Spatial Existential Positive Logics for Hyperedge Replacement Grammars

Yoshiki Nakamura
Tokyo Institute of Technology, Japan

Abstract

We study a (first-order) spatial logic based on graphs of conjunctive queries for expressing (hyper-)graph languages. In this logic, each primitive positive (resp. existential positive) formula plays a role of an expression of a graph (resp. a finite language of graphs) modulo graph isomorphism. First, this paper presents a sound- and complete axiomatization for the equational theory of primitive/existential positive formulas under this spatial semantics. Second, we show Kleene theorems between this logic and hyperedge replacement grammars (HRGs), namely that over graphs, the class of existential positive first-order (resp. least fixpoint, transitive closure) formulas has the same expressive power as that of non-recursive (resp. all, linear) HRGs.

2012 ACM Subject Classification Theory of computation → Equational logic and rewriting

Keywords and phrases Existential positive logic, spatial logic, Kleene theorem

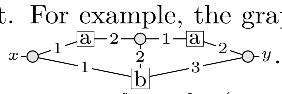
Digital Object Identifier 10.4230/LIPIcs.CSL.2022.30

Related Version *Full Version*: `https://yoshikinakamura.bitbucket.io/papers/ep_for_graph`

Funding This work was supported by JSPS KAKENHI Grant Number JP21K13828.

Acknowledgements We would like to thank the anonymous reviewers for their useful comments.

1 Introduction

Existential positive (EP) *formulas* are first-order formulas that are built up from atomic predicates, equality (=), top (tt), bottom (ff), conjunction (∧), disjunction (∨), and existential quantifier (∃). In particular, *primitive positive* (PP) *formulas* are EP formulas without ff nor ∨. PP formulas are semantically equivalent to *conjunctive queries* [1], which are at the core of query languages in database theory. In this paper, we focus on the *(hyper-)graphs* of conjunctive queries (a.k.a. *natural models* of conjunctive queries) [11][12, Fig. 1], which were introduced to characterize the semantical equivalence of conjunctive queries [11, Lemma 13][28] as follows: two PP formulas are semantically equivalent if and only if their graphs are *homomorphically* equivalent. For example, the graph of the PP formula $\exists z.\mathrm{a}(x,z) \wedge \mathrm{a}(z,y) \wedge \mathrm{b}(x,z,y)$ is the following: . This characterization can be generalized to EP formulas by using finite sets of graphs (see, e.g., [40, Sect. 2.6]).

In this paper, turning our attention to the correspondence between primitive positive logics and (hyper-)graphs, we study PP/EP formulas as *graph/graph-language expressions*. To this end, we introduce a *spatial* semantics (like that of graph logic [10] or separation logic [35, 38]), which is based on graphs of conjunctive queries, called *GI-semantics*. The semantics enables us to study graphs and graph languages through logical formulas in a natural way. The remarkable difference from classical semantics is the following (cf. the above): two PP formulas are equivalent under GI-semantics if and only if their graphs are *(graph-)isomorphically* equivalent. While the equational theory of PP/EP formulas under GI-semantics is subclassical, some formula transformations under classical semantics, in logic and database theory, still work under GI-semantics.

30th EACSL Annual Conference on Computer Science Logic (CSL 2022).
Editors: Florin Manea and Alex Simpson; Article No. 30; pp. 30:1–30:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Our first contribution is to present a sound- and complete axiomatization of the equational theory of PP/EP formulas under GI-semantics. Furthermore, we extend EP with the least-fixpoint operator and the transitive closure operator (see, e.g., [20, Sect. 8]), denoted by EP(LFP) and EP(TC), respectively. They can express possibly infinite graph languages. Our second contribution is to show that each of the logics above has the same expressive power as some class of *hyperedge replacement grammar* (HRG) [25, 36] (see also [19]), which is a generalization of context-free word grammar from words to graphs, as follows.

▶ **Theorem 1.** *Under GI-semantics, for every graph language $\mathcal{G}$ (closed under isomorphism):*

(1) *Some* EP *formula recognizes $\mathcal{G}$ iff some non-recursive HRG recognizes $\mathcal{G}$ (i.e., $\mathcal{G}$ is finite up to isomorphism). In particular, some* PP *formula recognizes $\mathcal{G}$ iff some deterministic and non-recursive HRG recognizes $\mathcal{G}$ (i.e., $\mathcal{G}$ is a singleton up to isomorphism).*

(2) *Some* EP(LFP) *formula recognizes $\mathcal{G}$ iff some HRG recognizes $\mathcal{G}$.*

(3) *Some* EP(TC) *formula recognizes $\mathcal{G}$ iff some linear HRG recognizes $\mathcal{G}$.*

This theorem is an analogy of *Kleene theorem* [27], that over words, for every language L: some regular word grammar (or equivalently, non-deterministic finite automaton) recognizes L if and only if some regular expression recognizes L. Such an equivalence between expressions and grammars/automata like Kleene theorem has also been widely studied for many other language classes (e.g., context-free word languages [29], ω-regular word languages [31], regular tree languages [13, Theorem 2.2.8], language classes over some specific graph classes [30, 6, 5]). To our knowledge, the Kleene theorem for HRGs and linear HRGs (namely, some syntax having the same expressive power) has not yet been investigated, whereas logical or algebraic characterizations are known, e.g., [3, 15].

Related work. This paper uses PP formulas as graph expressions and uses EP(LFP) formulas as graph language expressions. There also are some expressions for (bounded treewidth) graphs (or relational structures), e.g., HR-algebra [3, 16], SP-terms [34], 2p-algebra [14, 18], graphical (string diagrammatic) conjunctive queries [4]. As for the completeness result of PP (Theorem 19), Bauderon and Courcelle [3] have already presented a syntax and a complete axiomatization for graphs modulo isomorphism. However, our completeness proof (essentially [3] also) would have a sufficiently simple strategy relying on the transformation for obtaining conjunctive-queries from primitive positive formulas (under classical semantics); this is a reason that our expressions are based on logical formulas.

As for characterizing language classes by classical logics, it dates back to Büchi-Elgot-Trakhtenbrot Theorem [8, 9, 21, 43] (see also [23]), which states that over words, monadic second-order logic has the same expressive power as the class of regular expressions. See [16, Theorem 7.51][15] for a logical characterization of HRGs, by using monadic second-order logic as a graph transducer. However, the characterization presented in this paper uses logical formulas as *graph-language expressions.*

Also, the number of variables in formulas has a deep connection with the *treewidth* [39, 26] of (hyper-)graphs (or relational structures), which is a parameter indicating how much a graph is similar to a tree. It was mentioned in [28, Remark 5.3] that under the classical semantics, for every relational structure of treewidth k, its conjunctive query is semantically equivalent to an $\mathsf{PP}^{(k+1)(0)}$ formula. Here, $\mathsf{PP}^{k(l)}$ denotes the set of PP formulas using at most k variables and at most l free variables. In particular, it is shown in [32] that under the classical semantics, $\mathsf{PP}^{3(2)}$ has the same expressive power as the primitive positive calculus of relations, which is a fragment of Tarski's calculus of relations [41]. In [14, 18], a sound- and complete axiomatization is presented for 2p-algebra, which is intuitively the primitive positive calculus of relations under GI-semantics. In connection with them, it would be interesting to present a sound- and complete axiomatization of the equational theory of $\mathsf{PP}^{k(l)}$ formulas under GI-semantics, but it still remains open.

Outline. Section 2 presents preliminaries. Section 3 introduces GI-semantics. Section 4 presents an axiomatization of the equational theory under GI-semantics for PP/EP formulas. Section 5 (and 3) shows Kleene theorems between spatial existential positive logic and HRGs (Theorem 1(1)–(3)). Section 6 concludes this paper.

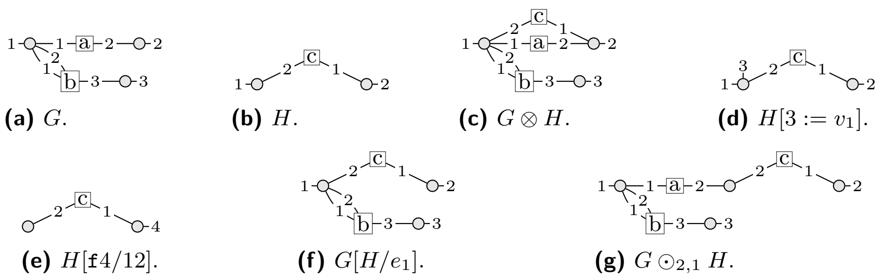
2 Preliminaries

We write $\mathbb{N}$ (resp. $\mathbb{N}_+$) for the set of all non-negative (resp. positive) integers. For $l, r \in \mathbb{N}$, we write $[l, r]$ for the set $\{i \in \mathbb{N} \mid l \leq i \leq r\}$. In particular, we write $[n]$ for $[1, n]$. The *cardinality* of a set A is denoted by $\#(A)$. For an equivalence relation $\sim$ on a set X, the quotient set of X by $\sim$ is denoted by $X/\!\sim$ and the equivalence class of an element x w.r.t. $\sim$ is denoted by $[x]_\sim$. For sets X_1 and X_2, the *disjoint union* $X_1 \uplus X_2$ is defined by $\{\langle i, a\rangle \mid i \in [2], a \in X_i\}$. We denote by $\vec{a} = \langle a_1, \dots, a_n\rangle$ (also denoted by $a_1 \dots a_n$ or $\langle a_i\rangle_{i=1}^{n}$) a *finite sequence*. The *length* $|\vec{a}|$ of $\vec{a}$ is n. We denote by $\mathrm{Occ}(\vec{a})$ the set $\{a_1, \dots, a_n\}$. We say that a sequence $\vec{a}$ is a *permutation* of a set A if $\mathrm{Occ}(\vec{a}) = A$ and the elements of $\vec{a}$ are pairwise distinct. We denote by $\mathrm{Perm}(A)$ the set of all permutations of a set A. We denote by A^* (resp. A^k) the set of all finite sequences (resp. sequences of length k) over a set A. Also, we denote by ι_n (or just by ι if n is obvious) the sequence $\langle 1, 2, \dots, n\rangle$. An *alphabet* A is a possibly infinite set. A *(finite-set-)typed alphabet* A is an alphabet with a function ty^A (or written ty for simplicity) from A to finite sets. In particular we say that a symbol a in A is *ordinal-typed* if $\mathrm{ty}^A(a) = [k]$ for some $k \in \mathbb{N}$. The *arity* of a in A is k, denoted by $\mathrm{ar}^A(a)$ (or just by $\mathrm{ar}(a)$).

Graphs. In the following, we define graphs (with ports) and graph languages.

▶ **Definition 2** (graph). *Given a typed alphabet A and a finite set τ, an A-labelled* graph *G of type τ is a tuple $\langle V^G, E^G, \mathrm{lab}^G, \mathrm{vert}^G, \mathrm{port}^G\rangle$, where V^G is a finite set of* vertices*, E^G is a finite set of* (hyper-)edges*, $\mathrm{lab}^G \colon E^G \to A$ is a function denoting the* label *of each edge, $\mathrm{vert}^G(e) \colon \mathrm{ty}^G(e) \to V^G$ is a function denoting the* vertices *of each edge, and $\mathrm{port}^G \colon \mathrm{ty}(G) \to V^G$ is a function denoting the* ports *of G. Here, $\mathrm{ty}(G) \triangleq \tau$ and $\mathrm{ty}^G \triangleq \mathrm{ty}^A \circ \mathrm{lab}^G$.*

▶ **Example 3.** Let $A = \{\mathrm{a}, \mathrm{b}, \mathrm{c}\}$ with type $\mathrm{ty}^A = \{\mathrm{a} \mapsto [2], \mathrm{b} \mapsto [3], \mathrm{c} \mapsto [2]\}$. Let $G = \langle \{v_1, v_2, v_3\}, \{e_1, e_2\}, \{e_1 \mapsto \mathrm{a}, e_2 \mapsto \mathrm{b}\}, \{e_1 \mapsto \lambda i \in [2].v_i, e_2 \mapsto \{1 \mapsto v_1, 2 \mapsto v_1, 3 \mapsto v_3\}\}, \lambda i \in [3].v_i\rangle$ and let $H = \langle \{v_1, v_2\}, \{e\}, \{e \mapsto \mathrm{c}\}, \{e \mapsto \{1 \mapsto v_2, 2 \mapsto v_1\}\}, \lambda i \in [2].v_i\rangle$ be A-labelled graphs (of type [3] and of type [2], respectively), where v_1, v_2, v_3, e_1, e_2 are pairwise distinct. Their graphical representations are in Figure 1a and 1b, respectively.



(a) G. (b) H. (c) $G \otimes H$. (d) $H[3 := v_1]$. (e) $H[\mathtt{f}4/12]$. (f) $G[H/e_1]$. (g) $G \odot_{2,1} H$.

Figure 1 Examples of graphs and operations on graphs.

Later (e.g., in Example 12), for binary edges and ports, we often use ○-[a]→○ to denote ○-1-[a]-2-○ for symbols a of the type [2] and use →○ ○→ to denote 1-○ ○-2. Also, for unlabelled non-hyper graphs, let $A_{\mathrm{E}} \triangleq \{\mathrm{E}\}$ with $\mathrm{ty}^{A_{\mathrm{E}}} = \{\mathrm{E} \mapsto [2]\}$ and we use ○→○ to denote ○-[E]→○.

We denote by GR_A^τ the set of all A-labelled graphs of type τ. An *(A-labelled) graph language* $\mathcal{G}$ (of type τ) is a subset of GR_A^τ. Given a system $\mathfrak{S}$ (e.g., HRGs, EP formulas, ...) over A (that defines a graph language $\mathcal{G}(E)$ for every E in $\mathfrak{S}$), we say that $\mathcal{G}$ is *recognized* by $\mathfrak{S}$ if there exists some element E in $\mathfrak{S}$ such that $\mathcal{G} = \mathcal{G}(E)$.

▶ **Definition 4** (homomorphism, isomorphism). *Let $G, H \in \mathsf{GR}_A^\tau$ be graphs. A pair $h = \langle h^{\mathrm{V}}, h^{\mathrm{E}} \rangle$ of $h^{\mathrm{V}} \colon V^G \to V^H$ and $h^{\mathrm{E}} \colon E^G \to E^H$ is a* homomorphism *from G to H if (1) $\mathrm{lab}^G = \mathrm{lab}^H \circ h^{\mathrm{E}}$, (2) $\mathrm{vert}^H(h^{\mathrm{E}}(e))(x) = h^{\mathrm{V}}(\mathrm{vert}^G(e)(x))$, and (3) $\mathrm{port}^H = h^{\mathrm{V}} \circ \mathrm{port}^G$. In particular, h is an* isomorphism *if both h^{V} and h^{E} are bijective. We say that G and H are* isomorphic, *written $G \cong H$ if there exists an isomorphism between G and H.*

In this paper, we will only focus on $\cong$*-closed* (i.e., if $G \in \mathcal{G}$ and $G \cong H$, then $H \in \mathcal{G}$) graph languages. We denote by $\mathcal{G}^{\cong}$ the minimal $\cong$-closed graph language including $\mathcal{G}$.

Some operations on graphs. In the following, we present some primitive operations on graphs. See Figure 1c-1g for graphical examples of Definition 5-8. In GI-semantics, $*$ uses glueing, $\exists$ uses forgetting, LFP uses hyperedge replacing, TC uses concatenating.

▶ **Definition 5** (glueing). *Let $G_1 \in \mathsf{GR}_A^\tau$ and $G_2 \in \mathsf{GR}_A^\upsilon$. Let $G_1 \otimes G_2 \in \mathsf{GR}_A^{\tau \cup \upsilon}$ be the graph such that $V^{G_1 \otimes G_2} = (V^{G_1} \uplus V^{G_2})/\simeq$, $E^{G_1 \otimes G_2} = E^{G_1} \uplus E^{G_2}$, $\mathrm{lab}^{G_1 \otimes G_2}(\langle k, e \rangle) = \mathrm{lab}^{G_k}(e)$, $\mathrm{vert}^{G_1 \otimes G_2}(\langle k, e \rangle)(x) = [\mathrm{vert}^{G_k}(e)(x)]_{\simeq}$, and $\mathrm{port}^{G_1 \otimes G_2}(x) = [\mathrm{port}^{G_k}(x)]_{\simeq}$. Here, $\simeq$ is the minimal equivalence relation such that for every $x \in \tau \cap \upsilon$, $\langle 1, \mathrm{port}^{G_1}(x) \rangle \simeq \langle 2, \mathrm{port}^{G_2}(x) \rangle$.*

▶ **Definition 6** (labelling/forgetting/renaming). *Let $G \in \mathsf{GR}_A^\tau$. For a vertex $v \in V^G$, a variable $z \notin \tau$, and a variable $x \in \tau$, we define the graphs $G[z := v] \in \mathsf{GR}_A^{\tau \cup \{z\}}$, $G[\mathtt{f}/x] \in \mathsf{GR}_A^{\tau \setminus \{x\}}$, $G[z/x] \in \mathsf{GR}_A^{(\tau \setminus \{x\}) \cup \{z\}}$ by $G[z := v] \triangleq \langle V^G, E^G, \mathrm{lab}^G, \mathrm{vert}^G, \mathrm{port}^G \cup \{z \mapsto v\} \rangle$, $G[\mathtt{f}/x] \triangleq \langle V^G, E^G, \mathrm{lab}^G, \mathrm{vert}^G, \mathrm{port}^G \setminus \{x \mapsto \mathrm{port}^G(x)\} \rangle$, $G[z/x] \triangleq G[\mathtt{f}/x][z := \mathrm{port}^G(x)]$.*

We write $G[y_1 \dots y_n/x_1 \dots x_n]$ for $G[z_1/x_1] \dots [z_n/x_n][y_1/z_1] \dots [y_n/z_n]$, where $z_1 \dots z_n$ is a sequence of fresh variables. For a sequence $z_1 \dots z_n$ of pairwise distinct variables, we write $G[z_1 \dots z_n := v_1 \dots v_n]$ for $G[z_1 := v_1] \dots [z_n := v_n]$.

▶ **Definition 7** (hyperedge replacing). *Let $G \in \mathsf{GR}_A^\tau$. For an edge $e \in E^G$ and a graph $H \in \mathsf{GR}_A^{\mathrm{ty}^G(e)}$, let $G[H/e] \in \mathsf{GR}_A^\tau$ be the graph $((G \setminus e)[\vec{z} := \mathrm{vert}^G(e)(x_1) \dots \mathrm{vert}^G(e)(x_n)] \otimes H[\vec{z}/x_1 \dots x_n])[\mathtt{f} \dots \mathtt{f}/\vec{z}]$, where $G \setminus e$ denotes the graph G in which the edge e has been removed. Here, $x_1 \dots x_n \in \mathrm{Perm}(\mathrm{ty}(H))$, and $\vec{z}$ is a sequence of fresh variables.*

We write $G[H_1 \dots H_n/e_1 \dots e_n]$ for $G[H_1/e_1][H_2 \dots H_n/\langle 1, e_2 \rangle \dots \langle 1, e_n \rangle]$ if $n \geq 1$, and G if $n = 0$.

▶ **Definition 8** (concatenating). *Let $G \in \mathsf{GR}_A^\tau$ and $H \in \mathsf{GR}_A^\upsilon$. Let $\vec{x} \in \mathrm{ty}(G)^k$ and $\vec{y} \in \mathrm{ty}(H)^k$ be sequences of pairwise distinct elements, where $k \geq 1$. Then, let $G \odot_{\vec{x}\vec{y}} H \in \mathsf{GR}_A^{(\tau \setminus \mathrm{Occ}(\vec{x})) \cup (\upsilon \setminus \mathrm{Occ}(\vec{y}))}$ be the graph $(G[\vec{z}/\vec{x}] \otimes H[\vec{z}/\vec{y}])[\mathtt{f} \dots \mathtt{f}/\vec{z}]$, where $\vec{z}$ is a sequence of fresh variables.*

Finally, we list some basic equations in the following.

▶ **Proposition 9.** *(1) $G_1 \otimes (G_2 \otimes G_3) \cong (G_1 \otimes G_2) \otimes G_3$; (2) $G \otimes H \cong H \otimes G$; (3) $(H_1 \otimes H_2)[G/\langle 1, e \rangle] \cong H_1[G/e] \otimes H_2$; (4) $G[z/x][H/e] \cong G[H/e][z/x]$; (5) $G[z/x] \otimes H \cong (G \otimes H)[z/x]$ if $x \notin \mathrm{ty}(H)$.*

Hyperedge Replacement Grammars. In the following, we present the definition of hyperedge replacement grammars (HRGs).

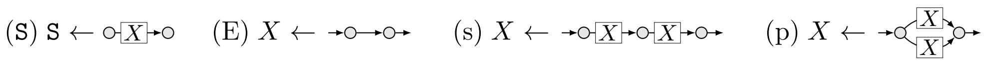
▶ **Definition 10** (e.g., [19]). *A* hyperedge replacement grammar *(HRG)* $\mathscr{H}$ *over a typed alphabet* A *is a tuple* $\langle \mathcal{X}^{\mathscr{H}}, \mathcal{R}^{\mathscr{H}}, \mathsf{S}^{\mathscr{H}} \rangle$*, where* $\mathcal{X}^{\mathscr{H}}$ *is a finite typed alphabet disjoint with* A *for* (non-terminal) labels*,* $\mathcal{R}^{\mathscr{H}}$ *is a finite set of pairs* $r = \langle X, G \rangle$ *(written* $X \leftarrow G$*) of* $X \in \mathcal{X}^{\mathscr{H}}$ *and* $G \in \mathsf{GR}^{\mathrm{ty}(X)}_{A \cup \mathcal{X}^{\mathscr{H}}}$ *for* rewriting rules*, and* $\mathsf{S}^{\mathscr{H}} \in \mathcal{X}^{\mathscr{H}}$ *denotes the* source label*.*

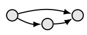
We also define the graph languages of HRGs as follows.

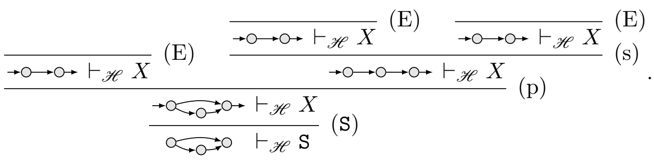
▶ **Definition 11** (cf. [19, Sect. 2.3.2]). *For an HRG* $\mathscr{H} = \langle \mathcal{X}, \mathcal{R}, \mathsf{S} \rangle$ *over a typed alphabet* A*, the binary relation* $\vdash_{\mathscr{H}} \subseteq \bigcup_{X \in \mathcal{X}} \mathsf{GR}^{\mathrm{ty}(X)}_{A} \times \{X\}$ *is defined as the least* $\cong$*-closed (i.e., if* $G \cong H$ *and* $G \vdash_{\mathscr{H}} X$*, then* $H \vdash_{\mathscr{H}} X$*) relation closed under the following rule: If* $X \leftarrow G \in \mathcal{R}$*, then* $\dfrac{H_1 \vdash_{\mathscr{H}} \mathrm{lab}^G(e_1) \ \ldots \ H_n \vdash_{\mathscr{H}} \mathrm{lab}^G(e_n)}{G[H_1 \ldots H_n / e_1 \ldots e_n] \vdash_{\mathscr{H}} X}$*. The* graph language *is defined by:* $\mathcal{G}(\mathscr{H}) \triangleq \{G \in \mathsf{GR}^{\mathrm{ty}(\mathsf{S})}_{A} \mid G \vdash_{\mathscr{H}} \mathsf{S}\}$*.*

For an HRG $\mathscr{H}$, we say that $\mathscr{H}$ is *linear* [36, Definition 3] if for every rule $X \leftarrow G \in \mathcal{R}^{\mathscr{H}}$, the number of non-terminal labels occurring in G is at most one. We say that $\mathscr{H}$ is *(n-)recursive* if there exist rules $X_0 \leftarrow G_0, \ldots, X_n \leftarrow G_n \in \mathcal{R}^{\mathscr{H}}$ such that X_i occurs in G_{i-1} for $i \in [0, n]$ where $n \in \mathbb{N}$ and G_{-1} denotes G_n.

▶ **Example 12.** Let $\mathscr{H}$ be the HRG over A_{E}, defined by $\mathrm{ty}^{\mathcal{X}^{\mathscr{H}}} = \{\mathsf{S} \mapsto [0], X \mapsto [2]\}$, $\mathcal{R}^{\mathscr{H}} = \{(\mathsf{S}), (\mathrm{E}), (\mathrm{s}), (\mathrm{p})\}$, and $\mathsf{S}^{\mathscr{H}} = \mathsf{S}$, where each rule in $\mathcal{R}^{\mathscr{H}}$ is as follows:



Then, $\mathcal{G}(\mathscr{H})$ is the set of all (directed) series-parallel graphs [24], e.g.,  $\in \mathcal{G}(\mathscr{H})$ is shown by:



3 Existential Positive Logics under GI-Semantics

In this section, we introduce the syntax and a *spatial semantics* of our existential positive logics. Let A be an ordinal-typed alphabet, $\mathcal{V}_1$ be a countably infinite set of *first-order variables*, and $\mathcal{V}_2$ be an ordinal-typed set of *second-order variables*, where for every $k \in \mathbb{N}_+$, the number of second-order variables of arity k is countably infinite. Here, A, $\mathcal{V}_1$, and $\mathcal{V}_2$ are disjoint. For $\tau \subseteq \mathcal{V}_1$ and $\mathcal{X} \subseteq A \cup \mathcal{V}_2$, we define $\mathsf{Fml}^{\tau}_{\mathcal{X}}$ as the least set closed under the rules as follows.[1]

$$\frac{}{\top \in \mathsf{Fml}^{\emptyset}_{\mathcal{X}}} \quad \frac{}{x = y \in \mathsf{Fml}^{\{x,y\}}_{\mathcal{X}}} \quad \frac{}{X\vec{x} \in \mathsf{Fml}^{\mathrm{Occ}(\vec{x})}_{\mathcal{X}}}\dagger_1 \quad \frac{\varphi \in \mathsf{Fml}^{\tau}_{\mathcal{X}} \quad \psi \in \mathsf{Fml}^{\upsilon}_{\mathcal{X}}}{\varphi * \psi \in \mathsf{Fml}^{\tau \cup \upsilon}_{\mathcal{X}}} \quad \frac{\varphi \in \mathsf{Fml}^{\tau \cup \{x\}}_{\mathcal{X}}}{\exists x.\varphi \in \mathsf{Fml}^{\tau}_{\mathcal{X}}}\dagger_2$$

$$\frac{}{\mathsf{ff} \in \mathsf{Fml}^{\tau}_{\mathcal{X}}} \quad \frac{\varphi \in \mathsf{Fml}^{\tau}_{\mathcal{X}} \quad \psi \in \mathsf{Fml}^{\tau}_{\mathcal{X}}}{\varphi \vee \psi \in \mathsf{Fml}^{\tau}_{\mathcal{X}}} \quad \frac{\varphi \in \mathsf{Fml}^{\mathrm{Occ}(\vec{x})}_{\mathcal{X} \cup \{X\}}}{[\mathsf{LFP}_{\vec{x},X}\varphi]\vec{y} \in \mathsf{Fml}^{\mathrm{Occ}(\vec{y})}_{\mathcal{X}}}\dagger_3 \quad \frac{\varphi \in \mathsf{Fml}^{\mathrm{Occ}(\vec{x}\vec{y})}_{\mathcal{X}}}{[\varphi]^{+}_{\vec{x}\vec{y}}\vec{u}\vec{w} \in \mathsf{Fml}^{\mathrm{Occ}(\vec{u}\vec{w})}_{\mathcal{X}}}\dagger_4$$

$\dagger_1$: $X \in \mathcal{X}$ and $\mathrm{ar}^{\mathcal{X}}(X) = |\vec{x}|$. $\dagger_2$: $x \notin \tau$. $\dagger_3$: $\mathrm{ar}(X) = |\vec{x}| = |\vec{y}| \geq 1$. $\vec{x}$ and $\vec{y}$ are sequences of pairwise distinct variables. $\dagger_4$: $|\vec{x}| = |\vec{y}| = |\vec{u}| = |\vec{w}| \geq 1$. $\vec{x}\vec{y}$ and $\vec{u}\vec{w}$ are sequences of pairwise distinct variables.

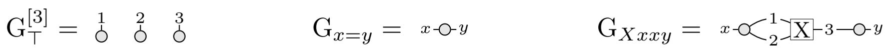
[1] We adopt the spatial conjunction symbol $*$ instead of $\wedge$.

We often use parentheses in ambiguous situations. We say that φ is a *formula* over A of *type* τ if $\varphi \in \mathsf{Fml}_A^\tau$. Note that, for a technical reason, ff has any type τ. We use $\mathsf{FV}_1(\varphi)/\mathsf{FV}_2(\varphi)$ (resp. $\mathsf{BV}_1(\varphi)/\mathsf{BV}_2(\varphi)$) to denote the set of first-/second-order free (resp. bound) variables of φ, and use $\mathsf{V}_l(\varphi)$ to denote the set $\mathsf{FV}_l(\varphi) \cup \mathsf{BV}_l(\varphi)$ for $l = 1, 2$. The set PP_A^τ *(resp.* EP_A^τ, $\mathsf{EP(LFP)}_A^\tau$, $\mathsf{EP(TC)}_A^\tau$*)* is defined as the set of all $\varphi \in \mathsf{Fml}_A^\tau$ such that φ is generated from the rules for $\top$, $=$, $X\vec{x}$, $*$, and $\exists$. (resp. the rules for PP with ff and $\vee$, the rules for EP with LFP, the rules for EP with TC). Note that some syntax restrictions exist, e.g., $\top \vee Xx \notin \mathsf{Fml}_{\mathcal{X}}^\tau$ for any $\mathcal{X}$ and τ. They are for simplifying the definition of GI-semantics.

For notational simplicity, we denote by $\mathop{\ast}_{i=1}^{n} \varphi_i$ (similarly for $\bigvee_{i=1}^{n} \varphi_i$) the formula $(\mathop{\ast}_{i=1}^{n-1} \varphi_i) * \varphi_n$ if $n \geq 1$ and the formula $\top$ if $n = 0$, by $x_1 \dots x_n = y_1 \dots y_n$ the formula $\mathop{\ast}_{i=1}^{n} x_i = y_i$, by $\exists x_1 \dots x_n.\varphi$ the formula $\exists x_1.\exists x_2.\dots\exists x_n.\varphi$, and by $\varphi[y_1 \dots y_n/x_1 \dots x_n]$ the formula φ in which each free variable x_i occurring in φ has been replaced with y_i where $i \in [n]$. A formula φ is *atomic* if φ forms $\top$, $x = y$, or $X\vec{x}$. Explicitly, we may use $\tilde{\varphi}$ to denote an atomic formula. We use atomic formulas to denote atomic graphs as follows.

▶ **Definition 13.** *For a finite set* τ*, let* $\mathrm{G}_\top^\tau \triangleq \langle \tau, \emptyset, \emptyset, \emptyset, \lambda x \in \tau.x \rangle$*. For an atomic formula* $\tilde{\varphi}$*, we define the graph* $\mathrm{G}_{\tilde{\varphi}}$ *by:* $\mathrm{G}_\top \triangleq \mathrm{G}_\top^\emptyset$*,* $\mathrm{G}_{x=y} \triangleq \langle \{\mathrm{v}\}, \emptyset, \emptyset, \emptyset, \lambda z \in \{x, y\}.\mathrm{v} \rangle$*, and* $\mathrm{G}_{X\vec{x}} \triangleq \langle \mathrm{Occ}(\vec{x}), \{\mathrm{e}\}, \{\mathrm{e} \mapsto X\}, \{\mathrm{e} \mapsto \lambda i \in \mathrm{ty}(X).x_i\}, \lambda y \in \mathrm{Occ}(\vec{x}).y \rangle$*.*

For example, $\mathrm{G}_\top^{[3]}$, $\mathrm{G}_{x=y}$, and G_{Xxxy} are as follows, where $x \neq y$:



In the following, we define a spatial semantics for graph languages, called *GI-semantics*.[2] Note that for every φ, if $G \models^{\mathrm{GI}} \varphi$, then $\mathrm{ty}(G)$ is determined to $\mathsf{FV}_1(\varphi)$.

▶ **Definition 14** (GI-semantics). *The binary relation* $\models^{\mathrm{GI}} \subseteq \bigcup_{\tau \subseteq \mathcal{V}_1;\, \mathcal{X} \subseteq A \cup \mathcal{V}_2} \mathsf{GR}_{\mathcal{X}}^\tau \times \mathsf{Fml}_{\mathcal{X}}^\tau$ *is defined as the least* $\cong$*-closed relation closed under the rules in Figure 2.*

$$\frac{}{\mathrm{G}_{\tilde{\varphi}} \models^{\mathrm{GI}} \tilde{\varphi}}\,(\mathrm{At}) \qquad \frac{G \models^{\mathrm{GI}} \varphi \quad H \models^{\mathrm{GI}} \psi}{G \otimes H \models^{\mathrm{GI}} \varphi * \psi}\,(*) \qquad \frac{\langle G_i \models^{\mathrm{GI}} \varphi \rangle_{i=1}^{n}}{(G_1 \odot_{\vec{y}\vec{x}} \cdots \odot_{\vec{y}\vec{x}} G_n)[\vec{u}\vec{w}/\vec{x}\vec{y}] \models^{\mathrm{GI}} [\varphi]_{\vec{x}\vec{y}}^{+}\vec{u}\vec{w}}\,(\mathsf{TC})\ \dagger_1$$

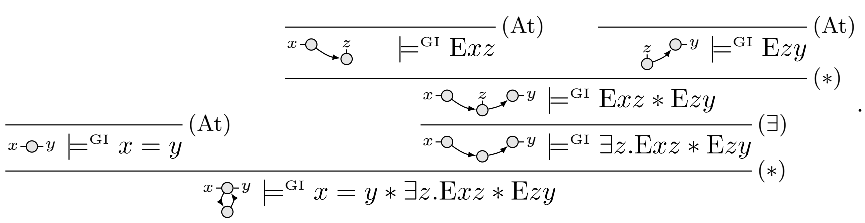
$$\frac{G \models^{\mathrm{GI}} \varphi}{G[\mathtt{f}/x] \models^{\mathrm{GI}} \exists x.\varphi}\,(\exists) \qquad \frac{G \models^{\mathrm{GI}} \varphi_i}{G \models^{\mathrm{GI}} \varphi_1 \vee \varphi_2}\,(\vee)\ \dagger_2 \qquad \frac{H \models^{\mathrm{GI}} \varphi \quad \langle G_i \models^{\mathrm{GI}} [\mathsf{LFP}_{\vec{x},X}\varphi]\iota \rangle_{i=1}^{n}}{H[G_1 \dots G_n/\vec{e}_X^H][\vec{y}/\vec{x}] \models^{\mathrm{GI}} [\mathsf{LFP}_{\vec{x},X}\varphi]\vec{y}}\,(\mathsf{LFP})\ \dagger_3$$

$\dagger_1$: $n \in \mathbb{N}_+$. $\dagger_2$: $i \in [2]$. $\dagger_3$: $n \in \mathbb{N}$ and $\vec{e}_X^H$ denotes a permutation of all the X-labelled edges in H.

Figure 2 Definition of GI-semantics.

The *graph language* of φ is defined by $\mathcal{G}(\varphi) \triangleq \{G \mid G \models^{\mathrm{GI}} \varphi\}$. We say that φ and ψ are *graph-isomorphically equivalent* (GI-equivalent), written $\varphi \cong^{\mathrm{GI}} \psi$ if $\mathcal{G}(\varphi) = \mathcal{G}(\psi)$.

▶ **Example 15.** Let $G \triangleq$ (graph) and $\varphi \triangleq x = y * \exists z.\mathrm{E}xz * \mathrm{E}zy$. Then, $G \models^{\mathrm{GI}} \varphi$ is shown by:



We will generalize this example in Definition 16, for expressing any graphs by PP formulas.

[2] See [33, Appendix A] for an alternative definition. Here, we adopt this style for extending to Definition 27.

3.1 PP/EP formulas as graph/finite-graph-language expressions

In this subsection, we show that PP (resp. EP) formulas under GI-semantics play a role as graph expressions (resp. finite graph language expressions).

▶ **Definition 16.** *Let G be a graph, $\vec{x} = x_1 \dots x_k \in \mathrm{Perm}(\mathrm{ty}(G))$, $\vec{v} = v_1 \dots v_n \in \mathrm{Perm}(V^G)$, and $\vec{e} = e_1 \dots e_m \in \mathrm{Perm}(E^G)$. Let $\varphi_G^{\vec{x},\vec{v},\vec{e}}$ (or written φ_G if they are not important) be the following* PP *formula, where $z_{v_1}, \dots, z_{v_n}$ are fresh variables:*

$$\exists z_{v_1} \dots z_{v_n}.(\mathop{\ast}_{i=1}^{k} z_{\mathrm{port}^G(x_i)} = x_i) * (\mathop{\ast}_{i=1}^{m} \mathrm{lab}^G(e_i)\ z_{\mathrm{vert}^G(e_i)(1)} \dots z_{\mathrm{vert}^G(e_i)(\mathrm{ar}^G(e_i))}).$$

Also, for a finite sequence $\vec{G} = G_1 \dots G_n$ of graphs, let $\varphi_{\vec{G}}$ be the EP *formula $\bigvee_{i=1}^{n} \varphi_{G_i}$.*

Then, $\mathcal{G}(\varphi_G) = \{G\}^{\cong}$ and $\mathcal{G}(\varphi_{\vec{G}}) = \mathrm{Occ}(\vec{G})^{\cong}$. By using them, the following holds.

▶ **Proposition 17** (Theorem 1(1))**.** *For every graph language $\mathcal{G}$ closed under isomorphism: (1): $\mathcal{G}$ is singleton up to isomorphism iff some* PP *formula recognizes $\mathcal{G}$. (2): $\mathcal{G}$ is finite up to isomorphism iff some* EP *formula recognizes $\mathcal{G}$.*

Proof. (1)(2)(⇒): By using φ_G and $\varphi_{\vec{G}}$, respectively. (1)(2)(⇐): By a straightforward induction on the structure of PP (resp. EP) formulas. ◀

▶ Remark 18. Indeed, GI-semantics characterizes the *graphs of* PP *formulas* [11] (see also [12, Figure 1]), namely, for every PP formula φ, $G \models^{\mathrm{GI}} \varphi$ iff G is isomorphic to the graph of φ. Thus, two PP formulas are GI-equivalent iff their graphs are isomorphically equivalent.

4 An Axiomatization of the Equational Theory of PP/EP

This section presents an axiomatization of the equational theory under GI-semantics (i.e., the binary relation $\cong^{\mathrm{GI}}$) of PP/EP formulas. Given an ordinal-typed alphabet A, we define the binary relation $\simeq \subseteq \bigcup_{\tau \subseteq \mathcal{V}_1} \mathsf{EP}_A^{\tau} \times \mathsf{EP}_A^{\tau}$ as the minimal relation closed under the rules in Figure 3.[3] Inference rules consist of the rules for equivalence relation and the rules for "α-equivalence" (see, e.g., [37, Sect. 4.1.] for λ-calculus).

Inference rules:

$$\frac{}{\varphi \simeq \varphi} \quad \frac{\varphi \simeq \psi}{\psi \simeq \varphi} \quad \frac{\varphi \simeq \psi \quad \psi \simeq \rho}{\varphi \simeq \rho} \quad \frac{\varphi \simeq \varphi' \quad \psi \simeq \psi'}{\varphi * \psi \simeq \varphi' * \psi'} \quad \frac{\varphi[z/x] \simeq \psi[z/y]}{\exists x.\varphi \simeq \exists y.\psi}\dagger_1 \quad \frac{\varphi \simeq \varphi' \quad \psi \simeq \psi'}{\varphi \vee \psi \simeq \varphi' \vee \psi'}$$

Axioms:

(=1) $x = y \simeq y = x$ (=2) $x = x * \varphi \simeq \varphi$ (=3) $x = y * \varphi[x/z] \simeq x = y * \varphi[y/z]$ (=4) $\exists x. x = y \simeq y = y$
(∗1) $\varphi * (\psi * \rho) \simeq (\varphi * \psi) * \rho$ (∗2) $\varphi * \psi \simeq \psi * \varphi$ (∗3) $\varphi * \top \simeq \varphi$ (∃1) $\exists x.\exists y.\varphi \simeq \exists y.\exists x.\varphi$
(∃2) $(\exists x.\varphi) * \psi \simeq \exists x.\varphi * \psi$ (∨1) $\varphi \vee (\psi \vee \rho) \simeq (\varphi \vee \psi) \vee \rho$ (∨2) $\varphi \vee \psi \simeq \psi \vee \varphi$ (∨3) $\varphi \vee \mathsf{ff} \simeq \varphi$
(∨4) $\varphi \vee \varphi \simeq \varphi$ (∨5) $\exists x.\varphi \vee \psi \simeq (\exists x.\varphi) \vee (\exists x.\psi)$ (∨6) $\varphi * (\psi \vee \rho) \simeq (\varphi * \psi) \vee (\varphi * \rho)$ (ff) $\mathsf{ff} * \varphi \simeq \mathsf{ff}$
$\dagger_1$: z is a fresh variable.

Figure 3 An axiomatization of the equational theory under GI-semantics of PP/EP formulas.

[3] We assume that the left- and right-hand side formulas have an identical type. This restriction implicitly implies the following: when their graph languages are not empty, $x \notin \mathsf{FV}_1(\psi)$ in (∃2), $x \in \mathsf{FV}_1(\varphi)$ in (=2), and $y \neq x$ in (=4), respectively. Also, note that we can use (ff) even if $\mathrm{ty}(\varphi) \neq \emptyset$, because ff has any type.

▶ **Theorem 19.** *The system in Figure 3 is sound and complete for the equational theory under GI-semantics of* PP/EP *formulas, that is, for every* $\varphi, \psi \in \mathsf{EP}^{\tau}_{A}$*,* $\varphi \simeq \psi$ *iff* $\varphi \cong^{\mathrm{GI}} \psi$*.*

In the next subsection, we prove this theorem. The following is a proof sketch.

Proof Sketch of Theorem 19. The soundness is straightforward. For completeness, we show by using the rules in Figure 3 that we can transform each formula into a normal form in two steps: (1) transform each EP formula into a disjunctive normal form of PP formulas; (2) transform each PP formula into a formula of the form φ_G in Definition 16.

4.1 Proof of Theorem 19

▶ **Proposition 20.** (1)*:* $\varphi_G^{\vec{x}_1,\vec{v}_1,\vec{e}_1} \simeq \varphi_G^{\vec{x}_2,\vec{v}_2,\vec{e}_2}$*.* (2)*: If there is an isomorphism* h *from* G *to* H*, then* $\varphi_G^{x_1\dots x_k, v_1\dots v_n, e_1\dots e_m} \simeq \varphi_H^{x_1\dots x_k, h^{\mathrm{V}}(v_1)\dots h^{\mathrm{V}}(v_n), h^{\mathrm{E}}(e_1)\dots h^{\mathrm{E}}(e_m)}$*.* (3)*: If* $G \cong H$*, then* $\varphi_G^{\vec{x}_1,\vec{v}_1,\vec{e}_1} \simeq \varphi_H^{\vec{x}_2,\vec{v}_2,\vec{e}_2}$*.*

Proof. (1): By permutating names using $(*1)(*2)$ for $\vec{x}_1$ and $\vec{x}_2$, $(\exists 1)$ for $\vec{v}_1$ and $\vec{v}_2$, $(*1)(*2)$ for $\vec{e}_1$ and $\vec{e}_2$, respectively. (2): Since they are the same up to variable names. (3): By (2)(1). ◀

Hereafter in this section, relying on this proposition, we write $\varphi_G^{\vec{x},\vec{v},\vec{e}}$ as φ_G, for simplicity.

▶ **Lemma 21.** *For every* PP *formula* φ*:* (1)*: Let* $x \in \mathsf{FV}_1(\varphi)$ *and* $y \neq x$*. Then,* $\exists x.x = y * \varphi \simeq \varphi[y/x]$*.* (2)*: Let* $z_1 \dots z_n \in \mathrm{Perm}(\mathsf{FV}_1(\varphi))$*,* $k \in \mathbb{N}$*, and* $f, g\colon [k] \to [n]$ *be maps. Let* $\sim$ *be the minimal equivalence relation on* $[n]$ *such that for every* $i \in [k]$*,* $f(i) \sim g(i)$ *and let* $I_1 \dots I_m$ *be a permutation of all the quotient classes of* $[n]$ *w.r.t.* $\sim$*. Then,* $\exists z_1 \dots z_n.(\mathop{\Large *}_{i=1}^{k} z_{f(i)} = z_{g(i)}) * \varphi \simeq \exists z_{I_1} \dots z_{I_m}.\varphi[z_{[1]_\sim} \dots z_{[n]_\sim}/z_1 \dots z_n]$*. Here,* $z_{I_1}, \dots, z_{I_m}$ *are pairwise distinct variables.*

Proof. (1): $\exists x.x = y * \varphi \simeq_{(=3)} \exists x.x = y * \varphi[y/x] \simeq_{(\exists 2)} (\exists x.x = y) * \varphi[y/x] \simeq_{(\exists 5)} y = y * \varphi[y/x] \simeq_{(=2)} \varphi[y/x]$. (2): By induction on k. Case $k = 0$. $\exists z_1 \dots z_n.\top * \varphi \simeq_{(*2)(*3)} \exists z_1 \dots z_n.\varphi \simeq \exists z_{\{1\}} \dots z_{\{n\}}.\varphi[z_{\{1\}} \dots z_{\{n\}}/z_1 \dots z_n]$. Case $k \geq 1$. Then,

$$\exists z_1 \dots z_n.(\mathop{\Large *}_{i=1}^{k} z_{f(i)} = z_{g(i)}) * \varphi \simeq_{(*1)} \exists z_1 \dots z_n.(\mathop{\Large *}_{i=1}^{k-1} z_{f(i)} = z_{g(i)}) * (z_{f(k)} = z_{g(k)} * \varphi)$$

$$\simeq \exists z_{I'_1} \dots z_{I'_{m'}}.z_{[f(k)]_{\sim'}} = z_{[g(k)]_{\sim'}} * \varphi[z_{[1]_{\sim'}} \dots z_{[n]_{\sim'}}/z_1 \dots z_n]$$

($\sim'$ and $I'_1 \dots I'_{m'}$ are the ones obtained by I.H. w.r.t. $k-1$.)

(Here, we assume without loss of generality by $(\exists 1)$ that $z_{I'_{m'}} = z_{[f(k)]_{\sim'}}$.)

$$\simeq \exists z_{I'_1} \dots z_{I'_m}.\varphi[z_{[1]_{\sim'}} \dots z_{[n]_{\sim'}}/z_1 \dots z_n][z_{[g(k)]_{\sim'}}/z_{[f(k)]_{\sim'}}]$$

(Apply (=2) if $[f(k)]_{\sim'} = [g(k)]_{\sim'}$ and (1) if $[f(k)]_{\sim'} \neq [g(k)]_{\sim'}$.)

(Here, $m = m'$ for (=2) and $m = m' - 1$ for (1).)

$$\simeq \exists z_{I_1} \dots z_{I_m}.\varphi[z_{[1]_\sim} \dots z_{[n]_\sim}/z_1 \dots z_n].$$

(They are the same up to variable names.) ◀

▶ **Lemma 22.** *For every* PP *formula* φ*, if* $G \models^{\mathrm{GI}} \varphi$*, then* $\varphi \simeq \varphi_G$*.*

Proof. By induction on the structure of PP formulas. Case $\varphi \equiv \top$. By $\varphi_{\mathrm{G}_\top} \equiv \top * \top \simeq_{(*3)} \top$. Case $\varphi \equiv x = x$. By $\varphi_{\mathrm{G}_{x=x}} \simeq_{(*3)} \exists z.z = x \simeq_{(=4)} x = x$. Case $\varphi \equiv x = y$ where $x \neq y$. By $\varphi_{\mathrm{G}_{x=y}} \simeq_{(*3)} \exists z.z = x * z = y \simeq_{\text{Lemma 21(1)}} x = y$. Case $\varphi \equiv a(x_{f(1)}, \dots, x_{f(n)})$ where $f\colon [n] \to [k]$ is a surjective map for some k. Then,

$$\varphi_{\mathrm{G}_{a(x_{f(1)},\dots,x_{f(n)})}} \equiv \exists z_k \dots z_1.(\mathop{\Large *}_{i=1}^{k} z_i = x_i) * a(z_{f(1)}, \dots, z_{f(n)})$$
$$\simeq_{\text{Lemma 21(1)}} \dots \simeq_{\text{Lemma 21(1)}} a(z_{f(1)}, \dots, z_{f(n)})[x_1 \dots x_k / z_1 \dots z_k] \equiv \varphi.$$

Case $\varphi \equiv \varphi_1 * \varphi_2$. Let G_1 and G_2 be such that $G \cong G_1 \otimes G_2$, $G_1 \models^{\mathrm{GI}} \varphi_1$, $G_2 \models^{\mathrm{GI}} \varphi_2$. By I.H., $\varphi_1 \simeq \varphi_{G_1}$ and $\varphi_2 \simeq \varphi_{G_2}$. We denote them by $\varphi_{G_1} \equiv \exists z_1 \dots z_{n'}. \mathop{\Large *}_{i=1}^{k'} z_{g_1(i)} = x_i * \mathop{\Large *}_{i=1}^{m'} \tilde{\varphi}_i$ and $\varphi_{G_2} \equiv \exists z_{n'+1} \dots z_n. \mathop{\Large *}_{i=1}^{k} z_{g_2(i)} = x_i * \mathop{\Large *}_{i=m'+1}^{m} \tilde{\varphi}_i$, respectively. Here, $g_1\colon [k'] \to [n']$ and $g_2\colon [k] \to [n]$ are some maps. We assume, without loss of generality that $z_1, \dots, z_n$ are pairwise distinct and $k' \leq k$ (by swapping G_1 and G_2 appropriately using $(*2)$). Then,

$$\varphi \simeq_{\text{I.H.}} \varphi_{G_1} \otimes \varphi_{G_2}$$
$$\simeq_{(\exists 1)(\exists 2)(*1)(*2)} \exists z_1 \dots z_n.(\mathop{\Large *}_{i=1}^{k'} z_{g_1(i)} = x_i) * (\mathop{\Large *}_{i=1}^{k} z_{g_2(i)} = x_i) * (\mathop{\Large *}_{i=1}^{m} \tilde{\varphi}_i)$$
$$\simeq_{(*1)(*2)(=3)} \exists z_1 \dots z_n.(\mathop{\Large *}_{i=1}^{k'} z_{g_1(i)} = z_{g_2(i)}) * (\mathop{\Large *}_{i=1}^{k} z_{g_2(i)} = x_i) * (\mathop{\Large *}_{i=1}^{m} \tilde{\varphi}_i)$$
$$\simeq_{\text{Lem. 21(2)}} \exists z_{I_1} \dots z_{I_m}.(\mathop{\Large *}_{i=1}^{k} z_{[g_2(i)]_\sim} = x_i) * (\mathop{\Large *}_{i=1}^{m} \tilde{\varphi}_i[[z_1]_\sim \dots [z_n]_\sim / z_1 \dots z_n]) \simeq \varphi_{G_1 \otimes G_2}$$

Here, $\sim$ and $I_1 \dots I_m$ the ones obtained from Lemma 21(2). Case $\varphi \equiv \exists y.\varphi_1$. Let G_1 be such that $G \cong G_1[\mathtt{f}/y]$ and $G_1 \models \varphi_1$. By I.H., $\varphi_1 \simeq \varphi_{G_1}$. We denote it by $\varphi_{G_1} \equiv \exists z_1 \dots z_n. \mathop{\Large *}_{i=1}^{k} z_{g(i)} = x_i * \mathop{\Large *}_{i=1}^{m} \tilde{\varphi}_i$. Here, $g\colon [k] \to [n]$ is a map, and we assume, without loss of generality that $y, z_1, \dots, z_n$ are pairwise distinct. Then, $y = x_l$ for some $l \in [k]$ (note $y \in \mathsf{FV}_1(\varphi_1)$). We assume, without loss of generality by $(*1)(*2)$ that $y = x_k$. Then, $\varphi \simeq_{\text{I.H.}} \exists y.\varphi_{G_1} \simeq_{(\exists 1)(=1)\ \text{Lem. 21(1)}} \exists z_1 \dots z_n.(\mathop{\Large *}_{i=1}^{k-1} z_{g(i)} = x_i) * (\mathop{\Large *}_{i=1}^{m} \tilde{\varphi}_i) \simeq \varphi_G$. ◀

Proof of Theorem 19 for PP formulas. Assume $\psi \cong^{\mathrm{GI}} \rho$. By Proposition 17(1), $\mathcal{G}(\psi) = \mathcal{G}(\rho) = \{G\}^{\cong}$ for some G. Then, $\psi \simeq_{\text{Lemma 22}} \varphi_G \simeq_{\text{Lemma 22}} \rho$. ◀

In the following, we consider EP formulas.

▶ **Lemma 23.** *If $\{G_1, \dots, G_n\}^{\cong} = \{H_1, \dots, H_m\}^{\cong}$, then $\varphi_{\langle G_i \rangle_{i=1}^n} \simeq \varphi_{\langle H_i \rangle_{i=1}^m}$.*

Proof. By the assumption, let $f\colon [n] \to [m]$ be a map such that $G_i \cong H_{f(i)}$ for every $i \in [n]$. Then, $\varphi_{\langle G_i \rangle_{i=1}^n} \equiv \bigvee_{i=1}^n \varphi_{G_i} \simeq_{\text{Prop. 20}} \bigvee_{i=1}^n \varphi_{H_{f(i)}} \simeq_{(\vee 1)(\vee 2)(\vee 4)} \bigvee_{i=1}^m \varphi_{H_i} \equiv \varphi_{\langle H_i \rangle_{i=1}^m}$. ◀

▶ **Lemma 24.** *For all $\varphi \in \mathsf{EP}_A^\tau$, there exists some $\langle \varphi_i \rangle_{i=1}^n \in (\mathsf{PP}_A^\tau)^*$ such that $\varphi \simeq \bigvee_{i=1}^n \varphi_i$.*

Proof. By induction on the structure of φ. Case $\varphi \equiv \mathsf{ff}$. By letting $n = 0$. Case $\varphi \equiv \tilde{\varphi}$. By letting $n = 1$. Case $\varphi \equiv \varphi^{(1)} * \varphi^{(2)}$. For $l \in [2]$, let $\langle \varphi_i^{(l)} \rangle_{i=1}^{n_l}$ be the one obtained by I.H. w.r.t. $\varphi^{(l)}$. If $n_1 = 0$ or $n_2 = 0$, then $\varphi \simeq_{(*2)(\mathsf{ff})} \mathsf{ff}$. Otherwise, $\varphi \simeq_{(\vee 1)(\vee 2)(\vee 6)} \bigvee_{i=1}^{n_1} \bigvee_{j=1}^{n_2} (\varphi_i^{(1)} * \varphi_j^{(2)})$ (and apply $(\vee 1)(\vee 2)$). Case $\varphi \equiv \varphi^{(1)} \vee \varphi^{(2)}$. Let $\langle \varphi_i \rangle_{i=1}^{n'}$ and $\langle \varphi_i \rangle_{i=n'+1}^{n}$ be the ones obtained by I.H. w.r.t. $\varphi^{(1)}$ and $\varphi^{(2)}$, respectively. Then, $\varphi \simeq_{(\vee 1)(\vee 2)(\vee 3)} \bigvee_{i=1}^n \varphi_i$. Case $\varphi \equiv \exists x.\varphi^{(1)}$. Let $\langle \varphi_i^{(1)} \rangle_{i=1}^n$ be the one obtained by I.H. w.r.t. $\varphi^{(1)}$. If $n = 0$, then $\varphi \equiv \exists x.\mathsf{ff} \simeq_{(\mathsf{ff})} \exists x.\mathsf{ff} * \mathsf{ff} \simeq_{(\exists 2)} (\exists x.\mathsf{ff}) * \mathsf{ff} \simeq_{(*2)(\mathsf{ff})} \mathsf{ff}$. Otherwise, $\varphi \equiv \exists x. \bigvee_{i=1}^n \varphi_i^{(1)} \simeq_{(\vee 5)} \bigvee_{i=1}^n \exists x.\varphi_i^{(1)}$. ◀

▶ **Lemma 25.** *For every* EP *formula* φ *and finite sequence* $\vec{G}$ *s.t.* $\mathcal{G}(\varphi) = \mathrm{Occ}(\vec{G})^{\cong}$, $\varphi \simeq \varphi_{\vec{G}}$.

Proof. By $\varphi \simeq_{\text{Lemma 24}} \bigvee_{i=1}^{n} \varphi_i \simeq_{\text{Lemma 22}} \bigvee_{i=1}^{n} \varphi_{G_i} \simeq_{\text{Lemma 23}} \varphi_{\vec{G}}$. Here, for each $i \in [n]$, φ_i is a PP formula and G_i is a graph such that $G_i \models^{\mathrm{GI}} \varphi_i$. ◀

Proof of Theorem 19 for EP formulas. Assume $\psi \cong^{\mathrm{GI}} \rho$. Let $\vec{G}$ be a finite sequence such that $\mathcal{G}(\psi) = \mathcal{G}(\rho) = \mathrm{Occ}(\vec{G})^{\cong}$. Then, $\psi \simeq_{\text{Lemma 25}} \varphi_{\vec{G}} \simeq_{\text{Lemma 25}} \rho$. ◀

5 Kleene Theorems Between EPs and HRGs

In this section, we show that EP(LFP) (resp. EP(TC)) has the same expressive power as the class of HRGs (resp. linear HRGs). To this end, we introduce *term (formula) rewriting systems* [2] (FRSs) and show the equivalence above via FRSs. Intuitively, FRSs play the same role as finite automata with transitions labelled by regular expressions [7] (so-called *extended finite automata*) in translating finite automata into regular expressions.[4]

5.1 Formula Rewriting Systems (FRSs)

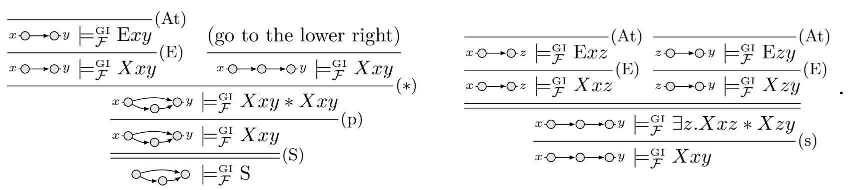
▶ **Definition 26.** *A* formula rewriting system (FRS[$\mathscr{C}$]) $\mathcal{F}$ *over an ordinal-typed alphabet* A *is a tuple* $\langle \mathcal{X}^{\mathcal{F}}, \mathcal{R}^{\mathcal{F}}, \mathfrak{s}^{\mathcal{F}} \rangle$, *where* $\mathcal{X}^{\mathcal{F}}$ *is an ordinal-typed alphabet disjoint with* A *for denoting* (non-terminal) labels, $\mathcal{R}^{\mathcal{F}}$ *is a finite set of pairs* $r = \langle X\vec{x}, \varphi \rangle$ *(written* $X\vec{x} \leftarrow \varphi$*) of a strictly atomic* $\mathcal{X}^{\mathcal{F}}$*-formula* $X\vec{x}$ *and a* $\mathscr{C}_{A \cup \mathcal{X}^{\mathcal{F}}}^{\mathrm{Occ}(\vec{x})}$*-formula* φ *for denoting* rewriting rules, *and* $\mathfrak{s}^{\mathcal{F}}$ *is a strictly atomic* $\mathcal{X}^{\mathcal{F}}$*-formula for denoting the* source formula. *Here, for an ordinal-typed alphabet* $\mathcal{X}$, *we say that* φ *is a* strictly atomic $\mathcal{X}$-formula *if* φ *is of the form* $X\vec{x}$, *where* $X \in \mathcal{X}$ *and the elements of* $\vec{x}$ *are pairwise distinct.*

▶ **Definition 27.** *For an* FRS[$\mathscr{C}$] $\mathcal{F} = \langle \mathcal{X}, \mathcal{R}, \mathfrak{s} \rangle$ *over an ordinal-typed alphabet* A, *the binary relation* $\models_{\mathcal{F}}^{\mathrm{GI}} \subseteq \bigcup_{\tau \subseteq \mathscr{V}_1;\, \mathcal{X} \subseteq A \cup \mathscr{V}_2} \mathsf{GR}_{\mathcal{X}}^{\tau} \times \mathsf{Fml}_{\mathcal{X}}^{\tau}$ *is defined as the least* $\cong$*-closed relation closed under all the rules of* $\models^{\mathrm{GI}}$ *(in Definition 14) and the following rule: If* $X\vec{x} \leftarrow \varphi \in \mathcal{R}$, *then* $\dfrac{G \models_{\mathcal{F}}^{\mathrm{GI}} \varphi[\vec{y}/\vec{x}]}{G \models_{\mathcal{F}}^{\mathrm{GI}} X\vec{y}}$. *We write* $G \models^{\mathrm{GI}} \mathcal{F}$ *for* $G \models_{\mathcal{F}}^{\mathrm{GI}} \mathfrak{s}$. *The graph language of* $\mathcal{F}$ *is defined by* $\mathcal{G}(\mathcal{F}) \triangleq \{G \mid G \models^{\mathrm{GI}} \mathcal{F}\}$.

▶ **Example 28** (cf. Example 12)**.** Let $\mathcal{F}$ be the FRS[PP] over A_{E}, defined by $\mathrm{ty}^{\mathcal{X}^{\mathcal{F}}} = \{\mathsf{S} \mapsto [0], X \mapsto [2]\}$, $\mathcal{R}^{\mathcal{F}} = \{(\mathsf{S}), (\mathrm{E}), (\mathrm{s}), (\mathrm{p})\}$, $\mathfrak{s}^{\mathcal{F}} = \mathsf{S}$, where each rule in $\mathcal{R}^{\mathcal{F}}$ is as follows:

$$(\mathsf{S})\ \mathsf{S} \leftarrow \exists xy.Xxy \qquad (\mathrm{E})\ Xxy \leftarrow \mathrm{E}xy \qquad (\mathrm{s})\ Xxy \leftarrow \exists z.Xxz * Xzy \qquad (\mathrm{p})\ Xxy \leftarrow Xxy * Xxy$$

Then, $\mathcal{G}(\mathcal{F})$ is the set of all series-parallel graphs. For example, [graph] $\models^{\mathrm{GI}} \mathcal{F}$ is shown by:[5]



In general, the following proposition is immediate from the translations between graphs and PP formulas in Proposition 17(1). Also, we use *linear/(n-)recursive* for FRS[PP]s in the same manner as for HRGs.

[4] FRS[$\mathscr{C}$] is essentially the same as *positive Datalog* [20, Section 9] if $\mathscr{C}$ is the class of conjunctive queries.
[5] Double line denotes that 0 or more rules are applied in the place.

▶ **Proposition 29.** *For every $\mathcal{G}$, some HRG (resp. linear HRG) recognizes $\mathcal{G}$ iff some* FRS[PP] *(resp. linear* FRS[PP]*) recognizes $\mathcal{G}$.*

An FRS $\mathcal{F}$ is *deterministic* if for every $X \in \mathcal{X}^{\mathcal{F}}$, the number of rules of the form $X\vec{x} \leftarrow \varphi$ is at most one. In Example 28, we can put together the three rules for X as follows in FRS[EP]:

$$(\mathtt{S})\ \mathtt{S} \leftarrow \exists xy.Xxy \qquad (X)\ Xxy \leftarrow (\mathrm{E}xy) \vee (Xxy * Xxy) \vee (\exists z.Xxz * Xzy).$$

▶ **Proposition 30.** *For every $\mathcal{G}$,* (i) *some* FRS[PP] *recognizes $\mathcal{G}$ iff* (ii) *some deterministic* FRS[EP] *recognizes $\mathcal{G}$ iff* (iii) *some* FRS[EP] *recognizes $\mathcal{G}$.*

Proof. (i) ⇒ (ii): By the same argument as above. (ii) ⇒ (iii): Trivial. (iii) ⇒ (i): By replacing each rule $X\vec{x} \leftarrow \varphi$ with $X\vec{x} \leftarrow \psi_1, \ldots, X\vec{x} \leftarrow \psi_n$. Here, $\psi_1, \ldots \psi_n$ are PP formulas such that $\varphi \cong^{\mathrm{GI}} \bigvee_{i=1}^{n} \psi_i$ (Lemma 24). ◀

The following are useful properties of hyperedge replacing and glueing.

▶ **Proposition 31.** *For every* FRS[EP(LFP)] $\mathcal{F}$*: (1): If there is a derivation tree that shows $G \models^{\mathrm{GI}}_{\mathcal{F}} \varphi$ from the assumptions $\langle H_i \models^{\mathrm{GI}}_{\mathcal{F}} \psi_i \rangle_{i=1}^{n}$ and $H_1, \ldots, H_n$ don't contain any $\mathsf{FV}_2(\varphi)$-labelled edges and have an ordinal type, then there exist some G' and $e_1 \ldots e_n$ such that $G \cong G'[H_1 \ldots H_n/e_1 \ldots e_n]$. (2): If there is a derivation tree that shows $G[H_1 \ldots H_n/\vec{e}] \models^{\mathrm{GI}}_{\mathcal{F}} \varphi$ from the assumptions $\langle H_i \models^{\mathrm{GI}}_{\mathcal{F}} \psi_i \rangle_{i=1}^{n}$ and $H_1, \ldots, H_n, H'_1, \ldots, H'_n$ don't contain any $\mathsf{FV}_2(\varphi)$-labelled edges and have an ordinal type, then there is a derivation tree that shows $G[H'_1 \ldots H'_n/\vec{e}] \models^{\mathrm{GI}}_{\mathcal{F}} \varphi$ from the assumptions $\langle H'_i \models^{\mathrm{GI}}_{\mathcal{F}} \psi_i \rangle_{i=1}^{n}$. For every* FRS[EP(TC)] $\mathcal{F}$*: (3): If there is a derivation tree that shows $G \models^{\mathrm{GI}}_{\mathcal{F}} \varphi$ from $H \models^{\mathrm{GI}}_{\mathcal{F}} \psi$ and $\mathrm{ty}(H) \cap \mathsf{BV}_1(\varphi) = \emptyset$, then there exist some G' such that $G \cong G' \otimes H$. (4): If there is a derivation tree that shows $G \otimes H \models^{\mathrm{GI}}_{\mathcal{F}} \varphi$ from $G' \otimes H \models^{\mathrm{GI}}_{\mathcal{F}} \psi$, $\mathrm{ty}(H) \cap \mathsf{BV}_1(\varphi) = \emptyset$, $\mathrm{ty}(H') \cap \mathsf{BV}_1(\varphi) = \emptyset$, and $\mathrm{ty}(H) = \mathrm{ty}(H')$, then there is a derivation tree that shows $G \otimes H' \models^{\mathrm{GI}}_{\mathcal{F}} \varphi$ from $G' \otimes H' \models^{\mathrm{GI}}_{\mathcal{F}} \psi$.*

Proof Sketch. By a straightforward induction on the structure of the derivation tree using Proposition 9. See [33, Appendix B] for more details. ◀

5.2 Equivalence of EP(LFP) formulas and HRGs (Theorem 1(2))

In the following, by using Proposition 29 and 30, we show that EP(LFP) has the same expressive power as (deterministic) FRS[EP].

From EP(LFP) formulas to FRS[EP]s. We say that an EP(LFP) formula φ is *simple* if (a) all the second-order variables X occurring in the form $[\mathsf{LFP}_{\vec{x},X}(\varphi)]\vec{y}$ are pairwise distinct, (b) $\vec{x} = \vec{y} = \iota$ for each subformula of the form $[\mathsf{LFP}_{\vec{x},X}(\varphi)]\vec{y}$, and (c) $\vec{x} = \iota$ for each subformula of the form $X\vec{x}$. This restriction simplifies the translation and the proof.

▶ **Lemma 32.** *Every* EP(LFP) *formula φ has a GI-equivalent simple* EP(LFP) *formula.*

Proof Sketch. For (a), rename variables appropriately. For (b)(c), use the following translations, respectively: $[\mathsf{LFP}_{\vec{x},X}(\varphi)]\vec{y} \rightsquigarrow \exists \vec{z}.\vec{z} = \vec{y} * \exists \iota.\iota = \vec{z} * [\mathsf{LFP}_{\iota,X}(\exists \vec{z}.\vec{z} = \iota * \exists \vec{x}.\vec{x} = \vec{z} * \varphi)]\iota$ and $X\vec{x} \rightsquigarrow \exists \vec{z}.\vec{z} = \vec{x} * \exists \iota.\iota = \vec{z} * X\iota$. Here, $\vec{z}$ is a sequence of fresh variables. ◀

Let $\vec{z}_{\bullet}$ be a map from each EP(LFP) formula φ to a permutation $\vec{z}_{\varphi}$ of $\mathsf{FV}_1(\varphi)$. Figure 4 gives a translation from a simple EP(LFP) formula φ into an FRS[EP] $\mathcal{F}_{\varphi} = \langle \mathcal{X}_{\varphi}, \mathcal{R}_{\varphi}, \mathfrak{s}_{\varphi} \rangle$.[6]

[6] This translation is essentially the same as the translation from existential fixpoint logic to Datalog, see, e.g., [20, Theorem 9.1.4]. The only difference is the semantics.

$$\mathcal{F}_{\tilde{\varphi}} \triangleq \langle \{\mathsf{S}_\varphi\}, \{\mathfrak{s}_\varphi \leftarrow \tilde{\varphi}\}, \mathsf{S}_\varphi \vec{z}_\varphi \rangle \quad \mathcal{F}_{\exists x.\psi} \triangleq \langle \{\mathsf{S}_\varphi\} \cup \mathcal{X}_\psi, \{\mathfrak{s}_\varphi \leftarrow \exists x.\mathfrak{s}_\psi\} \cup \mathcal{R}_\psi, \mathsf{S}_\varphi \vec{z}_\varphi \rangle$$

$$\mathcal{F}_{\psi \bullet \rho} \triangleq \langle \{\mathsf{S}_\varphi\} \cup \mathcal{X}_\psi \cup \mathcal{X}_\rho, \{\mathfrak{s}_\varphi \leftarrow \mathfrak{s}_\psi \bullet \mathfrak{s}_\rho\} \cup \mathcal{R}_\psi \cup \mathcal{R}_\rho, \mathsf{S}_\varphi \vec{z}_\varphi \rangle \quad (\bullet \in \{*, \vee\})$$

$$\mathcal{F}_{[\mathsf{LFP}_{\iota,X}(\psi)]\iota} \triangleq \langle \{\mathsf{S}_\varphi, X\} \cup \mathcal{X}_\psi, \{\mathfrak{s}_\varphi \leftarrow X\iota,\ X\iota \leftarrow \mathfrak{s}_\psi\} \cup \mathcal{R}_\psi, \mathsf{S}_\varphi \vec{z}_\varphi \rangle$$

Figure 4 A translation from EP(LFP) formulas into (deterministic) FRS[EP]s.

▶ **Lemma 33.** *For every simple* EP(LFP) *formula* φ, $\mathcal{G}(\varphi) = \mathcal{G}(\mathcal{F}_\varphi)$.

Proof. $G \models^{\mathrm{GI}} \varphi \Rightarrow G \models^{\mathrm{GI}}_{\mathcal{F}_\varphi} \mathfrak{s}_\varphi$: By induction on the size of the derivation tree of $G \models^{\mathrm{GI}} \varphi$. The only nontrivial case is when the last derivation rule is (LFP). Let $\varphi = [\mathsf{LFP}_{\iota,X}(\psi)]\iota$ (by the condition (b)) and let $G \cong H[G_1 \dots G_n/e_1 \dots e_n]$ be such that $H \models^{\mathrm{GI}} \psi$ and $G_i \models^{\mathrm{GI}} \varphi$ for $i \in [n]$. By I.H., $H \models^{\mathrm{GI}}_{\mathcal{F}_\psi} \mathfrak{s}_\psi$. Its derivation tree forms the left-hand side in the following (by the condition (c)). Also for $i \in [n]$, by I.H., $G_i \models^{\mathrm{GI}}_{\mathcal{F}_\varphi} \mathfrak{s}_\varphi$, so by the construction of $\mathcal{F}_\varphi$, ($\heartsuit$-i) $G_i \models^{\mathrm{GI}}_{\mathcal{F}_\varphi} X\iota$. Then, $G \models^{\mathrm{GI}}_{\mathcal{F}_\varphi} \mathfrak{s}_\varphi$ is shown by the right-hand side tree (Proposition 31(2)).

$$\dfrac{\dfrac{}{\mathrm{G}_{X\iota} \models^{\mathrm{GI}}_{\mathcal{F}_\psi} X\iota} \quad \dots \quad \dfrac{}{\mathrm{G}_{X\iota} \models^{\mathrm{GI}}_{\mathcal{F}_\psi} X\iota}}{\vdots\,(\spadesuit) \atop H \models^{\mathrm{GI}}_{\mathcal{F}_\psi} \mathfrak{s}_\psi} \quad \rightsquigarrow \quad \dfrac{\dfrac{\dfrac{}{\vdots\,(\heartsuit\text{-}1)}}{\mathrm{G}_{X\iota}[G_1/\mathrm{e}] \models^{\mathrm{GI}}_{\mathcal{F}_\varphi} X\iota} \quad \dots \quad \dfrac{\dfrac{}{\vdots\,(\heartsuit\text{-}n)}}{\mathrm{G}_{X\iota}[G_n/\mathrm{e}] \models^{\mathrm{GI}}_{\mathcal{F}_\varphi} X\iota}}{\vdots\,(\spadesuit) \atop \dfrac{H[G_1 \dots G_n/e_1 \dots e_n] \models^{\mathrm{GI}}_{\mathcal{F}_\varphi} \mathfrak{s}_\psi}{H[G_1 \dots G_n/e_1 \dots e_n] \models^{\mathrm{GI}}_{\mathcal{F}_\varphi} \mathfrak{s}_\varphi}(\mathfrak{s}_\varphi \leftarrow X\iota)(X\iota \leftarrow \mathfrak{s}_\psi)} \; .$$

$G \models^{\mathrm{GI}}_{\mathcal{F}_\varphi} \mathfrak{s}_\varphi \Rightarrow G \models^{\mathrm{GI}} \varphi$: By induction on the size of the derivation tree of $G \models^{\mathrm{GI}}_{\mathcal{F}_\varphi} \mathfrak{s}_\varphi$. We do a case analysis on the structure of φ. The only nontrivial case is when $\varphi = [\mathsf{LFP}_{\iota,X}(\psi)]\iota$. The derivation tree of $G \models^{\mathrm{GI}}_{\mathcal{F}_\varphi} \mathfrak{s}_\varphi$ should form the right-hand side above, where the rule for X is not applied in ($\spadesuit$). Note that $G \cong H[G_1 \dots G_n/e_1 \dots e_n]$ for some H and $e_1 \dots e_n$ (Proposition 31(1)). Then, from the derivation tree, we can obtain the derivation tree of the form on the left-hand side above (Proposition 31(2)). Thus by I.H., $H \models^{\mathrm{GI}} \psi$. Also by using ($\heartsuit$-$i$), $G_i \models^{\mathrm{GI}}_{\mathcal{F}_\varphi} \mathfrak{s}_\varphi$, and thus by I.H., $G_i \models^{\mathrm{GI}} \varphi$. Hence, $G \models^{\mathrm{GI}} \varphi$. ◀

Proof of Theorem 1(2)⇒. By Lemma 32 and 33 (with Proposition 29 and 30). ◀

From FRS[EP]s to EP(LFP) formulas. This part is shown by folding non-terminal labels for a given *deterministic* FRS[EP] as follows: for non-0-recursive labels X, replace each occurrence of X with the formula corresponding to X in the rule; for 0-recursive labels, use the LFP. Note that by Proposition 30, from an FRS[EP], we can obtain a deterministic one.

▶ **Lemma 34.** *Every deterministic* FRS[EP(LFP)] *has a GI-equivalent* EP(LFP) *formula.*

Proof. Let $\mathcal{F} = \langle \mathcal{X}, \mathcal{R}, \mathsf{S}\vec{z} \rangle$. Let $\#_{\mathrm{n}}(\mathcal{F}) \triangleq \#(\mathcal{X} \setminus \{\mathsf{S}\})$ and $\#_{\mathrm{r}}(\mathcal{F})$ be the number of 0-recursive labels in $\mathcal{F}$. We prove by induction on the pair $\langle \#_{\mathrm{n}}(\mathcal{F}), \#_{\mathrm{r}}(\mathcal{F}) \rangle$. Case $\#_{\mathrm{n}}(\mathcal{F}) = \#_{\mathrm{r}}(\mathcal{F}) = 0$. Let $\mathcal{R} = \{\mathsf{S}\vec{x} \leftarrow \psi\}$. Then, $\mathcal{G}(\mathcal{F}) = \mathcal{G}(\psi[\vec{z}/\vec{x}])$. Case $\#_{\mathrm{n}}(\mathcal{F}) > \#_{\mathrm{r}}(\mathcal{F})$. Then, there exists a non-0-recursive label $X_0 \in \mathcal{X} \setminus \{\mathsf{S}\}$. Let $X_0 \vec{x}_0 \leftarrow \psi_0 \in \mathcal{R}$. Let $\mathcal{F}' \triangleq \langle \mathcal{X} \setminus \{X_0\}, \{X\vec{x} \leftarrow \psi[\psi_0[-/\vec{x}_0]/X_0-] \mid X\vec{x} \leftarrow \psi \in \mathcal{R}, X \neq X_0\}, \mathsf{S}\vec{z} \rangle$, where $\psi[\psi_0[-/\vec{x}_0]/X_0-]$ denotes the formula ψ in which each $X_0\vec{y}$ has been replaced with $\psi_0[\vec{y}/\vec{x}_0]$. Then, $\mathcal{G}(\mathcal{F}) = \mathcal{G}(\mathcal{F}')$ because there is a trivial transformation between derivation trees of $\mathcal{F}$ and those of $\mathcal{F}'$. Also by I.H., there exists an EP(LFP) formula φ such that $\mathcal{G}(\mathcal{F}') = \mathcal{G}(\varphi)$. Hence, $\mathcal{G}(\mathcal{F}) = \mathcal{G}(\varphi)$. For the other case (i.e., $\#_{\mathrm{r}}(\mathcal{F}) \geq 1$), there exists a 0-recursive label $X_0 \in \mathcal{X}$. Let

$X_0\vec{x}_0 \leftarrow \psi_0 \in \mathcal{R}$. Let $\mathcal{F}' \triangleq \langle \mathcal{X}, \{X\vec{x} \leftarrow \psi \in \mathcal{R} \mid X \neq X_0\} \cup \{X_0\vec{x}_0 \leftarrow [\mathsf{LFP}_{\vec{x}_0,X_0}(\psi_0)]\vec{x}_0\}, \mathtt{S}\vec{z}\rangle$. Then, $\mathcal{G}(\mathcal{F}) = \mathcal{G}(\mathcal{F}')$ because there exists a transformation between derivation trees of $\mathcal{F}'$ and those of $\mathcal{F}$ in the same manner as the proof of Lemma 33. Also by I.H., there exists an $\mathsf{EP}(\mathsf{LFP})$ formula φ such that $\mathcal{G}(\mathcal{F}') = \mathcal{G}(\varphi)$. Hence, $\mathcal{G}(\mathcal{F}) = \mathcal{G}(\varphi)$. ◀

Proof of Theorem 1(2)⇐. By Lemma 34 (with Proposition 29 and 30). ◀

5.3 Equivalence of EP(TC) formulas and linear HRGs (Theorem 1(3)).

In the following, by using Proposition 29 and 30, we show that EP(TC) has the same expressive power as the class of linear FRS[PP].

From EP(TC) formulas to linear FRS[PP]s. We say that an EP(TC) formula φ is *simple* if all the variables x occurring in the form $\exists x.\psi$, the variables in $\vec{x}\vec{y}\vec{u}\vec{w}$ occurring in the form $[\varphi]^+_{\vec{x}\vec{y}}\vec{u}\vec{w}$, and the free variables in φ are pairwise distinct. As with Lemma 32, from a given EP(TC) formula, we can obtain a GI-equivalent simple one by renaming variables and using the following translation: $[\varphi]^+_{\vec{x}\vec{y}}\vec{u}\vec{w} \rightsquigarrow \exists \vec{z}.\vec{z} = \vec{u}\vec{w} * [\varphi[\vec{z}'/\vec{x}\vec{y}]]^+_{\vec{z}'}\vec{z}$. Here, elements of $\vec{z}$ and $\vec{z}'$ are fresh variables. Furthermore, the following holds.

▶ **Lemma 35.** *Every* EP(TC) *formula φ has a GI-equivalent simple* EP(TC) *formula of the form $\exists z_0.\varphi_0$ or $\top \vee \exists z_0.\varphi_0$.*

Proof. If $\mathsf{FV}_1(\varphi) \neq \emptyset$, then $\varphi \cong^{\mathrm{GI}} \exists z_0.z_0 = x * \varphi$, where $x \in \mathsf{FV}_1(\varphi)$ and z_0 is a fresh variable. Otherwise, let $\bigvee_{i=1}^{n} \varphi_i$ be a disjunctive normal form of φ, where each φ_i is a prenex normal form EP(TC) formula. Let $\rho_i \equiv \exists z_0.\psi_i$ if φ_i is of the form $\exists x.\psi_i$ and $\rho_i \equiv \top$ otherwise (note that then $\varphi_i \equiv \top$ should because $\mathsf{FV}_1(\varphi_i) = \emptyset$). Note that $\varphi_i \cong^{\mathrm{GI}} \rho_i$. Let $l_1 \dots l_m$ be the subsequence of ι_n such that for each $i \in [n]$, $i \in \{l_1, \dots, l_m\}$ iff $\rho_i \not\equiv \top$. If $m < n$, then $\varphi \cong^{\mathrm{GI}} \top \vee \bigvee_{j=1}^{m} \exists z_0.\psi_{l_j}$ $(\cong^{\mathrm{GI}} \top \vee \exists z_0. \bigvee_{j=1}^{m} \psi_{l_j})$. Otherwise, $\varphi \cong^{\mathrm{GI}} \bigvee_{i=1}^{n} \exists z_0.\psi_i$ $(\cong^{\mathrm{GI}} \exists z_0. \bigvee_{i=1}^{n} \psi_i)$. Hence, it has been proved. ◀

Let $\vec{z}$ be a sequence of pairwise distinct variables. For a simple EP(TC) formula φ such that $\mathsf{V}_1(\varphi) \subseteq \mathrm{Occ}(\vec{z})$, we define the linear FRS[PP] $\dot{\mathcal{F}}_\varphi = \langle \mathcal{X}_\varphi, \mathcal{R}_\varphi, \mathfrak{s}_\varphi\rangle$ (we may explicitly write $\dot{\mathcal{F}}^{\vec{z}}_\varphi = \langle \mathcal{X}^{\vec{z}}_\varphi, \mathcal{R}^{\vec{z}}_\varphi, \mathfrak{s}^{\vec{z}}_\varphi\rangle$) in Figure 5. Our construction is based on Thompson's construction [42] and the product construction (in translating regular expressions into finite automata), but is generalized for first-order variables.

$$\dot{\mathcal{F}}_{\tilde{\varphi}} \triangleq \langle \{\mathtt{S}_\varphi, \mathtt{T}_\varphi\}, \{\mathtt{S}_\varphi\vec{z} \leftarrow \tilde{\varphi} * \mathtt{T}_\varphi\vec{z}\}, \mathtt{S}_\varphi\vec{z}\rangle$$

$$\dot{\mathcal{F}}_{\exists x.\psi} \triangleq \langle \{\mathtt{S}_\varphi, \mathtt{T}_\varphi\} \cup \mathcal{X}_\psi, \{\mathtt{S}_\varphi\vec{z} \leftarrow x = x * \exists x.\mathtt{S}_\psi\vec{z}, \mathtt{T}_\psi\vec{z} \leftarrow \mathtt{T}_\varphi\vec{z}\} \cup \mathcal{R}_\psi, \mathtt{S}_\varphi\vec{z}\rangle$$

$$\dot{\mathcal{F}}_{\psi * \rho} \triangleq \langle \{\mathtt{S}_\varphi, \mathtt{T}_\varphi\} \cup (\mathcal{X}_\psi \times \mathcal{X}_\rho), \{\mathtt{S}_\varphi\vec{z} \leftarrow \langle \mathtt{S}_\psi, \mathtt{S}_\rho\rangle\vec{z}, \langle \mathtt{T}_\psi, \mathtt{T}_\rho\rangle\vec{z} \leftarrow \mathtt{T}_\varphi\vec{z}\} \cup \{r[\langle -, Y\rangle/-] \mid r \in \mathcal{R}_\psi, Y \in \mathcal{X}_\rho\} \cup \{r[\langle X, -\rangle/-] \mid r \in \mathcal{R}_\rho, X \in \mathcal{X}_\psi\}, \mathtt{S}_\varphi\vec{z}\rangle^{\dagger 1}$$

$$\dot{\mathcal{F}}_{\psi \vee \rho} \triangleq \langle \{\mathtt{S}_\varphi, \mathtt{T}_\varphi\} \cup \mathcal{X}_\psi \cup \mathcal{X}_\rho, \{\mathtt{S}_\varphi\vec{z} \leftarrow \mathtt{S}_\psi\vec{z}, \mathtt{S}_\varphi\vec{z} \leftarrow \mathtt{S}_\rho\vec{z}, \mathtt{T}_\psi\vec{z} \leftarrow \mathtt{T}_\varphi\vec{z}, \mathtt{T}_\rho\vec{z} \leftarrow \mathtt{T}_\varphi\vec{z}\} \cup \mathcal{R}_\psi \cup \mathcal{R}_\rho, \mathtt{S}_\varphi\vec{z}\rangle$$

$$\dot{\mathcal{F}}_{[\psi]^+_{\vec{x}\vec{y}}\vec{u}\vec{w}} \triangleq \langle \{\mathtt{S}_\varphi, \mathtt{T}_\varphi\} \cup \mathcal{X}_\psi, \{\mathtt{S}_\varphi\vec{z} \leftarrow \vec{x}\vec{y} = \vec{x}\vec{y} * \exists \vec{x}.\vec{x} = \vec{u} * \exists \vec{y}.\mathtt{S}_\psi\vec{z}\} \cup \{\mathtt{T}_\psi\vec{z} \leftarrow \vec{x}\vec{y} = \vec{x}\vec{y} * \exists \vec{x}.\vec{x} = \vec{y} * \exists \vec{y}.\mathtt{S}_\psi\vec{z}, \mathtt{T}_\psi\vec{z} \leftarrow \vec{y} = \vec{w} * \mathtt{T}_\varphi\vec{z}\} \cup \mathcal{R}_\psi, \mathtt{S}_\varphi\vec{z}\rangle$$

†1: $r[\langle -, Y\rangle/-]$ (resp. $r[\langle X, -\rangle/-]$) is the rule r in which each X (resp. Y) has been replaced with $\langle X, Y\rangle$.

Figure 5 Definition of linear FRS[PP] $\dot{\mathcal{F}}_\varphi$.

▶ **Lemma 36.** *For every simple* $\mathsf{EP}(\mathsf{TC})$ *formula* φ *and every* $G \in \mathsf{GR}^{\tau}_{A}$ *(where* $\varphi \in \mathsf{Fml}^{\tau}_{A}$*),* $G \models^{\mathrm{GI}} \varphi$ *iff there is a derivation tree that shows* $G \otimes \mathrm{G}^{\mathrm{Occ}(\vec{z})}_{\top} \models^{\mathrm{GI}}_{\dot{\mathcal{F}}^{\vec{z}}_{\varphi}} \mathtt{S}_{\varphi}\vec{z}$ *from* $\mathrm{G}^{\mathrm{Occ}(\vec{z})}_{\top} \models^{\mathrm{GI}}_{\dot{\mathcal{F}}^{\vec{z}}_{\varphi}} \mathtt{T}_{\varphi}\vec{z}$*.*

Proof. $\Rightarrow$: By induction on the structure of φ. The essential case is when $\varphi = [\psi]^{+}_{\vec{x}\vec{y}}\vec{u}\vec{w}$. Let $G \cong (G_1 \odot_{\vec{y}\vec{x}} \dots \odot_{\vec{y}\vec{x}} G_n)[\vec{u}\vec{w}/\vec{x}\vec{y}]$ be such that $G_i \models^{\mathrm{GI}} \psi$ for $i \in [n]$. For notational simplicity, let $G_{[i,n]} \triangleq G_i \odot_{\vec{y}\vec{x}} \dots \odot_{\vec{y}\vec{x}} G_n[\vec{w}/\vec{y}]$ for $i \in [n]$. Note that $G \cong G_{[1,n]}[\vec{u}/\vec{x}]$ and $G_{[i,n]} \cong (G_i \otimes G_{[i+1,n]}[\vec{y}/\vec{x}])[\mathtt{f}\dots\mathtt{f}/\vec{y}]$. For each $i \in [n]$, by I.H., there is a derivation tree ($\clubsuit$-i) that shows $G_i \otimes \mathrm{G}^{\mathrm{Occ}(\vec{z})}_{\top} \models^{\mathrm{GI}}_{\dot{\mathcal{F}}^{\vec{z}}_{\psi}} \mathtt{S}_{\psi}\vec{z}$ from $\mathrm{G}^{\mathrm{Occ}(\vec{z})}_{\top} \models^{\mathrm{GI}}_{\dot{\mathcal{F}}^{\vec{z}}_{\psi}} \mathtt{T}_{\psi}\vec{z}$. Then, we obtain a derivation tree that shows $G \otimes \mathrm{G}^{\mathrm{Occ}(\vec{z})}_{\top} \models^{\mathrm{GI}}_{\dot{\mathcal{F}}^{\vec{z}}_{\varphi}} \mathtt{S}_{\varphi}\vec{z}$ from $\mathrm{G}^{\mathrm{Occ}(\vec{z})}_{\top} \models^{\mathrm{GI}}_{\dot{\mathcal{F}}^{\vec{z}}_{\varphi}} \mathtt{T}_{\varphi}\vec{z}$ by concatenating ($\clubsuit$-1)-($\clubsuit$-n) using Proposition 31(4) as follows.

$$
\dfrac{\dfrac{\begin{array}{c}\dfrac{\text{(go to the lower right)}}{G_{[2,n]}[\vec{y}/\vec{x}] \otimes \mathrm{G}^{\mathrm{Occ}(\vec{z})}_{\top} \models^{\mathrm{GI}}_{\dot{\mathcal{F}}^{\vec{z}}_{\varphi}} \mathtt{T}_{\psi}\vec{z}}\\ \vdots\ (\clubsuit\text{-}1)\end{array}}{G_1 \otimes G_{[2,n]}[\vec{y}/\vec{x}] \otimes \mathrm{G}^{\mathrm{Occ}(\vec{z})}_{\top} \models^{\mathrm{GI}}_{\dot{\mathcal{F}}^{\vec{z}}_{\varphi}} \mathtt{S}_{\psi}\vec{z}}}{G_{[1,n]}[\vec{u}/\vec{x}] \otimes \mathrm{G}^{\mathrm{Occ}(\vec{z})}_{\top} \models^{\mathrm{GI}}_{\dot{\mathcal{F}}^{\vec{z}}_{\varphi}} \mathtt{S}_{\varphi}\vec{z}}
\quad \dots \quad
\dfrac{\dfrac{\begin{array}{c}\dfrac{\text{(go to the lower right)}}{\mathrm{G}_{\vec{y}=\vec{w}} \otimes \mathrm{G}^{\mathrm{Occ}(\vec{z})}_{\top} \models^{\mathrm{GI}}_{\dot{\mathcal{F}}^{\vec{z}}_{\varphi}} \mathtt{T}_{\psi}\vec{z}}\\ \vdots\ (\clubsuit\text{-}n)\end{array}}{G_n \otimes \mathrm{G}_{\vec{y}=\vec{w}} \otimes \mathrm{G}^{\mathrm{Occ}(\vec{z})}_{\top} \models^{\mathrm{GI}}_{\dot{\mathcal{F}}^{\vec{z}}_{\varphi}} \mathtt{S}_{\psi}\vec{z}}}{G_{[n,n]}[\vec{y}/\vec{x}] \otimes \mathrm{G}^{\mathrm{Occ}(\vec{z})}_{\top} \models^{\mathrm{GI}}_{\dot{\mathcal{F}}^{\vec{z}}_{\varphi}} \mathtt{T}_{\psi}\vec{z}}
\quad
\dfrac{\mathrm{G}^{\mathrm{Occ}(\vec{z})}_{\top} \models^{\mathrm{GI}}_{\dot{\mathcal{F}}^{\vec{z}}_{\varphi}} \mathtt{T}_{\varphi}\vec{z}}{\mathrm{G}_{\vec{y}=\vec{w}} \otimes \mathrm{G}^{\mathrm{Occ}(\vec{z})}_{\top} \models^{\mathrm{GI}}_{\dot{\mathcal{F}}^{\vec{z}}_{\varphi}} \mathtt{T}_{\psi}\vec{z}}\,.
$$

$\Leftarrow$: By induction on the structure of φ. We do case analysis on the structure of φ. The essential case is when $\varphi = [\psi]^{+}_{\vec{x}\vec{y}}\vec{u}\vec{w}$. Then, the derivation tree should be of the form like the above (by using Proposition 31(3)), where the rules for $\mathtt{T}_{\psi}$ are not applied in each ($\clubsuit$-i). Then by Proposition 31(4), each ($\clubsuit$-i) also shows $G_i \otimes \mathrm{G}^{\mathrm{Occ}(\vec{z})}_{\top} \models^{\mathrm{GI}}_{\dot{\mathcal{F}}^{\vec{z}}_{\psi}} \mathtt{S}_{\psi}\vec{z}$ from $\mathrm{G}^{\mathrm{Occ}(\vec{z})}_{\top} \models^{\mathrm{GI}}_{\dot{\mathcal{F}}^{\vec{z}}_{\psi}} \mathtt{T}_{\psi}\vec{z}$. By I.H., $G_i \models^{\mathrm{GI}} \psi$. Thus, $G \models^{\mathrm{GI}} \varphi$. ◀

▶ **Lemma 37.** *Every simple* $\mathsf{EP}(\mathsf{TC})$ *formula of the form* $\exists z_0.\varphi_0$ *or* $\top \vee \exists z_0.\varphi_0$ *has a GI-equivalent linear* FRS[PP]*.*

Proof. We only write the case of $\exists z_0.\varphi_0$ (the case of $\top\vee\exists z_0.\varphi_0$ is shown in the same way). Let us recall the linear FRS[PP] $\dot{\mathcal{F}}^{\vec{z}}_{\varphi_0} = \langle \mathcal{X}^{\vec{z}}_{\varphi_0}, \mathcal{R}^{\vec{z}}_{\varphi_0}, \mathfrak{s}^{\vec{z}}_{\varphi_0}\rangle$ in Figure 5, where $\vec{z}'z_0 \in \mathrm{Perm}(\mathsf{FV}_1(\varphi_0))$, $\vec{z}'' \in \mathrm{Perm}(\mathsf{BV}_1(\varphi_0))$, and $\vec{z} = \vec{z}'z_0\vec{z}''$. Let $\bar{\mathcal{F}}$ be the linear FRS[PP] $\langle \{\mathtt{S}\} \cup \mathcal{X}^{\vec{z}}_{\varphi_0}, \{\mathtt{S}\vec{z}' \leftarrow \exists z_0.\mathtt{S}_{\varphi_0}\vec{z}'z_0\dots z_0, \mathtt{T}_{\varphi_0}\vec{z} \leftarrow \vec{z} = \vec{z}\} \cup \mathcal{R}^{\vec{z}}_{\varphi_0}, \mathtt{S}\vec{z}'\rangle$. Then, $G[\mathtt{f}/z_0] \models^{\mathrm{GI}} \exists z_0.\varphi_0$ iff $G \models^{\mathrm{GI}} \varphi_0$ iff there exists a derivation tree that shows $G \otimes \mathrm{G}^{\vec{z}}_{\top} \models^{\mathrm{GI}}_{\dot{\mathcal{F}}^{\vec{z}}_{\varphi_0}} \mathtt{S}_{\varphi_0}\vec{z}$ from $\mathrm{G}^{\vec{z}}_{\top} \models^{\mathrm{GI}}_{\dot{\mathcal{F}}^{\vec{z}}_{\varphi_0}} \mathtt{T}_{\varphi_0}\vec{z}$ (Lemma 36) iff there exists a derivation tree that shows $G \models^{\mathrm{GI}}_{\dot{\mathcal{F}}^{\vec{z}}_{\varphi_0}} \mathtt{S}_{\varphi_0}\vec{z}'z_0\dots z_0$ from $\mathrm{G}^{\vec{z}}_{\top} \models^{\mathrm{GI}}_{\dot{\mathcal{F}}^{\vec{z}}_{\varphi_0}} \mathtt{T}_{\varphi_0}\vec{z}$ (because the name differences in the part $\vec{z}''$ do not affect to the construction of the derivation tree by $\vec{z}'' \in \mathrm{Perm}(\mathsf{BV}_1(\varphi_0))$) iff $G[\mathtt{f}/z_0] \models^{\mathrm{GI}}_{\bar{\mathcal{F}}} \mathtt{S}\vec{z}'$. Hence, $\mathcal{G}(\bar{\mathcal{F}}) = \mathcal{G}(\exists z_0.\varphi_0)$. ◀

Proof of Theorem 1(3)$\Rightarrow$. By Lemma 35 and 37 (with Proposition 29 and 30). ◀

From linear FRS[PP]s to EP(TC) formulas. This part is shown by generalizing *the state elimination method* in finite automata theory for linear FRS[PP]s. To this end, we introduce the following class based on transitions in finite automata. We say that an FRS[$\mathsf{EP}(\mathsf{TC})$] $\mathcal{F}$ is *FA-linear* if (a) there is a non-terminal label T (denoted by $\mathrm{T}^{\mathcal{F}}$) not equivalent to $\mathrm{S}^{\mathcal{F}}$ such that the label T has the single rule $\mathrm{T}\vec{x} \leftarrow \vec{x} = \vec{x}$; and (b) for every pair of $X \in \mathcal{X}^{\mathcal{F}} \setminus \{\mathrm{T}\}$ and $Y \in \mathcal{X}^{\mathcal{F}}$, there is exactly one rule of the form $X\vec{x} \leftarrow \exists\vec{y}.\psi * Y\vec{y}$ (we denote this ψ by $\varphi^{\mathcal{F}}_{X,Y}\vec{x}\vec{y}$; note that ψ does not have non-terminal labels), where the elements of $\vec{x}\vec{y}$ are pairwise distinct.

▶ **Lemma 38.** *Every linear* FRS[PP] *has a GI-equivalent FA-linear* FRS[EP]*.*

Proof. For the condition (a), we introduce a fresh non-terminal label T and introduce the rule $\mathrm{T}\vec{x} \leftarrow \vec{x} = \vec{x}$. For the condition (b), for each rule $X\vec{x} \leftarrow \varphi$, if φ does not have non-terminal labels, then we replace the rule with $X\vec{x} \leftarrow \exists \vec{z}.(\vec{z} = \vec{x} * \varphi) * \mathrm{T}\vec{z}$, where $\vec{z}$ is a sequence of fresh variables. Otherwise, let Y be the non-terminal label and transform the PP formula φ into a GI-equivalent formula of the form $\exists \vec{z}.\varphi' * Y\vec{u}$ by taking its prenex normal form and reordering the inner formulas appropriately. Then, transform it into the following formula: $\exists \vec{y}.(\exists \vec{z}.\vec{y} = \vec{u} * \varphi') * Y\vec{y}$, where $\vec{y}$ is a sequence of fresh variables. Next, for each pair $\langle X, Y \rangle$, let $\langle X\vec{x}_i \leftarrow \exists \vec{y}_i.\psi_i * Y\vec{y}_i \rangle_{i=1}^{n}$ be a permutation of all the rules of the form $X\vec{x} \leftarrow \exists \vec{y}.\psi * Y\vec{y}$. Without loss of generality, we can assume that $\vec{x}_1\vec{y}_1 = \dots = \vec{x}_n\vec{y}_n$ (so we denote it by $\vec{x}\vec{y}$) by renaming variables. Then, replace these rules with the single rule $X\vec{x} \leftarrow \exists \vec{y}.(\bigvee_{i=1}^{n} \psi_i) * Y\vec{y}$. ◀

Finally, we present a translation from FA-linear FRS[EP]s into EP(TC) formulas.

▶ **Lemma 39.** *Every FA-linear* FRS[EP(TC)] $\mathcal{F}$ *has a GI-equivalent* EP(TC) *formula.*

Proof. By induction on $\#(\mathcal{X}^{\mathcal{F}})$. If $\mathcal{X}^{\mathcal{F}} = \{\mathtt{S}^{\mathcal{F}}, \mathtt{T}^{\mathcal{F}}\}$, then $\mathcal{F}$ is denoted by $\langle \{\mathtt{S}^{\mathcal{F}}, \mathtt{T}^{\mathcal{F}}\}, \{\mathtt{S}^{\mathcal{F}}\vec{z} \leftarrow \exists \vec{x}.\varphi * \mathtt{T}^{\mathcal{F}}\vec{x}, \mathtt{T}^{\mathcal{F}}\vec{x} \leftarrow \vec{x} = \vec{x}\}, \mathfrak{s}^{\mathcal{F}} \rangle$. Thus, $\mathcal{F}$ is GI-equivalent to the EP(TC) formula $\exists \vec{x}.\varphi * \vec{x} = \vec{x}$ ($\cong^{\mathrm{GI}} \exists \vec{x}.\varphi$). Otherwise, there exists $Y_0 \in \mathcal{X}^{\mathcal{F}} \setminus \{\mathtt{S}^{\mathcal{F}}, \mathtt{T}^{\mathcal{F}}\}$. We define $\mathcal{F}' \triangleq \langle \mathcal{X}^{\mathcal{F}} \setminus \{Y_0\}, \{X\vec{x} \leftarrow \exists \vec{z}.(\varphi^{\mathcal{F}}_{X,Z}\vec{x}\vec{z} \vee \exists \vec{y}.\varphi^{\mathcal{F}}_{X,Y_0}\vec{x}\vec{y} * \exists \vec{y}'.[\varphi^{\mathcal{F}}_{Y_0,Y_0}\vec{y}\vec{y}']^{*}_{\vec{y}\vec{y}'}\vec{y}\vec{y}' * \varphi^{\mathcal{F}}_{Y_0,Z}\vec{y}'\vec{z}) * Z\vec{z} \mid X, Z \in \mathcal{X}^{\mathcal{F}} \setminus \{Y_0\}, X \neq \mathtt{T}^{\mathcal{F}}\} \cup \{\mathrm{T}^{\mathcal{F}}\vec{x} \leftarrow \vec{x} = \vec{x}\}, \mathfrak{s}^{\mathcal{F}} \rangle$, where elements of $\vec{x}\vec{z}\vec{y}\vec{y}'$ are pairwise distinct. Here, $[\varphi]^{*}_{\vec{x}\vec{y}}\vec{u}\vec{w}$ abbreviates the formula $\vec{u} = \vec{w} \vee [\varphi]^{+}_{\vec{x}\vec{y}}\vec{u}\vec{w}$. Then, the FA-linear FRS[EP(TC)] $\mathcal{F}'$ is GI-equivalent to $\mathcal{F}$ because there are transformations between derivation trees of $\mathcal{F}$ and those of $\mathcal{F}'$ in the same manner as the proof of Lemma 36. By I.H., $\mathcal{F}'$ has some GI-equivalent EP(TC) formula φ. Thus by using this φ, it has been proved. ◀

Proof of Theorem 1(3)⇐. By Lemma 38 and 39 (with Proposition 29 and 30). ◀

6 Conclusion

We have presented a perspective on graph languages via logical formulas by introducing *GI-semantics*. We have presented an axiomatization of the equational theory of PP/EP formulas under GI-semantics, and we have shown that several classes of existential positive logic formulas under GI-semantics have the same expressive power as those of HRGs. One future work is to find some axiomatization or some proof system of the (in)equational theory of EP(TC), or EP(LFP). Another possible future work is to study some classes of (bounded treewidth) graph languages by considering syntactic fragments, e.g., for finding decidable (or tractable) fragments of graph language problems. It would also be interesting to extend this logic to higher-order fixpoint logic (for a graph extension of higher-order grammars [17, 22]).

References

1 Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

2 Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998. doi:10.1017/cbo9781139172752.

3 Michel Bauderon and Bruno Courcelle. Graph expressions and graph rewritings. *Mathematical Systems Theory*, 20(1):83–127, 1987. doi:10.1007/BF01692060.

4 Filippo Bonchi, Jens Seeber, and Pawel Sobocinski. Graphical Conjunctive Queries. In *27th EACSL Annual Conference on Computer Science Logic (CSL '18)*, volume 119, pages 13:1–13:23. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018. doi:10.4230/LIPIcs.CSL.2018.13.

5 Francis Bossut, Max Dauchet, and Bruno Warin. A Kleene Theorem for a Class of Planar Acyclic Graphs. *Information and Computation*, 117(2):251–265, 1995. doi:10.1006/inco.1995.1043.

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

7 Janusz A. Brzozowski and Edward J. McCluskey. Signal Flow Graph Techniques for Sequential Circuit State Diagrams. *IEEE Transactions on Electronic Computers*, EC-12(2):67–76, 1963. doi:10.1109/PGEC.1963.263416.

8 J. Richard Büchi. Weak Second-Order Arithmetic and Finite Automata. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 6(1-6):66–92, 1960. doi:10.1002/malq.19600060105.

9 J. Richard Büchi. On a Decision Method in Restricted Second Order Arithmetic. In *International Congress on Logic, Methodology, and Philosophy of Science 1960*, pages 1–11. Stanford University Press, 1962.

10 Luca Cardelli, Philippa Gardner, and Giorgio Ghelli. A Spatial Logic for Querying Graphs. In *International Colloquium on Automata, Languages, and Programming (ICALP '02)*, pages 597–610. Springer Verlag, 2002. doi:10.1007/3-540-45465-9_51.

11 Ashok K. Chandra and Philip M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In *9th annual ACM symposium on Theory of computing (STOC '77)*, pages 77–90. ACM Press, 1977. doi:10.1145/800105.803397.

12 Chandra Chekuri and Anand Rajaraman. Conjunctive query containment revisited. *Theoretical Computer Science*, 239(2):211–229, 2000. doi:10.1016/S0304-3975(99)00220-0.

13 Hubert Comon, Max Dauchet, Rémi Gilleron, Florent Jacquemard, Denis Lugiez, Christof Löding, Sophie Tison, and Marc Tommasi. Tree Automata Techniques and Applications, 2007. URL: http://www.grappa.univ-lille3.fr/tata.

14 Enric Cosme Llópez and Damien Pous. K4-free Graphs as a Free Algebra. In *42nd International Symposium on Mathematical Foundations of Computer Science (MFCS '17)*, pages 76:1–76::14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPICS.MFCS.2017.76.

15 Bruno Courcelle and Joost Engelfriet. A Logical Characterization of the Sets of Hypergraphs Defined by Hyperedge Replacement Grammars. *Mathematical Systems Theory*, 28(6):515–552, 1995.

16 Bruno Courcelle and Joost Engelfriet. *Graph Structure and Monadic Second-Order Logic*. Cambridge University Press, 2012. doi:10.1017/CBO9780511977619.

17 Werner Damm. The IO- and OI-hierarchies. *Theoretical Computer Science*, 20(2):95–207, 1982. doi:10.1016/0304-3975(82)90009-3.

18 Christian Doczkal and Damien Pous. Treewidth-Two Graphs as a Free Algebra. In *43rd International Symposium on Mathematical Foundations of Computer Science (MFCS '18)*, volume 117, pages 60:1–60:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018. doi:10.4230/LIPIcs.MFCS.2018.60.

19 Frank Drewes, Hans-Jörg Kreowski, and Annegret Habel. Hyperedge Replacement Graph Grammars. In *Handbook of Graph Grammars and Computing by Graph Transformation, Volume 1: Foundations*, chapter 2, pages 95–162. World Scientific, 1997. doi:10.1142/9789812384720_0002.

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

21 Calvin C. Elgot. Decision Problems of Finite Automata Design and Related Arithmetics. *Transactions of the American Mathematical Society*, 98(1):21, 1961. doi:10.2307/1993511.

22 Joost Engelfriet. Iterated stack automata and complexity classes. *Information and Computation*, 95(1):21–75, 1991. doi:10.1016/0890-5401(91)90015-T.

23 Joost Engelfriet and Vincent van Oostrom. Logical Description of Context-free Graph Languages. *Journal of Computer and System Sciences*, 55(3):489–503, 1997. doi:10.1006/jcss.1997.1510.

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

25 Jerome Feder. Plex languages. *Information Sciences*, 3(3):225–241, 1971. doi:10.1016/S0020-0255(71)80008-7.

26 Tomás Feder and Moshe Y. Vardi. Monotone monadic SNP and constraint satisfaction. In *25th annual ACM symposium on Theory of computing (STOC '93)*, volume Part F1295, pages 612–622. ACM Press, 1993. doi:10.1145/167088.167245.

27 Stephen C. Kleene. Representation of Events in Nerve Nets and Finite Automata. In *Automata Studies. (AM-34)*, pages 3–42. Princeton University Press, 1956.

28 Phokion G. Kolaitis and Moshe Y. Vardi. Conjunctive-Query Containment and Constraint Satisfaction. *Journal of Computer and System Sciences*, 61(2):302–332, 2000. doi:10.1006/jcss.2000.1713.

29 Haas Leiß. Towards Kleene Algebra with recursion. In *5th International Workshop on Computer Science Logic (CSL '91)*, pages 242–256. Springer, 1991. doi:10.1007/BFb0023771.

30 Kamal Lodaya and Pascal Weil. Series-parallel languages and the bounded-width property. *Theoretical Computer Science*, 237(1-2):347–380, 2000. doi:10.1016/S0304-3975(00)00031-1.

31 Robert McNaughton. Testing and generating infinite sequences by a finite automaton. *Information and Control*, 9(5):521–530, 1966. doi:10.1016/S0019-9958(66)80013-X.

32 Yoshiki Nakamura. Expressive Power and Succinctness of the Positive Calculus of Relations. In *18th International Conference on Relational and Algebraic Methods in Computer Science (RAMiCS '20)*, volume 12062 of *LNCS*, pages 204–220. Springer, Cham, 2020. doi:10.1007/978-3-030-43520-2_13.

33 Yoshiki Nakamura. A full version of this paper, 2021. URL: https://yoshikinakamura.bitbucket.io/papers/ep_for_graph.

34 Mizuhito Ogawa, Zhenjiang Hu, and Isao Sasano. Iterative-free program analysis. In *8th ACM SIGPLAN international conference on Functional programming (ICFP '03)*, volume 8, pages 111–123. ACM Press, 2003. doi:10.1145/944705.944716.

35 Peter W. O'Hearn and David J. Pym. The Logic of Bunched Implications. *Bulletin of Symbolic Logic*, 5(2):215–244, 1999. doi:10.2307/421090.

36 Theodosios Pavlidis. Linear and Context-Free Graph Grammars. *Journal of the ACM*, 19(1):11–22, 1972. doi:10.1145/321679.321682.

37 Andrew M. Pitts. *Nominal Sets*. Cambridge University Press, 2013. doi:10.1017/CBO9781139084673.

38 David Pym. Resource semantics: logic as a modelling technology. *ACM SIGLOG News*, 6(2):5–41, 2019. doi:10.1145/3326938.3326940.

39 Neil Robertson and Paul D. Seymour. Graph minors. III. Planar tree-width. *Journal of Combinatorial Theory, Series B*, 36(1):49–64, 1984. doi:10.1016/0095-8956(84)90013-3.

40 Benjamin Rossman. Homomorphism preservation theorems. *Journal of the ACM*, 55(3):1–53, 2008. doi:10.1145/1379759.1379763.

41 Alfred Tarski. On the Calculus of Relations. *The Journal of Symbolic Logic*, 6(3):73–89, 1941. doi:10.2307/2268577.

42 Ken Thompson. Programming Techniques: Regular expression search algorithm. *Communications of the ACM*, 11(6):419–422, 1968. doi:10.1145/363347.363387.

43 Boris A. Trakhtenbrot. Finite automata and the logic of monadic predicates (in Russian). *Doklady Akademii Nauk SSSR*, 140:326–329, 1961.