Logical Methods in Computer Science
Volume 21, Issue 4, 2025, pp. 27:1–27:57
https://lmcs.episciences.org/

Submitted Aug. 18, 2024
Published Dec. 08, 2025

DERIVATIVES ON GRAPHS FOR THE POSITIVE CALCULUS OF RELATIONS WITH TRANSITIVE CLOSURE

YOSHIKI NAKAMURA

Institute of Science Tokyo, Japan
e-mail address: nakamura.yoshiki.ny@gmail.com

Abstract. We prove that the equational theory of the positive calculus of relations with transitive closure (PCoR*) is EXPSPACE-complete. Here, PCoR* terms consist of the following standard operators on binary relations: identity, empty, universality, union, intersection, composition, converse, and reflexive transitive closure (so, PCoR* terms subsume Kleene algebra and allegory terms as fragments). Additionally, we show that the equational theory of PCoR* extended with tests and nominals (in hybrid logic) is still EXPSPACE-complete; moreover, it is PSPACE-complete for its intersection-free fragment.

To this end, we design derivatives on *graphs* by extending derivatives on words for regular expressions. The derivatives give a finite automata construction on path decompositions, like those on words. Because the equational theory has a linearly bounded pathwidth model property, we can decide the equational theory of PCoR* using these automata.

1. Introduction

We consider *the positive calculus of relations with transitive closure* (PCoR*) [Pou18]: the algebraic system on binary relations, with the operators $\{1, 0, \top, ;, +, \cap, \_^{\smile}, \_^{*}\}$ of identity (1), empty (0), universality ($\top$), composition ($;$), union ($+$), intersection ($\cap$), converse ($\_^{\smile}$), and reflexive transitive closure ($\_^{*}$); so, PCoR* consist of the operators of Kleene algebra $\{1, 0, ;, +, \_^{*}\}$ [Kle56, Con71, Koz91], those of allegory $\{1, ;, \cap, \_^{\smile}\}$ [FS90], and $\top$. PCoR* terms without $\top$ are sometimes called *Kleene allegory* terms [BP15, BP17, Nak17].

If we added the complement operator ($\_^{-}$) to PCoR*, then we would obtain the calculus of relations [Tar41, TG87] with transitive closure [Ng84]. However, the equational theory of the calculus of relations is well-known undecidable [Tar41, TG87]. The undecidability result holds even for the terms with the three operators $\{;, +, \_^{-}\}$ [Hir18] and moreover even with one variable [Nak19]. Also, in PCoR* with $\_^{-}$, even if the complement only applies to atomic terms (variables or constants), the equational theory is still undecidable [Nak23]. In this paper, by excluding complement, we focus on *positive* fragments. For Kleene algebras (with respect to binary relations), the equational theory coincides with the language equivalence problem of regular expressions (see, *e.g.*, [Pou18, Theorem 4]), and thus the equational theory of Kleene algebras is decidable and PSPACE-complete, by

Key words and phrases: Relation algebra, Kleene algebra, Derivative, Relational intersection, Series-Parallel Graph, Pathwidth.

This paper is a revised and extended version of [Nak17].

DOI:10.46298/LMCS-21(4:27)2025

the results on regular expressions [MS72]. For representable allegories (with respect to binary relations), the equational theory coincides with the graph homomorphic equivalence problem via an encoding of terms as graphs [FS90, AB95, CM77], and thus the equational theory of representable allegories is also decidable. For the equational theory of PCoR*, Brunet and Pous extended the graph homomorphism characterizations above using graph languages [BP15, BP17]. By introducing Petri automata, which is an automata model for expressing graph languages, they have shown that the equational theory of *identity-free Kleene lattices* $\{0, ;, +, \cap, \_^{+}\}$ with respect to binary relations [AMN11] is decidable and EXPSPACE-complete [BP15, BP17] where $(\_^{+})$ denotes the transitive closure operator. However, the decidability and complexity for Kleene allegories (and PCoR*) were left open at LICS 2015 [BP15].

Our main contribution is to prove that the equational theory of PCoR* is still decidable and EXPSPACE-complete. We first note that the equational theory has the treewidth at most 2 model property (Proposition 2.8) and the linearly bounded pathwidth model property (Proposition 2.9). Thus, the decidability can be derived from the decidability of MSO over bounded treewidth structures [Cou88, Cou90, CE12]. However, the naive algorithm obtained from this fact has a non-elementary complexity (see Section 8, for more comparisons to other systems related to PCoR*). The EXPSPACE-hardness is shown by giving a reduction from the universality problem of regular expressions with intersection, which is EXPSPACE-complete [Fü80, Theorem 2]. (This was independently shown in [Nak17, BP17].)

To obtain the EXPSPACE upper bound of the equational theory of PCoR*, we use the idea of derivatives on words for regular expressions, *e.g.*, Brzozowski's derivatives [Brz64] and Antimirov's (partial) derivatives [Ant96, BBM+16], that are tools to obtain automata from regular expressions syntactically (we refer to the book [Sak09], for more details of derivatives on words). In this paper, we extend the derivatives from words to graphs. To this end, we consider Kleene lattices $\{1, 0, ;, +, \cap, \_^{*}\}$, which is a core fragment of PCoR*, and we extend the syntax of terms with *labels* for pointing multiple vertices on graphs (Section 3). Thanks to this extension, similar to those on words, we can give derivatives on graphs/structures (Section 4). Derivatives of labeled Kleene lattice terms can simulate left quotients of languages of runs (having a special form of DAG) (Section 2.2.1), as with that derivatives of regular expressions can simulate left quotients of word languages. Moreover, we show that our derivatives are decomposable on path decompositions (Theorem 5.5). Namely, from derivatives on (small) pre-glued graphs, we can compute the derivative on the (large) glued graph (in this sense, our approach can be viewed as a variant of the Feferman–Vaught decomposition theorem [Mak04, CE12]). Thanks to this decomposition with the linearly bounded pathwidth model property (Proposition 2.9), we can obtain an automata construction for Kleene lattice terms (Section 5). Using this construction, we have that the equational theory of Kleene lattices with respect to relations is EXPSPACE-complete (Corollary 5.13). Moreover, we show that on our automata construction, we can give encodings of the following operators: universality ($\top$), converse ($\_^{\smile}$), tests in Kleene algebra with tests (KAT), and nominals in hybrid logic. Thus, the equational theory of PCoR* is EXPSPACE-complete (Corollary 6.3), and moreover, the equational theory of PCoR* with tests and nominals is EXPSPACE-complete (Corollary 6.6). Additionally, if the intersection width [GLL09] is fixed (particularly, if $\cap$ does not occur), the equational theory of PCoR* with tests and nominals is PSPACE-complete (Corollary 6.7).

Differences with the version at LICS 2017. This is a revised and extended version of the paper [Nak17], presented at the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2017). The differences from the conference version are as follows.

- To express bounded pathwidth structures, we use a graph gluing operator ($\odot$) [BP16a], instead of "Sequential Graph Constructing Procedures" (SGCPs) [Nak17]. Roughly speaking, SGCPs enumerate "normalized" (but complete to generate all structures of pathwidth at most k) path decompositions of width at most k, but combinations of the gluing operator enumerate *all* path decompositions of width at most k. Due to this, the alphabet size, here, the number of structures of size at most $k+1$, is large, naively, but we can reduce it (see Section 6.3). Thanks to this separation of concerns, our formalizations are significantly simplified without changing the essence.
- Thanks to the above, we can easily encode some additional operators, *e.g.*, top ($\top$), converse ($\_^{\smile}$), tests in Kleene algebra with tests, and nominals in hybrid logic (Sections 6.1 and 6.2). In the body of this paper, relying on these encodings, instead of PCoR* terms, we mainly consider Kleene lattice terms—PCoR* terms with neither $\top$ nor $\_^{\smile}$. This separation also slightly simplifies our formalizations.
- We use runs (defined in Section 2.2.1) instead of "simple graphs" in [Nak17]. Because each vertex of runs precisely has one fun-in and one fun-out, we can more easily define left quotients, *cf.* [Nak17]. Our runs are essentially those of branching automata by Lodaya and Weil [LW98, LW00, LW01] without minor changes.
- We point out an error of [Nak17, Section IV] in Section 7.4. We give a counterexample that the decomposition rules corresponding to the argument in [Nak17, Lemma IV.10] are not complete (*cf.* Lemma 7.10), due to one-way reading. To avoid this problem, we use two-way alternating finite automata (2AFAs), not one-way.

Outline. In Section 2, we give basic definitions, including PCoR* terms and Kleene lattice terms (KL terms). In Section 3, we extend KL terms with labels. In Section 4, we define derivatives on structures (graphs without interfaces) for labeled KL terms. In Section 5, we give an automata construction using the derivatives. We then have that the equational theory of KL terms is EXPSPACE-complete. In Section 6, we give encodings of additional operators in our automata construction. Consequently, the equational theory of PCoR* terms with tests and nominals is also EXPSPACE-complete. In Section 7, we prove the decomposition theorem for the derivatives, which is the key to our automata construction in Section 5. In Section 8, we conclude this paper with related and future work.

2. Preliminaries

We write $\mathbb{N}$ for the set of non-negative integers. For $l, r \in \mathbb{N}$, we write $[l, r]$ for the set $\{i \in \mathbb{N} \mid l \leq i \leq r\}$. For $n \in \mathbb{N}$, we abbreviate $[1, n]$ to $[n]$. For a set X, we write $\#X$ for the cardinality of X and $\wp(X)$ for the power set of X. We use $\uplus$ to denote that the union $\cup$ is disjoint.

Word languages. We write wv for the concatenation of words w and v. We write ε for the *empty word*. For a set X of *letters*, we write X^* for the set of *words* over X. A *language* over X is a subset of X^*. We use w, v to denote words and use $\mathcal{L}, \mathcal{K}$ to denote languages. For languages $\mathcal{L}, \mathcal{K} \subseteq X^*$, the *concatenation* $\mathcal{L} ; \mathcal{K}$, the *n-th iteration* $\mathcal{L}^n$ (where $n \in \mathbb{N}$), the *Kleene plus* $\mathcal{L}^+$, and the *Kleene star* $\mathcal{L}^*$ are defined by:

$$\mathcal{L} ; \mathcal{K} \triangleq \{wv \mid w \in \mathcal{L} \wedge v \in \mathcal{K}\}, \quad \mathcal{L}^n \triangleq \begin{cases} \mathcal{L} ; \mathcal{L}^{n-1} & (n \geq 1) \\ \{\varepsilon\} & (n = 0) \end{cases}, \quad \mathcal{L}^+ \triangleq \bigcup_{n \geq 1} \mathcal{L}^n, \quad \mathcal{L}^* \triangleq \bigcup_{n \geq 0} \mathcal{L}^n.$$

Binary relations. We write $\triangle_A$ for the identity relation on a set A: $\triangle_A \triangleq \{\langle x, x\rangle \mid x \in A\}$. For binary relations R, S on a set B, the *composition* $R ; S$, the *converse* $R^{\smile}$, the *n-th iteration* R^n (where $n \in \mathbb{N}$), the *transitive closure* R^+, and the *reflexive transitive closure* R^* are defined by:

$$R ; S \triangleq \{\langle x, z\rangle \mid \exists y,\ \langle x, y\rangle \in R \wedge \langle y, z\rangle \in S\}, \qquad R^{\smile} \triangleq \{\langle x, y\rangle \mid \langle y, x\rangle \in R\},$$

$$R^n \triangleq \begin{cases} R ; R^{n-1} & (n \geq 1) \\ \triangle_B & (n = 0) \end{cases}, \qquad R^+ \triangleq \bigcup_{n \geq 1} R^n, \qquad R^* \triangleq \bigcup_{n \geq 0} R^n.$$

Graphs with bi-interface. We consider graphs with bi-interface, inspired by [BP16a, bi-interface graphs][BGK+16, symmetric monoidal structures]. Interfaces are used to define the series composition (Definition 2.1). Let A be a set with a map $\mathrm{ty}\colon A \to \mathbb{N}^2$; we also let ty_1 and ty_2 be such that $\mathrm{ty}(a) = \langle \mathrm{ty}_1(a), \mathrm{ty}_2(a)\rangle$ for each $a \in A$. For $n, m \in \mathbb{N}$, a *graph* G over A (with $\langle n, m\rangle$-interface) is a tuple $\langle |G|, \{a^G\}_{a \in A}, 1^G_1, \dots, 1^G_n, 2^G_1, \dots, 2^G_m\rangle$, where

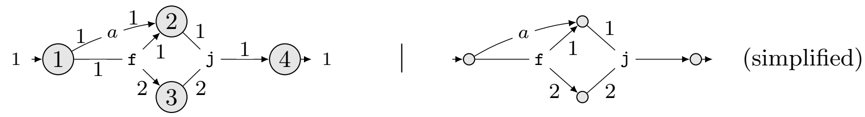
- $|G|$ is a (possibly empty) set;
- $a^G \subseteq |G|^{\mathrm{ty}_1(a)+\mathrm{ty}_2(a)}$ is a $(\mathrm{ty}_1(a) + \mathrm{ty}_2(a))$-ary relation for each $a \in A$;
- $1^G_l \in |G|$ is the l-th *source* vertex for each $l \in [n]$;
- $2^G_r \in |G|$ is the r-th *target* vertex for each $r \in [m]$.

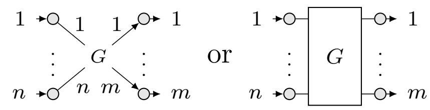
Particularly when $n = m = 0$, we say that G is a graph without interfaces or G has no interfaces. Note that, if $|G| = \emptyset$, then $n = m = 0$. We let $\mathrm{ty}(G) \triangleq \langle \mathrm{ty}_1(G), \mathrm{ty}_2(G)\rangle \triangleq \langle n, m\rangle$. We often abbreviate 1^G_1 to 1^G and 2^G_1 to 2^G, respectively.

Let G and H be graphs over a set A with $\langle n, m\rangle$-interface. We say that a map $f\colon |G| \to |H|$ is a *graph homomorphism* from G to H, written $f\colon G \longrightarrow H$, if (1) for each $a \in A$ and $x_1, \dots, x_p \in |G|$, if $\langle x_1, \dots, x_p\rangle \in a^G$, then $\langle f(x_1), \dots, f(x_p)\rangle \in a^H$, (2) for each $l \in [n]$, $f(1^G_l) = 1^H_l$, and (3) for each $r \in [m]$, $f(2^G_r) = 2^H_r$. Particularly, we say that f is a *graph isomorphism* from G to H, written $f\colon G \cong H$, if f is a bijective graph homomorphism from G to H, and for each $a \in A$ and $x_1, \dots, x_p \in |G|$, $\langle x_1, \dots, x_p\rangle \in a^G$ iff $\langle f(x_1), \dots, f(x_p)\rangle \in a^H$. We write $G \longrightarrow H$ (resp. $G \cong H$) if there is some $f\colon G \longrightarrow H$ (resp. $f\colon G \cong H$). In the sequel, we identify two graphs if there is a graph isomorphism between them (except for structures; see Section 3.1); we write $G = H$ when $G \cong H$.

For instance, when A is the set $\{a, \mathtt{f}, \mathtt{j}\}$ with $\mathrm{ty}(a) = \langle 1, 1\rangle$, $\mathrm{ty}(\mathtt{f}) = \langle 1, 2\rangle$, and $\mathrm{ty}(\mathtt{j}) = \langle 2, 1\rangle$ and the graph G is given by $|G| = \{1, 2, 3, 4\}$, $a^G = \{\langle 1, 2\rangle\}$, $\mathtt{f}^G = \{\langle 1, 2, 3\rangle\}$, $\mathtt{j}^G = \{\langle 2, 3, 4\rangle\}$, $\mathrm{ty}(G) = \langle 1, 1\rangle$, $1^G_1 = 1$, and $2^G_1 = 4$, we depict G as the figure on the left-hand side. Here, we will use the symbols $\mathtt{f}$ and $\mathtt{j}$ as special constants (Section 2.2.1).

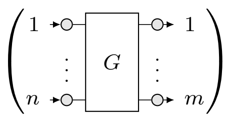
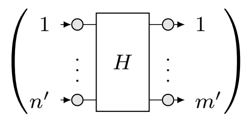
For notational simplicity, we often omit vertex labels and some "1" labels as the right-hand side.

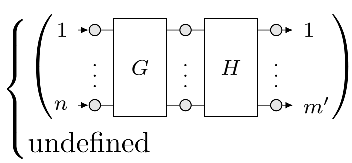
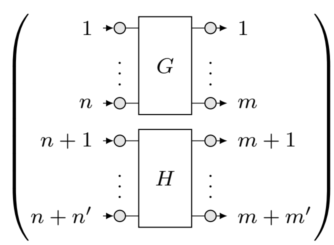


We may depict graphs G with $\mathrm{ty}(G) = \langle n, m\rangle$ as one edge:  .

Additionally, for an equivalence relation E on $|G|$, the *quotient graph* of G with respect to E is defined as the graph $G/E \triangleq \langle |G|/E, \{\langle X, Y\rangle \mid \exists x \in X, y \in Y, \langle x, y\rangle \in a^G\}_{a\in A}, [1^G_1]_E, \ldots, [1^G_n]_E, [2^G_1]_E, \ldots, [2^G_m]_E\rangle$ where $|G|/E$ denotes the set of equivalence classes of $|G|$ by E and $[x]_E$ denotes the equivalence class of x.

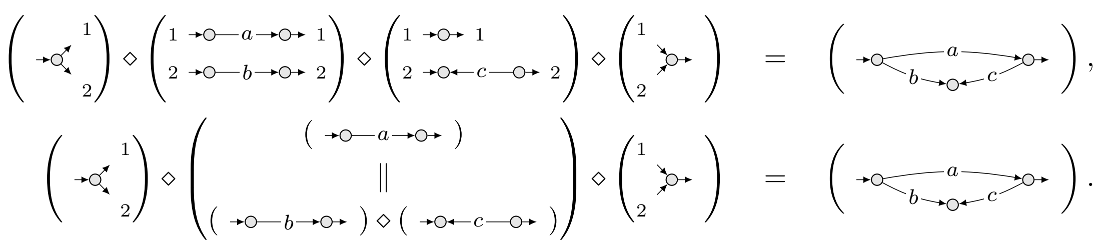
Series and parallel composition. For graphs, we use the series composition and the parallel composition, defined as follows.

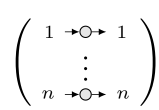
Definition 2.1. For graphs, $G =$  and $H =$ , the *series composition* $G \diamond H$ and the *parallel composition* $G \parallel H$ are defined as follows, respectively:

$G \diamond H \triangleq$  $(m = n')$, undefined (otherwise), $\quad G \parallel H \triangleq$  .

($G \diamond H$ is the graph obtained from G and H by merging 2^G_i and 1^H_i for $i \in [m]$.)

Additionally, we may denote $G \parallel H$ vertically as $\genfrac{}{}{0pt}{}{G}{\genfrac{}{}{0pt}{}{\parallel}{H}}$. For instance,



For $n \in \mathbb{N}$, we let 1^n be the following graph with $\langle n, n\rangle$-interface: $1^n \triangleq$ .

Each 1^n is an *empty* graph and the graph 1^0 is the *null* graph. Each 1^n is the left identity element with respect to $\diamond$ on graphs G with $\langle n, m\rangle$-interface, that is $1^n \diamond G = G$. The right identity element is given by 1^m, i.e., $G \diamond 1^m = G$. The identity element with respect to $\parallel$ on graphs with $\langle n, m\rangle$-interface is given by 1^0, i.e., $1^0 \parallel G = G \parallel 1^0 = G$.

Pathwidth and treewidth. We recall the notions of the pathwidth and treewidth of structures [CE12, Definition 9.12], which are defined based on the pathwidth [RS83] (see also, *e.g.*, [Bod98, Theorem 2][BP16a, Lemma 4.6] for alternative characterizations) and the treewidth [RS86] for graphs. Below, for graphs with bi-interface, we use a slightly generalized definition, based on [CE12, Definition 2.5.3] for treewidth and [Nak17] for pathwidth (these definitions will be useful in the constructions in Propositions 2.4 and 2.5, while we will always consider structures (graphs without interfaces)). For a graph G (with bi-interface), a *path decomposition* of G is a sequence $\vec{H} = H_1 \dots H_n$ of finite graphs such that

- $|G| = \bigcup_{i \in [n]} |H_i|$ and $a^G = \bigcup_{i \in [n]} a^{H_i}$ for each $a \in A$;
- $|H_i| \cap |H_k| \subseteq |H_j|$ for all $1 \le i \le j \le k \le n$;
- each source vertex of G is in $|H_1|$ and each target vertex of G is in $|H_n|$.

The *width* of $\vec{H}$ is $\max_{i \in [n]}(\#|H_i| - 1)$. The *pathwidth* $\mathrm{pw}(G)$ of a graph G is the minimum width among path decompositions of G. Each element of $\vec{H}$ is called a *bag*.

Similarly, for a graph G, a *tree decomposition* of G is a finite rooted tree $\vec{H} = \{H_w\}_{w \in \Gamma}$ of graphs, where $\Gamma \subseteq \mathbb{N}^*$ is a finite and prefix-closed ($ww' \in \Gamma$ implies $w \in \Gamma$) set, such that

- $|G| = \bigcup_{w \in \Gamma} |H_w|$ and $a^G = \bigcup_{w \in \Gamma} a^{H_w}$ for each $a \in A$;
- $|H_w| \cap |H_u| \subseteq |H_v|$ for all $w, v, u \in \Gamma$ such that v is on the unique path between w and u (i.e., v satisfies $\mathsf{LCA}(w, u) \preceq_{\mathrm{pref}} v \preceq_{\mathrm{pref}} w$ or $\mathsf{LCA}(w, u) \preceq_{\mathrm{pref}} v \preceq_{\mathrm{pref}} u$ where $w' \preceq_{\mathrm{pref}} v'$ denotes that w' is a prefix of v' and $\mathsf{LCA}(w, u)$ denotes the maximal common prefix word (i.e., the least common ancestor) of w and u);
- each source vertex and each target vertex of G is in the root $|H_\varepsilon|$.

The *width* of $\vec{H}$ is $\max_{w \in \Gamma}(\#|H_w| - 1)$. The *treewidth* $\mathrm{tw}(G)$ of a graph G is the minimum width among tree decompositions of G. Each element of $\vec{H}$ is called a *bag*.

Particularly when G has no interfaces, the pathwidth/treewidth of a graph G coincide with those of its Gaifman graph (i.e., the graph $\langle |G|, E \rangle$ where E is the binary relation $\bigcup_{a \in A} \bigcup_{\langle x_1, \dots, x_k \rangle \in a^G} \{\langle x_i, x_j \rangle \mid i, j \in [k] \wedge x_i \neq x_j\}$ [EF95, p. 26]), respectively.

Remark 2.2. By definition, the number of bags is finite; this is sufficient in this paper as we are mainly interested in finite structures (thanks to the Proposition 2.7).

2.1. PCoR*: The Positive Calculus of Relations with Transitive Closure.

We consider the positive calculus of relations with transitive closure (PCoR*). We use Σ to denote a set of *variables*. The set of *PCoR* terms* over Σ is defined as the set of terms over the signature $\{\mathsf{1}_{(0)}, \mathsf{0}_{(0)}, \top_{(0)}, ;_{(2)}, +_{(2)}, \cap_{(2)}, \_^{\smile}_{(1)}, \_^{*}_{(1)}\}$ and the variable set Σ:

$$t, s, u \;::=\; a \mid \mathsf{1} \mid \mathsf{0} \mid \top \mid t \,;\, s \mid t + s \mid t \cap s \mid t^{\smile} \mid t^{*}. \qquad (a \in \Sigma)$$

We often abbreviate $t \,;\, s$ to ts. We use parentheses in ambiguous situations (where $+$, $;$, and $\cap$ are left-associative). We write $\sum_{i=1}^{n} t_i$ for the term $\mathsf{0} + t_1 + \dots + t_n$, write $\mathop{;}_{i=1}^{n} t_i$ for the term $\mathsf{1} \,;\, t_1 \,;\, \dots \,;\, t_n$, write t^n for the term $\mathop{;}_{i=1}^{n} t$, and write t^+ for the term tt^*.

An *equation* $t = s$ is a pair of PCoR* terms. The *inequation* $t \le s$ abbreviates the equation $t + s = s$.

For a PCoR* term t, the *size* $\|t\|$ is the number of symbols occurring in t:

$$\|a\| \triangleq 1 \text{ for } a \in \Sigma \cup \{\mathsf{1}, \mathsf{0}, \top\}, \qquad \|t^{\heartsuit}\| \triangleq 1 + \|t\| \text{ for } \heartsuit \in \{\smile, *\},$$
$$\|t \heartsuit s\| \triangleq 1 + \|t\| + \|s\| \text{ for } \heartsuit \in \{;, +, \cap\}.$$

Additionally, the *intersection width* $\mathrm{iw}(t)$ [GLL09] is defined as follows:

$$\mathrm{iw}(a) \triangleq 1 \text{ for } a \in \Sigma \cup \{1, 0, \top\}, \qquad \mathrm{iw}(t^{\heartsuit}) \triangleq \mathrm{iw}(t) \text{ for } \heartsuit \in \{\smile, *\},$$
$$\mathrm{iw}(t \heartsuit s) \triangleq \max(\mathrm{iw}(t), \mathrm{iw}(s)) \text{ for } \heartsuit \in \{;, +\}, \qquad \mathrm{iw}(t \cap s) \triangleq \mathrm{iw}(t) + \mathrm{iw}(s).$$

For instance, $\mathrm{iw}(a \cap b \cap c \cap d) = 4$ and $\mathrm{iw}((a \cap b) ; (a \cap c) ; (a \cap d)) = 2$.

Proposition 2.3. *For all PCoR* terms t, we have* $\mathrm{iw}(t) \leq \|t\|$.

Proof. By easy induction on t. □

2.1.1. *Semantics: relational models.* A *structure (of binary relations)* is a non-empty graph without interfaces over the set Σ with $\mathrm{ty}(a) = \langle 1, 1\rangle$ for $a \in \Sigma$. We use $\mathfrak{A}, \mathfrak{B}$ to denote a structure. Let $\mathfrak{A}$ be a structure. For a PCoR* term t, the *semantics* $[\![t]\!]_{\mathfrak{A}} \subseteq |\mathfrak{A}|^2$ is defined as follows:

$$[\![a]\!]_{\mathfrak{A}} \triangleq a^{\mathfrak{A}}, \quad [\![1]\!]_{\mathfrak{A}} \triangleq \triangle_{|\mathfrak{A}|}, \quad [\![0]\!]_{\mathfrak{A}} \triangleq \emptyset, \quad [\![\top]\!]_{\mathfrak{A}} \triangleq |\mathfrak{A}|^2, \quad [\![t ; s]\!]_{\mathfrak{A}} \triangleq [\![t]\!]_{\mathfrak{A}} ; [\![s]\!]_{\mathfrak{A}},$$
$$[\![t + s]\!]_{\mathfrak{A}} \triangleq [\![t]\!]_{\mathfrak{A}} \cup [\![s]\!]_{\mathfrak{A}}, \quad [\![t \cap s]\!]_{\mathfrak{A}} \triangleq [\![t]\!]_{\mathfrak{A}} \cap [\![s]\!]_{\mathfrak{A}}, \quad [\![t^{\smile}]\!]_{\mathfrak{A}} \triangleq [\![t]\!]_{\mathfrak{A}}^{\smile}, \quad [\![t^*]\!]_{\mathfrak{A}} \triangleq [\![t]\!]_{\mathfrak{A}}^{*}.$$

We write REL for the class of all structures. For $\mathcal{C} \subseteq \mathsf{REL}$, we write $\mathcal{C} \models t = s$ if $[\![t]\!]_{\mathfrak{A}} = [\![s]\!]_{\mathfrak{A}}$ for all $\mathfrak{A} \in \mathcal{C}$. In the sequel, we consider the *equational theory* with respect to REL. For instance, the following equations hold with respect to REL:

$$(t^* s^*)^* = (t + s)^* \quad (2.1) \qquad t(s \cap u) \leq (ts) \cap (tu) \quad (2.3) \qquad t^+ \cap 1 \leq (tt)^+ \quad (2.5)$$
$$t^* = (tt)^*(1 + t) \quad (2.2) \qquad (ts) \cap u \leq (t \cap (us^{\smile}))s \quad (2.4) \qquad (t \cap s^{\smile})^+ \cap 1 \leq (t \cap s^+)^+ \quad (2.6)$$

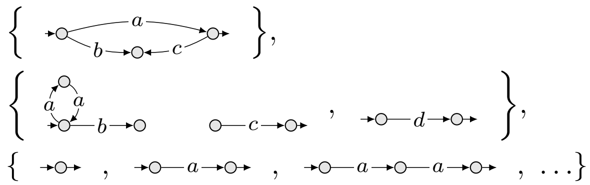
Notably, the equational theory of PCoR* contains that of Kleene algebras $\{1, 0, ;, +, \_^*\}$ [Koz91] (such as Equations (2.1) and (2.2)) and that of allegories $\{1, ;, \cap, \_^{\smile}\}$ [FS90, PV18] (such as Equation (2.3) (the semi-distributive law over $\cap$) and Equation (2.4) (the modularity law)); see also [BÉS95, BP16b] for Kleene algebras with converse $\{1, 0, ;, +, \_^*, \_^{\smile}\}$ and [AMN11, DP18] for identity-free Kleene lattices $\{0, ;, +, \cap, \_^+\}$. Equation (2.5) [Pou18, p. 14] and Equation (2.6) are non-trivial equations having transitive closure and intersection.

2.1.2. *An alternative semantics: graph languages.* We recall graph languages for PCoR* [BP17][Pou18, Definition 15]. The *graph language* $\mathcal{G}(t)$ of a PCoR* term t is the set of graphs with $\langle 1, 1\rangle$-interface over Σ, defined as follows:

$$\mathcal{G}(a) \triangleq \{ \to\!\circ\!-\!a\!\to\!\circ\!\to \} \text{ for } a \in \Sigma, \qquad \mathcal{G}(1) \triangleq \{ \to\!\circ\!\to \}, \quad \mathcal{G}(0) \triangleq \emptyset,$$
$$\mathcal{G}(t ; s) \triangleq \{ \to\!\circ\!-\!G\!\to\!\circ\!-\!H\!\to\!\circ\!\to \mid G \in \mathcal{G}(t) \wedge H \in \mathcal{G}(s)\}, \qquad \mathcal{G}(\top) \triangleq \{ \to\!\circ \quad \circ\!\to \},$$
$$\mathcal{G}(t \cap s) \triangleq \left\{ \to\!\circ\!\!<^{G}_{H}\!\!>\!\circ\!\to \mid G \in \mathcal{G}(t) \wedge H \in \mathcal{G}(s)\right\}, \qquad \mathcal{G}(t + s) \triangleq \mathcal{G}(t) \cup \mathcal{G}(s),$$
$$\mathcal{G}(t^{\smile}) \triangleq \{ \to\!\circ\!\leftarrow\!G\!-\!\circ\!\to \mid G \in \mathcal{G}(t)\}, \qquad \mathcal{G}(t^*) \triangleq \bigcup_{n \geq 0} \mathcal{G}(t^n).$$

For instance,

$\mathcal{G}(a \cap (bc^{\smile})) = \{\ \ldots\ \}$, $\mathcal{G}((((aa) \cap 1)b\top c) + d) = \{\ \ldots\ \}$, $\mathcal{G}(a^*) = \{\ \ldots\ \}$:



Proposition 2.4. *For all PCoR* terms t and $G \in \mathcal{G}(t)$, we have* $\mathrm{tw}(G) \leq 2$.

Proof (Corollary of [Bod98, Theorem 41]*).* By induction on $\{;, \cap\}$-terms, we have that every series-parallel graph (with one source and one target) has some tree decomposition of width 2 [Bod98, Theorem 41]. Similar to this, by induction on PCoR* terms t, each graph $G \in \mathcal{G}(t)$ has some tree decomposition of width 2. □

Proposition 2.5. *For all PCoR* terms t and $G \in \mathcal{G}(t)$, we have* $\mathrm{pw}(G) \leq \mathrm{iw}(t)$.

Proof. Similar to the above, by easy induction on t, each graph $G \in \mathcal{G}(t)$ has some path decomposition $\vec{H} = H_1 \ldots H_n$ of width $\mathrm{iw}(t)$ (where the source vertex is in $|H_1|$ and the target vertex is in $|H_n|$). □

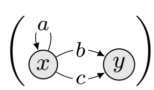
In this paper, we use the graph languages above as an alternative semantics. Let $\mathfrak{A}$ be a structure. For $x, y \in |\mathfrak{A}|$, we write $\mathsf{G}(\mathfrak{A}, x, y)$ for the graph defined by $\langle |\mathfrak{A}|, \{[\![a]\!]_{\mathfrak{A}}\}_{a \in \Sigma}, x, y\rangle$. For a graph H and a set $\mathcal{G}$ of graphs, we define $[\![H]\!]_{\mathfrak{A}}$ and $[\![\mathcal{G}]\!]_{\mathfrak{A}}$ as follows:

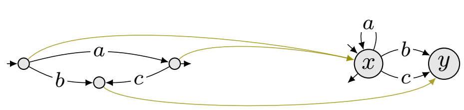
$$[\![H]\!]_{\mathfrak{A}} \triangleq \{\langle x, y\rangle \mid H \longrightarrow \mathsf{G}(\mathfrak{A}, x, y)\}, \qquad [\![\mathcal{G}]\!]_{\mathfrak{A}} \triangleq \bigcup_{H \in \mathcal{G}} [\![H]\!]_{\mathfrak{A}}.$$

We then have the following.

Proposition 2.6 [AMN11, Lemma 2.1 and 2.2] (for PCoR* without $\top$), [BP17, Lemma 2.3] (for PCoR*)**.** *Let $\mathfrak{A}$ be a structure. For all PCoR* terms t, we have $[\![t]\!]_{\mathfrak{A}} = [\![\mathcal{G}(t)]\!]_{\mathfrak{A}}$.*

Proof. By easy induction on t using distributivity of ;, $\cap$, and $\_^{\smile}$. □

For instance, when $\mathfrak{A} =$ , we have $\langle x, x\rangle \in [\![a \cap (bc^{\smile})]\!]_{\mathfrak{A}}$ by the following graph homomorphism:

$\mathcal{G}(a \cap (bc^{\smile})) \ni$  $= \mathsf{G}(\mathfrak{A}, x, x)$.

As a corollary of Proposition 2.6, we also have the following bounded model property.

Proposition 2.7 (Graph language bounded model property)**.** *For all PCoR* terms t, s,*

$$\mathsf{REL} \models t \leq s \iff \{\mathfrak{A} \mid \exists x, y \in |\mathfrak{A}|, \mathsf{G}(\mathfrak{A}, x, y) \in \mathcal{G}(t)\} \models t \leq s.$$

Proof. ($\Longrightarrow$): Trivial. ($\Longleftarrow$): We prove the contraposition. Assume $\langle x, y\rangle \in [\![t]\!]_{\mathfrak{A}} \setminus [\![s]\!]_{\mathfrak{A}}$ for some $\mathfrak{A} \in \mathsf{REL}$ and $x, y \in |\mathfrak{A}|$. By Proposition 2.6, $G' \longrightarrow \mathsf{G}(\mathfrak{A}, x, y)$ for some $G' \in \mathcal{G}(t)$, and $H \not\longrightarrow \mathsf{G}(\mathfrak{A}, x, y)$ for all $H \in \mathcal{G}(s)$. Let $\mathfrak{B}$, x', and y' be such that $\mathsf{G}(\mathfrak{B}, x', y') = G'$. Since $G' \longrightarrow \mathsf{G}(\mathfrak{B}, x', y')$, we have $\langle x', y'\rangle \in [\![t]\!]_{\mathfrak{B}}$. Also, by $\mathsf{G}(\mathfrak{B}, x', y') \longrightarrow \mathsf{G}(\mathfrak{A}, x, y)$, we have $H \not\longrightarrow \mathsf{G}(\mathfrak{B}, x', y')$ for all $H \in \mathcal{G}(s)$, and hence $\langle x', y'\rangle \notin [\![s]\!]_{\mathfrak{B}}$. Hence by $\mathsf{G}(\mathfrak{B}, x', y') \in \mathcal{G}(t)$, this completes the proof. □

For a class $\mathcal{C} \subseteq \mathsf{REL}$, we use the following notations:

$$\mathcal{C}_{\mathrm{pw}\leq k} = \{\mathfrak{A} \in \mathcal{C} \mid \mathrm{pw}(\mathfrak{A}) \leq k\}, \qquad \mathcal{C}_{\mathrm{tw}\leq k} = \{\mathfrak{A} \in \mathcal{C} \mid \mathrm{tw}(\mathfrak{A}) \leq k\}.$$

As a corollary of Proposition 2.7, we also have the following two model properties.

Proposition 2.8 (Treewidth at most 2 model property). *For all PCoR* terms t, s,*

$$\mathsf{REL} \models t \leq s \iff \mathsf{REL}_{\mathrm{tw}\leq 2} \models t \leq s.$$

Proof. ($\Longrightarrow$): Trivial. ($\Longleftarrow$): Because every graph of $\mathcal{G}(t)$ has treewidth at most 2 (Proposition 2.4), this direction is shown by Proposition 2.7. □

Proposition 2.9 (Linearly bounded pathwidth model property). *For all PCoR* terms t, s,*

$$\mathsf{REL} \models t \leq s \iff \mathsf{REL}_{\mathrm{pw}\leq \mathrm{iw}(t)} \models t \leq s.$$

Proof. ($\Longrightarrow$): Trivial. ($\Longleftarrow$): Because every graph of $\mathcal{G}(t)$ has pathwidth at most $\mathrm{iw}(t)$ (Proposition 2.5), this direction is shown by Proposition 2.7. □

Note that we can easily translate the equational theory of PCoR* terms into the theory of monadic second-order logic (MSO) formulas[1] as an analog of the standard translation from the equational theory of the calculus of relations into the theory of three-variable first-order logic formulas [Tar41]. Because the theory of MSO over bounded treewidth structures is decidable [Cou88, Cou90, CE12], Proposition 2.8 (or Proposition 2.9) implies that the equational theory of PCoR* is decidable. However, the naive algorithm obtained from the above has a non-elementary complexity. We will show that the equational theory of PCoR* is EXPSPACE-complete (Corollary 6.3).

Additionally, Brunet and Pous gave the following graph-theoretic characterization of the equational theory for PCoR*, which generalizes the characterizations known for conjunctive queries [CM77], representable allegories [FS90] and union-free relation algebras [AB95].

Proposition 2.10 [BP15, BP17]. *For all PCoR* terms t, s,*

$$\mathsf{REL} \models t \leq s \iff \forall G \in \mathcal{G}(t), \exists H \in \mathcal{G}(s), H \longrightarrow G.$$

2.1.3. *Comparison to word languages.* The word language $[t]_\Sigma$ of a Kleene lattice term (i.e., terms over $\{1, 0, ;, +, \cap, \_^*\}$) t is the set of words over Σ defined as follows:

$$[a]_\Sigma \triangleq \{a\} \text{ for } a \in \Sigma, \qquad [1]_\Sigma \triangleq \{\varepsilon\}, \qquad [0]_\Sigma \triangleq \emptyset,$$
$$[t+s]_\Sigma \triangleq [t]_\Sigma \cup [s]_\Sigma, \qquad [t\,;s]_\Sigma \triangleq [t]_\Sigma\,;[s]_\Sigma, \qquad [t^*]_\Sigma \triangleq [t]_\Sigma^*, \qquad [t \cap s]_\Sigma \triangleq [t]_\Sigma \cap [s]_\Sigma.$$

For Kleene algebra terms (i.e., terms over $\{1, 0, ;, +, \_^*\}$) t and s, the following is well-known (see, *e.g.*, [Pou18, Theorem 4]): $\mathsf{REL} \models t \leq s$ iff $[t]_\Sigma \subseteq [s]_\Sigma$. This equivalence can be slightly strengthened as follows.

Proposition 2.11 [Nak17, BP17]. *For all Kleene algebra terms t and Kleene lattice terms s, we have:*

$$\mathsf{REL} \models t \leq s \iff [t]_\Sigma \subseteq [s]_\Sigma.$$

[1]Precisely, with respect to binary relations, the expressive power of PCoR* is equivalent to that of three-variable existential positive logic with "variable-confined" monadic transitive closure [Nak22].

Proof. (In [Nak17, Theorem VI.3], this was shown via the tree unwinding technique in modal logic. The following is a simpler proof [BP17, Proposition 10.2] based on the graph language characterization.) For all Kleene lattice terms s and words $w \in \Sigma^*$, we have:

$$\exists H \in \mathcal{G}(s), H \longrightarrow \mathsf{G}(w) \iff w \in [s]_\Sigma$$

where $\mathsf{G}(w)$ denotes the unique graph such that $\mathcal{G}(w) = \{\mathsf{G}(w)\}$. It is shown by easy induction on s. Thus by Proposition 2.10, this completes the proof. □

Remark 2.12. The (word) language model LANG [BÉS95, AMN11, Bru17] is another interesting model of PCoR* terms, where the constant $\top$ is interpreted as the universal language, the unary operator $\_^{\smile}$ is interpreted as the reverse of the language: $L^{\smile} \triangleq \{a_n \dots a_2 a_1 \mid a_1 a_2 \dots a_n \in L\}$, and the other operators are defined based on the above $[\_]_\Sigma$. In LANG, each variable x may interpret not only $\{x\}$ but also a (possibly non-singleton) language. Notably for identity-free Kleene lattices, the equational theory with respect to LANG coincides with that with respect to REL [AMN11, Theorem 4.1]. However, for Kleene lattices and PCoR*, this coincidence is broken, *e.g.*, Equations (2.4) to (2.6) are counterexamples. For the equational theory of Kleene lattices (with complement), LANG can be characterized as a subclass of (linearly ordered) (generalized) relational models [Nak25].

Remark 2.13. The restriction of Proposition 2.11 is crucial. Proposition 2.11 fails when $\cap$, $\top$, or $\_^{\smile}$ occurs in t. For instance, if $t = a \cap b$ and $s = \mathsf{0}$ where $a \neq b$, then we have $[t]_\Sigma = [s]_\Sigma = \emptyset$ but $a \cap b \leq \mathsf{0}$ does not hold with respect to REL. Proposition 2.11 also fails when $\top$ or $\_^{\smile}$ occurs in s (where $\top$ expresses the universal language and $\_^{\smile}$ expresses the reverse, according to Remark 2.12); let us consider, $\mathsf{REL} \models a \leq a \top a$ or $\mathsf{REL} \models a \leq a a^{\smile} a$. In such cases, graph languages are needed (Proposition 2.10).

2.2. **Kleene lattice terms: a core fragment of PCoR* terms.**

In the sequel, we mainly consider Kleene lattice (KL) terms. We say that a PCoR* term t is a *KL term* if t contains neither $\top$ nor $\_^{\smile}$. Namely, the set of KL terms is given by:

$$t, s, u \ ::= \ a \mid \mathsf{1} \mid \mathsf{0} \mid t \mathbin{;} s \mid t + s \mid t \cap s \mid t^*. \qquad (a \in \Sigma)$$

2.2.1. *Second alternative semantics: run (DAG with vertices of one fan-in and one fan-out) languages.*

We consider run languages, which is a slightly modified semantics of graph languages. This is inspired by the runs in branching automata [LW98, LW00, LW01]. We use the symbols $\mathtt{f}$ and $\mathtt{j}$ for expressing the forking and joining of runs, respectively. Let $\Sigma_{\mathtt{f},\mathtt{j}}$ be the set $\Sigma \cup \{\mathtt{f}, \mathtt{j}\}$ with the map ty defined by: $\mathrm{ty}(a) = \langle 1, 1\rangle$ for $a \in \Sigma$, $\mathrm{ty}(\mathtt{f}) = \langle 1, 2\rangle$, and $\mathrm{ty}(\mathtt{j}) = \langle 2, 1\rangle$. A *run with $\langle n, m\rangle$-interface* G is a graph with $\langle n, m\rangle$-interface over $\Sigma_{\mathtt{f},\mathtt{j}}$ where

- G forms a directed acyclic graph (DAG), namely, the following holds:

$$\left(\bigcup_{a \in \Sigma_{\mathtt{f},\mathtt{j}}} \bigcup_{i \in [\mathrm{ty}_1(a)]} \bigcup_{j \in [\mathrm{ty}_2(a)]} \{\langle x_i, y_j\rangle \mid \langle x_1, \dots, x_{\mathrm{ty}_1(a)}, y_1, \dots, y_{\mathrm{ty}_2(a)}\rangle \in a^G\}\right)^+ \cap \triangle_{|G|} = \emptyset.$$

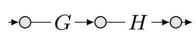
- $\mathsf{in}_G(z) = \mathsf{out}_G(z) = 1$ for each $z \in |G|$, where

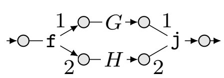
$$\mathsf{in}_G(z) \triangleq \#\{l \in [n] \mid 1_l^G = z\} + \sum_{\substack{a \in \Sigma_{\mathtt{f},\mathtt{j}} \\ i \in [\mathrm{ty}_1(a)]}} \#\{\langle x_1, \dots, x_{\mathrm{ty}_1(a)}, y_1, \dots, y_{\mathrm{ty}_2(a)} \rangle \in a^G \mid x_i = z\},$$

$$\mathsf{out}_G(z) \triangleq \#\{r \in [m] \mid 2_r^G = z\} + \sum_{\substack{a \in \Sigma_{\mathtt{f},\mathtt{j}} \\ j \in [\mathrm{ty}_2(a)]}} \#\{\langle x_1, \dots, x_{\mathrm{ty}_1(a)}, y_1, \dots, y_{\mathrm{ty}_2(a)} \rangle \in a^G \mid y_j = z\}.$$

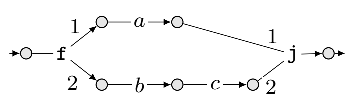
Definition 2.14. The *run language* $\mathcal{R}(t)$ of a KL term t is a set of runs with $\langle 1, 1\rangle$-interface, defined as follows:

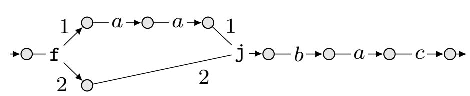
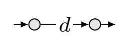
$\mathcal{R}(a) \triangleq \{$ $\}$ for $a \in \Sigma$, $\quad \mathcal{R}(1) \triangleq \{$ $\}$, $\mathcal{R}(0) \triangleq \emptyset$,

$\mathcal{R}(t \,;\, s) \triangleq \{$  $\mid G \in \mathcal{R}(t) \land H \in \mathcal{R}(s)\}$, $\quad \mathcal{R}(t + s) \triangleq \mathcal{R}(t) \cup \mathcal{R}(s)$,

$\mathcal{R}(t \cap s) \triangleq \{$  $\mid G \in \mathcal{R}(t) \land H \in \mathcal{R}(s)\}$, $\quad \mathcal{R}(t^*) \triangleq \bigcup_{n \geq 0} \mathcal{R}(t^n)$.

By construction, every run of KL terms is a directed two-terminal series-parallel graph [VTL79, Epp92], by viewing labels f and j as vertices. For instance (*cf.* Section 2.1.2),

$\mathcal{R}(a \cap (bc)) = \{$  $\}$,

$\mathcal{R}((((aa) \cap 1)bac) + d) = \{$  ,  $\}$,

$\mathcal{R}(a^*) = \{$, , $, \dots\}$.

We can also use the run languages as an alternative relational semantics. Let $\mathfrak{A}$ be a structure. For $\vec{x}, \vec{y} \in |\mathfrak{A}|^*$, let $\tilde{\mathsf{G}}(\mathfrak{A}, \vec{x}, \vec{y}) \triangleq \langle |\mathfrak{A}|, \{a^{\tilde{\mathsf{G}}(\mathfrak{A}, \vec{x}, \vec{y})}\}_{a \in \Sigma_{\mathtt{f},\mathtt{j}}}, \vec{x}, \vec{y} \rangle$ where $a^{\tilde{\mathsf{G}}(\mathfrak{A}, \vec{x}, \vec{y})} = [\![a]\!]_{\mathfrak{A}}$ for $a \in \Sigma$ and $\mathtt{f}^{\tilde{\mathsf{G}}(\mathfrak{A}, \vec{x}, \vec{y})} = \mathtt{j}^{\tilde{\mathsf{G}}(\mathfrak{A}, \vec{x}, \vec{y})} = \{\langle z, z, z \rangle \mid z \in |\mathfrak{A}|\}$.

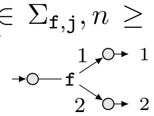
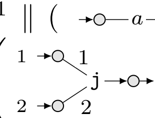
For a run H and a set $\mathcal{G}$ of runs, we define $[\![H]\!]_{\mathfrak{A}}$ and $[\![\mathcal{G}]\!]_{\mathfrak{A}}$ as follows:

$$[\![H]\!]_{\mathfrak{A}} \triangleq \{\vec{x}\vec{y} \mid \vec{x} \in |\mathfrak{A}|^{\mathrm{ty}_1(H)} \land \vec{y} \in |\mathfrak{A}|^{\mathrm{ty}_2(H)} \land H \longrightarrow \tilde{\mathsf{G}}(\mathfrak{A}, \vec{x}, \vec{y})\}, \qquad [\![\mathcal{G}]\!]_{\mathfrak{A}} \triangleq \bigcup_{H \in \mathcal{G}} [\![H]\!]_{\mathfrak{A}}.$$

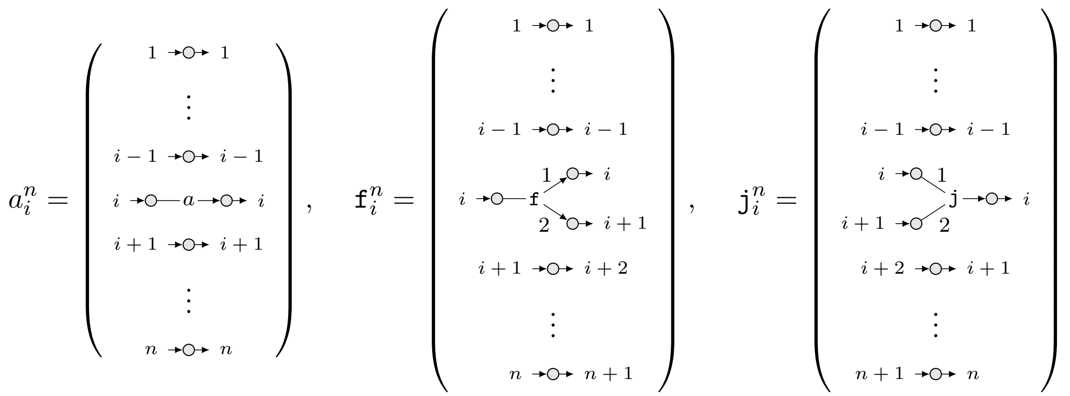
We then have the following as with Proposition 2.6.

Proposition 2.15. *Let $\mathfrak{A}$ be a structure. For all KL terms t, we have $[\![t]\!]_{\mathfrak{A}} = [\![\mathcal{R}(t)]\!]_{\mathfrak{A}}$.*

Proof. By easy induction on t using distributivity of ; and $\cap$. ☐

2.2.2. *Compositions on runs.* We recall the series composition $\diamond$, the parallel composition $\parallel$, and the identity elements 1^n (Definition 2.1). The set of *atomic runs* is defined as $\bar{\Sigma} \triangleq \{b_i^n \mid b \in \Sigma_{\mathtt{f},\mathtt{j}}, n \geq 1, i \in [n]\}$ where $a_i^n \triangleq 1^{i-1} \parallel ($ $) \parallel 1^{n-i}$ for $a \in \Sigma$, $\mathtt{f}_i^n \triangleq 1^{i-1} \parallel \Big($  $\Big) \parallel 1^{n-i}$, and $\mathtt{j}_i^n \triangleq 1^{i-1} \parallel \Big($  $\Big) \parallel 1^{n-i}$.

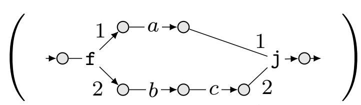
Namely, a_i^n, $\mathtt{f}_i^n$, and $\mathtt{j}_i^n$ are given as follows:



Every run can be expressed as series compositions of atomic runs.

Proposition 2.16. *For every run G with $n \geq 1$ edges, there exist some $a_1, \dots, a_n \in \bar{\Sigma}$ such that $G = a_1 \diamond \dots \diamond a_n$.*

Proof. By easy induction on n. □

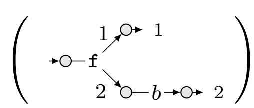
For instance,  $= \mathtt{f}_1^1 \diamond a_1^2 \diamond b_2^2 \diamond c_2^2 \diamond \mathtt{j}_1^1$. This decomposition is not unique if the "topological sort of edges" of the run (i.e., the linear ordering of edges such that for every vertex, each incoming edge of the vertex comes before each outgoing edge of the vertex in the ordering) is not unique. For instance, $\mathtt{f}_1^1 \diamond b_2^2 \diamond a_1^2 \diamond c_2^2 \diamond \mathtt{j}_1^1$ and $\mathtt{f}_1^1 \diamond b_2^2 \diamond c_2^2 \diamond a_1^2 \diamond \mathtt{j}_1^1$ also express the same run above.

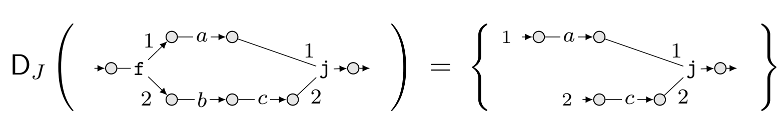
2.2.3. *Left quotients of runs.* For runs, we define *left quotients* with respect to series compositions. For runs G, H, and J, we write:

$$G \longrightarrow_J H \quad \stackrel{\triangle}{\iff} \quad J \diamond H \text{ is defined and } G = J \diamond H.$$

The *left quotient* $\mathsf{D}_J(G)$ of a run G with respect to a run J is the run language defined by:

$$\mathsf{D}_J(G) \ \stackrel{\triangle}{=} \ \{H \mid G \longrightarrow_J H\}.$$

For a run language $\mathcal{G}$, let $\mathsf{D}_J(\mathcal{G}) \stackrel{\triangle}{=} \bigcup_{G \in \mathcal{G}} \mathsf{D}_J(G)$. For instance, when $J =$ , we have:

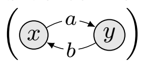


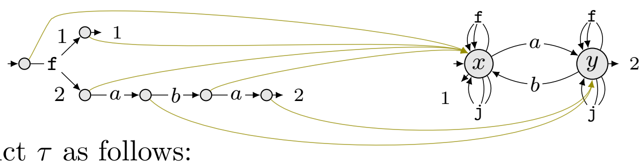
Let $n \in \mathbb{N}$. For a run G, the *emptiness property* $\mathsf{E}_n(G)$ is the truth value defined by:

$$\mathsf{E}_n(G) \ \stackrel{\triangle}{=} \begin{cases} \mathsf{true} & (G = 1^n) \\ \mathsf{false} & (\text{otherwise}). \end{cases}$$

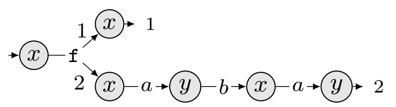
For a run language $\mathcal{G}$, let $\mathsf{E}_n(\mathcal{G}) \stackrel{\triangle}{=} \bigvee_{G \in \mathcal{G}} \mathsf{E}_n(G)$. The following is clear by definition.

Proposition 2.17. *For all runs G, we have $\mathsf{E}_{\mathrm{ty}_2(G)}(\mathsf{D}_G(G))$.*

2.3. **Runs on structures.** We also introduce *runs on structures* to simplify the use of the semantics using runs (Proposition 2.15). Let $\mathfrak{A}$ be a structure. An $\mathfrak{A}$-*run* with $\langle \vec{x}, \vec{y} \rangle$-interface is a graph homomorphism $\tau \colon H \longrightarrow \tilde{\mathsf{G}}(\mathfrak{A}, \vec{x}, \vec{y})$, where H is some run. We let $\mathrm{ty}(\tau) \triangleq \langle \mathrm{ty}_1(\tau), \mathrm{ty}_2(\tau) \rangle \triangleq \langle \vec{x}, \vec{y} \rangle$. We use τ, σ, ρ to denote $\mathfrak{A}$-runs. For brevity, we depict $\mathfrak{A}$-runs as vertex-labeled graphs where each vertex is labeled with a vertex of $\mathfrak{A}$. For instance, let $\mathfrak{A} =$  and we consider the following graph homomorphism τ:

 $= \tilde{\mathsf{G}}(\mathfrak{A}, x, xy)$.

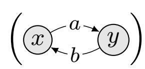
We then depict τ as follows:

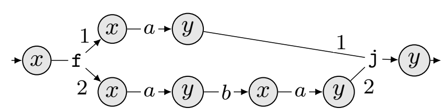


For two $\mathfrak{A}$-runs with $\langle \vec{x}, \vec{y} \rangle$-interface, $\tau \colon H \longrightarrow \tilde{\mathsf{G}}(\mathfrak{A}, \vec{x}, \vec{y})$ and $\sigma \colon J \longrightarrow \tilde{\mathsf{G}}(\mathfrak{A}, \vec{x}, \vec{y})$, we say that a map $f \colon |H| \to |J|$ is an $\mathfrak{A}$-*run isomorphism* from τ to σ, written $f \colon \tau \cong \sigma$, if $f \colon H \cong J$ and for all $x \in |H|$, $\tau(x) = \sigma(f(x))$. We write $\tau \cong \sigma$ if there is some f such that $f \colon \tau \cong \sigma$. In the sequel, we identify two $\mathfrak{A}$-runs if there is an $\mathfrak{A}$-run isomorphism between them; we write $\tau = \sigma$ when $\tau \cong \sigma$.

For a KL term t and $x, y \in |\mathfrak{A}|$, we write $\mathcal{R}^{\mathfrak{A}}_{x,y}(t)$ for the $\mathfrak{A}$-run language defined by:

$$\mathcal{R}^{\mathfrak{A}}_{x,y}(t) \;\triangleq\; \{\tau \colon H \longrightarrow \tilde{\mathsf{G}}(\mathfrak{A}, x, y) \mid H \in \mathcal{R}(t)\}.$$

For instance, when $\mathfrak{A} =$ , we have:

$$\mathcal{R}^{\mathfrak{A}}_{x,y}(a \cap (aba)) \;=\; \Bigg\{$$  $$\Bigg\}.$$

Using $\mathfrak{A}$-runs, we can rephrase Proposition 2.15 as follows.

Proposition 2.18. *Let $\mathfrak{A}$ be a structure. For all KL term t, we have:*

$$[\![t]\!]_{\mathfrak{A}} \;=\; \{\langle x, y \rangle \in |\mathfrak{A}|^2 \mid \mathcal{R}^{\mathfrak{A}}_{x,y}(t) \neq \emptyset\}.$$

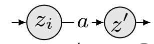
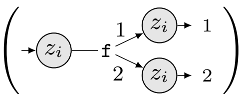
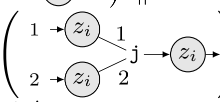
Proof. By easy induction on t (or an easy consequence of Proposition 2.15, as $\langle x, y \rangle \in [\![t]\!]_{\mathfrak{A}}$ iff $\exists H \in \mathcal{R}(t), \exists \tau, \tau \colon H \longrightarrow \tilde{\mathsf{G}}(\mathfrak{A}, x, y)$ iff $\mathcal{R}^{\mathfrak{A}}_{x,y}(t) \neq \emptyset$). ☐

2.3.1. *Compositions on $\mathfrak{A}$-runs.* We recall the series composition $\diamond$, the parallel composition $\parallel$, and the identity element 1^n (Definition 2.1) for graphs with bi-interface. Similarly, we define them also for $\mathfrak{A}$-runs.

Definition 2.19. Let $\tau \colon H \longrightarrow \tilde{\mathsf{G}}(\mathfrak{A}, \vec{x}, \vec{y})$ and $\sigma \colon J \longrightarrow \tilde{\mathsf{G}}(\mathfrak{A}, \vec{x}', \vec{y}')$ be $\mathfrak{A}$-runs. Under $\vec{y} = \vec{x}'$, the *series composition* $\tau \diamond \sigma \colon (H \diamond J) \longrightarrow \tilde{\mathsf{G}}(\mathfrak{A}, \vec{x}, \vec{y}')$ is defined (undefined when $\vec{y} \neq \vec{x}'$) as the map such that, for each vertex $x \in |H \diamond J|$, if x is induced from a vertex $y \in |H|$, then $(\tau \diamond \sigma)(x) = \tau(y)$, and if x is induced from a vertex $y \in |J|$, then $(\tau \diamond \sigma)(x) = \sigma(y)$. The *parallel composition* $\tau \parallel \sigma \colon (H \parallel J) \longrightarrow \tilde{\mathsf{G}}(\mathfrak{A}, \vec{x}\vec{x}', \vec{y}\vec{y}')$ is defined as the map such that if x is induced from a vertex $y \in |H|$, then $(\tau \parallel \sigma)(x) = \tau(y)$, and if x is induced from a vertex

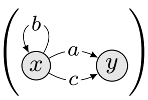
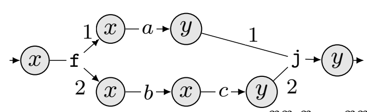
$y \in |J|$, then $(\tau \parallel \sigma)(x) = \sigma(y)$. For each $\vec{x} = x_1 \dots x_n \in |\mathfrak{A}|^*$, let $1^{\vec{x}} \colon 1^n \longrightarrow \tilde{\mathsf{G}}(\mathfrak{A}, \vec{x}, \vec{x})$ be the $\mathfrak{A}$-run such that for each $i \in [n]$, the i-th source/target vertex maps to x_i.

The left and right identity element with respect to $\diamond$ on $\mathfrak{A}$-runs with $\langle \vec{x}, \vec{y} \rangle$-interface are given by $1^{\vec{x}}$ and $1^{\vec{y}}$, respectively, i.e., $1^{\vec{x}} \diamond \tau = \tau \diamond 1^{\vec{y}} = \tau$. The identity element with respect to $\parallel$ on $\mathfrak{A}$-runs with $\langle \vec{x}, \vec{y} \rangle$-interface is given by 1^{ε}, i.e., $1^{\varepsilon} \parallel \tau = \tau \parallel 1^{\varepsilon} = \tau$.

For $\vec{z} = z_1 \dots z_n$, let $a_i^{\vec{z}, z'} \triangleq 1^{z_1 \dots z_{i-1}} \parallel ($  $) \parallel 1^{z_{i+1} \dots z_n}$, $\mathtt{f}_i^{\vec{z}} \triangleq 1^{z_1 \dots z_{i-1}} \parallel$ () $\parallel 1^{z_{i+1} \dots z_n}$, and $\mathtt{j}_i^{\vec{z}} \triangleq 1^{z_1 \dots z_{i-1}} \parallel$ () $\parallel 1^{z_{i+1} \dots z_n}$. The set of *atomic* $\mathfrak{A}$*-runs*, written $\bar{\Sigma}_{\mathfrak{A}}$, is defined as $\bar{\Sigma}_{\mathfrak{A}} \triangleq \{a_i^{\vec{z}, z'} \mid a \in \Sigma, n \geq 1, i \in [n], \vec{z} \in |\mathfrak{A}|^n, z' \in |\mathfrak{A}|, \langle z_i, z' \rangle \in [\![a]\!]_{\mathfrak{A}}\} \cup \{\mathtt{f}_i^{\vec{z}}, \mathtt{j}_i^{\vec{z}} \mid n \geq 1, i \in [n], \vec{z} \in |\mathfrak{A}|^n\}$. Every $\mathfrak{A}$-run can be expressed as series compositions of atomic $\mathfrak{A}$-runs, as follows (as with Proposition 2.16).

Proposition 2.20. *Let* $\mathfrak{A}$ *be a structure. For every* $\mathfrak{A}$*-run* τ *with* $n \geq 1$ *edges, there exist some* $a_1, \dots, a_n \in \bar{\Sigma}_{\mathfrak{A}}$ *such that* $\tau = a_1 \diamond \dots \diamond a_n$.

Proof. By easy induction on n. □

For instance, when $\mathfrak{A} =$ () and $\tau =$  , we have $\tau = \mathtt{f}_1^{x} \diamond a_1^{xx,y} \diamond b_2^{yx,x} \diamond c_2^{yx,y} \diamond \mathtt{j}_1^{y}$. The decomposition of τ is not unique, as $\mathtt{f}_1^{x} \diamond b_2^{xx,x} \diamond a_1^{xx,y} \diamond c_2^{yx,y} \diamond \mathtt{j}_1^{y}$ and $\mathtt{f}_1^{x} \diamond b_2^{xx,x} \diamond c_2^{xx,y} \diamond a_1^{xy,y} \diamond \mathtt{j}_1^{y}$ also express τ.

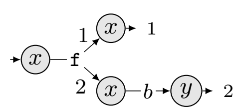
2.3.2. *Left quotients of* $\mathfrak{A}$*-runs.* Let $\mathfrak{A}$ be a structure. Similar to Section 2.2.3, we define the left quotients and emptiness property for $\mathfrak{A}$-runs. For $\mathfrak{A}$-runs τ, σ, and ρ, we write:

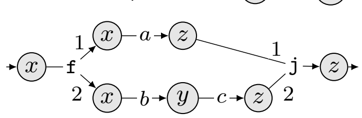
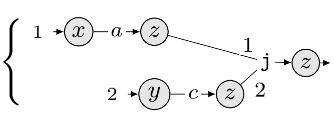
$$\tau \longrightarrow_{\rho}^{\mathfrak{A}} \sigma \quad \overset{\triangle}{\iff} \quad \rho \diamond \sigma \text{ is defined and } \tau \cong \rho \diamond \sigma.$$

For $\mathfrak{A}$-runs τ and ρ, the *left quotient* $\mathsf{D}_{\rho}^{\mathfrak{A}}(\tau)$ of τ with respect to ρ is the $\mathfrak{A}$-run language defined by:

$$\mathsf{D}_{\rho}^{\mathfrak{A}}(\tau) \;\triangleq\; \{\sigma \text{ an } \mathfrak{A}\text{-run} \mid \tau \longrightarrow_{\rho} \sigma\}.$$

For an $\mathfrak{A}$-run language $\mathcal{G}$, let $\mathsf{D}_{\rho}^{\mathfrak{A}}(\mathcal{G}) \triangleq \bigcup_{\tau \in \mathcal{G}} \mathsf{D}_{\rho}^{\mathfrak{A}}(\tau)$.

For instance, if $\rho =$ (), we have:

$$\mathsf{D}_{\rho}^{\mathfrak{A}}\Bigg($$  $$\Bigg) = \Bigg\{$$  $$\Bigg\}.$$

For an $\mathfrak{A}$-run τ and $\vec{z} \in |\mathfrak{A}|^*$, the *emptiness property* $\mathsf{E}_{\vec{z}}(\tau)$ is the truth value defined by:

$$\mathsf{E}_{\vec{z}}(\tau) \;\triangleq\; \begin{cases} \mathsf{true} & (\tau = 1^{\vec{z}}) \\ \mathsf{false} & (\text{otherwise}). \end{cases}$$

For an $\mathfrak{A}$-run language $\mathcal{G}$, let $\mathsf{E}_{\vec{z}}(\mathcal{G}) \triangleq \bigvee_{\tau \in \mathcal{G}} \mathsf{E}_{\vec{z}}(\tau)$. The following is clear by definition.

Proposition 2.21. *Let* $\mathfrak{A}$ *be a structure. For all* $\mathfrak{A}$*-runs* τ*, we have* $\mathsf{E}_{\mathrm{ty}_2(\tau)}(\mathsf{D}_{\tau}^{\mathfrak{A}}(\tau))$.

We will define derivatives based on these left quotient and emptiness property.

3. Labeled Kleene Lattice Terms

To present derivatives (which will be used for an automata construction for PCoR*), we consider simulating left quotients of $\mathfrak{A}$-runs (Section 2.3.2) by terms. To this end, we extend the syntax of terms with "labels". In a nutshell, labels are used for expressing runs with *multiple* source vertices.

Let L be a set of *labels*. We use x, y, z to denote labels. The set of *labeled Kleene lattice terms* (lKL terms) is defined by the following grammar:

$$\tilde{t}, \tilde{s}, \tilde{u} \in \tilde{\mathbf{T}}_L \ ::= \ @x.t \mid \tilde{t} \,;_1\, s \mid \tilde{t} \cap_1 \tilde{s} \qquad (x \in L \text{ and } t, s \text{ are KL terms})$$

Each label x indicates the vertex x on a structure (if exists). The notation $@x.t$ is inspired by the jump operator in hybrid logics [AtC07] (a minor difference is that x directly indicates a vertex name, here). We explicitly use $;_1$ and $\cap_1$ instead of $;$ and $\cap$ in lKL terms, as labeled terms interpret unary relations (*cf.* Section 3.1). We write $\mathsf{lab}_i(\tilde{t})$ for the i-th label occurring in $\tilde{t}$, and we write $\overrightarrow{\mathsf{lab}}(\tilde{t})$ for the sequence $\mathsf{lab}_1(\tilde{t}) \dots \mathsf{lab}_n(\tilde{t})$ where n is the number of labels occurring in $\tilde{t}$. For instance, if $\tilde{t} = ((@x.\tilde{s}_1) \cap_1 (((@x.\tilde{s}_2) \cap_1 (@y.\tilde{s}_3)) \,;_1\, u_1)) \,;_1\, u_2$, then $\mathsf{lab}_1(\tilde{t}) = x$, $\mathsf{lab}_2(\tilde{t}) = x$, $\mathsf{lab}_3(\tilde{t}) = y$, and $\overrightarrow{\mathsf{lab}}(\tilde{t}) = xxy$.

3.1. **Semantics: relational models.**

Let $\mathfrak{A} \in \mathsf{REL}$. For an lKL term $\tilde{t}$, the *semantics* $[\![\tilde{t}]\!]_{\mathfrak{A}} \subseteq |\mathfrak{A}|$ is a unary relation defined as follows:

$$[\![@x.t]\!]_{\mathfrak{A}} \triangleq \{y \mid \langle x, y\rangle \in [\![t]\!]_{\mathfrak{A}}\}, [\![\tilde{t} \,;_1\, s]\!]_{\mathfrak{A}} \triangleq \{z \mid y \in [\![\tilde{t}]\!]_{\mathfrak{A}}, \langle y, z\rangle \in [\![s]\!]_{\mathfrak{A}}\}, [\![\tilde{t} \cap_1 \tilde{s}]\!]_{\mathfrak{A}} \triangleq [\![\tilde{t}]\!]_{\mathfrak{A}} \cap [\![\tilde{s}]\!]_{\mathfrak{A}}.$$

Note that $[\![@x.t]\!]_{\mathfrak{A}} = \emptyset$ when $x \notin |\mathfrak{A}|$. Also, note that $[\![t]\!]_{\mathfrak{A}}$ is a *binary* relation for KL terms t and $[\![\tilde{t}]\!]_{\mathfrak{A}}$ is a *unary* relation for lKL terms $\tilde{t}$. For $\mathcal{C} \subseteq \mathsf{REL}$, we write $\mathcal{C} \models \tilde{t} = \tilde{s}$ if $[\![\tilde{t}]\!]_{\mathfrak{A}} = [\![\tilde{s}]\!]_{\mathfrak{A}}$ for $\mathfrak{A} \in \mathcal{C}$. We then have the following easy fact.

Proposition 3.1. *Let t and s be KL terms and x be a label. We then have:*

$$\mathsf{REL} \models t \leq s \iff \mathsf{REL} \models @x.t \leq @x.s.$$

Proof. ($\Longrightarrow$): Trivial. ($\Longleftarrow$): Let $\langle y, z\rangle \in [\![t]\!]_{\mathfrak{A}}$. Let f and $\mathfrak{B}$ be such that $f\colon \mathfrak{A} \cong \mathfrak{B}$ and $f(y) = x$. We then have that for all KL terms u, $\langle y, z\rangle \in [\![u]\!]_{\mathfrak{A}}$ iff $\langle x, f(z)\rangle \in [\![u]\!]_{\mathfrak{B}}$ (since $\mathfrak{A} \cong \mathfrak{B}$ and u is a (non-labelled) KL term) iff $f(z) \in [\![@x.u]\!]_{\mathfrak{B}}$ (by the definition of $[\![\_]\!]_{\mathfrak{B}}$). We thus have $f(z) \in [\![@x.t]\!]_{\mathfrak{A}}$. By the assumption, $f(z) \in [\![@x.s]\!]_{\mathfrak{B}}$. Hence, by the reverse direction of the above argument, we have $\langle y, z\rangle \in [\![s]\!]_{\mathfrak{A}}$. □

Note that for labeled KL terms $\tilde{t}$, $[\![\tilde{t}]\!]_{\mathfrak{A}}$ and $[\![\tilde{t}]\!]_{\mathfrak{B}}$ may not be equal even if $\mathfrak{A} \cong \mathfrak{B}$, because we use the names of vertices in the semantics (*cf.* nominals in hybrid logic [AtC07]). For that reason, for structures, we distinguish two structures even if they are isomorphic, *cf.* we identify two graphs/runs/$\mathfrak{A}$-runs if they are isomorphic.

3.2. **Runs on structures.**

We use the following notation: $X \heartsuit Y \triangleq \{\tau \heartsuit \sigma \mid \tau \in X \land \sigma \in Y\}$ where X and Y are sets of runs or sets of $\mathfrak{A}$-runs and $\heartsuit \in \{\diamond, \parallel\}$. In this set notation, we use terms as singleton sets: for instance, $\{\tau, \sigma\} \diamond \rho = \{\tau \diamond \rho, \sigma \diamond \rho\}$.

Definition 3.2. Let $\mathfrak{A}$ be a structure. For an lKL term $\tilde{t}$ and $z \in |\mathfrak{A}|$, the $\mathfrak{A}$*-run language* $\mathcal{R}^{\mathfrak{A}}_z(\tilde{t})$ is defined as follows:

$$\mathcal{R}^{\mathfrak{A}}_z(@x.t) \triangleq \mathcal{R}^{\mathfrak{A}}_{x,z}(t),\ \mathcal{R}^{\mathfrak{A}}_z(\tilde{t} \,;_1\, s) \triangleq \bigcup_{y \in |\mathfrak{A}|} \mathcal{R}^{\mathfrak{A}}_y(\tilde{t}) \diamond \mathcal{R}^{\mathfrak{A}}_{y,z}(s),\ \mathcal{R}^{\mathfrak{A}}_z(\tilde{t} \cap_1 \tilde{s}) \triangleq (\mathcal{R}^{\mathfrak{A}}_z(\tilde{t}) \parallel \mathcal{R}^{\mathfrak{A}}_z(\tilde{s})) \diamond \mathtt{j}^z_1.$$

Proposition 3.3 (*cf.* Proposition 2.18)**.** *Let $\mathfrak{A}$ be a structure. For all lKL term $\tilde{t}$, we have:*

$$[\![\tilde{t}]\!]_{\mathfrak{A}} \;=\; \{z \in |\mathfrak{A}| \mid \mathcal{R}^{\mathfrak{A}}_{z}(\tilde{t}) \neq \emptyset\}.$$

Proof. By easy induction on $\tilde{t}$ using Proposition 2.18. □

4. Derivatives on Graphs

We recall the left quotients of $\mathfrak{A}$-runs (defined in Section 2.3.2). In this section, we now define derivatives on $\mathfrak{A}$, which can simulate the left quotients. We first define the emptiness property in the term level (Definition 4.1), and then define the derivatives on graphs (Definition 4.3 and Proposition 4.4), based on derivatives on strings [Brz64, Ant96, Sak09].

Definition 4.1. For a label z and an lKL term $\tilde{t}$, the truth value $\mathsf{E}_z(\tilde{t}) \in \{\mathsf{false}, \mathsf{true}\}$ is defined as follows:[2]

$$\mathsf{E}_z(@x.a) \triangleq \mathsf{E}_z(@x.0) \triangleq \mathsf{E}_z(@x.s \cap u) \triangleq \mathsf{E}_z(\tilde{s} \cap_1 \tilde{u}) \triangleq \mathsf{false},$$

$$\mathsf{E}_z(@x.1) \triangleq \mathsf{E}_z(@x.s^*) \triangleq \begin{cases} \mathsf{true} & (x = z) \\ \mathsf{false} & (x \neq z), \end{cases} \qquad \mathsf{E}_z(\tilde{s} \,;_1 u) \triangleq \mathsf{E}_z(\tilde{s}) \wedge \mathsf{E}_z(@z.u),$$

$$\mathsf{E}_z(@x.s \,;\, u) \triangleq \mathsf{E}_z(@x.s) \wedge \mathsf{E}_z(@z.u), \qquad \mathsf{E}_z(@x.s + u) \triangleq \mathsf{E}_z(@x.s) \vee \mathsf{E}_z(@x.u).$$

For a label z and a set $\tilde{T}$ of lKL terms, let $\mathsf{E}_z(\tilde{T}) \triangleq \bigvee_{\tilde{t} \in \tilde{T}} \mathsf{E}_z(\tilde{t})$. This E_z (Definition 4.1) is defined so that we can encode the emptiness property E_z (in Section 2.2.3) as follows.

Lemma 4.2. *Let $\mathfrak{A}$ be a structure. For all $z \in |\mathfrak{A}|$ and lKL terms $\tilde{t}$, we have:*

$$\mathsf{E}_z(\mathcal{R}^{\mathfrak{A}}_{z}(\tilde{t})) \iff \mathsf{E}_z(\tilde{t}).$$

Proof. By easy induction on $\tilde{t}$, with unfolding the definitions.

- Case $\tilde{t} = @x.a, @x.0, @x.s \cap u, \tilde{s} \cap_1 \tilde{u}$: By $\mathsf{E}_z(\mathcal{R}^{\mathfrak{A}}_{z}(\tilde{t}))$ iff false.
- Case $\tilde{t} = @x.1, @x.s^*$: By $\mathsf{E}_z(\mathcal{R}^{\mathfrak{A}}_{z}(\tilde{t}))$ iff $\begin{cases} \mathsf{true} & (x = z) \\ \mathsf{false} & (x \neq z). \end{cases}$
- Case $\tilde{t} = @x.s \,;\, u$: We have:

$$\begin{aligned}
&\mathsf{E}_z(\mathcal{R}^{\mathfrak{A}}_{z}(@x.s \,;\, u)) \\
\iff\;& \mathsf{E}_z\Big(\bigcup_{y \in |\mathfrak{A}|} \mathcal{R}^{\mathfrak{A}}_{y}(@x.s) \diamond \mathcal{R}^{\mathfrak{A}}_{z}(@y.u)\Big) && \text{(Definition 3.2 of } \mathcal{R}^{\mathfrak{A}}_{z}) \\
\iff\;& \bigvee_{y \in |\mathfrak{A}|} \mathsf{E}_y(\mathcal{R}^{\mathfrak{A}}_{z}(@x.s)) \wedge \mathsf{E}_z(\mathcal{R}^{\mathfrak{A}}_{z}(@y.u)) && \text{(By } \mathsf{E}_z(\tau \diamond \sigma) \text{ iff } \mathsf{E}_z(\tau) \wedge \mathsf{E}_z(\sigma)) \\
\iff\;& \mathsf{E}_z(\mathcal{R}^{\mathfrak{A}}_{z}(@x.s)) \wedge \mathsf{E}_z(\mathcal{R}^{\mathfrak{A}}_{z}(@z.u)) && (\mathsf{E}_y(\mathcal{R}^{\mathfrak{A}}_{z}(@x.s)) \text{ is } \mathsf{false} \text{ when } y \neq z) \\
\iff\;& \mathsf{E}_z(@x.s) \wedge \mathsf{E}_z(@z.u) && \text{(IH)} \\
\iff\;& \mathsf{E}_z(@x.s \,;\, u). && \text{(Definition 4.1)}
\end{aligned}$$

- Case $\tilde{t} = \tilde{s} \,;_1 u$: Similar to the case above.

[2]The definition $\mathsf{E}_z(@x.s \cap u) \triangleq \mathsf{E}_z(\tilde{s} \cap_1 \tilde{u}) \triangleq \mathsf{false}$ is due to that each $\mathfrak{A}$-run of $\mathcal{R}^{\mathfrak{A}}_{z}(@x.s \cap u)$ and $\mathcal{R}^{\mathfrak{A}}_{z}(\tilde{s} \cap_1 \tilde{u})$ always contains $\mathsf{j}^{\tilde{z}}_{1}$ (Definition 3.2), so they do not contain the run having the emptiness property.

- Case $\tilde{t} = @x.s + u$: We have:

$$\begin{aligned}
\mathsf{E}_z(\mathcal{R}^{\mathfrak{A}}_z(@x.s+u)) &\iff \mathsf{E}_z(\mathcal{R}^{\mathfrak{A}}_z(@x.s) \cup \mathcal{R}^{\mathfrak{A}}_z(@z.u)) && \text{(Definition 3.2 of } \mathcal{R}^{\mathfrak{A}}_z\text{)}\\
&\iff \mathsf{E}_z(\mathcal{R}^{\mathfrak{A}}_z(@x.s)) \vee \mathsf{E}_z(\mathcal{R}^{\mathfrak{A}}_z(@z.u)) && \text{(By } \mathsf{E}_z(\mathcal{G}) = \textstyle\bigvee_{\tau \in \mathcal{G}} \mathsf{E}_z(\tau)\text{)}\\
&\iff \mathsf{E}_z(@x.s) \vee \mathsf{E}_z(@z.u) && \text{(IH)}\\
&\iff \mathsf{E}_z(@x.s+u). && \text{(Definition 4.1)}
\end{aligned}$$

Hence, this completes the proof. □

Definition 4.3. Let $\mathfrak{A}$ be a structure. The *derivative* relation $\tilde{t} \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{s}$, where $\tilde{t}$ and $\tilde{s}$ are lKL terms and ρ is an $\mathfrak{A}$-run, is defined as the smallest relation (of tuples of $\tilde{t}$, $\tilde{s}$, and ρ) closed under the following rules:

$$\frac{\langle x, y\rangle \in [\![a]\!]_{\mathfrak{A}}}{@x.a \longrightarrow^{\mathfrak{A}}_{a_1^{x,y}} @y.1} \text{ for } a \in \Sigma \qquad \frac{@x.t \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{t}'}{@x.t\,;s \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{t}'\,;s} \qquad \frac{\mathsf{E}_z(@x.t) \quad @z.s \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{s}'}{@x.t\,;s \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{s}'}$$

$$\frac{@x.t \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{t}'}{@x.t+s \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{t}'} \qquad \frac{@x.s \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{s}'}{@x.t+s \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{s}'} \qquad \frac{\tilde{t} \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{t}'}{\tilde{t}\,;_1 s \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{t}'\,;_1 s} \qquad \frac{\mathsf{E}_z(\tilde{t}) \quad @z.s \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{s}'}{\tilde{t}\,;_1 s \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{s}'}$$

$$\frac{@x.t \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{t}'}{@x.t^* \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{t}'\,;t^*} \qquad \frac{}{@x.t \cap s \longrightarrow^{\mathfrak{A}}_{\mathtt{f}_1^x} (@x.t) \cap_1 (@x.s)} \qquad \frac{\mathsf{E}_z(\tilde{t}) \quad \mathsf{E}_z(\tilde{s})}{\tilde{t} \cap_1 \tilde{s} \longrightarrow^{\mathfrak{A}}_{\mathtt{j}_1^z} @z.1}$$

$$\frac{\tilde{t} \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{t}'}{\tilde{t} \cap_1 \tilde{s} \longrightarrow^{\mathfrak{A}}_{\rho \| 1^{\overrightarrow{\mathsf{lab}}(\tilde{s})}} \tilde{t}' \cap_1 \tilde{s}} \qquad \frac{\tilde{s} \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{s}'}{\tilde{t} \cap_1 \tilde{s} \longrightarrow^{\mathfrak{A}}_{1^{\overrightarrow{\mathsf{lab}}(\tilde{t})} \| \rho} \tilde{t} \cap_1 \tilde{s}'} \qquad \frac{}{\tilde{t} \longrightarrow^{\mathfrak{A}}_{1^{\overrightarrow{\mathsf{lab}}(\tilde{t})}} \tilde{t}} \text{ R} \qquad \frac{\tilde{t} \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{u} \quad \tilde{u} \longrightarrow^{\mathfrak{A}}_{\rho'} \tilde{s}}{\tilde{t} \longrightarrow^{\mathfrak{A}}_{\rho \diamond \rho'} \tilde{s}} \text{ T}$$

Let $\mathfrak{A}$ be a structure and ρ be an $\mathfrak{A}$-run. For an lKL term $\tilde{t}$ (resp. a set $\tilde{T}$ of lKL terms), we define the derivative of $\tilde{t}$ (resp. $\tilde{T}$) with respect to ρ as follows:

$$\mathsf{D}^{\mathfrak{A}}_{\rho}(\tilde{t}) \triangleq \{\tilde{s} \text{ an lKL term} \mid \tilde{t} \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{s}\}, \qquad \mathsf{D}^{\mathfrak{A}}_{\rho}(\tilde{T}) \triangleq \bigcup_{\tilde{t} \in \tilde{T}} \mathsf{D}^{\mathfrak{A}}_{\rho}(\tilde{t}).$$

For notational simplicity, we use the following notation: $A \heartsuit B \triangleq \{a \heartsuit b \mid a \in A \wedge b \in B\}$ where A and B are sets of lKL terms and $\heartsuit \in \{;_1, \cap_1\}$. In this notation, we use terms as singleton sets: for instance, $\{a, b\} \cap_1 c = \{a \cap_1 c, b \cap_1 c\}$. Using this notation, for *atomic $\mathfrak{A}$-runs*, Definition 4.3 can be alternatively expressed as follows (*cf.* Antimirov's derivatives [Ant96]).

Proposition 4.4. *Let $\mathfrak{A}$ be a structure. For an atomic $\mathfrak{A}$-run ρ, we have the following:*

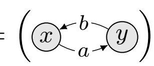
$$\begin{aligned}
\mathsf{D}^{\mathfrak{A}}_{\rho}(@x.a) &= \begin{cases} \{@y.1 \mid \langle x, y\rangle \in [\![a]\!]_{\mathfrak{A}}\} & (\rho = a_1^{x,y}) \\ \emptyset & (\textit{otherwise}) \end{cases} \quad \textit{for } a \in \Sigma,\\
\mathsf{D}^{\mathfrak{A}}_{\rho}(@x.1) &= \mathsf{D}^{\mathfrak{A}}_{\rho}(@x.0) = \emptyset,\\
\mathsf{D}^{\mathfrak{A}}_{\rho}(@x.s\,;u) &= (\mathsf{D}^{\mathfrak{A}}_{\rho}(@x.s)\,;_1 u) \cup \bigcup_{z \in |\mathfrak{A}|} \mathsf{D}^{\mathfrak{A}}_{\rho}(\{@z.u \mid \mathsf{E}_z(@x.s)\}),\\
\mathsf{D}^{\mathfrak{A}}_{\rho}(@x.s+u) &= \mathsf{D}^{\mathfrak{A}}_{\rho}(@x.s) \cup \mathsf{D}^{\mathfrak{A}}_{\rho}(@x.u),\\
\mathsf{D}^{\mathfrak{A}}_{\rho}(@x.s^*) &= \mathsf{D}^{\mathfrak{A}}_{\rho}(@x.s)\,;_1 s^*,
\end{aligned}$$

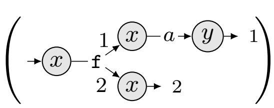
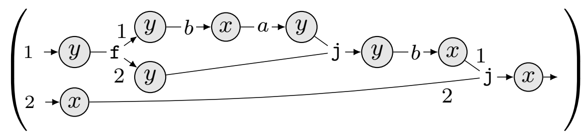
$$\mathsf{D}^{\mathfrak{A}}_{\rho}(@x.s \cap u) \;=\; \begin{cases} \{(@x.s) \cap_1 (@x.u)\} & (\rho = \mathtt{f}^{x}_{1}) \\ \emptyset & (\textit{otherwise}), \end{cases}$$
$$\mathsf{D}^{\mathfrak{A}}_{\rho}(\tilde{s} \,;_1\, u) \;=\; (\mathsf{D}^{\mathfrak{A}}_{\rho}(\tilde{s}) \,;_1\, u) \cup \bigcup_{z \in |\mathfrak{A}|} \mathsf{D}^{\mathfrak{A}}_{\rho}(\{@z.u \mid \mathsf{E}_z(\tilde{s})\}),$$
$$\mathsf{D}^{\mathfrak{A}}_{\rho}(\tilde{s} \cap_1 \tilde{u}) \;=\; \begin{cases} \mathsf{D}^{\mathfrak{A}}_{\rho'}(\tilde{s}) \cap_1 \tilde{u} & (\rho = (\rho' \parallel 1^{\overrightarrow{\mathsf{lab}}(\tilde{u})}) \textit{ for some } \rho') \\ \tilde{s} \cap_1 \mathsf{D}^{\mathfrak{A}}_{\rho'}(\tilde{u}) & (\rho = (1^{\overrightarrow{\mathsf{lab}}(\tilde{s})} \parallel \rho') \textit{ for some } \rho') \\ \{@z.1 \mid \mathsf{E}_z(\tilde{s}) \wedge \mathsf{E}_z(\tilde{u})\} & (\rho = \mathtt{j}^{z}_{1}) \\ \emptyset & (\textit{otherwise}). \end{cases}$$

Proof. By a routine verification. □

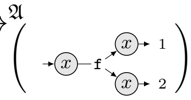
The derivatives can simulate left quotients of $\mathfrak{A}$-runs in that $\mathsf{D}^{\mathfrak{A}}_{\rho}(\tilde{t})$ expresses the left quotient of $\mathcal{R}^{\mathfrak{A}}_{z}(\tilde{t})$ with respect to ρ, as follows. See Section 4.1, for a detailed proof.

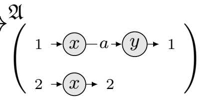
Lemma 4.5. *Let $\mathfrak{A}$ be a structure and $z \in |\mathfrak{A}|$. For all lKL terms $\tilde{t}$ and $\mathfrak{A}$-runs ρ, we have:*
$$\mathsf{D}^{\mathfrak{A}}_{\rho}(\mathcal{R}^{\mathfrak{A}}_{z}(\tilde{t})) \;=\; \mathcal{R}^{\mathfrak{A}}_{z}(\mathsf{D}^{\mathfrak{A}}_{\rho}(\tilde{t})).$$

Example 4.6 (*cf.* Section 2.3.2)**.** Let $\mathfrak{A} =$ , $\tilde{t} = @x.(a((ba) \cap 1)b) \cap 1$, $\rho =$

, and $\rho' =$ .

We can see $\rho' \in \mathsf{D}_{\rho}(\mathcal{R}^{\mathfrak{A}}_{x}(\tilde{t}))$ by $\rho \diamond \rho' \in \mathcal{R}^{\mathfrak{A}}_{x}(\tilde{t})$. We also have $\rho' \in \mathcal{R}^{\mathfrak{A}}_{x}(\mathsf{D}_{\rho}(\tilde{t}))$, because by letting $\tilde{t}' = (@y.((ba) \cap 1)b) \cap_1 @x.1$, we have $\rho' \in \mathcal{R}^{\mathfrak{A}}(\tilde{t}')$ and $\tilde{t}' \in \mathsf{D}_{\rho}(\tilde{t})$. Here, $\tilde{t}' \in \mathsf{D}_{\rho}(\tilde{t})$ is shown as follows:

$\tilde{t} \;=\; @x.(a((ba) \cap 1)b) \cap 1 \quad \longrightarrow^{\mathfrak{A}}$  $\quad (@x.a((ba) \cap 1)b) \cap_1 @x.1$

$\longrightarrow^{\mathfrak{A}}$  $\quad (@y.((ba) \cap 1)b) \cap_1 @x.1 \;=\; \tilde{t}'.$

We let $(\longrightarrow^{\mathfrak{A}}) \triangleq \bigcup_{\rho \text{ an } \mathfrak{A}\text{-run}} (\longrightarrow^{\mathfrak{A}}_{\rho})$. For an lKL term $\tilde{t}$ and a set $\tilde{T}$ of lKL terms, we let:
$$\mathsf{D}^{\mathfrak{A}}(\tilde{t}) \;\triangleq\; \{\tilde{s} \text{ an lKL term} \mid \tilde{t} \longrightarrow^{\mathfrak{A}} \tilde{s}\}, \qquad \mathsf{D}^{\mathfrak{A}}(\tilde{T}) \;\triangleq\; \bigcup_{\tilde{t} \in \tilde{T}} \mathsf{D}^{\mathfrak{A}}(\tilde{t}).$$

By Lemmas 4.2 and 4.5, derivatives can give an alternative semantics, as follows.

Theorem 4.7. *Let $\mathfrak{A}$ be a structure and let $\tilde{t}$ be an lKL term. We have:*
$$[\![\tilde{t}]\!]_{\mathfrak{A}} \;=\; \{z \in |\mathfrak{A}| \mid \mathsf{E}_z(\mathsf{D}^{\mathfrak{A}}(\tilde{t}))\}.$$

Proof. We have:
$$\begin{aligned} z \in [\![\tilde{t}]\!]_{\mathfrak{A}} &\iff \mathcal{R}^{\mathfrak{A}}_{z}(\tilde{t}) \neq \emptyset && \text{(Proposition 3.3)} \\ &\iff \mathsf{E}_z(\mathsf{D}^{\mathfrak{A}}_{\rho}(\mathcal{R}^{\mathfrak{A}}_{z}(\tilde{t}))) \text{ for some } \mathfrak{A}\text{-run } \rho && \text{(Proposition 2.21)} \\ &\iff \mathsf{E}_z(\mathsf{D}^{\mathfrak{A}}_{\rho}(\tilde{t})) \text{ for some } \mathfrak{A}\text{-run } \rho && \text{(Lemmas 4.2 and 4.5)} \\ &\iff \mathsf{E}_z(\mathsf{D}^{\mathfrak{A}}(\tilde{t})). && \text{(By } \mathsf{D}^{\mathfrak{A}}(\tilde{t}) = \textstyle\bigcup_{\rho \text{ an } \mathfrak{A}\text{-run}} \mathsf{D}^{\mathfrak{A}}_{\rho}(\tilde{t})\text{)} \end{aligned}$$

Hence this completes the proof. □

4.1. **Proof of Lemma 4.5: Derivatives can simulate left quotients.** We first prove the following proposition. For $\vec{x}$ and $\vec{y}$, we use $X_{\vec{x},\vec{y}}, Y_{\vec{x},\vec{y}}, \dots$ to denote sets of $\mathfrak{A}$-runs with $\langle \vec{x}, \vec{y} \rangle$-interface.

Proposition 4.8. *Let ρ be an atomic $\mathfrak{A}$-run. We then have the following.*

($\diamond$): $\mathsf{D}^{\mathfrak{A}}_{\rho}(X_{\vec{x},y} \diamond Y_{y,z}) = \begin{cases} (\mathsf{D}^{\mathfrak{A}}_{\rho}(X_{\vec{x},y}) \diamond Y_{y,z}) \cup \mathsf{D}^{\mathfrak{A}}_{\rho}(Y_{y,z}) & (\mathsf{E}_y(X_{\vec{x},y})) \\ (\mathsf{D}^{\mathfrak{A}}_{\rho}(X_{\vec{x},y}) \diamond Y_{y,z}) & (\textit{otherwise}); \end{cases}$

($\parallel$): $\mathsf{D}^{\mathfrak{A}}_{\rho}(X_{\vec{x}_1,z_1} \parallel Y_{\vec{x}_2,z_2}) = \begin{cases} \mathsf{D}^{\mathfrak{A}}_{\rho'}(X_{\vec{x}_1,z_1}) \parallel Y_{\vec{x}_2,z_2} & (\rho = (\rho' \parallel 1^{\overrightarrow{\mathsf{lab}}(\vec{x}_2)}) \textit{ for some } \rho') \\ X_{\vec{x}_1,z_1} \parallel \mathsf{D}^{\mathfrak{A}}_{\rho'}(Y_{\vec{x}_2,z_2}) & (\rho = (1^{\overrightarrow{\mathsf{lab}}(\vec{x}_1)} \parallel \rho') \textit{ for some } \rho') \\ \emptyset & (\textit{otherwise}). \end{cases}$

Proof. For ($\diamond$): Because for all τ, σ, ρ' such that $\mathrm{ty}_2(\tau) = \mathrm{ty}_1(\sigma) = y$, we have

$$\tau \diamond \sigma \longrightarrow^{\mathfrak{A}}_{\rho} \rho' \iff \bigvee \begin{cases} \exists \tau', \tau \longrightarrow^{\mathfrak{A}}_{\rho} \tau' \wedge \rho' = \tau' \diamond \sigma \\ \mathsf{E}_y(\tau) \wedge \sigma \longrightarrow^{\mathfrak{A}}_{\rho} \rho'. \end{cases}$$

For ($\parallel$): Because for all τ, σ, ρ', we have

$$\tau \parallel \sigma \longrightarrow^{\mathfrak{A}}_{\rho} \rho' \iff \bigvee \begin{cases} \exists \tau', \rho',\ \tau \longrightarrow^{\mathfrak{A}}_{\rho''} \tau' \wedge \rho' = (\tau' \parallel \sigma) \wedge \rho = (\rho'' \parallel 1^{\mathrm{ty}_1(\sigma)}) \\ \exists \sigma', \rho',\ \sigma \longrightarrow^{\mathfrak{A}}_{\rho''} \sigma' \wedge \rho' = (\tau \parallel \sigma') \wedge \rho = (1^{\mathrm{ty}_1(\tau)} \parallel \rho''). \end{cases}$$

Hence, this completes the proof. □

We first show Lemma 4.5 for atomic $\mathfrak{A}$-runs, and then we extend for $\mathfrak{A}$-runs.

Lemma 4.9. *Let $\mathfrak{A}$ be a structure and $z \in |\mathfrak{A}|$. For all lKL terms $\tilde{t}$ and atomic $\mathfrak{A}$-runs ρ,*

$$\mathsf{D}^{\mathfrak{A}}_{\rho}(\mathcal{R}^{\mathfrak{A}}_{z}(\tilde{t})) \ = \ \mathcal{R}^{\mathfrak{A}}_{z}(\mathsf{D}^{\mathfrak{A}}_{\rho}(\tilde{t})).$$

Proof. By easy induction on $\tilde{t}$ using Proposition 4.8.

- Case $\tilde{t} = @x.a$ for $a \in \Sigma$: We have:

$$\begin{aligned} \mathsf{D}^{\mathfrak{A}}_{\rho}(\mathcal{R}^{\mathfrak{A}}_{z}(@x.a)) \ &= \ \mathsf{D}^{\mathfrak{A}}_{\rho}(\{a_1^{x,z} \mid \langle x, z \rangle \in [\![a]\!]_{\mathfrak{A}}\}) && (\text{Definition 3.2 of } \mathcal{R}^{\mathfrak{A}}_{z}) \\ &= \begin{cases} \mathcal{R}^{\mathfrak{A}}_{z}(\{@z.1 \mid \langle x, z \rangle \in [\![a]\!]_{\mathfrak{A}}\}) & (\rho = a_1^{x,z}) \\ \emptyset & (\text{otherwise}) \end{cases} \ = \ \mathcal{R}^{\mathfrak{A}}_{z}(\mathsf{D}^{\mathfrak{A}}_{\rho}(@x.a)). \end{aligned}$$

- Case $\tilde{t} = @x.1, @x.0$: We have: $\mathsf{D}^{\mathfrak{A}}_{\rho}(\mathcal{R}^{\mathfrak{A}}_{z}(\tilde{t})) = \emptyset = \mathcal{R}^{\mathfrak{A}}_{z}(\mathsf{D}^{\mathfrak{A}}_{\rho}(\tilde{t}))$.
- Case $\tilde{t} = @x.s + u$: We have:

$$\begin{aligned} \mathsf{D}^{\mathfrak{A}}_{\rho}(\mathcal{R}^{\mathfrak{A}}_{z}(@x.s + u)) \ &= \ \mathsf{D}^{\mathfrak{A}}_{\rho}(\mathcal{R}^{\mathfrak{A}}_{z}(@x.s)) \cup \mathsf{D}^{\mathfrak{A}}_{\rho}(\mathcal{R}^{\mathfrak{A}}_{z}(@x.u)) && (\text{Definition 3.2 of } \mathcal{R}^{\mathfrak{A}}_{z}) \\ &= \ \mathcal{R}^{\mathfrak{A}}_{z}(\mathsf{D}^{\mathfrak{A}}_{\rho}(@x.s)) \cup \mathcal{R}^{\mathfrak{A}}_{z}(\mathsf{D}^{\mathfrak{A}}_{\rho}(@x.u)) = \mathcal{R}^{\mathfrak{A}}_{z}(\mathsf{D}^{\mathfrak{A}}_{\rho}(@x.s + u)). && (\text{IH}) \end{aligned}$$

- Case $\tilde{t} = @x.s\,;u$: We have:

$$\mathsf{D}^{\mathfrak{A}}_{\rho}(\mathcal{R}^{\mathfrak{A}}_{z}(@x.s\,;u)) = \bigcup_{y\in|\mathfrak{A}|} \mathsf{D}^{\mathfrak{A}}_{\rho}(\mathcal{R}^{\mathfrak{A}}_{y}(@x.s) \diamond \mathcal{R}^{\mathfrak{A}}_{y,z}(u)) \qquad \text{(Definition 3.2 of } \mathcal{R}^{\mathfrak{A}}_{z}\text{)}$$

$$= (\bigcup_{y\in|\mathfrak{A}|} \mathsf{D}^{\mathfrak{A}}_{\rho}(\mathcal{R}^{\mathfrak{A}}_{y}(@x.s)) \diamond \mathcal{R}^{\mathfrak{A}}_{y,z}(u)) \cup \bigcup_{y\in|\mathfrak{A}|;\mathsf{E}_y(\mathcal{R}^{\mathfrak{A}}_{y}(@x.s))} \mathsf{D}^{\mathfrak{A}}_{\rho}(\mathcal{R}^{\mathfrak{A}}_{z}(@y.u)) \quad \text{(Proposition 4.8 } (\diamond)\text{)}$$

$$= (\bigcup_{y\in|\mathfrak{A}|} \mathcal{R}^{\mathfrak{A}}_{y}(\mathsf{D}^{\mathfrak{A}}_{\rho}(@x.s)) \diamond \mathcal{R}^{\mathfrak{A}}_{y,z}(u)) \cup \bigcup_{y\in|\mathfrak{A}|;\mathsf{E}_y(@x.s)} \mathcal{R}^{\mathfrak{A}}_{z}(\mathsf{D}^{\mathfrak{A}}_{\rho}(@y.u)) \qquad \text{(Lemma 4.2, IH)}$$

$$= \mathcal{R}^{\mathfrak{A}}_{z}\left((\mathsf{D}^{\mathfrak{A}}_{\rho}(@x.s)\,;_1 u) \cup \bigcup_{y\in|\mathfrak{A}|;\mathsf{E}_y(@x.s)} \mathsf{D}^{\mathfrak{A}}_{\rho}(@y.u)\right) = \mathcal{R}^{\mathfrak{A}}_{z}(\mathsf{D}^{\mathfrak{A}}_{\rho}(@x.s\,;u)). \quad \text{(Definition 3.2)}$$

- Case $\tilde{t} = @x.s^*$: We have:

$$\begin{aligned}
\mathsf{D}^{\mathfrak{A}}_{\rho}(\mathcal{R}^{\mathfrak{A}}_{z}(@x.s^*)) &= \mathsf{D}^{\mathfrak{A}}_{\rho}(\bigcup_{y\in|\mathfrak{A}|} \{\tau \in \mathcal{R}^{\mathfrak{A}}_{y}(@x.s) \mid \neg\,\mathsf{E}_y(\tau)\} \diamond \mathcal{R}^{\mathfrak{A}}_{y,z}(s^*)) && \text{(As } \rho \text{ is not empty)}\\
&= \bigcup_{y\in|\mathfrak{A}|} \mathsf{D}^{\mathfrak{A}}_{\rho}(\{\tau \in \mathcal{R}^{\mathfrak{A}}_{y}(@x.s) \mid \neg\,\mathsf{E}_y(\tau)\}) \diamond \mathcal{R}^{\mathfrak{A}}_{y,z}(s^*) && \text{(Proposition 4.8 } (\diamond)\text{)}\\
&= \bigcup_{y\in|\mathfrak{A}|} \mathsf{D}^{\mathfrak{A}}_{\rho}(\mathcal{R}^{\mathfrak{A}}_{y}(@x.s)) \diamond \mathcal{R}^{\mathfrak{A}}_{y,z}(s^*) && \text{(As } \rho \text{ is not empty)}\\
&= \bigcup_{y\in|\mathfrak{A}|} \mathcal{R}^{\mathfrak{A}}_{y}(\mathsf{D}^{\mathfrak{A}}_{\rho}(@x.s)) \diamond \mathcal{R}^{\mathfrak{A}}_{y,z}(s^*) && \text{(IH)}\\
&= \mathcal{R}^{\mathfrak{A}}_{z}(\mathsf{D}^{\mathfrak{A}}_{\rho}(@x.s)\,;_1 s^*) = \mathcal{R}^{\mathfrak{A}}_{z}(\mathsf{D}^{\mathfrak{A}}_{\rho}(@x.s^*)). && \text{(Definition 3.2)}
\end{aligned}$$

- Case $\tilde{t} = @x.s \cap u$: We have:

$$\begin{aligned}
\mathsf{D}^{\mathfrak{A}}_{\rho}(\mathcal{R}^{\mathfrak{A}}_{z}(@x.s \cap u)) &= \mathsf{D}^{\mathfrak{A}}_{\rho}(\mathtt{f}^{x}_{1} \diamond (\mathcal{R}^{\mathfrak{A}}_{z}(@x.s) \parallel \mathcal{R}^{\mathfrak{A}}_{z}(@x.u)) \diamond \mathtt{j}^{z}_{1}) && \text{(Definition 3.2 of } \mathcal{R}^{\mathfrak{A}}_{z}\text{)}\\
&= \begin{cases} (\mathcal{R}^{\mathfrak{A}}_{z}(@x.s) \parallel \mathcal{R}^{\mathfrak{A}}_{z}(@x.u)) \diamond \mathtt{j}^{z}_{1} & (\rho = \mathtt{f}^{x}_{1})\\ \emptyset & (\text{otherwise}) \end{cases} = \mathcal{R}^{\mathfrak{A}}_{z}(\mathsf{D}^{\mathfrak{A}}_{\rho}(@x.s \cap u)).
\end{aligned}$$

- Case $\tilde{t} = \tilde{s}\,;_1 u$: Similar to Case $\tilde{t} = @x.s\,;u$.
- Case $\tilde{t} = \tilde{s} \cap_1 \tilde{u}$: We have:

$$\mathsf{D}^{\mathfrak{A}}_{\rho}(\mathcal{R}^{\mathfrak{A}}_{z}(\tilde{s} \cap_1 \tilde{u})) = \mathsf{D}^{\mathfrak{A}}_{\rho}((\mathcal{R}^{\mathfrak{A}}_{z}(\tilde{s}) \parallel \mathcal{R}^{\mathfrak{A}}_{z}(\tilde{u})) \diamond \mathtt{j}^{z}_{1}) \qquad \text{(Definition 3.2 of } \mathcal{R}^{\mathfrak{A}}_{z}\text{)}$$

$$= \begin{cases} (\mathcal{R}^{\mathfrak{A}}_{z}(\mathsf{D}^{\mathfrak{A}}_{\rho'}(\tilde{s})) \parallel \mathcal{R}^{\mathfrak{A}}_{z}(\tilde{u})) \diamond \mathtt{j}^{z}_{1} & (\rho = (\rho' \parallel 1^{\overrightarrow{\mathsf{lab}}(\tilde{u})}) \text{ for some } \rho')\\ (\mathcal{R}^{\mathfrak{A}}_{z}(\tilde{s}) \parallel \mathcal{R}^{\mathfrak{A}}_{z}(\mathsf{D}^{\mathfrak{A}}_{\rho'}(\tilde{u}))) \diamond \mathtt{j}^{z}_{1} & (\rho = (1^{\overrightarrow{\mathsf{lab}}(\tilde{s})} \parallel \rho') \text{ for some } \rho')\\ \{1^{z} \mid \mathsf{E}_z(\tilde{s}) \wedge \mathsf{E}_z(\tilde{u})\} & (\rho = \mathtt{j}^{z}_{1})\\ \emptyset & (\text{otherwise}) \end{cases} \qquad \text{(Proposition 4.8, IH)}$$

$$= \begin{cases} \mathcal{R}^{\mathfrak{A}}_{z}(\mathsf{D}^{\mathfrak{A}}_{\rho'}(\tilde{s}) \cap_1 \tilde{u}) & (\rho = (\rho' \parallel 1^{\overrightarrow{\mathsf{lab}}(\tilde{u})}) \text{ for some } \rho')\\ \mathcal{R}^{\mathfrak{A}}_{z}(\tilde{s} \cap_1 \mathsf{D}^{\mathfrak{A}}_{\rho'}(\tilde{u})) & (\rho = (1^{\overrightarrow{\mathsf{lab}}(\tilde{s})} \parallel \rho') \text{ for some } \rho')\\ \mathcal{R}^{\mathfrak{A}}_{z}(\{@z.1 \mid \mathsf{E}_z(\tilde{s}) \wedge \mathsf{E}_z(\tilde{u})\}) & (\rho = \mathtt{j}^{z}_{1})\\ \emptyset & (\text{otherwise}) \end{cases}$$

$$= \mathcal{R}^{\mathfrak{A}}_{z}(\mathsf{D}^{\mathfrak{A}}_{\rho}(\tilde{s} \cap_1 \tilde{u})).$$

Hence this completes the proof. □

Proof of Lemma 4.5. By easy induction on the number k of edges occurring in ρ using Lemma 4.9. We distinguish the following cases:

- Case $k = 0$: We have:

$$\mathsf{D}^{\mathfrak{A}}_{\rho}(\mathcal{R}^{\mathfrak{A}}_{z}(\tilde{t})) = \begin{cases} \mathcal{R}^{\mathfrak{A}}_{z}(\tilde{t}) & (\rho = 1^{\overrightarrow{\mathsf{lab}}(\tilde{t})}) \\ \emptyset & (\text{otherwise}) \end{cases} = \mathcal{R}^{\mathfrak{A}}_{z}(\mathsf{D}^{\mathfrak{A}}_{\rho}(\tilde{t})). \qquad \text{(By the rule (R))}$$

- Case $k \geq 1$: By Proposition 2.20, let $a_1, \dots, a_k$ be atomic $\mathfrak{A}$-runs such that $\rho = a_1 \diamond \dots \diamond a_k$. We have:

$$\begin{aligned} \mathsf{D}^{\mathfrak{A}}_{a_1 \diamond \dots \diamond a_k}(\mathcal{R}^{\mathfrak{A}}_{z}(\tilde{t})) &= \mathsf{D}^{\mathfrak{A}}_{a_k}(\mathsf{D}^{\mathfrak{A}}_{a_{k-1}}(\dots(\mathsf{D}^{\mathfrak{A}}_{a_1}(\mathcal{R}^{\mathfrak{A}}_{z}(\tilde{t}))))) && \text{(By the rule (T))} \\ &= \mathcal{R}^{\mathfrak{A}}_{z}(\mathsf{D}^{\mathfrak{A}}_{a_k}(\mathsf{D}^{\mathfrak{A}}_{a_{k-1}}(\dots(\mathsf{D}^{\mathfrak{A}}_{a_1}(\tilde{t}))))) && \text{(By Lemma 4.9, iteratively)} \\ &= \mathcal{R}^{\mathfrak{A}}_{z}(\mathsf{D}^{\mathfrak{A}}_{a_1 \diamond \dots \diamond a_k}(\tilde{t})). && \text{(By the rule (T))} \end{aligned}$$

Hence this completes the proof. □

4.2. **Closure property of derivatives.**

Moreover, the derivatives above have a closure property like Antimirov's derivatives [Ant96, BBM+16].

Definition 4.10. Let L be a set (of labels). For a KL term t, the *closure* $\mathrm{cl}_L(t)$ is the set of lKL terms defined by:

$$\begin{aligned} \mathrm{cl}_L(a) &\triangleq \{@y.a, @y.1 \mid y \in L\} \text{ for } a \in \Sigma, \\ \mathrm{cl}_L(\mathsf{1}) &\triangleq \{@y.\mathsf{1} \mid y \in L\}, \\ \mathrm{cl}_L(\mathsf{0}) &\triangleq \{\mathsf{0}\}, \\ \mathrm{cl}_L(s \mathbin{;} u) &\triangleq \{@y.s \mathbin{;} u \mid y \in L\} \cup (\mathrm{cl}_L(s) \mathbin{;_1} u) \cup \mathrm{cl}_L(u), \\ \mathrm{cl}_L(s + u) &\triangleq \{@y.s + u \mid y \in L\} \cup \mathrm{cl}_L(s) \cup \mathrm{cl}_L(u), \\ \mathrm{cl}_L(s^*) &\triangleq \{@y.s^* \mid y \in L\} \cup (\mathrm{cl}_L(s) \mathbin{;_1} s^*), \\ \mathrm{cl}_L(s \cap u) &\triangleq \{@y.s \cap u, @y.\mathsf{1} \mid y \in L\} \cup (\mathrm{cl}_L(s) \cap_1 \mathrm{cl}_L(u)). \end{aligned}$$

For an lKL term $\tilde{t} \in \tilde{\mathbf{T}}_L$, the closure $\mathrm{cl}_L(\tilde{t})$ is the set of lKL terms defined by:

$$\begin{aligned} \mathrm{cl}_L(@x.t) &\triangleq \mathrm{cl}_L(t), \\ \mathrm{cl}_L(\tilde{s} \mathbin{;_1} u) &\triangleq (\mathrm{cl}_L(\tilde{s}) \mathbin{;_1} u) \cup \mathrm{cl}_L(u), \\ \mathrm{cl}_L(\tilde{s} \cap_1 \tilde{u}) &\triangleq \{@y.\mathsf{1} \mid y \in L\} \cup (\mathrm{cl}_L(\tilde{s}) \cap_1 \mathrm{cl}_L(\tilde{u})). \end{aligned}$$

We then extend cl_L for sets of lKL terms, by $\mathrm{cl}_L(\tilde{T}) \triangleq \bigcup_{\tilde{t} \in \tilde{T}} \mathrm{cl}_L(\tilde{t})$.

The function cl_L is a closure operator, as follows.

Proposition 4.11. *For each set L, the function $\mathrm{cl}_L \colon \wp(\tilde{\mathbf{T}}_L) \to \wp(\tilde{\mathbf{T}}_L)$ is a closure operator.*

Proof. Because we have the following, respectively:

- Extensivity $\tilde{T} \subseteq \mathrm{cl}_L(\tilde{T})$: Because $\tilde{t} \in \mathrm{cl}_L(\tilde{t})$ (by definition).
- Monotonicity $\tilde{T} \subseteq \tilde{S} \Longrightarrow \mathrm{cl}_L(\tilde{T}) \subseteq \mathrm{cl}_L(\tilde{S})$: By $\mathrm{cl}_L(\tilde{T}) = \bigcup_{\tilde{t} \in \tilde{T}} \mathrm{cl}_L(\tilde{t}) \subseteq \bigcup_{\tilde{t} \in \tilde{S}} \mathrm{cl}_L(\tilde{t}) = \mathrm{cl}_L(\tilde{S})$.

- Idempotency $\mathrm{cl}_L(\mathrm{cl}_L(\tilde{T})) \subseteq \mathrm{cl}_L(\tilde{T})$: It suffices to prove that $\mathrm{cl}_L(\mathrm{cl}_L(t)) \subseteq \mathrm{cl}_L(t)$ and $\mathrm{cl}_L(\mathrm{cl}_L(\tilde{t})) \subseteq \mathrm{cl}_L(\tilde{t})$. This is shown by induction on the size of t (resp. $\tilde{t}$). □

Additionally, the closure operator cl_L is *algebraic*, i.e., $\mathrm{cl}_L(\tilde{T}) = \bigcup_{\tilde{S} \subseteq \tilde{T}; \tilde{S} \text{ is finite}} \mathrm{cl}_L(\tilde{S})$ holds, which is clear by $\mathrm{cl}_L(\tilde{S}) = \bigcup_{\tilde{t} \in \tilde{S}} \mathrm{cl}_L(\tilde{t})$ for each $\tilde{S}$.

We then have that the lKL terms obtained by derivatives of an lKL term $\tilde{t}$ are always in the closure of $\tilde{t}$, as follows.

Proposition 4.12. *Let $\mathfrak{A}$ be a structure and let ρ be a $\mathfrak{A}$-run. For all lKL terms $\tilde{t} \in \tilde{\mathbf{T}}_{|\mathfrak{A}|}$, we have $\mathsf{D}^{\mathfrak{A}}_{\rho}(\tilde{t}) \subseteq \mathrm{cl}_{|\mathfrak{A}|}(\tilde{t})$.*

Proof. When ρ is an atomic $\mathfrak{A}$-run, this is shown by easy induction on the size of $\tilde{t}$ (by comparing Proposition 4.4 and Definition 4.10). By extensivity and idempotency (Proposition 4.11), we can extend this property from atomic $\mathfrak{A}$-runs to $\mathfrak{A}$-runs. □

Moreover, on the closure size, we have the following result.

Proposition 4.13. *Let L be a set. For all KL terms t, $\#\,\mathrm{cl}_L(t) \leq (2\#L\|t\|)^{\mathrm{iw}(t)}$.*

Proof. By easy induction on t.

- Case $t = a$ where $a \in \Sigma \cup \{1, 0\}$: By $\#\,\mathrm{cl}_L(t) \leq 2\#L \leq (2\#L\|t\|)^1$.
- Case $t = s \heartsuit u$ where $\heartsuit \in \{;, +\}$: We have:

$$\begin{aligned}
\#\,\mathrm{cl}_L(s \heartsuit u) &\leq \#L + \#\,\mathrm{cl}_L(s) + \#\,\mathrm{cl}_L(u) && (\text{Definition 4.10 of } \mathrm{cl}_L)\\
&\leq \#L + (2\#L\|s\|)^{\mathrm{iw}(s)} + (2\#L\|u\|)^{\mathrm{iw}(u)} && (\text{IH})\\
&\leq (2\#L)^{\max(\mathrm{iw}(s),\mathrm{iw}(u))}(1 + \|s\|^{\max(\mathrm{iw}(s),\mathrm{iw}(u))} + \|u\|^{\max(\mathrm{iw}(s),\mathrm{iw}(u))})\\
&\leq (2\#L\|s \heartsuit u\|)^{\mathrm{iw}(s \heartsuit u)}.
\end{aligned}$$

- Case $t = s^*$: We have:

$$\begin{aligned}
\#\,\mathrm{cl}_L(s^*) &\leq \#L + \#\,\mathrm{cl}_L(s) && (\text{Definition 4.10 of } \mathrm{cl}_L)\\
&\leq \#L + (2\#L\|s\|)^{\mathrm{iw}(s)} && (\text{IH})\\
&\leq (2\#L)^{\mathrm{iw}(s)}(1 + \|s\|^{\mathrm{iw}(s)}) \leq (2\#L\|s^*\|)^{\mathrm{iw}(s^*)}.
\end{aligned}$$

- Case $t = s \cap u$: We have:

$$\begin{aligned}
\#\,\mathrm{cl}_L(s \cap u) &\leq \#L + \#\,\mathrm{cl}_L(s)\#\,\mathrm{cl}_L(u) + \#L && (\text{Definition 4.10 of } \mathrm{cl}_L)\\
&\leq 2\#L + (2\#L\|s\|)^{\mathrm{iw}(s)}(2\#L\|u\|)^{\mathrm{iw}(u)} && (\text{IH})\\
&\leq (2\#L)^{\mathrm{iw}(s)+\mathrm{iw}(u)}(1 + \|s\|^{\mathrm{iw}(s)}\|u\|^{\mathrm{iw}(u)})\\
&\leq (2\#L)^{\mathrm{iw}(s)+\mathrm{iw}(u)}(1 + \|s\| + \|u\|)^{\mathrm{iw}(s)+\mathrm{iw}(u)} \leq (2\#L\|s \cap u\|)^{\mathrm{iw}(s \cap u)}.
\end{aligned}$$

Hence, this completes the proof. □

5. Decomposing Derivatives: An Automata Construction

In this section, we show that the derivatives are decomposable on path decompositions (Theorem 5.5 in Section 5.2), and we give an automata construction (Section 5).

We let $\mathsf{REL}_k \triangleq \{\mathfrak{A} \in \mathsf{REL} \mid |\mathfrak{A}| \subseteq [k]\}$. For $\mathcal{C} \subseteq \mathsf{REL}$, we write $\mathsf{I}(\mathcal{C})$ for the *isomorphic closure* of $\mathcal{C}$, i.e., $\mathsf{I}(\mathcal{C}) \triangleq \{\mathfrak{B} \in \mathsf{REL} \mid \mathfrak{A} \in \mathcal{C} \text{ and } \mathfrak{A} \cong \mathfrak{B}\}$.

5.1. Gluing operator for path decompositions.

For $\mathfrak{A}_1, \dots, \mathfrak{A}_n \in \mathsf{REL}$, the *disjoint union* $\bigsqcup_{i=1}^{n} \mathfrak{A}_i$ is defined as the structure $\langle \{\langle i, x\rangle \mid i \in [n], x \in |\mathfrak{A}_i|\}, \{\langle \langle i, x\rangle, \langle i, y\rangle\rangle \mid i \in [n], \langle x, y\rangle \in a^{\mathfrak{A}_i}\}_{a \in \Sigma}\rangle$. We consider the following gluing operator.

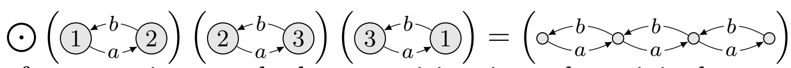
Definition 5.1. Let $\vec{\mathfrak{A}} = \mathfrak{A}_1 \dots \mathfrak{A}_n \in \mathsf{REL}^+$. The structure $\odot\vec{\mathfrak{A}}$ (or written $\odot_{i=1}^{n}\mathfrak{A}_i$) is defined as follows:

$$\odot\vec{\mathfrak{A}} \triangleq \left(\bigsqcup_{i=1}^{n} \mathfrak{A}_i\right) / \sim_{\vec{\mathfrak{A}}}$$

where $\sim_{\vec{\mathfrak{A}}}$ is the minimal equivalence relation on the set $\{\langle i, x\rangle \mid i \in [n], x \in |\mathfrak{A}_i|\}$ closed under the following rule:

For all x and $i \in [n-1]$, if $x \in |\mathfrak{A}_i| \cap |\mathfrak{A}_{i+1}|$, then $\langle i, x\rangle \sim_{\vec{\mathfrak{A}}} \langle i+1, x\rangle$.

(For notational simplicity, we abbreviate $\sim_{\vec{\mathfrak{A}}}$ to $\sim$ when $\vec{\mathfrak{A}}$ is clear from the context.)

For instance, . This gluing operator transforms a given path decomposition into the original structure up to isomorphisms.

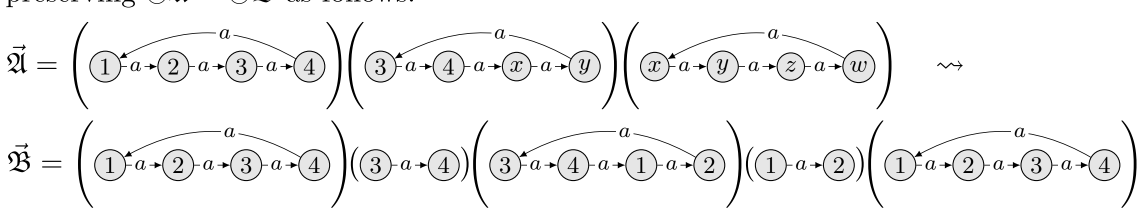
Proposition 5.2. *Let $\mathfrak{A}$ be a structure. If $\vec{\mathfrak{B}}$ is a path decomposition of $\mathfrak{A}$, then $\mathfrak{A} \cong \odot\vec{\mathfrak{B}}$.*

Proof. Easy by definition. □

From this, we have:

$$\mathsf{REL}_{\mathrm{pw} \le k} = \mathsf{I}(\{\odot\vec{\mathfrak{A}} \mid \vec{\mathfrak{A}} \in \mathsf{REL}_{k+1}^{+}\}).$$

The direction ($\supseteq$) is clear by the definition of pathwidth (Proposition 5.2). For the direction ($\subseteq$), let $\vec{\mathfrak{A}}$ be a path decomposition of width k. We then can replace names of vertices in each bag with numbers in $[k+1]$ by introducing an intermediate bag whose vertices consist of the intersection of the two adjacent bags. For instance, we can transform $\vec{\mathfrak{A}}$ into $\vec{\mathfrak{B}}$ with preserving $\odot\vec{\mathfrak{A}} \cong \odot\vec{\mathfrak{B}}$ as follows:



By the linearly bounded pathwidth model property (Proposition 2.9), the equational theory of KL terms can be reformulated as follows:

$$\mathsf{REL} \models t \le s \iff \{\odot\vec{\mathfrak{A}} \mid \vec{\mathfrak{A}} \in \mathsf{REL}_{\mathrm{iw}(t)+1}^{+}\} \models t \le s.$$

Because path decompositions are sequences (words) of structures, this characterization is compatible with the automata construction given later.

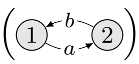
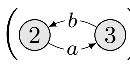
5.2. **Decomposing derivatives.** Let $\bullet$ be a fresh label for denoting a special isolated vertex. For a structure $\mathfrak{A}$, let $\mathfrak{A}_\bullet \triangleq \langle |\mathfrak{A}| \uplus \{\bullet\}, \{a^{\mathfrak{A}}\}_{a \in \Sigma}\rangle$. Clearly, for all $\tilde{t}, \tilde{s} \in \tilde{\mathbf{T}}_{|\mathfrak{A}|}$, $\tilde{t} \longrightarrow^{\mathfrak{A}} \tilde{s}$ iff $\tilde{t} \longrightarrow^{\mathfrak{A}_\bullet} \tilde{s}$. Let $\vec{\mathfrak{A}} = \mathfrak{A}_1 \dots \mathfrak{A}_n \in \mathsf{REL}^+$. In the sequel, we consider the structure $\odot((\mathfrak{A}_1)_\bullet \dots (\mathfrak{A}_n)_\bullet)$; we use $(\odot\vec{\mathfrak{A}})_\bullet$ to denote this structure (as they are isomorphic) and we may abbreviate $[\langle i, \bullet\rangle]_{\sim_{(\odot\vec{\mathfrak{A}})_\bullet}}$ to $\bullet$. For $i \in [n]$, we let $\tilde{\mathbf{T}}^{\vec{\mathfrak{A}}}_{(i)} \triangleq \tilde{\mathbf{T}}_{\{[\langle i,x\rangle]_{\sim_{(\odot\vec{\mathfrak{A}})_\bullet}} \mid x \in |(\mathfrak{A}_i)_\bullet|\}}$. For an lKL term $\tilde{t}$ and $i \in [n]$, let $\tilde{t}_{(i)}$ be the lKL term $\tilde{t}$ in which each x has been replaced with $[\langle i, x\rangle]_{\sim_{(\odot\vec{\mathfrak{A}})_\bullet}}$. We now consider the following composition of derivative relations.

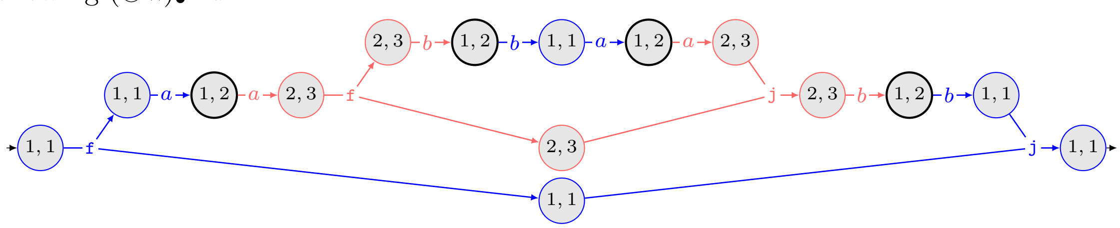
Definition 5.3. Let $\vec{\mathfrak{A}} = \mathfrak{A}_1 \dots \mathfrak{A}_n \in \mathsf{REL}^+$. The relation $\tilde{t}\ (\odot_{i=1}^n \longrightarrow^{(\mathfrak{A}_i)_\bullet})\ \tilde{s}$, where $\langle \tilde{t}, \tilde{s}\rangle \in \bigcup_{i=1}^n (\tilde{\mathbf{T}}^{\vec{\mathfrak{A}}}_{(i)})^2$, is defined as the smallest relation (of tuples of $i \in [1,n]$, $\tilde{t} \in \tilde{\mathbf{T}}^{\vec{\mathfrak{A}}}_{(i)}$, and $\tilde{s} \in \tilde{\mathbf{T}}^{\vec{\mathfrak{A}}}_{(i)}$) closed under the following rules:

$$\frac{\tilde{t} \longrightarrow^{(\mathfrak{A}_j)_\bullet} \tilde{s}}{\tilde{t}_{(j)}\ (\odot_{i=1}^n \longrightarrow^{(\mathfrak{A}_i)_\bullet})\ \tilde{s}_{(j)}}\ \mathrm{D} \qquad \frac{\tilde{t}[\bullet/l]\ (\odot_{i=1}^n \longrightarrow^{(\mathfrak{A}_i)_\bullet})\ \tilde{s}[\bullet/r]}{\tilde{t}[x/l]\ (\odot_{i=1}^n \longrightarrow^{(\mathfrak{A}_i)_\bullet})\ \tilde{s}[x/r]}\ \mathrm{L}$$

$$\frac{\tilde{t}\ (\odot_{i=1}^n \longrightarrow^{(\mathfrak{A}_i)_\bullet})\ \tilde{u} \qquad \tilde{u}\ (\odot_{i=1}^n \longrightarrow^{(\mathfrak{A}_i)_\bullet})\ \tilde{s}}{\tilde{t}\ (\odot_{i=1}^n \longrightarrow^{(\mathfrak{A}_i)_\bullet})\ \tilde{s}}\ \mathrm{T}.$$

Here, $\tilde{t}[y/j]$ denotes the lKL term $\tilde{t}$ in which the j-th label of $\tilde{t}$ has been replaced with y.

Example 5.4. Let $\vec{\mathfrak{A}} = \mathfrak{A}_1\mathfrak{A}_2$ where $\mathfrak{A}_1 = \Bigg($  $\Bigg)$ and $\mathfrak{A}_2 = \Bigg($  $\Bigg)$. Let $\tilde{t} = @[\langle 1,1\rangle]_\sim.(a^2((b^2a^2) \cap 1)b^2) \cap 1$ and let $\tilde{s} = @[\langle 1,1\rangle]_\sim.1$. We then have $\tilde{t} \longrightarrow^{(\odot\vec{\mathfrak{A}})_\bullet} \tilde{s}$ by the following $(\odot\vec{\mathfrak{A}})_\bullet$-run:



where we abbreviate $[\langle i, x\rangle]_\sim$ to "i, x" in each vertex. We let

$\tilde{t}_1 = (@[\langle 1,2\rangle]_\sim.a((b^2a^2) \cap 1)b^2) \cap_1 (@[\langle 1,1\rangle]_\sim.1)$, $\quad \tilde{t}'_1 = (@[\langle 1,2\rangle]_\sim.a((b^2a^2) \cap 1)b^2) \cap_1 (@\bullet.1)$,

$\tilde{t}'_2 = (((@[\langle 2,2\rangle]_\sim.ba^2) \cap_1 (@[\langle 2,3\rangle]_\sim.1))b^2) \cap_1 (@\bullet.1)$, $\quad \tilde{t}''_2 = (((@[\langle 1,2\rangle]_\sim.ba^2) \cap_1 (@\bullet.1))b^2) \cap_1 (@\bullet.1)$,

$\tilde{t}''_3 = (((@[\langle 1,2\rangle]_\sim.a) \cap_1 (@\bullet.1))b^2) \cap_1 (@\bullet.1)$, $\quad \tilde{t}'_3 = (((@[\langle 2,2\rangle]_\sim.a) \cap_1 (@[\langle 2,3\rangle]_\sim.1))b^2) \cap_1 (@\bullet.1)$,

$\tilde{t}'_4 = (@[\langle 2,2\rangle]_\sim.b) \cap_1 (@\bullet.1)$, $\quad \tilde{t}_4 = (@[\langle 1,2\rangle]_\sim.b) \cap_1 (@[\langle 1,1\rangle]_\sim.1)$.

We also have $\tilde{t}\ (\odot_{i=1}^2 \longrightarrow^{(\mathfrak{A}_i)_\bullet})\ \tilde{s}$, as follows:

$$\dfrac{\dfrac{}{\tilde{t} \rightsquigarrow \tilde{t}_1}\ \mathrm{D}\ (\mathfrak{A}_1) \qquad \dfrac{\dfrac{\dfrac{}{\tilde{t}'_1 \rightsquigarrow \tilde{t}'_2}\ \mathrm{D}\ (\mathfrak{A}_2) \qquad \dfrac{\dfrac{}{\tilde{t}''_2 \rightsquigarrow \tilde{t}''_3}\ \mathrm{D}\ (\mathfrak{A}_1)}{\tilde{t}'_2 \rightsquigarrow \tilde{t}'_3}\ \mathrm{L} \qquad \dfrac{}{\tilde{t}'_3 \rightsquigarrow \tilde{t}'_4}\ \mathrm{D}\ (\mathfrak{A}_2)}{\dfrac{\tilde{t}'_1 \rightsquigarrow \tilde{t}'_4}{\tilde{t}_1 \rightsquigarrow \tilde{t}_4}\ \mathrm{L}}\ \mathrm{T} \qquad \dfrac{}{\tilde{t}_4 \rightsquigarrow \tilde{s}}\ \mathrm{D}\ (\mathfrak{A}_1)}{\tilde{t} \rightsquigarrow \tilde{s}}\ \mathrm{T}$$

Note that we cannot use pairs of lKL terms that both labels $[\langle 1, 1\rangle]_\sim$ and $[\langle 2, 3\rangle]_\sim$ appear in each vertex of the derivation tree, because these labels do not occur simultaneously in $\mathfrak{A}_1$ or $\mathfrak{A}_2$; so, we should consider an appropriate strategy.

The following decomposition theorem shows that by composing derivative relations $\longrightarrow^{\mathfrak{A}_i}$ using the rules of Definition 5.3, we can obtain the derivative relation $\longrightarrow^{(\odot\vec{\mathfrak{A}})\bullet}$. Namely, we can decompose a derivative relation on a large structure into derivative relations on small structures, as follows. The soundness ($\Longleftarrow$) is easy. For the completeness ($\Longrightarrow$), we appropriately decompose derivative relations, as in Example 5.4. The proof will be given in Section 7.

Theorem 5.5 (Decomposition theorem)**.** *Let* $\vec{\mathfrak{A}} = \mathfrak{A}_1 \dots \mathfrak{A}_n \in \mathsf{REL}^+$. *For all* $j \in [n]$ *and lKL terms* $\tilde{t}, \tilde{s} \in \tilde{\mathbf{T}}^{\vec{\mathfrak{A}}}_{(j)}$, *we have:*

$$\tilde{t} \longrightarrow^{(\odot\vec{\mathfrak{A}})\bullet} \tilde{s} \iff \tilde{t}\ (\odot_{i=1}^{n} \longrightarrow^{(\mathfrak{A}_i)\bullet})\ \tilde{s}.$$

5.3. **Reducing to 2AFAs.** Using Theorem 5.5, we can give a reduction from the equational theory of PCoR* into the inclusion problem of two-way alternating finite word automata (2AFAs) [LLS78, LLS84] (the following definition is based on [GO14]).[3]

5.3.1. *2AFAs.* We use $\triangleright$ and $\triangleleft$ as the special characters denoting the left end-marker and right end-marker, respectively. A *2AFA* $\mathcal{A}$ over a finite set A is a tuple $\mathcal{A} = \langle |\mathcal{A}|, \delta^{\mathcal{A}}, 1^{\mathcal{A}}\rangle$, where

- $|\mathcal{A}|$ is a finite set of states;
- $\delta^{\mathcal{A}} \colon |\mathcal{A}| \times (A \uplus \{\triangleright, \triangleleft\}) \to \mathbb{B}_+(|\mathcal{A}| \times \{-1, 0, 1\})$ is a *transition function*, where $\mathbb{B}_+(X)$ denotes the set of positive boolean formulas over a set X given by
$$\varphi, \psi \in \mathbb{B}_+(X) \ ::= \ p \mid \mathsf{f} \mid \mathsf{t} \mid \varphi \vee \psi \mid \varphi \wedge \psi \qquad (p \in X);$$
- $1^{\mathcal{A}} \in |\mathcal{A}|$ is the initial state.

For a 2AFA $\mathcal{A}$ and a word $w = a_0 \dots a_{n-1}$ over $A \uplus \{\triangleright, \triangleleft\}$, the set $S^{\mathcal{A}}_w \subseteq |\mathcal{A}| \times [0, n-1]$ is defined as the smallest set closed under the following rule: For $\langle q, i\rangle \in |\mathcal{A}| \times [0, n-1]$ and propositional variables $\langle q_1, i_1\rangle, \dots, \langle q_m, i_m\rangle \in |\mathcal{A}| \times \{-1, 0, 1\}$, when the positive boolean formula $\delta^{\mathcal{A}}(q, a_i)$ is true under that each $\langle q_k, i_k\rangle$ is true, then

$$\frac{\langle q_1, i + i_1\rangle \in S^{\mathcal{A}}_w \quad \dots \quad \langle q_m, i + i_m\rangle \in S^{\mathcal{A}}_w}{\langle q, i\rangle \in S^{\mathcal{A}}_w}.$$

For a 2AFA $\mathcal{A}$, the language is defined as $[\mathcal{A}] \triangleq \{w \in A^* \mid \langle 1^{\mathcal{A}}, 0\rangle \in S^{\mathcal{A}}_{\triangleright w \triangleleft}\}$. We define the *size* $\|\mathcal{A}\|$ as $\sum_{\langle q,a\rangle \in |\mathcal{A}| \times (A \uplus \{\triangleright,\triangleleft\})} \|\delta^{\mathcal{A}}(q, a)\|$ where $\|\varphi\|$ denotes the number of symbols occurring in the positive boolean formula φ.[4]

[3] See also [KZ21] for a comparison of definitions of "2AFA". In their classification, our definition corresponds to monotone 2BFA, as we use monotone (but, possibly non-basic) formulas in transition functions.

[4] Note that $\|\delta^{\mathcal{A}}(q, a)\|$ has no limit in general. Also, even if we take $\delta^{\mathcal{A}}(q, a)$ as the minimum one among equivalent positive boolean formulas, the size is not bounded by a polynomial in $(\#|\mathcal{A}|\#A)$, because the number of monotone boolean functions (a.k.a. the Dedekind number) with $2n$ variables, is $\Omega(2^{\binom{2n}{n}})$ by considering antichains in which each set has exactly n elements (and hence, $2^{2^{\Omega(n)}}$ by Stirling's approximation). Due to this, we count up each $\|\delta^{\mathcal{A}}(q, a)\|$ in the definition of the size $\|\mathcal{A}\|$.

Proposition 5.6. *The* inclusion problem *for 2AFAs—given a finite set A and given two 2AFAs $\mathcal{A}$ and $\mathcal{B}$ over A, does $[\mathcal{A}] \subseteq [\mathcal{B}]$ hold?—is decidable in* PSPACE.

Proof. Let $\mathcal{A}'$ be the (one-way) non-deterministic finite automata (1NFA) of $2^{\mathcal{O}(n \log n)}$ states such that $[\mathcal{A}'] = [\mathcal{A}]$ obtained by [GO14, Lemma 1], and let $\bar{\mathcal{B}}'$ be the 1NFA of $2^{\mathcal{O}(n)}$ states such that $[\bar{\mathcal{B}}'] = A^* \setminus [\mathcal{B}]$ obtained by [GO14, Lemma 5] (see also [CDGLV02, Theorem 8]). Here, n is the number of states in the input 2AFA. We have $[\mathcal{A}] \not\subseteq [\mathcal{B}]$ iff $[\mathcal{A}'] \cap [\bar{\mathcal{B}}'] \neq \emptyset$. Then the right-hand side can be decided in a non-deterministic polynomial space algorithm by the "on-the-fly" checking of the non-emptiness problem of the product 1NFA of $\mathcal{A}'$ and $\bar{\mathcal{B}}'$.[5] Hence, by Savitch's theorem ((co-)NPSPACE = PSPACE) [Sav70], this completes the proof. □

5.3.2. *2AFAs construction.* Using 2AFAs, we can naturally encode the rules of Definition 5.3.

Definition 5.7. For $k \geq 1$ and a KL term u, we define the 2AFA $\mathcal{A}^u_k$ over REL_k as follows:

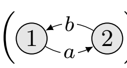
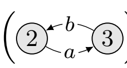
- $|\mathcal{A}^u_k| \triangleq \{1\} \uplus (\mathrm{cl}_{[k] \cup \{\bullet\}}(u)^2 \times \{?, \checkmark\})$, we abbreviate $\langle \tilde{t}, \tilde{s}, p \rangle$ to $\langle \tilde{t}, \tilde{s} \rangle_p$;
- $1^{\mathcal{A}^u_k} \triangleq 1$;
- $\delta^{\mathcal{A}^u_k}(q, a) \triangleq \bigvee X(q, a)$ where $X(q, a) \subseteq \mathbb{B}_+(|\mathcal{A}^u_k| \times \{-1, 0, +1\})$ is the smallest set closed under the following rules:

$$\frac{\mathsf{E}_y(\tilde{s})}{\langle \langle @x.\ u, \tilde{s} \rangle_?, +1 \rangle \in X(1^{\mathcal{A}^u_k}, \triangleright)}\ \mathrm{I}\ (x, y \in [k]) \qquad \frac{\overrightarrow{\mathsf{lab}}(\tilde{t}), \overrightarrow{\mathsf{lab}}(\tilde{s}) \in |\mathfrak{A}_\bullet|^+}{\langle \langle \tilde{t}, \tilde{s} \rangle_\checkmark, 0 \rangle \in X(\langle \tilde{t}, \tilde{s} \rangle_?, \mathfrak{A})}\ \checkmark$$

$$\frac{}{\langle \langle \tilde{t}, \tilde{s} \rangle_?, -1 \rangle \in X(\langle \tilde{t}, \tilde{s} \rangle_\checkmark, \mathfrak{A})}\ -1 \qquad \frac{}{\langle \langle \tilde{t}, \tilde{s} \rangle_?, +1 \rangle \in X(\langle \tilde{t}, \tilde{s} \rangle_\checkmark, \mathfrak{A})}\ +1 \qquad \frac{\tilde{t} \longrightarrow^{\mathfrak{A}_\bullet} \tilde{s}}{\mathsf{t} \in X(\langle \tilde{t}, \tilde{s} \rangle_\checkmark, \mathfrak{A})}\ \mathrm{D}$$

$$\frac{\overrightarrow{\mathsf{lab}}(\tilde{u}') \in |\mathfrak{A}_\bullet|^+}{\langle \langle \tilde{t}, \tilde{u}' \rangle_\checkmark, 0 \rangle \wedge \langle \langle \tilde{u}', \tilde{s} \rangle_\checkmark, 0 \rangle \in X(\langle \tilde{t}, \tilde{s} \rangle_\checkmark, \mathfrak{A})}\ \mathrm{T} \qquad \frac{}{\langle \langle \tilde{t}[\bullet/l], \tilde{s}[\bullet/r] \rangle_\checkmark, 0 \rangle \in X(\langle \tilde{t}[x/l], \tilde{s}[x/r] \rangle_\checkmark, \mathfrak{A})}\ \mathrm{L}$$

The rules are defined based on Definition 5.3. Intuitively, the check mark "✓" in $\langle \tilde{t}, \tilde{s} \rangle_\checkmark$ expresses that $\overrightarrow{\mathsf{lab}}(\tilde{t}), \overrightarrow{\mathsf{lab}}(\tilde{s}) \in |\mathfrak{A}_\bullet|^+$ holds (where $\mathfrak{A}$ is the structure in the current position); the question mark "?" in $\langle \tilde{t}, \tilde{s} \rangle_?$ expresses that it has not yet been checked. They are introduced to check it after the rule -1 or $+1$ (as 2AFAs cannot read sibling structures in one step).

Example 5.8. Let $\vec{\mathfrak{A}} = \mathfrak{A}_1 \mathfrak{A}_2$ where $\mathfrak{A}_1 = \Big($  $\Big)$ and $\mathfrak{A}_2 = \Big($  $\Big)$. Let

$$\tilde{t}_1 = @1.(a^2 b^2) \cap 1, \qquad \tilde{t}_2 = (@2.ab^2) \cap_1 (@1.1), \qquad \tilde{t}'_2 = (@2.ab^2) \cap_1 (@\bullet.1),$$
$$\tilde{t}'_3 = (@2.b) \cap_1 (@\bullet.1), \qquad \tilde{t}_3 = (@2.b) \cap_1 (@1.1), \qquad \tilde{t}_4 = @1.1.$$

Then $(\tilde{t}_1)_{(1)} \rightsquigarrow (\tilde{t}_4)_{(1)}$ holds by the following derivation tree (with respect to the rules Definition 5.3), where we abbreviate $(\odot_{i=1}^2 \longrightarrow^{(\mathfrak{A}_i)_\bullet})$ to $(\rightsquigarrow)$:

[5] See [GO14] for the precise constructions of $\mathcal{A}'$ and $\bar{\mathcal{B}}'$. For $\mathcal{A}'$ [GO14, Lemma 1], we can encode each state of $\mathcal{A}'$ (expressed as a sequence over subsets of $|\mathcal{A}|$) in $\mathcal{O}(n \log n)$-space because its length is at most $2n$ and each state of $\mathcal{A}$ occurs at most 2 times. Given $q, q' \in |\mathcal{A}'|$ and $a \in A \uplus \{\triangleright, \triangleleft\}$, the membership $\langle q, q' \rangle \in a^{\mathcal{A}'}$ can be decided in polynomial space, by construction. Similarly for $\bar{\mathcal{B}}'$ [GO14, Lemma 5], the membership $\langle q, q' \rangle \in a^{\bar{\mathcal{B}}'}$ can be decided in polynomial space. Thus, the "on-the-fly" checking is indeed possible.

$$\dfrac{\dfrac{}{(\tilde{t}_1)_{(1)} \rightsquigarrow (\tilde{t}_2)_{(1)}}\ \mathrm{D}\ ({\mathfrak{A}_1}) \qquad \dfrac{\dfrac{}{(\tilde{t}'_2)_{(2)} \rightsquigarrow (\tilde{t}'_3)_{(2)}}\ \mathrm{D}\ ({\mathfrak{A}_2})}{(\tilde{t}_2)_{(1)} \rightsquigarrow (\tilde{t}_3)_{(1)}}\ \mathrm{L} \qquad \dfrac{}{(\tilde{t}_3)_{(1)} \rightsquigarrow (\tilde{t}_4)_{(1)}}\ \mathrm{D}\ ({\mathfrak{A}_1})}{(\tilde{t}_1)_{(1)} \rightsquigarrow (\tilde{t}_4)_{(1)}}\ \mathrm{T}$$

Let $u = a^2b^2 \cap 1$. This tree can be simulated in $\mathcal{A}_k^u$ as follows, where we omit "$\in S_{\triangleright\vec{\mathfrak{A}}\triangleleft}^{\mathcal{A}_k^u}$":

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\langle\langle \tilde{t}_1, \tilde{t}_2\rangle_{\checkmark}, 1\rangle}\ \mathrm{D}\ (\mathfrak{A}_1) \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\langle\langle \tilde{t}'_2, \tilde{t}'_3\rangle_{\checkmark}, 2\rangle}\ \mathrm{D}\ (\mathfrak{A}_2)}{\langle\langle \tilde{t}'_2, \tilde{t}'_3\rangle_{?}, 2\rangle}\ \checkmark}{\langle\langle \tilde{t}'_2, \tilde{t}'_3\rangle_{\checkmark}, 1\rangle}\ {+1}}{\langle\langle \tilde{t}_2, \tilde{t}_3\rangle_{\checkmark}, 1\rangle}\ \mathrm{L} \qquad \dfrac{}{\langle\langle \tilde{t}_3, \tilde{t}_4\rangle_{\checkmark}, 1\rangle}\ \mathrm{D}\ (\mathfrak{A}_1)}{\langle\langle \tilde{t}_1, \tilde{t}_4\rangle_{\checkmark}, 1\rangle}\ \mathrm{T}}{\langle\langle \tilde{t}_1, \tilde{t}_4\rangle_{?}, 1\rangle}\ \checkmark}{\langle 1^{\mathcal{A}_k^u}, 0\rangle}\ \mathrm{I}$$

Since the rules of $\mathcal{A}_k^u$ are defined based on the rules of Definition 5.3, we have the following, as expected. Both directions are shown by easy induction on the derivative trees.

Proposition 5.9 (Appendix A)**.** *Let $k \geq 1$ and let u be an lKL term. Let $\mathfrak{A}_1 \ldots \mathfrak{A}_n \in \mathsf{REL}_k^+$. For all $j \in [n]$ and $\tilde{t}, \tilde{s} \in \mathrm{cl}_{|\mathfrak{A}_j| \cup \{\bullet\}}(u)$, we have:*

$$\tilde{t}_{(j)}\ (\odot_{i=1}^{n} \longrightarrow^{(\mathfrak{A}_i)_\bullet})\ \tilde{s}_{(j)} \iff \langle\langle \tilde{t}, \tilde{s}\rangle_{\checkmark}, j\rangle \in S_{\triangleright\mathfrak{A}_1 \ldots \mathfrak{A}_n\triangleleft}^{\mathcal{A}_k^u}.$$

Thus, we have the following lemma.

Lemma 5.10. *For $k \geq 1$ and a KL term t, we have: $[\mathcal{A}_k^t] = \{\vec{\mathfrak{A}} = \mathfrak{A}_1 \ldots \mathfrak{A}_n \in \mathsf{REL}_k^+ \mid \langle [\langle 1, x\rangle]_\sim, [\langle 1, y\rangle]_\sim\rangle \in [\![t]\!]_{\odot\vec{\mathfrak{A}}}$ for some $x, y \in |\mathfrak{A}_1|\}$.*

Proof. Let $\vec{\mathfrak{A}} = \mathfrak{A}_1 \ldots \mathfrak{A}_n \in \mathsf{REL}_k^+$. We have:

$$\begin{aligned}
&\vec{\mathfrak{A}} \in [\mathcal{A}_k^t] \\
&\iff \exists x, y \in |\mathfrak{A}_1|, \exists \tilde{s} \in \mathrm{cl}_{[k]}(t),\ \langle\langle @x.t, \tilde{s}\rangle_{\checkmark}, 1\rangle \in S_{\triangleright\vec{\mathfrak{A}}\triangleleft}^{\mathcal{A}_k^t} \wedge \mathsf{E}_y(\tilde{s}) && \text{(By the form of } \mathcal{A}_k^t\text{)} \\
&\iff \exists x, y \in |\mathfrak{A}_1|, \exists \tilde{s} \in \mathrm{cl}_{[k]}(t),\ (@x.t)_{(1)}\ (\odot_{i=1}^{n} \longrightarrow^{(\mathfrak{A}_i)_\bullet})\ \tilde{s}_{(1)} \wedge \mathsf{E}_y(\tilde{s}) && \text{(Proposition 5.9)} \\
&\iff \exists x, y \in |\mathfrak{A}_1|, \exists \tilde{s} \in \mathrm{cl}_{[k]}(t),\ (@x.t)_{(1)} \longrightarrow^{(\odot\vec{\mathfrak{A}})_\bullet}\ \tilde{s}_{(1)} \wedge \mathsf{E}_y(\tilde{s}) && \text{(Theorem 5.5)} \\
&\iff \exists x, y \in |\mathfrak{A}_1|, \exists \tilde{s} \in \mathrm{cl}_{[k]}(t),\ (@x.t)_{(1)} \longrightarrow^{\odot\vec{\mathfrak{A}}}\ \tilde{s}_{(1)} \wedge \mathsf{E}_{[\langle 1, y\rangle]_\sim}(\tilde{s}_{(1)}) \\
&\iff \exists x, y \in |\mathfrak{A}_1|,\ [\langle 1, y\rangle]_\sim \in [\![@[\langle 1, x\rangle]_\sim.t]\!]_{\odot\vec{\mathfrak{A}}}. && \text{(Theorem 4.7)}
\end{aligned}$$

Hence, this completes the proof. □

When l and r are fresh variables, we have: $\mathsf{REL} \models t \leq s$ iff $\mathsf{REL} \models \top l t r \top \leq \top l s r \top$. For PCoR* terms of the form $\top u \top$, either $[\![\top u \top]\!]_{\mathfrak{A}} = \emptyset$ or $[\![\top u \top]\!]_{\mathfrak{A}} = |\mathfrak{A}|^2$ holds. Thus, if there is a KL term $u_\top$ such that $[\![u_\top]\!]_{\mathfrak{A}} = [\![\top]\!]_{\mathfrak{A}}$ (note that $\top$ is not a KL term), then we have: $\mathsf{REL}_{\mathrm{pw} \leq k-1} \models t \leq s$ iff $\{\vec{\mathfrak{A}} \in \mathsf{REL}_k^+ \mid [\![u_\top l t r u_\top]\!]_{\odot\vec{\mathfrak{A}}} \neq \emptyset\} \subseteq \{\vec{\mathfrak{A}} \in \mathsf{REL}_k^+ \mid [\![u_\top l s r u_\top]\!]_{\odot\vec{\mathfrak{A}}} \neq \emptyset\}$ iff $[\mathcal{A}_k^{u_\top l t r u_\top}] \subseteq [\mathcal{A}_k^{u_\top l s r u_\top}]$ (by Lemma 5.10).

From this, if we can encode $\top$, then we can encode the equational theory of KL terms. In the following, we consider encoding $\top$ on our automata construction.

5.3.3. *Normalized path decompositions.* Let $\mathcal{L}_k^{\mathrm{N}} \triangleq \mathsf{REL}_k^+ \setminus (\mathcal{L}_k^{\mathrm{Inac}} \cup \mathcal{L}_k^{\mathrm{Incon}})$ where

$$\mathcal{L}_k^{\mathrm{Inac}} \triangleq \mathsf{REL}_k^* \,;\, \left\{\mathfrak{A}_1\mathfrak{A}_2 \in \mathsf{REL}_k^2 \mid |\mathfrak{A}_1| \cap |\mathfrak{A}_2| = \emptyset\right\} ; \mathsf{REL}_k^*,$$

$$\mathcal{L}_k^{\mathrm{Incon}} \triangleq \mathsf{REL}_k^* \,;\, \left\{\mathfrak{A}_1\mathfrak{A}_2 \in \mathsf{REL}_k^2 \mid \exists b \in \Sigma,\ b^{\mathfrak{A}_1} \cap (|\mathfrak{A}_1| \cap |\mathfrak{A}_2|)^2 \neq b^{\mathfrak{A}_2} \cap (|\mathfrak{A}_1| \cap |\mathfrak{A}_2|)^2\right\} ; \mathsf{REL}_k^*.$$

Intuitively, $\mathcal{L}_k^{\mathrm{Inac}}$ detects adjacent structures not sharing any vertices, and $\mathcal{L}_k^{\mathrm{Incon}}$ detects adjacent structures such that their relations are not the same in the sharing part. The following proposition shows that $\mathcal{L}_k^{\mathrm{N}}$ is sufficient to enumerate all structures of pathwidth at most $k-1$.

Proposition 5.11. *For $k \geq 2$, we have* $\mathsf{REL}_{\mathrm{pw}\leq k-1} = \mathsf{I}(\{\odot\vec{\mathfrak{A}} \mid \vec{\mathfrak{A}} \in \mathcal{L}_k^{\mathrm{N}}\})$.

Proof. Let $\vec{\mathfrak{A}} = \mathfrak{A}_1 \dots \mathfrak{A}_n \in \mathsf{REL}_k^+$. If $|\mathfrak{A}_i| \cap |\mathfrak{A}_{i+1}| = \emptyset$ for some i, then by putting a new component consisting of one vertex in $\mathfrak{A}_i$ and one vertex in $\mathfrak{A}_{i+1}$ between $\mathfrak{A}_i$ and $\mathfrak{A}_{i+1}$, we can avoid the condition of $\mathcal{L}_k^{\mathrm{Inac}}$. Moreover, for each $\langle [\langle i, x\rangle]_\sim, [\langle i, y\rangle]_\sim\rangle \in b^{\odot\vec{\mathfrak{A}}}$, we add a new edge for $\langle x, y\rangle \in b^{\mathfrak{A}_i}$ in $\mathfrak{A}_i$; then, we can avoid the condition of $\mathcal{L}_k^{\mathrm{Incon}}$. □

In the sequel, we consider $\mathcal{L}_k^{\mathrm{N}}$ instead of REL_k^+. This is useful for encoding additional operators (Sections 5.3.4 and 6). We use the condition of $\mathcal{L}_k^{\mathrm{Inac}}$ to encode $\top$ and we use the condition of $\mathcal{L}_k^{\mathrm{Incon}}$ to encode tests and nominals.

5.3.4. *Encoding the equational theory of KL.* Let $c_\top$ be a special letter for encoding $\top$. We consider the following language:

$$\mathcal{L}_k^{\top} \triangleq \mathsf{REL}_k^* \,;\, \{\mathfrak{A} \in \mathsf{REL}_k \mid c_\top^{\mathfrak{A}} \neq |\mathfrak{A}|^2\} ; \mathsf{REL}_k^*.$$

Note that for every $\vec{\mathfrak{A}} \in \mathsf{REL}_k^+ \setminus (\mathcal{L}_k^{\top} \cup \mathcal{L}_k^{\mathrm{Inac}})$, we have $[\![c_\top^*]\!]_{\odot\vec{\mathfrak{A}}} = [\![\top]\!]_{\odot\vec{\mathfrak{A}}}$. We then can reduce the equational theory into the inclusion problem for 2AFAs, as follows.

Lemma 5.12. *Let $k \geq 2$. Let t and s be KL terms. Let $l, r, c_\top$ be fresh variables. Then,*

$$\mathsf{REL}_{\mathrm{pw}\leq k-1} \models t \leq s \iff [\mathcal{A}_k^{c_\top^* l t r c_\top^*}] \subseteq [\mathcal{A}_k^{c_\top^* l s r c_\top^*}] \cup \mathcal{L}_k^{\mathrm{Inac}} \cup \mathcal{L}_k^{\mathrm{Incon}} \cup \mathcal{L}_k^{\top}.$$

Proof. We have:

$$\begin{aligned}
&\mathsf{REL}_{\mathrm{pw}\leq k-1} \models t \leq s \iff \mathsf{REL}_{\mathrm{pw}\leq k-1} \models \top l t r \top \leq \top l s r \top \\
&\iff \{\odot\vec{\mathfrak{A}} \mid \vec{\mathfrak{A}} \in \mathcal{L}_k^{\mathrm{N}} \setminus \mathcal{L}_k^{\top}\} \models \top l t r \top \leq \top l s r \top && \text{(Section 5.3.3 and } c_\top \text{ is fresh)} \\
&\iff \forall \vec{\mathfrak{A}} \in \mathcal{L}_k^{\mathrm{N}} \setminus \mathcal{L}_k^{\top}, [\![c_\top^* l t r c_\top^*]\!]_{\odot\vec{\mathfrak{A}}} \neq \emptyset \text{ implies } [\![c_\top^* l s r c_\top^*]\!]_{\odot\vec{\mathfrak{A}}} \neq \emptyset && ([\![\top]\!]_{\odot\vec{\mathfrak{A}}} = [\![c_\top^*]\!]_{\odot\vec{\mathfrak{A}}}) \\
&\iff [\mathcal{A}_k^{c_\top^* l t r c_\top^*}] \cap (\mathcal{L}_k^{\mathrm{N}} \setminus \mathcal{L}_k^{\top}) \subseteq [\mathcal{A}_k^{c_\top^* l s r c_\top^*}] && \text{(Lemma 5.10, and } [\![\top u \top]\!]_{\odot\vec{\mathfrak{A}}} \text{ is } \emptyset \text{ or } |\odot\vec{\mathfrak{A}}|^2) \\
&\iff [\mathcal{A}_k^{c_\top^* l t r c_\top^*}] \subseteq [\mathcal{A}_k^{c_\top^* l s r c_\top^*}] \cup \mathcal{L}_k^{\mathrm{Inac}} \cup \mathcal{L}_k^{\mathrm{Incon}} \cup \mathcal{L}_k^{\top}. && \text{(By } \mathcal{L}_k^{\mathrm{N}} \triangleq \mathsf{REL}_k^+ \setminus (\mathcal{L}_k^{\mathrm{Inac}} \cup \mathcal{L}_k^{\mathrm{Incon}}))
\end{aligned}$$

Hence, this completes the proof. □

5.3.5. *Complexity: On the size of the translated 2AFA.* For the number of states of $\mathcal{A}^t_k$, by Proposition 4.13, we have $\#|\mathcal{A}^t_k| = 2^{\mathcal{O}(\mathrm{iw}(t)\log(k\|t\|))}$. For the alphabet size of $\mathcal{A}^t_k$, we have $\#\mathsf{REL}_k = 2^{\mathcal{O}(k^2\#\Sigma)}$ (the number of structures with universe S is $2^{\#S^2\#\Sigma}$, as every element of Σ denotes a binary relation; thus, $\#\mathsf{REL}_k = \sum_{S\subseteq[k];S\neq\emptyset} 2^{\#S^2\#\Sigma} \leq 2^k \times 2^{k^2\#\Sigma} = 2^{\mathcal{O}(k^2\#\Sigma)}$). For the sizes of the positive boolean formulas, we have $\|\delta^{\mathcal{A}^t_k}(q,a)\| = \mathcal{O}(k^2\#|\mathcal{A}^t_k|) = 2^{\mathcal{O}(\mathrm{iw}(t)\log(k\|t\|))}$ (the rule (I) and the rule (T) are critical). Thus, we have $\|\mathcal{A}^t_k\| = 2^{\mathcal{O}(\mathrm{iw}(t)\log(k\|t\|)+k^2\#\Sigma)}$. Hence, under $k = \mathcal{O}(\|t\|)$ and $\#\Sigma = \mathcal{O}(\|t\|)$, we have $\|\mathcal{A}^t_k\| = 2^{\mathrm{poly}(\|t\|)}$.

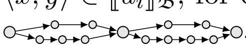
Additionally, for $\mathcal{L}^{\mathrm{Inac}}_k$, there is a 1DFA such that the number of states is $\mathcal{O}(2^k)$ (each state corresponds to the universe in the current structure). For $\mathcal{L}^{\mathrm{Incon}}_k$, there is a 1DFA such that the number of states is $\mathcal{O}(\#\mathsf{REL}_k)$ (each state corresponds to the current structure). For $\mathcal{L}^{\top}_k$, there is a 1DFA such that the number of states is 2 (hence, a constant size with respect to k). Finally, we have obtained the following complexity result.

Corollary 5.13. *The equational theory for KL terms with respect to binary relations—given a finite set Σ and KL terms t and s over Σ, does $\mathsf{REL} \models t \leq s$ hold?—is* EXPSPACE*-complete.*

Proof. (In EXPSPACE): By Lemma 5.12 with the linearly bounded pathwidth model property (Proposition 2.9), we can reduce the problem into the inclusion problem for 2AFAs:

$$\mathsf{REL} \models t \leq s \iff [\mathcal{A}^{c_\top^* ltrc_\top^*}_{\mathrm{iw}(t)+1}] \subseteq [\mathcal{A}^{c_\top^* lsrc_\top^*}_{\mathrm{iw}(t)+1}] \cup \mathcal{L}^{\mathrm{Inac}}_{\mathrm{iw}(t)+1} \cup \mathcal{L}^{\mathrm{Incon}}_{\mathrm{iw}(t)+1} \cup \mathcal{L}^{\top}_{\mathrm{iw}(t)+1}.$$

Here, we take the union construction. By the discussion above with $k = \mathrm{iw}(t) + 1 \leq \|t\| + 1$ (Proposition 2.3), the 2AFAs have exponential sizes to the input size. By Proposition 5.6, this completes the proof. (Hardness) [Nak17, BP17]: By $\mathsf{REL} \models (\sum_{a\in\Sigma} a)^* \leq t$ iff $\Sigma^* \subseteq [t]_\Sigma$ (Proposition 2.11) iff $\Sigma^* = [t]_\Sigma$, we can give a reduction from the universality problem for regular expressions with intersection, which is EXPSPACE-hard [Fü80, Theorem 2]. □

Remark 5.14. In Corollary 5.13, by [Fü80, Theorem 2], the equational theory for KL terms with respect to binary relations is EXPSPACE-hard even when $\#\Sigma = 2$. Furthermore, even when $\#\Sigma = 1$, the equational theory is EXPSPACE-hard; given KL terms t and s, by letting t' and s' be the terms t and s in which each variable a_i (where $\Sigma = \{a_1, a_2\}$) has been replaced with $u_i \triangleq b^3 \cap b^{3+i}$, we have $\mathsf{REL} \models t \leq s$ iff $\mathsf{REL} \models t' \leq s'$. (($\Longrightarrow$): Because $t' \leq s'$ is obtained by a substitution from $t \leq s$. ($\Longleftarrow$): We show the contraposition. Assume $[\![t]\!]_{\mathfrak{A}} \not\subseteq [\![s]\!]_{\mathfrak{A}}$. Let $\mathfrak{B}$ be the structure obtained by extending fresh vertices for $\langle x, y\rangle \in [\![u_i]\!]_{\mathfrak{B}}$, for each edge $\langle x, y\rangle \in [\![a_i]\!]_{\mathfrak{A}}$; for instance, if $\mathfrak{A}$ is $\circ \xrightarrow{a_1} \circ \xrightarrow{a_2} \circ$, then $\mathfrak{B}$ is . We then have $[\![t']\!]_{\mathfrak{B}} = [\![t]\!]_{\mathfrak{A}}$ or $[\![t']\!]_{\mathfrak{B}} = [\![t]\!]_{\mathfrak{A}} \cup \triangle_{|\mathfrak{B}|\setminus|\mathfrak{A}|}$, by easy induction on t; thus, $[\![t']\!]_{\mathfrak{B}} \not\subseteq [\![s']\!]_{\mathfrak{B}}$.)

6. Encoding Additional Operators

In this section, similar to how $\top$ was encoded in the reduction of Lemma 5.12 above, we provide encodings for additional operators within our automata construction. As a consequence, we have that the equational theory of PCoR* is decidable in EXPSPACE.

6.1. **Encoding PCoR*.**

In this subsection, we give encodings of $\top$ and $\_^{\smile}$. Using them, we can encode the equational theory of PCoR*.

6.1.1. *Encoding top* ($\top$). We recall $\mathcal{L}_k^{\top}$ and $c_{\top}$ (Section 5.3.4). For a term t, let t' be the unique normal form with respect to the term rewriting system: $\top \rightsquigarrow c_{\top}^*$. For $\vec{\mathfrak{A}} \in \mathsf{REL}_k^+ \setminus (\mathcal{L}_k^{\top} \cup \mathcal{L}_k^{\mathrm{Inac}})$, by $[\![c_{\top}^*]\!]_{\odot\vec{\mathfrak{A}}} = [\![\top]\!]_{\odot\vec{\mathfrak{A}}}$, we have $[\![t']\!]_{\odot\vec{\mathfrak{A}}} = [\![t]\!]_{\odot\vec{\mathfrak{A}}}$. Thus we can encode $\top$.

6.1.2. *Encoding converse* ($\_^{\smile}$). For each variable $a \in \Sigma$, we introduce a fresh variable $\breve{a}$. Let $\mathsf{REL}^{\smile} \triangleq \{\mathfrak{A} \in \mathsf{REL} \mid \forall a \in \Sigma, \breve{a} = a^{\smile}\}$. We consider the following language:

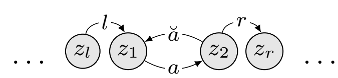
$$\mathcal{L}_k^{\smile} \triangleq \mathsf{REL}_k^* \,;\, \{\mathfrak{A} \in \mathsf{REL}_k \mid \breve{a}^{\mathfrak{A}} \neq (a^{\mathfrak{A}})^{\smile} \text{ for some } a \in \Sigma\} \,;\, \mathsf{REL}_k^*.$$

We then have $\mathsf{REL}_{\mathrm{pw} \leq k-1}^{\smile} = \{\odot\vec{\mathfrak{A}} \mid \vec{\mathfrak{A}} \in \mathsf{REL}_k^+ \setminus (\mathcal{L}_k^{\mathrm{Incon}} \cup \mathcal{L}_k^{\smile})\}$. Moreover, we consider the *converse normal form* [BP17] of PCoR*. For a PCoR* term t, we consider the unique normal form t' with respect to the following term rewriting system:

$$a^{\smile} \rightsquigarrow \breve{a}, \qquad \mathsf{1}^{\smile} \rightsquigarrow \mathsf{1}, \qquad \mathsf{0}^{\smile} \rightsquigarrow \mathsf{0}, \qquad \top^{\smile} \rightsquigarrow \top, \qquad (t \,;\, s)^{\smile} \rightsquigarrow s^{\smile} \,;\, t^{\smile},$$
$$(t + s)^{\smile} \rightsquigarrow t^{\smile} + s^{\smile}, \qquad (t \cap s)^{\smile} \rightsquigarrow t^{\smile} \cap s^{\smile}, \qquad (t^{\smile})^{\smile} \rightsquigarrow t, \qquad (t^*)^{\smile} \rightsquigarrow (t^{\smile})^*.$$

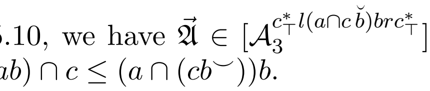
We then have $\mathsf{REL}^{\smile} \models t' = t$. Thus, within $\mathsf{REL}^{\smile}$, we can encode $\_^{\smile}$. Additionally, for the language $\mathcal{L}_k^{\smile}$, there is a 1DFA such that the number of states is 2 (a constant size).

Example 6.1. As a toy example, consider $\mathsf{REL} \models a \leq aa^{\smile}a$ [BÉS95, (10)]. We observe that this holds iff $\mathsf{REL}_{\mathrm{pw}\leq 1} \models a \leq aa^{\smile}a$ (Proposition 2.5) iff $\mathsf{REL}_{\mathrm{pw}\leq 1}^{\smile} \models a \leq a\breve{a}a$ (because $\breve{a}$ is fresh) iff $[\mathcal{A}_2^{c_{\top}^* larc_{\top}^*}] \subseteq [\mathcal{A}_2^{c_{\top}^* la\breve{a}arc_{\top}^*}] \cup \mathcal{L}_2^{\mathrm{Inac}} \cup \mathcal{L}_2^{\mathrm{Incon}} \cup \mathcal{L}_2^{\top} \cup \mathcal{L}_2^{\smile}$ (by Lemma 5.12 where $l, r, c_{\top}$ are fresh variables). If $\vec{\mathfrak{A}} \in [\mathcal{A}_2^{c_{\top}^* larc_{\top}^*}] \setminus (\mathcal{L}_2^{\mathrm{Inac}} \cup \mathcal{L}_2^{\mathrm{Incon}} \cup \mathcal{L}_2^{\top} \cup \mathcal{L}_2^{\smile})$, then by Lemma 5.10 with $[\![c_{\top}^*]\!]_{\odot\vec{\mathfrak{A}}} = [\![\top]\!]_{\odot\vec{\mathfrak{A}}}$ (by $\mathcal{L}_2^{\top}$), there are z_l, z_1, z_2, and z_r such that $\langle z_l, z_1\rangle \in [\![l]\!]_{\odot\vec{\mathfrak{A}}}$, $\langle z_1, z_2\rangle \in [\![a]\!]_{\odot\vec{\mathfrak{A}}}$, and $\langle z_2, z_r\rangle \in [\![r]\!]_{\odot\vec{\mathfrak{A}}}$. By $[\![\breve{a}]\!]_{\odot\vec{\mathfrak{A}}} = [\![a]\!]_{\odot\vec{\mathfrak{A}}}^{\smile}$ (by $\mathcal{L}_2^{\smile}$), we have $\langle z_2, z_1\rangle \in [\![\breve{a}]\!]_{\odot\vec{\mathfrak{A}}}$. Thus, the structure $\odot\vec{\mathfrak{A}}$ can be illustrated as follows:



where some of them may be the same vertex and some vertices/edges (including the universal relation $[\![c_{\top}^*]\!]_{\odot\vec{\mathfrak{A}}}$) are omitted. By the form of $\odot\vec{\mathfrak{A}}$ with Lemma 5.10, we have $\vec{\mathfrak{A}} \in [\mathcal{A}_2^{c_{\top}^* la\breve{a}arc_{\top}^*}]$. Hence, from this language inclusion, we have $\mathsf{REL} \models a \leq aa^{\smile}a$.

Example 6.2. We recall $\mathsf{REL} \models (ab) \cap c \leq (a \cap (cb^{\smile}))b$ (*cf.* Equation (2.4)). Similar to Example 6.1, this holds iff $[\mathcal{A}_3^{c_{\top}^* l(ab\cap c)rc_{\top}^*}] \subseteq [\mathcal{A}_3^{c_{\top}^* l(a\cap c\breve{b})brc_{\top}^*}] \cup \mathcal{L}_3^{\mathrm{Inac}} \cup \mathcal{L}_3^{\mathrm{Incon}} \cup \mathcal{L}_3^{\top} \cup \mathcal{L}_3^{\smile}$. If $\vec{\mathfrak{A}} \in [\mathcal{A}_3^{c_{\top}^* l(ab\cap c)rc_{\top}^*}] \setminus (\mathcal{L}_3^{\mathrm{Inac}} \cup \mathcal{L}_3^{\mathrm{Incon}} \cup \mathcal{L}_3^{\top} \cup \mathcal{L}_3^{\smile})$, then by Lemma 5.10 with $[\![c_{\top}^*]\!]_{\odot\vec{\mathfrak{A}}} = [\![\top]\!]_{\odot\vec{\mathfrak{A}}}$ (by $\mathcal{L}_3^{\top}$), there are z_l, z_1, z_2, z_3, and z_r such that $\langle z_l, z_1\rangle \in [\![l]\!]_{\odot\vec{\mathfrak{A}}}$, $\langle z_1, z_2\rangle \in [\![a]\!]_{\odot\vec{\mathfrak{A}}}$, $\langle z_2, z_3\rangle \in [\![b]\!]_{\odot\vec{\mathfrak{A}}}$, $\langle z_1, z_3\rangle \in [\![c]\!]_{\odot\vec{\mathfrak{A}}}$, and $\langle z_3, z_r\rangle \in [\![r]\!]_{\odot\vec{\mathfrak{A}}}$. By $[\![\breve{b}]\!]_{\odot\vec{\mathfrak{A}}} = [\![b]\!]_{\odot\vec{\mathfrak{A}}}^{\smile}$ (by $\mathcal{L}_3^{\smile}$), we have $\langle z_3, z_2\rangle \in [\![\breve{b}]\!]_{\odot\vec{\mathfrak{A}}}$. Thus, the structure $\odot\vec{\mathfrak{A}}$ can be illustrated as follows:



where some of them may be the same vertex and some vertices/edges are omitted. By the form of $\odot\vec{\mathfrak{A}}$ with Lemma 5.10, we have $\vec{\mathfrak{A}} \in [\mathcal{A}_3^{c_{\top}^* l(a\cap c\breve{b})brc_{\top}^*}]$. Hence, from this language inclusion, we have $\mathsf{REL} \models (ab) \cap c \leq (a \cap (cb^{\smile}))b$.

6.1.3. *Encoding the equational theory of PCoR*.* By using the encodings above, we can encode the equational theory of PCoR*. Thus, we have the following.

Corollary 6.3. *The equational theory of PCoR* is decidable and* EXPSPACE-*complete.*

Proof. (EXPSPACE-hardness): By Corollary 5.13. (In EXPSPACE): Let t and s be the KL terms obtained from given PCoR* terms t' and s' by applying the translations of Sections 6.1.1 and 6.1.2. Similar to Corollary 5.13, we can reduce into the inclusion problem for 2AFAs, as follows: $\mathsf{REL} \models t' \leq s'$ iff $\mathsf{REL}_{\mathrm{pw} \leq \mathrm{iw}(t')} \models t' \leq s'$ (Proposition 2.9) iff

$$[\mathcal{A}^{c_\top^* ltrc_\top^*}_{\mathrm{iw}(t)+1}] \subseteq [\mathcal{A}^{c_\top^* lsrc_\top^*}_{\mathrm{iw}(t)+1}] \cup \mathcal{L}^{\mathrm{Inac}}_{\mathrm{iw}(t)+1} \cup \mathcal{L}^{\mathrm{Incon}}_{\mathrm{iw}(t)+1} \cup \mathcal{L}^{\top}_{\mathrm{iw}(t)+1} \cup \mathcal{L}^{\smile}_{\mathrm{iw}(t)+1}.$$

Because these 2AFAs have exponential sizes, this completes the proof by Proposition 5.6. □

6.2. **Encoding tests and nominals.** In this subsection, moreover, we give encodings of tests in Kleene algebra with tests [KS96, Section 4] and nominals in hybrid logic [AtC07].

6.2.1. *Encoding tests.* Let $B \subseteq \Sigma$ be a finite set of *atomic tests*. Let $\mathsf{REL}^{\mathrm{tests}_B} \triangleq \{\mathfrak{A} \in \mathsf{REL} \mid \forall b \in B, b^{\mathfrak{A}} \subseteq \triangle_{|\mathfrak{A}|}\}$. The set of *PCoR* terms with tests* is given by:

$$t, s \ ::= \ p \mid a \mid \mathsf{1} \mid \mathsf{0} \mid t \,;\, s \mid t + s \mid t^*, \qquad (a \in \Sigma \setminus B)$$

$$p, q \ ::= \ b \mid \mathsf{1} \mid \mathsf{0} \mid p \,;\, q \mid p + q \mid p^-. \qquad (b \in B)$$

where the term p^- expresses the complement of p with respect to the identity—the semantics is extended by $[\![p^-]\!]_{\mathfrak{A}} = \triangle_{|\mathfrak{A}|} \setminus [\![p]\!]_{\mathfrak{A}}$ (the others are the same as PCoR* terms). Using tests, we can encode propositional while programs [FL79, Koz97]:

$$\textbf{while } p \textbf{ do } t \triangleq (pt)^* p^-, \qquad \textbf{if } p \textbf{ then } t \textbf{ else } s \triangleq (pt) + (p^- s).$$

For each $b \in B$, we introduce a fresh variable $\bar{b}$; let $\bar{B}$ be the set $\{\bar{b} \mid b \in B\}$. Let $\mathsf{REL}^{\mathrm{tests}_{B,\bar{B}}} \triangleq \{\mathfrak{A} \in \mathsf{REL} \mid \forall b \in B, b^{\mathfrak{A}} \uplus \bar{b}^{\mathfrak{A}} = \triangle_{|\mathfrak{A}|}\}$. For a term t, we consider the unique normal form t' in PCoR* terms with respect to the following term rewriting system:

$$b^- \rightsquigarrow \bar{b}, \qquad \mathsf{1}^- \rightsquigarrow \mathsf{0}, \qquad \mathsf{0}^- \rightsquigarrow \mathsf{1}, \qquad (p \,;\, q)^- \rightsquigarrow p^- + q^-, \qquad (p + q)^- \rightsquigarrow p^- \,;\, q^-.$$

Then for $\mathfrak{A} \in \mathsf{REL}^{\mathrm{tests}_{B,\bar{B}}}$, we have $[\![t']\!]_{\mathfrak{A}} = [\![t]\!]_{\mathfrak{A}}$. Let

$$\mathcal{L}_k^{\mathrm{tests}_{B,\bar{B}}} \triangleq \mathsf{REL}_k^* \,;\, \{\mathfrak{A} \in \mathsf{REL}_k \mid \exists b \in B,\ b^{\mathfrak{A}} \uplus \bar{b}^{\mathfrak{A}} \neq \triangle_{|\mathfrak{A}|}\} \,;\, \mathsf{REL}_k^*.$$

Then, $\mathsf{REL}^{\mathrm{tests}_{B,\bar{B}}}_{\mathrm{pw} \leq k-1} = \{\odot \vec{\mathfrak{A}} \mid \vec{\mathfrak{A}} \in \mathsf{REL}_k^+ \setminus (\mathcal{L}_k^{\mathrm{tests}_{B,\bar{B}}} \cup \mathcal{L}_k^{\mathrm{Incon}})\}$. For $\mathcal{L}_k^{\mathrm{tests}_{B,\bar{B}}}$, there is a 1NFA such that the number of states is a constant size. Thus, we can encode tests.

Remark 6.4. For the 1NFA of $\mathcal{L}_k^{\mathrm{tests}_{B,\bar{B}}}$, the smallest number of states is 2 (hence, a constant size) but the number of transitions is naively exponential in the size of the alphabet for tests. In Section 6.3, to obtain the PSPACE upper bound, we will give an additional technique for reducing the number of the alphabet size to polynomial to the input (Section 6.3). The modified encoding works even if the number of tests is not fixed.

6.2.2. *Encoding nominals.* Let $L \subseteq \Sigma$ be a finite set of *nominals*. Let $\mathsf{REL}^{\mathrm{noms}_L} \triangleq \{\mathfrak{A} \in \mathsf{REL} \mid \forall l \in L, \exists x \in |\mathfrak{A}|, l^{\mathfrak{A}} = \{\langle x, x\rangle\}\}$. Using nominals, we can point a vertex. In $\mathsf{REL}^{\mathrm{noms}_L}$, we can also encode the jump operator "$@l.t$" in the definition of [AtC07], by $\mathsf{REL}^{\mathrm{noms}_L} \models @l.t = \top lt$.

We define the following language:

$$\mathcal{L}_k^{\mathrm{noms}_L} \triangleq \bigcup_{l \in L} \left(\begin{array}{l} \{\mathfrak{A} \in \mathsf{REL}_k \mid l^{\mathfrak{A}} = \emptyset\}^+ \cup (\mathsf{REL}_k^* \,;\, \{\mathfrak{A} \in \mathsf{REL}_k \mid l^{\mathfrak{A}} \not\subseteq \triangle_{|\mathfrak{A}|}\} \,;\, \mathsf{REL}_k^*) \cup \\ \mathsf{REL}_k^* \,;\, \left\{ \begin{array}{l} \mathfrak{A}_1 \dots \mathfrak{A}_n \in \mathsf{REL}_k^+ \mid \exists x, \exists y, \langle x, x\rangle \in l^{\mathfrak{A}_1} \wedge \langle y, y\rangle \in l^{\mathfrak{A}_n} \\ \qquad \wedge (x \neq y \vee (x = y \wedge \exists j \in [n], x \notin |\mathfrak{A}_j|)) \end{array} \right\} ; \mathsf{REL}_k^* \end{array} \right).$$

Then, $\mathsf{REL}^{\mathrm{noms}_L}_{\mathrm{pw} \le k-1} = \{\odot\vec{\mathfrak{A}} \mid \vec{\mathfrak{A}} \in \mathsf{REL}_k^+ \setminus \mathcal{L}_k^{\mathrm{noms}_L}\}$. (By the first line, $l^{\odot\vec{\mathfrak{A}}} \neq \emptyset$ and $l^{\odot\vec{\mathfrak{A}}} \subseteq \triangle_{|\odot\vec{\mathfrak{A}}|}$. By the second line, $\#\{x \mid \langle x, x\rangle \in l^{\odot\vec{\mathfrak{A}}}\} \le 1$.) For $\mathcal{L}_k^{\mathrm{noms}_L}$, there is a 1NFA such that the number of states is $\mathcal{O}(\#L \times k)$ (intuitively, we use the states to remember $l \in L$, $x \in [k]$, and to track whether $x \notin |\mathfrak{A}_j|$ held in past structures $\mathfrak{A}_j$). Thus, we can encode nominals.

6.2.3. *Encoding the equational theory of PCoR* with tests and nominals.* From the above two, to use tests and nominals, we consider PCoR* terms in the class $\mathsf{REL}^{\mathrm{tests}_{B,\bar{B}}, \mathrm{noms}_L} \triangleq \mathsf{REL}^{\mathrm{tests}_{B,\bar{B}}} \cap \mathsf{REL}^{\mathrm{noms}_L}$. For the class, we have the following proposition (*cf.* Proposition 2.9).

Proposition 6.5. *For all PCoR* terms t, s, we have:*

$$\mathsf{REL}^{\mathrm{tests}_{B,\bar{B}}, \mathrm{noms}_L} \models t \le s \iff \mathsf{REL}^{\mathrm{tests}_{B,\bar{B}}, \mathrm{noms}_L}_{\mathrm{pw} \le \mathrm{iw}(t) + \#L} \models t \le s.$$

Proof. ($\Longrightarrow$): Trivial. ($\Longleftarrow$): We prove the contraposition. Assume $\langle x, y\rangle \in [\![t]\!]_{\mathfrak{A}} \setminus [\![s]\!]_{\mathfrak{A}}$ for some $\mathfrak{A} \in \mathsf{REL}^{\mathrm{tests}_{B,\bar{B}}, \mathrm{noms}_L}$. By Proposition 2.6, $G' \longrightarrow \mathsf{G}(\mathfrak{A}, x, y)$ for some $G' \in \mathcal{G}(t)$, and $H \not\longrightarrow \mathsf{G}(\mathfrak{A}, x, y)$ for all $H \in \mathcal{G}(s)$. Let G be the graph G' extended with L isolated vertices having $l_1, \dots, l_{\#L}$ labelled looping edges, respectively, where $L = \{l_1, \dots, l_{\#L}\}$. Then there is a graph homomorphism $h \colon G \longrightarrow \mathsf{G}(\mathfrak{A}, x, y)$ by extending $G' \longrightarrow \mathsf{G}(\mathfrak{A}, x, y)$. Let $\mathcal{S}_{B,\bar{B}}(h) \triangleq \langle |G|, \{a^{\mathcal{S}_{B,\bar{B}}(h)}\}_{a \in \Sigma}, 1^G, 2^G\rangle$ where the binary relation $a^{\mathcal{S}_{B,\bar{B}}(h)}$ is defined by

$$a^{\mathcal{S}_{B,\bar{B}}(h)} \triangleq \begin{cases} a^G & (a \in \Sigma \setminus (B \cup \bar{B})) \\ a^G \cup \{\langle x, x\rangle \in \triangle_{|G|} \mid \langle h(x), h(x)\rangle \in a^H\} & (a \in B \cup \bar{B}). \end{cases}$$

Note that $h \colon \mathcal{S}_{B,\bar{B}}(h) \longrightarrow \mathsf{G}(\mathfrak{A}, x, y)$. Let $\sim$ be the minimal equivalence relation satisfying the following two conditions:

- for each $l \in L$, if $\langle x, y\rangle, \langle x', y'\rangle \in l^{\mathcal{S}_{B,\bar{B}}(h)}$, then $x \sim y'$ (hence, $x \sim y \sim x' \sim y'$);
- for each $b \in B \cup \bar{B}$, if $\langle x, y\rangle \in b^{\mathcal{S}_{B,\bar{B}}(h)}$, then $x \sim y$.

Then $\mathcal{S}_{B,\bar{B}}(h)/{\sim} \longrightarrow \mathsf{G}(\mathfrak{A}, x, y)$ holds because $h(x) = h(y)$ holds for all $x \sim y$. Let $\mathfrak{B}$, x', and y' be such that $\mathsf{G}(\mathfrak{B}, x', y') = \mathcal{S}_{B,\bar{B}}(h)/{\sim}$. By $G' \longrightarrow G \longrightarrow \mathcal{S}_{B,\bar{B}}(h) \longrightarrow \mathsf{G}(\mathfrak{B}, x', y')$, we have $\langle x', y'\rangle \in [\![t]\!]_{\mathfrak{B}}$ (Proposition 2.6). By $\mathsf{G}(\mathfrak{B}, x', y') \longrightarrow \mathsf{G}(\mathfrak{A}, x, y)$, we also have $\langle x', y'\rangle \notin [\![s]\!]_{\mathfrak{B}}$ (Proposition 2.6). Because $\mathfrak{B} \in \mathsf{REL}^{\mathrm{tests}_{B,\bar{B}}, \mathrm{noms}_L}_{\mathrm{pw} \le \mathrm{iw}(t) + \#L}$ (note that by $\mathrm{pw}(\mathcal{S}_{B,\bar{B}}(h)) = \mathrm{pw}(G) = \mathrm{pw}(G') \le \mathrm{iw}(t)$ (Proposition 2.5), we have $\mathrm{pw}(\mathcal{S}_{B,\bar{B}}(h)/{\sim}) \le \mathrm{iw}(t) + \#L$), this completes the proof. □

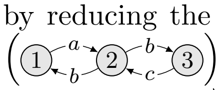
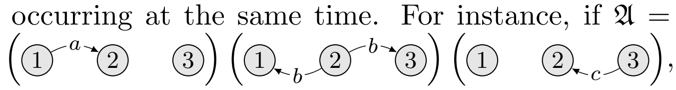
Using this bounded pathwidth model property, we have the following corollary.

Corollary 6.6. *The equational theory of PCoR* terms with tests and nominals is decidable and* EXPSPACE*-complete.*

Proof. (EXPSPACE-hardness): By Corollary 5.13. (In EXPSPACE): Given terms t'' and s'' (where B and $\bar{B}$ are used for tests and L is used for nominals), let t' and s' be the PCoR* terms obtained by applying the translations of Section 6.2.1, and let t and s be the KL terms obtained by applying the translations of Sections 6.1.1 and 6.1.2, respectively. Similar to Corollary 5.13, we can reduce into the inclusion problem for 2AFAs, as follows: $\mathsf{REL}^{\mathrm{tests}_B, \mathrm{noms}_L} \models t'' \leq s''$ iff $\mathsf{REL}^{\mathrm{tests}_{B,\bar{B}}, \mathrm{noms}_L} \models t' \leq s'$ (Section 6.2.1) iff $\mathsf{REL}^{\mathrm{tests}_{B,\bar{B}}, \mathrm{noms}_L}_{\mathrm{pw} \leq \mathrm{iw}(t) + \#L} \models t' \leq s'$ (Proposition 6.5) iff

$$[\mathcal{A}_k^{c_\top^* ltrc_\top^*}] \subseteq [\mathcal{A}_k^{c_\top^* lsrc_\top^*}] \cup \mathcal{L}_k^{\mathrm{Inac}} \cup \mathcal{L}_k^{\mathrm{Incon}} \cup \mathcal{L}_k^{\top} \cup \mathcal{L}_k^{\smile} \cup \mathcal{L}_k^{\mathrm{tests}_{B,\bar{B}}} \cup \mathcal{L}_k^{\mathrm{noms}_L}$$

where $k \triangleq \mathrm{iw}(t) + \#L + 1$. Because these 2AFAs have exponential sizes, this completes the proof by Proposition 5.6. □

6.3. **Reducing the alphabet size for PSPACE upper bound.** In this subsection, we note that we can reduce the alphabet size from $\#\mathsf{REL}_k = 2^{\mathcal{O}(\#\Sigma \times k^2)}$ into $\#\Sigma \times 2^{\mathcal{O}(k^2)}$, by reducing the number of variables occurring at the same time. For instance, if $\mathfrak{A} =$ , then by letting $\vec{\mathfrak{B}} =$ , we have $\mathfrak{A} \cong \odot\vec{\mathfrak{B}}$. In the sequel, we consider partitions of Σ given as follows:

$$\{b, \bar{b}, \breve{b}, \breve{\bar{b}}\} \text{ for } b \in B, \qquad \{l, \breve{l}\} \text{ for } l \in L, \qquad \{a, \breve{a}\} \text{ for other variables } a.$$

We enumerate them as a sequence $\Sigma_0, \dots, \Sigma_{m-1}$. Let $\mathsf{supp}(\mathfrak{A}) \triangleq \{a \in \Sigma \mid a^{\mathfrak{A}} \neq \emptyset\}$ and let $\mathsf{REL}'_k \triangleq \{\mathfrak{A} \in \mathsf{REL}_k \mid \exists i < m, \mathsf{supp}(\mathfrak{A}) \subseteq \Sigma_i\}$. As above, to enumerate all structures of pathwidth at most $k-1$, it suffices to consider REL'_k. Since $\#\Sigma_i$ is bounded by a fixed number ($\#\Sigma_i \leq 4$), we have $\#\mathsf{REL}'_k = \#\Sigma \times 2^{\mathcal{O}(k^2)}$. However, this decomposition breaks the condition of $\mathcal{L}_k^{\mathrm{Incon}}$. For that reason, we consider the following well-aligned universe:

$$\mathcal{L}_k^{\mathrm{U}'} \triangleq \{\mathfrak{A}_0 \dots \mathfrak{A}_{m-1} \mid |\mathfrak{A}_0| = \dots = |\mathfrak{A}_{m-1}| \wedge \forall i < m, \mathsf{supp}(\mathfrak{A}_i) \subseteq \Sigma_i\}^+,$$

and then we replace $\mathcal{L}_k^{\mathrm{Incon}}$ with the language $\mathcal{L}_k^{\mathrm{Incon}'}$ defined by:

$$(\mathsf{REL}'_k)^* ; \bigcup_{a \in \Sigma} \left\{ \mathfrak{A}_0 \dots \mathfrak{A}_m \in (\mathsf{REL}'_k)^{m+1} \;\middle|\; \begin{array}{l} a^{\mathfrak{A}_0} \cap (|\mathfrak{A}_0| \cap |\mathfrak{A}_m|)^2 \\ \neq a^{\mathfrak{A}_m} \cap (|\mathfrak{A}_0| \cap |\mathfrak{A}_m|)^2 \end{array} \right\} ; (\mathsf{REL}'_k)^*.$$

According to this, similarly for the others, we construct them as follows:

$$\begin{aligned}
\mathcal{L}_k^{\mathrm{Inac}'} &\triangleq (\mathsf{REL}'_k)^* ; \{\mathfrak{A}_1 \mathfrak{A}_2 \in (\mathsf{REL}'_k)^2 \mid |\mathfrak{A}_1| \cap |\mathfrak{A}_2| = \emptyset\} ; (\mathsf{REL}'_k)^*, \\
\mathcal{L}_k^{\top'} &\triangleq ((\mathsf{REL}'_k)^m)^* ; (\mathsf{REL}'_k)^{i_\top - 1} ; \{\mathfrak{A} \in \mathsf{REL}'_k \mid c_\top^{\mathfrak{A}} \neq |\mathfrak{A}|^2\} ; (\mathsf{REL}'_k)^* \\
&\qquad\qquad \text{where } i_\top \text{ is such that } c_\top \in \Sigma_{i_\top}, \\
\mathcal{L}_k^{\smile'} &\triangleq (\mathsf{REL}'_k)^* ; \{\mathfrak{A} \in \mathsf{REL}'_k \mid \breve{a}^{\mathfrak{A}} \neq (a^{\mathfrak{A}})^{\smile}\} ; (\mathsf{REL}'_k)^*, \\
\mathcal{L}_k^{\mathrm{tests}'_{B,\bar{B}}} &\triangleq ((\mathsf{REL}'_k)^m)^* ; (\bigcup_{b \in B} (\mathsf{REL}'_k)^{i_b - 1} ; \{\mathfrak{A} \in \mathsf{REL}'_k \mid b^{\mathfrak{A}} \uplus \bar{b}^{\mathfrak{A}} \neq \triangle_{|\mathfrak{A}|}\}) ; (\mathsf{REL}'_k)^* \\
&\qquad\qquad \text{where } i_b \text{ is such that } b \in \Sigma_{i_b},
\end{aligned}$$

$$\mathcal{L}_k^{\mathrm{noms}'_L} \triangleq \bigcup_{l \in L} \left(\begin{array}{l} \{\mathfrak{A} \in \mathsf{REL}'_k \mid l^{\mathfrak{A}} = \emptyset\}^+ \cup \\ ((\mathsf{REL}'_k)^* ; \{\mathfrak{A} \in \mathsf{REL}'_k \mid l^{\mathfrak{A}} \not\subseteq \triangle_{|\mathfrak{A}|}\} ; (\mathsf{REL}'_k)^*) \cup \\ (\mathsf{REL}'_k)^* ; \left\{ \begin{array}{l} \mathfrak{A}_1 \dots \mathfrak{A}_n \in (\mathsf{REL}'_k)^+ \mid \exists x, \exists y, \\ \langle x, x\rangle \in l^{\mathfrak{A}_1} \wedge \langle y, y \rangle \in l^{\mathfrak{A}_n} \\ \wedge (x \neq y \vee (x = y \wedge \exists j \in [n], x \notin |\mathfrak{A}_j|)) \end{array} \right\} ; (\mathsf{REL}'_k)^* \end{array} \right).$$

Each of them is a regular language, and there exists a 1NFA of $\mathcal{O}(2^k m)$ states for $\mathcal{L}_k^{\mathrm{U}'}$, $\mathcal{O}(2^k m)$ for $\mathcal{L}_k^{\mathrm{Incon}'}$, $\mathcal{O}(2^k)$ for $\#\mathcal{L}_k^{\mathrm{Inac}'}$, $\mathcal{O}(m)$ for $\#\mathcal{L}_k^{\top'}$, $\mathcal{O}(1)$ for $\#\mathcal{L}_k^{\smile'}$, $\mathcal{O}(m)$ for $\#\mathcal{L}_k^{\mathrm{tests}'_{B,\bar{B}}}$, and $\mathcal{O}(k\#L)$ for $\#\mathcal{L}_k^{\mathrm{noms}'_L}$. Thus, assuming k is fixed, for each language, there is a 1NFA such that the number of states is $\mathcal{O}(\|t\|)$. The alphabet size REL'_k is also $\mathcal{O}(\|t\|)$.

6.3.1. *The bounded intersection width fragment.* We moreover show that when the intersection width [GLL09] is fixed, the equational theory is PSPACE-complete.

Corollary 6.7. *Let $k \geq 1$. The equational theory of PCoR* terms with tests and nominals of intersection width at most k is* PSPACE-*complete.*

Proof. (PSPACE-hardness): Because the universality problem for regular expressions is PSPACE-complete [MS72]. (In PSPACE): Similar to Corollary 6.6, from given PCoR* terms with tests and nominals t' and s', let t and s be the KL terms obtained by applying the translations of Sections 6.1.1, 6.1.2 and 6.2.1. By using the automata above with $\mathrm{iw}(t) \leq k$, we can reduce into the inclusion problem for 2AFAs, as follows: $\mathsf{REL} \models t' \leq s'$ iff

$$[\mathcal{A}_{k+1}^{c_\top^* l t' r c_\top^*}] \cap \mathcal{L}_{k+1}^{\mathrm{U}'} \subseteq [\mathcal{A}_{k+1}^{c_\top^* l s' r c_\top^*}] \cup \mathcal{L}_{k+1}^{\mathrm{Inac}'} \cup \mathcal{L}_{k+1}^{\mathrm{Incon}'} \cup \mathcal{L}_{k+1}^{\top'} \cup \mathcal{L}_{k+1}^{\smile'} \cup \mathcal{L}_{k+1}^{\mathrm{tests}'_{B,\bar{B}}} \cup \mathcal{L}_{k+1}^{\mathrm{noms}'_L}.$$

Because these 2AFAs have polynomial sizes, this completes the proof by Proposition 5.6. □

Particularly when $k = 1$, we have the following, which slightly extends the PSPACE upper bound of Kleene algebra with top and tests [PW24] and of Kleene algebra with top and converse [Nak23].

Corollary 6.8. *The equational theory for Kleene algebra with top, converse, tests, and nominals with respect to binary relations is* PSPACE-*complete.*

7. Proof of Theorem 5.5: The Decomposition Theorem

In this section, we prove Theorem 5.5, which is the only remaining part.

Theorem 5.5 (Decomposition theorem). *Let $\vec{\mathfrak{A}} = \mathfrak{A}_1 \dots \mathfrak{A}_n \in \mathsf{REL}^+$. For all $j \in [n]$ and lKL terms $\tilde{t}, \tilde{s} \in \tilde{\mathbf{T}}_{(j)}^{\vec{\mathfrak{A}}}$, we have:*

$$\tilde{t} \longrightarrow^{(\odot \vec{\mathfrak{A}})\bullet} \tilde{s} \iff \tilde{t} \left(\odot_{i=1}^{n} \longrightarrow^{(\mathfrak{A}_i)\bullet}\right) \tilde{s}.$$

We first show the decomposition theorem for a subclass of runs (Lemma 7.10 in Section 7.1), and then we extend this result for lKL terms.

7.1. **Decomposition for local sub-SP runs.** In this subsection, we first show the decomposition theorem for *local sub-series-parallel runs* (local sub-SP runs). In Section 7.1.1, we define sub-SP runs. In Section 7.1.2, we define *local* sub-SP runs. In Section 7.1.3, we show this theorem.

7.1.1. *Sub-SP runs.* The set SPR of *series-parallel (SP) runs* is the subclass of runs with $\langle 1, 1\rangle$-interface defined by:

$$G, H, J \in \mathsf{SPR} \ ::= \ 1^1 \mid a^1_1 \mid G \diamond H \mid \mathtt{f}^1_1 \diamond (G \parallel H) \diamond \mathtt{j}^1_1 \qquad (a \in \Sigma)$$

For runs G and G', we say that G' is a *sub-run* of G if there are runs H and J such that $H \diamond G' \diamond J = G$. We write subSPR for the set of all *sub-series-parallel runs* (sub-SP runs):

$$\mathsf{subSPR} \ \triangleq \ \{G' \mid \text{there are runs } H,\ J \text{ and an SP run } G \text{ such that } G = H \diamond G' \diamond J\}.$$

Similarly, we define the sets of *sub-SP-l runs* and *sub-SP-r runs*, written $\mathsf{subSPRl}$ and $\mathsf{subSPRr}$, as follows:

$$\mathsf{subSPRl} \ \triangleq \ \{G' \mid \text{there are a run } H \text{ and an SP run } G \text{ such that } G = H \diamond G'\},$$
$$\mathsf{subSPRr} \ \triangleq \ \{G' \mid \text{there are a run } J \text{ and an SP run } G \text{ such that } G = G' \diamond J\}.$$

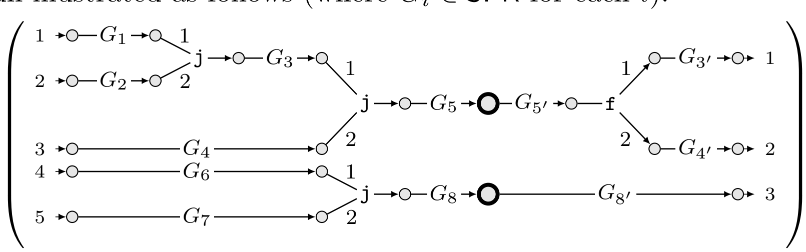
Namely, sub-SP-l runs are runs obtained by left quotients of SP runs. Similarly, sub-SP-r runs are runs obtained by "right quotients" of SP runs.

Below, we present alternative definitions of subSPR, $\mathsf{subSPRl}$, and $\mathsf{subSPRr}$.

$$G, H, J \in \mathsf{subSPR}' \ ::= \ (H^{\mathrm{l}} \diamond J^{\mathrm{r}}) \mid (H \parallel J) \qquad (H^{\mathrm{l}} \in \mathsf{subSPRl}',\ J^{\mathrm{r}} \in \mathsf{subSPRr}',\ H, J \in \mathsf{subSPR}')$$
$$G^{\mathrm{l}}, H^{\mathrm{l}}, J^{\mathrm{l}} \in \mathsf{subSPRl}' \ ::= \ J \mid (G^{\mathrm{l}} \parallel H^{\mathrm{l}}) \diamond \mathtt{j}^1_1 \diamond J \qquad (G^{\mathrm{l}}, H^{\mathrm{l}} \in \mathsf{subSPRl}',\ J \in \mathsf{SPR})$$
$$G^{\mathrm{r}}, H^{\mathrm{r}}, J^{\mathrm{r}} \in \mathsf{subSPRr}' \ ::= \ J \mid J \diamond \mathtt{f}^1_1 \diamond (G^{\mathrm{r}} \parallel H^{\mathrm{r}}) \qquad (G^{\mathrm{r}}, H^{\mathrm{r}} \in \mathsf{subSPRr}',\ J \in \mathsf{SPR})$$

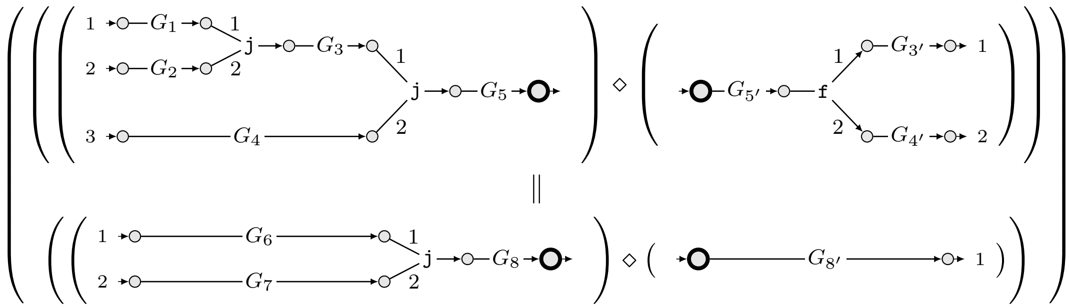
Proposition 7.1 (Appendix B)**.** *We have* $\mathsf{subSPRl} = \mathsf{subSPRl}'$*,* $\mathsf{subSPRr} = \mathsf{subSPRr}'$*, and* $\mathsf{subSPR} = \mathsf{subSPR}'$*.*

Thanks to this alternative definition, we have that every sub-SP run is obtained by the (multiple) parallel compositions of SP runs having some vertex such that all sources are reachable to the vertex and all targets are reachable from the vertex (note that $\mathrm{ty}_2(G) = 1$ for every $G \in \mathsf{subSPRl}$ and $\mathrm{ty}_1(G) = 1$ for every $G \in \mathsf{subSPRr}$). For instance, let us consider the sub-SP run illustrated as follows (where $G_i \in \mathsf{SPR}$ for each i):

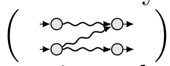


.

This is clearly a sub-SP run by filling up both the left-hand side and the right-hand side, appropriately. This sub-SP run is decomposed to the parallel compositions of runs $G \diamond H$ where $G \in \mathsf{subSPRl}$ and $H \in \mathsf{subSPRr}$, as follows.



.

Remark 7.2. Every (sub-)SP run does not contain the four pairwise distinct vertices of the form , *cf.* "N-free" [VTL79, p. 17][LW98]. Here, $\circ\!\rightsquigarrow\!\circ$ expresses that the target vertex is reachable from the source vertex.

7.1.2. *Labeled KL terms and local sub-SP runs.* We first note the following proposition.

Proposition 7.3. *Let $\mathfrak{A}$ be a structure.*

- *For all KL terms t, every $\mathfrak{A}$-run of t is SP.*
- *For all lKL terms $\tilde{t}$, every $\mathfrak{A}$-run of $\tilde{t}$ is sub-SP-l.*

Proof. By easy induction on terms. □

Moreover, each $(\odot\vec{\mathfrak{A}})_\bullet$-run of lKL terms is *local* in the following sense.

Definition 7.4. Let $\vec{\mathfrak{A}} = \mathfrak{A}_1 \dots \mathfrak{A}_n$. We say that an $(\odot\vec{\mathfrak{A}})_\bullet$-run $\tau \colon G \longrightarrow (\odot\vec{\mathfrak{A}})_\bullet$ is *local* if for all $\langle v_1, \dots, v_k\rangle \in a^G$, there is some $i \in [n]$ s.t. $\{\tau(v_1), \dots, \tau(v_k)\} \subseteq \{[\langle i, x\rangle]_\sim \mid x \in |\mathfrak{A}_i|\}$.

Proposition 7.5. *Let $\tilde{t}$ be an lKL term. Every $(\odot\vec{\mathfrak{A}})_\bullet$-run of $\tilde{t}$ is local sub-SP-l.*

Proof. For "sub-SP-l": By Proposition 7.3. For "local": By definition of $(\odot\vec{\mathfrak{A}})_\bullet$-run. □

7.1.3. *Decomposition of local sub-SP runs.* Let $\vec{\mathfrak{A}} = \mathfrak{A}_1 \dots \mathfrak{A}_n \in \mathsf{REL}^+$. Let $\mathsf{lsubSPR}^{(\odot\vec{\mathfrak{A}})_\bullet}$ be the set of local sub-SP $(\odot\vec{\mathfrak{A}})_\bullet$-runs. Similarly, we use $\mathsf{lSPR}^{(\odot\vec{\mathfrak{A}})_\bullet}$, $\mathsf{lsubSPRl}^{(\odot\vec{\mathfrak{A}})_\bullet}$, and $\mathsf{lsubSPRr}^{(\odot\vec{\mathfrak{A}})_\bullet}$. We now define the decompositions of local sub-SP $(\odot\vec{\mathfrak{A}})_\bullet$-runs. For $i \in [n]$, we let $\mathsf{lsubSPR}^{\vec{\mathfrak{A}}}_{(i)} \triangleq \{\tau \in \mathsf{lsubSPR}^{(\odot\vec{\mathfrak{A}})_\bullet} \mid \mathrm{ty}_1(\tau)\mathrm{ty}_2(\tau) \in \{[\langle i, x\rangle]_\sim \mid x \in |(\mathfrak{A}_i)_\bullet|\}^*\}$. For $i \in [n]$ and an $(\mathfrak{A}_i)_\bullet$-run τ, we write $\tau_{(i)}$ for the $(\odot\vec{\mathfrak{A}})_\bullet$-run τ in which each label x has been replaced with $[\langle i, x\rangle]_\sim$. We then consider the following decomposition of $(\odot\vec{\mathfrak{A}})_\bullet$-run.

Definition 7.6. Let $\vec{\mathfrak{A}} = \mathfrak{A}_1 \dots \mathfrak{A}_n \in \mathsf{REL}^+$. The set $\odot_{i=1}^n \mathsf{lsubSPR}^{(\mathfrak{A}_i)_\bullet}$ is defined as the smallest subset of $\bigcup_{i=1}^n \mathsf{lsubSPR}^{\vec{\mathfrak{A}}}_{(i)}$ closed under the following rules:

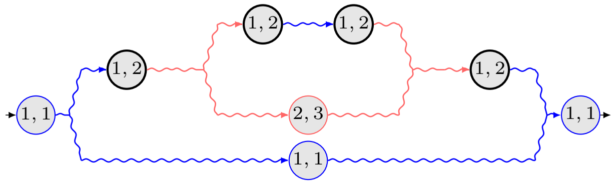
$$\frac{\tau \in \mathsf{lsubSPR}^{\mathfrak{A}_j}}{\tau_{(j)} \in \odot_{i=1}^n \mathsf{lsubSPR}^{(\mathfrak{A}_i)_\bullet}}\ \mathrm{D} \qquad \frac{\tau[\bullet/\langle l, r\rangle] \in \odot_{i=1}^n \mathsf{lsubSPR}^{(\mathfrak{A}_i)_\bullet}}{\tau[x/\langle l, r\rangle] \in \odot_{i=1}^n \mathsf{lsubSPR}^{(\mathfrak{A}_i)_\bullet}}\ \mathrm{L}$$

$$\frac{\tau \in \odot_{i=1}^n \mathsf{lsubSPR}^{(\mathfrak{A}_i)_\bullet} \qquad \sigma \in \odot_{i=1}^n \mathsf{lsubSPR}^{(\mathfrak{A}_i)_\bullet}}{\tau \diamond \sigma \in \odot_{i=1}^n \mathsf{lsubSPR}^{(\mathfrak{A}_i)_\bullet}}\ \mathrm{T}$$

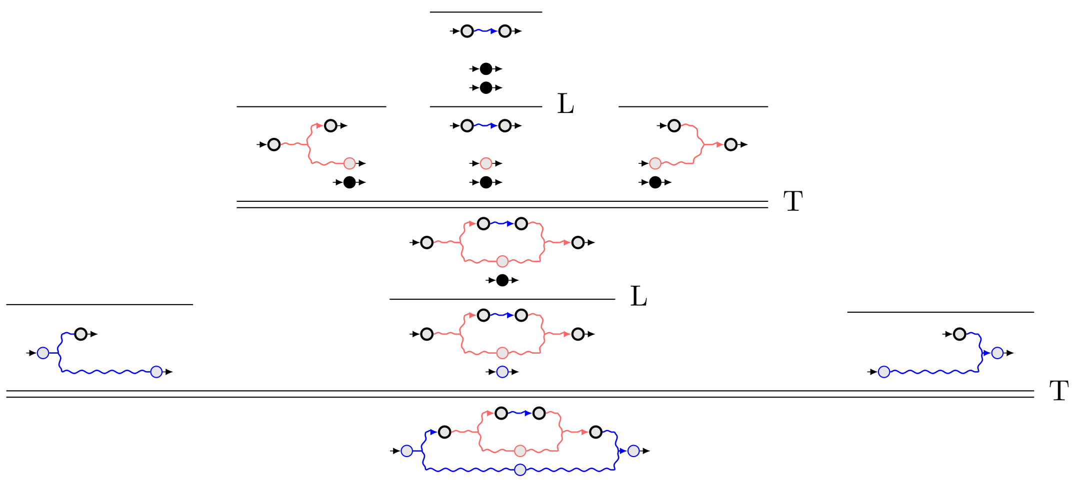
Here, $\tau[y/\langle l, r\rangle]$ denotes the $(\odot\vec{\mathfrak{A}})_\bullet$-run τ in which $\tau(1^\tau_l)$ and $\tau(2^\tau_r)$ has been replaced with y.

We then have that every $(\odot\vec{\mathfrak{A}})_\bullet$-run of $\bigcup_{i=1}^{n} \mathsf{lsubSPR}^{\vec{\mathfrak{A}}}_{(i)}$ can be derived using the rules of Definition 7.6 (Lemma 7.10). Note that each $(\odot\vec{\mathfrak{A}})_\bullet$-run occurring in the derivation tree should be in $\mathsf{lsubSPR}^{\vec{\mathfrak{A}}}_{(i)}$ for some i.

Example 7.7. We recall the sequence $\vec{\mathfrak{A}} = \mathfrak{A}_1\mathfrak{A}_2$ and the following $(\odot\vec{\mathfrak{A}})_\bullet$-run $\tau \in \mathsf{lsubSPR}^{\vec{\mathfrak{A}}}_{(1)}$ in Example 5.4:



We then also have $\tau \in \odot_{i=1}^{n} \mathsf{lsubSPR}^{(\mathfrak{A}_i)\bullet}$ by the following derivation tree (*cf.* Example 5.4):

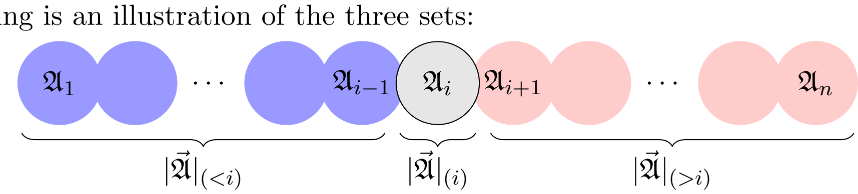


The most technical point in Example 7.7 is to find an appropriate interpolation. In the following, we show that we can always take an appropriate interpolation (Lemma 7.10).

For $\vec{\mathfrak{A}} = \mathfrak{A}_1 \dots \mathfrak{A}_n \in \mathsf{REL}^+$ and $i \in [n]$, we consider the following three disjoint sets:

$$|\vec{\mathfrak{A}}|_{(i)} \triangleq \{[\langle i, x\rangle]_\sim \mid x \in |\mathfrak{A}_i|\}, \quad |\vec{\mathfrak{A}}|_{(<i)} \triangleq (\bigcup_{j<i} |\vec{\mathfrak{A}}|_{(j)}) \setminus |\vec{\mathfrak{A}}|_{(i)}, \quad |\vec{\mathfrak{A}}|_{(>i)} \triangleq (\bigcup_{j>i} |\vec{\mathfrak{A}}|_{(j)}) \setminus |\vec{\mathfrak{A}}|_{(i)}.$$

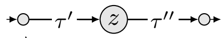
The following is an illustration of the three sets:



For $\tau \in \mathsf{lsubSPR}^{(\odot\vec{\mathfrak{A}})\bullet}$, we say that τ is *i-split* if there are some $x \in |\vec{\mathfrak{A}}|_{(<i)}$ and $y \in |\vec{\mathfrak{A}}|_{(>i)}$ occurring in the sequence $\mathrm{ty}_1(\tau)\mathrm{ty}_2(\tau)$. We first show that we can decompose each i-split $\mathsf{lSPR}^{(\odot\vec{\mathfrak{A}})\bullet}$ run into two $\mathsf{lSPR}^{(\odot\vec{\mathfrak{A}})\bullet}$ runs such that each glued vertex is $|\vec{\mathfrak{A}}|_{(i)}$-labeled.

Lemma 7.8. *Let $\vec{\mathfrak{A}} = \mathfrak{A}_1 \dots \mathfrak{A}_n \in \mathsf{REL}^+$ and $i \in [n]$. For all i-split $\tau \in \mathsf{lSPR}^{(\odot\vec{\mathfrak{A}})\bullet}$, there are non-empty σ and ρ such that $\tau = \sigma \diamond \rho$ and for each k, we have $\sigma(2^\sigma_k) \in |\vec{\mathfrak{A}}|_{(i)}$.*

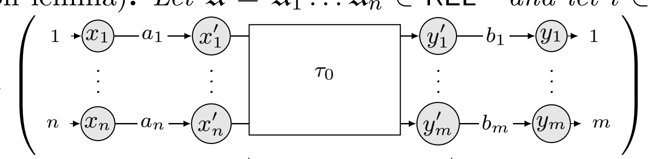
Proof. By induction on the derivation tree of $G \in \mathsf{SPR}$ where $\tau\colon G \longrightarrow (\odot\vec{\mathfrak{A}})_{\bullet}$.

- Case $\tau = a_1^{x,y}$, Case $\tau = 1^x$: Then τ is local. However, as this contradicts that τ is i-split, this case does not occur.
- Case $\tau =$ (): We distinguish the following cases:
  - Case $z \in |\vec{\mathfrak{A}}|_{(i)}$: By letting $\sigma = \tau'$ and $\rho = \tau''$, this case has been proved.
  - Otherwise: Because z is $|\vec{\mathfrak{A}}|_{(<i)}$- or $|\vec{\mathfrak{A}}|_{(>i)}$-labeled vertex, either τ' or τ'' is i-split.
    - Case τ' is i-split: Let σ' and ρ' be the ones obtained by IH with respect to τ'. Then by letting $\sigma = \sigma'$ and $\rho = \rho' \diamond \tau''$, this case has been proved.
    - Case τ'' is i-split: Similarly, let σ'' and ρ'' be the ones obtained by IH with respect to τ''. Then by letting $\sigma = \tau' \diamond \sigma''$ and $\rho = \rho''$, this case has been proved.
- Case $\tau = \mathtt{f}_1^x \diamond (\tau' \parallel \tau'') \diamond \mathtt{j}_1^y$: Since both τ' and τ'' are i-split, let σ' and ρ' be the ones obtained by IH with respect to τ' and let σ'' and ρ'' be the ones obtained by IH with respect to τ''. Then by letting $\sigma = \mathtt{f}_1^x \diamond (\sigma' \parallel \sigma'')$ and $\rho = (\rho' \parallel \rho'') \diamond \mathtt{j}_1^y$, this case has been proved.

Hence, this completes the proof. □

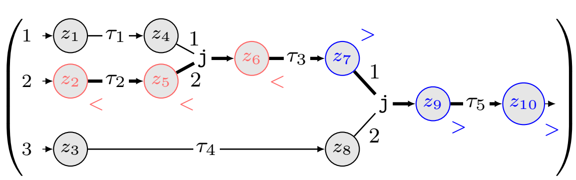
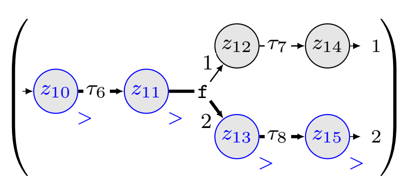
We then extend the lemma above for (slightly specialized) $\mathsf{IsubSPR}$. This is the most crucial lemma for Lemma 7.10.

Lemma 7.9 (Interpolation lemma)**.** *Let $\vec{\mathfrak{A}} = \mathfrak{A}_1 \dots \mathfrak{A}_n \in \mathsf{REL}^+$ and let $i \in [n]$. For all $\tau \in \mathsf{IsubSPR}^{\vec{\mathfrak{A}}}_{(i)}$ of the form*

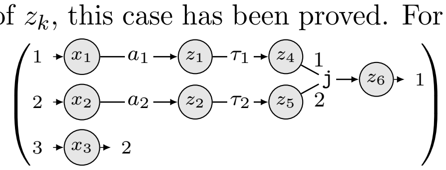


such that τ_0 is i-split and connected *(i.e., $\tau_0 \in \mathsf{IsubSPRl}^{(\odot\vec{\mathfrak{A}})_\bullet} \diamond \mathsf{IsubSPRr}^{(\odot\vec{\mathfrak{A}})_\bullet}$ by Proposition 7.1), there are non-empty σ and ρ such that $\tau = \sigma \diamond \rho$ and $\sigma, \rho \in \mathsf{IsubSPR}^{\vec{\mathfrak{A}}}_{(i)}$.*

Proof. Let $\tau_0 = (\sigma_0 \diamond \rho_0)$ where $\sigma_0 \in \mathsf{IsubSPRl}^{(\odot\vec{\mathfrak{A}})_\bullet}$ and $\rho_0 \in \mathsf{IsubSPRr}^{(\odot\vec{\mathfrak{A}})_\bullet}$. Note that σ_0 and ρ_0 can be viewed as trees of $\mathsf{lSPR}^{(\odot\vec{\mathfrak{A}})_\bullet}$ runs. Let v_k be such vertices in the tree and let z_k be the label of v_k. Let $z_<$ and $z_>$ be an $|\vec{\mathfrak{A}}|_{(<i)}$-labeled vertex and an $|\vec{\mathfrak{A}}|_{(>i)}$-labeled vertex in source or target vertices of τ (as τ is i-split). For instance, we consider the following σ_0 and ρ_0 (where $z_< = z_2$ and $z_> = z_{15}$):

$\sigma_0 =$  , $\rho_0 =$  .

We distinguish the following cases:

- Case $z_k \in |\vec{\mathfrak{A}}|_{(i)}$ for some v_k on σ_0: By letting σ and ρ be such that $\tau = \sigma \diamond \rho$ and σ consists of the edges on the left of z_k, this case has been proved. For instance, if $z_6 \in |\vec{\mathfrak{A}}|_{(i)}$ in the instance above, then $\sigma =$  .
- Case $z_k \in |\vec{\mathfrak{A}}|_{(i)}$ for some v_k on ρ_0: In the same way as above.

- Otherwise, every z_k is either $|\vec{\mathfrak{A}}|_{(<i)}$- or $|\vec{\mathfrak{A}}|_{(>i)}$-labeled. We consider a path between $z_<$ and $z_>$ (the bold edges in the above). Then on the path, there is some k such that there are an $|\vec{\mathfrak{A}}|_{(<i)}$-labeled vertex and an $|\vec{\mathfrak{A}}|_{(>i)}$-labeled vertex in source or target vertices of τ_k (*e.g.*, in the instance above, τ_3 is an expected $\mathsf{ISPR}^{(\odot\vec{\mathfrak{A}})\bullet}$ run where each $<$ and $>$ indicates $|\vec{\mathfrak{A}}|_{(<i)}$-labeled and $|\vec{\mathfrak{A}}|_{(>i)}$-labeled, respectively). We distinguish the following cases:
  - Case τ_k is on σ_0: Let σ_k and ρ_k be the ones obtained from τ_k by Lemma 7.8. Let $\sigma', \rho', \vec{z}_1, \vec{z}_2$ be such that $\sigma_0 = \sigma' \diamond (1^{\vec{z}_1} \parallel \tau_k \parallel 1^{\vec{z}_2}) \diamond \rho'$ and σ' consists of the edges on the left of the source of τ_k. Then by letting $\sigma = \sigma' \diamond (1^{\vec{z}_1} \parallel \sigma_k \parallel 1^{\vec{z}_2})$ and $\rho = (1^{\vec{z}_1} \parallel \rho_k \parallel 1^{\vec{z}_2}) \diamond \rho' \diamond \rho_0$, this case has been proved.
  - Otherwise (Case τ_k is on ρ_0): In the same way as above.

Hence, this completes the proof. □

By Lemma 7.9, we can show the decomposition theorem for local sub-SP runs, as follows.

Lemma 7.10 (Decomposition theorem for local sub-SP runs)**.** *Let* $\vec{\mathfrak{A}} = \mathfrak{A}_1 \dots \mathfrak{A}_n \in \mathsf{REL}^+$*.*

$$\bigcup_{i=1}^{n} \mathsf{IsubSPR}^{\vec{\mathfrak{A}}}_{(i)} \ = \ \odot_{i=1}^{n} \mathsf{IsubSPR}^{\mathfrak{A}_i}.$$

Proof. ($\supseteq$): Trivial by the definition of them. ($\subseteq$): We show the following by induction on the number of edges of τ: for all $i \in [n]$, if $\tau \in \mathsf{IsubSPR}^{\vec{\mathfrak{A}}}_{(i)}$, then $\tau \in \odot_{i=1}^{n} \mathsf{IsubSPR}^{\mathfrak{A}_i}$. We distinguish the following cases.

- Case τ has no edges: Then $\tau = 1^{\vec{z}}$. By the rule (R), this case has been proved.
- Case there is an a-labeled edge of $\mathfrak{A}_i$ adjacent to a source vertex:
  - Case $a \in \Sigma$: Then $\tau = a_k^{\vec{z},z'} \diamond \tau'$ where $\vec{z}z' \in |\vec{\mathfrak{A}}|^+_{(i)}$. We then have:

$$\dfrac{\dfrac{}{a_k^{\vec{z},z'}}\,\mathrm{D} \qquad \overline{\overline{\tau'}}\,(\mathrm{IH})}{a_k^{\vec{z},z'} \diamond \tau'}\,\mathrm{T}.$$

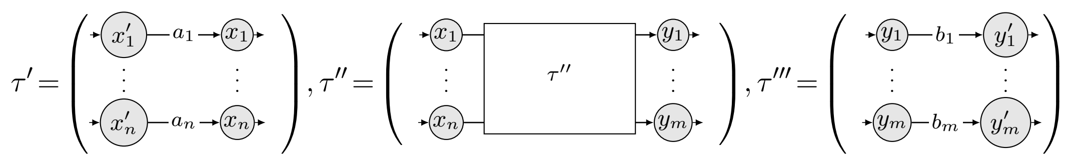
  - Case $a = \mathtt{f}$: In the same way as above.
  - Case $a = \mathtt{j}$: Then $\tau = ((1^{\vec{z}_1} \parallel (1^z \parallel \rho) \parallel 1^{\vec{z}_2}) \diamond \mathsf{j}_k^{\vec{z}_1 z \vec{z}_2}) \diamond \tau'$ or $\tau = ((1^{\vec{z}_1} \parallel (\rho \parallel 1^z) \parallel 1^{\vec{z}_2}) \diamond \mathsf{j}_k^{\vec{z}_1 z \vec{z}_2}) \diamond \tau'$ where $\rho \in \mathsf{IsubSPRl}^{\vec{\mathfrak{A}}}_{(i)}$, $\vec{z}_1 z \vec{z}_2 \in |\vec{\mathfrak{A}}|^+_{(i)}$, and $k = \|\vec{z}_1\| + 1$. Then we have:

$$\dfrac{\overline{\overline{1^{\vec{z}_1} \parallel \rho' \parallel 1^{\vec{z}_2}}}\,(\mathrm{IH}) \qquad \dfrac{}{\mathsf{j}_k^{\vec{z}_1 z \vec{z}_2}}\,\mathrm{D} \qquad \overline{\overline{\tau'}}\,(\mathrm{IH})}{((1^{\vec{z}_1} \parallel \rho' \parallel 1^{\vec{z}_2}) \diamond \mathsf{j}_k^{\vec{z}_1 z \vec{z}_2}) \diamond \tau'}\,\mathrm{T} \quad \text{where } \rho' = 1^z \parallel \rho \text{ or } \rho' = \rho \parallel 1^z.$$

- Case there is an edge of $\mathfrak{A}_i$ adjacent to a target vertex: Similar to the case above.
- Case τ has two or more weakly connected components, that is, $\tau = \tau_1 \parallel \dots \parallel \tau_p$, where $p \geq 1$, $\tau_1, \dots, \tau_p \in \mathsf{IsubSPRl}^{(\odot\vec{\mathfrak{A}})\bullet} \diamond \mathsf{IsubSPRr}^{(\odot\vec{\mathfrak{A}})\bullet}$, and τ_i is non-empty for two or more $i \in [p]$: Then we can easily take some non-empty τ' and τ'' such that $\tau = \tau' \parallel \tau''$ holds. By letting $\sigma = \tau' \parallel 1^{\mathrm{ty}_1(\tau'')}$ and $\rho = 1^{\mathrm{ty}_2(\tau')} \parallel \tau''$, we have:

$$\dfrac{\overline{\overline{\sigma}}\,(\mathrm{IH}) \qquad \overline{\overline{\rho}}\,(\mathrm{IH})}{\sigma \diamond \rho}\,\mathrm{T}.$$

- Otherwise, after applying the rule (L) for each vertex on both source and target vertices, $\tau = \mathsf{1}^{\bullet\cdots\bullet} \parallel (\tau' \diamond \tau'' \diamond \tau''') \parallel \mathsf{1}^{\bullet\cdots\bullet}$ holds (by the fourth case) where



Here, $n, m \geq 1$ (as τ has some edge by the first case) and $x_1, \dots, x_n, y_1, \dots, y_m \in |\vec{\mathfrak{A}}|_{(<i)} \cup |\vec{\mathfrak{A}}|_{(>i)}$ (by the second and third cases). We then distinguish the following cases:
  - Case $x_1, \dots, x_n, y_1, \dots, y_m \in |\vec{\mathfrak{A}}|_{(<i)}$: Let $j < i$ be the maximum j such that some edge of $a_1, \dots, a_n, b_1, \dots, b_m$ occurs in $\mathfrak{A}_j$. Let σ' and ρ' be such that $\tau' = \sigma' \diamond \rho'$ and σ' contains edges of $\mathfrak{A}_j$. Similarly, let σ''' and ρ''' be such that $\tau''' = \sigma''' \diamond \rho'''$ and ρ''' contains edges of $\mathfrak{A}_j$. Then either σ' or ρ''' is not empty and $\sigma', \rho' \diamond \tau'' \diamond \sigma''', \rho''' \in \mathsf{lsubSPR}^{\vec{\mathfrak{A}}}_{(j)}$ ($x_1, \dots, x_n, y_1, \dots, y_m \in |\vec{\mathfrak{A}}|_{(j)}$ holds by the maximality of j). Then we have:

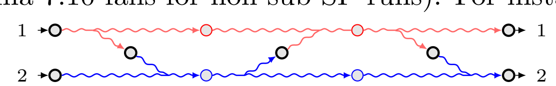
$$\dfrac{\dfrac{}{\mathsf{1}^{\bullet\cdots\bullet} \parallel \sigma' \parallel \mathsf{1}^{\bullet\cdots\bullet}}\,(\mathrm{IH}) \quad \dfrac{}{\mathsf{1}^{\bullet\cdots\bullet} \parallel (\rho' \diamond \tau'' \diamond \sigma''') \parallel \mathsf{1}^{\bullet\cdots\bullet}}\,(\mathrm{IH}) \quad \dfrac{}{\mathsf{1}^{\bullet\cdots\bullet} \parallel \rho''' \parallel \mathsf{1}^{\bullet\cdots\bullet}}\,(\mathrm{IH})}{\mathsf{1}^{\bullet\cdots\bullet} \parallel (\sigma' \diamond (\rho' \diamond \tau'' \diamond \sigma''') \diamond \rho''') \parallel \mathsf{1}^{\bullet\cdots\bullet}}\,\mathrm{T}.$$

  - Case $x_1, \dots, x_n, y_1, \dots, y_m \in |\vec{\mathfrak{A}}|_{(>i)}$: In the same way as above.
  - Otherwise, by the interpolation lemma (Lemma 7.9), there are non-empty σ and ρ such that $\tau = \mathsf{1}^{\bullet\cdots\bullet} \parallel (\sigma \diamond \rho) \parallel \mathsf{1}^{\bullet\cdots\bullet}$ and $\sigma, \rho \in \mathsf{lsubSPR}^{\vec{\mathfrak{A}}}_{(i)}$. Thus we have:

$$\dfrac{\dfrac{}{\mathsf{1}^{\bullet\cdots\bullet} \parallel \sigma \parallel \mathsf{1}^{\bullet\cdots\bullet}}\,(\mathrm{IH}) \quad \dfrac{}{\mathsf{1}^{\bullet\cdots\bullet} \parallel \rho \parallel \mathsf{1}^{\bullet\cdots\bullet}}\,(\mathrm{IH})}{\mathsf{1}^{\bullet\cdots\bullet} \parallel (\sigma \diamond \rho) \parallel \mathsf{1}^{\bullet\cdots\bullet}}\,\mathrm{T}.$$

Hence, this completes the proof. □

Remark 7.11 (On non-sub-SP runs)**.** If a local $\odot\vec{\mathfrak{A}}$-run τ is not sub-SP, we may not be able to decompose τ (and hence, Lemma 7.10 fails for non-sub-SP runs). For instance, we cannot decompose for the following τ:



7.2. **Properties of derivatives.** In this subsection, we get back to derivatives, and we show some properties of derivatives. To prove Theorem 5.5, we will use them for each rule: Lemma 7.12 for (D), Lemma 7.13 for (L), and Lemma 7.14 for (T), respectively.

Lemma 7.12. *Let $\vec{\mathfrak{A}} = \mathfrak{A}_1 \dots \mathfrak{A}_n \in \mathsf{REL}^+$ and $j \in [n]$. For all lKL terms $\tilde{t}, \tilde{s}$ and $(\mathfrak{A}_j)_\bullet$-runs τ, if $\tilde{t}_{(j)} \longrightarrow^{(\odot\vec{\mathfrak{A}})_\bullet}_{\tau_{(j)}} \tilde{s}_{(j)}$, then $\tilde{t} \longrightarrow^{(\mathfrak{A}_j)_\bullet}_{\tau} \tilde{s}$.*

Proof. By easy induction on the derivation tree with respect to Definition 4.3. Note that $\langle \tilde{t}_{(j)}, \tau_{(j)}, \tilde{s}_{(j)} \rangle$ and $\langle \tilde{t}, \tau, \tilde{s} \rangle$ are the same up to changing the names of vertices' labels. □

Lemma 7.13. *Let $\mathfrak{A}$ be a structure. For all lKL terms $\tilde{t}$ and $\tilde{s}$, $\mathfrak{A}_\bullet$-runs τ, l and r such that $1^\tau_l = 2^\tau_r$, and $x, y \in |\mathfrak{A}_\bullet|$, if $\tilde{t}[x/l] \longrightarrow^{\mathfrak{A}_\bullet}_{\tau[x/\langle l,r \rangle]} \tilde{s}[x/r]$, then $\tilde{t}[y/l] \longrightarrow^{\mathfrak{A}_\bullet}_{\tau[y/\langle l,r \rangle]} \tilde{s}[y/r]$.*

Proof. By easy induction on the derivation tree with respect to Definition 4.3. Since $1^{\tau}_{l} = 2^{\tau}_{r} = v$ for some v, the label x is not used on the derivation tree of $\longrightarrow^{\mathfrak{A}\bullet}_{\tau[x/\langle l,r\rangle]}$, thus we can replace the label of v with any label (in particular, with $\bullet$). □

Lemma 7.14. *Let $\mathfrak{A}$ be a structure. For all lKL terms $\tilde{t}$ and $\tilde{s}$ and $\mathfrak{A}$-runs τ and σ, if $\tilde{t} \longrightarrow^{\mathfrak{A}}_{\tau\diamond\sigma} \tilde{s}$, then there exists some lKL term $\tilde{u}$ such that $\tilde{t} \longrightarrow^{\mathfrak{A}}_{\tau} \tilde{u}$ and $\tilde{u} \longrightarrow^{\mathfrak{A}}_{\sigma} \tilde{s}$.*

To prove Lemma 7.14, we use the following alternative definition of $(\longrightarrow^{\mathfrak{A}}_{\tau})$. Intuitively, this is the "big-step" version of $(\longrightarrow^{\mathfrak{A}}_{\tau})$, obtained by eliminating the transitivity rule (T).

Definition 7.15. Let $\mathfrak{A}$ be a structure. The derivative relation $\tilde{t} \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{s}$, where $\tilde{t}$ and $\tilde{s}$ are lKL terms and ρ is an $\mathfrak{A}$-run, is defined as the smallest relation closed under the rules:

$$\frac{\langle x, y\rangle \in a^{\mathfrak{A}}}{@x.a \rightsquigarrow^{\mathfrak{A}}_{a^{x,y}_{1}} @y.1} \text{ for } a \in \Sigma \qquad \frac{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}'}{@x.t \,;\, s \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}' \,;_1\, s} \qquad \frac{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}' \quad \mathsf{E}_z(\tilde{t}') \quad @z.s \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{s}'}{@x.t \,;\, s \rightsquigarrow^{\mathfrak{A}}_{\rho\diamond\rho'} \tilde{s}'}$$

$$\frac{@x_0.t \rightsquigarrow^{\mathfrak{A}}_{\rho_0} \tilde{t}'_1 \quad \mathsf{E}_{x_1}(\tilde{t}'_1) \quad \dots \quad @x_{n-1}.t \rightsquigarrow^{\mathfrak{A}}_{\rho_{n-1}} \tilde{t}'_n \quad \mathsf{E}_{x_n}(\tilde{t}'_n) \quad @x_n.t \rightsquigarrow^{\mathfrak{A}}_{\rho_n} \tilde{t}'_{n+1}}{@x_0.t^* \rightsquigarrow^{\mathfrak{A}}_{\rho_0\diamond\dots\diamond\rho_n} \tilde{t}'_{n+1} \,;_1\, t^*}$$

$$\frac{@x_0.t \rightsquigarrow^{\mathfrak{A}}_{\rho_0} \tilde{t}'_1 \quad \mathsf{E}_{x_1}(\tilde{t}'_1) \quad \dots \quad @x_{n-1}.t \rightsquigarrow^{\mathfrak{A}}_{\rho_{n-1}} \tilde{t}'_n \quad \mathsf{E}_{x_n}(\tilde{t}'_n)}{@x_0.t^* \rightsquigarrow^{\mathfrak{A}}_{\rho_0\diamond\dots\diamond\rho_{n-1}} @x_n.t^*}$$

$$\frac{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}'}{@x.t + s \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}'} \qquad \frac{@x.s \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{s}'}{@x.t + s \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{s}'} \qquad \frac{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}' \quad \mathsf{E}_z(\tilde{t}') \quad @x.s \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{s}' \quad \mathsf{E}_z(\tilde{s}')}{@x.t \cap s \rightsquigarrow^{\mathfrak{A}}_{\mathtt{f}^{x}_{1}\diamond(\rho\|\rho')\diamond\mathtt{j}^{z}_{1}} @z.1}$$

$$\frac{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}' \quad @x.s \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{s}'}{@x.t \cap s \rightsquigarrow^{\mathfrak{A}}_{\mathtt{f}^{x}_{1}\diamond(\rho\|\rho')} \tilde{t}' \cap_1 \tilde{s}'} \qquad \frac{\tilde{t} \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}' \;\; \mathsf{E}_z(\tilde{t}') \quad \tilde{s} \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{s}' \;\; \mathsf{E}_z(\tilde{s}')}{\tilde{t} \cap_1 \tilde{s} \rightsquigarrow^{\mathfrak{A}}_{(\rho\|\rho')\diamond\mathtt{j}^{z}_{1}} @z.1} \qquad \frac{\tilde{t} \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}' \quad \tilde{s} \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{s}'}{\tilde{t} \cap_1 \tilde{s} \rightsquigarrow^{\mathfrak{A}}_{(\rho\|\rho')} \tilde{t}' \cap_1 \tilde{s}'}$$

$$\frac{\tilde{t} \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}'}{\tilde{t} \,;_1\, s \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}' \,;_1\, s} \qquad \frac{\tilde{t} \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}' \quad \mathsf{E}_z(\tilde{t}') \quad @z.s \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{s}}{\tilde{t} \,;_1\, s \rightsquigarrow^{\mathfrak{A}}_{\rho\diamond\rho'} \tilde{s}} \qquad \frac{}{\tilde{t} \rightsquigarrow^{\mathfrak{A}}_{1^{\overrightarrow{\mathsf{lab}}(\tilde{t})}} \tilde{t}} \text{ R}$$

Proposition 7.16 (Appendix C)**.** *For all $\mathfrak{A}$-runs τ, we have: $(\longrightarrow^{\mathfrak{A}}_{\tau}) = (\rightsquigarrow^{\mathfrak{A}}_{\tau})$.*

Proof Sketch. ($\supseteq$): Using the transitivity rule (T), we have that each rule of $\rightsquigarrow^{\mathfrak{A}}_{\tau}$ is admissible in $\longrightarrow^{\mathfrak{A}}_{\tau}$. ($\subseteq$): We have that the transitivity rule $\dfrac{\tilde{t} \rightsquigarrow^{\mathfrak{A}}_{\tau} \tilde{u} \quad \tilde{u} \rightsquigarrow^{\mathfrak{A}}_{\sigma} \tilde{s}}{\tilde{t} \rightsquigarrow^{\mathfrak{A}}_{\tau\diamond\sigma} \tilde{s}}$ T is admissible in $\rightsquigarrow^{\mathfrak{A}}_{\tau}$, which is shown by induction on the size of the derivation tree. □

Using Proposition 7.16, we can show Lemma 7.14, as follows.[6]

Proof of Lemma 7.14. By induction on the derivation tree of $\rightsquigarrow^{\mathfrak{A}}_{\rho}$.

- Case $\dfrac{\langle x, y\rangle \in a^{\mathfrak{A}}}{@x.a \rightsquigarrow^{\mathfrak{A}}_{\rho} @y.1}$: Then $\rho = \tau \diamond \sigma = a^{x,y}_{1}$. We distinguish the following sub-cases:
  - Case $\tau = a^{x,y}_{1}$ and $\sigma = 1^{y}$: By letting $\tilde{u} = @y.1$.
  - Case $\tau = 1^{x}$ and $\sigma = a^{x,y}_{1}$: By letting $\tilde{u} = @x.a$.

[6]Our method can be viewed as an analog of Maehara's method [Mae61, p. 237] in that, after eliminating the rule (T) (which corresponds to the cut rule), we construct an interpolant by induction on derivation trees.

- Case $\dfrac{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}'}{@x.t + s \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}'}$: By IH, there is some $\tilde{t}''$ such that $@x.t \rightsquigarrow^{\mathfrak{A}}_{\tau} \tilde{t}''$ and $\tilde{t}'' \rightsquigarrow^{\mathfrak{A}}_{\sigma} \tilde{t}'$. We then have $\dfrac{@x.t \rightsquigarrow^{\mathfrak{A}}_{\tau} \tilde{t}''}{@x.t + s \rightsquigarrow^{\mathfrak{A}}_{\tau} \tilde{t}''}$. Hence by letting $\tilde{u} = \tilde{t}''$, this case has been proved.
- Case $\dfrac{@x.s \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{s}'}{@x.t + s \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{s}'}$: Similar to the case above.
- Case $\dfrac{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}'}{@x.t \,;\, s \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}' \,;_1\, s}$: By IH, there is some $\tilde{t}''$ such that $@x.t \rightsquigarrow^{\mathfrak{A}}_{\tau} \tilde{t}''$ and $\tilde{t}'' \rightsquigarrow^{\mathfrak{A}}_{\sigma} \tilde{t}'$. We then have:
$$\frac{@x.t \rightsquigarrow^{\mathfrak{A}}_{\tau} \tilde{t}''}{@x.t \,;\, s \rightsquigarrow^{\mathfrak{A}}_{\tau} \tilde{t}'' \,;_1\, s} \quad \text{and} \quad \frac{\tilde{t}'' \rightsquigarrow^{\mathfrak{A}}_{\sigma} \tilde{t}'}{\tilde{t}'' \,;_1\, s \rightsquigarrow^{\mathfrak{A}}_{\sigma} \tilde{t}' \,;_1\, s}.$$
Thus by letting $\tilde{u} = \tilde{t}'' \,;_1\, s$, this case has been proved.
- Case $\dfrac{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}' \quad \mathsf{E}_z(\tilde{t}') \quad @z.s \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{s}'}{@x.t \,;\, s \rightsquigarrow^{\mathfrak{A}}_{\rho \diamond \rho'} \tilde{s}'}$: We distinguish the following cases:
  - Case $\rho = \tau \diamond \rho''$ (and $\sigma = \rho'' \diamond \rho'$): By IH, there is $\tilde{t}''$ such that $@x.t \rightsquigarrow^{\mathfrak{A}}_{\tau} \tilde{t}''$ and $\tilde{t}'' \rightsquigarrow^{\mathfrak{A}}_{\rho''} \tilde{t}'$. Then, $\dfrac{\tilde{t}'' \rightsquigarrow^{\mathfrak{A}}_{\rho''} \tilde{t}' \quad \mathsf{E}_z(\tilde{t}') \quad @z.s \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{s}'}{\tilde{t}'' \rightsquigarrow^{\mathfrak{A}}_{\rho'' \diamond \rho'} \tilde{s}'}$. By letting $\tilde{u} = \tilde{t}''$, this case has been proved.
  - Case $\rho' = \rho'' \diamond \sigma$ (and $\tau = \rho \diamond \rho''$): By IH, there is $\tilde{s}''$ such that $@z.s \rightsquigarrow^{\mathfrak{A}}_{\rho''} \tilde{s}''$ and $\tilde{s}'' \rightsquigarrow^{\mathfrak{A}}_{\sigma} \tilde{s}'$. Then, $\dfrac{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}' \quad \mathsf{E}_z(\tilde{t}') \quad @z.s \rightsquigarrow^{\mathfrak{A}}_{\rho''} \tilde{s}''}{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho \diamond \rho''} \tilde{s}''}$. By letting $\tilde{u} = \tilde{s}''$, this case has been proved.
- Case $\dfrac{\tilde{t} \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}'}{\tilde{t} \,;_1\, u \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}' \,;_1\, u}$, Case $\dfrac{\tilde{t} \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}' \quad \mathsf{E}_z(\tilde{t}') \quad @z.s \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{s}'}{\tilde{t} \,;_1\, s \rightsquigarrow^{\mathfrak{A}}_{\rho \diamond \rho'} \tilde{s}'}$: Similar to the two cases above.
- Case $\dfrac{@x_0.t \rightsquigarrow^{\mathfrak{A}}_{\rho_0} \tilde{t}'_1 \quad \mathsf{E}_{x_1}(\tilde{t}'_1) \quad \dots \quad @x_{n-1}.t \rightsquigarrow^{\mathfrak{A}}_{\rho_{n-1}} \tilde{t}'_n \quad \mathsf{E}_{x_n}(\tilde{t}'_n) \quad @x_n.t \rightsquigarrow^{\mathfrak{A}}_{\rho_n} \tilde{t}'_{n+1}}{@x_0.t^* \rightsquigarrow^{\mathfrak{A}}_{\rho_0 \diamond \dots \diamond \rho_n} \tilde{t}'_{n+1} \,;_1\, t^*}$:
Then there are some $i \in [0, n]$, ρ'_i, and ρ''_i such that $\rho_i = \rho'_i \diamond \rho''_i$, $\tau = \rho_0 \diamond \dots \diamond \rho_{i-1} \diamond \rho'_i$, and $\sigma = \rho''_i \diamond \rho_{i+1} \diamond \dots \diamond \rho_n$. By IH, there is some $\tilde{u}'$ such that $@x_i.t_i \rightsquigarrow^{\mathfrak{A}}_{\rho'_i} \tilde{u}'$ and $\tilde{u}' \rightsquigarrow^{\mathfrak{A}}_{\rho''_i} \tilde{t}'_{i+1}$. We then have:
$$\frac{@x_0.t \rightsquigarrow^{\mathfrak{A}}_{\rho_0} \tilde{t}'_1 \quad \mathsf{E}_{x_1}(\tilde{t}'_1) \quad \dots \quad @x_{i-1}.t \rightsquigarrow^{\mathfrak{A}}_{\rho_{i-1}} \tilde{t}'_i \quad \mathsf{E}_{x_i}(\tilde{t}'_i) \quad @x_i.t \rightsquigarrow^{\mathfrak{A}}_{\rho'_i} \tilde{u}'}{@x_0.t^* \rightsquigarrow^{\mathfrak{A}}_{\rho_0 \diamond \dots \diamond \rho_{i-1} \diamond \rho'_i} \tilde{u}' \,;_1\, t^*} \text{ and}$$
$$\frac{\tilde{u}' \rightsquigarrow^{\mathfrak{A}}_{\rho''_i} \tilde{t}'_{i+1} \quad \mathsf{E}_{x_{i+1}}(\tilde{t}'_{i+1}) \quad \dfrac{@x_{i+1}.t \rightsquigarrow^{\mathfrak{A}}_{\rho_{i+1}} \tilde{t}'_{i+2} \quad \mathsf{E}_{x_{i+2}}(\tilde{t}'_{i+2}) \quad \dots \quad \mathsf{E}_{x_n}(\tilde{t}'_n) \quad @x_n.t \rightsquigarrow^{\mathfrak{A}}_{\rho_n} \tilde{t}'_{n+1}}{@x_{i+1}.t^* \rightsquigarrow^{\mathfrak{A}}_{\rho_{i+1} \diamond \dots \diamond \rho_n} \tilde{t}'_{n+1} \,;_1\, t^*}}{\tilde{u}' \,;_1\, t^* \rightsquigarrow^{\mathfrak{A}}_{\rho''_i \diamond \rho_{i+1} \diamond \dots \diamond \rho_n} \tilde{t}'_{n+1} \,;_1\, t^*}$$
Hence by letting $\tilde{u} = \tilde{u}' \,;_1\, t^*$, this case has been proved.

- Case $\dfrac{@x_0.t \rightsquigarrow^{\mathfrak{A}}_{\rho_0} \tilde{t}'_1 \;\; \mathsf{E}_{x_1}(\tilde{t}'_1) \;\; \dots \;\; @x_{n-1}.t \rightsquigarrow^{\mathfrak{A}}_{\rho_{n-1}} \tilde{t}'_n \;\; \mathsf{E}_{x_n}(\tilde{t}'_n)}{@x_0.t^* \rightsquigarrow^{\mathfrak{A}}_{\rho_0 \diamond \dots \diamond \rho_{n-1}} @x_n.t^*}$: Similar to the case above.
- Case $\dfrac{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{t}' \quad \mathsf{E}_z(\tilde{t}') \quad @x.s \rightsquigarrow^{\mathfrak{A}}_{\rho''} \tilde{s}' \quad \mathsf{E}_z(\tilde{s}')}{@x.t \cap s \rightsquigarrow^{\mathfrak{A}}_{\mathtt{f}^x_1 \diamond (\rho' \parallel \rho'') \diamond \mathtt{j}^z_1} @z.1}$: We distinguish the following cases:
  - Case $\tau = \mathtt{f}^x_1 \diamond (\rho' \parallel \rho'') \diamond \mathtt{j}^z_1$ and $\sigma = 1^z$: By $@x.t \cap s \rightsquigarrow^{\mathfrak{A}}_{\tau} @z.1$ and $@z.1 \rightsquigarrow^{\mathfrak{A}}_{\sigma} @z.1$.
  - Case $\tau = 1^x$ and $\sigma = \mathtt{f}^x_1 \diamond (\rho' \parallel \rho'') \diamond \mathtt{j}^z_1$: Similar to the case above.
  - Case $\tau = \mathtt{f}^x_1 \diamond (\tau' \parallel \tau'')$ and $\sigma = (\sigma' \parallel \sigma'') \diamond \mathtt{j}^z_1$ (then, $\rho' = \tau' \diamond \sigma'$ and $\rho'' = \tau'' \diamond \sigma''$): By IH with $\rho' = \tau' \diamond \sigma'$, there is some $\tilde{t}''$ such that $@x.t \rightsquigarrow^{\mathfrak{A}}_{\tau'} \tilde{t}''$ and $\tilde{t}'' \rightsquigarrow^{\mathfrak{A}}_{\sigma'} \tilde{t}'$. By IH with $\rho'' = \tau'' \diamond \sigma''$, there is some $\tilde{s}''$ such that $@x.s \rightsquigarrow^{\mathfrak{A}}_{\tau''} \tilde{s}''$ and $\tilde{s}'' \rightsquigarrow^{\mathfrak{A}}_{\sigma''} \tilde{s}'$. Then we have:
  $$\frac{@x.t \rightsquigarrow^{\mathfrak{A}}_{\tau'} \tilde{t}'' \quad @x.s \rightsquigarrow^{\mathfrak{A}}_{\tau''} \tilde{s}''}{@x.t \cap s \rightsquigarrow^{\mathfrak{A}}_{\mathtt{f}^x_1 \diamond (\tau' \parallel \tau'')} \tilde{t}'' \cap_1 \tilde{s}''} \quad \text{and} \quad \frac{\tilde{t}'' \rightsquigarrow^{\mathfrak{A}}_{\sigma'} \tilde{t}' \;\; \mathsf{E}_z(\tilde{t}') \quad \tilde{s}'' \rightsquigarrow^{\mathfrak{A}}_{\sigma''} \tilde{s}' \;\; \mathsf{E}_z(\tilde{s}')}{\tilde{t}'' \cap_1 \tilde{s}'' \rightsquigarrow^{\mathfrak{A}}_{(\sigma' \parallel \sigma'') \diamond \mathtt{j}^z_1} @z.1}$$
  Hence by letting $\tilde{u} = \tilde{t}'' \cap_1 \tilde{s}''$, this case has been proved.
- Case $\dfrac{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{t}' \quad @x.s \rightsquigarrow^{\mathfrak{A}}_{\rho''} \tilde{s}'}{@x.t \cap s \rightsquigarrow^{\mathfrak{A}}_{\mathtt{f}^x_1 \diamond (\rho' \parallel \rho'')} \tilde{t}' \cap_1 \tilde{s}'}$, Case $\dfrac{\tilde{t} \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{t}' \quad \mathsf{E}_z(\tilde{t}') \quad \tilde{s} \rightsquigarrow^{\mathfrak{A}}_{\rho''} \tilde{s}' \quad \mathsf{E}_z(\tilde{s}')}{\tilde{t} \cap_1 \tilde{s} \rightsquigarrow^{\mathfrak{A}}_{(\rho' \parallel \rho'') \diamond \mathtt{j}^z_1} @z.1}$, and Case $\dfrac{\tilde{t} \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{t}' \quad \tilde{s} \rightsquigarrow^{\mathfrak{A}}_{\rho''} \tilde{s}'}{\tilde{t} \cap_1 \tilde{s} \rightsquigarrow^{\mathfrak{A}}_{(\rho' \parallel \rho'')} \tilde{t}' \cap_1 \tilde{s}'}$: Similar to the case above.
- Case $\dfrac{}{\tilde{t} \rightsquigarrow^{\mathfrak{A}}_{1^{\overrightarrow{\mathsf{lab}}(\tilde{t})}} \tilde{t}}$: Then $\tau = \sigma = 1^{\overrightarrow{\mathsf{lab}}(\tilde{t})}$. This case has been proved by the assumption.

Hence, this completes the proof. □

7.3. **Proof of Theorem 5.5: decomposition of derivatives.** We now prove Theorem 5.5. First, we extend Definition 5.3 with annotations $(-)_\tau$ where τ is a $(\odot\vec{\mathfrak{A}})_\bullet$-run.

Definition 7.17. Let $\vec{\mathfrak{A}} = \mathfrak{A}_1 \dots \mathfrak{A}_n \in \mathsf{REL}^+$. The relation $\tilde{t}\ (\odot_{i=1}^n \longrightarrow^{(\mathfrak{A}_i)_\bullet})_\tau\ \tilde{s}$, where $\langle \tilde{t}, \tau, \tilde{s} \rangle \in \bigcup_{j=1}^n (\tilde{\mathbf{T}}^{\vec{\mathfrak{A}}}_{(j)} \times \mathsf{lsubSPR}^{\vec{\mathfrak{A}}}_{(j)} \times \tilde{\mathbf{T}}^{\vec{\mathfrak{A}}}_{(j)})$, is defined as the smallest relation (of tuples of $\tilde{t}$, τ, and $\tilde{s}$) closed under the following rules:

$$\frac{\tilde{t} \longrightarrow^{(\mathfrak{A}_j)_\bullet}_{\tau} \tilde{s}}{\tilde{t}_{(j)}\ (\odot_{i=1}^n \longrightarrow^{(\mathfrak{A}_i)_\bullet})_{\tau_{(j)}}\ \tilde{s}_{(j)}}\ \mathrm{D} \qquad \frac{\tilde{t}[\bullet/l]\ (\odot_{i=1}^n \longrightarrow^{(\mathfrak{A}_i)_\bullet})_{\tau[\bullet/\langle l,r \rangle]}\ \tilde{s}[\bullet/r]}{\tilde{t}[x/l]\ (\odot_{i=1}^n \longrightarrow^{(\mathfrak{A}_i)_\bullet})_{\tau[x/\langle l,r \rangle]}\ \tilde{s}[x/r]}\ \mathrm{L}$$
$$\frac{\tilde{t}\ (\odot_{i=1}^n \longrightarrow^{(\mathfrak{A}_i)_\bullet})_\tau\ \tilde{u} \quad \tilde{u}\ (\odot_{i=1}^n \longrightarrow^{(\mathfrak{A}_i)_\bullet})_\sigma\ \tilde{s}}{\tilde{t}\ (\odot_{i=1}^n \longrightarrow^{(\mathfrak{A}_i)_\bullet})_{\tau \diamond \sigma}\ \tilde{s}}\ \mathrm{T}.$$

The rules of Definition 7.17 $(\odot_{i=1}^n \longrightarrow^{(\mathfrak{A}_i)_\bullet})_\tau$ are the same as those of Definition 5.3 $(\odot_{i=1}^n \longrightarrow^{(\mathfrak{A}_i)_\bullet})$ except that an $(\odot\vec{\mathfrak{A}})_\bullet$-run is annotated. Thus, the following immediately holds:

Proposition 7.18. *Let $\vec{\mathfrak{A}} = \mathfrak{A}_1 \dots \mathfrak{A}_n \in \mathsf{REL}^+$. For all $j \in [n]$ and $\tilde{t}, \tilde{s} \in \tilde{\mathbf{T}}^{\vec{\mathfrak{A}}}_{(j)}$, we have:*

$$\tilde{t}\ (\odot_{i=1}^n \longrightarrow^{(\mathfrak{A}_i)_\bullet})\ \tilde{s} \iff \exists \tau \in \mathsf{lsubSPR}^{\vec{\mathfrak{A}}}_{(j)},\ \tilde{t}\ (\odot_{i=1}^n \longrightarrow^{(\mathfrak{A}_i)_\bullet})_\tau\ \tilde{s}.$$

Proof. Both directions are shown by easy induction on the derivation trees. □

We also have the following by using the properties shown in Section 7.2.

Lemma 7.19. *Let* $\vec{\mathfrak{A}} = \mathfrak{A}_1 \dots \mathfrak{A}_n \in \mathsf{REL}^+$*. For* $\langle \tilde{t}, \tau, \tilde{s} \rangle \in \bigcup_{j=1}^{n} \tilde{\mathbf{T}}^{\vec{\mathfrak{A}}}_{(j)} \times \mathsf{lsubSPR}^{\vec{\mathfrak{A}}}_{(j)} \times \tilde{\mathbf{T}}^{\vec{\mathfrak{A}}}_{(j)}$,

$$\tilde{t} \longrightarrow_{\tau}^{(\odot\vec{\mathfrak{A}})\bullet} \tilde{s} \iff \tilde{t}\ (\odot_{i=1}^{n} \longrightarrow^{(\mathfrak{A}_i)\bullet})_{\tau}\ \tilde{s}.$$

Proof. ($\Longleftarrow$): By easy induction on the derivation tree of Definition 7.17. ($\Longrightarrow$): Using Lemma 7.10, we show this by induction on the derivation tree of $\tau \in \odot_{i=1}^{n} \mathsf{lsubSPR}^{(\mathfrak{A}_i)\bullet}$ (of the rules in Definition 7.6).

- Case $\dfrac{\tau \in \mathsf{lsubSPR}^{(\mathfrak{A}_j)\bullet}}{\tau_{(j)} \in \odot_{i=1}^{n} \mathsf{lsubSPR}^{(\mathfrak{A}_i)\bullet}}$ D: By $\tilde{t} \longrightarrow_{\tau_{(j)}}^{(\odot\vec{\mathfrak{A}})\bullet} \tilde{s}$ (so $\tilde{t}, \tilde{s} \in \tilde{\mathbf{T}}^{\vec{\mathfrak{A}}}_{(j)}$), let $\tilde{t}'$ and $\tilde{s}'$ be such that $\tilde{t} = \tilde{t}'_{(j)}$ and $\tilde{s} = \tilde{s}'_{(j)}$. By $\tilde{t}'_{(j)} \longrightarrow_{\tau_{(j)}}^{(\odot\vec{\mathfrak{A}})\bullet} \tilde{s}'_{(j)}$, we have $\tilde{t}' \longrightarrow_{\tau}^{(\mathfrak{A}_j)\bullet} \tilde{s}'$ (Lemma 7.12). Thus, by the rule (D), this case has been proved.
- Case $\dfrac{\tau[\bullet/\langle l, r\rangle] \in \odot_{i=1}^{n} \mathsf{lsubSPR}^{(\mathfrak{A}_i)\bullet}}{\tau[x/\langle l, r\rangle] \in \odot_{i=1}^{n} \mathsf{lsubSPR}^{(\mathfrak{A}_i)\bullet}}$ L: By $\tilde{t} \longrightarrow_{\tau[x/\langle l,r\rangle]}^{(\odot\vec{\mathfrak{A}})\bullet} \tilde{s}$, let $\tilde{t}'$ and $\tilde{s}'$ be such that $\tilde{t} = \tilde{t}'[x/l]$ and $\tilde{s} = \tilde{s}'[x/r]$. Then we have $\tilde{t}'[\bullet/l] \longrightarrow_{\tau[\bullet/\langle l,r\rangle]}^{(\odot\vec{\mathfrak{A}})\bullet} \tilde{s}'[\bullet/r]$ (Lemma 7.13). Thus, by the rule (L) with IH, this case has been proved.
- Case $\dfrac{\tau \in \odot_{i=1}^{n} \mathsf{lsubSPR}^{(\mathfrak{A}_i)\bullet} \qquad \sigma \in \odot_{i=1}^{n} \mathsf{lsubSPR}^{(\mathfrak{A}_i)\bullet}}{\tau \diamond \sigma \in \odot_{i=1}^{n} \mathsf{lsubSPR}^{(\mathfrak{A}_i)\bullet}}$ T: By $\tilde{t} \longrightarrow_{\tau\diamond\sigma}^{(\odot\vec{\mathfrak{A}})\bullet} \tilde{s}$, there exists some lKL term $\tilde{u}$ such that $\tilde{t} \longrightarrow_{\tau}^{(\odot\vec{\mathfrak{A}})\bullet} \tilde{u}$ and $\tilde{u} \longrightarrow_{\sigma}^{(\odot\vec{\mathfrak{A}})\bullet} \tilde{s}$ (Lemma 7.14). Thus by the rule (T) with IH, this case has been proved.

Hence, this completes the proof. □

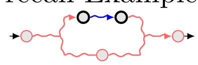
Proof of Theorem 5.5. For all $\tilde{t}, \tilde{s} \in \tilde{\mathbf{T}}^{\vec{\mathfrak{A}}}_{(j)}$, we have:

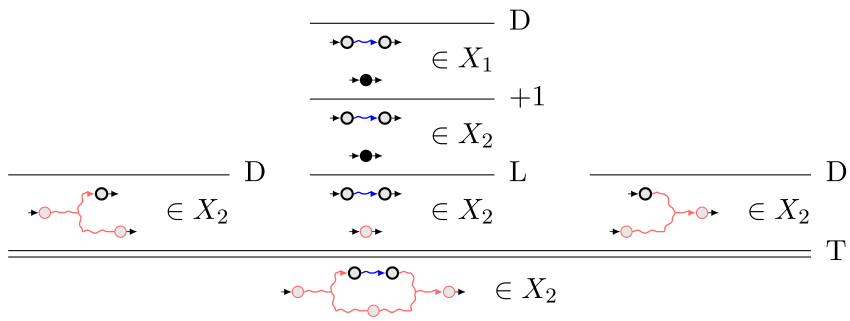
$$\begin{aligned}
\tilde{t} \longrightarrow^{(\odot\vec{\mathfrak{A}})\bullet} \tilde{s} &\iff \exists \tau \in \mathsf{lsubSPR}^{\vec{\mathfrak{A}}}_{(j)},\ \tilde{t} \longrightarrow_{\tau}^{(\odot\vec{\mathfrak{A}})\bullet} \tilde{s} && \text{(Proposition 7.5)}\\
&\iff \exists \tau \in \mathsf{lsubSPR}^{\vec{\mathfrak{A}}}_{(j)},\ \tilde{t}\ (\odot_{i=1}^{n} \longrightarrow^{(\mathfrak{A}_i)\bullet})_{\tau}\ \tilde{s} && \text{(Lemma 7.19)}\\
&\iff \tilde{t}\ (\odot_{i=1}^{n} \longrightarrow^{(\mathfrak{A}_i)\bullet})\ \tilde{s}. && \text{(Proposition 7.18)}
\end{aligned}$$

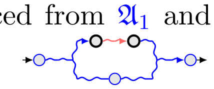
Hence, this completes the proof. □

7.4. **Incompleteness of the decomposition rules in** [Nak17]**.** In this subsection, we point out an error of the decomposition of $(\odot\vec{\mathfrak{A}})_\bullet$-runs in [Nak17, Section IV (Lemma IV.10)]. In [Nak17], left quotients of $(\odot\vec{\mathfrak{A}})_\bullet$-runs with respect to SGCPs are considered, but events of SGCPs are read in one-way. This approach essentially corresponds to the following decomposition rules of $(\odot\vec{\mathfrak{A}})_\bullet$-runs where $\vec{\mathfrak{A}} = \mathfrak{A}_1 \dots \mathfrak{A}_n$ and each X_j ($j \in [n]$) is the smallest subset of $\bigcup_{i=1}^{n} \mathsf{lsubSPR}^{\vec{\mathfrak{A}}}_{(i)}$ closed under the following rules (*cf.* Definition 7.6):

$$\frac{\tau \in \mathsf{lsubSPR}^{\mathfrak{A}_j}}{\tau_{(j)} \in X_j}\ \mathrm{D} \qquad \frac{\tau[\bullet/\langle l, r\rangle] \in X_j}{\tau[x/\langle l, r\rangle] \in X_j}\ \mathrm{L} \qquad \frac{\tau \in X_j \qquad \sigma \in X_j}{\tau \diamond \sigma \in X_j}\ \mathrm{T} \qquad \frac{\tau \in X_{j-1}}{\tau \in X_j}\ {+1}$$

(In a nutshell, the rule "-1" does not exist.) However, the set $\bigcup_{i=1}^{n} X_i$ is incomplete, as the direction $\Longleftarrow$ of Lemma 7.10 does not hold. We recall Example 7.7 and the sequence $\vec{\mathfrak{A}} = \mathfrak{A}_1\mathfrak{A}_2$. For instance, the $(\odot\vec{\mathfrak{A}})_\bullet$-run of the form  is derived by:



(Here, blue-colored edges and red-colored edges are the edges induced from $\mathfrak{A}_1$ and $\mathfrak{A}_2$, respectively.) However, we cannot derive the $(\odot\vec{\mathfrak{A}})_\bullet$-run τ of the form  in this system because the rule "-1" does not exist. To avoid this problem, instead of one-way automata, we use 2AFAs.

8. Conclusion

We have presented derivatives on structures/graphs and have shown a decomposition theorem of the derivatives (Theorem 5.5). Consequently, we have shown that the equational theory of the positive calculus of relations with transitive closure (PCoR*) is decidable and EXPSPACE-complete (Corollary 6.3).

Related and future work. In the following, we present related and future work.

Identity-free Kleene lattices. Brunet and Pous have shown that the equational theory of identity-free Kleene lattices $\{0, ;, +, \cap, \_^+\}$ is EXPSPACE-complete [BP15, BP17]. They presented an algorithm for comparing downward closed sets of runs (denoting acyclic and connected graphs) with respect to graph homomorphisms using Petri automata. However, their approach is problematic when terms have non-acyclic or non-connected graphs (*e.g.*, when the identity 1, the converse $\smile$, or the universality $\top$ occurs) [BP17, Section 9.3]. In our approach, based on the linearly bounded pathwidth model property (Proposition 2.9), we consider decomposing $\mathfrak{A}$-runs (instead of runs) on path decompositions. It would also be possible to connect our derivatives to branching automata [LW98, LW00, LW01] or Petri automata [BP17].

PDL with intersection. Propositional dynamic logic with intersection (IPDL) [Dan84a] is propositional dynamic logic (PDL) of regular programs with intersection. The theory of IPDL is decidable and 2EXPTIME-complete [Dan84a, LL05, GLL09]. The known algorithms [Dan84a, Lut05, GLL09] are based on the treewidth at most 2 model property of IPDL (*cf.* Proposition 2.8). In [GLL09], Göeller, Lohrey, and Lutz presented a polynomial-time reduction from the theory of IPDL with converse (ICPDL) into that over ω-regular trees and presented a reduction to the universality problem of two-way alternating parity tree automata. To express relational intersection ($\cap$), their approach is based on the product construction of automata (the product construction naively fails, but by adding transitions so that there always exists the shortest run if a run exists between two vertices on an input tree, the product construction works on trees [GLL09, p. 288]), while our approach uses branches of sub SP runs. Recently by extending their approach, the elementary decidability result of ICPDL has been extended to CPDL+ [FFP23]. Note that we can naturally reduce the equational theory of KL terms (with respect to relations) into the theory of IPDL formulas because $\mathsf{REL} \models t = s$ iff the IPDL formula $\langle t \rangle p \leftrightarrow \langle s \rangle p$ is valid (as with [FL79, p. 209] for PDL) where p is a propositional variable (disjoint from variables) and $\langle - \rangle$ denotes the diamond modality. Thus, the 2EXPTIME upper bound of the equational theory of KL terms can be obtained via this reduction. To obtain the EXPSPACE upper bound using this approach, we need to give a reduction from the theory of KL terms over linearly bounded pathwidth structures into that over words, *cf.* the inclusion problem is in PSPACE for (*word*) 2AFAs (Proposition 5.6) whereas the inclusion problem is EXPTIME-hard for *tree* (two-way alternating) automata [CDG+07, Theorem 1.7.7]. This approach would be possible by modifying the encoding of tree decompositions but is not immediate, because the reduction [Lut05, GLL09] always generates trees (not words) even when a given tree decomposition is a path decomposition. Conversely, it would also be possible to extend our decomposition approach to the extensions of PDL above, but we leave it as a future work.

Regular queries. Regular queries [RRV17] are binary non-recursive positive Datalog programs extended with Kleene star. The containment problem for regular queries is decidable and 2EXPSPACE-complete [RRV17]. Because we can naturally translate KL terms (with respect to relations) and also PCoR* terms into regular queries, we can also reduce the equational theory of PCoR* terms into the containment problem for regular queries.

The existential calculus of relations with transitive closure. The existential calculus of relations with transitive closure (ECoR*) [Nak23] is PCoR* with variable and constant complements (i.e., the complement operator only applies to variables or constants). ECoR* has the same expressive power as 3-variable existential first-order logic with variable-confined monodic transitive closure [Nak20, Nak22]. This extension is also related to boolean modal logic [GP90] and (I)PDL with negation of atomic programs [LW05, GLL09]. The equational theory of PCoR* with variable complements is Π^0_1-complete [Nak23] (an open problem left in the conference version [Nak17]) and the equational theory of PCoR* with constant complements (i.e., with the difference constant) is Π^0_1-complete [Nak24]. Nevertheless, the equational theory is decidable for the intersection-free fragment of ECoR* [Nak23].

On Kleene algebra with loop. The (graph) loop operator $t^{\circlearrowleft}$ [Dan84b] is the operator defined by: $t^{\circlearrowleft} \triangleq t \cap 1$. We leave it open whether the equational theory of Kleene algebras with loop (with respect to relations) is PSPACE-complete. Related to this, PDL with loop (loop-PDL) [Dan84b, GLL09] is EXPTIME-complete, whereas PDL with intersection is 2EXPSPACE-complete. For loop-PDL, the formulas can be translated into ICPDL formulas of intersection width at most 2, so the EXPTIME upper bound of loop-PDL is a corollary of the fact that the bounded intersection width fragment of ICPDL is in EXPTIME [GLL09, Corollary 4.9]. However, this translation essentially needs the diamond modality as a primitive, so Corollary 6.7 does not imply the PSPACE decidability, immediately; the closure size (with respect to the closure function of Definition 4.10) is still exponential because the number of labels occurring in Kleene algebra terms with loop is not bounded. A related problem is posed by Sedlár [Sed23, p. 16] for the complexity of the equational theory of Kleene algebra with the domain operator (dKA) with respect to relations. Here, the domain operator $\mathsf{d}(t)$ is the operator $\mathsf{d}(t) \triangleq (t\top) \cap \mathsf{1}$. Note that, because $\mathsf{REL} \models \mathsf{d}(t) = (t\top)^{\circlearrowleft}$, we can reduce the equational theory of dKA into that of Kleene algebras with loop and top.

On axiomatization. Another interesting question is the axiomatizability. Doumane and Pous presented a finite axiomatization for the equational theory of identity-free Kleene lattices [DP18]. However, the finite axiomatizability is still an open problem for the equational theory of Kleene lattices (with additional operators) with respect to relations [DP18, p. 15][Pou18].

Acknowledgment

We would like to thank anonymous referees for useful comments. This work was supported by JSPS KAKENHI Grant Numbers JP16J08119, JP21K13828, JP25K14985.

References

[AB95] Hajnal Andréka and D. A. Bredikhin. The equational theory of union-free algebras of relations. *Algebra Universalis*, 33(4):516–532, 1995. doi:10.1007/BF01225472.

[AMN11] Hajnal Andréka, Szabolcs Mikulás, and István Németi. The equational theory of Kleene lattices. *Theoretical Computer Science*, 412(52):7099–7108, 2011. doi:10.1016/J.TCS.2011.09.024.

[Ant96] Valentin Antimirov. Partial derivatives of regular expressions and finite automaton constructions. *Theoretical Computer Science*, 155(2):291–319, 1996. doi:10.1016/0304-3975(95)00182-4.

[AtC07] Carlos Areces and Balder ten Cate. Hybrid logics. In *Handbook of Modal Logic*, volume 3 of *Studies in Logic and Practical Reasoning*, pages 821–868. Elsevier, 2007. doi:10.1016/S1570-2464(07)80017-6.

[BBM+16] Rafaela Bastos, Sabine Broda, António Machiavelo, Nelma Moreira, and Rogério Reis. On the state complexity of partial derivative automata for regular expressions with intersection. In *DCFS*, volume 9777 of *LNTCS*, pages 45–59. Springer, 2016. doi:10.1007/978-3-319-41114-9_4.

[BÉS95] S. L. Bloom, Z. Ésik, and Gh. Stefanescu. Notes on equational theories of relations. *algebra universalis*, 33(1):98–126, 1995. doi:10.1007/BF01190768.

[BGK+16] Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Paweł Sobociński, and Fabio Zanasi. Rewriting modulo symmetric monoidal structure. In *LICS*, page 710–719. ACM, 2016. doi:10.1145/2933575.2935316.

[Bod98] Hans L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1-2):1–45, 1998. doi:10.1016/S0304-3975(97)00228-4.

[BP15] Paul Brunet and Damien Pous. Petri automata for Kleene allegories. In *LICS*, pages 68–79. IEEE, 2015. doi:10.1109/LICS.2015.17.

[BP16a] Mikołaj Bojańczyk and Michał Pilipczuk. Definability equals recognizability for graphs of bounded treewidth. In *LICS*, pages 407–416. ACM, 2016. doi:10.1145/2933575.2934508.

[BP16b] Paul Brunet and Damien Pous. Algorithms for Kleene algebra with converse. *Journal of Logical and Algebraic Methods in Programming*, 85(4):574–594, 2016. doi:10.1016/J.JLAMP.2015.07.005.

[BP17] Paul Brunet and Damien Pous. Petri automata. *Logical Methods in Computer Science*, 13(3), 2017. doi:10.23638/LMCS-13(3:33)2017.

[Bru17] Paul Brunet. Reversible Kleene lattices. In *MFCS*, volume 83 of *LIPIcs*, pages 66:1–66:14. Schloss Dagstuhl, 2017. doi:10.4230/LIPICS.MFCS.2017.66.

[Brz64] Janusz A. Brzozowski. Derivatives of regular expressions. *Journal of the ACM*, 11(4):481–494, 1964. doi:10.1145/321239.321249.

[CDG+07] Hubert Comon, Max Dauchet, Rémi Gilleron, Florent Jacquemard, Denis Lugiez, Christof Löding, Sophie Tison, and Marc Tommasi. *Tree Automata Techniques and Applications*. 2007. URL: https://inria.hal.science/hal-03367725.

[CDGLV02] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Moshe Y. Vardi. View-based query answering and query containment over semistructured data. In *DBPL*, volume 2397 of *LNCS*, pages 40–61. Springer, 2002. doi:10.1007/3-540-46093-4_3.

[CE12] Bruno Courcelle and Joost Engelfriet. *Graph Structure and Monadic Second-Order Logic*. Number 138 in Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2012. doi:10.1017/CBO9780511977619.

[CM77] Ashok K. Chandra and Philip M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In *STOC*, pages 77–90. ACM, 1977. doi:10.1145/800105.803397.

[Con71] John H. Conway. *Regular Algebra and Finite Machines*. Chapman and Hall, 1971.

[Cou88] Bruno Courcelle. The monadic second-order logic of graphs, II: Infinite graphs of bounded width. *Mathematical Systems Theory*, 21(1):187–221, 1988. doi:10.1007/BF02088013.

[Cou90] Bruno Courcelle. The monadic second-order logic of graphs. I. recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.

[Dan84a] Ryszard Danecki. Nondeterministic propositional dynamic logic with intersection is decidable. In *SCT*, volume 208 of *LNCS*, pages 34–53. Springer, 1984. doi:10.1007/3-540-16066-3_5.

[Dan84b] Ryszard Danecki. Propositional dynamic logic with strong loop predicate. In *MFCS*, volume 176 of *LNCS*, pages 573–581. Springer, 1984. doi:10.1007/BFb0030342.

[DP18] Amina Doumane and Damien Pous. Completeness for identity-free Kleene lattices. In *CONCUR*, volume 118 of *LIPIcs*, pages 18:1–18:17. Schloss Dagstuhl, 2018. doi:10.4230/LIPICS.CONCUR.2018.18.

[EF95] Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite Model Theory*. Springer Monographs in Mathematics. Springer, 2 edition, 1995. doi:10.1007/3-540-28788-4.

[Epp92] David Eppstein. Parallel recognition of series-parallel graphs. *Information and Computation*, 98(1):41–55, 1992. doi:10.1016/0890-5401(92)90041-D.

[FFP23] Diego Figueira, Santiago Figueira, and Edwin Pin. PDL on steroids: on expressive extensions of PDL with intersection and converse. In *LICS*, pages 1–13. IEEE, 2023. doi:10.1109/LICS56636.2023.10175813.

[FL79] Michael J. Fischer and Richard E. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18(2):194–211, 1979. doi:10.1016/0022-0000(79)90046-1.

[FS90] Peter J. Freyd and Andre Scedrov. *Categories, Allegories*, volume 39. North-Holland, 1990. doi:10.1016/S0924-6509(08)70049-7.

[Fü80] Martin Fürer. The complexity of the inequivalence problem for regular expressions with intersection. In *ICALP*, volume 85 of *LNCS*, pages 234–245. Springer, 1980. doi:10.1007/3-540-10003-2_74.

[GLL09] Stefan Göller, Markus Lohrey, and Carsten Lutz. PDL with intersection and converse: satisfiability and infinite-state model checking. *The Journal of Symbolic Logic*, 74(1):279–314, 2009. doi:10.2178/jsl/1231082313.

[GO14] Viliam Geffert and Alexander Okhotin. Transforming two-way alternating finite automata to one-way nondeterministic automata. In *MFCS*, volume 8634 of *LNTCS*, pages 291–302. Springer, 2014. doi:10.1007/978-3-662-44522-8_25.

[GP90] George Gargov and Solomon Passy. A note on boolean modal logic. In *Mathematical Logic*, pages 299–309. Springer, 1990. doi:10.1007/978-1-4613-0609-2_21.

[Hir18] Robin Hirsch. Decidability of equational theories for subsignatures of relation algebra. In *RAMICS*, volume 11194 of *LNCS*, pages 87–96. Springer, 2018. doi:10.1007/978-3-030-02149-8_6.

[Kle56] Stephen C. Kleene. Representation of events in nerve nets and finite automata. In *Automata Studies. (AM-34)*, pages 3–42. Princeton University Press, 1956. doi:10.1515/9781400882618-002.

[Koz91] Dexter Kozen. A completeness theorem for Kleene algebras and the algebra of regular events. In *LICS*, pages 214–225. IEEE, 1991. doi:10.1109/LICS.1991.151646.

[Koz97] Dexter Kozen. Kleene algebra with tests. *ACM Transactions on Programming Languages and Systems*, 19(3):427–443, 1997. doi:10.1145/256167.256195.

[KS96] Dexter Kozen and Frederick Smith. Kleene algebra with tests: Completeness and decidability. In *CSL*, volume 1258 of *LNCS*, pages 244–259. Springer, 1996. doi:10.1007/3-540-63172-0_43.

[KZ21] Christos Kapoutsis and Mohammad Zakzok. Alternation in two-way finite automata. *Theoretical Computer Science*, 870:75–102, 2021. doi:10.1016/j.tcs.2020.12.011.

[LL05] Martin Lange and Carsten Lutz. 2-ExpTime lower bounds for propositional dynamic logics with intersection. *The Journal of Symbolic Logic*, 70(04):1072–1086, 2005. doi:10.2178/jsl/1129642115.

[LLS78] Richard E. Ladner, Richard J. Lipton, and Larry J. Stockmeyer. Alternating pushdown automata. In *SFCS*, pages 92–106. IEEE, 1978. doi:10.1109/SFCS.1978.6.

[LLS84] Richard E. Ladner, Richard J. Lipton, and Larry J. Stockmeyer. Alternating pushdown and stack automata. *SIAM Journal on Computing*, 13(1):135–155, 1984. doi:10.1137/0213010.

[Lut05] Carsten Lutz. PDL with intersection and converse is decidable. In *CSL*, volume 3634 of *LNTCS*, pages 413–427. Springer, 2005. doi:10.1007/11538363_29.

[LW98] K. Lodaya and P. Weil. Series-parallel posets: Algebra, automata and languages. In *STACS*, volume 1373 of *LNCS*, pages 555–565. Springer, 1998. doi:10.1007/BFb0028590.

[LW00] K. Lodaya and P. Weil. Series–parallel languages and the bounded-width property. *Theoretical Computer Science*, 237(1):347–380, 2000. doi:10.1016/S0304-3975(00)00031-1.

[LW01] K. Lodaya and P. Weil. Rationality in algebras with a series operation. *Information and Computation*, 171(2):269–293, 2001. doi:10.1006/inco.2001.3077.

[LW05] Carsten Lutz and Dirk Walther. PDL with negation of atomic programs. *Journal of Applied Non-Classical Logics*, 15(2):189–213, 2005. doi:10.3166/jancl.15.189-213.

[Mae61] Shoji Maehara. Craig's interpolation theorem (in Japanese). *Sugaku*, 12(4):235–237, 1961. doi:10.11429/sugaku1947.12.235.

[Mak04] J. A. Makowsky. Algorithmic uses of the feferman–vaught theorem. *Annals of Pure and Applied Logic*, 126(1):159–213, 2004. doi:10.1016/j.apal.2003.11.002.

[MS72] A. R. Meyer and L. J. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *SWAT*, pages 125–129. IEEE, 1972. doi:10.1109/SWAT.1972.29.

[Nak17] Yoshiki Nakamura. Partial derivatives on graphs for Kleene allegories. In *LICS*, pages 1–12. IEEE, 2017. doi:10.1109/LICS.2017.8005132.

[Nak19] Yoshiki Nakamura. The undecidability of FO3 and the calculus of relations with just one binary relation. In *ICLA*, volume 11600 of *LNTCS*, pages 108–120. Springer, 2019. doi:10.1007/978-3-662-58771-3_11.

[Nak20] Yoshiki Nakamura. Expressive power and succinctness of the positive calculus of relations. In *RAMICS*, volume 12062 of *LNTCS*, pages 204–220. Springer, 2020. doi:10.1007/978-3-030-43520-2_13.

[Nak22] Yoshiki Nakamura. Expressive power and succinctness of the positive calculus of binary relations. *Journal of Logical and Algebraic Methods in Programming*, 127:100760, 2022. doi:10.1016/j.jlamp.2022.100760.

[Nak23] Yoshiki Nakamura. Existential calculi of relations with transitive closure: Complexity and edge saturations. In *LICS*, pages 1–13. IEEE, 2023. doi:10.1109/LICS56636.2023.10175811.

[Nak24] Yoshiki Nakamura. Undecidability of the positive calculus of relations with transitive closure and difference: Hypothesis elimination using graph loops. In *RAMICS*, volume 14787 of *LNTCS*, pages 207–224. Springer, 2024. doi:10.1007/978-3-031-68279-7_13.

[Nak25] Yoshiki Nakamura. Finite relational semantics for language Kleene algebra with complement. In *CSL*, volume 326 of *LIPIcs*, page 37:1–37:23. Schloss Dagstuhl, 2025. doi:10.4230/LIPIcs.CSL.2025.37.

[Ng84] Kan Ching Ng. *Relation algebras with transitive closure*. PhD thesis, University of California, 1984.

[Pou18] Damien Pous. On the positive calculus of relations with transitive closure. In *STACS*, volume 96 of *LIPIcs*, pages 3:1–3:16. Schloss Dagstuhl, 2018. doi:10.4230/LIPICS.STACS.2018.3.

[PV18] Damien Pous and Valeria Vignudelli. Allegories: decidability and graph homomorphisms. In *LICS*, pages 829–838. ACM, 2018. doi:10.1145/3209108.3209172.

[PW24] Damien Pous and Jana Wagemaker. Completeness theorems for kleene algebra with tests and top. *Logical Methods in Computer Science*, Volume 20, Issue 3, 2024. doi:10.46298/lmcs-20(3:27)2024.

[RRV17] Juan L. Reutter, Miguel Romero, and Moshe Y. Vardi. Regular queries on graph databases. *Theory of Computing Systems*, 61(1):31–83, 2017. doi:10.1007/s00224-016-9676-2.

[RS83] Neil Robertson and Paul D. Seymour. Graph minors. I. excluding a forest. *Journal of Combinatorial Theory, Series B*, 35(1):39–61, 1983. doi:10.1016/0095-8956(83)90079-5.

[RS86] Neil Robertson and Paul D. Seymour. Graph minors. II. algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3):309–322, 1986. doi:10.1016/0196-6774(86)90023-4.

[Sak09] Jacques Sakarovitch. *Elements of Automata Theory.* Cambridge University Press, 2009. doi:10.1017/CBO9781139195218.

[Sav70] Walter Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177–192, 1970. doi:10.1016/S0022-0000(70)80006-X.

[Sed23] Igor Sedlár. Kleene algebra with dynamic tests: Completeness and complexity, 2023. doi:10.48550/arXiv.2311.06937.

[Tar41] Alfred Tarski. On the calculus of relations. *The Journal of Symbolic Logic*, 6(3):73–89, 1941. doi:10.2307/2268577.

[TG87] Alfred Tarski and Steven Givant. *A Formalization of Set Theory without Variables*, volume 41. American Mathematical Society, 1987. doi:10.1090/coll/041.

[VTL79] Jacobo Valdes, Robert E. Tarjan, and Eugene L. Lawler. The recognition of series parallel digraphs. In *STOC*, page 1–12. ACM, 1979. doi:10.1145/800135.804393.

Appendix A. Proof of Proposition 5.9

Proof. Both directions are shown by easy induction on the derivative trees, respectively.

($\Longrightarrow$): By induction on the derivation tree.

- Case $\dfrac{\tilde{t} \longrightarrow^{(\mathfrak{A}_j)\bullet} \tilde{s}}{\tilde{t}_{(j')} \ (\odot_{i=1}^{n} \longrightarrow^{(\mathfrak{A}_i)\bullet}) \ \tilde{s}_{(j')}}$ D where j' satisfies $\tilde{t}_{(j')} = \tilde{t}_{(j)}$ and $\tilde{s}_{(j')} = \tilde{s}_{(j)}$: Then,

$$\dfrac{\dfrac{}{\langle\langle \tilde{t}, \tilde{s}\rangle_{\checkmark}, j'\rangle \in S^{\mathcal{A}_k^u}_{\triangleright\mathfrak{A}_1\ldots\mathfrak{A}_n\triangleleft}}\ \mathrm{D}}{\langle\langle \tilde{t}, \tilde{s}\rangle_{\checkmark}, j\rangle \in S^{\mathcal{A}_k^u}_{\triangleright\mathfrak{A}_1\ldots\mathfrak{A}_n\triangleleft}}\ -1,\ +1,\ \checkmark.$$

- Case $\dfrac{\tilde{t}_{(j')} \ (\odot_{i=1}^{n} \longrightarrow^{(\mathfrak{A}_i)\bullet}) \ \tilde{u}'_{(j')} \qquad \tilde{u}'_{(j')} \ (\odot_{i=1}^{n} \longrightarrow^{(\mathfrak{A}_i)\bullet}) \ \tilde{s}_{(j')}}{\tilde{t}_{(j')} \ (\odot_{i=1}^{n} \longrightarrow^{(\mathfrak{A}_i)\bullet}) \ \tilde{s}_{(j')}}$ T where j' satisfies $\tilde{t}_{(j')} = \tilde{t}_{(j)}$ and $\tilde{s}_{(j')} = \tilde{s}_{(j)}$: Then we have:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\langle\langle \tilde{t}, \tilde{u}'\rangle_{\checkmark}, j'\rangle \in S^{\mathcal{A}_k^u}_{\triangleright\mathfrak{A}_1\ldots\mathfrak{A}_n\triangleleft}}\ (\mathrm{IH})}{\langle\langle \tilde{t}, \tilde{u}'\rangle_{?}, j'\rangle \in S^{\mathcal{A}_k^u}_{\triangleright\mathfrak{A}_1\ldots\mathfrak{A}_n\triangleleft}}\ \checkmark \qquad \dfrac{\dfrac{}{\langle\langle \tilde{u}', \tilde{s}\rangle_{\checkmark}, j'\rangle \in S^{\mathcal{A}_k^u}_{\triangleright\mathfrak{A}_1\ldots\mathfrak{A}_n\triangleleft}}\ (\mathrm{IH})}{\langle\langle \tilde{u}', \tilde{s}\rangle_{?}, j'\rangle \in S^{\mathcal{A}_k^u}_{\triangleright\mathfrak{A}_1\ldots\mathfrak{A}_n\triangleleft}}\ \checkmark}{\langle\langle \tilde{t}, \tilde{s}\rangle_{\checkmark}, j'\rangle \in S^{\mathcal{A}_k^u}_{\triangleright\mathfrak{A}_1\ldots\mathfrak{A}_n\triangleleft}}\ \mathrm{T}}{\langle\langle \tilde{t}, \tilde{s}\rangle_{\checkmark}, j\rangle \in S^{\mathcal{A}_k^u}_{\triangleright\mathfrak{A}_1\ldots\mathfrak{A}_n\triangleleft}}\ -1,\ +1,\ \checkmark\ .$$

- Case $\dfrac{\tilde{t}[\bullet/l]_{(j')}\ (\odot_{i=1}^{n} \longrightarrow^{(\mathfrak{A}_i)\bullet})\ \tilde{s}[\bullet/r]_{(j')}}{\tilde{t}[x/l]_{(j')}\ (\odot_{i=1}^{n} \longrightarrow^{(\mathfrak{A}_i)\bullet})\ \tilde{s}[x/r]_{(j')}}$ L where j' satisfies that $\tilde{t}[x/l]_{(j')} = \tilde{t}[x/l]_{(j)}$ and $\tilde{s}[x/r]_{(j')} = \tilde{s}[x/r]_{(j)}$: Then we have:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\langle\langle \tilde{t}[\bullet/l], \tilde{s}[\bullet/r]\rangle_{\checkmark}, j'\rangle \in S^{\mathcal{A}_k^u}_{\triangleright\mathfrak{A}_1\ldots\mathfrak{A}_n\triangleleft}}\ \text{(IH)}}{\langle\langle \tilde{t}[\bullet/l], \tilde{s}[\bullet/r]\rangle_{?}, j'\rangle \in S^{\mathcal{A}_k^u}_{\triangleright\mathfrak{A}_1\ldots\mathfrak{A}_n\triangleleft}}\ \checkmark}{\langle\langle \tilde{t}[x/l], \tilde{s}[x/r]\rangle_{\checkmark}, j'\rangle \in S^{\mathcal{A}_k^u}_{\triangleright\mathfrak{A}_1\ldots\mathfrak{A}_n\triangleleft}}\ \text{L}}{\langle\langle \tilde{t}[x/l], \tilde{s}[x/r]\rangle_{\checkmark}, j\rangle \in S^{\mathcal{A}_k^u}_{\triangleright\mathfrak{A}_1\ldots\mathfrak{A}_n\triangleleft}}\ -1,\ +1,\ \checkmark \quad .$$

($\Longleftarrow$): By induction on the derivation tree.

- Case $\dfrac{}{\langle\langle \tilde{t}, \tilde{s}\rangle_{\checkmark}, j\rangle \in S^{\mathcal{A}_k^u}_{\triangleright\mathfrak{A}_1\ldots\mathfrak{A}_n\triangleleft}}$ D where $\tilde{t} \longrightarrow^{\mathfrak{A}} \tilde{s}$: Then, $\dfrac{\tilde{t} \longrightarrow^{(\mathfrak{A}_j)\bullet} \tilde{s}}{\tilde{t}_{(j)}\ (\odot_{i=1}^{n} \longrightarrow^{(\mathfrak{A}_i)\bullet})\ \tilde{s}_{(j)}}$ D.
- Case $\dfrac{\dfrac{\langle\langle \tilde{t}, \tilde{s}\rangle_{\checkmark}, j-1\rangle \in S^{\mathcal{A}_k^u}_{\triangleright\mathfrak{A}_1\ldots\mathfrak{A}_n\triangleleft}}{\langle\langle \tilde{t}, \tilde{s}\rangle_{?}, j-1\rangle \in S^{\mathcal{A}_k^u}_{\triangleright\mathfrak{A}_1\ldots\mathfrak{A}_n\triangleleft}}\ \checkmark}{\langle\langle \tilde{t}, \tilde{s}\rangle_{\checkmark}, j\rangle \in S^{\mathcal{A}_k^u}_{\triangleright\mathfrak{A}_1\ldots\mathfrak{A}_n\triangleleft}}\ -1$: (Note that $\checkmark$ is the only rule that can apply to $\langle \_, \_\rangle_{?}$.) We then have:

$$\dfrac{}{\tilde{t}_{(j-1)}\ (\odot_{i=1}^{n} \longrightarrow^{(\mathfrak{A}_i)\bullet})\ \tilde{s}_{(j-1)}}\ \text{(IH)}.$$

By $\overrightarrow{\mathsf{lab}}(\tilde{t}), \overrightarrow{\mathsf{lab}}(\tilde{s}) \in |(\mathfrak{A}_{j-1})_{\bullet}|^{+} \cap |(\mathfrak{A}_{j})_{\bullet}|^{+}$, we have $\tilde{t}_{(j-1)} = \tilde{t}_{(j)}$ and $\tilde{s}_{(j-1)} = \tilde{s}_{(j)}$, and thus this case has been proved.

- Case $\dfrac{\langle\langle \tilde{t}, \tilde{s}\rangle_{?}, j+1\rangle \in S^{\mathcal{A}_k^u}_{\triangleright\mathfrak{A}_1\ldots\mathfrak{A}_n\triangleleft}}{\langle\langle \tilde{t}, \tilde{s}\rangle_{\checkmark}, j\rangle \in S^{\mathcal{A}_k^u}_{\triangleright\mathfrak{A}_1\ldots\mathfrak{A}_n\triangleleft}}$ +1: In the same way as the case above.
- Case $\dfrac{\dfrac{\langle\langle \tilde{t}, \tilde{u}'\rangle_{\checkmark}, j\rangle \in S^{\mathcal{A}_k^u}_{\triangleright\mathfrak{A}_1\ldots\mathfrak{A}_n\triangleleft}}{\langle\langle \tilde{t}, \tilde{u}'\rangle_{?}, j\rangle \in S^{\mathcal{A}_k^u}_{\triangleright\mathfrak{A}_1\ldots\mathfrak{A}_n\triangleleft}}\ \checkmark \qquad \dfrac{\langle\langle \tilde{u}', \tilde{s}\rangle_{\checkmark}, j\rangle \in S^{\mathcal{A}_k^u}_{\triangleright\mathfrak{A}_1\ldots\mathfrak{A}_n\triangleleft}}{\langle\langle \tilde{u}', \tilde{s}\rangle_{?}, j\rangle \in S^{\mathcal{A}_k^u}_{\triangleright\mathfrak{A}_1\ldots\mathfrak{A}_n\triangleleft}}\ \checkmark}{\langle\langle \tilde{t}, \tilde{s}\rangle_{\checkmark}, j\rangle \in S^{\mathcal{A}_k^u}_{\triangleright\mathfrak{A}_1\ldots\mathfrak{A}_n\triangleleft}}\ \text{T}$: Then we have:

$$\dfrac{\dfrac{}{\tilde{t}_{(j)}\ (\odot_{i=1}^{n} \longrightarrow^{(\mathfrak{A}_i)\bullet})\ \tilde{u}'_{(j)}}\ \text{(IH)} \qquad \dfrac{}{\tilde{u}'_{(j)}\ (\odot_{i=1}^{n} \longrightarrow^{(\mathfrak{A}_i)\bullet})\ \tilde{s}_{(j)}}\ \text{(IH)}}{\tilde{t}_{(j)}\ (\odot_{i=1}^{n} \longrightarrow^{(\mathfrak{A}_i)\bullet})\ \tilde{s}_{(j)}}\ \text{T} \quad .$$

- Case $\dfrac{\dfrac{\langle\langle \tilde{t}[\bullet/l], \tilde{s}[\bullet/r]\rangle_{\checkmark}, j\rangle \in S^{\mathcal{A}^u_k}_{\triangleright \mathfrak{A}_1 \dots \mathfrak{A}_n \triangleleft}}{\langle\langle \tilde{t}[\bullet/l], \tilde{s}[\bullet/r]\rangle_{?}, j\rangle \in S^{\mathcal{A}^u_k}_{\triangleright \mathfrak{A}_1 \dots \mathfrak{A}_n \triangleleft}} \checkmark}{\langle\langle \tilde{t}[x/l], \tilde{s}[x/r]\rangle_{\checkmark}, j\rangle \in S^{\mathcal{A}^u_k}_{\triangleright \mathfrak{A}_1 \dots \mathfrak{A}_n \triangleleft}} \mathrm{L}$: Then, $\dfrac{\overline{\overline{\tilde{t}[\bullet/l]_{(j)}\ (\odot^n_{i=1} \longrightarrow^{(\mathfrak{A}_i)\bullet})\ \tilde{s}[\bullet/r]_{(j)}}}\ (\mathrm{IH})}{\tilde{t}[x/l]_{(j)}\ (\odot^n_{i=1} \longrightarrow^{(\mathfrak{A}_i)\bullet})\ \tilde{s}[x/r]_{(j)}} \mathrm{L}$.

Hence, this completes the proof. □

Appendix B. Proof of Proposition 7.1

For a run G, we let

$$\begin{aligned} \mathsf{subl}(G) &\triangleq \{G' \mid \text{there is a run } H \text{ such that } G = H \diamond G'\}, \\ \mathsf{subr}(G) &\triangleq \{G' \mid \text{there is a run } J \text{ such that } G = G' \diamond J\}, \\ \mathsf{sub}(G) &\triangleq \{G' \mid \text{there are runs } H \text{ and } J \text{ such that } G = H \diamond G' \diamond J\}. \end{aligned}$$

Proof of $\mathsf{subSPRl} = \mathsf{subSPRl}'$. ($\mathsf{subSPRl}' \subseteq \mathsf{subSPRl}$): By induction on the derivation tree of $G \in \mathsf{subSPRl}'$, it suffices to show: for every $G \in \mathsf{subSPRl}'$, there is some $G' \in \mathsf{subSPRr}'$ such that $G' \diamond G \in \mathsf{SPR}$. We distinguish the following cases:

- Case $G \in \mathsf{SPR}$: By letting $G' = 1^1$, we have $G' \diamond G = G \in \mathsf{SPR}$.
- Case $G = (G^{\mathrm{l}} \parallel H^{\mathrm{l}}) \diamond \mathtt{j}^1_1 \diamond J$ where $G^{\mathrm{l}}, H^{\mathrm{l}} \in \mathsf{subSPRl}'$ and $J \in \mathsf{SPR}$: Let $G^{\mathrm{r}}, H^{\mathrm{r}} \in \mathsf{subSPRr}'$ be the ones obtained by IH with respect to G^{l} and H^{l}, respectively. Then by letting $G' = \mathtt{f}^1_1 \diamond (G^{\mathrm{r}} \parallel H^{\mathrm{r}})$, we have $G' \diamond G = (\mathtt{f}^1_1 \diamond ((G^{\mathrm{r}} \diamond G^{\mathrm{l}}) \parallel (H^{\mathrm{r}} \diamond H^{\mathrm{l}})) \diamond \mathtt{j}^1_1) \diamond J \in \mathsf{SPR}$.

Hence, this completes the proof.

($\mathsf{subSPRl} \subseteq \mathsf{subSPRl}'$): By induction on the derivation tree of $G_0 \in \mathsf{SPR}$, we show: for every $G_0 \in \mathsf{SPR}$ and $G \in \mathsf{subl}(G_0)$, we have $G \in \mathsf{subSPRl}'$. We distinguish the following cases:

- Case $G_0 = 1^1$, Case $G_0 = a^1_1$: Clear by $\mathsf{subl}(G_0) \subseteq \mathsf{SPR} \subseteq \mathsf{subSPRl}'$.
- Case $G_0 = H_0 \diamond J_0$ where $H_0, J_0 \in \mathsf{SPR}$: We distinguish the following cases:
  - Case $G = H \diamond J_0$ where $H \in \mathsf{subl}(H_0)$: By IH, we have $H \in \mathsf{subSPRl}'$. Then (1) $H = J'$ or (2) $H = (G^{\mathrm{l}} \parallel H^{\mathrm{l}}) \diamond \mathtt{j}^1_1 \diamond J'$ holds where $G^{\mathrm{l}}, H^{\mathrm{l}} \in \mathsf{subSPRl}'$, $J' \in \mathsf{SPR}$. For (1), we have $H \diamond J_0 = J' \diamond J_0 \in \mathsf{SPR} \subseteq \mathsf{subSPRl}'$. For (2), we have $H \diamond J_0 = (G^{\mathrm{l}} \parallel H^{\mathrm{l}}) \diamond \mathtt{j}^1_1 \diamond (J' \diamond J_0) \in \mathsf{subSPRl}'$. Hence, in both cases, we have $H \diamond J_0 \in \mathsf{subSPRl}'$.
  - Case $G \in \mathsf{subl}(J_0)$: By IH.
- Case $G_0 = \mathtt{f}^1_1 \diamond (H_0 \parallel J_0) \diamond \mathtt{j}^1_1$ where $H_0, J_0 \in \mathsf{SPR}$: We distinguish the following cases:
  - Case $G = G_0$: Clear by $G_0 \in \mathsf{SPR} \subseteq \mathsf{subSPRl}'$.
  - Case $G = (H \parallel J) \diamond \mathtt{j}^1_1$ where $H \in \mathsf{subl}(H_0)$ and $J \in \mathsf{subl}(J_0)$: By IH.
  - Case $G = 1^1$: Clear.

Hence, this completes the proof. □

Proof of $\mathsf{subSPRr} = \mathsf{subSPRr}'$. Similar to the proof of $\mathsf{subSPRl} = \mathsf{subSPRl}'$. □

Proof of $\mathsf{subSPR} = \mathsf{subSPR}'$. ($\mathsf{subSPR}' \subseteq \mathsf{subSPR}$): By the proof of ($\mathsf{subSPRl}' \subseteq \mathsf{subSPRl}$), we have:

- for every $G \in \mathsf{subSPRl}'$, there is some $G' \in \mathsf{subSPRr}'$ such that $G' \diamond G \in \mathsf{SPR}$.

Similarly, we have:

- for every $G \in \mathsf{subSPRr}'$, there is some $G' \in \mathsf{subSPRl}'$ such that $G \diamond G' \in \mathsf{SPR}$.

By induction on the derivation tree of $G \in \mathsf{subSPR}'$, we show: for every $G \in \mathsf{subSPR}'$, there are some $H \in \mathsf{subSPRl}'$ and $J \in \mathsf{subSPRr}'$ such that $H \diamond G \diamond J \in \mathsf{SPR}$.

- Case $G = H^{\mathrm{l}} \diamond J^{\mathrm{r}}$ where $H^{\mathrm{l}} \in \mathsf{subSPRl}'$ and $J^{\mathrm{r}} \in \mathsf{subSPRr}'$: Let $H^{\mathrm{r}} \in \mathsf{subSPRr}'$ be such that $H^{\mathrm{r}} \diamond H^{\mathrm{l}} \in \mathsf{SPR}$ and let $J^{\mathrm{l}} \in \mathsf{subSPRl}$ be such that $J^{\mathrm{r}} \diamond J^{\mathrm{l}} \in \mathsf{SPR}$. We then have $H^{\mathrm{r}} \diamond (H^{\mathrm{l}} \diamond J^{\mathrm{r}}) \diamond J^{\mathrm{l}} \in \mathsf{SPR}$.
- Case $G = (G_1 \parallel G_2)$ where $G_1, G_2 \in \mathsf{subSPR}'$: By IH, there are some $H_k \in \mathsf{subSPRr}'$, $J_k \in \mathsf{subSPRl}'$ such that $H_k \diamond G_k \diamond J_k \in \mathsf{SPR}$ for each $k \in [1, 2]$. We then have

$$\begin{aligned}&(\mathtt{f}_1^1 \diamond (H_1 \parallel H_2)) \diamond (G_1 \parallel G_2) \diamond ((J_1 \parallel J_2) \diamond \mathtt{j}_1^1)\\ &= \mathtt{f}_1^1 \diamond ((H_1 \diamond G_1 \diamond J_1) \parallel (H_2 \diamond G_2 \diamond J_2)) \diamond \mathtt{j}_1^1 \in \mathsf{SPR}.\end{aligned}$$

Hence, this completes the proof.

($\mathsf{subSPR} \subseteq \mathsf{subSPR}'$): By induction on the derivation tree of $G_0 \in \mathsf{SPR}$, we show: for every $G_0 \in \mathsf{SPR}$ and $G \in \mathsf{sub}(G_0)$, we have $G \in \mathsf{subSPR}'$. We distinguish the following cases:

- Case $G_0 = 1^1$, Case $G_0 = a_1^1$: Clear by $\mathsf{sub}(G_0) \subseteq \mathsf{SPR} \subseteq \mathsf{subSPR}'$.
- Case $G_0 = H_0 \diamond J_0$ where $H_0, J_0 \in \mathsf{SPR}$: We distinguish the following cases:
  - Case $G \in \mathsf{sub}(H_0)$ or $G \in \mathsf{sub}(J_0)$: By IH.
  - Case $G = H \diamond J$ where $H \in \mathsf{subl}(H_0)$ and $J \in \mathsf{subr}(J_0)$. From the above, we also have $H \in \mathsf{subSPRl}'$, $J \in \mathsf{subSPRr}'$. Thus, $H \diamond J \in \mathsf{subSPR}'$.
- Case $G_0 = \mathtt{f}_1^1 \diamond (H_0 \parallel J_0) \diamond \mathtt{j}_1^1$ where $H_0, J_0 \in \mathsf{SPR}$: We distinguish the following cases:
  - Case $G = G_0$: Clear by $G_0 \in \mathsf{SPR} \subseteq \mathsf{subSPR}'$.
  - Case $G = (H \parallel J) \diamond \mathtt{j}_1^1$ where $H \in \mathsf{subl}(H_0)$ and $J \in \mathsf{subl}(J_0)$: From the above, we also have $H \in \mathsf{subSPRl}'$, $J \in \mathsf{subSPRl}'$. Thus, $G \in \mathsf{subSPRl}' \subseteq \mathsf{subSPR}'$.
  - Case $G = \mathtt{f}_1^1 \diamond (H \parallel J)$ where $H \in \mathsf{subr}(H_0)$ and $J \in \mathsf{subr}(J_0)$: Similarly.
  - Case $G = (H \parallel J)$ where $H \in \mathsf{sub}(H_0)$ and $J \in \mathsf{sub}(J_0)$: By IH, $H, J \in \mathsf{subSPR}'$. We thus have $G \in \mathsf{subSPR}'$.
  - Case $G = 1^1$: Clear.

Hence, this completes the proof. ∎

Appendix C. Proof of Proposition 7.16

Proof. ($\supseteq$): We show that each rule of $\rightsquigarrow_\tau$ is admissible in $\longrightarrow_\tau$.

- Case $\dfrac{\langle x, y\rangle \in a^{\mathfrak{A}}}{@x.a \longrightarrow^{\mathfrak{A}}_{a_1^{x,y}} @y.1}$ for $a \in \Sigma$, Case $\dfrac{@x.t \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{t}'}{@x.t + s \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{t}'}$, Case $\dfrac{@x.s \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{s}'}{@x.t + s \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{s}'}$, Case $\dfrac{}{\tilde{t} \longrightarrow^{\mathfrak{A}}_{1^{\overrightarrow{\mathsf{lab}}(\tilde{t})}} \tilde{t}}$, Case $\dfrac{@x.t \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{t}'}{@x.t \mathbin{;} s \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{t}' \mathbin{;_1} s}$, Case $\dfrac{\tilde{t} \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{s}}{\tilde{t} \mathbin{;_1} u \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{s} \mathbin{;_1} u}$: Trivial.
- Case $\dfrac{@x.t \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{t}' \quad \mathsf{E}_z(\tilde{t}') \quad @z.s \longrightarrow^{\mathfrak{A}}_{\rho'} \tilde{s}'}{@x.t \mathbin{;} s \longrightarrow^{\mathfrak{A}}_{\rho \diamond \rho'} \tilde{s}'}$: By $\dfrac{@x.t \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{t}' \qquad \dfrac{\mathsf{E}_z(\tilde{t}') \quad @z.s \longrightarrow^{\mathfrak{A}}_{\rho'} \tilde{s}'}{\tilde{t}' \longrightarrow^{\mathfrak{A}}_{\rho'} \tilde{s}'}}{@x.t \mathbin{;} s \longrightarrow^{\mathfrak{A}}_{\rho \diamond \rho'} \tilde{s}'}\ \mathrm{T}$.
- Case $\dfrac{\tilde{t} \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{t}' \quad \mathsf{E}_z(\tilde{t}') \quad @z.s \longrightarrow^{\mathfrak{A}}_{\rho'} \tilde{s}}{\tilde{t} \mathbin{;_1} s \longrightarrow^{\mathfrak{A}}_{\rho \diamond \rho'} \tilde{s}}$: Similar to the case above.

- Case $\dfrac{@x_0.t \longrightarrow^{\mathfrak{A}}_{\rho_0} \tilde{t}'_1 \quad \mathsf{E}_{x_1}(\tilde{t}'_1) \quad \dots \quad @x_{n-1}.t \longrightarrow^{\mathfrak{A}}_{\rho_{n-1}} \tilde{t}'_n \quad \mathsf{E}_{x_n}(\tilde{t}'_n) \quad @x_n.t \longrightarrow^{\mathfrak{A}}_{\rho_n} \tilde{t}'_{n+1}}{@x_0.t^* \longrightarrow^{\mathfrak{A}}_{\rho_0 \diamond \dots \diamond \rho_n} \tilde{t}'_{n+1} ;_1 t^*}$:

  By the following derivation tree:

$$\dfrac{\dfrac{@x_0.t \longrightarrow^{\mathfrak{A}}_{\rho_0} \tilde{t}'_1}{@x_0.t^* \longrightarrow^{\mathfrak{A}}_{\rho_0} \tilde{t}'_1 ;_1 t^*} \quad \dots \quad \dfrac{\mathsf{E}_{x_n}(\tilde{t}'_n) \quad \dfrac{@x_n.t \longrightarrow^{\mathfrak{A}}_{\rho_n} \tilde{t}'_{n+1}}{@x_n.t^* \longrightarrow^{\mathfrak{A}}_{\rho_n} \tilde{t}'_{n+1} ;_1 t^*}}{\tilde{t}'_n ;_1 t^* \longrightarrow^{\mathfrak{A}}_{\rho_n} \tilde{t}'_{n+1} ;_1 t^*}}{@x_0.t^* \longrightarrow^{\mathfrak{A}}_{\rho_0 \diamond \dots \diamond \rho_n} \tilde{t}'_{n+1} ;_1 t^*}\ \mathrm{T}$$

- Case $\dfrac{@x.t \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{t}' \quad \mathsf{E}_z(\tilde{t}') \qquad @x.s \longrightarrow^{\mathfrak{A}}_{\rho'} \tilde{s}' \quad \mathsf{E}_z(\tilde{s}')}{@x.t \cap s \longrightarrow^{\mathfrak{A}}_{\mathtt{f}^x_1 \diamond (\rho \| \rho') \diamond \mathtt{j}^z_1} @z.1}$: We have:

$$\dfrac{}{@x.t \cap s \longrightarrow^{\mathfrak{A}}_{\mathtt{f}^x_1} (@x.t) \cap_1 (@x.s)}, \qquad \dfrac{@x.t \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{t}'}{(@x.t) \cap_1 (@x.s) \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{t}' \cap_1 (@x.s)},$$

$$\dfrac{@x.s \longrightarrow^{\mathfrak{A}}_{\rho'} \tilde{s}'}{\tilde{t}' \cap_1 (@x.s) \longrightarrow^{\mathfrak{A}}_{\rho'} \tilde{t}' \cap_1 \tilde{s}'}, \text{ and } \qquad \dfrac{\mathsf{E}_z(\tilde{t}') \quad \mathsf{E}_z(\tilde{s}')}{\tilde{t}' \cap_1 \tilde{s}' \longrightarrow^{\mathfrak{A}}_{\mathtt{j}^z_1} @z.1}.$$

  Hence by (T), we have obtained $@x.t \cap s \longrightarrow^{\mathfrak{A}}_{\mathtt{f}^x_1 \diamond (\rho \| \rho') \diamond \mathtt{j}^z_1} @z.1$.

- Case $\dfrac{@x.t \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{t}' \qquad @x.s \longrightarrow^{\mathfrak{A}}_{\rho'} \tilde{s}'}{@x.t \cap s \longrightarrow^{\mathfrak{A}}_{\mathtt{f}^x_1 \diamond (\rho \| \rho')} \tilde{t} \cap_1 \tilde{s}}$, Case $\dfrac{\tilde{t} \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{t}' \qquad \mathsf{E}_z(\tilde{t}') \qquad \tilde{s} \longrightarrow^{\mathfrak{A}}_{\rho'} \tilde{s}' \qquad \mathsf{E}_z(\tilde{s}')}{\tilde{t} \cap_1 \tilde{s} \longrightarrow^{\mathfrak{A}}_{(\rho \| \rho') \diamond \mathtt{j}^z_1} @z.1}$,

  Case $\dfrac{\tilde{t} \longrightarrow^{\mathfrak{A}}_{\rho} \tilde{t}' \qquad \tilde{s} \longrightarrow^{\mathfrak{A}}_{\rho'} \tilde{s}'}{\tilde{t} \cap_1 \tilde{s} \longrightarrow^{\mathfrak{A}}_{(\rho \| \rho')} \tilde{t}' \cap_1 \tilde{s}'}$: Similar to the case above.

Hence, this direction has been proved.

($\subseteq$): We show that the transitivity rule $\dfrac{\tilde{t} \rightsquigarrow^{\mathfrak{A}}_{\tau} \tilde{u} \qquad \tilde{u} \rightsquigarrow^{\mathfrak{A}}_{\sigma} \tilde{s}}{\tilde{t} \rightsquigarrow^{\mathfrak{A}}_{\tau \diamond \sigma} \tilde{s}}$ T is admissible, by induction on the sum of the sizes of the derivation trees of $\tilde{t} \rightsquigarrow^{\mathfrak{A}}_{\tau} \tilde{u}$ and $\tilde{u} \rightsquigarrow^{\mathfrak{A}}_{\sigma} \tilde{s}$.

- Case $\dfrac{\tilde{t} \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{s} \qquad \dfrac{}{\tilde{s} \rightsquigarrow^{\mathfrak{A}}_{1^{\overrightarrow{\mathsf{lab}}(\tilde{s})}} \tilde{s}}\ \mathrm{R}}{\tilde{t} \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{s}}\ \mathrm{T}$, Case $\dfrac{\dfrac{}{\tilde{t} \rightsquigarrow^{\mathfrak{A}}_{1^{\overrightarrow{\mathsf{lab}}(\tilde{t})}} \tilde{t}}\ \mathrm{R} \qquad \tilde{t} \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{s}}{\tilde{t} \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{s}}\ \mathrm{T}$ (i.e., when the immediate above rule contains (R)): Clear.

- Case $\dfrac{\dfrac{}{@x.a \rightsquigarrow^{\mathfrak{A}}_{\rho} @y.1} \qquad @y.1 \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{s}}{@x.a \rightsquigarrow^{\mathfrak{A}}_{\rho \diamond \rho'} \tilde{s}}\ \mathrm{T}$: Then the right-hand side rule should be the rule (R); thus, this case has already been proved.

- Case $\dfrac{\dfrac{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}''}{@x.t \,;\, s \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}'' \,;_1 s} \qquad \tilde{t}'' \,;_1 s \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{u}}{@x.t \,;\, s \rightsquigarrow^{\mathfrak{A}}_{\rho\diamond\rho'} \tilde{u}}\,\mathrm{T}$: For the form of the above of $\tilde{t}'' \,;_1 s \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{u}$, we distinguish the following cases (except for the rule (R)):
  - Case $\dfrac{\tilde{t}'' \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{t}'}{\tilde{t}'' \,;_1 s \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{t}' \,;_1 s}$: Then we have: $\dfrac{\dfrac{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}'' \qquad \tilde{t}'' \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{t}'}{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho\diamond\rho'} \tilde{t}'}\,\mathrm{T}}{@x.t \,;\, s \rightsquigarrow^{\mathfrak{A}}_{\rho\diamond\rho'} \tilde{t}' \,;_1 s}$. By IH, this case has been proved.
  - Case $\dfrac{\tilde{t}'' \rightsquigarrow^{\mathfrak{A}}_{\rho'_1} \tilde{t}' \qquad \mathsf{E}_z(\tilde{t}') \qquad @z.s \rightsquigarrow^{\mathfrak{A}}_{\rho'_2} \tilde{u}}{\tilde{t}'' \,;_1 s \rightsquigarrow^{\mathfrak{A}}_{\rho'_1\diamond\rho'_2} \tilde{u}}$: Then we have:
  $\dfrac{\dfrac{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}'' \qquad \tilde{t} \rightsquigarrow^{\mathfrak{A}}_{\rho'_1} \tilde{t}'}{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho\diamond\rho'_1} \tilde{t}'}\,\mathrm{T} \qquad \mathsf{E}_z(\tilde{t}') \qquad @z.s \rightsquigarrow^{\mathfrak{A}}_{\rho'_2} \tilde{u}}{@x.t \,;\, s \rightsquigarrow^{\mathfrak{A}}_{(\rho\diamond\rho'_1)\diamond\rho'_2} \tilde{u}}$. By IH, this case has been proved.
- Case $\dfrac{\dfrac{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}' \qquad \mathsf{E}_z(\tilde{t}') \qquad @z.s \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{s}''}{@x.t \,;\, s \rightsquigarrow^{\mathfrak{A}}_{\rho\diamond\rho'} \tilde{s}''} \qquad \tilde{s}'' \rightsquigarrow^{\mathfrak{A}}_{\rho''} \tilde{s}'}{@x.t \,;\, s \rightsquigarrow^{\mathfrak{A}}_{(\rho\diamond\rho')\diamond\rho''} \tilde{s}'}\,\mathrm{T}$: Then we have:
$\dfrac{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}' \quad \mathsf{E}_z(\tilde{t}') \quad \dfrac{@z.s \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{s}'' \quad \tilde{s}'' \rightsquigarrow^{\mathfrak{A}}_{\rho''} \tilde{s}'}{@z.s \rightsquigarrow^{\mathfrak{A}}_{\rho'\diamond\rho''} \tilde{s}'}\,\mathrm{T}}{@x.t \,;\, s \rightsquigarrow^{\mathfrak{A}}_{\rho;(\rho'\diamond\rho'')} \tilde{s}'}$. By IH, this case has been proved.
- Case $\dfrac{\dfrac{\tilde{t} \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}'}{\tilde{t} \,;_1 s \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}' \,;_1 s} \quad \tilde{t}' \,;_1 s \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{s}'}{\tilde{t} \rightsquigarrow^{\mathfrak{A}}_{\rho\diamond\rho'} \tilde{s}'}\,\mathrm{T}$, Case $\dfrac{\dfrac{\tilde{t} \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}' \quad \mathsf{E}_z(\tilde{t}') \quad @z.s \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{s}''}{\tilde{t} \,;_1 s \rightsquigarrow^{\mathfrak{A}}_{\rho\diamond\rho'} \tilde{s}''} \qquad \tilde{s}'' \rightsquigarrow^{\mathfrak{A}}_{\rho''} \tilde{s}'}{\tilde{t} \,;_1 s \rightsquigarrow^{\mathfrak{A}}_{(\rho\diamond\rho')\diamond\rho''} \tilde{s}'}\,\mathrm{T}$:
Similar to the two cases above.
- Case
$\dfrac{\dfrac{@x_0.t \rightsquigarrow^{\mathfrak{A}}_{\rho_0} \tilde{t}'_1 \qquad \mathsf{E}_{x_1}(\tilde{t}'_1) \qquad \dots \qquad \mathsf{E}_{x_n}(\tilde{t}'_n) \qquad @x_n.t \rightsquigarrow^{\mathfrak{A}}_{\rho'_n} \tilde{t}''_{n+1}}{@x_0.t^* \rightsquigarrow^{\mathfrak{A}}_{\rho_0\diamond\ldots\diamond\rho_{n-1}\diamond\rho'_n} \tilde{t}''_{n+1} \,;_1 t^*} \qquad \tilde{t}''_{n+1} \,;_1 t^* \rightsquigarrow^{\mathfrak{A}}_{\rho''} \tilde{u}}{@x_0.t^* \rightsquigarrow^{\mathfrak{A}}_{(\rho_0\diamond\ldots\diamond\rho_{n-1}\diamond\rho'_n)\diamond\rho''} \tilde{u}}\,\mathrm{T}$:

  For the form of the above of $\tilde{t}'_{n+1} \,;_1 t^* \rightsquigarrow^{\mathfrak{A}}_{\rho''} \tilde{u}$, we distinguish the following cases (except for the rule (R)):
  - Case $\dfrac{\tilde{t}''_{n+1} \rightsquigarrow^{\mathfrak{A}}_{\rho''} \tilde{t}'_{n+1}}{\tilde{t}''_{n+1} \,;_1 t^* \rightsquigarrow^{\mathfrak{A}}_{\rho''} \tilde{t}'_{n+1} \,;_1 t^*}$: Then we have:

$$\dfrac{@x_0.t \rightsquigarrow^{\mathfrak{A}}_{\rho_0} \tilde{t}'_1 \quad \mathsf{E}_{x_1}(\tilde{t}'_1) \quad \ldots \quad \mathsf{E}_{x_n}(\tilde{t}'_n) \quad \dfrac{@x_n.t \rightsquigarrow^{\mathfrak{A}}_{\rho'_n} \tilde{t}''_{n+1} \quad \tilde{t}''_{n+1} \rightsquigarrow^{\mathfrak{A}}_{\rho''} \tilde{t}'_{n+1}}{@x_n.t \rightsquigarrow^{\mathfrak{A}}_{\rho'_n \diamond \rho''} \tilde{t}'_{n+1}}\,\mathrm{T}}{@x_0.t^* \rightsquigarrow^{\mathfrak{A}}_{\rho_0 \diamond \ldots \diamond (\rho'_n \diamond \rho'')} \tilde{t}'_{n+1} \,;_1\, t^*}.$$

Thus by IH, this case has been proved.

– Case $\dfrac{\tilde{t}''_{n+1} \rightsquigarrow^{\mathfrak{A}}_{\rho''_1} \tilde{t}'_{n+1} \quad \mathsf{E}_{x_{n+1}}(\tilde{t}'_{n+1}) \quad @x_{n+1}.t^* \rightsquigarrow^{\mathfrak{A}}_{\rho''_2} \tilde{u}}{\tilde{t}'_{n+1} \,;_1\, t^* \rightsquigarrow^{\mathfrak{A}}_{\rho''_1 \diamond \rho''_2} \tilde{u}}$: For the form of the above of $@x_{n+1}.t^* \rightsquigarrow^{\mathfrak{A}}_{\rho''_2} \tilde{u}$, we distinguish the following cases:

∗ Case $\dfrac{}{@x_{n+1}.t^* \rightsquigarrow^{\mathfrak{A}}_{1^{x_{n+1}}} @x_{n+1}.t^*}\,\mathrm{R}$ (then, $\rho'' = \rho''_1$): Then we have:

$$\dfrac{@x_0.t \rightsquigarrow^{\mathfrak{A}}_{\rho_0} \tilde{t}'_1 \quad \mathsf{E}_{x_1}(\tilde{t}'_1) \ \ldots \quad \dfrac{@x_n.t \rightsquigarrow^{\mathfrak{A}}_{\rho'_n} \tilde{t}''_{n+1} \quad \tilde{t}''_{n+1} \rightsquigarrow^{\mathfrak{A}}_{\rho''} \tilde{t}'_{n+1}}{@x_n.t \rightsquigarrow^{\mathfrak{A}}_{\rho'_n \diamond \rho''} \tilde{t}'_{n+1}}\,\mathrm{T} \quad \mathsf{E}_{x_{n+1}}(\tilde{t}'_{n+1})}{@x_0.t^* \rightsquigarrow^{\mathfrak{A}}_{\rho_0 \diamond \ldots \diamond \rho_{n-1} \diamond \rho'_n \diamond \rho''} @x_{n+1}.t^*}.$$

Thus by IH, this case has been proved.

∗ Case
$$\dfrac{@x_{n+1}.t \rightsquigarrow^{\mathfrak{A}}_{\rho_{n+1}} \tilde{t}'_{n+2} \quad \mathsf{E}_{x_{n+2}}(\tilde{t}'_{n+2}) \quad \ldots \quad \mathsf{E}_{x_{m+1}}(\tilde{t}'_{m+1}) \quad @x_{m+1}.t \rightsquigarrow^{\mathfrak{A}}_{\rho_{m+1}} \tilde{t}'_{m+2}}{@x_{n+1}.t^* \rightsquigarrow^{\mathfrak{A}}_{\rho_{n+1} \diamond \ldots \diamond \rho_{m+1}} \tilde{t}'_{m+2} \,;_1\, t^*}:$$

Then by letting $\rho_n = \rho'_n \diamond \rho''_1$, we have:

$$\dfrac{@x_0.t \rightsquigarrow^{\mathfrak{A}}_{\rho_0} \tilde{t}'_1 \quad \mathsf{E}_{x_1}(\tilde{t}'_1) \quad \ldots \quad \ldots \quad \mathsf{E}_{x_{m+1}}(\tilde{t}'_{m+1}) \quad @x_{m+1}.t \rightsquigarrow^{\mathfrak{A}}_{\rho_{m+1}} \tilde{t}'_{m+2}}{@x_0.t^* \rightsquigarrow^{\mathfrak{A}}_{\rho_0 \diamond \ldots \diamond \rho_{m+1}} \tilde{t}'_{m+2} \,;_1\, t^*}.$$

Thus by IH, this case has been proved.

- Case $\dfrac{\dfrac{@x_0.t \rightsquigarrow^{\mathfrak{A}}_{\rho_0} \tilde{t}'_1 \ \mathsf{E}_{x_1}(\tilde{t}'_1) \ \ldots \ \mathsf{E}_{x_n}(\tilde{t}'_n)}{@x_0.t^* \rightsquigarrow^{\mathfrak{A}}_{\rho_0 \diamond \ldots \diamond \rho_{n-1}} @x_n t^*} \quad @x_n t^* \,;_1\, t^* \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{u}}{@x_0.t^* \rightsquigarrow^{\mathfrak{A}}_{\rho_0 \diamond \ldots \diamond \rho_{n-1}} \tilde{u}}\,\mathrm{T}$: Similar to the case above.
- Case $\dfrac{\dfrac{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}'}{@x.t + s \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}'} \quad \tilde{t}' \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{u}}{@x.t + s \rightsquigarrow^{\mathfrak{A}}_{\rho \diamond \rho'} \tilde{u}}\,\mathrm{T}$: We then have: $\dfrac{\dfrac{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{t}' \quad \tilde{t}' \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{u}}{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho \diamond \rho'} \tilde{u}}\,\mathrm{T}}{@x.t + s \rightsquigarrow^{\mathfrak{A}}_{\rho \diamond \rho'} \tilde{u}}$. Thus by IH, this case has been proved.
- Case $\dfrac{\dfrac{@x.s \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{s}'}{@x.t + s \rightsquigarrow^{\mathfrak{A}}_{\rho} \tilde{s}'} \quad \tilde{s}' \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{u}}{@x.t + s \rightsquigarrow^{\mathfrak{A}}_{\rho \diamond \rho'} \tilde{u}}\,\mathrm{T}$: Similar to the case above.
- Case $\dfrac{\dfrac{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho_1} \tilde{t}' \quad \mathsf{E}_z(\tilde{t}') \quad @x.s \rightsquigarrow^{\mathfrak{A}}_{\rho_2} \tilde{s}' \quad \mathsf{E}_z(\tilde{s}')}{@x.t \cap s \rightsquigarrow^{\mathfrak{A}}_{\mathtt{f}^x_1 \diamond (\rho_1 \| \rho_2) \diamond \mathtt{j}^z_1} @z.1} \quad @z.1 \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde{u}}{@x.t \cap s \rightsquigarrow^{\mathfrak{A}}_{(\mathtt{f}^x_1 \diamond (\rho_1 \| \rho_2) \diamond \mathtt{j}^z_1) \diamond \rho'} \tilde{u}}\,\mathrm{T}$,

Case $\dfrac{\dfrac{\tilde t \rightsquigarrow^{\mathfrak{A}}_{\rho_1} \tilde t' \quad \mathsf{E}_z(\tilde t') \quad \tilde s \rightsquigarrow^{\mathfrak{A}}_{\rho_2} \tilde s' \quad \mathsf{E}_z(\tilde s')}{\tilde t \cap_1 \tilde s \rightsquigarrow^{\mathfrak{A}}_{(\rho_1\|\rho_2)\diamond \mathtt{j}_1^z} @z.1} \qquad @z.1 \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde u}{\tilde t\cap_1 \tilde s \rightsquigarrow^{\mathfrak{A}}_{((\rho_1\|\rho_2)\diamond\mathtt{j}_1^z)\diamond\rho'} \tilde u}\,\mathrm{T}$: Then the right-hand side rule should be the rule (R); thus, this case has already been proved.

- Case $\dfrac{\dfrac{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho_1} \tilde t'' \quad @x.s \rightsquigarrow^{\mathfrak{A}}_{\rho_2} \tilde s''}{@x.t\cap s \rightsquigarrow^{\mathfrak{A}}_{\mathtt{f}_1^x\diamond(\rho_1\|\rho_2)} \tilde t''\cap_1 \tilde s''} \qquad \tilde t''\cap_1 \tilde s'' \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde u}{@x.t\cap s \rightsquigarrow^{\mathfrak{A}}_{(\mathtt{f}_1^x\diamond(\rho_1\|\rho_2))\diamond\rho'} \tilde u}\,\mathrm{T}$: For the form of the above of $\tilde t''\cap_1 \tilde s'' \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde u$, we distinguish the following cases (except for the rule (R)):
  - Case $\dfrac{\tilde t'' \rightsquigarrow^{\mathfrak{A}}_{\rho_1'} \tilde t' \quad \tilde s'' \rightsquigarrow^{\mathfrak{A}}_{\rho_2'} \tilde s'}{\tilde t''\cap_1 \tilde s'' \rightsquigarrow^{\mathfrak{A}}_{\rho_1'\|\rho_2'} \tilde t'\cap_1 \tilde s'}$: Then we have:

$$\dfrac{\dfrac{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho_1} \tilde t'' \quad \tilde t'' \rightsquigarrow^{\mathfrak{A}}_{\rho_1'} \tilde t'}{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho_1\diamond\rho_1'} \tilde t'}\,\mathrm{T} \qquad \dfrac{@x.s \rightsquigarrow^{\mathfrak{A}}_{\rho_1} \tilde s'' \quad \tilde s'' \rightsquigarrow^{\mathfrak{A}}_{\rho_1'} \tilde s'}{@x.s \rightsquigarrow^{\mathfrak{A}}_{\rho_2\diamond\rho_2'} \tilde s'}\,\mathrm{T}}{@x.t\cap s \rightsquigarrow^{\mathfrak{A}}_{\mathtt{f}_1^x\diamond((\rho_1\diamond\rho_1')\|(\rho_2\diamond\rho_2'))} \tilde t'\cap_1 \tilde s'}.$$

    Note that $\mathtt{f}_1^x \diamond ((\rho_1\diamond\rho_1') \parallel (\rho_2\diamond\rho_2')) = (\mathtt{f}_1^x\diamond(\rho_1\parallel\rho_2))\diamond(\rho_1'\parallel\rho_2')$. Thus by IH, this case has been proved.
  - Case $\dfrac{\tilde t'' \rightsquigarrow^{\mathfrak{A}}_{\rho_1'} \tilde t' \quad \mathsf{E}_z(\tilde t') \quad \tilde s'' \rightsquigarrow^{\mathfrak{A}}_{\rho_2'} \tilde s' \quad \mathsf{E}_z(\tilde s')}{\tilde t''\cap_1 \tilde s'' \rightsquigarrow^{\mathfrak{A}}_{(\rho_1'\|\rho_2')\diamond\mathtt{j}_1^z} @z.1}$: Then we have:

$$\dfrac{\dfrac{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho_1} \tilde t'' \quad \tilde t'' \rightsquigarrow^{\mathfrak{A}}_{\rho_1'} \tilde t'}{@x.t \rightsquigarrow^{\mathfrak{A}}_{\rho_1\diamond\rho_1'} \tilde t'}\,\mathrm{T} \quad \mathsf{E}_z(\tilde t') \quad \dfrac{@x.s \rightsquigarrow^{\mathfrak{A}}_{\rho_1} \tilde s'' \quad \tilde s'' \rightsquigarrow^{\mathfrak{A}}_{\rho_1'} \tilde s'}{@x.s \rightsquigarrow^{\mathfrak{A}}_{\rho_2\diamond\rho_2'} \tilde s'}\,\mathrm{T} \quad \mathsf{E}_z(\tilde s')}{@x.t\cap s \rightsquigarrow^{\mathfrak{A}}_{\mathtt{f}_1^x\diamond((\rho_1\diamond\rho_1')\|(\rho_2\diamond\rho_2'))\diamond\mathtt{j}_1^z} \tilde t'\cap_1 \tilde s'}.$$

    Note that $\mathtt{f}_1^x \diamond ((\rho_1\diamond\rho_1') \parallel (\rho_2\diamond\rho_2'))\diamond\mathtt{j}_1^z = (\mathtt{f}_1^x\diamond(\rho_1\parallel\rho_2))\diamond((\rho_1'\parallel\rho_2')\diamond\mathtt{j}_1^z)$. Thus by IH, this case has been proved.
- Case $\dfrac{\dfrac{\tilde t \rightsquigarrow^{\mathfrak{A}}_{\rho_1} \tilde t'' \quad \tilde s \rightsquigarrow^{\mathfrak{A}}_{\rho_2} \tilde s''}{\tilde t\cap s \rightsquigarrow^{\mathfrak{A}}_{\rho_1\|\rho_2} \tilde t''\cap_1 \tilde s''} \qquad \tilde t''\cap_1 \tilde s'' \rightsquigarrow^{\mathfrak{A}}_{\rho'} \tilde u}{\tilde t\cap_1 \tilde s \rightsquigarrow^{\mathfrak{A}}_{(\rho_1\|\rho_2)\diamond\rho'} \tilde u}\,\mathrm{T}$: Similar to the case above.

Hence, this completes the proof. □