From Innermost to Full Almost-Sure Termination of Probabilistic Term Rewriting[⋆]

Jan-Christoph Kassing(✉), Florian Frohn(✉), and Jürgen Giesl(✉)

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

Abstract. There are many evaluation strategies for term rewrite systems, but proving termination automatically is usually easiest for innermost rewriting. Several syntactic criteria exist when innermost termination implies full termination. We adapt these criteria to the probabilistic setting, e.g., we show when it suffices to analyze almost-sure termination (AST) w.r.t. innermost rewriting to prove full AST of probabilistic term rewrite systems. These criteria also apply to other notions of termination like positive AST. We implemented and evaluated our new contributions in the tool AProVE.

1 Introduction

Termination analysis is one of the main tasks in program verification, and techniques and tools to analyze termination 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 [3, 20]. This framework allows for modular termination proofs by decomposing the original termination problem into sub-problems whose termination can then be analyzed independently using different techniques. Thus, DPs are used in essentially all current termination tools for TRSs (e.g., AProVE [21], MuTerm [25], NaTT [46], TTT2 [33]). To allow certification of termination proofs with DPs, they have been formalized in several proof assistants and there exist several corresponding certification tools for termination proofs with DPs (e.g., CeTA [43]).

On the other hand, *probabilistic* programs are used to describe randomized algorithms and probability distributions, with applications in many areas, see, e.g., [23]. To use TRSs also for such programs, *probabilistic term rewrite systems* (PTRSs) were introduced in [4, 9, 10]. In the probabilistic setting, there are several notions of "termination". In this paper, we mostly focus on analyzing *almost-sure termination* (AST), i.e., we want to prove automatically that the probability for termination is 1.

⋆ funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 235950644 (Project GI 274/6-2) and the DFG Research Training Group 2236 UnRAVeL

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-57231-9_10.

While there exist many automatic approaches to prove (P)AST of imperative programs on numbers (e.g., [2, 5, 11, 16, 22, 26–28, 36–38, 40]), there are only few automatic approaches for programs with complex non-tail recursive structure [8, 12, 13]. The approaches that are also suitable for algorithms on recursive data structures [7, 35, 45] are mostly specialized for specific data structures and cannot easily be adjusted to other (possibly user-defined) ones, or are not yet fully automated.

For innermost AST (i.e., AST restricted to rewrite sequences where one only evaluates at innermost positions), we recently presented an adaption of the DP framework which allows us to benefit from a similar modularity as in the non-probabilistic setting [29, 32]. Unfortunately, there is no such modular powerful approach available for *full* AST (i.e., AST when considering arbitrary rewrite sequences). Up to now, full AST of PTRSs can only be proved via a direct application of orderings [4, 29], but there is no corresponding adaption of dependency pairs. (As explained in [29], a DP framework to analyze full instead of innermost AST would be "considerably more involved".) Indeed, also in the non-probabilistic setting, innermost termination is usually substantially easier to prove than full termination, see, e.g., [3, 20]. To lift innermost termination proofs to full rewriting, in the non-probabilistic setting, there exist several sufficient criteria which ensure that innermost termination implies full termination [24].

Up to now no such results were known in the probabilistic setting. Our paper presents the first sufficient criteria for PTRSs which ensure that AST coincide 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 [29, 32]. In this way, we obtain a modular powerful technique that can also prove AST for *full* rewriting automatically.

We will also consider the stronger notion of *positive almost-sure termination* (PAST) [10, 42], which requires that the expected runtime is finite, and show that our criteria for the relationship between full and innermost probabilistic rewriting hold for PAST as well. In contrast to AST, PAST is not modular, i.e., the sequence of two programs that are PAST may yield a program that is not PAST (see, e.g., [27]). Therefore, up to now there is no variant of DPs that allows to prove PAST of PTRSs, but there only exist techniques to apply polynomial or matrix orderings directly [4].

We start with preliminaries on term rewriting in Sect. 2. Then we recapitulate PTRSs based on [4, 10, 14, 15, 29] in Sect. 3. In Sect. 4 we show that the properties of [24] 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, as well as for strategies like leftmost-innermost evaluation. In Sect. 5 we show how 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 Sect. 6. We refer to [30] for all proofs.

2 Preliminaries

We assume familiarity with term rewriting [6] and regard (possibly infinite) TRSs over a (possibly infinite) signature Σ and a set of variables $\mathcal{V}$. Consider the TRS $\mathcal{R}_{\mathsf{d}}$ that doubles a natural number (represented by the terms s and $\mathcal{O}$) with the rewrite rules $\mathsf{d}(\mathsf{s}(x)) \to \mathsf{s}(\mathsf{s}(\mathsf{d}(x)))$ and $\mathsf{d}(\mathcal{O}) \to \mathcal{O}$ as an example. A TRS $\mathcal{R}$ induces a *rewrite relation* $\to_{\mathcal{R}} \subseteq \mathcal{T}(\Sigma, \mathcal{V}) \times \mathcal{T}(\Sigma, \mathcal{V})$ on terms where $s \to_{\mathcal{R}} t$ holds if there is a position π, a rule $\ell \to r \in \mathcal{R}$, and a substitution σ such that $s|_\pi = \ell\sigma$ and $t = s[r\sigma]_\pi$. A rewrite step $s \to_{\mathcal{R}} t$ is an *innermost* rewrite step (denoted $s \xrightarrow{\mathsf{i}}_{\mathcal{R}} t$) if all proper subterms of the used redex $\ell\sigma$ are in normal form w.r.t. $\mathcal{R}$ (i.e., they do not contain redexes themselves and thus, they cannot be reduced with $\to_{\mathcal{R}}$). For example, we have $\mathsf{d}(\mathsf{s}(\mathsf{d}(\mathsf{s}(\mathcal{O})))) \xrightarrow{\mathsf{i}}_{\mathcal{R}_{\mathsf{d}}} \mathsf{d}(\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{d}(\mathcal{O})))))$.

Let $<$ be the prefix ordering on positions and let $\leq$ be its reflexive closure. Then for two parallel positions τ and π we define $\tau \prec \pi$ 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 $s \xrightarrow{\mathsf{i}}_{\mathcal{R}} t$ at position π is *leftmost* (denoted $s \xrightarrow{\mathsf{li}}_{\mathcal{R}} t$) if there exists no redex at a position τ with $\tau \prec \pi$.

We call a TRS $\mathcal{R}$ *strongly (innermost/leftmost innermost) normalizing* (SN / iSN / liSN) if $\to_{\mathcal{R}}$ ($\xrightarrow{\mathsf{i}}_{\mathcal{R}}$ / $\xrightarrow{\mathsf{li}}_{\mathcal{R}}$) is well founded. SN is also called "*terminating*" and iSN/liSN are called "*innermost/leftmost innermost terminating*". If every term $t \in \mathcal{T}(\Sigma, \mathcal{V})$ has a normal form (i.e., we have $t \to_{\mathcal{R}}^* t'$ where t' is in normal form) then we call $\mathcal{R}$ *weakly normalizing* (WN). Two terms s, t are *joinable* via $\mathcal{R}$ (denoted $s \downarrow_{\mathcal{R}} t$) if there exists a term w such that $s \to_{\mathcal{R}}^* w \leftarrow_{\mathcal{R}}^* t$. Two rules $\ell_1 \to r_1, \ell_2 \to r_2 \in \mathcal{R}$ with renamed variables such that $\mathcal{V}(\ell_1) \cap \mathcal{V}(\ell_2) = \varnothing$ are *overlapping* if there exists a non-variable position π of ℓ_1 such that $\ell_1|_\pi$ and ℓ_2 are unifiable with a mgu σ. If $(\ell_1 \to r_1) = (\ell_2 \to r_2)$, then we require that $\pi \neq \varepsilon$. $\mathcal{R}$ is *non-overlapping* (NO) if it has no overlapping rules. As an example, the TRS $\mathcal{R}_{\mathsf{d}}$ 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 iSN, SN, liSN, and WN in the non-probabilistic setting. We start with the relation between iSN and SN.

Counterexample 1 (Toyama's Counterexample [44]). The TRS $\mathcal{R}_1$ with the rules $\mathsf{f}(\mathsf{a}, \mathsf{b}, x) \to \mathsf{f}(x, x, x)$, $\mathsf{g} \to \mathsf{a}$, and $\mathsf{g} \to \mathsf{b}$ is not SN since we have $\mathsf{f}(\mathsf{a}, \mathsf{b}, \mathsf{g}) \to_{\mathcal{R}_1} \mathsf{f}(\mathsf{g}, \mathsf{g}, \mathsf{g}) \to_{\mathcal{R}_1} \mathsf{f}(\mathsf{a}, \mathsf{g}, \mathsf{g}) \to_{\mathcal{R}_1} \mathsf{f}(\mathsf{a}, \mathsf{b}, \mathsf{g}) \to_{\mathcal{R}_1} \ldots$ But the only innermost rewrite sequences starting with $\mathsf{f}(\mathsf{a}, \mathsf{b}, \mathsf{g})$ are $\mathsf{f}(\mathsf{a}, \mathsf{b}, \mathsf{g}) \xrightarrow{\mathsf{i}}_{\mathcal{R}_1} \mathsf{f}(\mathsf{a}, \mathsf{b}, \mathsf{a}) \xrightarrow{\mathsf{i}}_{\mathcal{R}_1} \mathsf{f}(\mathsf{a}, \mathsf{a}, \mathsf{a})$ and $\mathsf{f}(\mathsf{a}, \mathsf{b}, \mathsf{g}) \xrightarrow{\mathsf{i}}_{\mathcal{R}_1} \mathsf{f}(\mathsf{a}, \mathsf{b}, \mathsf{b}) \xrightarrow{\mathsf{i}}_{\mathcal{R}_1} \mathsf{f}(\mathsf{b}, \mathsf{b}, \mathsf{b})$, i.e., both reach normal forms in the end. Thus, $\mathcal{R}_1$ is iSN as we have to rewrite the inner g before we can use the f-rule.

The first property known to ensure equivalence of SN and iSN is orthogonality. A TRS is *orthogonal* if it is non-overlapping and left-linear.

Theorem 2 (From iSN to SN (1), [41]). *If a TRS $\mathcal{R}$ is orthogonal, then $\mathcal{R}$ is SN iff $\mathcal{R}$ is iSN.*

Then, in [24] it was shown that one can remove the left-linearity requirement.

Theorem 3 (From iSN to SN (2), [24]). *If a TRS $\mathcal{R}$ is non-overlapping, then $\mathcal{R}$ is SN iff $\mathcal{R}$ is iSN.*

Finally, [24] also refined Thm. 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 Ex. 1 one can see that the overlaps occur at non-root positions, i.e., $\mathcal{R}_1$ is not an overlay system. Furthermore, a TRS is *locally confluent* (or *weakly Church-Rosser*, abbreviated WCR) if for all terms s, t_1, t_2 such that $t_1 \;{}_{\mathcal{R}}\!\leftarrow s \to_{\mathcal{R}} t_2$ the terms t_1 and t_2 are joinable. So $\mathcal{R}_1$ is *not* WCR, as we have $\mathsf{f}(\mathsf{a}, \mathsf{b}, \mathsf{a}) \;{}_{\mathcal{R}_1}\!\!\leftarrow \mathsf{f}(\mathsf{a}, \mathsf{b}, \mathsf{g}) \to_{\mathcal{R}_1} \mathsf{f}(\mathsf{a}, \mathsf{b}, \mathsf{b})$, but $\mathsf{f}(\mathsf{a}, \mathsf{b}, \mathsf{a}) \not\downarrow_{\mathcal{R}_1} \mathsf{f}(\mathsf{a}, \mathsf{b}, \mathsf{b})$. If a TRS has both of these properties, then iSN and SN are again equivalent.

Theorem 4 (From iSN to SN (3), [24]). *If a TRS $\mathcal{R}$ is a locally confluent overlay system, then $\mathcal{R}$ is SN iff $\mathcal{R}$ is iSN.*

Thm. 4 is stronger than Thm. 3 as every non-overlapping TRS is a locally confluent overlay system. We recapitulate the relation between WN and SN next.

Counterexample 5. Consider the TRS $\mathcal{R}_2$ with the rules $\mathsf{f}(x) \to \mathsf{b}$ and $\mathsf{a} \to \mathsf{f}(\mathsf{a})$. This TRS is not SN since we can always rewrite the inner a to get $\mathsf{a} \to_{\mathcal{R}_2} \mathsf{f}(\mathsf{a}) \to_{\mathcal{R}_2} \mathsf{f}(\mathsf{f}(\mathsf{a})) \to_{\mathcal{R}_2} \ldots$, but it is WN since we can also rewrite the outer $\mathsf{f}(\ldots)$ 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 $\mathsf{f}(\mathsf{a}) \to \mathsf{b}$ and $\mathsf{a} \to \mathsf{f}(\mathsf{a})$, the situation is similar.

The TRS $\mathcal{R}_2$ from Ex. 5 is erasing and $\mathcal{R}_3$ is overlapping. For TRSs with neither of those two properties, SN and WN are equivalent.

Theorem 6 (From WN to SN [24]). *If a TRS $\mathcal{R}$ is non-overlapping and non-erasing, then $\mathcal{R}$ is SN iff $\mathcal{R}$ is WN.*

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

Theorem 7 (From liSN to iSN [34]). *For all TRSs $\mathcal{R}$ we have that $\mathcal{R}$ is iSN iff $\mathcal{R}$ is liSN.*

The relations between the different properties for non-probabilistic TRSs (given in Thm. 4, 6, and 7) are summarized below.

$$\begin{tikzcd}
\mathbf{liSN} \arrow[r, Leftrightarrow] & \mathbf{iSN} \arrow[r, Rightarrow, bend left, "\mathbf{OS + WCR}"] \arrow[rr, Rightarrow, bend right] & \mathbf{SN} \arrow[l, Rightarrow, bend left] \arrow[r, Rightarrow, bend right] & \mathbf{WN} \arrow[l, Rightarrow, bend right, "\mathbf{NO+NE}"']
\end{tikzcd}$$

3 Probabilistic Term Rewriting

In this section, we recapitulate *probabilistic TRSs* [4, 10, 29]. In contrast to TRSs, a PTRS has finite multi-distributions[1] on the right-hand sides of its rewrite rules.[2] A finite *multi-distribution* μ on a set $A \neq \varnothing$ is a finite multiset of pairs $(p : a)$, where $0 < p \leq 1$ is a probability and $a \in A$, such that $\sum_{(p:a) \in \mu} p = 1$. $\mathrm{FDist}(A)$ is the set of all finite multi-distributions on A. For $\mu \in \mathrm{FDist}(A)$, its *support* is the multiset $\mathrm{Supp}(\mu) = \{a \mid (p : a) \in \mu \text{ for some } p\}$. A *probabilistic rewrite rule* is a pair $\ell \to \mu \in \mathcal{T}(\Sigma, \mathcal{V}) \times \mathrm{FDist}(\mathcal{T}(\Sigma, \mathcal{V}))$ such that $\ell \notin \mathcal{V}$ and $\mathcal{V}(r) \subseteq \mathcal{V}(\ell)$ for every $r \in \mathrm{Supp}(\mu)$. A *probabilistic TRS* (PTRS) is a (possibly infinite) set $\mathcal{S}$ of probabilistic rewrite rules. Similar to TRSs, the PTRS $\mathcal{S}$ induces a *rewrite relation* $\to_{\mathcal{S}} \subseteq \mathcal{T}(\Sigma, \mathcal{V}) \times \mathrm{FDist}(\mathcal{T}(\Sigma, \mathcal{V}))$ where $s \to_{\mathcal{S}} \{p_1 : t_1, \ldots, p_k : t_k\}$ if there is a position π, a rule $\ell \to \{p_1 : r_1, \ldots, p_k : r_k\} \in \mathcal{S}$, 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 $s \to_{\mathcal{S}} \mu$ an *innermost* rewrite step (denoted $s \xrightarrow{\mathsf{i}}_{\mathcal{S}} \mu$) if all proper subterms of the used redex $\ell\sigma$ are in normal form w.r.t. $\mathcal{S}$. We have $s \xrightarrow{\mathsf{li}}_{\mathcal{S}} \mu$ if the rewrite step $s \xrightarrow{\mathsf{i}}_{\mathcal{S}} \mu$ at position π is leftmost (i.e., there is no redex at a position τ with $\tau \prec \pi$). For example, the PTRS $\mathcal{S}_{\mathsf{rw}}$ with the only rule $\mathsf{g} \to \{1/2 : \mathsf{c}(\mathsf{g}, \mathsf{g}),\ 1/2 : \bot\}$ corresponds to a symmetric random walk on the number of g-symbols in a term.

As in [4, 14, 15, 29], we *lift* $\to_{\mathcal{S}}$ to a rewrite relation 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 $\mu \in \mathrm{FDist}(A)$, let $p \cdot \mu = \{(p \cdot q : a) \mid (q : a) \in \mu\}$.

Definition 8 (Lifting). *The* lifting $\rightrightarrows \subseteq \mathrm{FDist}(\mathcal{T}(\Sigma, \mathcal{V})) \times \mathrm{FDist}(\mathcal{T}(\Sigma, \mathcal{V}))$ *of a relation* $\to \subseteq \mathcal{T}(\Sigma, \mathcal{V}) \times \mathrm{FDist}(\mathcal{T}(\Sigma, \mathcal{V}))$ *is the smallest relation with:*

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

For a PTRS $\mathcal{S}$, we write $\rightrightarrows_{\mathcal{S}}$, $\overset{\mathsf{i}}{\rightrightarrows}_{\mathcal{S}}$, and $\overset{\mathsf{li}}{\rightrightarrows}_{\mathcal{S}}$ for the liftings of $\to_{\mathcal{S}}$, $\xrightarrow{\mathsf{i}}_{\mathcal{S}}$, and $\xrightarrow{\mathsf{li}}_{\mathcal{S}}$, respectively.

Example 9. For example, we obtain the following $\rightrightarrows_{\mathcal{S}_{\mathsf{rw}}}$-rewrite sequence (which is also a $\overset{\mathsf{i}}{\rightrightarrows}_{\mathcal{S}_{\mathsf{rw}}}$-sequence, but not a $\overset{\mathsf{li}}{\rightrightarrows}_{\mathcal{S}_{\mathsf{rw}}}$-sequence).

$$\begin{aligned}
& \{1 : \mathsf{g}\} \\
\rightrightarrows_{\mathcal{S}_{\mathsf{rw}}} & \{1/2 : \mathsf{c}(\mathsf{g}, \mathsf{g}), 1/2 : \bot\} \\
\rightrightarrows_{\mathcal{S}_{\mathsf{rw}}} & \{1/4 : \mathsf{c}(\mathsf{c}(\mathsf{g}, \mathsf{g}), \mathsf{g}), 1/4 : \mathsf{c}(\bot, \mathsf{g}), 1/2 : \bot\} \\
\rightrightarrows_{\mathcal{S}_{\mathsf{rw}}} & \{1/8 : \mathsf{c}(\mathsf{c}(\mathsf{g}, \mathsf{g}), \mathsf{c}(\mathsf{g}, \mathsf{g})), 1/8 : \mathsf{c}(\mathsf{c}(\mathsf{g}, \mathsf{g}), \bot), 1/8 : \mathsf{c}(\bot, \mathsf{c}(\mathsf{g}, \mathsf{g})), 1/8 : \mathsf{c}(\bot, \bot), 1/2 : \bot\}
\end{aligned}$$

[1] The restriction to finite multi-distributions allows us to simplify the handling of PTRSs in the proofs.

[2] A different form of probabilistic rewrite rules was proposed in PMaude [1], where numerical extra variables in right-hand sides of rules are instantiated according to a probability distribution.

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

Definition 10 ($|\mu|_{\mathcal{S}}$)**.** *For a PTRS $\mathcal{S}$, $\mathtt{NF}_{\mathcal{S}} \subseteq \mathcal{T}(\Sigma, \mathcal{V})$ denotes the set of all* normal forms *w.r.t. $\mathcal{S}$. For any $\mu \in \mathrm{FDist}(\mathcal{T}(\Sigma, \mathcal{V}))$, let $|\mu|_{\mathcal{S}} = \sum_{(p:t)\in\mu, t\in\mathtt{NF}_{\mathcal{S}}} p$.*

Example 11. Consider $\{1/8 : \mathsf{c}(\mathsf{c}(\mathsf{g},\mathsf{g}),\mathsf{c}(\mathsf{g},\mathsf{g})), 1/8 : \mathsf{c}(\mathsf{c}(\mathsf{g},\mathsf{g}),\bot), 1/8 : \mathsf{c}(\bot,\mathsf{c}(\mathsf{g},\mathsf{g})), 1/8 : \mathsf{c}(\bot,\bot), 1/2 : \bot\} = \mu$ from Ex. 9. Then $|\mu|_{\mathcal{S}_{\mathsf{rw}}} = 1/8 + 1/2 = 5/8$, since $\mathsf{c}(\bot,\bot)$ and $\bot$ are both normal forms w.r.t. $\mathcal{S}_{\mathsf{rw}}$.

Definition 12 (AST). *Let $\mathcal{S}$ be a PTRS and $\vec{\mu} = (\mu_n)_{n\in\mathbb{N}}$ be an infinite $\rightrightarrows_{\mathcal{S}}$-rewrite sequence, i.e., $\mu_n \rightrightarrows_{\mathcal{S}} \mu_{n+1}$ for all $n \in \mathbb{N}$. We say that $\vec{\mu}$* converges with probability $\lim_{n\to\infty} |\mu_n|_{\mathcal{S}}$*. $\mathcal{S}$ is* almost-surely terminating (AST) *(*innermost AST (iAST) / leftmost innermost AST (liAST)*) if $\lim_{n\to\infty} |\mu_n|_{\mathcal{S}} = 1$ holds for every infinite $\rightrightarrows_{\mathcal{S}}$- ($\overset{\mathsf{i}}{\rightrightarrows}_{\mathcal{S}}$- / $\overset{\mathsf{li}}{\rightrightarrows}_{\mathcal{S}}$-) rewrite sequence $(\mu_n)_{n\in\mathbb{N}}$. To highlight the consideration of AST for* full *(instead of innermost) rewriting, we also speak of* full AST *(fAST) instead of "AST". We say that $\mathcal{S}$ is* weakly AST *(wAST) if for every term t there exists an infinite $\rightrightarrows_{\mathcal{S}}$-rewrite sequence $(\mu_n)_{n\in\mathbb{N}}$ with $\lim_{n\to\infty} |\mu_n|_{\mathcal{S}} = 1$ and $\mu_0 = \{1 : t\}$.*

Example 13. For every infinite extension $(\mu_n)_{n\in\mathbb{N}}$ of the $\rightrightarrows_{\mathcal{S}_{\mathsf{rw}}}$-rewrite sequence in Ex. 9, we have $\lim_{n\to\infty} |\mu_n|_{\mathcal{S}} = 1$. Indeed, $\mathcal{S}_{\mathsf{rw}}$ is fAST and thus also iAST, liAST, and wAST.

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

Definition 14 (edl**, PAST).** *Let $\mathcal{S}$ be a PTRS and $\vec{\mu} = (\mu_n)_{n\in\mathbb{N}}$ be an infinite $\rightrightarrows_{\mathcal{S}}$-rewrite sequence. By $\mathrm{edl}(\vec{\mu}) = \sum_{n=0}^{\infty}(1 - |\mu_n|_{\mathcal{S}})$ we denote the* expected derivation length *of $\vec{\mu}$. $\mathcal{S}$ is* positively almost-surely terminating (PAST) *(*innermost PAST (iPAST) / leftmost innermost AST (liPAST)*) if $\mathrm{edl}(\vec{\mu})$ is finite for every infinite $\rightrightarrows_{\mathcal{S}}$- ($\overset{\mathsf{i}}{\rightrightarrows}_{\mathcal{S}}$- / $\overset{\mathsf{li}}{\rightrightarrows}_{\mathcal{S}}$-) rewrite sequence $\vec{\mu} = (\mu_n)_{n\in\mathbb{N}}$.*[3] *Again, we also speak of* full PAST *(fPAST) when considering PAST for the* full *rewrite relation $\rightrightarrows_{\mathcal{S}}$. We say that $\mathcal{S}$ is* weakly PAST *(wPAST) if for every term t there exists an infinite $\rightrightarrows_{\mathcal{S}}$-rewrite sequence $\vec{\mu} = (\mu_n)_{n\in\mathbb{N}}$ such that $\mathrm{edl}(\vec{\mu})$ is finite and $\mu_0 = \{1 : t\}$.*

It is well known that PAST implies AST, but not vice versa.

Example 15. For every infinite extension $\vec{\mu} = (\mu_n)_{n\in\mathbb{N}}$ of the $\rightrightarrows_{\mathcal{S}_{\mathsf{rw}}}$-rewrite sequence in Ex. 9, the expected derivation length $\mathrm{edl}(\vec{\mu})$ is infinite, hence $\mathcal{S}_{\mathsf{rw}}$ is not PAST w.r.t. any of the strategies regarded in this paper.

[3] This definition is from [4], where it is also explained why this definition of PAST is equivalent to the one of, e.g., [10].

In [4, 18], PAST was strengthened further to *bounded* or *strong almost-sure termination* (SAST). Indeed, our results on PAST can also be adapted to SAST (see [30]).

Many properties of TRSs from Sect. 2 can be lifted to PTRSs in a straightforward way: A PTRS $\mathcal{S}$ is right-linear (non-erasing) iff the TRS $\{\ell \to r \mid \ell \to \mu \in \mathcal{S}, r \in \operatorname{Supp}(\mu)\}$ has the respective property. Moreover, all properties that just consider the left-hand sides, e.g., 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.

4 Relating Variants of AST

Our goal is to relate AST of full rewriting to restrictions of fAST, i.e., to iAST (Sect. 4.1), wAST (Sect. 4.2), and liAST (Sect. 4.3). More precisely, we want to find properties of PTRSs which are suitable for automated checking and which guarantee that two variants of AST are equivalent. Then for example, we can use existing tools that analyze iAST in order to prove fAST. Clearly, we have to impose at least the same requirements as in the non-probabilistic setting, as every TRS $\mathcal{R}$ can be transformed into a PTRS $\mathcal{S}$ by replacing every rule $\ell \to r$ with $\ell \to \{1 : r\}$. Then $\mathcal{R}$ is SN / iSN / liSN iff $\mathcal{S}$ is fAST / iAST / liAST. While we mostly focus on AST, all results and counterexamples in this section also hold for PAST.

4.1 From iAST to fAST

Again, we start by analyzing the relation between iAST and fAST. The following example shows that Thm. 2 does not carry over to the probabilistic setting, i.e., orthogonality is not sufficient to ensure that iAST implies fAST.

Counterexample 16 (Orthogonality Does Not Suffice). Consider the orthogonal PTRS $\mathcal{S}_1$ with the two rules:

$$\mathsf{g} \to \{{}^3\!/\!_4 : \mathsf{d}(\mathsf{g}), {}^1\!/\!_4 : \bot\} \qquad\qquad \mathsf{d}(x) \to \{1 : \mathsf{c}(x,x)\}$$

This PTRS is not fAST (and thus, also not fPAST), as we have $\{1 : \mathsf{g}\} \rightrightarrows^2_{\mathcal{S}_1} \{{}^3\!/\!_4 : \mathsf{c}(\mathsf{g},\mathsf{g}), {}^1\!/\!_4 : \bot\}$, which corresponds to a random walk biased towards non-termination (since $\frac{3}{4} > \frac{1}{2}$).

However, the d-rule can only duplicate normal forms in innermost evaluations. To see that $\mathcal{S}_1$ is iPAST (and thus, also iAST), consider the following rewrite sequence $\vec{\mu}$:

$$\{1 : \mathsf{g}\} \overset{\mathsf{i}}{\rightrightarrows}_{\mathcal{S}_1} \{{}^3\!/\!_4 : \mathsf{d}(\mathsf{g}), {}^1\!/\!_4 : \bot\} \overset{\mathsf{i}}{\rightrightarrows}_{\mathcal{S}_1} \{({}^3\!/\!_4)^2 : \mathsf{d}(\mathsf{d}(\mathsf{g})), {}^1\!/\!_4 \cdot {}^3\!/\!_4 : \mathsf{d}(\bot), {}^1\!/\!_4 : \bot\} \overset{\mathsf{i}}{\rightrightarrows}_{\mathcal{S}_1} \ldots$$

We can also view this rewrite sequence as a tree:

$$\begin{tikzcd}[column sep=small]
\mu_0 : & & & & \boxed{1 \mid \mathsf{g}} \arrow[dl] \arrow[drr] & & \\
\mu_1 : & & & \boxed{{}^3\!/_4 \mid \mathsf{d}(\mathsf{g})} \arrow[dl] \arrow[dr] & & & \boxed{{}^1\!/_4 \mid \bot} \\
\mu_2 : & & \boxed{({}^3\!/_4)^2 \mid \mathsf{d}(\mathsf{d}(\mathsf{g}))} \arrow[dl] \arrow[dr] & & \boxed{{}^1\!/_4 \cdot {}^3\!/_4 \mid \mathsf{d}(\bot)} \arrow[drr] & & \\
\mu_3 : & \boxed{({}^3\!/_4)^3 \mid \mathsf{d}(\mathsf{d}(\mathsf{d}(\mathsf{g})))} \arrow[d] & & \boxed{{}^1\!/_4 \cdot ({}^3\!/_4)^2 \mid \mathsf{d}(\mathsf{d}(\bot))} \arrow[drr] & & & \dots \\
& \dots & & & & \dots &
\end{tikzcd}$$

The branch to the right that starts with $\bot$ stops after 0 innermost steps, the branch that starts with $\mathsf{d}(\bot)$ stops after 1 innermost steps, the branch that starts with $\mathsf{d}(\mathsf{d}(\bot))$ stops after 2 innermost steps, and so on. So if we start with the term $\mathsf{d}^n(\bot)$, then we reach a normal form after n steps, and we reach $\mathsf{d}^n(\bot)$ after $n+1$ steps from the initial term g, where $\mathsf{d}^n(\bot) = \underbrace{\mathsf{d}(\dots(\mathsf{d}}_{n\text{-times}}(\bot))\dots)$. Hence, for every $k \in \mathbb{N}$ we have $|\mu_{2\cdot k+1}|_{\mathcal{S}_1} = |\mu_{2\cdot k+2}|_{\mathcal{S}_1} = \sum_{n=0}^{k} {}^1\!/_4 \cdot ({}^3\!/_4)^n$ and thus

$$\begin{aligned}
\mathrm{edl}(\vec{\mu}) &= \textstyle\sum_{n=0}^{\infty}(1 - |\mu_n|_{\mathcal{S}_1}) &&= 1 + 2 \cdot \textstyle\sum_{k\in\mathbb{N}}(1 - |\mu_{2\cdot k+1}|_{\mathcal{S}_1}) \\
&= 1 + 2 \cdot \textstyle\sum_{k\in\mathbb{N}}(1 - \sum_{n=0}^{k} {}^1\!/_4 \cdot ({}^3\!/_4)^n) &&= 1 + 2 \cdot \textstyle\sum_{k\in\mathbb{N}}({}^3\!/_4)^{k+1} \\
&= (2 \cdot \textstyle\sum_{k\in\mathbb{N}}({}^3\!/_4)^k) - 1 &&= 7
\end{aligned}$$

Analogously, in all other innermost rewrite sequences, the d-rule can also only duplicate normal forms. Thus, all possible innermost rewrite sequences have finite expected derivation length. Therefore, $\mathcal{S}_1$ is iPAST and thus, also iAST. The latter can also be proved automatically by our implementation of the probabilistic DP framework for iAST [29] in AProVE.

To construct a counterexample for AST of $\mathcal{S}_1$, we exploited the fact that $\mathcal{S}_1$ is not right-linear. Indeed, requiring right-linearity yields our desired result. For reasons of space, here we only give a proof sketch. As mentioned, all full proofs can be found in [30].

Theorem 17 (From iAST/iPAST to fAST/fPAST (1)). *If a PTRS $\mathcal{S}$ is orthogonal and right-linear (i.e., non-overlapping and linear), then:*

$$\mathcal{S} \textit{ is fAST} \iff \mathcal{S} \textit{ is iAST}$$
$$\mathcal{S} \textit{ is fPAST} \iff \mathcal{S} \textit{ is iPAST}$$

Proof Sketch. We only have to prove the non-trivial direction "$\Longleftarrow$". The proofs for all theorems in this section (for both AST and PAST) follow a similar structure. We always iteratively replace rewrite steps by steps that use the desired strategy and ensure that this does not increase the probability of termination (resp. 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.

We first consider fAST and iAST. Let $\mathcal{S}$ be a PTRS that is non-overlapping, linear, and not fAST. Thus, there exists an infinite rewrite sequence $\vec{\mu} = (\mu_n)_{n\in\mathbb{N}}$ such that $\lim_{n\to\infty} |\mu_n|_{\mathcal{S}} = c$ for some $c \in \mathbb{R}$ with $0 \leq c < 1$. Our goal is to transform this sequence into an innermost sequence that converges at most with 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 $\mathcal{S}$ 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 $\lim_{n\to\infty} |\mu_n^{(1)}|_{\mathcal{S}} = \lim_{n\to\infty} |\mu_n|_{\mathcal{S}} = c$. Here, linearity is needed to ensure that the probability of termination 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 termination probability, etc. In the end, the limit of all these rewrite sequences $\lim_{i\to\infty}(\mu_n^{(i)})_{n\in\mathbb{N}}$ is an innermost rewrite sequence that converges with probability at most $c < 1$, and hence, the PTRS $\mathcal{S}$ is not innermost AST.

For fPAST and iPAST, we start with an infinite rewrite sequence $\vec{\mu}$ such that $\mathrm{edl}(\vec{\mu}) = \infty$. Again, we replace the first non-innermost rewrite step with an innermost rewrite step using exactly the same construction as before to obtain $\vec{\mu}^{(1)}$, etc., since $\vec{\mu}^{(1)}$ does not only have the same termination probability as $\vec{\mu}$, but we also have $\mathrm{edl}(\vec{\mu}^{(1)}) \geq \mathrm{edl}(\vec{\mu})$. In the end, the limit of all these rewrite sequences $\lim_{i\to\infty} \vec{\mu}^{(i)}$ is an innermost rewrite sequence such that $\mathrm{edl}(\lim_{i\to\infty} \vec{\mu}^{(i)}) \geq \mathrm{edl}(\vec{\mu}) = \infty$, and hence, the PTRS $\mathcal{S}$ is not innermost PAST. $\square$

One may wonder whether we can remove the left-linearity requirement from Thm. 17, as in the non-probabilistic setting. It turns out that this is not possible.

Counterexample 18 (Left-Linearity Cannot be Removed). Consider the PTRS $\mathcal{S}_2$ with the rules:

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

$\mathcal{S}_2$ is not fAST (hence also not fPAST), since $\{1 : \mathsf{f}(\mathsf{a},\mathsf{a})\} \rightrightarrows_{\mathcal{S}_2} \{1 : \mathsf{f}(\mathsf{a},\mathsf{a})\} \rightrightarrows_{\mathcal{S}_2} \ldots$ is an infinite rewrite sequence that converges with probability 0. However, it is iPAST (and hence, iAST) since the corresponding innermost sequence has the form $\{1 : \mathsf{f}(\mathsf{a},\mathsf{a})\} \overset{\mathrm{i}}{\rightrightarrows}_{\mathcal{S}_2} \{\frac{1}{2} : \mathsf{f}(\mathsf{b},\mathsf{a}), \frac{1}{2} : \mathsf{f}(\mathsf{c},\mathsf{a})\} \overset{\mathrm{i}}{\rightrightarrows}_{\mathcal{S}_2} \{\frac{1}{4} : \mathsf{f}(\mathsf{b},\mathsf{b}), \frac{1}{4} : \mathsf{f}(\mathsf{b},\mathsf{c}), \frac{1}{4} : \mathsf{f}(\mathsf{c},\mathsf{b}), \frac{1}{4} : \mathsf{f}(\mathsf{c},\mathsf{c})\}$. Here, the last distribution contains two normal forms $\mathsf{f}(\mathsf{b},\mathsf{c})$ and $\mathsf{f}(\mathsf{c},\mathsf{b})$ that did not occur in the previous rewrite sequence. Since all innermost rewrite sequences keep on adding such normal forms after a certain number of steps for each start term, they always have finite expected derivation length and thus, converge with probability 1 (again, iAST 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 either, as shown by the PTRS $\mathcal{S}_3$ which results from $\mathcal{S}_2$ by replacing the f-rule with $\mathsf{f}(x,x) \to \{1 : \mathsf{d}(\mathsf{f}(\mathsf{a},\mathsf{a}),x)\}$.

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 is impossible in the non-probabilistic setting.

Next, one could try to adapt Thm. 4 to the probabilistic setting (when requiring linearity in addition). So one could investigate whether iAST implies fAST for PTRSs that are linear locally confluent overlay systems. A PTRS $\mathcal{S}$ is *locally confluent* if for all multi-distributions μ, μ_1, μ_2 such that $\mu_1 \leftleftarrows_{\mathcal{S}} \mu \rightrightarrows_{\mathcal{S}} \mu_2$, there exists a multi-distribution μ' such that $\mu_1 \rightrightarrows_{\mathcal{S}}^{*} \mu' \leftleftarrows_{\mathcal{S}}^{*} \mu_2$, see [14]. Note that in contrast to the probabilistic setting, there are non-overlapping PTRSs that are not locally confluent (e.g., the variant $\mathcal{S}_2'$ of $\mathcal{S}_2$ that consists of the rules $\mathsf{f}(x,x) \to \{1:\mathsf{d}\}$ and $\mathsf{a} \to \{1/2:\mathsf{b}, 1/2:\mathsf{c}\}$, since we have $\{1:\mathsf{d}\} \leftleftarrows_{\mathcal{S}_2'} \{1: \mathsf{f}(\mathsf{a},\mathsf{a})\} \rightrightarrows_{\mathcal{S}_2'} \{1/2:\mathsf{f}(\mathsf{b},\mathsf{a}), 1/2:\mathsf{f}(\mathsf{c},\mathsf{a})\}$ and the two resulting multi-distributions are not joinable). Thus, such an adaption of Thm. 4 would not subsume Thm. 17.

In contrast to the proof of Thm. 2, the proof of Thm. 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 position π of t such that there is an infinite rewrite sequence of t starting with the redex $t|_\pi$, but no infinite rewrite sequence of t starting with a redex at a position $\tau > \pi$ which is strictly below π. In other words, if t starts an infinite rewrite sequence, then there is a "minimal" infinite rewrite sequence starting in t, i.e., as soon as one reduces a proper subterm of one of the redexes in the sequence, then one obtains a term which is terminating. However, such minimal infinite sequences do not always exist in the probabilistic setting.

Example 19 (No Minimal Infinite Rewrite Sequence for AST). Reconsider the PTRS $\mathcal{S}_1$ from Ex. 16, which is not fAST. However, there is no "minimal" rewrite sequence with convergence probability < 1 such that one rewrite step at a proper subterm of a redex would modify the multi-distribution in such a way that now only rewrite sequences with convergence probability 1 are possible. We have $\{1:\mathsf{g}\} \rightrightarrows_{\mathcal{S}_1} \{3/4:\mathsf{d}(\mathsf{g}), 1/4:\bot\}$. In Ex. 16, we now alternated between the d- and the g-rule, resulting in a biased random walk, i.e., we obtained $\{3/4:\mathsf{d}(\mathsf{g}), 1/4: \bot\} \rightrightarrows_{\mathcal{S}_1} \{3/4:\mathsf{c}(\mathsf{g},\mathsf{g}), 1/4:\bot\} \rightrightarrows_{\mathcal{S}_1} \{3/4:\mathsf{c}(\mathsf{d}(\mathsf{g}),\mathsf{g}), 1/4:\bot\} \rightrightarrows_{\mathcal{S}_1} \ldots$ The steps with the d-rule use redexes that have g as a proper subterm.

However, there does not exist any "minimal" non-fAST sequence. If we rewrite the proper subterm g of a redex $\mathsf{d}(\mathsf{g})$, then this still yields a multi-distribution that is not fAST, i.e., it can still start a rewrite sequence with convergence probability < 1. For example, we have $\{3/4:\mathsf{d}(\mathsf{g}), 1/4:\bot\} \rightrightarrows_{\mathcal{S}_1} \{(3/4)^2:\mathsf{d}(\mathsf{d}(\mathsf{g})), 1/4 \cdot 3/4 : \mathsf{d}(\bot), 1/4:\bot\}$, but the obtained multi-distribution still contains the subterm g, and thus, one can still continue the rewrite sequence in such a way that its convergence probability is < 1. Again, the same example also shows that there is no "minimal" non-fPAST sequence.

It remains open whether one can also adapt Thm. 4 to the probabilistic setting (e.g., if one can replace non-overlappingness in Thm. 17 by the requirement of locally confluent overlay systems). There are two main difficulties when trying to adapt the proof of this theorem 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 rewriting below a position that was reduced in the original (minimal) infinite rewrite sequence leads to a strongly normalizing rewrite sequence. Second, the original proof of Thm. 4 uses Newman's Lemma [39] 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 adaptions of the unique normal form property for the probabilistic setting have been studied in [14, 15], which concluded that obtaining an analogous statement to Newman's Lemma for PTRSs that are AST (or PAST) 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 (or PAST), since these rewrite sequences may be infinite.

4.2 From wAST to fAST

Next, we investigate wAST. Since iAST implies wAST, we essentially have the same problems as for innermost AST, i.e., in addition to non-overlappingness, we need linearity, as seen in Ex. 16 and 18, as $\mathcal{S}_1$ and $\mathcal{S}_3$ are iAST (and hence wAST) but not fAST, while they 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 Ex. 5.

Theorem 20 (From wAST/wPAST to fAST/fPAST). *If a PTRS $\mathcal{S}$ is non-overlapping, linear, and non-erasing, then*

$$\mathcal{S} \textit{ is fAST} \Longleftrightarrow \mathcal{S} \textit{ is wAST}$$
$$\mathcal{S} \textit{ is fPAST} \Longleftrightarrow \mathcal{S} \textit{ is wPAST}$$

4.3 From liAST to fAST

Finally, we look at leftmost-innermost AST 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 liAST and iAST are not equivalent in general. The counterexample is similar to Ex. 18, which illustrated that fAST and iAST are not equivalent without left-linearity.

Counterexample 21. Consider the PTRS $\mathcal{S}_4$ with the five rules:

$$\mathsf{a} \to \{1 : \mathsf{c}_1\} \qquad \mathsf{b} \to \{1/2 : \mathsf{d}_1, 1/2 : \mathsf{d}_2\}$$
$$\mathsf{f}(\mathsf{c}_1, \mathsf{d}_1) \to \{1 : \mathsf{f}(\mathsf{a}, \mathsf{b})\}$$
$$\mathsf{a} \to \{1 : \mathsf{c}_2\} \qquad \mathsf{f}(\mathsf{c}_2, \mathsf{d}_2) \to \{1 : \mathsf{f}(\mathsf{a}, \mathsf{b})\}$$

This PTRS is not iAST (and hence not iPAST) since there exists the infinite rewrite sequence $\{1 : \mathsf{f}(\mathsf{a}, \mathsf{b})\} \overset{\mathsf{i}}{\rightrightarrows}_{\mathcal{S}_4} \{1/2 : \mathsf{f}(\mathsf{a}, \mathsf{d}_1), 1/2 : \mathsf{f}(\mathsf{a}, \mathsf{d}_2)\} \overset{\mathsf{i}}{\rightrightarrows}^2_{\mathcal{S}_4} \{1/2 : \mathsf{f}(\mathsf{c}_1, \mathsf{d}_1), 1/2 : \mathsf{f}(\mathsf{c}_2, \mathsf{d}_2)\} \overset{\mathsf{i}}{\rightrightarrows}^2_{\mathcal{S}_4} \{1/2 : \mathsf{f}(\mathsf{a}, \mathsf{b}), 1/2 : \mathsf{f}(\mathsf{a}, \mathsf{b})\} \overset{\mathsf{i}}{\rightrightarrows}_{\mathcal{S}_4} \ldots$, which converges with probability 0. It first "splits" the term $\mathsf{f}(\mathsf{a}, \mathsf{b})$ 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. For example, we have $\{1 : \mathsf{f}(\mathsf{a}, \mathsf{b})\} \overset{\mathsf{li}}{\rightrightarrows}_{\mathcal{S}_4} \{1 : \mathsf{f}(\mathsf{c}_1, \mathsf{b})\} \overset{\mathsf{li}}{\rightrightarrows}_{\mathcal{S}_4} \{1/2 : \mathsf{f}(\mathsf{c}_1, \mathsf{d}_1), 1/2 : \mathsf{f}(\mathsf{c}_1, \mathsf{d}_2)\}$. Here, the second term $\mathsf{f}(\mathsf{c}_1, \mathsf{d}_2)$ 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, the PTRS is liAST (and also liPAST).

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, liAST and iAST are again equivalent for non-overlapping TRSs. For such TRSs, at most one rule can be used to rewrite at a given position, which prevents the problem illustrated in Ex. 21.

Theorem 22 (From liAST/liPAST to iAST/iPAST). *If a PTRS $\mathcal{S}$ is non-overlapping, then*

$$\mathcal{S} \text{ is iAST} \Longleftrightarrow \mathcal{S} \text{ is liAST}$$

$$\mathcal{S} \text{ is iPAST} \Longleftrightarrow \mathcal{S} \text{ is liPAST}$$

The relations between the different properties for AST of PTRSs (given in Thm. 17, 20, and 22) are summarized below. An analogous figure also holds for PAST.

$$\begin{tikzcd}
\textbf{liAST} \arrow[r, Rightarrow, bend left, "\textbf{NO}"] \arrow[rrr, Rightarrow, bend right=40] & \textbf{iAST} \arrow[l, Rightarrow, bend left] \arrow[r, Rightarrow, bend left, "\textbf{NO+LL+RL}"] \arrow[rr, Rightarrow, bend right=30] & \textbf{fAST} \arrow[l, Rightarrow, bend left] \arrow[r, Rightarrow, bend right] & \textbf{wAST} \arrow[l, Rightarrow, bend right, "\textbf{NO+LL+RL+NE}"']
\end{tikzcd}$$

5 Improving Applicability

In this section, we improve the applicability of Thm. 17, which relates fAST and iAST. The results of Sect. 5.1 allow us to remove the requirement of left-linearity by modifying the rewrite relation to *simultaneous rewriting*. Then in Sect. 5.2 we show that the requirement of right-linearity can be weakened to *spareness* if one only considers rewrite sequences that start with *basic terms*.

5.1 Removing Left-Linearity by Simultaneous Rewriting

First, we will see that we do not need to require left-linearity if we allow the simultaneous reduction of several copies of identical redexes. For a PTRS $\mathcal{S}$, this results in the notion of *simultaneous rewriting*, denoted $\rightarrowtail_{\mathcal{S}}$. While $\overset{\mathsf{i}}{\rightarrowtail}_{\mathcal{S}}$ over-approximates $\overset{\mathsf{i}}{\to}_{\mathcal{S}}$, existing techniques for proving iAST [29, 32] (except for the rewriting processor[4]) do not distinguish between both notions of rewriting, i.e., these techniques even prove that every rewrite sequence with the lifting $\overset{\mathsf{i}}{\rightrightarrows}_{\mathcal{S}}$ of $\overset{\mathsf{i}}{\rightarrowtail}_{\mathcal{S}}$ converges with probability 1. So for non-overlapping and right-linear PTRSs, these techniques can be used to prove innermost almost-sure termination w.r.t.

[4] This processor is an optional transformation technique which was added in [32] when improving the DP framework further since it sometimes helps to increase power, but all other (major) DP processors do not distinguish between $\overset{\mathsf{i}}{\to}_{\mathcal{S}}$ and $\overset{\mathsf{i}}{\rightarrowtail}_{\mathcal{S}}$.

$\rightarrowtail_{\mathcal{S}}$, which then implies fAST. The following example illustrates our approach for handling non-left-linear PTRSs by applying the same rewrite rule at parallel positions simultaneously.

Example 23 (Simultaneous Rewriting). Reconsider the PTRS $\mathcal{S}_2$ from Ex. 18 with the rules $\mathsf{f}(x,x) \to \{1 : \mathsf{f}(\mathsf{a},\mathsf{a})\}$ and $\mathsf{a} \to \{1/2 : \mathsf{b}, 1/2 : \mathsf{c}\}$ which is iAST, but not fAST. Our new rewrite relation $\rightrightarrows_{\mathcal{S}_2}$ allows us to reduce several copies of the same redex simultaneously, so that we get $\{1 : \mathsf{f}(\mathsf{a},\mathsf{a})\} \overset{\mathsf{i}}{\rightrightarrows}_{\mathcal{S}_2} \{\frac{1}{2} : \mathsf{f}(\mathsf{b},\mathsf{b}), \frac{1}{2} : \mathsf{f}(\mathsf{c},\mathsf{c})\} \overset{\mathsf{i}}{\rightrightarrows}^2_{\mathcal{S}_2} \{1/2 : \mathsf{f}(\mathsf{a},\mathsf{a}), 1/2 : \mathsf{f}(\mathsf{a},\mathsf{a})\}$, i.e., this $\overset{\mathsf{i}}{\rightrightarrows}_{\mathcal{S}_2}$-sequence converges with probability 0 and thus, $\mathcal{S}_2$ is *not* iAST w.r.t. $\rightarrowtail_{\mathcal{S}_2}$. Note that we simultaneously reduced both occurrences of a in the first step.

Definition 24 (Simultaneous Rewriting). *Let $\mathcal{S}$ be a PTRS. A term s rewrites* simultaneously *to a multi-distribution $\mu = \{p_1 : t_1, \ldots, p_k : t_k\}$ (denoted $s \rightarrowtail_{\mathcal{S}} \mu$) if there is a non-empty set of parallel positions Π, a rule $\ell \to \{p_1 : r_1, \ldots, p_k : r_k\} \in \mathcal{S}$, and a substitution σ such that $s|_\pi = \ell\sigma$ and $t_j = s[r_j\sigma]_\pi$ for every position $\pi \in \Pi$ and for all $1 \leq j \leq k$. We call $s \rightarrowtail_{\mathcal{S}} \mu$ an* innermost simultaneous *rewrite step (denoted $s \overset{\mathsf{i}}{\rightarrowtail}_{\mathcal{S}} \mu$) if all proper subterms of the redex $\ell\sigma$ are in normal form w.r.t. $\mathcal{S}$.*

Clearly, if the set of positions Π from Def. 24 is a singleton, then the resulting simultaneous rewrite step is an “ordinary” probabilistic rewrite step, i.e., $\to_{\mathcal{S}} \subseteq \rightarrowtail_{\mathcal{S}}$ and $\overset{\mathsf{i}}{\to}_{\mathcal{S}} \subseteq \overset{\mathsf{i}}{\rightarrowtail}_{\mathcal{S}}$.

Corollary 25 (From $\rightarrowtail_{\mathcal{S}}$ to $\to_{\mathcal{S}}$). *If $\mathcal{S}$ is fAST (iAST) w.r.t. $\rightarrowtail_{\mathcal{S}}$, i.e., every infinite $\rightrightarrows_{\mathcal{S}}$- (resp. $\overset{\mathsf{i}}{\rightrightarrows}_{\mathcal{S}}$-) rewrite sequence converges with probability 1, then $\mathcal{S}$ is fAST (iAST). Analogously, if $\mathcal{S}$ is fPAST (iPAST) w.r.t. $\rightarrowtail_{\mathcal{S}}$, i.e., every infinite $\rightrightarrows_{\mathcal{S}}$- (resp. $\overset{\mathsf{i}}{\rightrightarrows}_{\mathcal{S}}$-) rewrite sequence has finite expected derivation length, then $\mathcal{S}$ is fPAST (iPAST).*

However, the converse of Cor. 25 does not hold. Ex. 23 shows that $\overset{\mathsf{i}}{\rightarrowtail}_{\mathcal{S}}$ allows for rewrite sequences that are not possible with $\overset{\mathsf{i}}{\to}_{\mathcal{S}}$, and the following example shows the same for $\rightarrowtail_{\mathcal{S}}$ and $\to_{\mathcal{S}}$.

Counterexample 26. Consider the PTRS $\overline{\mathcal{S}}_2$ with the three rules:

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

This PTRS is fAST. But as in Ex. 23, we have $\{1 : \mathsf{f}(\mathsf{a},\mathsf{a})\} \overset{\mathsf{i}}{\rightrightarrows}_{\overline{\mathcal{S}}_2} \{\frac{1}{2} : \mathsf{f}(\mathsf{b},\mathsf{b}), \frac{1}{2} : \mathsf{f}(\mathsf{c},\mathsf{c})\} \overset{\mathsf{i}}{\rightrightarrows}^2_{\overline{\mathcal{S}}_2} \{1/2 : \mathsf{f}(\mathsf{a},\mathsf{a}), 1/2 : \mathsf{f}(\mathsf{a},\mathsf{a})\}$, i.e., there are rewrite sequences with $\overset{\mathsf{i}}{\rightrightarrows}_{\overline{\mathcal{S}}_2}$ and thus, also with $\rightrightarrows_{\overline{\mathcal{S}}_2}$ that converge with probability 0. Hence, $\overline{\mathcal{S}}_2$ is not iAST or fAST w.r.t. $\rightarrowtail_{\overline{\mathcal{S}}_2}$. Again, the same example also shows that fPAST and fPAST w.r.t. simultaneous rewriting are not equivalent either.

Note that this kind of simultaneous rewriting is different from the “ordinary” parallelism used for non-probabilistic rewriting, which is typically denoted by $\to_{||}$. There, one may reduce multiple parallel redexes in a single rewrite step. Here, we do not only allow reducing multiple redexes, but in addition we “merge” the corresponding terms in the multi-distributions that result from rewriting the different 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. For example, for $\mathcal{S}_2$ from Ex. 18 we have $\{1 : \mathsf{f}(\mathsf{a},\mathsf{a})\} \overset{\mathsf{i}}{\rightarrowtail\!\!\rightrightarrows}_{\mathcal{S}_2} \{\frac{1}{2} : \mathsf{f}(\mathsf{b},\mathsf{b}), \frac{1}{2} : \mathsf{f}(\mathsf{c},\mathsf{c})\}$, whereas using ordinary parallel rewriting we would get $\{1 : \mathsf{f}(\mathsf{a},\mathsf{a})\} \overset{\mathsf{i}}{\rightrightarrows}_{||\mathcal{S}_2} \{\frac{1}{4} : \mathsf{f}(\mathsf{b},\mathsf{b}), \frac{1}{4} : \mathsf{f}(\mathsf{b},\mathsf{c}), \frac{1}{4} : \mathsf{f}(\mathsf{c},\mathsf{b}), \frac{1}{4} : \mathsf{f}(\mathsf{c},\mathsf{c})\}$.

The following theorem shows that indeed, we do not need to require left-linearity when moving from iAST/iPAST w.r.t. $\rightarrowtail_{\mathcal{S}}$ to fAST/fPAST w.r.t. $\rightarrow_{\mathcal{S}}$.

Theorem 27 (From iAST/iPAST to fAST/fPAST (2)). *If a PTRS $\mathcal{S}$ is non-overlapping and right-linear, then*

$$\mathcal{S} \textit{ is fAST} \Longleftarrow \mathcal{S} \textit{ is iAST w.r.t. } \rightarrowtail_{\mathcal{S}}$$
$$\mathcal{S} \textit{ is fPAST} \Longleftarrow \mathcal{S} \textit{ is iPAST w.r.t. } \rightarrowtail_{\mathcal{S}}$$

Proof Sketch. We use an analogous construction as for the proof of Thm. 17, but in addition, if we replace a non-innermost rewrite step by an innermost one, then we check whether in the original rewrite sequence, the corresponding innermost redex is "inside" the substitution used for the non-innermost rewrite step. In that case, if this rewrite step applied a non-left-linear rule, then we identify all other (equal) innermost redexes and use $\overset{\mathsf{i}}{\rightarrowtail}_{\mathcal{S}}$ to rewrite them simultaneously (as we did for the innermost redex a in Ex. 23). $\square$

Note that Ex. 26 shows that the direction " $\Longrightarrow$ " does not hold in Thm. 27. The following example shows that right-linearity in Thm. 27 cannot be weakened to the requirement that $\mathcal{S}$ is *non-duplicating* (i.e., that no variable occurs more often in a term on the right-hand side of a rule than on its left-hand side).

Counterexample 28 (Non-Duplicating Does Not Suffice). Let $\mathsf{d}(\mathsf{f}(\mathsf{a},\mathsf{a})^3)$ abbreviate $\mathsf{d}(\mathsf{f}(\mathsf{a},\mathsf{a}),\mathsf{f}(\mathsf{a},\mathsf{a}),\mathsf{f}(\mathsf{a},\mathsf{a}))$. Consider the PTRS $\mathcal{S}_5$ with the four rules:

$$\mathsf{f}(x,x) \rightarrow \{1 : \mathsf{g}(x,x)\} \qquad \mathsf{g}(\mathsf{b},\mathsf{c}) \rightarrow \{1 : \mathsf{d}(\mathsf{f}(\mathsf{a},\mathsf{a})^3)\}$$
$$\mathsf{a} \rightarrow \{1/2 : \mathsf{b}, 1/2 : \mathsf{c}\} \qquad \mathsf{g}(\mathsf{c},\mathsf{b}) \rightarrow \{1 : \mathsf{d}(\mathsf{f}(\mathsf{a},\mathsf{a})^3)\}$$

$\mathcal{S}_5$ is not fAST (and thus, also not fPAST), since the infinite rewrite sequence $\{1 : \mathsf{f}(\mathsf{a},\mathsf{a})\} \rightrightarrows_{\mathcal{S}_5} \{1 : \mathsf{g}(\mathsf{a},\mathsf{a})\} \rightrightarrows^2_{\mathcal{S}_5} \{1/4 : \mathsf{g}(\mathsf{b},\mathsf{b}), 1/4 : \mathsf{g}(\mathsf{b},\mathsf{c}), 1/4 : \mathsf{g}(\mathsf{c},\mathsf{b}), 1/4 : \mathsf{g}(\mathsf{c},\mathsf{c})\} \rightrightarrows^2_{\mathcal{S}_5} \{1/4 : \mathsf{g}(\mathsf{b},\mathsf{b}), 1/4 : \mathsf{d}(\mathsf{f}(\mathsf{a},\mathsf{a})^3), 1/4 : \mathsf{d}(\mathsf{f}(\mathsf{a},\mathsf{a})^3), 1/4 : \mathsf{g}(\mathsf{c},\mathsf{c})\}$ can be seen as a biased random walk on the number of $\mathsf{f}(\mathsf{a},\mathsf{a})$-subterms that is not AST. However, for every innermost evaluation with $\overset{\mathsf{i}}{\rightarrow}_{\mathcal{S}_5}$ or $\overset{\mathsf{i}}{\rightarrowtail}_{\mathcal{S}_5}$ we have to rewrite the inner a-symbols first. Afterwards, the f-rule can only be used on redexes $\mathsf{f}(t,t)$ where the resulting term $\mathsf{g}(t,t)$ is a normal form. Thus, $\mathcal{S}_5$ is iPAST (and hence, iAST) w.r.t. $\rightarrowtail_{\mathcal{S}_5}$.

Note that for wAST, the direction of the implication in Cor. 25 is reversed, since wAST requires that for each start term, there *exists* an infinite rewrite sequence that is almost-surely terminating, whereas fAST requires that *all* infinite rewrite sequences are almost-surely terminating. Thus, if there exists an infinite $\rightrightarrows_{\mathcal{S}}$-rewrite sequence that converges with probability 1 (showing that $\mathcal{S}$ is wAST), then this is also a valid $\rightarrowtail\!\!\rightrightarrows_{\mathcal{S}}$-rewrite sequence that converges with probability 1 (showing that $\mathcal{S}$ is wAST w.r.t. $\rightarrowtail_{\mathcal{S}}$).

Corollary 29 (From $\to_{\mathcal{S}}$ to $\rightarrowtail_{\mathcal{S}}$ for wAST/wPAST). *If $\mathcal{S}$ is wAST (wPAST), then $\mathcal{S}$ is wAST (wPAST) w.r.t. $\rightarrowtail_{\mathcal{S}}$.*

One may wonder whether simultaneous rewriting could also be used to improve Thm. 20 by removing the requirement of left-linearity, but Ex. 30 shows this is not possible.

Counterexample 30. Consider the non-left-linear PTRS $\mathcal{S}_6$ with the two rules:

$$\mathsf{g} \to \{{}^{3}/{}_{4} : \mathsf{d}(\mathsf{g}, \mathsf{g}), {}^{1}/{}_{4} : \bot\} \qquad\qquad \mathsf{d}(x, x) \to \{1 : x\}$$

This PTRS is not fAST (and thus, also not fPAST), as we have $\{1 : \mathsf{g}\} \rightrightarrows_{\mathcal{S}_6} \{{}^{3}/{}_{4} : \mathsf{d}(\mathsf{g}, \mathsf{g}), {}^{1}/{}_{4} : \bot\}$, which corresponds to a random walk biased towards non-termination if we never use the d-rule (since $\frac{3}{4} > \frac{1}{2}$). However, if we always use the d-rule directly after the g-rule, then we essentially end up with a PTRS whose only rule is $\mathsf{g} \to \{{}^{3}/{}_{4} : \mathsf{c}(\mathsf{g}), {}^{1}/{}_{4} : \bot\}$, which corresponds to flipping a biased coin until heads comes up. This proves that $\mathcal{S}_6$ is wPAST and hence, also wAST. As $\mathcal{S}_6$ is non-overlapping, right-linear, and non-erasing, this shows that a variant of Thm. 20 without the requirement of left-linearity needs more than just moving to simultaneous rewriting.

5.2 Weakening Right-Linearity to Spareness

To improve our results further, we introduce the notion of *spareness*. The idea of spareness is to require that variables which occur non-linear in right-hand sides may only be instantiated by normal forms. We already used spareness for non-probabilistic TRSs in [17] to find classes of TRSs where innermost and full runtime complexity coincide. For a PTRS $\mathcal{S}$, we decompose its signature $\Sigma = \Sigma_C \uplus \Sigma_D$ such that $f \in \Sigma_D$ iff $f = \mathrm{root}(\ell)$ for some rule $\ell \to \mu \in \mathcal{S}$. The symbols in Σ_C and Σ_D are called *constructors* and *defined symbols*, respectively.

Definition 31 (Spareness). *Let $\ell \to \mu \in \mathcal{S}$. A rewrite step $\ell\sigma \to_{\mathcal{S}} \mu\sigma$ is* spare *if $\sigma(x)$ is in normal form w.r.t. $\mathcal{S}$ for every $x \in \mathcal{V}$ that occurs more than once in some $r \in \mathrm{Supp}(\mu)$. A $\rightrightarrows_{\mathcal{S}}$-sequence is spare if each of its $\to_{\mathcal{S}}$-steps is spare. $\mathcal{S}$ is spare if each $\rightrightarrows_{\mathcal{S}}$-sequence that starts with $\{1 : t\}$ for a basic term t is spare. A term $t \in \mathcal{T}(\Sigma, \mathcal{V})$ is* basic *if $t = f(t_1, \ldots, t_n)$ such that $f \in \Sigma_D$ and $t_i \in \mathcal{T}(\Sigma_C, \mathcal{V})$ for all $1 \leq i \leq n$.*

Example 32. Consider the PTRS $\mathcal{S}_7$ with the two rules:

$$\mathsf{g} \to \{{}^{3}/{}_{4} : \mathsf{d}(\bot), {}^{1}/{}_{4} : \mathsf{g}\} \qquad\qquad \mathsf{d}(x) \to \{1 : \mathsf{c}(x, x)\}$$

It is similar to the PTRS $\mathcal{S}_1$ from Ex. 16, but we exchanged the symbols g and $\bot$ in the right-hand side of the g-rule. This PTRS is orthogonal but duplicating due to the d-rule. However, in any rewrite sequence that starts with $\{1 : t\}$ for a basic term t we can only duplicate the constructor symbol $\bot$ but no defined symbol. Hence, $\mathcal{S}_7$ is spare.

In general, it is undecidable whether a PTRS is spare, since spareness is already undecidable for non-probabilistic TRSs. However, there exist computable sufficient conditions for spareness, see [17].

If a PTRS 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 AST, the probability of termination. As in [17], which analyzed runtime complexity, we have to restrict ourselves to rewrite sequences that start with basic terms. So we only consider start terms where a single algorithm is applied to data, i.e., we may not have any nested defined symbols in our start terms. This leads to the following theorem, where "*on basic terms*" means that one only considers rewrite sequences that start with $\{1 : t\}$ for a basic term t. It can be proved by an analogous limit construction as in the proof of Thm. 17.

Theorem 33 (From iAST/iPAST to fAST/fPAST (3)). *If a PTRS $\mathcal{S}$ is orthogonal and spare, then*

$$\mathcal{S} \textit{ is fAST on basic terms} \iff \mathcal{S} \textit{ is iAST on basic terms}$$
$$\mathcal{S} \textit{ is fPAST on basic terms} \iff \mathcal{S} \textit{ is iPAST on basic terms}$$

While iAST on basic terms is the same as iAST in general, the requirement of basic start terms is real restriction for fAST, i.e., there exists PTRSs that are fAST on basic terms, but not fAST in general.

Counterexample 34. Consider the PTRS $\mathcal{S}_8$ with the two rules:

$$\mathsf{g} \to \{{}^3\!/\!{}_4 : \mathsf{s}(\mathsf{g}), {}^1\!/\!{}_4 : \bot\} \qquad\qquad \mathsf{f}(\mathsf{s}(x)) \to \{1 : \mathsf{c}(\mathsf{f}(x), \mathsf{f}(x))\}$$

This PTRS behaves similarly to $\mathcal{S}_1$ (see Ex. 16). It is not fAST (and thus, also not fPAST), as we have $\{1 : \mathsf{f}(\mathsf{g})\} \rightrightarrows_{\mathcal{S}_8}^2 \{{}^3\!/\!{}_4 : \mathsf{c}(\mathsf{f}(\mathsf{g}), \mathsf{f}(\mathsf{g})), {}^1\!/\!{}_4 : \mathsf{f}(\bot)\}$, which corresponds to a random walk biased towards non-termination (since $\frac{3}{4} > \frac{1}{2}$).

However, the only basic terms for this PTRS are g and $\mathsf{f}(t)$ for terms t that do not contain g or f. A sequence starting with g corresponds to flipping a biased coin and a sequence starting with $\mathsf{f}(t)$ will clearly terminate. Hence, $\mathcal{S}_8$ is fAST (and even fPAST) on basic terms. Furthermore, note that $\mathcal{S}_8$ is iPAST (and thus, also iAST) analogous to $\mathcal{S}_1$. This shows that Thm. 33 cannot be extended to fAST or fPAST in general.

One may wonder whether Thm. 33 can nevertheless be used in order to prove fAST of a PTRS $\mathcal{S}$ on all terms by using a suitable transformation from $\mathcal{S}$ to another PTRS $\mathcal{S}'$ such that $\mathcal{S}$ is fAST on all terms iff $\mathcal{S}'$ is fAST on basic terms.

There is an analogous difference in the complexity analysis of non-probabilistic term rewrite systems. There, the concept of *runtime complexity* is restricted to rewrite sequences that start with a basic term, whereas the concept of *derivational complexity* allows arbitrary start terms. In [19], a transformation was presented that extends any (non-probabilistic) TRS $\mathcal{R}$ by so-called 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 fAST *on basic terms* to fAST in general.

Lemma 35. *A PTRS $\mathcal{S}$ is fAST iff $\mathcal{S} \cup \mathcal{G}(\mathcal{S})$ is fAST on basic terms.*

For every defined symbol f, the idea of the transformation is to introduce a new constructor symbol cons_f and for every function symbol f it introduces a new defined symbol enc_f. As an example for $\mathcal{S}_8$ from Ex. 32, then instead of starting with the non-basic term $\mathsf{c}(\mathsf{g}, \mathsf{f}(\mathsf{g}))$, we start with the basic term $\mathsf{enc}_\mathsf{c}(\mathsf{cons}_\mathsf{g}, \mathsf{cons}_\mathsf{f}(\mathsf{cons}_\mathsf{g}))$, its so-called *basic variant*. The new defined symbol enc_c is used to first build the term $\mathsf{c}(\mathsf{g}, \mathsf{f}(\mathsf{g}))$ 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 $\mathsf{c}(\mathsf{g}, \mathsf{f}(\mathsf{g}))$ 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 [19] (just with trivial probabilities in the right-hand sides $\ell \to \{1 : r\}$), since we do not need any probabilities during this initial construction of the original start term.

Definition 36 (Generator Rules $\mathcal{G}(\mathcal{S})$). *Let $\mathcal{S}$ be a PTRS over the signature Σ. Its* generator rules *$\mathcal{G}(\mathcal{S})$ are the following set of rules*

$$\begin{aligned}
&\{\mathsf{enc}_f(x_1,\ldots,x_n) \to \{1 : f(\mathsf{argenc}(x_1),\ldots,\mathsf{argenc}(x_n))\} \mid f \in \Sigma\}\\
\cup\ &\{\mathsf{argenc}(\mathsf{cons}_f(x_1,\ldots,x_n)) \to \{1 : f(\mathsf{argenc}(x_1),\ldots,\mathsf{argenc}(x_n))\} \mid f \in \Sigma_D\}\\
\cup\ &\{\mathsf{argenc}(f(x_1,\ldots,x_n)) \to \{1 : f(\mathsf{argenc}(x_1),\ldots,\mathsf{argenc}(x_n))\} \mid f \in \Sigma_C\},
\end{aligned}$$

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

Example 37. For the PTRS $\mathcal{S}_8$ from Ex. 34, we obtain the following generator rules $\mathcal{G}(\mathcal{S}_8)$:

$$\begin{aligned}
\mathsf{enc}_\mathsf{g} &\to \{1 : \mathsf{g}\}\\
\mathsf{enc}_\mathsf{f}(x_1) &\to \{1 : \mathsf{f}(\mathsf{argenc}(x_1))\}\\
\mathsf{enc}_\mathsf{c}(x_1, x_2) &\to \{1 : \mathsf{c}(\mathsf{argenc}(x_1), \mathsf{argenc}(x_2))\}\\
\mathsf{enc}_\mathsf{s}(x_1) &\to \{1 : \mathsf{s}(\mathsf{argenc}(x_1))\}\\
\mathsf{enc}_\bot &\to \{1 : \bot\}\\
\mathsf{argenc}(\mathsf{cons}_\mathsf{g}) &\to \{1 : \mathsf{g}\}\\
\mathsf{argenc}(\mathsf{cons}_\mathsf{f}(x_1)) &\to \{1 : \mathsf{f}(\mathsf{argenc}(x_1))\}\\
\mathsf{argenc}(\mathsf{c}(x_1, x_2)) &\to \{1 : \mathsf{c}(\mathsf{argenc}(x_1), \mathsf{argenc}(x_2))\}\\
\mathsf{argenc}(\mathsf{s}(x_1)) &\to \{1 : \mathsf{s}(\mathsf{argenc}(x_1))\}\\
\mathsf{argenc}(\bot) &\to \{1 : \bot\}
\end{aligned}$$

As mentioned, using the symbols cons_f and enc_f, as in [19] every term over Σ can be transformed into a basic term over $\Sigma \cup \Sigma_{\mathcal{G}(\mathcal{S})}$.

However, even if $\mathcal{S}$ is spare, the PTRS $\mathcal{S} \cup \mathcal{G}(\mathcal{S})$ 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 $\mathsf{enc_f}(x_1) \to \{1 : \mathsf{f}(\mathsf{argenc}(x_1))\}$ where a defined symbol argenc occurs below the duplicating symbol f on the right-hand side. Indeed, while $\mathcal{S}_8$ is spare, $\mathcal{S}_8 \cup \mathcal{G}(\mathcal{S}_8)$ is not. For example, when starting with the basic term $\mathsf{enc_f}(\mathsf{s}(\mathsf{cons_g}))$, we have

$$\begin{aligned} \{1 : \mathsf{enc_f}(\mathsf{s}(\mathsf{cons_g}))\} &\rightrightarrows^2_{\mathcal{G}(\mathcal{S}_8)} \{1 : \mathsf{f}(\mathsf{s}(\mathsf{argenc}(\mathsf{cons_g})))\} \\ &\rightrightarrows_{\mathcal{S}_8} \{1 : \mathsf{c}(\mathsf{f}(\mathsf{argenc}(\mathsf{cons_g})), \mathsf{f}(\mathsf{argenc}(\mathsf{cons_g}))), \end{aligned}$$

where the last step is not spare. In general, $\mathcal{S} \cup \mathcal{G}(\mathcal{S})$ is guaranteed to be spare if $\mathcal{S}$ is right-linear. So we could modify Thm. 33 into a theorem which states that $\mathcal{S}$ is fAST on all terms iff $\mathcal{S} \cup \mathcal{G}(\mathcal{S})$ is iAST on basic terms (and thus, on all terms) for orthogonal and right-linear PTRSs $\mathcal{S}$. However, this theorem would be subsumed by Thm. 17, where we already showed the equivalence of fAST and iAST if $\mathcal{S}$ is orthogonal and right-linear. Indeed, our goal in Thm. 33 was to find a weaker requirement than right-linearity. Hence, such a transformational approach to move from fAST on all start terms to fAST on basic terms does not seem viable for Thm. 33.

Finally, we can also combine our results on simultaneous rewriting and spareness to relax both left- and right-linearity in case of basic start terms. The proof for the following theorem combines the proofs for Thm. 27 and Thm. 33.

Theorem 38 (From iAST/iPAST to fAST/fPAST (4)). *If $\mathcal{S}$ is non-overlapping and spare, then*

$$\mathcal{S} \textit{ is fAST on basic terms} \Longleftarrow \mathcal{S} \textit{ is iAST w.r.t. } \rightarrowtail_{\mathcal{S}} \textit{ on basic terms}$$
$$\mathcal{S} \textit{ is fPAST on basic terms} \Longleftarrow \mathcal{S} \textit{ is iPAST w.r.t. } \rightarrowtail_{\mathcal{S}} \textit{ on basic terms}$$

6 Conclusion and Evaluation

In this paper, we presented numerous new results on the relationship between full and restricted forms of AST, including several criteria for PTRSs such that innermost AST implies full AST. All of our results also hold for PAST, and all of our criteria are suitable for automation (for spareness, there exist sufficient conditions that can be checked automatically).

We implemented our new criteria in our termination prover AProVE [21]. For every PTRS, one can indicate whether one wants to analyze its termination behavior for all start terms or only for basic start terms. Up to now, AProVE's main technique for termination analysis of PTRSs was the probabilistic DP framework from [29, 32] which however can only prove iAST. If one wants to analyze fAST for a PTRS $\mathcal{S}$, then AProVE now first tries to prove that the conditions of Thm. 33 are satisfied if one is restricted to basic start terms, or that the conditions of Thm. 17 hold if one wants to consider arbitrary start terms. If this succeeds, then we can use the full probabilistic DP framework in order to prove iAST, which then implies fAST. Otherwise, we try to prove all conditions of Thm. 38 or Thm. 27, respectively. If this succeeds, then we can use most of the processors from the probabilistic DP framework to prove iAST, which again implies fAST. If none of these theorems can be applied, then AProVE tries to prove fAST using a direct application of polynomial orderings [29]. Note that for AST w.r.t. basic start terms, Thm. 33 generalizes Thm. 17 and Thm. 38 generalizes Thm. 27, since right-linearity implies spareness.

For our evaluation, we compare the *old* AProVE without any of the new theorems (which only uses direct applications of polynomial orderings to prove fAST), to variants of AProVE where we activated each of the theorems individually, and finally to the *new* AProVE strategy explained above. The following diagram shows the theoretical subsumptions of each of these strategies for basic start terms, where an arrow from strategy A to strategy B means that B is strictly better than A.

$$\begin{tikzcd}
 & \text{Thm. 17} \arrow[r] & \text{Thm. 33} \arrow[dr] & \\
\textit{old}\ \mathsf{AProVE} \arrow[ur] \arrow[dr] & & & \textit{new}\ \mathsf{AProVE} \\
 & \text{Thm. 27} \arrow[r] & \text{Thm. 38} \arrow[ur] &
\end{tikzcd}$$

We used the benchmark set of 100 PTRSs from [32], and extended it by 15 new PTRSs that contain all the examples presented in this paper and some additional examples which illustrate the power of each strategy. AProVE can prove iAST for 93 of these 118 PTRSs. The following table shows for how many of these 93 PTRSs the respective strategy allows us to conclude fAST for basic start terms from AProVE's proof of iAST.

old AProVE	Thm. 17	Thm. 27	Thm. 33	Thm. 38	*new* AProVE
36	48	44	58	56	61

From the 61 examples that we can solve by using both Thm. 33 and Thm. 38 in "*new* AProVE", 5 examples (that are all right-linear) can only be solved by Thm. 33, 3 examples (where one is right-linear and the others only spare) can only be solved by Thm. 38, and 53 examples can be solved by both. If one considers arbitrary start terms, then the *new* AProVE can conclude fAST (using only Thm. 17 and Thm. 27) for 49 examples.

Currently, we only use the switch from full to innermost rewriting as a preprocessing step before applying the DP framework. As future work, we want to develop a processor within the DP framework that can perform this switch in a modular way. Then, the criteria of our theorems do not have to be required for the whole PTRS anymore, but just for specific sub-problems within the termination proof. This, however, requires developing a DP framework for fAST directly, which we will investigate in future work.

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/

In addition, an artifact is available at [31].

Acknowledgements. We thank Stefan Dollase for pointing us to [19].

References

[1] G. Agha, J. Meseguer, and K. Sen. “PMaude: Rewrite-based Specification Language for Probabilistic Object Systems”. In: *Proc. QAPL ’05*. ENTCS 153. 2006, pp. 213–239. DOI: 10.1016/j.entcs.2005.10.040.

[2] S. Agrawal, K. Chatterjee, and P. Novotný. “Lexicographic Ranking Supermartingales: An Efficient Approach to Termination of Probabilistic Programs”. In: *Proc. ACM Program. Lang.* 2.POPL (2017). DOI: 10.1145/3158122.

[3] T. Arts and J. Giesl. “Termination of Term Rewriting Using Dependency Pairs”. In: *Theor. Comput. Sc.* 236.1-2 (2000), pp. 133–178. DOI: 10.1016/S0304-3975(99)00207-8.

[4] M. Avanzini, U. Dal Lago, and A. Yamada. “On Probabilistic Term Rewriting”. In: *Sci. Comput. Program.* 185 (2020). DOI: 10.1016/j.scico.2019.102338.

[5] M. Avanzini, G. Moser, and M. Schaper. “A Modular Cost Analysis for Probabilistic Programs”. In: *Proc. ACM Program. Lang.* 4.OOPSLA (2020). DOI: 10.1145/3428240.

[6] F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998. DOI: 10.1017/CBO9781139172752.

[7] K. Batz, B. L. Kaminski, J.-P. Katoen, C. Matheja, and L. Verscht. “A Calculus for Amortized Expected Runtimes”. In: *Proc. ACM Program. Lang.* 7.POPL (2023). DOI: 10.1145/3571260.

[8] R. Beutner and L. Ong. “On Probabilistic Termination of Functional Programs with Continuous Distributions”. In: *Proc. PLDI ’21*. 2021, pp. 1312–1326. DOI: 10.1145/3453483.3454111.

[9] O. Bournez and C. Kirchner. “Probabilistic Rewrite Strategies. Applications to ELAN”. In: *Proc. RTA ’02*. LNCS 2378. 2002, pp. 252–266. DOI: 10.1007/3-540-45610-4_18.

[10] O. Bournez and F. Garnier. “Proving Positive Almost-Sure Termination”. In: *Proc. RTA ’05*. LNCS 3467. 2005, pp. 323–337. DOI: 10.1007/978-3-540-32033-3_24.

[11] K. Chatterjee, H. Fu, and P. Novotný. “Termination Analysis of Probabilistic Programs with Martingales”. In: *Foundations of Probabilistic Programming*. Ed. by G. Barthe, J. Katoen, and A. Silva. Cambridge University Press, 2020, 221–258. DOI: 10.1017/9781108770750.008.

[12] U. Dal Lago and C. Grellois. “Probabilistic Termination by Monadic Affine Sized Typing”. In: *Proc. ESOP ’17*. LNCS 10201. 2017, pp. 393–419. DOI: 10.1007/978-3-662-54434-1_15.

[13] U. Dal Lago, C. Faggian, and S. R. Della Rocca. “Intersection Types and (Positive) Almost-Sure Termination”. In: *Proc. ACM Program. Lang.* 5.POPL (2021). DOI: 10.1145/3434313.

[14] A. Díaz-Caro and G. Martínez. “Confluence in Probabilistic Rewriting”. In: *Proc. LSFA ’17*. ENTCS 338. 2018, pp. 115–131. DOI: 10.1016/j.entcs.2018.10.008.

[15] C. Faggian. "Probabilistic Rewriting and Asymptotic Behaviour: On Termination and Unique Normal Forms". In: *Log. Methods in Comput. Sci.* 18.2 (2022). DOI: 10.46298/lmcs-18(2:5)2022.

[16] L. M. Ferrer Fioriti and H. Hermanns. "Probabilistic Termination: Soundness, Completeness, and Compositionality". In: *Proc. POPL '15.* 2015, pp. 489–501. DOI: 10.1145/2676726.2677001.

[17] F. Frohn and J. Giesl. "Analyzing Runtime Complexity via Innermost Runtime Complexity". In: *Proc. LPAR '17.* EPiC 46. 2017, pp. 249–228. DOI: 10.29007/1nbh.

[18] H. Fu and K. Chatterjee. "Termination of Nondeterministic Probabilistic Programs". In: *Proc. VMCAI '19.* LNCS 11388. 2019, pp. 468–490. DOI: 10.1007/978-3-030-11245-5_22.

[19] C. Fuhs. "Transforming Derivational Complexity of Term Rewriting to Runtime Complexity". In: *Proc. FroCoS '19.* LNCS 11715. 2019, pp. 348–364. DOI: 10.1007/978-3-030-29007-8_20.

[20] J. Giesl, R. Thiemann, P. Schneider-Kamp, and S. Falke. "Mechanizing and Improving Dependency Pairs". In: *J. Autom. Reason.* 37.3 (2006), pp. 155–203. DOI: 10.1007/s10817-006-9057-7.

[21] J. Giesl, C. Aschermann, M. Brockschmidt, F. Emmes, F. Frohn, C. Fuhs, J. Hensel, C. Otto, M. Plücker, P. Schneider-Kamp, T. Ströder, S. Swiderski, and R. Thiemann. "Analyzing Program Termination and Complexity Automatically with AProVE". In: *J. Autom. Reason.* 58.1 (2017), pp. 3–31. DOI: 10.1007/s10817-016-9388-y.

[22] J. Giesl, P. Giesl, and M. Hark. "Computing Expected Runtimes for Constant Probability Programs". In: *Proc. CADE '19.* LNCS 11716. 2019, pp. 269–286. DOI: 10.1007/978-3-030-29436-6_16.

[23] A. D. Gordon, T. A. Henzinger, A. V. Nori, and S. K. Rajamani. "Probabilistic Programming". In: *Proc. FOSE '14.* 2014, pp. 167–181. DOI: 10.1145/2593882.2593900.

[24] B. Gramlich. "Abstract Relations between Restricted Termination and Confluence Properties of Rewrite Systems". In: *Fundamenta Informaticae* 24 (1995), pp. 2–23. DOI: 10.3233/FI-1995-24121.

[25] R. Gutiérrez and S. Lucas. "MU-TERM: Verify Termination Properties Automatically (System Description)". In: *Proc. IJCAR '20.* LNCS 12167. 2020, pp. 436–447. DOI: 10.1007/978-3-030-51054-1_28.

[26] M. Huang, H. Fu, K. Chatterjee, and A. K. Goharshady. "Modular Verification for Almost-Sure Termination of Probabilistic Programs". In: *Proc. ACM Program. Lang.* 3.OOPSLA (2019). DOI: 10.1145/3360555.

[27] B. L. Kaminski, J.-P. Katoen, C. Matheja, and F. Olmedo. "Weakest Precondition Reasoning for Expected Runtimes of Randomized Algorithms". In: *J. ACM* 65 (2018), pp. 1–68. DOI: 10.1145/3208102.

[28] B. L. Kaminski, J. Katoen, and C. Matheja. "Expected Runtime Analyis by Program Verification". In: *Foundations of Probabilistic Programming.* Ed. by G. Barthe, J. Katoen, and A. Silva. Cambridge University Press, 2020, 185–220. DOI: 10.1017/9781108770750.007.

[29] J.-C. Kassing and J. Giesl. "Proving Almost-Sure Innermost Termination of Probabilistic Term Rewriting Using Dependency Pairs". In: *Proc. CADE '23*. LNCS 14132. 2023, pp. 344–364. DOI: 10.1007/978-3-031-38499-8_20.

[30] J.-C. Kassing, F. Frohn, and J. Giesl. "From Innermost to Full Almost-Sure Termination of Probabilistic Term Rewriting". In: *CoRR* abs/2310.06121 (2023). DOI: 10.48550/arXiv.2310.06121.

[31] J.-C. Kassing, F. Frohn, and J. Giesl. *From Innermost to Full Almost-Sure Termination of Probabilistic Term Rewriting* - AProVE *Artifact*. 2024. DOI: 10.5281/zenodo.10449299.

[32] J.-C. Kassing, S. Dollase, and J. Giesl. "A Complete Dependency Pair Framework for Almost-Sure Innermost Termination of Probabilistic Term Rewriting". In: *Proc. FLOPS '24*. LNCS. To appear. Long version at *CoRR* abs/2309.00344. 2024. DOI: 10.48550/arXiv.2309.00344.

[33] M. Korp, C. Sternagel, H. Zankl, and A. Middeldorp. "Tyrolean Termination Tool 2". In: *Proc. RTA '09*. LNCS 5595. 2009, pp. 295–304. DOI: 10.1007/978-3-642-02348-4_21.

[34] M. R. K. Krishna Rao. "Some Characteristics of Strong Innermost Normalization". In: *Theor. Comput. Sc.* 239 (2000), pp. 141–164. DOI: 10.1016/S0304-3975(99)00215-7.

[35] L. Leutgeb, G. Moser, and F. Zuleger. "Automated Expected Amortised Cost Analysis of Probabilistic Data Structures". In: *Proc. CAV '22*. LNCS 13372. 2022, pp. 70–91. DOI: 10.1007/978-3-031-13188-2_4.

[36] A. McIver, C. Morgan, B. L. Kaminski, and J.-P. Katoen. "A New Proof Rule for Almost-Sure Termination". In: *Proc. ACM Program. Lang.* 2.POPL (2018). DOI: 10.1145/3158121.

[37] F. Meyer, M. Hark, and J. Giesl. "Inferring Expected Runtimes of Probabilistic Integer Programs Using Expected Sizes". In: *Proc. TACAS '21*. LNCS 12651. 2021, pp. 250–269. DOI: 10.1007/978-3-030-72016-2_14.

[38] M. Moosbrugger, E. Bartocci, J. Katoen, and L. Kovács. "Automated Termination Analysis of Polynomial Probabilistic Programs". In: *Proc. ESOP '21*. LNCS 12648. 2021, pp. 491–518. DOI: 10.1007/978-3-030-72019-3_18.

[39] M. H. A. Newman. "On Theories with a Combinatorial Definition of Equivalence". In: *Annals of Mathematics* 43.2 (1942), pp. 223–242. URL: http://www.ens-lyon.fr/LIP/REWRITING/TERMINATION/NEWMAN/Newman.pdf.

[40] V. C. Ngo, Q. Carbonneaux, and J. Hoffmann. "Bounded Expectations: Resource Analysis for Probabilistic Programs". In: *Proc. PLDI '18*. 2018, pp. 496–512. DOI: 10.1145/3192366.3192394.

[41] M. J. O'Donnell. *Computing in Systems Described by Equations*. LNCS 58. 1977. DOI: 10.1007/3-540-08531-9.

[42] N. Saheb-Djahromi. "Probabilistic LCF". In: *Proc. MFCS '78*. LNCS 64. 1978, pp. 442–451. DOI: 10.1007/3-540-08921-7_92.

[43] R. Thiemann and C. Sternagel. “Certification of Termination Proofs Using CeTA”. In: *Proc. TPHOLs ’09*. LNCS 5674. 2009, pp. 452–468. DOI: 10.1007/978-3-642-03359-9_31.

[44] Y. Toyama. “Counterexamples to the Termination for the Direct Sum of Term Rewriting Systems”. In: *Inf. Proc. Lett.* 25 (1987), pp. 141–143. DOI: 10.1016/0020-0190(87)90122-0.

[45] D. Wang, D. M. Kahn, and J. Hoffmann. “Raising Expectations: Automating Expected Cost Analysis with Types”. In: *Proc. ACM Program. Lang.* 4.ICFP (2020). DOI: 10.1145/3408992.

[46] A. Yamada, K. Kusakari, and T. Sakabe. “Nagoya Termination Tool”. In: *Proc. RTA-TLCA ’14*. LNCS 8560. 2014, pp. 466–475. DOI: 10.1007/978-3-319-08918-8_32.