf{wPSN}_{\mathfrak{f} \rightarrow \mathcal{P}}$$

Proof. We only have to prove the non-trivial direction “ \Leftarrow ”. Let \mathcal{P} be a PTRS that is non-overlapping, left-linear, and non-erasing.

$\mathbf{AST}_{\mathfrak{f} \rightarrow \mathcal{P}} \Leftarrow \mathbf{wAST}_{\mathfrak{f} \rightarrow \mathcal{P}}$:

Assume for a contradiction that we have $\mathbf{wAST}_{\mathfrak{f} \rightarrow \mathcal{P}}$ but not $\mathbf{AST}_{\mathfrak{f} \rightarrow \mathcal{P}}$. This means that there exists an $\mathfrak{f} \rightarrow \mathcal{P}$ -RST \mathfrak{T} such that $|\mathfrak{T}| = c$ for some $0 \leq c < 1$. Let $t \in \mathcal{T}$ such that the root of \mathfrak{T} is labeled with $(1 : t)$. Since we have $\mathbf{wAST}_{\mathfrak{f} \rightarrow \mathcal{P}}$, there exists another $\mathfrak{f} \rightarrow \mathcal{P}$ -RST $\tilde{\mathfrak{T}} = (\tilde{N}, \tilde{E}, \tilde{L})$ such that $|\tilde{\mathfrak{T}}| = 1$ and the root of $\tilde{\mathfrak{T}}$ is also labeled with $(1 : t)$. Hence, in \mathfrak{T} at least one rewrite step is performed at some node x that is different to the rewrite step performed in $\tilde{\mathfrak{T}}$. The core steps of the proof are the same as for the proof of Lemma 4.3. We iteratively push the rewrite steps that would be performed in $\tilde{\mathfrak{T}}$ at node x to this node in \mathfrak{T} . Then, the limit of this construction is exactly $\tilde{\mathfrak{T}}$, which would mean that $|\tilde{\mathfrak{T}}| \leq c < 1$, which is the desired contradiction. For this, we have to adjust the construction $\Phi(\cdot)$. The rest of the proof remains completely the same.

1.1 Construction of $\Phi(-)$

Let $\mathfrak{T}_x = \mathfrak{T}[xE^*]$ be an $\xrightarrow{\mathcal{P}}$ -RST that performs a rewrite step at position ζ at the root node x , i.e., $t_x^{\mathfrak{T}_x} \xrightarrow{\mathcal{P}, \zeta} \{p_{y_1}^{\mathfrak{T}_x} : t_{y_1}^{\mathfrak{T}_x}, \dots, p_{y_k}^{\mathfrak{T}_x} : t_{y_k}^{\mathfrak{T}_x}\}$ using the rule $\bar{\ell} \rightarrow \{\bar{p}_1 : \bar{r}_1, \dots, \bar{p}_k : \bar{r}_k\}$, and the substitution $\bar{\sigma}$ such that $t_x^{\mathfrak{T}_x}|_{\zeta} = \bar{\ell}\bar{\sigma}$. Then $t_{y_j}^{\mathfrak{T}_x} = t_x^{\mathfrak{T}_x}[\bar{r}_j\bar{\sigma}]_{\zeta}$ for all $1 \leq j \leq k$. Furthermore, assume that $\tilde{T}_x = \tilde{\mathfrak{T}}[xE^*]$ rewrites at position τ (using a rule $\hat{\ell} \rightarrow \{\hat{p}_1 : \hat{r}_1, \dots, \hat{p}_h : \hat{r}_h\} \in \mathcal{P}$ and the substitution $\hat{\sigma}$) with $\tau \neq \zeta$. (Note that if $\tau = \zeta$, then by non-overlappingness the rewrite step would be the same.) Instead of applying the rewrite step at position ζ at the root x we want to directly apply the rewrite step at position τ .

The construction creates a new RST $\Phi(\mathfrak{T}_x) = (N', E', L')$ such that $|\Phi(\mathfrak{T}_x)| = |\mathfrak{T}_x|$, and that directly performs a rewrite step at position τ at the root of the tree, by pushing it from the original nodes in the tree \mathfrak{T}_x to the root. This push only results in the same convergence probability due to our restriction that \mathcal{P} is linear and non-erasing. We need the non-erasing property now, because τ may be above ζ , which was not possible in the proof of Lemma 4.3.

Again, since we are allowed to rewrite above τ in the original tree \mathfrak{T}_x , the actual position of the redex that was originally at position τ might change during the application of a rewrite step. Hence, we recursively define the position $\varphi_{\tau}(v)$ that contains precisely this redex for each node v in \mathfrak{T}_x until we rewrite at this position. Compared to the proof of Lemma 4.3, since the rules in \mathcal{P} are non-erasing, we only have $\varphi_{\tau}(v) \in \mathbb{N}^*$ or $\varphi_{\tau}(v) = \perp$. The option $\varphi_{\tau}(v) = \top$ is not possible anymore. Initially, we have $\varphi_{\tau}(x) = \tau$. Whenever we have defined $\varphi_{\tau}(v)$ for some node v , and we have $t_v^{\mathfrak{T}_x} \xrightarrow{\mathcal{P}} \{p_{w_1}^{\mathfrak{T}_x} : t_{w_1}^{\mathfrak{T}_x}, \dots, p_{w_m}^{\mathfrak{T}_x} : t_{w_m}^{\mathfrak{T}_x}\}$ for the direct successors $vE = \{w_1, \dots, w_m\}$, using the rule $\ell \rightarrow \{p_1 : r_1, \dots, p_m : r_m\}$, the substitution σ , and position π , we do the following: If $\varphi_{\tau}(v) = \pi$, then we set $\varphi_{\tau}(w_j) = \perp$ for all $1 \leq j \leq m$ to indicate that we have rewritten the redex. If we have $\varphi_{\tau}(v) \perp \pi$, meaning that the rewrite step takes place parallel to $\varphi_{\tau}(v)$, then we set $\varphi_{\tau}(w_j) = \varphi_{\tau}(v)$ for all $1 \leq j \leq m$, as the position did not change. If we have $\varphi_{\tau}(v) < \pi$, then we set $\varphi_{\tau}(w_j) = \varphi_{\tau}(v)$ for all $1 \leq j \leq m$ as well, as the position did not change either. If we have $\pi < \varphi_{\tau}(v)$, then there exists a $\chi \in \mathbb{N}^+$ such that $\pi.\chi = \varphi_{\tau}(v)$. Since the rules of \mathcal{P} are non-overlapping, the redex must be completely “inside” the used substitution σ , and we can find a position α_q of a variable q in ℓ and another position β such that $\chi = \alpha_q.\beta$. Furthermore, since the rule is linear and non-erasing, q only occurs once in ℓ and once in r_j for all $1 \leq j \leq m$. Let ρ_q^j be the position of q in r_j . Here, we set $\varphi_{\tau}(w_j) = \rho_q^j.\beta$. Finally, if $\varphi_{\tau}(v) = \perp$, then we set $\varphi_{\tau}(w_j) = \varphi_{\tau}(v) = \perp$ for all $1 \leq j \leq m$ as well.

Again, we now define the $\xrightarrow{\mathcal{P}}$ -RST $\Phi(\mathfrak{T}_x)$ whose root is labeled with $(1 : t_x^{\mathfrak{T}_x})$ such that $|\Phi(\mathfrak{T}_x)| = |\mathfrak{T}_x|$, and that directly performs the rewrite step $t_x^{\mathfrak{T}_x} \xrightarrow{\mathcal{P}, \tau} \{\hat{p}_1 : t_{1.x}^{\Phi(\mathfrak{T}_x)}, \dots, \hat{p}_h : t_{h.x}^{\Phi(\mathfrak{T}_x)}\}$, with the rule $\hat{\ell} \rightarrow \{\hat{p}_1 : \hat{r}_1, \dots, \hat{p}_h : \hat{r}_h\} \in \mathcal{P}$, the substitution $\hat{\sigma}$, and the position τ , at the new root \hat{v} . Here, we have $t_x^{\mathfrak{T}_x}|_{\tau} = \hat{\ell}\hat{\sigma}$. Let Z be the set of all nodes v such that $\varphi_{\tau}(v) \neq \perp$. For each of these nodes $z \in Z$ and each $1 \leq e \leq h$, we create a new node $e.z \in N'$ with edges as in \mathfrak{T}_x for the nodes in Z . Furthermore, we add the edges from the new root \hat{v} to the nodes $e.x$ for all $1 \leq e \leq h$. Remember that x was the root in the tree \mathfrak{T}_x and has to be contained in Z . We define the labeling of the nodes in $\Phi(\mathfrak{T}_x)$ as follows for all nodes z in Z :

- (T-1) $t_{e.z}^{\Phi(\mathfrak{T}_x)} = t_z^{\mathfrak{T}_x}[\hat{r}_e\hat{\sigma}]_{\varphi_{\tau}(z)}$ for the substitution $\hat{\sigma}$ such that $t_z^{\mathfrak{T}_x}|_{\varphi_{\tau}(z)} = \hat{\ell}\hat{\sigma}$
- (T-2) $p_{e.z}^{\Phi(\mathfrak{T}_x)} = p_z^{\mathfrak{T}_x} \cdot \hat{p}_e$

Now, for a leaf $e.z' \in N'$ either $z' \in N$ is also a leaf or we rewrite at the position $\varphi_\tau(z')$ in node z' in \mathfrak{T}_x . If we rewrite $t_{z'}^{\mathfrak{T}_x} \xrightarrow{\mathcal{P}, \varphi_\tau(z')} \{p_{w'_1}^{\mathfrak{T}_x} : t_{w'_1}^{\mathfrak{T}_x}, \dots, p_{w'_h}^{\mathfrak{T}_x} : t_{w'_h}^{\mathfrak{T}_x}\}$, then we have $t_{w'_e}^{\mathfrak{T}_x} = t_{z'}^{\mathfrak{T}_x}[\hat{r}_e\delta]_{\varphi_\tau(z')} \stackrel{(T-1)}{=} t_{e.z'}^{\Phi(\mathfrak{T}_x)}$ for the substitution δ such that $t_{z'}^{\mathfrak{T}_x}|_{\varphi_\tau(z')} = \hat{\ell}\delta$ and $p_{w'_e}^{\mathfrak{T}_x} \stackrel{(T-2)}{=} p_{z'}^{\mathfrak{T}_x} \cdot \hat{p}_e = p_{e.z'}^{\Phi(\mathfrak{T}_x)}$. Thus, we can copy the rest of this subtree of \mathfrak{T}_x in our newly generated tree $\Phi(\mathfrak{T}_x)$. Again, we just had to define how to construct $\Phi(\mathfrak{T}_x)$ for the part of the tree before we reach the nodes v with $\varphi_\tau(v) = \perp$ in \mathfrak{T}_x . As in the proof of Lemma 4.3, we obtain $|\Phi(\mathfrak{T}_x)| = |\mathfrak{T}_x|$. We only need to show that $\Phi(\mathfrak{T}_x)$ is a valid $\xrightarrow{\mathcal{P}}$ -RST, i.e., that $t_{e.z}^{\Phi(\mathfrak{T}_x)} \xrightarrow{\mathcal{P}} \{p_{e.w_1}^{\Phi(\mathfrak{T}_x)} : t_{e.w_1}^{\Phi(\mathfrak{T}_x)}, \dots, p_{e.w_m}^{\Phi(\mathfrak{T}_x)} : t_{e.w_m}^{\Phi(\mathfrak{T}_x)}\}$ for all $z \in Z$ as in the proof of Lemma 4.3.

1.3 We show that $\Phi(\mathfrak{T}_x)$ is indeed a valid RST.

Let $z \in Z$. In the following, we distinguish three different cases for a rewrite step at node z of \mathfrak{T}_x :

- (A) We rewrite at a position parallel to $\varphi_\tau(z)$.
- (B) We rewrite at a position above $\varphi_\tau(z)$.
- (C) We rewrite at a position below $\varphi_\tau(z)$.

The cases (A) and (B) are analogous to our earlier proof, but in Case (A) we cannot erase the redex anymore. We only need to look at the new Case (C).

(C) If we have $t_z^{\mathfrak{T}_x} \xrightarrow{\mathcal{P}} \{p_z^{\mathfrak{T}_x} \cdot p_1 : t_{w_1}^{\mathfrak{T}_x}, \dots, p_z^{\mathfrak{T}_x} \cdot p_m : t_{w_m}^{\mathfrak{T}_x}\}$, then there is a rule $\ell \rightarrow \{p_1 : r_1, \dots, p_m : r_m\} \in \mathcal{P}$, a substitution σ , and a position π with $t_z^{\mathfrak{T}_x}|_\pi = \ell\sigma$. Then $t_{w_j}^{\mathfrak{T}_x} = t_z^{\mathfrak{T}_x}[r_j\sigma]_\pi$ for all $1 \leq j \leq m$. Additionally, we assume that $\varphi_\tau(z) < \pi$, and thus there exists a $\chi \in \mathbb{N}^+$ such that $\pi = \varphi_\tau(z) \cdot \chi$. By (T-1), we get $t_z^{\mathfrak{T}_x}[\hat{r}_e\delta]_{\varphi_\tau(z)} = t_{e.z}^{\Phi(\mathfrak{T}_x)}$ for the substitution δ such that $t_z^{\mathfrak{T}_x}|_{\varphi_\tau(z)} = \hat{\ell}\delta$. Since the rules of \mathcal{P} are non-overlapping, including the rule $\hat{\ell} \rightarrow \{\hat{p}_1 : \hat{r}_1, \dots, \hat{p}_h : \hat{r}_h\}$ that we use at the root of $\Phi(\mathfrak{T}_x)$, the redex for the current rewrite step must be completely “inside” the substitution δ , and we can find a variable position α_q of a variable q in $\hat{\ell}$ and another position β such that $\chi = \alpha_q \cdot \beta$. Furthermore, since the rule is also linear and non-erasing, q occurs exactly once in $\hat{\ell}$ and exactly once in \hat{r}_j for all $1 \leq j \leq m$. Let ρ_q^j be the position of q in r_j . We can rewrite $t_{e.z}^{\Phi(\mathfrak{T}_x)}$ using the same rule, the same substitution, and the same position, as we have $t_{e.z}^{\Phi(\mathfrak{T}_x)}|_\pi = t_z^{\mathfrak{T}_x}[\hat{r}_e\delta]_{\varphi_\tau(z)}|_\pi = \hat{r}_e\delta|_\chi = \delta(z)|_\beta = t_z^{\mathfrak{T}_x}|_{\varphi_\tau(z) \cdot \alpha_q \cdot \beta} = t_z^{\mathfrak{T}_x}|_{\varphi_\tau(z) \cdot \chi} = t_z^{\mathfrak{T}_x}|_\pi = \ell\sigma$.

It remains to show that $t_{e.w_j}^{\Phi(\mathfrak{T}_x)} = t_{e.z}^{\Phi(\mathfrak{T}_x)}[r_j\sigma']_\pi$ for all $1 \leq j \leq m$, i.e., that the labeling we defined for $\Phi(\mathfrak{T}_x)$ corresponds to this rewrite step. Let $1 \leq j \leq m$. We have $t_{e.z}^{\Phi(\mathfrak{T}_x)}[r_j\sigma]_\pi = t_z^{\mathfrak{T}_x}[\hat{r}_e\delta]_{\varphi_\tau(z)}[r_j\sigma]_\pi = t_z^{\mathfrak{T}_x}[\hat{r}_e\delta']_{\varphi_\tau(z)} = t_{e.w_j}^{\Phi(\mathfrak{T}_x)}$ for the substitution δ' with $\delta'(q) = \delta(q)[r_j\sigma]_\beta$ and $\delta'(q') = \delta(q')$ for all other variables $q' \neq q$. With this new substitution, we get $t_{w_j}^{\mathfrak{T}_x}|_{\varphi_\tau(w_j)} = t_z^{\mathfrak{T}_x}[r_j\sigma]_{\varphi_\tau(z) \cdot \alpha_q \cdot \beta}|_{\varphi_\tau(z)} = t_z^{\mathfrak{T}_x}|_{\varphi_\tau(z)}[r_j\sigma]_{\alpha_q \cdot \beta} = (\hat{\ell}\delta)[r_j\sigma]_{\alpha_q \cdot \beta} = \hat{\ell}\delta'$. Finally, note that the probabilities of the labeling are correct, as we are using the same rule with the same probabilities in both trees.

$\text{PAST} \xrightarrow{\mathcal{P}} \leftarrow \text{wPAST} \xrightarrow{\mathcal{P}} :$

By the same construction as for $\text{wAST} \xrightarrow{\mathcal{P}}$, we also get $\text{edl}(\mathfrak{T}^{(\infty)}) \geq \text{edl}(\mathfrak{T})$. □

Lemma 4.12 (From Leftmost-Innermost to Innermost Rewriting). *If a PTRS \mathcal{P} is NO, then for every infinite $\xrightarrow{\mathcal{P}}$ -rewrite sequence $\vec{\mu} = (\mu_n)_{n \in \mathbb{N}}$ there exists an infinite $\xrightarrow{\mathcal{P}}$ -rewrite sequence $\vec{\nu} = (\nu_n)_{n \in \mathbb{N}}$ such that*

- (i) $\lim_{n \rightarrow \infty} |\mu_n|_{\mathcal{P}} \geq \lim_{n \rightarrow \infty} |\nu_n|_{\mathcal{P}}$

$$(ii) \quad \text{edl}(\vec{\mu}) \leq \text{edl}(\vec{\nu})$$

Proof. The idea and the construction of this proof are completely analogous to the one of Lemma 4.3. We iteratively move leftmost-innermost rewrite steps to a higher position in the innermost RST. Hence, the resulting tree is a leftmost $\xrightarrow{\text{P}}$ -RST.

The construction of $\Phi(\cdot)$ is also analogous to the one in Lemma 4.3. The only difference to the proof of Lemma 4.3 is that the original tree \mathfrak{T}_x is already an $\xrightarrow{\text{P}}$ -RST. This means that during our construction only Case (A) can occur, as we cannot rewrite above a redex in an $\xrightarrow{\text{P}}$ -RST. And for Case (A), we only need the property of being non-overlapping. \square

Next, we prove the new results from Section 5.

Lemma 5.10 (From Innermost to Full Rewriting Starting in \mathcal{T}_B). *If a PTRS \mathcal{P} is OR and SP, then for every infinite $\xrightarrow{\text{P}}$ -rewrite sequence $\vec{\mu} = (\mu_n)_{n \in \mathbb{N}}$ that starts with a basic term there exists an infinite $\xrightarrow{\text{P}}$ -rewrite sequence $\vec{\nu} = (\nu_n)_{n \in \mathbb{N}}$ that starts with a basic term such that*

- (i) $\lim_{n \rightarrow \infty} |\mu_n|_{\mathcal{P}} \geq \lim_{n \rightarrow \infty} |\nu_n|_{\mathcal{P}}$
- (ii) $\text{edl}(\vec{\mu}) \leq \text{edl}(\vec{\nu})$

Proof. The proof is completely analogous to the one of Lemma 4.3. We iteratively move the innermost rewrite steps to a higher position using the construction $\Phi(\cdot)$. Note that since \mathcal{P} is spare and left-linear, in the construction of $\Phi(\cdot)$ and in the proof of Case (B), if the innermost redex is below a redex $\ell\sigma$ that is reduced next via a rule $\ell \rightarrow \{p_1 : r_1, \dots, p_m : r_m\}$, then the innermost redex is completely “inside” the used substitution σ , and it corresponds to a variable q which occurs only once in ℓ and at most once in r_j for all $1 \leq j \leq m$, due to sparseness of \mathcal{P} and the fact that we started with a basic term. Hence, we can use the same construction as in Lemma 4.3. \square

For Lemma 5.15 we need some more auxiliary functions from [Fuh19] to decode a basic term over $\Sigma \cup \Sigma_{\mathcal{G}(\mathcal{P})}$ into the original term over Σ .

Definition A.8 (Constructor Variant, Basic Variant, Decoded Variant). Let \mathcal{P} be a PTRS over the signature Σ . For a term $t \in \mathcal{T}$, we define its *constructor variant* $\text{cv}(t)$ inductively as follows:

- $\text{cv}(x) = x$ for $x \in \mathcal{V}$
- $\text{cv}(f(t_1, \dots, t_n)) = f(\text{cv}(t_1), \dots, \text{cv}(t_n))$ for $f \in \Sigma_C$
- $\text{cv}(f(t_1, \dots, t_n)) = \text{cons}_f(\text{cv}(t_1), \dots, \text{cv}(t_n))$ for $f \in \Sigma_D$

For a term $t \in \mathcal{T}$ with $t = f(t_1, \dots, t_n)$, we define its *basic variant* $\text{bv}(f(t_1, \dots, t_n)) = \text{enc}_f(\text{cv}(t_1), \dots, \text{cv}(t_n))$. For a term $t \in \mathcal{T}(\Sigma \cup \Sigma_{\mathcal{G}(\mathcal{P})}, \mathcal{V})$, we define its *decoded variant* $\text{dv}(t) \in \mathcal{T}$ as follows:

- $\text{dv}(x) = x$ for $x \in \mathcal{V}$
- $\text{dv}(\text{argenc}(t)) = \text{dv}(t)$
- $\text{dv}(f(t_1, \dots, t_n)) = g(\text{dv}(t_1), \dots, \text{dv}(t_n))$ for $f \in \{g, \text{cons}_g, \text{enc}_g\}$ with $g \in \Sigma_D$
- $\text{dv}(f(t_1, \dots, t_n)) = f(\text{dv}(t_1), \dots, \text{dv}(t_n))$ for $f \in \Sigma_C$

The only difference to the auxiliary functions from [Fuh19] is that dv now also removes the argenc symbols from a term. This was handled differently in [Fuh19] in order to ensure $|\text{dv}(t)| = |t|$, but this is irrelevant for our proofs regarding $\text{PSN}_{\xrightarrow{\text{P}}}$.

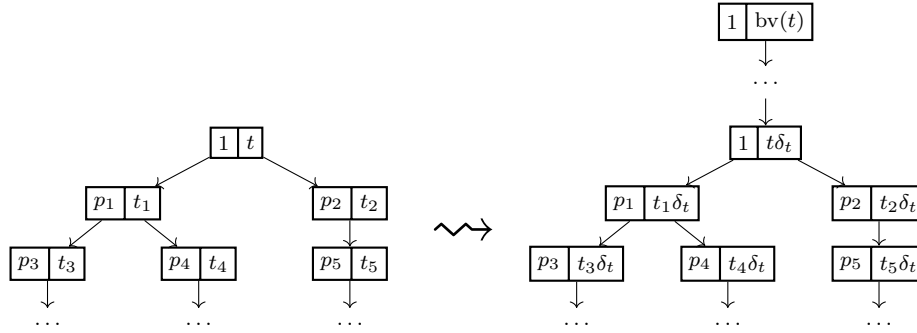
Lemma 5.15 (From all Terms to Basic Terms (1)). *For any PTRS \mathcal{P} we have:*

$$\text{AST}_{\mathcal{P}} \xleftrightarrow{\quad} \text{AST}_{\mathcal{P} \cup \mathcal{G}(\mathcal{P})} \text{ starting in } \mathcal{T}_{\mathcal{B}}.$$

Proof. “ \Leftarrow ”

Let $\mathfrak{T} = (N, E, L)$ be an \mathcal{P} -RST whose root is labeled with $(1 : t)$ for some $t \in \mathcal{T}$. We construct an $(\mathcal{P} \cup \mathcal{G}(\mathcal{P}))$ -RST $\mathfrak{T}' = (N', E', L')$ whose root is labeled with $(1 : \text{bv}(t))$ where $|\mathfrak{T}'| = |\mathfrak{T}|$ and $\text{edl}(\mathfrak{T}') \geq \text{edl}(\mathfrak{T})$.

In [Fuh19] it was shown that $\{1 : \text{bv}(t)\} \xrightarrow{\mathcal{G}(\mathcal{P})^*} \{1 : t\delta_t\}$. (PTRSs were not considered in [Fuh19], but since all the probabilities in the rules of $\mathcal{G}(\mathcal{P})$ are trivial, the proof in [Fuh19] directly translates to the probabilistic setting.) Here, for any term $t \in \mathcal{T}$ the substitution δ_t is defined by $\delta_t(x) = \text{argenc}(x)$ if $x \in \mathcal{V}(t)$ and $\delta_t(x) = x$ otherwise. The $(\mathcal{P} \cup \mathcal{G}(\mathcal{P}))$ -RST \mathfrak{T}' first performs these innermost rewrite steps to get from $\text{bv}(t)$ to $t\delta_t$, and then we can mirror the rewrite steps from \mathfrak{T} . To be precise, we use the same underlying tree structure and an adjusted labeling such that $p_x^{\mathfrak{T}} = p_x^{\mathfrak{T}'}$ and $t_x^{\mathfrak{T}}\delta_t = t_x^{\mathfrak{T}'}$ for all $x \in N$. Since the tree structure and the probabilities are the same, we then obtain $|\mathfrak{T}| = |\mathfrak{T}'|$.



“ \Rightarrow ”

Let $\mathfrak{T} = (N, E, L)$ be an $(\mathcal{P} \cup \mathcal{G}(\mathcal{P}))$ -RST whose root is labeled with $(1 : t)$ for some term $t \in \mathcal{T}(\Sigma \cup \Sigma_{\mathcal{G}(\mathcal{P})}, \mathcal{V})$. We construct an \mathcal{P} -RST $\mathfrak{T}' = (N', E', L')$ inductively such that for all leaves x of \mathfrak{T}' during the construction there exists a node $\varphi(x)$ of \mathfrak{T} such that

$$t_x^{\mathfrak{T}'} = \text{dv}(t_{\varphi(x)}^{\mathfrak{T}}) \quad \text{and} \quad p_x^{\mathfrak{T}'} = p_{\varphi(x)}^{\mathfrak{T}}. \quad (\text{A.1})$$

Here, φ is injective, i.e., every leaf x of \mathfrak{T}' is mapped to a (unique) node $\varphi(x)$ of \mathfrak{T} . Furthermore, after this construction, if x is still a leaf in \mathfrak{T}' , then $\varphi(x)$ is also a leaf in \mathfrak{T} . Hence, we obtain $|\mathfrak{T}| = \sum_{x \in \text{Leaf}^{\mathfrak{T}}} p_x^{\mathfrak{T}} \geq \sum_{x \in \text{Leaf}^{\mathfrak{T}'}} p_x^{\mathfrak{T}'} = |\mathfrak{T}'|$. Moreover, we only add rewrite steps in the beginning, so that $\text{edl}(\mathfrak{T}) \leq \text{edl}(\mathfrak{T}')$.

We label the root of \mathfrak{T}' with $(1 : \text{dv}(t))$. By letting φ map the root of \mathfrak{T}' to the root of \mathfrak{T} , the claim (A.1) is clearly satisfied. As long as there is still a node x in \mathfrak{T}' such that $\varphi(x)$ is not a leaf in \mathfrak{T} , we do the following. If we perform a rewrite step with $\mathcal{G}(\mathcal{P})$ at node $\varphi(x)$, i.e., $t_{\varphi(x)}^{\mathfrak{T}} \xrightarrow{\mathcal{G}(\mathcal{P})} \{1 : t_y^{\mathfrak{T}}\}$ for the only successor y of $\varphi(x)$, then we have $\text{dv}(t_{\varphi(x)}^{\mathfrak{T}}) = \text{dv}(t_y^{\mathfrak{T}})$, i.e., we do nothing in \mathfrak{T}' but simply change the definition of φ such that $\varphi(x)$ is now y . To see why $\text{dv}(t_{\varphi(x)}^{\mathfrak{T}}) = \text{dv}(t_y^{\mathfrak{T}})$ holds, note that we have $\text{dv}(\ell) = \text{dv}(r)$ for all rules $\ell \rightarrow \{1 : r\} \in \mathcal{G}(\mathcal{P})$, since $\text{dv}(\text{enc}_f(x_1, \dots, x_n)) = f(x_1, \dots, x_n) = \text{dv}(f(\text{argenc}(x_1), \dots, \text{argenc}(x_n)))$, and analogously $\text{dv}(\text{argenc}(\text{cons}_f(x_1, \dots, x_n))) = f(x_1, \dots, x_n) = \text{dv}(f(\text{argenc}(x_1), \dots, \text{argenc}(x_n)))$. This can then be lifted to arbitrary rewrite steps.

Otherwise, let $\varphi(x)E = \{y_1, \dots, y_k\}$ be the set of successors of $\varphi(x)$ in \mathfrak{T} , and we have $t_{\varphi(x)}^{\mathfrak{T}} \xrightarrow{\mathbf{f}}_{\mathcal{P}} \{ \frac{p_{y_1}^{\mathfrak{T}}}{p_{\varphi(x)}^{\mathfrak{T}}} : t_{y_1}^{\mathfrak{T}}, \dots, \frac{p_{y_k}^{\mathfrak{T}}}{p_{\varphi(x)}^{\mathfrak{T}}} : t_{y_k}^{\mathfrak{T}} \}$. Since $t_x^{\mathfrak{T}'} = \text{dv}(t_{\varphi(x)}^{\mathfrak{T}})$ and $p_x^{\mathfrak{T}'} = p_{\varphi(x)}^{\mathfrak{T}}$ by the induction hypothesis, then we also have $t_x^{\mathfrak{T}'} \xrightarrow{\mathbf{f}}_{\mathcal{P}} \{ \frac{p_{y_1}^{\mathfrak{T}'}}{p_x^{\mathfrak{T}'}} : t_{y_1}^{\mathfrak{T}'}, \dots, \frac{p_{y_k}^{\mathfrak{T}'}}{p_x^{\mathfrak{T}'}} : t_{y_k}^{\mathfrak{T}'} \}$ for terms $t_{y_j}^{\mathfrak{T}'}$ with $t_{y_j}^{\mathfrak{T}'} = \text{dv}(t_{y_j}^{\mathfrak{T}})$ and for $p_{y_j}^{\mathfrak{T}'} = p_{y_j}^{\mathfrak{T}}$. Thus, when defining $\varphi(y_j) = y_j$ for all $1 \leq j \leq k$, (A.1) is satisfied for the new leaves y_1, \dots, y_k of \mathfrak{T}' . \square

Lemma 5.16 (From all Terms to Basic Terms (2)). *For any PTRS \mathcal{P} we have:*

$$\begin{aligned} \text{PAST} \xrightarrow{\mathbf{f}}_{\mathcal{P}} &\Leftarrow \text{PAST} \xrightarrow{\mathbf{f}}_{\mathcal{P} \cup \mathcal{G}(\mathcal{P})} \text{ starting in } \mathcal{T}_{\mathcal{B}} \\ \text{SAST} \xrightarrow{\mathbf{f}}_{\mathcal{P}} &\Leftarrow \text{SAST} \xrightarrow{\mathbf{f}}_{\mathcal{P} \cup \mathcal{G}(\mathcal{P})} \text{ starting in } \mathcal{T}_{\mathcal{B}} \\ \text{edc} \xrightarrow{\mathbf{f}}_{\mathcal{P}} &\leq \text{erc} \xrightarrow{\mathbf{f}}_{\mathcal{P} \cup \mathcal{G}(\mathcal{P})} \end{aligned}$$

Proof. Let \mathfrak{T} be an $\xrightarrow{\mathbf{f}}_{\mathcal{P}}$ -RST whose root is labeled with $(1 : t)$ for some $t \in \mathcal{T}$. The construction from the previous proof yields an $(\mathcal{P} \cup \mathcal{G}(\mathcal{P}))$ -RST \mathfrak{T}' whose root is labeled with $(1 : \text{bv}(t))$ where $|\mathfrak{T}'| = |\mathfrak{T}|$. Since the tree structure and the probabilities are the same, and we only add rewrite steps in the beginning, we get $\text{edl}(\mathfrak{T}') \geq \text{edl}(\mathfrak{T})$. \square

A.3. Proofs for Section 7. We start by proving the *cutting lemma* that is needed to prove our modularity results for $\text{AST}_{\xrightarrow{s}_{\mathcal{P}}}$. It states that if there exists an \rightarrow -RST \mathfrak{T} that converges with probability < 1 and a partitioning of its inner nodes into two sets N_1 and N_2 such that every subtree of \mathfrak{T} that only contains inner nodes from N_1 converges with probability 1, then we can create a subtree of \mathfrak{T} that converges with probability < 1 as well such that every infinite path contains an infinite number of N_2 nodes (i.e., it does not contain any infinite path with eventually only nodes from N_1).

Lemma A.9 (Cutting Lemma). *Let $\rightarrow \subseteq \mathcal{T} \times \text{FDist}(\mathcal{T})$, and let \mathfrak{T} be an \rightarrow -RST with $|\mathfrak{T}| < 1$. Suppose we can partition its inner nodes into $N_1 \uplus N_2$ such that $|\mathfrak{T}_1| = 1$ holds for every subtree \mathfrak{T}_1 of \mathfrak{T} which only contains inner nodes from N_1 . Then there exists a subtree \mathfrak{T}' of \mathfrak{T} with $|\mathfrak{T}'| < 1$ such that every infinite path of \mathfrak{T}' has an infinite number of nodes from N_2 .*

Proof. Let $\mathfrak{T} = (N, E, L)$ be an \rightarrow -RST with $|\mathfrak{T}| = c < 1$ for some $c \in \mathbb{R}$. Since we have $0 \leq c < 1$, there is an $\varepsilon > 0$ such that $c + \varepsilon < 1$. Remember that the formula for the geometric series is:

$$\sum_{n=1}^{\infty} \left(\frac{1}{d}\right)^n = \frac{1}{d-1}, \text{ for all } d \in \mathbb{R} \text{ such that } \frac{1}{|d|} < 1$$

Let $d = \frac{1}{\varepsilon} + 2$. Then we have $\frac{1}{d} = \frac{1}{\frac{1}{\varepsilon} + 2} < 1$ and:

$$\frac{1}{\varepsilon} + 1 < \frac{1}{\varepsilon} + 2 \Leftrightarrow \frac{1}{\varepsilon} + 1 < d \Leftrightarrow \frac{1}{\varepsilon} < d - 1 \Leftrightarrow \frac{1}{d-1} < \varepsilon \Leftrightarrow \sum_{n=1}^{\infty} \left(\frac{1}{d}\right)^n < \varepsilon \quad (\text{A.2})$$

We will now construct a subtree $\mathfrak{T}' = (N', E', L')$ such that every infinite path has an infinite number of N_2 nodes and such that

$$|\mathfrak{T}'| \leq |\mathfrak{T}| + \sum_{n=1}^{\infty} \left(\frac{1}{d}\right)^n \quad (\text{A.3})$$

and then, we finally have

$$|\mathfrak{T}'| \stackrel{(A.3)}{\leq} |\mathfrak{T}| + \sum_{n=1}^{\infty} \left(\frac{1}{d}\right)^n = c + \sum_{n=1}^{\infty} \left(\frac{1}{d}\right)^n \stackrel{(A.2)}{<} c + \varepsilon < 1$$

The idea of this construction is that we cut infinite subtrees of pure N_1 nodes as soon as the probability for normal forms is high enough. In this way, one obtains paths where after finitely many N_1 nodes, there is a N_2 node, or we reach a leaf.

The construction works as follows. For any node $x \in N$, let $\mathcal{L}_2(x)$ be the number of N_2 nodes in the path from the root to x . Furthermore, for any set $W \subseteq N$ and $k \in \mathbb{N}$, let $\mathfrak{L}_2(W, k) = \{x \in W \mid \mathcal{L}_2(x) \leq k \vee (x \in N_2 \wedge \mathcal{L}_2(x) \leq k + 1)\}$ be the set of all nodes in W that have at most k nodes from N_2 in the path from the root to its predecessor. So if $x \in \mathfrak{L}_2(W, k)$ is not in N_2 , then we have at most k nodes from N_2 in the path from the root to x and if $x \in \mathfrak{L}_2(W, k)$ is in N_2 , then we have at most $k + 1$ nodes from N_2 in the path from the root to x . We will inductively define a set $U_k \subseteq N$ such that $U_k \subseteq \mathfrak{L}_2(N, k)$ and then define the subtree as $\mathfrak{T}' = \mathfrak{T}[\bigcup_{k \in \mathbb{N}} U_k]$.

We start by considering the subtree $\mathfrak{T}_0 = \mathfrak{T}[\mathfrak{L}_2(N, 0)]$. This tree only contains inner nodes from N_1 . While the node set $\mathfrak{L}_2(N, 0)$ itself may contain nodes from N_2 , they can only occur at the leaves of \mathfrak{T}_0 . Using the prerequisite of the lemma, we get $|\mathfrak{T}_0| = 1$. In Fig. 7 one can see the different possibilities for \mathfrak{T}_0 . Either \mathfrak{T}_0 is finite or \mathfrak{T}_0 is infinite. In the first case, we can add all the nodes to U_0 since there is no infinite path of pure N_1 nodes. Hence, we define $U_0 = \mathfrak{L}_2(N, 0)$. In the second case, we have to cut the tree at a specific depth once the probability of leaves is high enough. Let $d_0(y)$ be the depth of the node y in the tree \mathfrak{T}_0 . Moreover, let $D_0(k) = \{x \in \mathfrak{L}_2(N, 0) \mid d_0(y) \leq k\}$ be the set of nodes in T_0 that have a depth of at most k . Since $|\mathfrak{T}_0| = 1$ and $|\cdot|$ is monotonic w.r.t. the depth of the tree \mathfrak{T}_0 , we can find an $N_0 \in \mathbb{N}$ such that

$$\sum_{x \in \text{Leaf}^{\mathfrak{T}_0}, d_0(x) \leq N_0} p_x^{\mathfrak{T}_0} \geq 1 - \frac{1}{d}$$

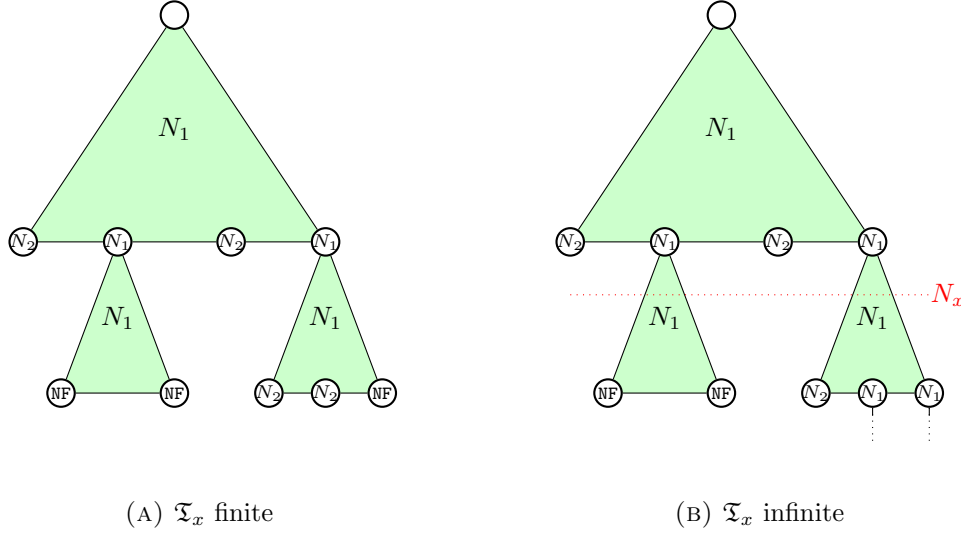
We include all nodes from $D_0(N_0)$ in U_0 and delete every other node of \mathfrak{T}_0 . In other words, we cut the tree after depth N_0 . This cut can be seen in Fig. 7, indicated by the dotted line. We now know that this cut may increase the probability of leaves by at most $\frac{1}{d}$. Therefore, we define $U_0 = D_0(N_0)$ in this case.

For the induction step, assume that we have already defined a subset $U_i \subseteq \mathfrak{L}_2(N, i)$. Let $H_i = \{x \in U_i \mid x \in N_2, \mathcal{L}_2(x) = i + 1\}$ be the set of leaves in $\mathfrak{T}[U_i]$ that are in N_2 . For each $x \in H_i$, we consider the subtree that starts at x until we reach the next node from N_2 , including the node itself. Everything below such a node will be cut. To be precise, we regard the tree $\mathfrak{T}_x = (N_x, E_x, L_x) = \mathfrak{T}[\mathfrak{L}_2(xE^*, i + 1)]$. Here, xE^* is the set of all nodes that are reachable from x by arbitrary many steps.

First, we show that $|\mathfrak{T}_x| = 1$. For every direct successor y of x , the subtree $\mathfrak{T}_y = \mathfrak{T}_x[yE_x^*]$ of \mathfrak{T}_x that starts at y does not contain any inner nodes from N_2 . Hence, we have $|\mathfrak{T}_y| = 1$ by the prerequisite of the lemma, and hence

$$|\mathfrak{T}_x| = \sum_{y \in xE} p_y \cdot |\mathfrak{T}_y| = \sum_{y \in xE} p_y \cdot 1 = \sum_{y \in xE} p_y = 1.$$

For the construction of U_{i+1} , we have the same cases as before, see Fig. 7. Either \mathfrak{T}_x is finite or \mathfrak{T}_x is infinite. Let Z_x be the set of nodes that we want to add to our node set U_{i+1} from the tree \mathfrak{T}_x . In the first case we can add all the nodes again and set $Z_x = N_x$.

Figure 7: Possibilities for \mathfrak{T}_x

In the second case, we once again cut the tree at a specific depth once the probability for leaves is high enough. Let $d_x(z)$ be the depth of the node z in the tree \mathfrak{T}_x . Moreover, let $D_x(k) = \{x \in N_x \mid d_x(z) \leq k\}$ be the set of nodes in \mathfrak{T}_x that have a depth of at most k . Since $|\mathfrak{T}_x| = 1$ and $|\cdot|$ is monotonic w.r.t. the depth of the tree \mathfrak{T}_x , we can find an $N_x \in \mathbb{N}$ such that

$$\sum_{y \in \text{Leaf}^{\mathfrak{T}_x}, d_x(y) \leq N_x} p_y^{\mathfrak{T}_x} \geq 1 - \left(\frac{1}{d}\right)^{i+1} \cdot \frac{1}{|H_i|}$$

We include all nodes from $D_x(N_x)$ in U_{i+1} and delete every other node of \mathfrak{T}_x . In other words, we cut the tree after depth N_x . We now know that this cut may increase the probability of leaves by at most $\left(\frac{1}{d}\right)^{i+1} \cdot \frac{1}{|H_i|}$. Therefore, we set $Z_x = D_x(N_x)$.

We do this for each $x \in H_i$ and in the end, we set $U_{i+1} = U_i \cup \bigcup_{x \in H} Z_x$.

It is straightforward to see that $\bigcup_{k \in \mathbb{N}} U_k$ satisfies the conditions of Def. A.7, as we only cut after certain nodes in our construction. Hence, $\bigcup_{k \in \mathbb{N}} U_k$ is non-empty and weakly connected, and for each of its nodes, it either contains no or all successors. Furthermore, $\mathfrak{T}' = \mathfrak{T}[\bigcup_{k \in \mathbb{N}} U_k]$ is a subtree which does not contain an infinite path of pure N_1 nodes as we cut every such path after a finite depth.

It remains to prove that $|\mathfrak{T}'| \leq |\mathfrak{T}| + \sum_{n=1}^{\infty} \left(\frac{1}{d}\right)^n$ holds. During the i -th iteration of the construction, we may increase the value of $|\mathfrak{T}|$ by the sum of all probabilities corresponding to the new leaves resulting from the cuts. As we cut at most $|H_i|$ trees in the i -th iteration and for each such tree, we added at most a total probability of $\left(\frac{1}{d}\right)^{i+1} \cdot \frac{1}{|H_i|}$ for the new leaves, the value of $|\mathfrak{T}|$ might increase by

$$|H_i| \cdot \left(\frac{1}{d}\right)^{i+1} \cdot \frac{1}{|H_i|} = \left(\frac{1}{d}\right)^{i+1}$$

in the i -th iteration, and hence in total, we then obtain

$$|\mathfrak{T}'| \leq |\mathfrak{T}| + \sum_{n=1}^{\infty} \left(\frac{1}{d}\right)^n,$$

as desired (see (A.3)). \square

With the cutting lemma, we can now prove the following lemma regarding the parallel execution of rewrite sequences that are $\text{AST}_{\mathcal{P}}$.

Lemma A.10 (Parallel Execution Lemma for $\text{AST}_{\mathcal{P}}$). *Let \mathcal{P} be a PTRS and $s \in \mathbb{S}$. Furthermore, let $q_1, \dots, q_n \in \mathcal{T}$ be terms such that for every $\xrightarrow{s}_{\mathcal{P}}$ -RST \mathfrak{T}_i that starts with $(1 : q_i)$ for some $1 \leq i \leq n$ we have $|\mathfrak{T}_i| = 1$. Then, every $\xrightarrow{s}_{\mathcal{P}}$ -RST \mathfrak{T} that starts with $(1 : c(q_1, \dots, q_n))$ for some symbol c , where we do not rewrite at the root position, converges with probability 1.*

Proof. By $\xrightarrow{s \neg \xi}_{\mathcal{P}}$ we denote the restriction of $\xrightarrow{s}_{\mathcal{P}}$ that does not perform any rewrite steps at the root position. We prove that every $\xrightarrow{s \neg \xi}_{\mathcal{P}}$ -RST which starts with $(1 : c(q_1, \dots, q_n))$ for some symbol c converges with probability 1. Note that if we rewrite a term q_i to $\{p_1 : q_{i,1}, \dots, p_k : q_{i,k}\}$, then we obtain a distribution $\{p_1 : c(q_1, \dots, q_{i,1}, \dots, q_n), \dots, p_k : c(q_1, \dots, q_{i,k}, \dots, q_n)\}$. Now, the terms q_j with $j \neq i$ occur multiple times in this distribution, and we may use different rules to rewrite them. Hence, the order in which we rewrite the different q_i matters and cannot be chosen arbitrarily (as seen in Counterex. 4.10). Therefore, the proof is much more complex than in the non-probabilistic setting for termination. (This is not the case for leftmost-innermost rewriting, where an easier induction step than the following one would also be possible.) Let $\xrightarrow{e}_{\mathcal{P}}$ be the restriction of $\xrightarrow{s \neg \xi}_{\mathcal{P}}$ where we can only rewrite on or below a position $1 \leq i \leq e$. Note that $\xrightarrow{n}_{\mathcal{P}} = \xrightarrow{s \neg \xi}_{\mathcal{P}}$. By induction on e , we prove that every $\xrightarrow{e}_{\mathcal{P}}$ -RST which starts with $(1 : c(q_1, \dots, q_n))$ converges with probability 1.

In the base case we have $e = 1$ and only allow rewrite steps on or below position 1. Obviously, since every $\xrightarrow{s}_{\mathcal{P}}$ -RST that starts with $(1 : q_1)$ converges with probability 1 by our assumption, so does every $\xrightarrow{1}_{\mathcal{P}}$ -RST that starts with $(1 : c(q_1, \dots, q_n))$.

In the induction step, we assume that the statement holds for $e - 1 < n$, i.e., every $\xrightarrow{e-1}_{\mathcal{P}}$ -RST which starts with $(1 : c(q_1, \dots, q_n))$ converges with probability 1. Now consider an arbitrary $\xrightarrow{e}_{\mathcal{P}}$ -RST \mathfrak{T} that starts with $(1 : c(q_1, \dots, q_n))$. Assume for a contradiction that it converges with a probability < 1 . The rest of the induction step has the following structure:

- 1) We first use the cutting lemma (Lemma A.9) to get rid of infinite paths that only use rewrite steps on or below position e , resulting in an $\xrightarrow{e}_{\mathcal{P}}$ -RST tree \mathfrak{T}' , which uses infinitely many $\xrightarrow{e-1}_{\mathcal{P}}$ -steps in each infinite path.
- 2) Then, for each finite height $H \in \mathbb{N}$ we split the tree \mathfrak{T}'_H , which consists of the first H layers of the tree \mathfrak{T}' , into multiple (finitely many) $\xrightarrow{e-1}_{\mathcal{P}}$ -RSTs of height at most H that all start with $(1 : c(q_1, \dots, q_n))$. Furthermore, we show that for each $H \in \mathbb{N}$, at least one of these $\xrightarrow{e-1}_{\mathcal{P}}$ -RSTs converges with low enough probability.
- 3) We create an infinite, finitely-branching tree whose nodes are labeled with RSTs. More precisely, its nodes at depth $H \in \mathbb{N}$ are labeled with $\xrightarrow{e-1}_{\mathcal{P}}$ -RSTs with low enough probability.

- 4) Finally, we use König's Lemma to obtain an infinite path in this tree, which corresponds to an infinite $\xrightarrow{e-1}_{\mathcal{P}}$ -RST that starts with $(1 : c(q_1, \dots, q_n))$ and converges with probability < 1 , which is our desired contradiction.

1) Use the cutting lemma

We can partition the inner nodes of our RST \mathfrak{T} into the sets

- $N_1 := \{x \in N^{\mathfrak{T}} \setminus \text{Leaf}^{\mathfrak{T}} \mid \text{the rewrite step at } x \text{ is on or below position } e\}$
- $N_2 := N \setminus N_1 = \{x \in N^{\mathfrak{T}} \setminus \text{Leaf}^{\mathfrak{T}} \mid \text{the rewrite step at } x \text{ is on or below position } k \text{ with } 1 \leq k < e\}$

We know that every $\xrightarrow{e}_{\mathcal{P}}$ -RST that starts with $(1 : q_e)$ converges with probability 1 by assumption. Hence, also every subtree of \mathfrak{T} that only contains nodes from N_1 as inner nodes converges with probability 1. Thus, we can apply the cutting lemma (Lemma A.9) and obtain a subtree \mathfrak{T}' of \mathfrak{T} that converges with probability < 1 and contains infinitely many nodes of N_2 in each infinite path.

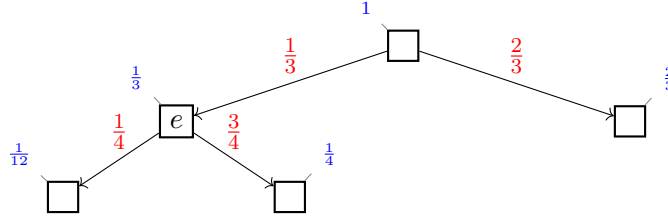
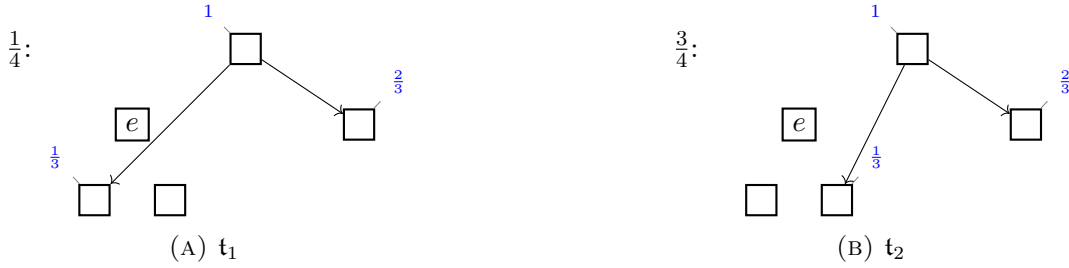
2) Split the tree \mathfrak{T}_H for every $H \in \mathbb{N}$

Let $H \in \mathbb{N}$ and let \mathfrak{T}'_H be the finite tree consisting of the first H layers of \mathfrak{T}' , where the subterm on position e remains the same in every node and is equal to the subterm on position e in the root of \mathfrak{T}' . Since \mathfrak{T}' converges with probability < 1 , there exists an $0 < \alpha \leq 1$ such that $|\mathfrak{T}'| = \lim_{H \rightarrow \infty} \sum_{x \in \text{Leaf}^{\mathfrak{T}'} \wedge d(x) \leq H} p_x^{\mathfrak{T}'} < 1 - \alpha$, and hence $\sum_{x \in \text{Leaf}^{\mathfrak{T}'} \wedge d(x) \leq H} p_x^{\mathfrak{T}'} = \sum_{x \in \text{Leaf}^{\mathfrak{T}'} \wedge x \in \text{Leaf}^{\mathfrak{T}'_H}} p_x^{\mathfrak{T}'_H} < 1 - \alpha$ for every $H \in \mathbb{N}$. Note that \mathfrak{T}'_H is not a valid $\xrightarrow{e}_{\mathcal{P}}$ -RST, as we now have steps where no term changes, i.e., where we would have performed a rewrite step below position e .

From \mathfrak{T}'_H we generate a set \mathbb{T}_H of pairs (p_t, t) where $p_t \in (0, 1]$ is a probability and t is an $\xrightarrow{e-1}_{\mathcal{P}}$ -RST of height at most H (i.e., we do not perform any rewrite steps on or below position e anymore). The set \mathbb{T}_H contains all $\xrightarrow{e-1}_{\mathcal{P}}$ -RSTs t resulting from \mathfrak{T}'_H when skipping the nodes of N_1 (where the rewrite step is on or below the position e). Instead, one uses one of its child nodes. Moreover, p_t is the probability for choosing the RST t . We will give more intuition on \mathbb{T}_H below. Since a rewrite step on or below the position e may create several children (one for each term in the support of the multi-distribution on the right-hand side of the applied rewrite rule), the set \mathbb{T}_H may contain several pairs. \mathbb{T}_H will satisfy the following two properties:

- (Prop-1) We have $\sum_{(p_t, t) \in \mathbb{T}_H} p_t = 1$. This means that the sum of the probabilities for all trees in \mathbb{T}_H sum up to one.
- (Prop-2) For all $x \in \text{Leaf}^{\mathfrak{T}'_H} \cup N_2$ we have $p_x^{\mathfrak{T}'_H} = \sum_{(p_t, t) \in \mathbb{T}_H} p_t \cdot p_x^t$. As mentioned above, p_t represents the probability that the RST t is chosen and p_x^t is the probability of the node x in the RST t . Whenever x is not a node of t (i.e., $x \notin N^t$), then we define $p_x^t = 0$. This means that the probability for a leaf $x \in \text{Leaf}^{\mathfrak{T}'_H}$ or an inner node $x \in N_2$ in our cut tree \mathfrak{T}'_H is equal to the sum over all trees t that contain x , where we multiply the probability p_t of the tree t by the probability of node x in t .

Assume now that for every $(p_t, t) \in \mathbb{T}_H$ we have $\sum_{x \in \text{Leaf}^t \wedge x \in \text{Leaf}^{\mathfrak{T}'}} p_x^t \geq 1 - \alpha$. Then, with the two properties (Prop-1) and (Prop-2) we obtain the following:

Figure 8: Example tree \mathfrak{T}'_2 with a step on or below position e in the left nodeFigure 9: The set $\mathbb{T}_2 = \{(\frac{1}{4}, t_1), (\frac{3}{4}, t_2)\}$

$$\begin{aligned}
& \text{(by (Prop-2))} & = & \sum_{x \in \text{Leaf}^{\mathfrak{T}'_H} \wedge x \in \text{Leaf}^{\mathfrak{T}'}} p_x^{\mathfrak{T}'_H} \\
& & = & \sum_{x \in \text{Leaf}^{\mathfrak{T}'_H} \wedge x \in \text{Leaf}^{\mathfrak{T}'}} \sum_{(p_t, t) \in \mathbb{T}_H} p_t \cdot p_x^t \\
& & = & \sum_{(p_t, t) \in \mathbb{T}_H} \sum_{x \in \text{Leaf}^{\mathfrak{T}'_H} \wedge x \in \text{Leaf}^{\mathfrak{T}'}} p_t \cdot p_x^t \\
& & = & \sum_{(p_t, t) \in \mathbb{T}_H} p_t \cdot \sum_{x \in \text{Leaf}^{\mathfrak{T}'_H} \wedge x \in \text{Leaf}^{\mathfrak{T}'}} p_x^t \\
& & = & \sum_{(p_t, t) \in \mathbb{T}_H} p_t \cdot \sum_{x \in \text{Leaf}^t \wedge x \in \text{Leaf}^{\mathfrak{T}'}} p_x^t \\
& \text{(by assumption)} & \geq & \sum_{(p_t, t) \in \mathbb{T}_H} p_t \cdot (1 - \alpha) \\
& & = & (1 - \alpha) \cdot \sum_{(p_t, t) \in \mathbb{T}_H} p_t \\
& \text{(by (Prop-1))} & = & (1 - \alpha) \cdot 1 \\
& & = & 1 - \alpha
\end{aligned} \tag{A.4}$$

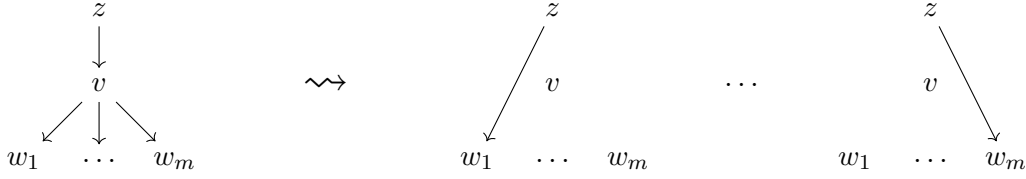
which is a contradiction. Hence, for each $H \in \mathbb{N}$ there exists a $(p_t, t) \in \mathbb{T}_H$ such that

$$\sum_{x \in \text{Leaf}^t \wedge x \in \text{Leaf}^{\mathfrak{T}'}} p_x^t < 1 - \alpha \tag{A.5}$$

We will use this in Step 3) and 4) to generate our infinite $\xrightarrow{\mathcal{P}}^{e-1}$ -RST that converges with probability < 1 . But first we have to define the set \mathbb{T}_H that satisfies (Prop-1) and (Prop-2).

Idea of \mathbb{T}_H

The idea of the split can be seen in Fig. 8 and 9. There, the probabilities of the nodes are indicated by the **small numbers** and the probabilities for the edges are indicated by the **big numbers**. Whenever we encounter a node from N_1 with successors y_1, \dots, y_m , meaning that we rewrite on or below position e , then we split the tree into m different trees, since the terms below positions 1 to $e - 1$ did not change, and we may use different rules on the different resulting terms afterwards.

Figure 10: Skipping inner node $v \in N_2$ to create m different trees**Constructing \mathbb{T}_H**

To construct \mathbb{T}_H , we repeatedly remove all nodes of N_1 from the tree \mathfrak{T}'_H and create a set M inductively that satisfies the following properties.

(Ind-1) $\sum_{(p_t, t) \in M} p_t = 1$.

(Ind-2) For all $(p_t, t) \in M$ and all inner nodes $x \in N^t$ with successors y_1, \dots, y_k in t , the edge relation either represents a valid rewrite step on or below a position $1 \leq j \leq e-1$, or the subterms on or below the positions $1 \leq j \leq e-1$ remain the same, but the probabilities of the successors still sum up to the probability of node x , i.e., $\sum_{j=1}^k p_{y_j}^t = p_x^t$.

(Ind-3) For all $x \in \text{Leaf}^{\mathfrak{T}'_H} \cup N_2$ we have $p_x^{\mathfrak{T}'_H} = \sum_{(p_t, t) \in M} p_t \cdot p_x^t$.

We stop the construction once every $t \in M$ is a valid $\xrightarrow{e-1}$ -RST, i.e., once we have removed all nodes where the terms do not change as we would have performed a rewrite step on or below position e . In the end, we result in a set M that satisfies (Ind-1), (Ind-2), and (Ind-3), and t is a valid $\xrightarrow{e-1}$ -RST for every $(p_t, t) \in M$. Then M is our desired set, and we define $\mathbb{T}_H = M$. Moreover, in the end, (Prop-1) and (Prop-2) follow from (Ind-1) and (Ind-3).

We start with $M = \{(1, \mathfrak{T}'_H)\}$. Here, clearly all three properties (Ind-1)-(Ind-3) are satisfied. Now, assume that there is still a pair $(p_l, l) \in M$ such that l contains a node $v \in N_1$ that is not a leaf in l . We will now split l into multiple trees that do not contain v anymore but move directly to one of its children, as illustrated in Fig. 10.

First, assume that v is not the root of l . Let $vE^l = \{w_1, \dots, w_m\}$ be the direct successors of v in l and let z be the predecessor of v in l . Instead of one tree l with the edges $(z, v), (v, w_1), \dots, (v, w_m)$, we split the tree into m different trees l_1, \dots, l_m such that for every $1 \leq h \leq m$, the tree l_h contains a direct edge from z to w_h . In addition to that, the unreachable nodes are removed, and we also have to adjust the probabilities of all (not necessarily direct) successors of w_h (including w_h itself) in l_h . More precisely, we set $l_h = (N^{l_h}, E^{l_h}, L^{l_h})$, with

$$\begin{aligned} N^{l_h} &= (N^l \setminus v(E^l)^*) \cup w_h(E^l)^* \\ E^{l_h} &= (E^l \setminus (v(E^l)^* \times v(E^l)^*)) \cup \{(z, w_h)\} \cup (E^l \cap (w_h(E^l)^* \times w_h(E^l)^*)) \end{aligned}$$

Furthermore, let $p_h = \frac{p_{w_h}^l}{p_v^l}$. Then, the labeling is defined by

$$L^{l_h}(x) = \begin{cases} (\frac{1}{p_h} \cdot p_x^l, t_x^l) & \text{if } x \in w_h(E^l)^* \\ (p_x^l, t_x^l) & \text{otherwise} \end{cases}$$

Note that

$$\sum_{1 \leq h \leq m} p_h = \sum_{1 \leq h \leq m} \frac{p_{w_h}^l}{p_v^l} = \frac{1}{p_v^l} \cdot \sum_{1 \leq h \leq m} p_{w_h}^l \stackrel{(Ind-2)}{=} \frac{1}{p_v^l} \cdot p_v^l = 1 \quad (\text{A.6})$$

If v is the root of \mathfrak{l} , then we use the same construction, but we have no predecessor z of v and directly start with the node w_h as the new root. Hence, we have to use the node set $N^{l_h} = w_h(E^l)^*$ and edge relation $E^{l_h} = (E \cap (w_h(E^l)^* \times w_h(E^l)^*))$ and the rest stays the same.

In the end, we set

$$M' = M \setminus \{(p_l, \mathfrak{l})\} \cup \{(p_l \cdot p_1, \mathfrak{l}_1), \dots, (p_l \cdot p_m, \mathfrak{l}_m)\}$$

This construction is exactly what we did in Fig. 8 and Fig. 9 for the only node where we would have rewritten on or below position e . It remains to prove that (Ind-1), (Ind-2), and (Ind-3) are still satisfied for M' .

(Ind-1): We have

$$\begin{aligned} & \sum_{(p_t, \mathfrak{t}) \in M'} p_t \\ &= \sum_{(p_t, \mathfrak{t}) \in M \setminus \{(p_l, \mathfrak{l})\}} p_t + \sum_{(p_t, \mathfrak{t}) \in \{(p_l \cdot p_1, \mathfrak{l}_1), \dots, (p_l \cdot p_m, \mathfrak{l}_m)\}} p_t \\ &= \sum_{(p_t, \mathfrak{t}) \in M \setminus \{(p_l, \mathfrak{l})\}} p_t + \sum_{1 \leq h \leq m} p_l \cdot p_h \\ &= \sum_{(p_t, \mathfrak{t}) \in M \setminus \{(p_l, \mathfrak{l})\}} p_t + p_l \cdot \sum_{1 \leq h \leq m} p_h \\ (\text{by (A.6)}) &= \sum_{(p_t, \mathfrak{t}) \in M \setminus \{(p_l, \mathfrak{l})\}} p_t + p_l \cdot 1 \\ &= \sum_{(p_t, \mathfrak{t}) \in M \setminus \{(p_l, \mathfrak{l})\}} p_t + p_l \\ &= \sum_{(p_t, \mathfrak{t}) \in M} p_t \\ &\stackrel{IH}{=} 1 \end{aligned}$$

(Ind-2): Let $1 \leq h \leq m$. We only split nodes whenever we would have rewritten on or below position e . Hence, we only have to prove that in this case the probabilities of the successors for a node add up to the probability for the node itself with our new labeling. We constructed \mathfrak{l}_h by skipping the node v and directly moving from z to w_h (or starting with w_h if v was the root node). Hence, (N^{l_h}, E^{l_h}) is still a finitely branching tree. Let $x \in N^{l_h}$ with $xE^{l_h} \neq \emptyset$. If $x \in w_h(E^l)^*$, then $xE^{l_h} = xE^l$ and thus

$$\sum_{y \in xE^{l_h}} p_y^{l_h} = \sum_{y \in xE^{l_h}} \frac{1}{p_h} \cdot p_y^l = \frac{1}{p_h} \cdot \sum_{y \in xE^{l_h}} p_y^l = \frac{1}{p_h} \cdot \sum_{y \in xE^l} p_y^l = \frac{1}{p_h} \cdot p_x^l = p_x^{l_h}$$

If v was not the root and $x = z$, then $zE^{l_h} = (zE^l \setminus \{v\}) \cup \{w_h\}$ and thus

$$\begin{aligned} \sum_{y \in zE^{l_h}} p_y^{l_h} &= \sum_{y \in (zE^l \setminus \{v\}) \cup \{w_h\}} p_y^{l_h} = \sum_{y \in (zE^l \setminus \{v\})} p_y^{l_h} + p_{w_h}^{l_h} = \sum_{y \in (zE^l \setminus \{v\})} p_y^l + \frac{1}{p_h} \cdot p_{w_h}^l \\ &= \sum_{y \in (zE^l \setminus \{v\})} p_y^l + \frac{p_v^l}{p_{w_h}^l} \cdot p_{w_h}^l = \sum_{y \in (zE^l \setminus \{v\})} p_y^l + p_v^l = \sum_{y \in zE^l} p_y^l = p_z^l = p_z^{l_h} \end{aligned}$$

Otherwise, we have $x \in N^l \setminus (v(E^l)^* \cup \{z\})$. This means $p_y^{l_h} = p_y^l$ for all $y \in xE^{l_h}$ and $xE^{l_h} = xE^l$, and thus

$$\sum_{y \in xE^{l_h}} p_y^{l_h} = \sum_{y \in xE^{l_h}} p_y^l = \sum_{y \in xE^l} p_y^l = p_x^l = p_x^{l_h}$$

For the last property ($p_x^l = p_x^{l_h}$), note that if v is not the root in l , then the root and its labeling did not change, so that we have $p_{\mathfrak{r}^{l_h}}^{l_h} = p_{\mathfrak{r}^l}^l = 1$, where $\mathfrak{r}^{l_h} = \mathfrak{r}^l$ is the root of l_h and l . If v was the root, then w_h is the new root with

$$p_{w_h}^{l_h} = \frac{1}{p_h} \cdot p_{w_h}^l = \frac{p_v^l}{p_{w_h}^l} \cdot p_{w_h}^l = p_v^l = 1$$

(Ind-3): Let $x \in \text{Leaf}^{\mathfrak{T}'_H} \cup N_2$. If we have $x \notin N^l$, then also $x \notin N^{l_h}$ for all $1 \leq h \leq m$ and thus

$$\begin{aligned} & \sum_{(p_t, t) \in M'} p_t \cdot p_x^t \\ = & \sum_{(p_t, t) \in M \setminus \{(p_l, l)\} \cup \{(p_l, p_1, l_1), \dots, (p_l, p_m, l_m)\}} p_t \cdot p_x^t \\ = & \sum_{(p_t, t) \in M \setminus \{(p_l, l)\}} p_t \cdot p_x^t + \sum_{(p_t, t) \in \{(p_l, p_1, l_1), \dots, (p_l, p_m, l_m)\}} p_t \cdot p_x^t \\ = & \sum_{(p_t, t) \in M \setminus \{(p_l, l)\}} p_t \cdot p_x^t + \sum_{(p_t, t) \in \{(p_l, p_1, l_1), \dots, (p_l, p_m, l_m)\}} p_t \cdot 0 \\ = & \sum_{(p_t, t) \in M \setminus \{(p_l, l)\}} p_t \cdot p_x^t \\ = & \sum_{(p_t, t) \in M \setminus \{(p_l, l)\}} p_t \cdot p_x^t + p_l \cdot 0 \\ = & \sum_{(p_t, t) \in M \setminus \{(p_l, l)\}} p_t \cdot p_x^t + p_l \cdot p_x^l \\ = & \sum_{(p_t, t) \in M} p_t \cdot p_x^t \\ \stackrel{IH}{=} & p_x^{\mathfrak{T}'_H} \end{aligned}$$

If we have $x \in N^l$ and $x \notin v(E^l)^*$, then $x \in N^{l_h}$ for all $1 \leq h \leq m$ and $p_x^l = p_x^{l_h}$. Hence

$$\begin{aligned} & \sum_{(p_t, t) \in M'} p_t \cdot p_x^t \\ = & \sum_{(p_t, t) \in M \setminus \{(p_l, l)\} \cup \{(p_l, p_1, l_1), \dots, (p_l, p_m, l_m)\}} p_t \cdot p_x^t \\ = & \sum_{(p_t, t) \in M \setminus \{(p_l, l)\}} p_t \cdot p_x^t + \sum_{(p_t, t) \in \{(p_l, p_1, l_1), \dots, (p_l, p_m, l_m)\}} p_t \cdot p_x^t \\ = & \sum_{(p_t, t) \in M \setminus \{(p_l, l)\}} p_t \cdot p_x^t + \sum_{1 \leq h \leq m} p_l \cdot p_h \cdot p_x^{l_h} \\ = & \sum_{(p_t, t) \in M \setminus \{(p_l, l)\}} p_t \cdot p_x^t + \sum_{1 \leq h \leq m} p_l \cdot p_h \cdot p_x^l \\ = & \sum_{(p_t, t) \in M \setminus \{(p_l, l)\}} p_t \cdot p_x^t + p_l \cdot p_x^l \cdot \sum_{1 \leq h \leq m} p_h \\ = & \sum_{(p_t, t) \in M \setminus \{(p_l, l)\}} p_t \cdot p_x^t + p_l \cdot p_x^l \cdot 1 \\ = & \sum_{(p_t, t) \in M} p_t \cdot p_x^t \\ \stackrel{IH}{=} & p_x^{\mathfrak{T}'_H} \end{aligned}$$

Otherwise we have $x \in N^l$ and $x \in v(E^l)^*$. This means that we have $x \in N^{l_h}$ and $x \in w_h(E^l)^*$ for some $1 \leq h \leq m$ and $x \notin N^{l_{h'}}$ for all $h' \neq h$. Furthermore, we have $p_x^{l_h} = \frac{1}{p_h} \cdot p_x^l$, and hence

$$\begin{aligned} & \sum_{(p_t, t) \in M'} p_t \cdot p_x^t \\ = & \sum_{(p_t, t) \in M \setminus \{(p_l, l)\} \cup \{(p_l, p_1, l_1), \dots, (p_l, p_m, l_m)\}} p_t \cdot p_x^t \\ = & \sum_{(p_t, t) \in M \setminus \{(p_l, t)\}} p_t \cdot p_x^t + \sum_{(p_t, t) \in \{(p_l, p_1, l_1), \dots, (p_l, p_m, l_m)\}} p_t \cdot p_x^t \\ = & \sum_{(p_t, t) \in M \setminus \{(p_l, t)\}} p_t \cdot p_x^t + p_l \cdot p_h \cdot p_x^{l_h} \\ = & \sum_{(p_t, t) \in M \setminus \{(p_l, t)\}} p_t \cdot p_x^t + p_l \cdot p_h \cdot \frac{1}{p_h} \cdot p_x^l \\ = & \sum_{(p_t, t) \in M \setminus \{(p_l, t)\}} p_t \cdot p_x^t + p_l \cdot p_x^l \\ = & \sum_{(p_t, t) \in M} p_t \cdot p_x^t \\ \stackrel{IH}{=} & p_x^{\mathfrak{T}'_H} \end{aligned}$$

3) Create the finitely-branching infinite tree of RSTs

Next, we create a tree \mathfrak{F} whose nodes are labeled with RSTs. More precisely, its nodes at depth H represent $\xrightarrow{e-1}_{\mathcal{P}}$ -RSTs $t \in \mathbb{T}_H$ that converge with a small enough probability, i.e.,

$\sum_{x \in \text{Leaf}^t \wedge x \in \text{Leaf}^{\mathfrak{T}'}} p_x^t < 1 - \alpha$. The root of \mathfrak{F} is labeled with the subtree \mathfrak{T}'_0 of \mathfrak{T}' that only consists of its root. \mathfrak{T}'_0 is a finite subtree and $\sum_{x \in \text{Leaf}^{\mathfrak{T}'_0} \wedge x \in \text{Leaf}^{\mathfrak{T}'}} p_x^{\mathfrak{T}'_0} = 0 \leq 1 - \alpha$, since the root of \mathfrak{T}' cannot be a leaf in \mathfrak{T}' .

Let $H \in \mathbb{N}$ with $H > 1$. For the tree \mathfrak{F} , we have a node at depth H for every tree $t \in \mathbb{T}_H$ such that $\sum_{x \in \text{Leaf}^t \wedge x \in \text{Leaf}^{\mathfrak{T}'}} p_x^T < 1 - \alpha$. We draw an edge from a node X at depth $H - 1$ to the node Y at depth H if the corresponding RSTs in the labels are the same, or if the RST for node Y is an extension of the RST for node X (i.e., we evaluate the leaves in the RST of X further to obtain Y).

Each \mathbb{T}_H for every $H \in \mathbb{N}$ is finite, so \mathfrak{F} is finitely branching. Furthermore, there is a node in each layer of the tree \mathfrak{F} , since for every $H \in \mathbb{N}$, we can find an $(p_t, t) \in \mathbb{T}_H$ such that $\sum_{x \in \text{Leaf}^t \wedge x \in \text{Leaf}^{\mathfrak{T}'}} p_x^t < 1 - \alpha$, see (A.5).

4) Use König's Lemma

Now we have an infinite tree \mathfrak{F} that is finitely branching, which means that the tree has an infinite path by König's Lemma. This path represents an $\xrightarrow{e-1}_{\mathcal{P}}$ -RST that does not converge with probability 1, which is our desired contradiction. To see this, let t_1, t_2, \dots be the finite $\xrightarrow{e-1}_{\mathcal{P}}$ -RSTs in the labels of the nodes in the infinite path in \mathfrak{F} , where t_i is a prefix of t_{i+1} for all $i \geq 1$. Hence, we can define the tree $t_{\text{lim}} = \lim_{i \rightarrow \infty} t_i$. Furthermore, there exists no infinite path in \mathfrak{F} such that the sequence t_1, t_2, \dots eventually stays the same finite tree forever, due to the fact that there are no infinite paths in \mathfrak{T}' that eventually only contain nodes from N_1 (due to the application of the cutting lemma to create \mathfrak{T}'). Finally, we have $|t_{\text{lim}}| = \lim_{i \rightarrow \infty} \sum_{x \in \text{Leaf}^{t_i} \wedge x \in \text{Leaf}^{\mathfrak{T}'}} p_x^{t_i} < 1 - \alpha$ since $\sum_{x \in \text{Leaf}^{t_i} \wedge x \in \text{Leaf}^{\mathfrak{T}'}} p_x^{t_i} < 1 - \alpha$ holds for all $i \in \mathbb{N}$. \square

We also obtain a corresponding parallel execution lemma for $\text{SAST}_{\rightarrow \mathcal{P}}$. Note that the parallel execution lemma does not hold for $\text{PAST}_{\rightarrow \mathcal{P}}$, see Ex. 3.15.

Lemma A.11 (Parallel Execution Lemma for $\text{SAST}_{\rightarrow \mathcal{P}}$). *Let \mathcal{P} be a PTRS and $s \in \mathbb{S}$. Furthermore, let $q_1, \dots, q_n \in \mathcal{T}$ be terms and $C_1, \dots, C_n \in \mathbb{R}$ constants such that for every $1 \leq i \leq n$ and every $\rightarrow_{\mathcal{P}}$ -RST \mathfrak{T}_i that starts with $(1 : q_i)$ we have $\text{edl}(\mathfrak{T}_i) \leq C_i$. Then, every $\rightarrow_{\mathcal{P}}$ -RST \mathfrak{T} that starts with $(1 : c(q_1, \dots, q_n))$ for some symbol c , where we do not use rewrite steps at the root position, has a finite expected derivation length, which is bounded by $\sum_{i=0}^n C_i \in \mathbb{R}$, i.e., $\text{edl}(\mathfrak{T}) \leq \sum_{i=0}^n C_i$.*

Proof. We use the same induction on e as in the proof for the parallel execution lemma for $\text{AST}_{\rightarrow \mathcal{P}}$ (Lemma A.10). Note that we now need to prove an upper bound (the expected derivation length is smaller than $\sum_{i=0}^n C_i$), while for $\text{AST}_{\rightarrow \mathcal{P}}$ we had to prove a lower bound (probability of convergence is (at least) 1). Therefore, we do not use the cutting lemma and the proof is a bit different in the induction step, while the base case is again trivial.

In the induction step, we assume that the statement holds for $e - 1$, i.e., there exists a bound $C'_{e-1} = \sum_{i=0}^{e-1} C_i \in \mathbb{R}$ such that for every $\xrightarrow{e-1}_{\mathcal{P}}$ -RST \mathfrak{T} that starts with $(1 : c(q_1, \dots, q_n))$ we have $\text{edl}(\mathfrak{T}) \leq C'_{e-1}$. Now consider an arbitrary $\xrightarrow{e}_{\mathcal{P}}$ -RST \mathfrak{T} . We prove that its expected derivation length is bounded by $C'_{e-1} + C_e$.

Let $H \in \mathbb{N}$ and let \mathfrak{T}_H be the tree consisting of the first H layers of \mathfrak{T} . We partition the inner nodes of \mathfrak{T}_H into N_1 and N_2 , analogous to the proof of the parallel execution lemma for $\text{AST}_{\rightarrow \mathcal{P}}$ (Lemma A.10). As in that proof, for each $H \in \mathbb{N}$ we split the tree \mathfrak{T}_H into multiple

(finitely many) $\xrightarrow{e-1}_{\mathcal{P}}$ -RSTs of height at most H that all start with $(1 : c(q_1, \dots, q_n))$. This again leads to a set of pairs \mathbb{T}_H such that:

(Prop-1) $\sum_{(p_t, t) \in \mathbb{T}_H} p_t = 1$

(Prop-2) For all $x \in N_2$ we have $p_x^{\mathfrak{T}_H} = \sum_{(p_t, t) \in \mathbb{T}_H} p_t \cdot p_x^t$

using the same notation as before. Furthermore, we can use the same construction to create a set \mathbb{T}_H^e containing (finitely many) pairs of probabilities and $\xrightarrow{\cdot}_{\mathcal{P}}$ -RSTs that start with $(1 : q_e)$ of height at most H , by simply switching when to split the tree and when to perform the rewrite step. To be precise, we split the tree if we encounter a rewrite step on or below a position $1 \leq j \leq e-1$, and perform the rewrite step if it is on or below position e . Again, we get

(Prop-1-e) $\sum_{(p_t, t) \in \mathbb{T}_H^e} p_t = 1$

(Prop-2-e) For all $x \in N_1$ we have $p_x^{\mathfrak{T}_H} = \sum_{(p_t, t) \in \mathbb{T}_H^e} p_t \cdot p_x^t$

Now we can bound the expected runtime after height H by $C'_{e-1} + C_e$ for every $H \in \mathbb{N}$, as we have

$$\begin{aligned}
 & \text{edl}(\mathfrak{T}_H) \\
 &= \sum_{x \in N^{\mathfrak{T}_H} \setminus \text{Leaf}^{\mathfrak{T}_H}} p_x^{\mathfrak{T}_H} \\
 &= \sum_{x \in N_1} p_x^{\mathfrak{T}_H} + \sum_{x \in N_2} p_x^{\mathfrak{T}_H} \\
 &= \sum_{x \in N_1} \sum_{(p_t, t) \in \mathbb{T}_H^e} p_t \cdot p_x^t \\
 &\quad + \sum_{x \in N_2} \sum_{(p_t, t) \in \mathbb{T}_H} p_t \cdot p_x^t \\
 &= \sum_{(p_t, t) \in \mathbb{T}_H^e} p_t \cdot \sum_{x \in N^t \setminus \text{Leaf}^t} p_x^t \\
 &\quad + \sum_{(p_t, t) \in \mathbb{T}_H} p_t \cdot \sum_{x \in N^t \setminus \text{Leaf}^t} p_x^t \\
 &\leq \sum_{(p_t, t) \in \mathbb{T}_H^e} p_t \cdot C'_{e-1} \\
 &\quad + \sum_{(p_t, t) \in \mathbb{T}_H} p_t \cdot C_e \\
 &= C'_{e-1} \cdot \sum_{(p_t, t) \in \mathbb{T}_H^e} p_t \\
 &\quad + C_e \cdot \sum_{(p_t, t) \in \mathbb{T}_H} p_t \\
 &= C'_{e-1} \cdot 1 + C_e \cdot 1 \\
 &= C'_{e-1} + C_e
 \end{aligned}$$

(since $N_1 \uplus N_2 = N^{\mathfrak{T}_H} \setminus \text{Leaf}^{\mathfrak{T}_H}$)
 (by (Prop-2-e) and (Prop-2))
 (as in (A.4))
 (by assumption and induction hypothesis)
 (by (Prop-1-e) and (Prop-1))

This gives us $\text{edl}(\mathfrak{T}) = \lim_{H \rightarrow \infty} \text{edl}(\mathfrak{T}_H) \leq C'_{e-1} + C_e$, as desired. \square

With the parallel execution lemmas for $\text{AST}_{\rightarrow \mathcal{P}}$ and $\text{SAST}_{\rightarrow \mathcal{P}}$ we can finally prove our modularity results. We start with $\text{AST}_{\rightarrow \mathcal{P}}$.

Theorem 7.2 (Modularity of $\text{AST}_{\rightarrow \mathcal{P}}$ for Disjoint Unions). *Let $\mathcal{P}^{(1)}$ and $\mathcal{P}^{(2)}$ be PTRSs with $\Sigma^{\mathcal{P}^{(1)}} \cap \Sigma^{\mathcal{P}^{(2)}} = \emptyset$. Then we have:*

$$\text{AST}_{\rightarrow \mathcal{P}^{(1)} \cup \mathcal{P}^{(2)}} \iff \text{AST}_{\rightarrow \mathcal{P}^{(1)}} \text{ and } \text{AST}_{\rightarrow \mathcal{P}^{(2)}}$$

Proof. The direction “ \implies ” is trivial and thus, we only prove “ \impliedby ”. So let $\mathcal{P} = \mathcal{P}^{(1)} \cup \mathcal{P}^{(2)}$, where both $\text{AST}_{\rightarrow \mathcal{P}^{(1)}}$ and $\text{AST}_{\rightarrow \mathcal{P}^{(2)}}$ hold.

By Lemma A.3, for $\text{AST}_{\rightarrow \mathcal{P}}$ it suffices to regard only rewrite sequences that start with multi-distributions of the form $\{1 : t\}$. Thus, we show by structural induction on the term structure that for every $t \in \mathcal{T}(\Sigma^{\mathcal{P}}, \mathcal{V})$, all $\xrightarrow{\cdot}_{\mathcal{P}}$ -rewrite sequences starting with $\{1 : t\}$ converge with probability 1.

If $t \in \mathcal{V}$, then t is in normal form. If t is a constant, then w.l.o.g. let $t \in \mathcal{P}^{(1)}$. Since we have $\text{AST} \xrightarrow{\mathcal{P}^{(1)}} \{1 : t\}$, $\{1 : t\}$ cannot start an infinite $\xrightarrow{\mathcal{P}}$ -rewrite sequence that converges with probability < 1 .

Now we regard the induction step, and consider the case where $t = f(q_1, \dots, q_n)$. By the induction hypothesis, every $\xrightarrow{\mathcal{P}}$ -RST \mathfrak{T} that starts with $(1 : q_i)$ for some $1 \leq i \leq k$ converges with probability 1. Let \mathfrak{T} be a fully evaluated $\xrightarrow{\mathcal{P}}$ -RST that starts with $(1 : f(q_1, \dots, q_n))$. We prove that for every $0 < \delta < 1$ we can find an $M \in \mathbb{N}$ such that $\sum_{x \in \text{Leaf}^{\mathfrak{T}}, d(x) \leq N} p_x^{\mathfrak{T}} > 1 - \delta$, which means that $|\mathfrak{T}| = \lim_{k \rightarrow \infty} \sum_{x \in \text{Leaf}^{\mathfrak{T}}, d(x) \leq k} p_x^{\mathfrak{T}} = 1$. (Recall that for proving $\text{AST} \xrightarrow{\mathcal{P}}$, it suffices to consider only fully evaluated RSTs, see Cor. A.4.)

Let $0 < \delta < 1$. By the induction hypothesis and the parallel execution lemma (Lemma A.10), the maximal subtree $\mathfrak{T}_{-\varepsilon}$ of \mathfrak{T} that starts with \mathfrak{T} 's root node and only performs rewrite steps at non-root positions converges with probability 1. Since $|\mathfrak{T}_{-\varepsilon}| = 1$, there exists a depth H such that $\sum_{x \in \text{Leaf}^{\mathfrak{T}_{-\varepsilon}}, d(x) \leq H} p_x^{\mathfrak{T}} \geq \beta$ with $\beta = \sqrt{1 - \delta}$. Let \mathfrak{T}_H be the tree resulting from cutting the tree \mathfrak{T} at depth H , and let $Z^{\mathfrak{T}_H}$ be the set of leaves in \mathfrak{T}_H that were already leaves in $\mathfrak{T}_{-\varepsilon}$. So we have $\sum_{x \in Z^{\mathfrak{T}_H}} p_x^{\mathfrak{T}} = \sum_{x \in \text{Leaf}^{\mathfrak{T}_{-\varepsilon}}, d(x) \leq H} p_x^{\mathfrak{T}} \geq \beta$. For each leaf $x \in Z^{\mathfrak{T}_H}$, every proper subterm of t_x is in normal form w.r.t. \mathcal{P} . This is due to the fact that we use an innermost rewrite strategy and the RST \mathfrak{T} is fully evaluated. Let us look at the induced subtree \mathfrak{T}_x of \mathfrak{T} that starts at x (i.e., $\mathfrak{T}_x = \mathfrak{T}[xE^*]$). W.l.o.g., let the root symbol f of t_x be from $\Sigma^{\mathcal{P}^{(1)}}$. Let $\mathfrak{T}^{(1)}$ result from \mathfrak{T}_x by labeling the root with t'_x , where t'_x results from t_x by replacing all its maximal (i.e., topmost) subterms with root symbols from $\Sigma^{\mathcal{P}^{(2)}}$ by fresh variables (using the same variable for the same subterm). Obviously, since both PTRSs have disjoint signatures and all proper subterms of t_x are in normal form, we can still apply the same rules as in \mathfrak{T}_x , such that $\mathfrak{T}^{(1)}$ is an $\mathcal{P}^{(1)}$ -RST with $|\mathfrak{T}^{(1)}| = |\mathfrak{T}_x|$. Since every $\mathcal{P}^{(1)}$ -RST converges with probability 1, we obtain $1 = |\mathfrak{T}^{(1)}| = |\mathfrak{T}_x|$.

For any node y of \mathfrak{T}_x , let $d_x(y)$ be the depth of the node y in the tree \mathfrak{T}_x . Moreover, let $D_x(k) = \{y \in N^{\mathfrak{T}_x} \mid d_x(y) \leq k\}$ be the set of nodes in \mathfrak{T}_x that have a depth of at most k . Since $|\mathfrak{T}_x| = 1$ and $|\cdot|$ increases weakly monotonically with the depth of the tree, we can find an $M_x \in \mathbb{N}$ such that $|\mathfrak{T}_x[D_x(M_x)]| > \beta$.

Note that $Z^{\mathfrak{T}_H}$ is finite and thus $M_{\max} = \max\{M_x \mid x \in Z^{\mathfrak{T}_H}\}$ exists. Now, for $N = H + M_{\max}$ we finally have

$$\begin{aligned}
& \sum_{x \in \text{Leaf}^{\mathfrak{T}}, d^{\mathfrak{T}}(x) \leq N} p_x^{\mathfrak{T}} \\
&= \sum_{x \in \text{Leaf}^{\mathfrak{T}}, d^{\mathfrak{T}}(x) \leq H + M_{\max}} p_x^{\mathfrak{T}} \\
&\geq \sum_{x \in Z^{\mathfrak{T}_H}} \sum_{y \in \text{Leaf}^{\mathfrak{T}_x} \wedge d_x(y) \leq M_{\max}} p_y^{\mathfrak{T}} \\
&= \sum_{x \in Z^{\mathfrak{T}_H}} \sum_{y \in \text{Leaf}^{\mathfrak{T}_x} \wedge d_x(y) \leq M_{\max}} p_x^{\mathfrak{T}} \cdot p_y^{\mathfrak{T}_x} \\
&= \sum_{x \in Z^{\mathfrak{T}_H}} p_x^{\mathfrak{T}} \cdot \sum_{y \in \text{Leaf}^{\mathfrak{T}_x} \wedge d_x(y) \leq M_{\max}} p_y^{\mathfrak{T}_x} \\
&\geq \sum_{x \in Z^{\mathfrak{T}_H}} p_x^{\mathfrak{T}} \cdot \sum_{y \in \text{Leaf}^{\mathfrak{T}_x} \wedge d_x(y) \leq M_x} p_y^{\mathfrak{T}_x} \\
&> \sum_{x \in Z^{\mathfrak{T}_H}} p_x^{\mathfrak{T}} \cdot \beta \\
&= \beta \cdot \sum_{x \in Z^{\mathfrak{T}_H}} p_x^{\mathfrak{T}} \\
&\geq \beta \cdot \beta \\
&= 1 - \delta
\end{aligned}$$

The first inequality holds since every leaf in \mathfrak{T}_x with a depth of at most M_{\max} (in \mathfrak{T}_x) for some $x \in Z^{\mathfrak{T}_H}$ must also be a leaf in \mathfrak{T} with a depth of at most $H + M_{\max}$, since x is at a depth of at most H . \square

Before we can prove modularity for disjoint unions w.r.t. $\text{SAST} \xrightarrow{i}_{\mathcal{P}}$, we have to give some more definitions regarding the disjoint union abstraction. In addition to Def. 7.5, we also want to label the function symbols in the abstraction, to indicate from which position the function symbol originated from in the original term. For a term t , let $\text{Pos}_{\mathcal{V}}(t)$ be the set of all its variable positions and let $\text{Pos}_{\Sigma}(t)$ be the set of all those positions of t where t has function symbols instead of variables.

Definition A.12 (Labeled Disjoint Union Abstraction). Let $\mathcal{P}^{(1)}, \mathcal{P}^{(2)}$ be PTRSs with $\Sigma^{\mathcal{P}^{(1)}} \cap \Sigma^{\mathcal{P}^{(2)}} = \emptyset$. For any $d \in \{1, 2\}$, $t \in \mathcal{T}(\Sigma^{\mathcal{P}^{(1)} \cup \mathcal{P}^{(2)}}, \mathcal{V})$, and position $\pi \in \text{Pos}_{\Sigma}(t)$, $A_d^{\pi}(t)$ and $\text{Abs}_d(t)$ are multisets of terms from $\mathcal{T}(L(\Sigma^{\mathcal{P}^{(d)}}), \mathcal{V})$, which are defined as follows. Here, we use the signature $L(\Sigma^{\mathcal{P}^{(d)}}) = \Sigma^{\mathcal{P}^{(d)}} \times \text{Pos}(t)$.

$$\begin{aligned} A_d^{\pi}(y) &= \{x\}, \text{ if } y \in \mathcal{V}, \text{ where } x \text{ is always a new fresh variable} \\ A_d^{\pi}(f(t_1, \dots, t_k)) &= \{f^{\pi}(q_1, \dots, q_k) \mid q_1 \in A_d^{\pi \cdot 1}(t_1), \dots, q_k \in A_d^{\pi \cdot k}(t_k)\}, \text{ if } f \in \Sigma^{\mathcal{P}^{(d)}} \\ A_d^{\pi}(f(t_1, \dots, t_k)) &= \{x\} \cup A_d^{\pi \cdot 1}(t_1) \cup \dots \cup A_d^{\pi \cdot k}(t_k), \text{ otherwise, where } x \text{ is always a new variable} \end{aligned}$$

So $A_d^{\pi}(t)$ is always a linear term, i.e., it never contains multiple occurrences of the same variable.

For any function $\varphi : X \rightarrow X$ with $X \subseteq \mathcal{V}$, let σ_{φ} be the substitution that replaces every variable $x \in X$ by $\varphi(x) \in X$ and leaves all other variables unchanged, i.e., $\sigma_{\varphi}(x) = \varphi(x)$ if $x \in X$ and $\sigma_X(x) = x$ otherwise. Then we define

$$\text{Abs}_d(t) = \{\sigma_{\varphi}(q) \mid q \in A_d^{\varepsilon}(t), \varphi : \mathcal{V}(q) \rightarrow \mathcal{V}(q)\}$$

and $\text{Abs}(t) = \text{Abs}_1(t) \cup \text{Abs}_2(t)$. The *(labeled) disjoint union abstraction* of t is the multiset $\text{Abs}_1(t) \cup \text{Abs}_2(t)$.

Example A.13. Consider the PTRS $\mathcal{P}_{14}^{(1)}$ with the rules $\mathbf{a} \rightarrow \{1 : \mathbf{f}(\mathbf{b})\}$, $\mathbf{b} \rightarrow \{1 : \mathbf{0}\}$, and $\mathbf{h}(y, y) \rightarrow \{1 : y\}$, the PTRS $\mathcal{P}_{14}^{(2)}$ with the rule $\mathbf{g}(y) \rightarrow \{1 : \mathbf{c}\}$, and the term $\mathbf{h}(\mathbf{g}(\mathbf{a}), \mathbf{g}(x))$. We obtain¹⁴

$$\begin{aligned} A_1^{\varepsilon}(\mathbf{h}(\mathbf{g}(\mathbf{a}), \mathbf{g}(x))) &= \{\mathbf{h}^{\varepsilon}(\mathbf{a}^{1.1}, z), \mathbf{h}^{\varepsilon}(y, z)\} \\ \text{Abs}_1(\mathbf{h}(\mathbf{g}(\mathbf{a}), \mathbf{g}(x))) &= \{\mathbf{h}^{\varepsilon}(\mathbf{a}^{1.1}, z), \mathbf{h}^{\varepsilon}(z, y), \mathbf{h}^{\varepsilon}(y, z), \mathbf{h}^{\varepsilon}(z, z), \mathbf{h}^{\varepsilon}(y, y)\} \\ A_2^{\varepsilon}(\mathbf{h}(\mathbf{g}(\mathbf{a}), \mathbf{g}(x))) &= \{x', \mathbf{g}^1(y), \mathbf{g}^2(z)\} \\ \text{Abs}_2(\mathbf{h}(\mathbf{g}(\mathbf{a}), \mathbf{g}(x))) &= \{x', \mathbf{g}^1(y), \mathbf{g}^2(z)\} \end{aligned}$$

Furthermore, we will define which terms $q \in \text{Abs}(t)$ “cover” which of the nodes from $N^{\mathfrak{T}}$ for an arbitrary $\xrightarrow{i}_{\mathcal{P}(1) \cup \mathcal{P}(2)}$ -RST \mathfrak{T} starting with $(1 : t)$. The idea is that every step in a rewrite sequence starting with t (corresponding to an inner node of \mathfrak{T}) can also be performed when starting with a suitable $q \in \text{Abs}(t)$. For this, we first define the notion of an *origin graph*. For every rewrite sequence starting with t , the graph indicates which subterm “originates” from which subterm of t . Moreover, every function symbol in the origin graph is labeled by the position of the corresponding symbol on the right-hand side of the rule that created it.

Definition A.14 (Origin Graph). Let \mathcal{P} be a PTRS and let \mathfrak{T} be an $\xrightarrow{i}_{\mathcal{P}}$ -RST. The *origin graph* for \mathfrak{T} is a labeled graph with the nodes (x, π) for all $x \in N^{\mathfrak{T}}$ and all $\pi \in \text{Pos}_{\Sigma}(t_x)$, where the edges and labels are defined as follows: For the root \mathbf{r} of \mathfrak{T} we label the node (\mathbf{r}, π) by ε . For $x \in N^{\mathfrak{T}}$, let the rewrite step $t_x \xrightarrow{i}_{\mathcal{P}} \{p_1 : t_{y_1}, \dots, p_k : t_{y_k}\}$ be performed using the

¹⁴To be precise, $A_1^{\varepsilon}(\mathbf{h}(\mathbf{g}(\mathbf{a}), \mathbf{g}(x)))$ contains two additional terms $\mathbf{h}^{\varepsilon}(\mathbf{a}^{1.1}, z')$ and $\mathbf{h}^{\varepsilon}(y, z')$ since $A_1^2(\mathbf{g}(x)) = \{z, z'\}$. However, to ease readability, we disregarded them here.

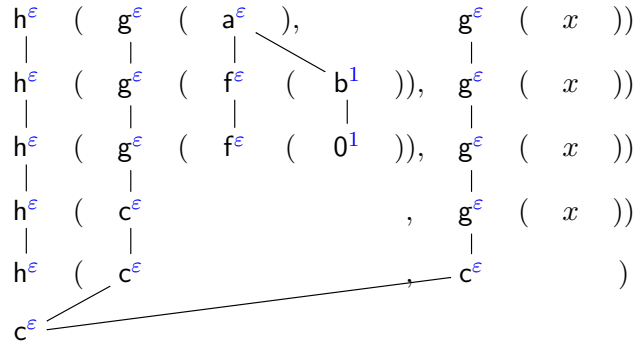
rule $\ell \rightarrow \{p_1 : r_{y_1}, \dots, p_k : r_{y_k}\}$, the position τ , and the substitution σ , i.e., $t_x|_\tau = \ell\sigma$ and $t_{y_j} = t_x[r_j\sigma]_\tau$ for all $1 \leq j \leq k$. Let $\pi \in \text{Pos}_\Sigma(t_x)$.

- (a) : If $\pi < \tau$ or $\pi \perp \tau$ (i.e., π is above or parallel to τ), then there is an edge from (x, π) to (y_j, π) . If (x, π) was labeled by γ , then (y_j, π) is labeled by γ as well.
- (b) : For $\pi = \tau$, there is an edge from (x, π) to $(y_j, \pi.\alpha)$ for all $\alpha \in \text{Pos}_\Sigma(r_j)$. If (x, π) was labeled by γ , then $(y_j, \pi.\alpha)$ is labeled by $\gamma.\alpha$.
- (c) : For every variable position $\alpha_\ell \in \text{Pos}_\Sigma(\ell)$, for all positions $\alpha_{r_j} \in \text{Pos}_\Sigma(r_j)$ with $r_j|_{\alpha_{r_j}} = \ell|_{\alpha_\ell}$, and for all $\beta \in \mathbb{N}^*$ with $\alpha_\ell.\beta \in \text{Pos}_\Sigma(\ell\sigma)$, there is an edge from $(x, \tau.\alpha_\ell.\beta)$ to $(y_j, \tau.\alpha_{r_j}.\beta)$.
- (d) : For all other positions $\pi \in \text{Pos}(t_x)$, i.e., the positions that are inside the redex $\ell\sigma$ but neither at the root of ℓ nor inside the substitution σ , there is no outgoing edge from the node (x, π) .

Example A.15. Reconsider the PTRSs $\mathcal{P}_{14}^{(1)}$ with the rules $\mathbf{a} \rightarrow \{1 : \mathbf{f}(\mathbf{b})\}$, $\mathbf{b} \rightarrow \{1 : \mathbf{0}\}$, and $\mathbf{h}(y, y) \rightarrow \{1 : y\}$, the PTRS $\mathcal{P}_{14}^{(2)}$ with the rule $\mathbf{g}(y) \rightarrow \{1 : \mathbf{c}\}$, and the term $\mathbf{h}(\mathbf{g}(\mathbf{a}), \mathbf{g}(x))$ from Ex. A.13. Furthermore, consider the following RST, where we omitted the (trivial) probabilities, and numbered each rewrite step.

$$\mathbf{h}(\mathbf{g}(\mathbf{a}), \mathbf{g}(x)) \xrightarrow{(1)} \mathbf{h}(\mathbf{g}(\mathbf{f}(\mathbf{b})), \mathbf{g}(x)) \xrightarrow{(2)} \mathbf{h}(\mathbf{g}(\mathbf{f}(\mathbf{0})), \mathbf{g}(x)) \xrightarrow{(3)} \mathbf{h}(\mathbf{c}, \mathbf{g}(x)) \xrightarrow{(4)} \mathbf{h}(\mathbf{c}, \mathbf{c}) \xrightarrow{(5)} \mathbf{c}$$

This RST yields the following origin graph.



Note that the root of every subterm that is not in normal form is reachable from exactly one node (\mathbf{r}, π) . Only nodes (x, τ) where $t_x|_\tau$ is in normal form may have multiple incoming transitions.

Our goal is to split up the RST that started with $(1 : t)$ into RSTs that start with $(1 : q)$ for $q \in \text{Abs}(t)$. The reason is that these terms q only contain symbols from either $\Sigma^{\mathcal{P}^{(1)}}$ or $\Sigma^{\mathcal{P}^{(2)}}$ and hence, there is a bound on the expected derivation length of all these RSTs.

The labels in the origin graph and the labels in the terms $q \in \text{Abs}(t)$ can now be used to construct the new RSTs that start with $(1 : q)$ for $q \in \text{Abs}(t)$ from the original RST that starts with $(1 : t)$. Then every rewrite step in the original RST corresponds to at least one step in one of these new RSTs. Let us illustrate this with our running example.

Example A.16. From the $\xrightarrow{\mathcal{P}_{14}}$ -RST in Ex. A.15 we obtain the following $\xrightarrow{\mathcal{P}_{14}^{(1)}}$ -RSTs and $\xrightarrow{\mathcal{P}_{14}^{(2)}}$ -RSTs¹⁵

¹⁵In addition to the RST starting with $\mathbf{h}^\epsilon(z, z)$ we also obtain the corresponding RST starting with $\mathbf{h}^\epsilon(y, y)$, but omitted it here for readability.

$$\begin{array}{ccc}
h^\varepsilon(\underline{a^{1.1}}, z) & \xrightarrow{(1')} & h(f(\underline{b}), z) \xrightarrow{(2')} h(f(0), z) \\
h^\varepsilon(z, z) & \xrightarrow{(5')} & z \\
\underline{g^1(z)} & \xrightarrow{(3')} & c \\
\underline{g^2(z)} & \xrightarrow{(4')} & c
\end{array}$$

where the labels are ignored for the rewrite steps, but they are used to determine which start term from $\text{Abs}(t)$ to use for which step. The rewrite step (i) in the original RST corresponds to the rewrite step (i') in our new RSTs. In the original RST, one first performs rewrite steps for the symbols a and b from $\Sigma^{\mathcal{P}_{14}^{(1)}}$, then one rewrites the symbol g from $\Sigma^{\mathcal{P}_{14}^{(2)}}$ above, and finally one rewrites the top symbol h from $\Sigma^{\mathcal{P}_{14}^{(1)}}$. In contrast, these rewrite steps are now separated such that in each of the above RSTs, one either only rewrites symbols from $\Sigma^{\mathcal{P}_{14}^{(1)}}$ or only symbols from $\Sigma^{\mathcal{P}_{14}^{(2)}}$.

Let us explain how to detect that the rewrite step (2) that is performed at position $\pi = 1.1.1$ in $h(g(f(b)), g(x))$ in the original RST should be applied at position 1.1 in $h(f(b), z)$ for rewrite step $(2')$. We consider the (unique) predecessor $(t_\tau, 1.1)$ (i.e., a^ε) of $(x_2, 1.1.1)$ (i.e., b^1), where τ is the root of the original RST and x_2 denotes the second node of the original RST containing the term $h(g(f(b)), g(x))$. This indicates that the rewrite step (2) in the original RST corresponds to a rewrite step in a new RST that starts with $(1 : q)$ for a term q containing $\text{root}(t_\tau|_{1.1})^{1.1} = \text{root}(h(g(a), g(x))|_{1.1})^{1.1} = a^{1.1}$. Hence, we have to consider the new RST starting with $h^\varepsilon(\underline{a^{1.1}}, z)$.

To find the actual rewrite step that corresponds to (2) in this new RST, we determine the position of the symbol $a^{1.1}$ within $h^\varepsilon(a^{1.1}, x)$, which is $\gamma = 1$, and the label of $(x_2, 1.1.1)$ (i.e., b^1), which is $\chi = 1$. This indicates that in the new RST we need to rewrite at position $\gamma \cdot \chi = 1.1$.

Now, we define this covering formally in order to determine which term q from $\text{Abs}(t)$ to use for the current rewrite step. More precisely, for any $q \in \text{Abs}(t)$ we define the *abstraction cover* $AC_q \subseteq N^\mathfrak{T}$ which contains all inner nodes of the original RST whose rewrite step can be simulated by a rewrite step in the new RST for q . Moreover, for every node x , we define the term $\psi_x(q)$ which we would obtain instead of t_x if we had started the RST with q instead of t . Thus, if there is an edge from the node x to the node y in the original RST \mathfrak{T} , then $\psi_x(q)$ rewrites to $\psi_y(q)$ or they are equal. Therefore, for the root τ of \mathfrak{T} , the term $\psi_\tau(q)$ rewrites (in zero or more steps) to $\psi_x(q)$ for every node of \mathfrak{T} .

Definition A.17 (Abstraction Cover). Let $\mathcal{P} = \mathcal{P}^{(1)} \cup \mathcal{P}^{(2)}$ be a PTRS with $\Sigma^{\mathcal{P}^{(1)}} \cap \Sigma^{\mathcal{P}^{(2)}} = \emptyset$ and let \mathfrak{T} be an $\xrightarrow{\mathcal{P}}$ -RST that starts with $(1 : t)$. We define the *abstraction cover* with $\bigcup_{q \in \text{Abs}(t)} AC_q = N^\mathfrak{T} \setminus \text{Leaf}^\mathfrak{T}$ recursively, where we additionally define a term $\psi_x(q)$ for each node $x \in N^\mathfrak{T}$ that corresponds to t_x in the RST starting with $(1 : q)$. So to reach $\psi_x(q)$ from q , one performs the same rewrite steps as in the path from the root to x in \mathfrak{T} whenever possible, where the appropriate position of the new rewrite step is indicated by the labels of the symbols in q and the labels in the origin graph. For instance, in Ex. A.16 for $q = h(a, z)$ we have $\psi_{x_2}(q) = h(f(b), z)$ and $\psi_{x_i}(q) = h(f(0), z)$ for all $i \in \{3, 4, 5, 6\}$ (we remove the labels in $\psi_x(q)$).

For the root node τ of \mathfrak{T} , we initially set $\psi_\tau(q) = q$.

Now consider the rewrite step at an arbitrary node $x \in N^\mathfrak{T}$, which uses the rule $\ell \rightarrow \{\dots, p_j : r_j, \dots\}$ at position π in t_x with substitution δ . Hence, we have $t_{y_j} = t_x[r_j\delta]_\pi$.

Let the origin graph of \mathfrak{T} contain a path from (\mathbf{r}, τ) to (x, π) for some position τ and let $Q \subseteq \text{Abs}(t)$ be the set of all terms from $\text{Abs}(t)$ that contain a function symbol labeled with τ . Moreover, let (x, π) be labeled with position χ in the origin graph of \mathfrak{T} . Then we add x to AC_q for all those $q \in Q$ where there exists a substitution δ' with $\psi_x(q)|_{\tau, \chi} = \ell\delta'$. For these q , we set $\psi_{y_j}(q) = \psi_x(q)[r_j\delta']_{\tau, \chi}$. Note that now indeed, $\psi_x(q)$ rewrites to $\psi_{y_j}(q)$. For all other $q \in \text{Abs}(t)$ we set $\psi_{y_j}(q) = \psi_x(q)$.

Lemma A.18 (AC_q Covers all Inner Nodes). *Let $\mathcal{P} = \mathcal{P}^{(1)} \cup \mathcal{P}^{(2)}$ be a PTRS with $\Sigma^{\mathcal{P}^{(1)}} \cap \Sigma^{\mathcal{P}^{(2)}} = \emptyset$ and let \mathfrak{T} be an $\rightarrow_{\mathcal{P}}$ -RST that starts with $(1 : t)$. Then $N^{\mathfrak{T}} \setminus \text{Leaf}^{\mathfrak{T}} = \bigcup_{q \in \text{Abs}(t)} AC_q$.*

Proof. In order to show that $\bigcup_{q \in \text{Abs}(t)} AC_q$ is indeed a cover of $N^{\mathfrak{T}} \setminus \text{Leaf}^{\mathfrak{T}}$, we have to show that for every node $x \in N^{\mathfrak{T}} \setminus \text{Leaf}^{\mathfrak{T}}$ there exists a $q \in \text{Abs}(t)$ with $x \in AC_q$.

In other words, we have to show that there exists at least one term $q \in Q$ such that there is a substitution δ' with $\psi_x(q)|_{\tau, \chi} = \ell\delta'$. W.l.o.g., let $\text{root}(\ell) \in \Sigma^{\mathcal{P}^{(1)}}$. Let C_1 be the maximal context containing no symbols from $\Sigma^{\mathcal{P}^{(2)}}$ such that there exists another context C and terms s_1, \dots, s_m with $\text{root}(s_i) \in \Sigma^{\mathcal{P}^{(2)}}$ for all $1 \leq i \leq m$ such that $t_x = C[C_1[s_1, \dots, s_m]]$ and the position π is within the context C_1 .

Let Φ be the set of positions in $\text{Pos}_{\Sigma}(t_x)$ that are within the context C_1 . There exists a term $q \in \text{Abs}(t)$ that contains all function symbols that are labeled with positions φ^{-1} where there is a path in the origin graph from $(\mathbf{r}, \varphi^{-1})$ to (x, φ) for some $\varphi \in \Phi$. This also holds if there is a normal form at position φ which may be reached from several positions φ^{-1} . Moreover, let $\Psi = \{\psi_1, \dots, \psi_m\}$ be the set of root positions of the s_1, \dots, s_m within $\text{Pos}(t_x)$, i.e., the positions of the holes in C_1 within t_x . We can choose q in such a way that the fresh variables for symbols from $\Sigma^{\mathcal{P}^{(2)}}$ at positions χ and χ' are the same whenever there are paths in the origin graph from (\mathbf{r}, χ) to (x, ψ_i) and from (\mathbf{r}, χ') to $(x, \psi_{i'})$, and $s_i = s_{i'}$.

Finally, for this specific q we have $\psi_x(q)|_{\tau, \chi} = \ell\delta'$ where δ' is like δ , but we replace every occurrence of the terms s_i with the corresponding fresh variables. \square

Example A.19. As a final example, consider the rewrite step (5), i.e., $\mathbf{h}(\mathbf{c}, \mathbf{c}) \rightarrow \mathbf{c}$. Following the notation from the proof above, we have $C_1 = \mathbf{h}(\square, \square)$, $s_1 = \mathbf{c}$, and $s_2 = \mathbf{c}$. Furthermore, we have $\psi_1 = \chi = 1$, $\psi_2 = \chi' = 2$, and the position φ of the \mathbf{h} in the context C_1 is $\varphi = \varepsilon$ with $\varphi^{-1} = \varepsilon$. Hence, we choose $\mathbf{h}^\varepsilon(z, z) \in \text{Abs}(t)$, as this term contains the function symbol \mathbf{h} labeled with $\varphi^{-1} = \varepsilon$, and uses the same variable z for both subterms, as they are equal ($s_1 = s_2 = \mathbf{c}$).

Theorem 7.8 (Modularity of $\text{SAST}_{\rightarrow_{\mathcal{P}}}$ for Disjoint Unions). *Let $\mathcal{P}^{(1)}$ and $\mathcal{P}^{(2)}$ be PTRSs with $\Sigma^{\mathcal{P}^{(1)}} \cap \Sigma^{\mathcal{P}^{(2)}} = \emptyset$. Then we have:*

$$\text{SAST}_{\rightarrow_{\mathcal{P}^{(1)} \cup \mathcal{P}^{(2)}}} \iff \text{SAST}_{\rightarrow_{\mathcal{P}^{(1)}}} \text{ and } \text{SAST}_{\rightarrow_{\mathcal{P}^{(2)}}}$$

Proof. Let $\mathcal{P} = \mathcal{P}^{(1)} \cup \mathcal{P}^{(2)}$ and assume that both $\text{SAST}_{\rightarrow_{\mathcal{P}^{(1)}}}$ and $\text{SAST}_{\rightarrow_{\mathcal{P}^{(2)}}}$ hold. Let $\mathfrak{T} = (N, E, L)$ be an arbitrary $\rightarrow_{\mathcal{P}}$ -RST that starts with $(1 : t)$. We prove that $\text{edl}(\mathfrak{T})$ is bounded by some constant which does not depend on \mathfrak{T} but just on t . This proves $\text{SAST}_{\rightarrow_{\mathcal{P}}}$.

In Def. A.17 we defined sets $AC_q \subseteq N^{\mathfrak{T}}$ for each $q \in \text{Abs}(t)$ such that $x \in AC_q$ if the rewrite step at node $x \in N^{\mathfrak{T}}$ is performed at some position π in t_x , q contains a function symbol labeled with τ where there is a path from (\mathbf{r}, τ) to (x, π) in the origin graph of \mathfrak{T} , (x, π) is labeled with some position χ in the origin graph, and we can perform the same rewrite step on $\psi_x(q)$ at position τ, χ . In Lemma A.18 we showed that $\bigcup_{q \in \text{Abs}(t)} AC_q = N^{\mathfrak{T}} \setminus \text{Leaf}^{\mathfrak{T}}$.

With the definition of AC_q we can now prove the upper bound on the expected derivation height of t . Let $H \in \mathbb{N}$ and let \mathfrak{T}_H be the tree consisting of the first H layers of \mathfrak{T} . As in the proof of the parallel execution lemma for $\mathbf{SAST}_{\rightarrow \mathcal{P}}$ (Lemma A.11), for each $H \in \mathbb{N}$ and each $q \in \text{Abs}(t)$ we split the tree \mathfrak{T}_H , into (finitely many) sets \mathbb{T}_H^q containing (finitely many) pairs of $\xrightarrow{\mathcal{P}(1)}$ -RSTs and $\xrightarrow{\mathcal{P}(2)}$ -RSTs \mathfrak{t} with certain probabilities $p_{\mathfrak{t}}$. Each of these RSTs \mathfrak{t} starts with $(1 : q)$, has height at most H , and

(Prop-1) $\sum_{(p_{\mathfrak{t}}, \mathfrak{t}) \in \mathbb{T}_H^q} p_{\mathfrak{t}} = 1$

(Prop-2) For all $x \in AC_q$ we have $p_x^{\mathfrak{T}_H} = \sum_{(p_{\mathfrak{t}}, \mathfrak{t}) \in \mathbb{T}_H^q} p_{\mathfrak{t}} \cdot p_x^{\mathfrak{t}}$.

(Prop-3) For all $\mathfrak{t} \in \mathbb{T}_H^q$, \mathfrak{t} starts with $(1 : q)$ and is a valid $\xrightarrow{\mathcal{P}(d)}$ -RST for some $d \in \{1, 2\}$.

The construction is analogous to the ones in Lemmas A.10 and A.11. Now we keep all nodes that are in AC_q and split a tree into multiple ones if the node is not in AC_q , i.e., if the rewrite step required a different starting term in the beginning.

For example, consider the PTRS $\mathcal{P}^{(1)}$ containing the rule $f(x, x) \rightarrow \{1 : f(b, c)\}$, $\mathcal{P}^{(2)}$ containing the rule $g(x) \rightarrow a$, and the innermost rewrite sequence $f(g(b), g(c)) \rightarrow f(a, g(c)) \rightarrow f(a, a) \rightarrow f(b, c)$, where we wrote \rightarrow instead of $\xrightarrow{\mathcal{P}(1) \cup \mathcal{P}(2)}$ and omitted the probabilities for readability. We split this sequence (RST) into the three sequences $g(x_1) \rightarrow a$, $g(x_2) \rightarrow a$, and $f(x_3, x_3) \rightarrow c$, where all start terms are contained in $\text{Abs}(f(g(b), g(c)))$.

Since $|\text{Abs}(t)| = K$ is finite and the expected derivation lengths of all $\xrightarrow{\mathcal{P}(d)}$ -RSTs with $d \in \{1, 2\}$ that start with $(1 : q)$ for a term $q \in \text{Abs}(t)$ are bounded by some constant $C_q < \omega$, there is a $C_{\max} < \omega$ such that for all RSTs \mathfrak{t} with $(\mathfrak{t}, p_{\mathfrak{t}}) \in \mathbb{T}_H^q$ we have $\text{edl}(\mathfrak{t}) \leq C_{\max}$ by (Prop-3). Hence, we obtain for each $H \in \mathbb{N}$:

$$\begin{aligned}
 & \text{edl}(\mathfrak{T}_H) \\
 &= \sum_{x \in N^{\mathfrak{T}_H} \setminus \text{Leaf}^{\mathfrak{T}_H}} p_x^{\mathfrak{T}_H} \\
 &\quad (\text{since } \bigcup_{q \in \text{Abs}(t)} AC_q = N^{\mathfrak{T}_H} \setminus \text{Leaf}^{\mathfrak{T}_H}) \leq \sum_{q \in \text{Abs}(t)} \sum_{x \in AC_q} p_x^{\mathfrak{T}_H} \\
 &\quad \quad (\text{by (Prop-2)}) = \sum_{q \in \text{Abs}(t)} \sum_{x \in AC_q} \sum_{(p_{\mathfrak{t}}, \mathfrak{t}) \in \mathbb{T}_H^q} p_{\mathfrak{t}} \cdot p_x^{\mathfrak{t}} \\
 &\quad \quad (\text{as in (A.4)}) = \sum_{q \in \text{Abs}(t)} \sum_{(p_{\mathfrak{t}}, \mathfrak{t}) \in \mathbb{T}_H^q} p_{\mathfrak{t}} \cdot \sum_{x \in AC_q} p_x^{\mathfrak{t}} \\
 &\quad \quad (\text{by assumption}) \leq \sum_{q \in \text{Abs}(t)} \sum_{(p_{\mathfrak{t}}, \mathfrak{t}) \in \mathbb{T}_H^q} p_{\mathfrak{t}} \cdot C_{\max} \\
 &\quad \quad (\text{by (Prop-1)}) = \sum_{q \in \text{Abs}(t)} C_{\max} \cdot \sum_{(p_{\mathfrak{t}}, \mathfrak{t}) \in \mathbb{T}_H^q} p_{\mathfrak{t}} \\
 &= \sum_{q \in \text{Abs}(t)} C_{\max} \cdot 1 \\
 &= \sum_{q \in \text{Abs}(t)} C_{\max} \\
 &= K \cdot C_{\max}
 \end{aligned}$$

Thus, we have $\text{edl}(\mathfrak{T}) = \lim_{H \rightarrow \infty} \text{edl}(\mathfrak{T}_H) \leq K \cdot C_{\max}$. \square

Finally, we prove the theorem on signature extensions.

Theorem 7.15 (Signature Extensions for $\mathbf{AST}_{\rightarrow \mathcal{P}}$ and $\mathbf{SAST}_{\rightarrow \mathcal{P}}$). *Let \mathcal{P} be a PTRS, $s \in \{\mathbf{f}, \mathbf{i}\}$, and let Σ' be some signature. Then we have:*

$$\begin{aligned}
 \mathbf{AST}_{\rightarrow \mathcal{P}} \text{ over } \Sigma^{\mathcal{P}} &\iff \mathbf{AST}_{\rightarrow \mathcal{P}} \text{ over } \Sigma^{\mathcal{P}} \cup \Sigma' \\
 \mathbf{SAST}_{\rightarrow \mathcal{P}} \text{ over } \Sigma^{\mathcal{P}} &\iff \mathbf{SAST}_{\rightarrow \mathcal{P}} \text{ over } \Sigma^{\mathcal{P}} \cup \Sigma'
 \end{aligned}$$

Proof. We only prove the non-trivial direction “ \implies ” and consider the following three cases:

- (i) For innermost rewriting, the theorem is implied by modularity of $\mathbf{AST}_{\xrightarrow{\mathcal{P}}}$ and $\mathbf{SAST}_{\xrightarrow{\mathcal{P}}}$ for disjoint unions (Thm. 7.2 and 7.8).
- (ii) For full rewriting, we first consider the case where $\Sigma^{\mathcal{P}}$ only contains constants and unary symbols. Let Σ' be another signature containing fresh symbols, i.e., w.l.o.g. we

have $\Sigma^{\mathcal{P}} \cap \Sigma' = \emptyset$. For any term from $\mathcal{T}(\Sigma^{\mathcal{P}} \cup \Sigma', \mathcal{V})$ we now compute a multiset of terms from $\mathcal{T}(\Sigma^{\mathcal{P}}, \mathcal{V})$ which can be regarded instead. Thus, we define a corresponding mapping A from $\mathcal{T}(\Sigma^{\mathcal{P}} \cup \Sigma', \mathcal{V})$ to multisets of terms from $\mathcal{T}(\Sigma^{\mathcal{P}}, \mathcal{V})$. For this definition, we need two auxiliary mappings. The mapping $C : \mathcal{T}(\Sigma^{\mathcal{P}} \cup \Sigma', \mathcal{V}) \rightarrow \mathcal{T}(\Sigma^{\mathcal{P}}, \mathcal{V})$ replaces all topmost subterms with a root f from Σ' by the fresh variable x_f and the mapping B maps any term $t \in \mathcal{T}(\Sigma^{\mathcal{P}} \cup \Sigma', \mathcal{V})$ to the multiset that unites $A(r)$ for the topmost subterms r with $\text{root}(r) \in \Sigma^{\mathcal{P}}$ occurring below a symbol from Σ' in t .

$$\begin{aligned} C(x) &= x, \text{ if } x \in \mathcal{V} \\ C(f(t_1, \dots, t_k)) &= f(C(t_1), \dots, C(t_k)), \text{ if } f \in \Sigma^{\mathcal{P}} \\ C(f(t_1, \dots, t_k)) &= x_f, \text{ if } f \in \Sigma' \end{aligned}$$

$$\begin{aligned} A(x) &= \{x\}, \text{ if } x \in \mathcal{V} \\ A(f(t_1, \dots, t_k)) &= \{f(C(t_1), \dots, C(t_k))\} \cup B(t_1) \cup \dots \cup B(t_k), \text{ if } f \in \Sigma^{\mathcal{P}} \\ A(f(t_1, \dots, t_k)) &= A(t_1) \cup \dots \cup A(t_k), \text{ if } f \in \Sigma' \end{aligned}$$

$$\begin{aligned} B(x) &= \emptyset, \text{ if } x \in \mathcal{V} \\ B(f(t_1, \dots, t_k)) &= B(t_1) \cup \dots \cup B(t_k), \text{ if } f \in \Sigma^{\mathcal{P}} \\ B(f(t_1, \dots, t_k)) &= A(t_1) \cup \dots \cup A(t_k), \text{ if } f \in \Sigma' \end{aligned}$$

Let $t \in \mathcal{T}(\Sigma^{\mathcal{P}} \cup \Sigma', \mathcal{V})$ and let $A(t) = \{q_1, \dots, q_n\}$. Let c be a fresh n -ary constructor symbol. Then instead of t , we consider the term $c(q_1, \dots, q_n)$. Every $\xrightarrow{\mathcal{P}}$ -RST \mathfrak{T} that starts with $(1 : t)$ gives rise to an $\xrightarrow{\mathcal{P}}$ -RST \mathfrak{T}' that starts with $(1 : c(q_1, \dots, q_n))$ with $|\mathfrak{T}| = |\mathfrak{T}'|$ and $\text{edl}(\mathfrak{T}) = \text{edl}(\mathfrak{T}')$. To see this, suppose that $t \xrightarrow{\mathcal{P}} \{p_1 : s_1, \dots, p_k : s_k\}$, $A(t) = \{q_1, \dots, q_n\}$, and $A(s_i) = \{q_1^i, \dots, q_{m_i}^i\}$ for all $1 \leq i \leq k$. Then there exists a $1 \leq j \leq n$ with $q_j \xrightarrow{\mathcal{P}} \{p_1 : u_1, \dots, p_k : u_k\}$ where $u_i \in A(s_i)$ for all $1 \leq i \leq k$. For every $v_i \in A(s_i) \setminus \{u_i\}$ there exists a $j' \neq j$ with $1 \leq j' \leq n$ such that $v_i = q_{j'}$ for all $1 \leq i \leq k$. Note that $m_i \leq n$, and we might even have $m_i < n$ if we use an erasing rule. Hence, every step in \mathfrak{T} in a term t can be mirrored by a step in \mathfrak{T}' in a term $c(q_1, \dots, q_n)$.

Since the q_i are terms from $\mathcal{T}(\Sigma^{\mathcal{P}}, \mathcal{V})$, every $\xrightarrow{\mathcal{P}}$ -RST that starts with $(1 : q_i)$ converges with probability 1 (resp. has bounded expected derivation length). Hence, by the parallel execution lemmas (Lemmas A.10 and A.11) this also holds for every $\xrightarrow{\mathcal{P}}$ -RST \mathfrak{T}' that starts with $(1 : c(q_1, \dots, q_n))$ (note that there cannot be any rewrite steps at the root since c is a constructor). But then this also holds for every $\xrightarrow{\mathcal{P}}$ -RST \mathfrak{T} that starts with $(1 : t)$.

- (iii) Now we consider the case where $\Sigma^{\mathcal{P}}$ contains at least one symbol g of arity at least 2. If g has an arity greater than two, then we use the term $g(-, -, x, \dots, x)$ instead, where $x \in \mathcal{V}$. Again, let Σ' be another signature containing fresh symbols, i.e., w.l.o.g. we have $\Sigma^{\mathcal{P}} \cap \Sigma' = \emptyset$.

We now define a function $\phi : \mathcal{T}(\Sigma^{\mathcal{P}} \cup \Sigma', \mathcal{V}) \rightarrow \mathcal{T}(\Sigma^{\mathcal{P}}, \mathcal{V})$ such that $\text{edh}_{\xrightarrow{\mathcal{P}}}(t) \leq \text{edh}_{\xrightarrow{\mathcal{P}}}(\phi(t))$:

$$\begin{aligned} \phi(x) &= x, \text{ if } x \in \mathcal{V} \\ \phi(f(t_1, \dots, t_k)) &= f(\phi(t_1), \dots, \phi(t_k)), \text{ if } f \in \Sigma^{\mathcal{P}} \\ \phi(f) &= x_f, \text{ if } f \in \Sigma' \text{ has arity } 0 \\ \phi(f(t)) &= g(\phi(t), x_f), \text{ if } f \in \Sigma' \text{ has arity } 1 \\ \phi(f(t_1, \dots, t_k)) &= g(\phi(t_1), g(\phi(t_2), \dots, g(\phi(t_k), x_f) \dots)), \text{ if } f \in \Sigma' \text{ has arity } k > 1 \end{aligned}$$

Every $\xrightarrow{\mathcal{P}}$ -RST that starts with $(1 : t)$ gives rise to an $\xrightarrow{\mathcal{P}}$ -RST that starts with $(1 : \phi(t))$ using exactly the same rules, leading to the same convergence probability and the same expected derivation length. To see this, note that whenever $t \xrightarrow{\mathcal{P}} \{p_1 : s_1, \dots, p_k : s_k\}$, then also $\phi(t) \xrightarrow{\mathcal{P}} \{p_1 : \phi(s_1), \dots, p_k : \phi(s_k)\}$. \square