Partial Derivatives on Graphs for Kleene Allegories

Yoshiki Nakamura
Tokyo Institute of Technology, Japan
Email: nakamura.y.ay@m.titech.ac.jp

Abstract—**Brunet and Pous showed at LICS 2015 that the equational theory of identity-free relational Kleene lattices (a fragment of *Kleene allegories*) is decidable in EXPSPACE. In this paper, we show that the equational theory of Kleene allegories is decidable, and is EXPSPACE-complete, answering the first open question posed by their work. The proof proceeds by designing *partial derivatives on graphs*, which are generalizations of partial derivatives on strings for regular expressions, called Antimirov's partial derivatives. The partial derivatives on graphs give a finite automata construction algorithm as with the partial derivatives on strings.**

I. Introduction

The terms of *Kleene allegories* [9] are built from variables, with the constants the empty 0 and the identity 1, the unary operations converse ($\bullet^{\smile}$) and reflexive-transitive closure ($\bullet^{\star}$), and the binary operations composition ($\cdot$), union ($\cup$), and intersection ($\cap$). Each term interprets a binary relation, whereas each term in regular expression interprets a set of strings.

Kleene allegories subsume *Kleene algebra* [11] and (representable distributive) *allegories* (or called *positive relation algebras*) [13]. Kleene algebra, that is an algebraic generalization of regular expressions, is a fragment of Kleene allegories, where one removes intersection and converse. Allegories are also fragments of Kleene allegories, where one remove reflexive-transitive closure. Their equational theories are decidable (see e.g., [3], [18] for Kleene algebra, and [13] for allegories). Especially, in [9], it is shown that *identity-free Kleene lattice* (a fragment of Kleene allegories that the terms are built from variables, with 0, $\cdot$, $\cup$, $\cap$, and transitive closure ($\bullet^{+}$)) is decidable in EXPSPACE by using Petri automata. However the decidability of Kleene allegories is open.

Our main contribution is to resolve the open question positively. The idea is based on *derivatives* on strings for regular expressions, e.g., Brzozowski's derivatives [10] and Antimirov's partial derivatives [3], that are tools to obtain the decidability of decision problems for regular expressions. In this work, we extend the derivatives from strings to graphs, and show the decidability of Kleene allegories by the following steps: (1) we extend some existing definitions with "labels"; this extension is effective for defining partial derivatives on graphs (Section II); (2) we design procedures for constructing finite graphs (called "Sequential Graph Construction Procedures") by using labels (Section III); (3) we give derivatives on graphs constructed by Sequential Graph Construction Procedures, and show that the equational theory is also decidable in EXPSPACE (Section V).

978-1-5090-3018-7/17/$31.00 ©2017 IEEE

A. Derivatives on Strings

The definition of derivatives on graphs depends on derivatives on strings [3], [10]. In fact, Section IV and V proceed in the same manner as this subsection. We first introduce derivatives on strings for comparing with derivatives on graphs.

Let A be an *alphabet* (a set of *characters*). A *string* over A is a finite sequence of characters, where ϵ is the *empty string*. A^* denotes the set of all strings over A, and let A^n be all strings of length n over A. A *language* over A is a subset of A^*. We use $a \in A$ to denote a character, $s \in A^*$ to denote a string, and $\mathcal{L} \subseteq A^*$ to denote a language. The set of *terms* is defined by the following grammar:

(Terms) $t ::= 0 \mid 1 \mid a \in A \mid t_1 \cdot t_2 \mid t_1 \cup t_2 \mid t_1 \cap t_2 \mid t_1^* \mid t_1^{\smile}$

We may omit the symbol $\cdot$, for short. We use T to denote a set of terms. The *concatenation* of T_1 and T_2, is defined by $T_1 \cdot T_2 = \{t_1 t_2 \mid t_1 \in T_1, t_2 \in T_2\}$. $|t|$ denotes the *length* of a term t, i.e., the number of symbols in t.

Definition I.1 (Languages)**.** The *language* of a term t, written $\mathcal{L}(t)$, is a set of strings defined inductively as follows:

$$\mathcal{L}(1) := \{\epsilon\};\ \mathcal{L}(0) := \emptyset;\ \mathcal{L}(a) := \{a\};$$
$$\mathcal{L}(t_1 \cdot t_2) := \{s_1 \cdot s_2 \mid s_1 \in \mathcal{L}(t_1), s_2 \in \mathcal{L}(t_2)\};$$
$$\mathcal{L}(t_1 \cup t_2) := \mathcal{L}(t_1) \cup \mathcal{L}(t_2);\ \mathcal{L}(t_1 \cap t_2) := \mathcal{L}(t_1) \cap \mathcal{L}(t_2);$$
$$\mathcal{L}(t_1^*) := \mathcal{L}(1) \cup \{s_1 \cdot \ldots \cdot s_n \mid n \geq 1, \forall i. s_i \in \mathcal{L}(t_1)\};$$
$$\mathcal{L}(t_1^{\smile}) := \{a_n \ldots a_1 \mid a_1 \ldots a_n \in \mathcal{L}(t_1)\}.$$

$\mathcal{L}(T)$ denotes the language $\bigcup_{t \in T} \mathcal{L}(t)$.

Remark. In the language model, it is inessential whether terms contain $\bullet^{\smile}$ or not because a term without $\bullet^{\smile}$ having the same language can be obtained from any term by the following rewriting rules:

$$1^{\smile} \to 1; 0^{\smile} \to 0; a^{\smile} \to a;\ (t_1 \cdot t_2)^{\smile} \to t_2^{\smile} \cdot t_1^{\smile};\ (t_1^*)^{\smile} \to (t_1^{\smile})^*;$$
$$(t_1 \cup t_2)^{\smile} \to t_1^{\smile} \cup t_2^{\smile};\ (t_1 \cap t_2)^{\smile} \to t_1^{\smile} \cap t_2^{\smile};\ t_1^{\smile\smile} \to t_1.$$

Derivatives on strings and partial derivatives on strings were developed by Brzozowski [10] and Antimirov [3], called Brzozowski's derivatives and Antimirov's partial derivatives, respectively. In this paper, we employ Antimirov's notation because the algorithm obtained from Antimirov's partial derivatives is easy to analyze computational complexity (Proposition I.8). For the sake of brevity, in this subsection, we only consider about terms without $\cap$ nor $\bullet^{\smile}$, i.e., terms of Kleene algebra. (See [4], for partial derivatives of terms containing $\cap$.) Partial derivatives on strings are closely related to the following operations.

Definition I.2 (Left quotients on strings)**.** The *left quotient* of a language $\mathcal{L}$ with respect to a string s, written $s^{-1}\mathcal{L}$, is a language, defined as $\{s' \mid ss' \in \mathcal{L}\}$.

Partial derivatives on strings consist of *empty string property* E and *partial derivatives on characters* D_a.

Definition I.3 (Empty string property (Reg1 in [3]))**.** The *empty string property* of a term t, written $E(t)$, is a truth value, defined inductively as follows:

$$E(0) := E(a) := false;\ E(1) := E(t^*) := true;$$
$$E(t_1 \cdot t_2) := E(t_1) \wedge E(t_2);\ E(t_1 \cup t_2) := E(t_1) \vee E(t_2).$$

$E(T)$ denotes the truth value $\bigvee_{t \in T} E(t)$.

Definition I.4 (Derivatives on characters [3])**.** The *derivative* of a term t with respect to a character $a \in A$, written $D_a(t)$, is a set of terms, defined inductively as follows:

$$D_a(a) := \{1\};\ D_a(a') := \emptyset \text{ for } a' \neq a;$$
$$D_a(0) := D_a(1) := \emptyset;$$
$$D_a(t_1 \cup t_2) := D_a(t_1) \cup D_a(t_2);\ D_a(t_1^*) := D_a(t_1) \cdot \{t_1^*\};$$
$$D_a(t_1 \cdot t_2) := D_a(t_1) \cdot \{t_2\} \cup D_a(t_2) \text{ if } E(t_1);$$
$$D_a(t_1 \cdot t_2) := D_a(t_1) \cdot \{t_2\} \text{ if } \neg E(t_1).$$

$D_a(T)$ denotes the set of terms $\bigcup_{t \in T} D_a(t)$.

Definition I.5 (Derivatives on strings [3])**.** The *derivative* of a term t with respect to a string $s \in A^*$, written $D_s(t)$, is a set of terms, defined inductively as follows:
$D_s(t) = \{t\}$ if $s = \epsilon$, and $D_s = D_a \circ D_{s'}$ if $s = s'a$.
$D_s(T)$ denotes the set of terms $\bigcup_{t \in T} D_s(t)$.

The next proposition shows left quotients can be characterized by the derivatives.

Proposition I.6 ([3, Proposition 2.10])**.**
(1) $E(t) \iff \epsilon \in \mathcal{L}(t)$.
(2) $\mathcal{L}(D_s(t)) = s^{-1}\mathcal{L}(t)$.
(3) $E(D_s(t)) \iff s \in \mathcal{L}(t)$.

The derivatives give some algorithms for language problems. For example, the *membership problem* (i.e., given a term t and a string s, the problem to decide whether $s \in \mathcal{L}(t)$ or not) is determined by checking whether $E(D_s(t))$ holds or not. Moreover, problems with searching a string, for example, the *(language) inclusion problem* (i.e., given two terms, t_1 and t_2, the problem to decide whether $\mathcal{L}(t_1) \subseteq \mathcal{L}(t_2)$ or not) and the *universality problem* (i.e., given a term t_1, the problem to decide whether $\mathcal{L}(t_1) = A^*$ or not) can be also determined by using the derivatives.

Proposition I.8 enables us to restrict the search space.

Definition I.7. The *closure* of a term t, written $\mathrm{cl}(t)$, is a set of terms, defined inductively as follows:

$$\mathrm{cl}(0) := \{0\};\ \mathrm{cl}(1) := \{1\};\ \mathrm{cl}(a) := \{1, a\};$$
$$\mathrm{cl}(t_1^*) := \mathrm{cl}(t_1) \cdot \{t_1^*\};\ \mathrm{cl}(t_1 \cup t_2) := \mathrm{cl}(t_1) \cup \mathrm{cl}(t_2);$$
$$\mathrm{cl}(t_1 \cdot t_2) := \mathrm{cl}(t_1) \cdot \{t_2\} \cup \mathrm{cl}(t_2).$$

$\mathrm{cl}(T)$ denotes the set of terms $\bigcup_{t \in T} \mathrm{cl}(t)$. cl is a *closure operator*, i.e., cl satisfies (a) $T \subseteq \mathrm{cl}(T)$; (b) $T_1 \subseteq T_2 \implies \mathrm{cl}(T_1) \subseteq \mathrm{cl}(T_2)$; and (c) $\mathrm{cl}(\mathrm{cl}(T)) = \mathrm{cl}(T)$.

Proposition I.8 ([3, Theorem 3.4])**.**
(1) $D_s(t) \subseteq \mathrm{cl}(t)$.
(2) $\#\,\mathrm{cl}(t) \leq 2 \times |t|$.

By Proposition I.8 and that cl is a closure operator, on derivatives of t, it is enough to consider terms in $\mathrm{cl}(t)$ because $D_s \restriction \mathrm{cl}(t) = D_{a_n} \restriction \mathrm{cl}(t) \circ \cdots \circ D_{a_1} \restriction \mathrm{cl}(t)$ holds by $D_s(t) \subseteq \mathrm{cl}(t)$ for any $s = a_1 \dots a_n$, where $f \restriction A$ denotes the *restriction* of a (partial) function f. Under the restriction, the number of patterns of functions from $\mathrm{cl}(t)$ to $\wp(\mathrm{cl}(t))$ is at most $2^{\#\,\mathrm{cl}(t)^2}$.
The finiteness justifies Algorithm 1.

Algorithm 1 The language inclusion problem

```
Ensure: L(t1) ⊆ L(t2)?
  CL ⇐ cl(t1) ∪ cl(t2)
  (D, d) ⇐ ({t' ↦ {t'} | t' ∈ CL}, 0)
  while d < 2^(#CL^2) do
    if E(D(t1)) ∧ ¬E(D(t2)) then
      return false
    end if
    pickup a ∈ A nondeterministically
    (D, d) ⇐ (D_a ∘ D, d + 1)
  end while
  return true
```

Theorem I.9 (e.g., [3])**.** *The language inclusion problem for terms of Kleene algebra is in PSPACE.*

Proof. By Algorithm 1 and Proposition I.8 (2), the problem is in coNPSPACE. Since coNPSPACE = PSPACE by Savitch's theorem [22], the problem is also in PSPACE. ☐

In fact, the problem is PSPACE-complete (see e.g., [17, Proposition 2.4]). By using derivatives on strings, any language of each term applied any left quotient is representable by finitely many 'sub'terms, whereas the language may be infinite.

B. Relation Models

The (in)equational theory of *Kleene allegories* [9] is defined by relation models.

Let V be a set of *nodes*. A *binary relation* (relation, for short) R on V is a subset of $V \times V$ (in other words, a set of *directed edges*). The *identity relation* on a set V' is defined by $\Delta(V') = \{(v, v) \mid v \in V'\}$. The *composition* of R_1 and R_2, is a relation, defined by $R_1 \cdot R_2 = \{(v, v'') \mid \exists v'.(v, v') \in R_1 \wedge (v', v'') \in R_2\}$, the *iteration* of R_1, written R_1^*, is the reflexive transitive closure of R_1, and the *converse* of R_1 is a relation, defined by $R_1^{\smile} = \{(v', v) \mid (v, v') \in R_1\}$.

Definition I.10 (Relation models)**.** A *relation model* M is a tuple $(V, \{R_a\}_{a \in A})$, where V is a nonempty set of nodes and each R_a is a relation on V. The *relation of a term t* with respect to M, written R_t^M, is defined inductively as follows:

$$R_1^M := \Delta(V^M);\ R_0^M := \emptyset;\ R_{t_1\cdot t_2}^M := R_{t_1}^M \cdot R_{t_2}^M;$$
$$R_{t_1\cup t_2}^M := R_{t_1}^M \cup R_{t_2}^M;\ R_{t_1\cap t_2}^M := R_{t_1}^M \cap R_{t_2}^M;$$
$$R_{t_1^*}^M := R_{t_1}^{M\,*};\ R_{t_1^{\smile}}^M := R_{t_1}^{M\,\smile}.$$

For an alphabet $A' \subseteq A$, $R_{A'}$ denotes $\bigcup_{a\in A'} R_a$. $M \models t_1 \le t_2$ denotes $R_{t_1}^M \subseteq R_{t_2}^M$, and $\mathrm{REL} \models t_1 \le t_2$ denotes that, for any relation model M, $M \models t_1 \le t_2$.

Definition I.11 (Homomorphism)**.** Let $M_i = (V^i, \{R_a^i\}_{a\in A})$ be a relation model for $i = 1, 2$. A function $h : V^2 \to V^1$ is called a *homomorphism* from M_2 to M_1 if, for any nodes, $v, v' \in V^2$ and any character a,

$$(v, v') \in R_a^2 \Longrightarrow (h(v), h(v')) \in R_a^1.$$

$M_1 \triangleleft M_2$ denotes there is a homomorphism from M_2 to M_1.

The next proposition shows a relationship between the equational theory on the language model and the equational theory over relation models.

Proposition I.12 (see e.g., [9], [19, Theorem 6])**.** *Let t_1 and t_2 be any two terms of Kleene algebra.*

$$\mathcal{L}(t_1) \subseteq \mathcal{L}(t_2) \iff \mathrm{REL} \models t_1 \le t_2.$$

However, when terms contain $\cap$ or $\bullet^{\smile}$, the proposition like Proposition I.12 does not hold. Let consider the next examples: (a) $a \cap aa = a \cap aaa$; (b) $a = a^{\smile}$; and (c) $a \le aa^{\smile}a$. Each (a) and (b) holds on the language model, however does not over relation models. In (a), the terms of both sides interpret the empty language on the language model, however it is easy to construct a relation model such that these terms are not equivalent [2, (1.8)]. In (b), the terms of both sides interpret the singleton language $\{a\}$, however the equation does not hold unless the relation of the character a on a relation model is symmetric [9]. In contrast, (c) holds over relation models, however does not on the language model [6, (10)].

Although the converse $\bullet^{\smile}$ over relation models cannot be completely erased like the converse $\bullet^{\smile}$ on the language model, any terms can be modified so that $\bullet^{\smile}$ is only applied to characters, by using the following rewriting rule [9]:

$$1^{\smile} \to 1; 0^{\smile} \to 0; (t_1 \cdot t_2)^{\smile} \to t_2^{\smile} \cdot t_1^{\smile}; (t_1^*)^{\smile} \to (t_1^{\smile})^*;$$
$$(t_1 \cup t_2)^{\smile} \to t_1^{\smile} \cup t_2^{\smile}; (t_1 \cap t_2)^{\smile} \to t_1^{\smile} \cap t_2^{\smile}; t_1^{\smile\smile} \to t_1.$$

For that reason, we assume that the set of terms is defined by the following grammar:

(Terms) $t ::= 0 \mid 1 \mid a \mid a^{\smile} \mid t_1 \cdot t_2 \mid t_1 \cup t_2 \mid t_1 \cap t_2 \mid t_1^*$

C. Graph Languages

Each term of Kleene allegories expresses a set of graphs with a single source and a single target. The inclusion problem for Kleene allegories can be solved by comparing their graph languages (Theorem I.17).

We define some mathematical notations needed later. $X_1 \uplus X_2$ denotes the *disjoint union* of two sets, X_1 and X_2, defined by $X_1 \uplus X_2 = \{(x_1, 1) \mid x_1 \in X_1\} \cup \{(x_2, 2) \mid x_2 \in X_2\}$. When i is clear from the context, (x_i, i) may be abbreviated as x_i. The *equivalence class* of an element x on X with a equivalence relation Θ, written $[x]_\Theta$, is defined by $[x]_\Theta = \{x' \in X \mid (x, x') \in \Theta\}$. When Θ is clear from the context, the subscript Θ may be abbreviated. Then the *quotient set* of X by Θ, written X/Θ, is defined as $\{[x] \mid x \in X\}$. $[X']$ denotes the set $\{[x] \mid x \in X'\}$. $R^{=}$ denotes the equivalence relation $(R \cup R^{\smile})^*$. For example, $\{1, 2, 3\}/\{(1, 2)\}^{=} = \{1, 2, 3\}/\{(1, 1), (2, 2), (3, 3), (1, 2), (2, 1)\} = \{\{1, 2\}, \{3\}\}$.

Let $\breve{a}$ be a character denoting the converse of a character a and $\mathtt{1}$ be the special constant character. $\breve{A}$ denotes the set $\{\breve{a} \mid a \in A\}$, $\breve{A}_1$ denotes $\breve{A} \cup \{\mathtt{1}\}$, $\dot{A}$ denotes $A \cup \breve{A}$, and $\dot{A}_1$ denotes $\dot{A} \cup \{\mathtt{1}\}$.

Definition I.13. A *graph* is a tuple $G = (V, \{R_{\dot a}\}_{\dot a \in \dot A_1}, v_s, v_t)$, where V is a nonempty set of nodes, each $R_{\dot a}$ is a relation on V, $v_s \in V$ is a node called the *source node*, and $v_t \in V$ is a node called the *target node*.

A graph G is said to be *normal formed* if $R_{\breve{A}_1} = \emptyset$. G may be denoted as (M, v_s, v_t) by using a relation model M. $M_1 \uplus M_2$ denotes the *disjoint union of relation models*, defined as $(V^{M_1} \uplus V^{M_2}, \bigcup_{i\in\{1,2\}}\{((v_i, i), (v_i', i)) \mid (v_i, v_i') \in R^{M_i}\})$ and M/Θ denotes the *quotient relation model*, defined as $(V^M/\Theta, \{([x], [x']) \mid (x, x') \in R^M\})$.

The *series-composition* of G_1 and G_2, written $G_1 \cdot G_2$, is a graph, defined as $((M^{G_1} \uplus M^{G_2})/\Theta, [v_s^{G_1}], [v_t^{G_2}])$, where $\Theta = \{(v_t^{G_1}, v_s^{G_2})\}^{=}$.

The *parallel-composition* of G_1 and G_2, written $G_1 \parallel G_2$, is a graph, defined as $((M^{G_1} \uplus M^{G_2})/\Theta, [v_s^{G_1}], [v_t^{G_1}])$, where $\Theta = \{(v_s^{G_1}, v_s^{G_2}), (v_t^{G_1}, v_t^{G_2})\}^{=}$.

A *graph language* $\mathcal{G}$ is a set of graphs. $\mathcal{G}_1 \bullet \mathcal{G}_2$ denotes the set of graphs $\{G_1 \bullet G_2 \mid G_1 \in \mathcal{G}_1, G_2 \in \mathcal{G}_2\}$ for $\bullet \in \{\cdot, \parallel\}$.

Definition I.14 (Graph languages)**.** The *graph language* of a term $\mathcal{G}(t)$ is defined inductively as follows:

$$\mathcal{G}(1) := \left\{ \rightarrow\!\circ \xrightarrow{\mathtt{1}} \circ\!\rightarrow \right\}; \mathcal{G}(\dot a) := \left\{ \rightarrow\!\circ \xrightarrow{\dot a} \circ \xrightarrow{\mathtt{1}} \circ\!\rightarrow \right\};$$
$$\mathcal{G}(0) := \emptyset;\ \mathcal{G}(t_1 \cdot t_2) := \mathcal{G}(t_1) \cdot \mathcal{G}(t_2);$$
$$\mathcal{G}(t_1^*) := \mathcal{G}(1) \cup \{G_1 \cdot \ldots \cdot G_n \mid n \ge 1, \forall i. G_i \in \mathcal{G}(t_1)\};$$
$$\mathcal{G}(t_1 \cup t_2) := \mathcal{G}(t_1) \cup \mathcal{G}(t_2);\ \mathcal{G}(t_1 \cap t_2) := \mathcal{G}(t_1) \parallel \mathcal{G}(t_2).$$

Any graph in the graph language of any term is a (directed) series-parallel graph [12].

Remark. $\mathcal{G}(\dot a)$ in the above definition may seem strange since the graph in the definition have an extra edge with the constant label $\mathtt{1}$. However, by the modification, some discussions are simplified because any labelled graph in a labelled graph language are "simple" (defined in Definition IV.1).

Definition I.15. The *normal form* of a graph G, written $\langle\!\langle G \rangle\!\rangle$, is defined as $((V^G, \{R_{\dot a}\}_{\dot a \in \dot A_1})/(R_{\mathtt{1}}^G)^{=}, [v_s^G], [v_t^G])$, where $R_a = R_a^G \cup R_{\breve a}^{G\,\smile}$ if $a \in A$, and $R_{\dot a} = \emptyset$ if $\dot a \in \breve{A} \cup \{\mathtt{1}\}$.

$\langle\!\langle \mathcal{G} \rangle\!\rangle$ denotes $\{\langle\!\langle G \rangle\!\rangle \mid G \in \mathcal{G}\}$. Note that $\langle\!\langle \mathcal{G}(t) \rangle\!\rangle$ is coincides with the graph language of t defined in [9].

Definition I.16 (Homomorphism [9])**.** Let G_i be a *normal formed* graph for $i = 1, 2$. A function $h \colon V^{G_2} \to V^{G_1}$ is called a *homomorphism* from G_2 to G_1 if the following are satisfied: (a) h is a homomorphism from M^{G_2} to M^{G_1}; (b) $h(v_s^{G_2}) = v_s^{G_1}$; (c) $h(v_t^{G_2}) = v_t^{G_1}$.

$G_1 \lhd G_2$ denotes that there exists a homomorphism from $\langle\!\langle G_2 \rangle\!\rangle$ to $\langle\!\langle G_1 \rangle\!\rangle$, where G_i is a graph for $i = 1, 2$. $\mathcal{G}_1 \lhd \mathcal{G}_2$ denotes that, for any graph $G_1 \in \mathcal{G}_1$, there exists a graph $G_2 \in \mathcal{G}_2$ such that $G_1 \lhd G_2$.

Theorem I.17 ([1, Theorem 1]).

$$\mathrm{REL} \models t_1 \le t_2 \iff \mathcal{G}(t_1) \lhd \mathcal{G}(t_2).$$

In the next section, we extend the above theorem with labels.

II. Extension with Labels

We extend relation models, terms, and graph languages with labels. In a nutshell, the extension is for expressing terms and graphs having multiple source nodes. This extension is effective to define Sequential Graph Construction Procedures and derivatives on graphs in later sections.

Let L be a set of *labels*, and $l \in L$ denotes a label. The set of *labelled terms* $\tilde{\mathcal{T}}$ is defined by the following grammar:

(Terms) $t ::= 0 \mid 1 \mid a \mid a^{\smile} \mid t_1 \cdot t_2 \mid t_1 \cup t_2 \mid t_1 \cap t_2 \mid t_1^*$

(Labeled Terms) $\tilde{t} ::= @l.t \mid \tilde{t}_1 \cdot t_2 \mid \tilde{t}_1 \cap \tilde{t}_2$

Intuitively $@l.t$ denotes t which the start point of t is forcibly regarded as the node with the label l. This is used for fixing the source nodes of terms and expressing terms which have multiple source nodes. This notation is derived from the jump operator in hybrid logics [5]. $A(\tilde{t})$ denotes the set of all characters occurring in $\tilde{t}$, and $L(\tilde{t})$ denotes the set of all labels occurring in $\tilde{t}$. We use $\tilde{T}$ to denote a set of labelled terms. $\tilde{\mathcal{T}}_L$ denotes the set $\{\tilde{t} \in \tilde{\mathcal{T}} \mid L(\tilde{t}) \subseteq L\}$, $\tilde{T}_1 \cdot T_2$ denotes the set $\{\tilde{t}_1 \cdot t_2 \mid \tilde{t}_1 \in \tilde{T}_1, t_2 \in T_2\}$, and $\tilde{T}_1 \Cap \tilde{T}_2$ denotes the set $\{\tilde{t}_1 \cap \tilde{t}_2 \mid \tilde{t}_1 \in \tilde{T}_1, \tilde{t}_2 \in \tilde{T}_2\}$.

Definition II.1 (Labeled relation models)**.** A *labelled relation model* $\tilde{M}$ is a tuple (M, m), where $M = (V, \{R_a\}_{a \in A})$ is a relation model and m is an injective and partial function from L to V. $L(\tilde{M})$ denotes the domain of m. The relation of a labelled term $\tilde{t}$ with respect to $\tilde{M}$, written $R_{\tilde{t}}^{\tilde{M}}$, is defined inductively as follows:

$$(v, v') \in R_{@l.t}^{\tilde{M}} :\Leftrightarrow m(l) \text{ is defined and } (m(l), v') \in R_t^M;$$
$$R_{\tilde{t}_1 \cdot t_2}^{\tilde{M}} := R_{\tilde{t}_1}^{\tilde{M}} \cdot R_{t_2}^{M};\ R_{\tilde{t}_1 \cap \tilde{t}_2}^{\tilde{M}} := R_{\tilde{t}_1}^{\tilde{M}} \cap R_{\tilde{t}_2}^{\tilde{M}}.$$

$\tilde{M} \models \tilde{t}_1 \le \tilde{t}_2$ denotes $R_{\tilde{t}_1}^{\tilde{M}} \subseteq R_{\tilde{t}_2}^{\tilde{M}}$, and $\mathrm{REL}^{\sim} \models \tilde{t}_1 \le \tilde{t}_2$ denotes that, for any labelled relation model $\tilde{M}$, $\tilde{M} \models \tilde{t}_1 \le \tilde{t}_2$.

It is easy to see the next proposition holds.

Proposition II.2. *For any label l_s,*

$$\mathrm{REL}^{\sim} \models @l_s.t_1 \le @l_s.t_2 \iff \mathrm{REL} \models t_1 \le t_2.$$

Sketch. It is shown by that the following hold for any M: $(M, m) \models @l_s.t_1 \le @l_s.t_2 \Leftrightarrow (M, m \restriction \{l_s\}) \models @l_s.t_1 \le @l_s.t_2$; $M \models t_1 \le t_2 \Leftrightarrow \forall v_s.(M, \{l_s \mapsto v_s\}) \models @l_s.t_1 \le @l_s.t_2$. □

Definition II.3 (Labeled graphs)**.** A *labelled graph* $\tilde{G}$ is a tuple $(V, \{R_{\dot{a}}\}_{\dot{a} \in \dot{A}_1}, m, v_t)$, where V is a nonempty set of nodes, each $R_{\dot{a}}$ is a relation on V; m is a function from L to $\wp(V)$ such that m is injective viewed as a binary relation over L and V (i.e., if $(l, v) \in m$ and $(l', v) \in m$, then $l = l'$); and $v_t \in V$ is a node called the *target node*. $\mathrm{dom}(m)$ and $\mathrm{cod}(m)$ denote the *domain* and the *codomain* of m, i.e., $\mathrm{dom}(m) = \{l \mid \exists v.(l, v) \in m\}$ and $\mathrm{cod}(m) = \{v \mid \exists l.(l, v) \in m\}$, respectively. $L(\tilde{G})$ denotes $\mathrm{dom}(m^{\tilde{G}})$. $\tilde{G}$ may be denoted as (M, m, v_t) by using a relation model M, or denoted as $(\tilde{M}, v_t)$ by using a labelled relation model $\tilde{M}$ when $m^{\tilde{G}}$ is functional (i.e., if $(l, v) \in m^{\tilde{G}}$ and $(l, v') \in m^{\tilde{G}}$, then $v = v'$).

The *series-composition* of $\tilde{G}_1$ and G_2, written $\tilde{G}_1 \cdot G_2$, is defined as $((M^{\tilde{G}_1} \uplus M^{G_2})/\{(v_t^{\tilde{G}_1}, v_s^{G_2})\}^{=}, m, [v_t^{G_2}])$, where m is defined by $m(l) = [m^{\tilde{G}_1}(l)]$.

The *parallel-composition* of $\tilde{G}_1$ and $\tilde{G}_2$, written $\tilde{G}_1 \parallel \tilde{G}_2$, is defined as $((M^{\tilde{G}_1} \uplus M^{\tilde{G}_2})/\{(v_t^{\tilde{G}_1}, v_t^{\tilde{G}_2})\}^{=}, m, [v_t^{\tilde{G}_1}])$, where m is defined by $m(l) = [m^{\tilde{G}_1}(l) \cup m^{\tilde{G}_2}(l)]$.

A *labelled graph language* $\tilde{\mathcal{G}}$ is a set of labelled graphs. $\tilde{\mathcal{G}}_1 \cdot \mathcal{G}_2$ denotes $\{\tilde{G}_1 \cdot G_2 \mid \tilde{G}_1 \in \tilde{\mathcal{G}}_1, G_2 \in \mathcal{G}_2\}$ and $\tilde{\mathcal{G}}_1 \parallel \tilde{\mathcal{G}}_2$ denotes $\{\tilde{G}_1 \parallel \tilde{G}_2 \mid \tilde{G}_1 \in \tilde{\mathcal{G}}_1, \tilde{G}_2 \in \tilde{\mathcal{G}}_2\}$. $\mathfrak{G}$ denotes the set of all labelled graphs and $\mathfrak{G}_{L'}$ denotes the set $\{\tilde{G} \in \mathfrak{G} \mid L(\tilde{G}) \subseteq L'\}$.

Definition II.4. The *labelled graph language of a labelled term* $\tilde{t}$, written $\tilde{\mathcal{G}}(\tilde{t})$, is defined inductively as follows:

$$\tilde{\mathcal{G}}(@l.t) := \{(g, \{(l, v_s)\}, v_t) \mid (g, v_s, v_t) \in \mathcal{G}(t)\};$$
$$\tilde{\mathcal{G}}(\tilde{t}_1 \cdot t_2) := \tilde{\mathcal{G}}(\tilde{t}_1) \cdot \mathcal{G}(t_2);\ \tilde{\mathcal{G}}(\tilde{t}_1 \cap \tilde{t}_2) := \tilde{\mathcal{G}}(\tilde{t}_1) \parallel \tilde{\mathcal{G}}(\tilde{t}_2).$$

$\tilde{G}$ is said to be *normal formed* if $R_{\breve{A}_1}^{\tilde{G}} = \emptyset$ and $m^{\tilde{G}}$ is functional.

Definition II.5. The *normal form* of a labelled graph $\tilde{G}$, written $\langle\!\langle \tilde{G} \rangle\!\rangle$, is defined as $((V^G, \{R_{\dot{a}}\}_{\dot{a} \in \dot{A}_1})/\Theta, m, [v_t^{\tilde{G}}])$, where (a) $\Theta = (\bigcup_{l \in L}(m^{\tilde{G}}(l) \times m^{\tilde{G}}(l)) \cup R_1^{\tilde{G}})^{=}$; (b) $R_a = R_a^{\tilde{G}} \cup R_{\breve{a}}^{\tilde{G}\smile}$ if $a \in A$, and $R_{\dot{a}} = \emptyset$ if $\dot{a} \in \breve{A}_1$; and (c) $m(l) = [m^{\tilde{G}}(l)]$.

$\langle\!\langle \tilde{G} \rangle\!\rangle$ may be undefined since m may not be injective in the above definition. $\langle\!\langle \tilde{\mathcal{G}} \rangle\!\rangle$ denotes the set $\{\langle\!\langle \tilde{G} \rangle\!\rangle \mid \tilde{G} \in \tilde{\mathcal{G}}\}$.

Definition II.6. Let $\tilde{G}_i$ be a normal formed labelled graph for $i = 1, 2$. A function $h\colon V^{\tilde{G}_2} \to V^{\tilde{G}_1}$ is called a *homomorphism* from $\tilde{G}_2$ to $\tilde{G}_1$ if the following are satisfied: (a) h is a homomorphism from $M^{\tilde{G}_2}$ to $M^{\tilde{G}_1}$; (b) $L(\tilde{G}_2) \subseteq L(\tilde{G}_1)$; (c) for any label $l \in L(\tilde{G}_2)$, $h(m^{\tilde{G}_2}(l)) = m^{\tilde{G}_1}(l)$; (d) $h(v_t^{\tilde{G}_2}) = v_t^{\tilde{G}_1}$.

$\tilde{G}_1 \lhd \tilde{G}_2$ denotes that, if $\langle\!\langle \tilde{G}_1 \rangle\!\rangle$ is defined, $\langle\!\langle \tilde{G}_2 \rangle\!\rangle$ is also defined and there exists a homomorphism from $\langle\!\langle \tilde{G}_2 \rangle\!\rangle$ to $\langle\!\langle \tilde{G}_1 \rangle\!\rangle$. $\tilde{\mathcal{G}}_1 \lhd \tilde{\mathcal{G}}_2$ denotes that, for any labelled graph $\tilde{G}_1 \in \tilde{\mathcal{G}}_1$, there exists a labelled graph $\tilde{G}_2 \in \tilde{\mathcal{G}}_2$ such that $\tilde{G}_1 \lhd \tilde{G}_2$.

Now we show a relationship between relation inclusions and homomorphisms of labelled graphs (Theorem II.9).

Proposition II.7 (cf. [1, Lemma 2])**.**

(1) $(M, v_s, v_t) \lhd G_1 \cdot G_2 \Leftrightarrow \exists v.(M, v_s, v) \lhd G_1 \wedge (M, v, v_t) \lhd G_2$.
(2) $(M, v_s, v_t) \lhd G_1 \parallel G_2 \Leftrightarrow (M, v_s, v_t) \lhd G_1 \wedge (M, v_s, v_t) \lhd G_2$.
(3) $(\tilde{M}, v_t) \lhd \tilde{G}_1 \cdot G_2 \Leftrightarrow \exists v.(\tilde{M}, v) \lhd \tilde{G}_1 \wedge (M^{\tilde{M}}, v, v_t) \lhd G_2$.
(4) $(\tilde{M}, v_t) \lhd \tilde{G}_1 \parallel \tilde{G}_2 \Leftrightarrow (\tilde{M}, v_t) \lhd \tilde{G}_1 \wedge (\tilde{M}, v_t) \lhd \tilde{G}_2$.

Sketch. (1) and (2) are proved in [1, Lemma 2]. (3) and (4) are proved in the same way as the proof of (1) and (2). ☐

Proposition II.8 (cf. [1, Lemma 3])**.**
(1) $(v_s, v_t) \in R_t^M \iff (M, v_s, v_t) \lhd \mathcal{G}(t)$.
(2) $v_t \in \mathrm{cod}(R_{\tilde{t}}^{\tilde{M}}) \iff (\tilde{M}, v_t) \lhd \tilde{\mathcal{G}}(\tilde{t})$.

Sketch. (1) is proved in [1, Lemma 3]. (2) is proved by induction on the length of the labelled term $\tilde{t}$ using (1), and (3) and (4) of Proposition II.7. ☐

Theorem II.9 (cf. Theorem I.17)**.**

$$\mathrm{REL}^{\sim} \models \tilde{t}_1 \leq \tilde{t}_2 \iff \tilde{\mathcal{G}}(\tilde{t}_1) \lhd \tilde{\mathcal{G}}(\tilde{t}_2).$$

Proof. ($\Rightarrow$): Let $\tilde{G}$ be any labelled graph in $\tilde{\mathcal{G}}(\tilde{t}_1)$ such that $\langle\!\langle \tilde{G} \rangle\!\rangle$ is defined. By $(\tilde{M}^{\langle\!\langle \tilde{G} \rangle\!\rangle}, v_t^{\langle\!\langle \tilde{G} \rangle\!\rangle}) \lhd \tilde{G}$ ($(\tilde{M}^{\langle\!\langle \tilde{G} \rangle\!\rangle}, v_t^{\langle\!\langle \tilde{G} \rangle\!\rangle})$ is equivalent to $\langle\!\langle \tilde{G} \rangle\!\rangle$) and Proposition II.8, $v_t^{\langle\!\langle \tilde{G} \rangle\!\rangle} \in \mathrm{cod}(R_{\tilde{t}_1}^{\tilde{M}^{\langle\!\langle \tilde{G} \rangle\!\rangle}})$. By $\mathrm{REL}^{\sim} \models \tilde{t}_1 \leq \tilde{t}_2$, $v_t^{\langle\!\langle \tilde{G} \rangle\!\rangle} \in \mathrm{cod}(R_{\tilde{t}_2}^{\tilde{M}^{\langle\!\langle \tilde{G} \rangle\!\rangle}})$. By Proposition II.8, $(\tilde{M}^{\langle\!\langle \tilde{G} \rangle\!\rangle}, v_t^{\langle\!\langle \tilde{G} \rangle\!\rangle}) \lhd \tilde{\mathcal{G}}(\tilde{t}_2)$. Therefore $\tilde{G} \lhd \tilde{\mathcal{G}}(\tilde{t}_2)$, and thus $\tilde{\mathcal{G}}(\tilde{t}_1) \lhd \tilde{\mathcal{G}}(\tilde{t}_2)$. ($\Leftarrow$): Let $\tilde{M}$ be any labelled relation model and let v_t be any node in $\mathrm{cod}(R_{\tilde{t}_1}^{\tilde{M}})$. By Proposition II.8, $(\tilde{M}, v_t) \lhd \tilde{\mathcal{G}}(\tilde{t}_1)$. By $\tilde{\mathcal{G}}(\tilde{t}_1) \lhd \tilde{\mathcal{G}}(\tilde{t}_2)$ and that $\lhd$ is transitive, $(\tilde{M}, v_t) \lhd \tilde{\mathcal{G}}(\tilde{t}_2)$. By Proposition II.8, $v_t \in \mathrm{cod}(R_{\tilde{t}_2}^{\tilde{M}})$. Therefore $\mathrm{REL}^{\sim} \models \tilde{t}_1 \leq \tilde{t}_2$. ☐

Definition II.10. The labeled graph of $\tilde{G}$ with a target label l_t, written $\tilde{G}^{l_t}$, is defined as $(V^{\tilde{G}}, \{R_{\dot{a}}^{\tilde{G}}\}_{\dot{a} \in \dot{A}_1}, m^{\tilde{G}} \cup \{(l_t, v_t)\}, v_t)$. $\tilde{\mathcal{G}}^{l_t}$ denotes $\{\tilde{G}^{l_t} \mid \tilde{G} \in \tilde{\mathcal{G}}\}$.

$\tilde{G}^{l_t}$ may be undefined since $m^{\tilde{G}} \cup \{(l_t, v_t)\}$ may not be functional.

Theorem II.11. $\tilde{\mathcal{G}}(\tilde{t}_1) \lhd \tilde{\mathcal{G}}(\tilde{t}_2) \iff \forall l_t \in L. \tilde{\mathcal{G}}(\tilde{t}_1)^{l_t} \lhd \tilde{\mathcal{G}}(\tilde{t}_2)^{l_t}$.

Proof. ($\Rightarrow$): Let $\tilde{G}_1$ be any graph such that $\tilde{G}_1^{l_t} \in \tilde{\mathcal{G}}(\tilde{t}_1)^{l_t}$ is defined, and let $\tilde{G}_2$ be a graph such that $\tilde{G}_1 \lhd \tilde{G}_2$. Then, $\tilde{G}_2^{l_t}$ is defined and thus $\tilde{G}_1^{l_t} \lhd \tilde{G}_2^{l_t}$ by using the same homomorphism. ($\Leftarrow$): Let $\tilde{G}_1$ be any labeled graph in $\tilde{\mathcal{G}}(\tilde{t}_1)$. When $v_t^{\tilde{G}_1}$ is labeled with a label l, there is a labelled graph $\tilde{G}_2 \in \tilde{\mathcal{G}}(\tilde{t}_2)$ such that $\tilde{G}_1 \lhd \tilde{G}_2^l$, by $\tilde{G}_1 \in \tilde{\mathcal{G}}(\tilde{t}_1)^l$ and $\tilde{\mathcal{G}}(\tilde{t}_1)^l \lhd \tilde{\mathcal{G}}(\tilde{t}_2)^l$. Then $\tilde{G}_1 \lhd \tilde{G}_2$ by using the same homomorphism. Otherwise, let l be a label not in $L(\tilde{t}_1)$ nor $L(\tilde{t}_2)$ and let $\tilde{G}_2^l$ be a labelled graph such that $\tilde{G}_1^l \lhd \tilde{G}_2^l$. Then $\tilde{G}_1 \lhd \tilde{G}_2$ by using the same homomorphism. ☐

III. Sequential Graph Construction Procedures

In this section, we define *Sequential Graph Construction Procedures*. In a nutshell, the purpose of the procedures is to express graphs by strings. A Sequential Graph Construction Procedure (SGCP) is a string of the following events: **(Adding)**, **(Connecting)**, or **(Forgetting)**. The intuition of each event is as follows:

(Adding: $\mathtt{A}(l_1)$**)** Add a node labelled with l_1.
(Connecting: $\mathtt{C}(l_1, l_2, a)$**)** Add an edge labelled with a from the node labelled with l_1 to the node labelled with l_2.
(Forgetting: $\mathtt{F}(l_1)$**)** Remove the label l_1.

We use e to denote an *event* ($\mathtt{A}(l_1)$, $\mathtt{C}(l_1, l_2, a)$, or $\mathtt{F}(l_1)$) and use ρ to denote a *string of events*.

Definition III.1. A *Sequential Graph Construction Procedure* (SGCP) ρ is defined as follows: (1) $\mathtt{A}(l_1)$ is a SGCP; (2) if ρ is a SGCP and $l_1 \notin \lambda(\rho)$, then $\rho\mathtt{A}(l_1)$ is a SGCP; (3) if ρ is a SGCP and $l_1, l_2 \in \lambda(\rho)$, then $\rho\mathtt{C}(l_1, l_2, a)$ is a SGCP; (4) if ρ is a SGCP and $l_1 \in \lambda(\rho)$, then $\rho\mathtt{F}(l_1)$ is a SGCP, where the *set of active labels* of a SGCP ρ, written $\lambda(\rho)$, is defined inductively by: (a) $\lambda(\mathtt{A}(l_1)) = \{l_1\}$; (b) $\lambda(\rho\mathtt{A}(l_1)) = \lambda(\rho) \cup \{l_1\}$; (c) $\lambda(\rho\mathtt{C}(l_1, l_2, a)) = \lambda(\rho)$; (d) $\lambda(\rho\mathtt{F}(l_1)) = \lambda(\rho) \setminus \{l_1\}$.

The *set of labels* of ρ, written $L(\rho)$, is defined as the set of all labels occurring in ρ. $\rho[l'/l]$ denotes ρ in which each occurrence of l has been replaced by l' and $\rho[\rho''/\rho']$ denotes ρ in which each occurrence of ρ' has been replaced by ρ''. Note that any SGCP ρ such that $\lambda(\rho) = \emptyset$ is in form of $\mathtt{A}(l)\rho'\mathtt{F}(l')$.

We explain the two main reasons of introducing labels.

1) Considering strings as relations, any string is corresponded to a path model. Any path graph can be constructed by repeating adding an edge at target node from the singleton graph. On the above procedure, given a path graph, it is enough to know which node is target to continue the procedure. Then we can assume that the other nodes are invisible. For that reason, we do not need to use labels for path models because the number of nodes that are needed to consider continuing the procedure is unique.
However, in general, given a procedure, the number of nodes that are needed to continue the procedure may not be unique, e.g., procedures constructing tree graphs. One of the reasons to introduce labels is to distinguishes multiple nodes in procedures.
2) Considering decidability or computational complexity, it is important to get an upperbound for the space complexity. SGCPs can make graphs with many nodes in low space complexity under some condition related to pathwidth [21] by forgetting unnecessary nodes using **Forgetting**. That is the second reason to introduce labels. In fact, any graph of a term of Kleene Allegories can be constructed in low space complexity (Theorem III.10).

Definition III.2. The *labelled graph of a SGCP* ρ, written $\tilde{G}(\rho) = (V^{\rho}, \{R_{\dot{a}}^{\rho}\}_{\dot{a} \in \dot{A}_1}, m^{\rho}, v_t^{\rho})$, is a normal formed labelled graph, defined inductively as follows:
(1) if $\rho = \mathtt{A}(l_1)$, $\tilde{G}(\rho) := (\{0\}, \{\emptyset\}_{\dot{a} \in \dot{A}_1}, \{(l_1, 0)\}, 0)$;
(2) if $\rho = \rho'\mathtt{A}(l_1)$, $\tilde{G}(\rho) :=$
$(V^{\rho'} \cup \{\#V^{\rho'}\}, \{R_{\dot{a}}^{\rho'}\}_{\dot{a} \in \dot{A}_1}, m^{\rho'} \cup \{(l_1, \#V^{\rho'})\}, 0)$;
(3) if $\rho = \rho'\mathtt{C}(l_1, l_2, a)$, $\tilde{G}(\rho) := (V^{\rho'}, \{R_{\dot{a}}\}_{\dot{a} \in \dot{A}_1}, m^{\rho'}, 0)$, where $R_{\dot{a}} = R_{\dot{a}}^{\rho'} \cup \{(m^{\rho'}(l_1), m^{\rho'}(l_2))\}$ if $\dot{a} = a$, and $R_{\dot{a}} = R_{\dot{a}}^{\rho'}$ otherwise;
(4) if $\rho = \rho'\mathtt{F}(l_1)$,
$\tilde{G}(\rho) := (V^{\rho'}, \{R_{\dot{a}}^{\rho'}\}_{\dot{a} \in \dot{A}_1}, m^{\rho'} \restriction (\mathrm{dom}(m^{\rho'}) \setminus \{l_1\}), 0)$.

Note that nodes in $\tilde{G}(\rho)$ are named by natural numbers $(0, 1, \dots)$ and 0 is always used for the target node. See Fig.1, for a graphical description of Definition III.2.

Definition III.3. The *graph of a SGCP* ρ such that $\rho = \rho'\mathtt{F}(l_s)$ and $\lambda(\rho) = \emptyset$, written $G(\rho)$, is a normal formed graph, defined as $(V^{\tilde{G}(\rho')}, \{R_{\dot{a}}^{\tilde{G}(\rho')}\}_{\dot{a} \in \dot{A}_1}, m^{\tilde{G}(\rho')}(l_s), v_t^{\tilde{G}(\rho')})$.

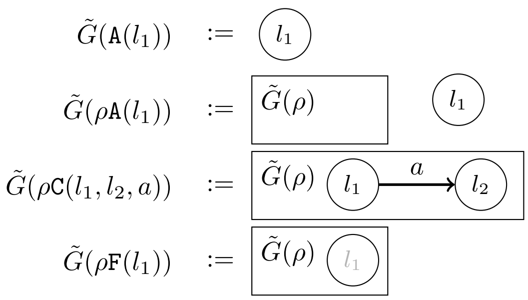


Fig. 1. Inductive construction of the labelled graph of a SGCP

The next proposition shows a completeness of SGCPs with respect to express finite relations.

Proposition III.4. *For any finite normal formed labelled graph $\tilde{G}$, there is a SGCP ρ such that $\tilde{G}(\rho)$ is isomorphic to $\tilde{G}$.*

Proof. Such SGCP can be constructed as follows: (a) make up nodes as many as the nodes in $\tilde{G}$ by repeating **Adding**; (b) make the same relation as $\tilde{G}$ by repeating **Connecting**; (c) remove labels not in $\tilde{G}$ by repeating **Forgetting**. □

Forgetting may seem inessential to construct graphs. However it is useful for saving the number of using labels.

Definition III.5 (n-SGCP)**.** The *labelwidth* of a SGCP $\rho = e_1 \dots e_n$, written $\tilde{\#}(\rho)$, is defined as $\max\{\#\lambda(e_1 \dots e_k) \mid 1 \leq k \leq n\}$. A SGCP ρ is called an *n-SGCP* if the labelwidth of ρ is less than or equal to n.

Example III.6. We consider a few examples about SGCPs.

- Let ρ_1 be $\mathtt{A}(l_n)\mathtt{A}(l_{n-1})\mathtt{C}(l_{n-1}, l_n, a_n)\mathtt{F}(l_n) \dots \mathtt{A}(l_0)$-$\mathtt{C}(l_0, l_1, a_1)\mathtt{F}(l_1)$. Then ρ_1 is a 2-SGCP and $\tilde{G}(\rho_1)$ is a path graph.
- Let ρ_2 be $\mathtt{A}(l')\mathtt{A}(l_1)\mathtt{A}(l_2)\mathtt{C}(l_1, l', a_1)\mathtt{C}(l_2, l', a_2)\mathtt{F}(l')\mathtt{A}(l)$-$\mathtt{C}(l, l_1, a_1)\mathtt{C}(l, l_2, a_2)\mathtt{F}(l_1)\mathtt{F}(l_2)\mathtt{F}(l)$.
Then ρ_2 is a 3-SGCP and $G(\rho_2)$ is a series-parallel graph.
- Let ρ_3 be $\mathtt{A}(l_1) \dots \mathtt{A}(l_5)\mathtt{C}(l_1, l_2, a) \dots \mathtt{C}(l_5, l_4, a)$.
Then ρ_3 is a 5-SGCP and $\tilde{G}(\rho_3)$ is the complete directed graph with 5 nodes.

See Fig.2 for graphs and labelled graphs expressed by the SGCPs in Example III.6.

A labelled relation model $\tilde{G}$ is called *n-expressible* if there exists an n-SGCP ρ such that $\tilde{G}(\rho)$ and $\tilde{G}$ are isomorphic.

Intuitively, the smaller n that a labelled graph is n-expressible, the more the graph is "elongated" like path graphs. Theorem III.10 shows that any labelled graph in the graph language of each term of Kleene allegories is "elongated".

We call that SGCPs $\rho_1, \dots, \rho_n$ (maybe $n = 1$) are *disjoint* if, each event $\mathtt{A}(l)$ and $\mathtt{F}(l)$ occur in the SGCPs at most once.

The *series-composition* of two disjoint SGCPs, ρ_1 and ρ_2, written $\rho_1 \diamond \rho_2$, is defined as follows:

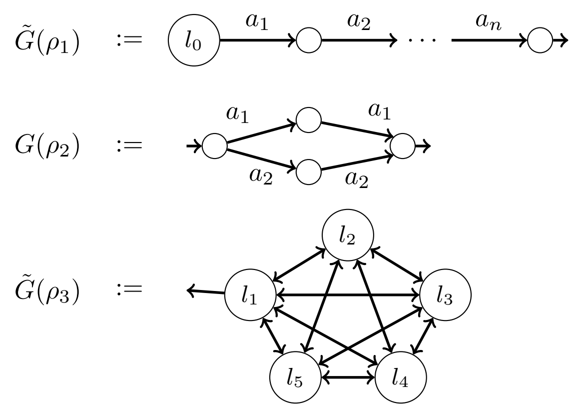


Fig. 2. Examples of (labelled) graphs constructed by a SGCP

- if $\lambda(\rho_2) = \emptyset$, $\rho_1 \diamond \rho_2 := (\rho_2[\epsilon/\mathtt{F}(l_2^s)][l_1^t/l_2^s])(\rho_1[\epsilon/\mathtt{A}(l_1^t)])$, where $\rho_1 = \mathtt{A}(l_1^t)\rho_1'$ and $\rho_2 = \mathtt{A}(l_2^t)\rho_2'\mathtt{F}(l_2^s)$;
- otherwise, $\rho_1 \diamond \rho_2$ is undefined.

The *parallel-composition* of two disjoint SGCPs, ρ_1 and ρ_2, written $\rho_1 \parallel \rho_2$, is defined as follows:

- if $\lambda(\rho_1) = \lambda(\rho_2) = \emptyset$, $\rho_1 \parallel \rho_2 :=$
$(\rho_1[\epsilon/\mathtt{F}(l_1^t)][\epsilon/\mathtt{F}(l_1^s)][l_2^t/l_1^t][l_2^s/l_1^s])(\rho_2[\epsilon/\mathtt{A}(l_2^t)][\epsilon/\mathtt{A}(l_2^s)])$, where $\rho_1 = \mathtt{A}(l_1^t)\rho_1'\mathtt{F}(l_1^s)$ and $\rho_2 = \mathtt{A}(l_2^t)\rho_2'\mathtt{F}(l_2^s)$;
- otherwise, $\rho_1 \parallel \rho_2 := (\rho_1[\epsilon/\mathtt{F}(l_1^t)][l_2^t/l_1^t])(\rho_2[\epsilon/\mathtt{A}(l_2^t)])$, where $\rho_1 = \mathtt{A}(l_1^t)\rho_1'$ and $\rho_2 = \mathtt{A}(l_2^t)\rho_2'$.

Proposition III.7. *Let ρ_1 and ρ_2 be disjoint SGCPs.*

(1) $G(\rho_1 \diamond \rho_2)$ is isomorphic to $G(\rho_1) \cdot G(\rho_2)$, where $\lambda(\rho_1) = \lambda(\rho_2) = \emptyset$.
(2) $\tilde{G}(\rho_1 \diamond \rho_2)$ is isomorphic to $\tilde{G}(\rho_1) \cdot G(\rho_2)$, where $\lambda(\rho_2) = \emptyset$.
(3) $G(\rho_1 \parallel \rho_2)$ is isomorphic to $G(\rho_1) \parallel G(\rho_2)$, where $\lambda(\rho_1) = \lambda(\rho_2) = \emptyset$.
(4) $\tilde{G}(\rho_1 \parallel \rho_2)$ is isomorphic to $\langle\!\langle \tilde{G}(\rho_1) \parallel \tilde{G}(\rho_2) \rangle\!\rangle$.

Proposition III.8. *Let ρ_1 and ρ_2 be disjoint SGCPs.*

(1) $\tilde{\#}(\rho_1 \diamond \rho_2) = \max(\tilde{\#}(\rho_1), \tilde{\#}(\rho_2))$ if $\rho_1 \diamond \rho_2$ is defined.
(2) $\tilde{\#}(\rho_1 \parallel \rho_2) = \tilde{\#}(\rho_1) + \tilde{\#}(\rho_2)$.

Definition III.9 (cf. [15])**.** The *intersection width* of a term (or a labelled term), written $\mathrm{iw}(t)$ (or $\mathrm{iw}(\tilde{t})$), is defined inductively as follows:

$$\mathrm{iw}(0) := \mathrm{iw}(1) := \mathrm{iw}(a) := \mathrm{iw}(a^{\smile}) := 1;$$
$$\mathrm{iw}(t_1 \cup t_2) := \mathrm{iw}(t_1 \cdot t_2) := \max(\mathrm{iw}(t_1), \mathrm{iw}(t_2));$$
$$\mathrm{iw}(\tilde{t}_1 \cdot t_2) := \max(\mathrm{iw}(\tilde{t}_1), \mathrm{iw}(t_2));$$
$$\mathrm{iw}(t_1^*) := \mathrm{iw}(@l.t_1) := \mathrm{iw}(t_1);$$
$$\mathrm{iw}(t_1 \cap t_2) := \mathrm{iw}(t_1) + \mathrm{iw}(t_2);\ \ \mathrm{iw}(\tilde{t}_1 \cap \tilde{t}_2) := \mathrm{iw}(\tilde{t}_1) + \mathrm{iw}(\tilde{t}_2).$$

It is easy to see that $\mathrm{iw}(t) \leq |t|$ and $\mathrm{iw}(\tilde{t}) \leq |\tilde{t}|$ hold.

Theorem III.10.

(1) Any graph $G \in \langle\!\langle \mathcal{G}(t) \rangle\!\rangle$ is $(\mathrm{iw}(t) + 1)$-expressible.
(2) Any labelled graph $\tilde{G} \in \langle\!\langle \tilde{\mathcal{G}}(\tilde{t}) \rangle\!\rangle$ is $(\mathrm{iw}(\tilde{t}) + 1)$-expressible.

Proof. (1) is proved by induction on the length of t using Proposition III.7 and III.8. (2) is proved by induction on the length of $\tilde{t}$ using (1) and Proposition III.7 and III.8. $\square$

Corollary III.11. *Let l_t be any label. Any labelled graph $\tilde{G} \in \langle\!\langle \tilde{\mathcal{G}}(\tilde{t})^{l_t} \rangle\!\rangle$ is $(\mathrm{iw}(\tilde{t})+2)$-expressible.*

Proof. Let $\rho = A(l)\rho'$ be a disjoint $(\mathrm{iw}(\tilde{t})+1)$-SGCP such that ρ expresses a labelled graph $\tilde{G} \in \langle\!\langle \tilde{\mathcal{G}}(\tilde{t}) \rangle\!\rangle$. If $\tilde{G}^{l_t}$ is defined, $\rho[\epsilon/\mathtt{F}(l_t)]$ is a $(\mathrm{iw}(\tilde{t})+2)$-SGCP such that expresses $\tilde{G}^{l_t}$. $\square$

The labelwidth of a SGCP is closely related with space complexity. Corollary III.11 is useful for analysing computational complexity in Section V.

Remark. Note that the following are equivalent:

(1) $\tilde{G}$ is expressible by a n-SGCP ρ;

(2) $\tilde{G}$ is expressible by a SGCP ρ such that $\#L(\rho) \leq n$.

(2) $\Rightarrow$ (1) is obvious. (1) $\Rightarrow$ (2) is shown by relabelling each label in ρ adequately. Thus it is enough to prepare at most n labels to express a n-expressible graph $\tilde{G}$.

A. Labelwidth and pathwidth

We mention that the labelwidth of a graph can be characterized by the *pathwidth* in graph theory [21]. (It has some alternative characterizations, see e.g., [7, Theorem 2] and [8, Lemma 4.6].) The pathwidth of a graph is determined by an optimal *path-decomposition*.

Definition III.12 (cf. [21])**.** Let G be a normal formed graph. A sequence $X_1, \dots, X_n$ of subsets of V^G is a *path-decomposition* if the following condition are satisfied:

(a) for any $(v, v') \in R_{\dot{A}}^G$, there is i such that $\{v, v'\} \subseteq X_i$;

(b) for any $1 \leq i \leq i' \leq i'' \leq n$, $X_i \cap X_{i''} \subseteq X_{i'}$;

(c) $v_t^G \in X_1$; $v_s^G \in X_n$.

(Note that (c) is added from the definition in [21] for the source node and the target node.) Let the *width* of a path-decomposition $X_1, \dots, X_n$ be the maximum of $\#X_1 - 1, \dots, \#X_n - 1$. Then the *pathwidth* of a graph G is defined as the minimum of the widths of path-decompositions of G.

Example III.13. Fig.3 shows examples of path-decompositions for graphs of Fig.2. These path-decompositions are optimal, i.e., the pathwidths are just 1, 2, and 4 shown by these path-decompositions, respectively.

Proposition III.14. *Let $G = (V, \{R_{\dot{a}}\}_{\dot{a} \in \dot{A}_1}, v_s, v_t)$ be any normal formed graph. The following are equivalent:*

(1) G is expressible by a n-SGCP;

(2) the pathwidth of G is at most $n-1$.

Proof. (1) $\Rightarrow$ (2): Let $\rho = e_1 \dots e_k$ be a n-SGCP such that ρ expresses G, let $\rho_i = e_1 \dots e_i$ be a n-SGCP, let $\tilde{G}(\rho_i) = (V^i, \{R_{\dot{a}}^i\}_{\dot{a} \in \dot{A}_1}, m^i, 0)$ be a normal formed labelled graph, and let $X_i = \mathrm{cod}(m^i)$. Then $X_1, \dots, X_{k-1}$ is a path-decomposition of width $n-1$ of G. (Note that the condition (b) in Definition III.12 is justified by that, once a label is forgotten, the node associated with the label will be never labelled again.) Thus the pathwidth of G is at most $n-1$.

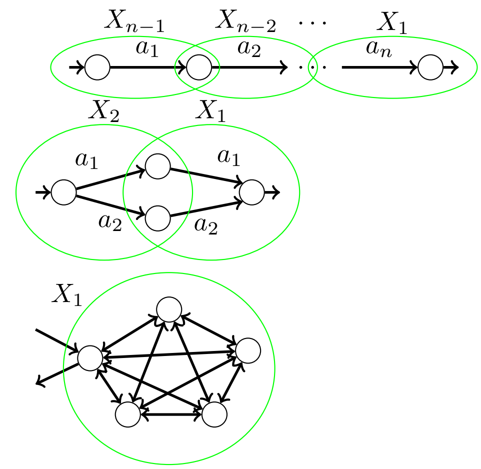


Fig. 3. Examples of path-decomposition

(2) $\Rightarrow$ (1): Let $X_1, \dots, X_k$ be a path-decomposition of width at most $n-1$ of G. For convenience, let X_0 be $\emptyset$. We construct a SGCP according to the path-decomposition. The SGCP by repeating the next procedure from $i = 0$ to $i = k-1$ is a n-SGCP that expresses G:

(i) Forgetting any label associated with nodes in $X_i \setminus X_{i+1}$;

(ii) Adding new nodes as many as the size of $X_{i+1} \setminus X_i$;

(iii) For any pair $(v, v') \in R_a \cap ((X_{i+1} \times X_{i+1}) \setminus (X_i \times X_i))$, connecting the pair with an edge labelled with a,

where v_t is added at first when $i = 0$, and v_s is forgotten at last when $i = k-1$. $\square$

IV. Left Quotients on Graphs

In this section, we only consider graphs in the following class. The restriction is not critical because any labelled graph in the labelled graph language of each labelled term of Kleene allegories is simple.

Definition IV.1. A labelled graph $\tilde{G} = (V, \{R_{\dot{a}}\}_{\dot{a} \in \dot{A}_1}, m, v_t)$ is called *simple* if the following are satisfied:

(a) $\tilde{G}$ is acyclic: $R_{\dot{A}_1}^{+} \cap \Delta(V) = \emptyset$;

(b) any node is reachable to v_t: $V = \mathrm{dom}(R_{\dot{A}_1}^{*} \cdot \Delta(\{v_t\}))$;

(c) $\sum_{\dot{a} \in \dot{A}} \#\{v \mid (v, v') \in R_{\dot{a}}\} \leq 1$, for any node v';

(d) any left most node is labelled with some label, and vice versa: $V \setminus \mathrm{cod}(R_{\dot{A}_1}) = \mathrm{cod}(m)$.

Similarly, a graph $G = (V, \{R_{\dot{a}}\}_{\dot{a} \in \dot{A}_1}, v_s, v_t)$ is called *simple* if (a),(b),(c) and (d') $V \setminus \mathrm{cod}(R_{\dot{A}_1}) = \{v_s\}$ are satisfied.

It is easy to see that the following are hold, where $X_a^{l_1, l_2}$, ρ^{-1}, $\tilde{G} \downarrow v$, and $\tilde{G} \uparrow m$ are defined in this section.

Proposition IV.2. *If $\tilde{G}$ is simple, the following graphs are all simple: (1) any labelled graph in $X_a^{l_1, l_2}(\tilde{G})$; (2) any labelled graph in $\rho^{-1}(\tilde{G})$; (3) $\tilde{G} \downarrow v$; (4) $\tilde{G} \uparrow m$.*

Now we define *left quotients on graphs* (generated by SGCPs). In the next section, derivatives on graphs will be defined based on the left quotients in a natural way.

Definition IV.3 (l-empty)**.** A labelled graph $\tilde{G}$ is said to be *l-empty* if $\tilde{M}^{\langle\!\langle \tilde{G} \rangle\!\rangle}$ is isomorphic to the labelled relation $(\{0\}, \{\emptyset\}_{\dot{a} \in \dot{A}_1}, \{(0, l)\})$.

Definition IV.4. The *lower labelled graph* of a labelled graph $\tilde{G}$ in $\mathfrak{G}_L$ with respect to a node $v \in V^{\tilde{G}}$, written $\tilde{G}\!\downarrow\! v$, is a labelled graph, defined as $(V', \{R_{\dot{a}}^{\tilde{G}} \cap (V' \times V')\}_{\dot{a} \in \dot{A}_1}, m^{\tilde{G}} \cap (L \times V'), v)$, where $V' = \mathrm{dom}((R_{\dot{A}_1}^{\tilde{M}})^* \cdot \Delta(\{v\}))$.

Definition IV.5 (Left quotients on edges)**.** The *left quotient* of a labelled graph $\tilde{G}$ with respect to a character a and two labels, l_1 and l_2, written $X_a^{l_1,l_2}(\tilde{G})$, is the smallest set of labelled graphs that the following are satisfied:

(1) Let v_1 and v_2 be two nodes such that $\tilde{G}\!\downarrow\! v_1$ is l_1-empty and $(v_1, v_2) \in R_a^{\tilde{G}}$. Then $(V^{\tilde{G}}, \{R'_{\dot{a}}\}_{\dot{a} \in \dot{A}_1}, m^{\tilde{G}} \cup \{(l_2, v_2)\}, v_t^{\tilde{G}})\!\downarrow\! v_t^{\tilde{G}} \in X_a^{l_1,l_2}(\tilde{G})$, where $R'_{\dot{a}} = R_{\dot{a}}^{\tilde{G}} \setminus \{(v_1, v_2)\}$ if $\dot{a} = a$, and $R'_{\dot{a}} = R_{\dot{a}}^{\tilde{G}}$ otherwise.

(2) Let v_1 and v_2 be two nodes such that $\tilde{G}\!\downarrow\! v_2$ is l_2-empty and $(v_2, v_1) \in R_{a^\smile}^{\tilde{G}}$. Then $(V^{\tilde{G}}, \{R'_{\dot{a}}\}_{\dot{a} \in \dot{A}_1}, m^{\tilde{G}} \cup \{(l_1, v_1)\}, v_t^{\tilde{G}})\!\downarrow\! v_t^{\tilde{G}} \in X_a^{l_1,l_2}(\tilde{G})$, where $R'_{\dot{a}} = R_{\dot{a}}^{\tilde{G}} \setminus \{(v_2, v_1)\}$ if $\dot{a} = \breve{a}$, and $R'_{\dot{a}} = R_{\dot{a}}^{\tilde{G}}$ otherwise.

$X_a^{l_1,l_2}(\tilde{\mathcal{G}})$ denotes $\bigcup_{\tilde{G} \in \tilde{\mathcal{G}}} X_a^{l_1,l_2}(\tilde{G})$.

Intuitively, $X_a^{l_1,l_2}$ means moving once from left most nodes labelled with l_1 (or l_2) to a node connected with an edge labelled with a (or $\breve{a}$), and then erasing the passed edge. Fig.4 is an example of left quotients on edges. The green colored edges are the edges which the left quotient $X_a^{l_1,l_2}$ can be triggered. The graph on the left side generates the two graphs on the right side by $X_a^{l_1,l_2}$.

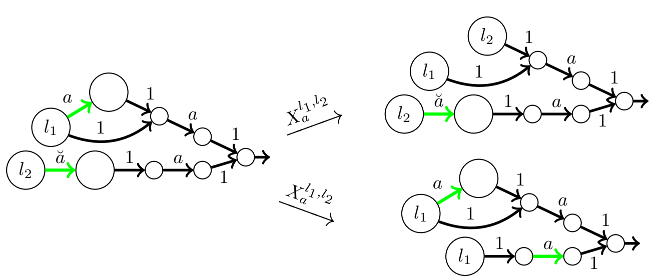


Fig. 4. An example of left quotients on edges

Now we extend the left quotients from edges to graphs. Let $l^\perp$ be the *dummy label*, that is used only to define left quotients. $\tilde{\mathfrak{G}}_L^\perp$ denotes the set of labelled graphs $\tilde{\mathfrak{G}}_{L \cup \{l^\perp\}}$. $\tilde{G}^\perp$ denotes the labelled graph $(V^{\tilde{G}} \cup \{\perp\}, \{R_{\dot{a}}^{\tilde{G}}\}_{\dot{a} \in \dot{A}_1}, m_s^{\tilde{G}} \cup \{(l^\perp, \perp)\}, v_t^{\tilde{G}})$, where $\perp \notin V^{\tilde{G}}$. $m[l'/l]$ denotes m in which each pair $(l, v) \in m$ has been replaced by (l', v), and $\tilde{G}[l'/l]$ denotes $(V^{\tilde{G}}, \{R_{\dot{a}}^{\tilde{G}}\}_{\dot{a} \in \dot{A}_1}, m^{\tilde{G}}[l'/l], v_t^{\tilde{G}})$.

Definition IV.6 (Left quotients on graphs)**.** The *left quotient* of $\tilde{G} \in \tilde{\mathfrak{G}}^\perp$ with respect to a SGCP ρ, written $\rho^{-1}(\tilde{G})$ is a set of labelled graphs, defined inductively as follows:

if $\rho = \mathtt{A}(l_1)$, $\rho^{-1} := \Delta(\tilde{\mathfrak{G}}_{\{l_1\}}^\perp)$;
if $\rho = \rho'\mathtt{A}(l_1)$, $\tilde{G}' \in \rho^{-1}(\tilde{G}) :\Leftrightarrow$
$\tilde{G}'[l^\perp/l_1] \in \rho'^{-1}(\tilde{G}[l^\perp/l_1]) \wedge m^{\tilde{G}'}(l_1) = m^{\tilde{G}}(l_1)$;
if $\rho = \rho'\mathtt{C}(l_1, l_2, a)$, $\rho^{-1} := (\rho'^{-1} \circ X_a^{l_1,l_2})^* \circ \rho'^{-1}$;
if $\rho = \rho'\mathtt{F}(l_1)$, $\rho^{-1} := \rho'^{-1} \cap (\tilde{\mathfrak{G}}_{\lambda(\rho)}^\perp \times \tilde{\mathfrak{G}}_{\lambda(\rho)}^\perp)$.

$\rho^{-1}(\tilde{\mathcal{G}})$ denotes $\bigcup_{\tilde{G} \in \tilde{\mathcal{G}}} \rho^{-1}(\tilde{G})$. ρ^{-1} is a closure operator.

Intuitively, ρ^{-1} means moving zero or more times between labelled nodes, and then erasing the passed edges, where each moving can be simulated by the relation model of ρ.

Example IV.7. Let $\rho_1 := \mathtt{A}(l_1)\mathtt{A}(l_2)\mathtt{C}(l_1, l_2, a)$. Then ρ_1^{-1} of the labelled graph on the left side of Fig.4 is the set of all labelled graphs obtained by applying $X_a^{l_1,l_2}$ zero or more times to the labelled graph.

Let $\rho_2 := \mathtt{A}(l_1)\mathtt{A}(l_2)\mathtt{A}(l_3)\mathtt{C}(l_3, l_2, a)\mathtt{F}(l_3)$. Then ρ_2^{-1} of the labelled graph on the left side of Fig.4 contains the labelled graphs by applying $X_a^{l_3,l_2}$ zero or two times, but does not contain the labelled graph by applying $X_a^{l_3,l_2}$ one time because this graph has the forgotten label l_3.

Now we show a relationship between these left quotients and homomorphisms (Theorem IV.11).

Lemma IV.8. *Let ρ be any SGCP and $\tilde{G}$ be any simple labelled graph in $\mathfrak{G}_{\lambda(\rho)}^\perp$.*

(1) if $(m^{\tilde{G}(\rho)^\perp}(l_1), m^{\tilde{G}(\rho)^\perp}(l_2)) \in R_a^{\tilde{G}(\rho)^\perp}$, $\tilde{G}(\rho)^\perp \lhd X_a^{l_1,l_2}(\tilde{G}) \Longrightarrow \tilde{G}(\rho)^\perp \lhd \tilde{G}$.
(2) $\tilde{G}(\rho)^\perp \lhd \rho^{-1}(\tilde{G}) \Longrightarrow \tilde{G}(\rho)^\perp \lhd \tilde{G}$.

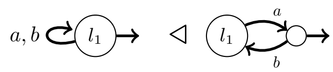
Proof. (1): Let $\tilde{G}' \in X_a^{l_1,l_2}(\tilde{G})$ be a labelled graph such that $\tilde{G}(\rho)^\perp \lhd \tilde{G}'$. By the definition of $X_a^{l_1,l_2}$ and that any node is reachable to $v_t^{\tilde{G}}$ (Definition IV.1 (b)), the subgraph with the nodes eliminated by $X_a^{l_1,l_2}$ from $\tilde{G}$ is l_1-empty (or l_2-empty). Then a homomorphism from $\langle\!\langle \tilde{G}' \rangle\!\rangle$ to $\langle\!\langle \tilde{G}(\rho)^\perp \rangle\!\rangle$ can be extend to a homomorphism from $\langle\!\langle \tilde{G} \rangle\!\rangle$ to $\langle\!\langle \tilde{G}(\rho)^\perp \rangle\!\rangle$ by mapping each eliminated node to the node labelled with l_1 (or l_2).

(2): It is proved by induction on the length of ρ.

- if $\rho = \mathtt{A}(l_1)$, it is obvious since $\rho^{-1}(\tilde{G}) = \{\tilde{G}\}$.
- if $\rho = \rho'\mathtt{A}(l_1)$, let $\tilde{G}' \in \rho^{-1}(\tilde{G})$ be a labelled graph such that $\tilde{G}(\rho)^\perp \lhd \tilde{G}'$. Then $\tilde{G}(\rho')^\perp \lhd \tilde{G}'[l^\perp/l_1]$ since both the node labelled with $l^\perp$ and the node labelled with l_1 in $\langle\!\langle \tilde{G}(\rho)^\perp \rangle\!\rangle$ are isolated from any other node. From this and $\tilde{G}'[l^\perp/l_1] \in \rho'^{-1}(\tilde{G}[l^\perp/l_1])$, $\tilde{G}(\rho')^\perp \lhd \rho'^{-1}(\tilde{G}[l^\perp/l_1])$. By I.H. and $\tilde{G}(\rho)^\perp \lhd \tilde{G}(\rho')^\perp$, $\tilde{G}(\rho)^\perp \lhd \tilde{G}[l^\perp/l_1]$.
  Let h be a homomorphism from $\langle\!\langle \tilde{G}[l^\perp/l_1] \rangle\!\rangle$ to $\langle\!\langle \tilde{G}(\rho)^\perp \rangle\!\rangle$. Then a homomorphism from $\langle\!\langle \tilde{G} \rangle\!\rangle$ to $\langle\!\langle \tilde{G}(\rho)^\perp \rangle\!\rangle$ can be obtained from h by remapping each node labelled with l_1 in $\langle\!\langle \tilde{G} \rangle\!\rangle$ to the node labelled with l_1 in $\langle\!\langle \tilde{G}(\rho)^\perp \rangle\!\rangle$. Therefore, $\tilde{G}(\rho)^\perp \lhd \tilde{G}$.
- if $\rho = \rho'\mathtt{C}(l_1, l_2, a)$, there exists n such that $\tilde{G}(\rho)^\perp \lhd ((\rho'^{-1} \circ X_a^{l_1,l_2})^n \circ \rho'^{-1})(\tilde{G})$. By I.H. and $\tilde{G}(\rho)^\perp \lhd \tilde{G}(\rho')^\perp$, $\tilde{G}(\rho)^\perp \lhd (X_a^{l_1,l_2} \circ \rho'^{-1})^n(\tilde{G})$. By Lemma IV.8 (1), $\tilde{G}(\rho)^\perp \lhd ((\rho'^{-1} \circ X_a^{l_1,l_2})^{n-1} \circ \rho'^{-1})(\tilde{G})$. By repeating the above inference, $\tilde{G}(\rho)^\perp \lhd \tilde{G}$ is proved.
- if $\rho = \rho'\mathtt{F}(l_1)$, by $\rho^{-1}(\tilde{G}) \subseteq \rho'^{-1}(\tilde{G})$, $\tilde{G}(\rho')^\perp \lhd \tilde{G}(\rho)^\perp$, and I.H., $\tilde{G}(\rho')^\perp \lhd \tilde{G}$. Then by using the same homomorphism, $\tilde{G}(\rho)^\perp \lhd \tilde{G}$ since $l_1 \notin L(\tilde{G})$.

□

The above lemma shows that, if a graph can be erased by a left quotient ρ^{-1}, there exists a homomorphism from the graph to $\langle\!\langle \tilde{G}(\rho) \rangle\!\rangle$. However the converse does not hold in general. For example, the below right labelled graph can not be erased by $X_a^{l_1,l_1}$, whereas there exists a homomorphism.



For that reason, the restriction to simple graphs is meaningful.

Definition IV.9. The *upper labelled graph* of a labelled graph $\tilde{G}$ with respect to a function m from L to $\wp(V^{\tilde{G}})$, written $\tilde{G}\uparrow m$, is a labelled graph, defined as $(V', \{R_{\dot{a}}^{\tilde{G}} \cap (V' \times V')\}_{\dot{a} \in \dot{A}_1}, (m^{\tilde{G}} \cup m) \cap (L \times V'), v_t^{\tilde{G}})$, where $V' = \mathrm{cod}((R_{\dot{A}_1}^{\tilde{M}} \cap (V \times (V \setminus \mathrm{cod}(m))))^* \cdot \Delta(v_t^{\tilde{G}}))$.

The *complement* of $\tilde{G}\uparrow m$, written $(\tilde{G}\uparrow m)^{\mathsf{c}}$, denotes the labelled graph $(V', \{R_{\dot{a}}^{\tilde{G}} \setminus R_{\dot{a}}^{\tilde{G}\uparrow m}\}_{\dot{a} \in \dot{A}_1}, (m^{\tilde{G}} \cup m) \cap (L \times V'), v_\bullet)$, where $V' = (V^{\tilde{G}} \setminus V^{\tilde{G}\uparrow m}) \cup \mathrm{cod}(m)$ and $v_\bullet$ is some node in V' ($v_\bullet$ is used only for defining the labelled graph).

Lemma IV.10. *Let ρ be any SGCP, let $\tilde{G}$ be any simple labelled graph in $\mathfrak{G}_{\lambda(\rho)}^{\perp}$, and let m be any function such that ($\star$): $m(l) \subseteq \mathrm{cod}(R_{\dot{A}}^{\tilde{G}})$.*

$$\tilde{M}^{\tilde{G}(\rho)^\perp} \lhd \tilde{M}^{\langle\!\langle (\tilde{G}\uparrow m)^{\mathsf{c}} \rangle\!\rangle} \Longrightarrow \tilde{G}\uparrow m \in \rho^{-1}(\tilde{G}).$$

Proof. It is proved by induction on the pair (the length of ρ, the number of edges in $\tilde{G}\uparrow m$).

- if $\rho = \mathtt{A}(l_1)$, $R_{\dot{A}}^{(\tilde{G}\uparrow m)^{\mathsf{c}}} = \emptyset$ by $R_{\dot{A}}^{\tilde{G}(\rho)^\perp} = \emptyset$ and the hypothesis. By ($\star$) and Definition IV.9, $R_{\dot{A}}^{\tilde{G}\uparrow m} = R_{\dot{A}}^{\tilde{G}}$, and thus $m = \emptyset$. Since $\tilde{G}\uparrow m = \tilde{G}$, $\tilde{G}\uparrow m \in \rho^{-1}(\tilde{G})$.
- if $\rho = \rho'\mathtt{A}(l_1)$, since both the node labelled with l_1 and the node labelled with $l^\perp$ are isolated from any other node in $\tilde{G}(\rho)^\perp$, $\tilde{M}^{\tilde{G}(\rho)^\perp} \lhd \tilde{M}^{\langle\!\langle (\tilde{G}[l^\perp/l_1]\uparrow m[l^\perp/l_1])^{\mathsf{c}} \rangle\!\rangle}$ by the hypothesis, and $m^{\tilde{G}\uparrow m}(l_1) = m^{\tilde{G}}(l_1)$ by ($\star$) and Definition IV.9. By I.H., $\tilde{G}[l^\perp/l_1]\uparrow m[l^\perp/l_1] \in {\rho'}^{-1}(\tilde{G}[l^\perp/l_1])$, and thus $(\tilde{G}\uparrow m)[l^\perp/l_1] \in {\rho'}^{-1}(\tilde{G}[l^\perp/l_1])$. Since $m^{\tilde{G}\uparrow m}(l_1) = m^{\tilde{G}}(l_1)$, $\tilde{G}\uparrow m \in \rho^{-1}(\tilde{G})$.
- if $\rho = \rho'\mathtt{F}(l_1)$, by $\tilde{M}^{\tilde{G}(\rho')^\perp} \lhd \tilde{M}^{\tilde{G}(\rho)^\perp}$ and I.H., $\tilde{G}\uparrow m \in {\rho'}^{-1}(\tilde{G})$. Since $l_1 \notin L(\tilde{M}^{\tilde{G}(\rho)^\perp})$ and $\tilde{M}^{\tilde{G}(\rho)^\perp} \lhd \tilde{M}^{\langle\!\langle (\tilde{G}\uparrow m)^{\mathsf{c}} \rangle\!\rangle}$, $l_1 \notin L((\tilde{G}\uparrow m)^{\mathsf{c}})$. From this and $l_1 \notin L(\tilde{G})$, $l_1 \notin L(\tilde{G}\uparrow m)$. Therefore $\tilde{G}\uparrow m \in \rho^{-1}(\tilde{G})$.
- if $\rho = \rho'\mathtt{C}(l_1, l_2, a)$, Let m' be the function such that,

  ($\star$1) m' satisfies ($\star$);
  ($\star$2) $\tilde{M}^{\tilde{G}(\rho')^\perp} \lhd \tilde{M}^{\langle\!\langle (\tilde{G}\uparrow m')^{\mathsf{c}} \rangle\!\rangle}$;
  ($\star$3) $V^{\tilde{G}\uparrow m'} \supseteq V^{\tilde{G}\uparrow m}$ is minimized.

  For short, we write $\tilde{G}\uparrow m'$ as $\tilde{G}'$. From ($\star$2) and I.H., $\tilde{G}' \in {\rho'}^{-1}(\tilde{G})$. When $m' = m$, it is immediately proved. Otherwise, by $m' \neq m$, ($\star$3), and ($\star$1), there exists two nodes, v_1 and v_2, such that $(v_1, v_2) \in R_{\dot{A}}^{\tilde{G}'}$ and $v_2 \in V^{(\tilde{G}\uparrow m)^{\mathsf{c}}}$. From this, the hypothesis, ($\star$2), and that $\tilde{G}'$ is acyclic, there exist two nodes $v_1, v_2 \in V^{\tilde{G}'}$ such that (i) $(v_1, v_2) \in R_a^{\tilde{G}'}$; (ii) $v_2 \in V^{(\tilde{G}\uparrow m)^{\mathsf{c}}}$; and (iii) $\tilde{G}'\downarrow v_1$ is l_1-empty, (or that (i) $(v_2, v_1) \in R_{\breve{a}}^{\tilde{G}'}$; (ii) $v_1 \in V^{(\tilde{G}\uparrow m)^{\mathsf{c}}}$; and (iii) $\tilde{G}'\downarrow v_2$ is l_2-empty.). Then there exists a labelled graph $\tilde{G}'' \in X_a^{l_1,l_2}(\tilde{G}')$ such that $\tilde{G}(\rho)^\perp \lhd \tilde{M}^{\langle\!\langle (\tilde{G}''\uparrow m)^{\mathsf{c}} \rangle\!\rangle}$. Note that $\tilde{G}''$ is simple and the number of edges in $\tilde{G}''\uparrow m$ is strictly smaller than the number in $\tilde{G}\uparrow m$. By I.H., $\tilde{G}''\uparrow m \in \rho^{-1}(\tilde{G}'')$. Since (ii), $\tilde{G}''\uparrow m = \tilde{G}'\uparrow m$. Since ($\star$3), $\tilde{G}'\uparrow m = \tilde{G}\uparrow m$. Also $\rho^{-1}(\tilde{G}'') \subseteq \rho^{-1}(X_a^{l_1,l_2}({\rho'}^{-1}(\tilde{G}))) \subseteq \rho^{-1}(\tilde{G})$. Therefore $\tilde{G}\uparrow m \in \rho^{-1}(\tilde{G})$.

□

Theorem IV.11. *Let ρ be any SGCP, let $\tilde{G}$ be any simple labelled graph in $\mathfrak{G}_{\lambda(\rho)}$, and let l_t be any label such that $\tilde{G}^{l_t}$ is defined. The following are equivalent:*

(1) there exists an l_t-empty labelled graph in $\rho^{-1}(\tilde{G})$;
(2) $\tilde{G}(\rho) \lhd \tilde{G}^{l_t}$.

Proof. (1) ⇒ (2): There also exists an l_t-empty labelled graph in $\rho^{-1}(\tilde{G}^{l_t})$ because it does not matter whether the target node is labelled or not in the operation ρ^{-1}. By Lemma IV.8 (2), $\tilde{G}(\rho)^\perp \lhd \tilde{G}^{l_t}$. Therefore $\tilde{G}(\rho) \lhd \tilde{G}^{l_t}$ since $l^\perp \notin L(\tilde{G}^{l_t})$.
(2) ⇒ (1): Let $m_t = \{l_t \mapsto \mathrm{cod}(R_{\dot{A}}^{\tilde{G}}) \cap ((R_1^*)^{\tilde{G}} \cdot \Delta(\{v_t^{\tilde{G}}\}))\}$. Then, since $\langle\!\langle (\tilde{G}\uparrow m_t)^{\mathsf{c}} \rangle\!\rangle = \langle\!\langle \tilde{G}^{l_t} \rangle\!\rangle$ and $l^\perp \notin L(\tilde{G})$, $M^{\tilde{G}(\rho)} \lhd M^{\langle\!\langle (\tilde{G}\uparrow m_t)^{\mathsf{c}} \rangle\!\rangle}$. By Lemma IV.10, $\tilde{G}\uparrow m_t \in \rho^{-1}(\tilde{G})$. Since $\tilde{G}\uparrow m_t$ is l_t-empty, (1) is proved. □

V. Derivatives on Graphs

In this section, we define derivatives on graphs (generated by SGCPs). The derivatives on graphs characterize the left quotient on graphs just like the derivative on strings characterize the left quotient on strings. The derivatives on graphs consist of *l-empty graph property E_l* and *partial derivatives on edges $D_a^{l_1,l_2}$*.

Definition V.1 (l-empty graph property)**.** Let l be a label. The *l-empty graph property* of a labelled term, written $E_l(\tilde{t})$, is a truth value, defined inductively as follows:

$$\begin{aligned}
E_l(@l.\dot{a}) &:= E_l(@l.0) := \mathit{false};\\
E_l(@l'.t) &:= \mathit{false} \text{ for } l' \neq l;\\
E_l(@l.1) &:= E_l(@l.t_1^*) := \mathit{true};\\
E_l(@l.t_1 \cup t_2) &:= E_l(@l.t_1) \vee E_l(@l.t_2);\\
E_l(@l.t_1 \cdot t_2) &:= E_l(@l.t_1 \cap t_2) := E_l(@l.t_1) \wedge E_l(@l.t_2);\\
E_l(\tilde{t}_1 \cdot t_2) &:= E_l(\tilde{t}_1) \wedge E_l(@l.t_2);\\
E_l(\tilde{t}_1 \cap \tilde{t}_2) &:= E_l(\tilde{t}_1) \wedge E_l(\tilde{t}_2).
\end{aligned}$$

$E_l(\tilde{T})$ denotes the truth value $\bigvee_{\tilde{t} \in \tilde{T}} E_l(\tilde{t})$.

The next proposition shows that E_{l_t} characterizes l_t-empty. It is shown by simple induction on the length of $\tilde{t}$.

Proposition V.2. *The following are equivalent:*

(1) $E_{l_t}(\tilde{t})$ is true;
(2) there exists a l_t-empty labelled graph in $\tilde{G}(\tilde{t})$.

Definition V.3 (Derivatives on edges)**.** The *derivatives on edges* of a labelled term $\tilde{t}$ with respect to a character a and two labels, l_1 and l_2, written $D_a^{l_1,l_2}(\tilde{t})$, is a set of labelled terms, defined inductively as follows:

$$D_a^{l_1,l_2}(@l.0) := D_a^{l_1,l_2}(@l.1) := \emptyset;$$
$$D_a^{l_1,l_2}(@l_1.a) := \{@l_2.1\};\ D_a^{l_1,l_2}(@l_2.\breve{a}) := \{@l_1.1\};$$
$$D_a^{l_1,l_2}(@l.a') := \emptyset \text{ for } a' \neq a \text{ or } l \neq l_1;$$
$$D_a^{l_1,l_2}(@l.\breve{a}') := \emptyset \text{ for } a' \neq a \text{ or } l \neq l_2;$$
$$D_a^{l_1,l_2}(@l.t_1 \cup t_2) := D_a^{l_1,l_2}(@l.t_1) \cup D_a^{l_1,l_2}(@l.t_2);$$
$$D_a^{l_1,l_2}(@l.t_1 \cdot t_2) := D_a^{l_1,l_2}((@l.t_1) \cdot t_2);$$
$$D_a^{l_1,l_2}(@l.t_1^*) := D_a^{l_1,l_2}(@l.t_1) \cdot \{t_1^*\};$$
$$D_a^{l_1,l_2}(@l.t_1 \cap t_2) := D_a^{l_1,l_2}((@l.t_1) \cap (@l.t_2));$$
$$D_a^{l_1,l_2}(\tilde{t}_1 \cap \tilde{t}_2) := (D_a^{l_1,l_2}(\tilde{t}_1) \Cap \{\tilde{t}_2\}) \cup (\{\tilde{t}_1\} \Cap D_a^{l_1,l_2}(\tilde{t}_2));$$

$D_a^{l_1,l_2}(\tilde{t}_1 \cdot t_2)$ is the smallest set such that
(1) $D_a^{l_1,l_2}(\tilde{t}_1) \cdot \{t_2\} \subseteq D_a^{l_1,l_2}(\tilde{t}_1 \cdot t_2)$,
(2) if $E_{l_i}(\tilde{t}_1)$, $D_a^{l_1,l_2}(@l_i.t_2) \subseteq D_a^{l_1,l_2}(\tilde{t}_1 \cdot t_2)$, for $i = 1, 2$.
$D_a^{l_1,l_2}(\tilde{T})$ denotes the set of labelled terms $\bigcup_{\tilde{t}\in\tilde{T}} D_a^{l_1,l_2}(\tilde{t})$.

The above derivatives can characterize the left quotient on edges. The next proposition is shown by case analysis on where an erased edge by $X_a^{l_1,l_2}$ is derived from.

Proposition V.4. *Let $\tilde{G}_1$ and $\tilde{G}_2$ be any simple labelled graphs and let G_2 be any simple graph.*
(1) $X_a^{l_1,l_2}(\tilde{G}_1 \parallel \tilde{G}_2) = (X_a^{l_1,l_2}(\tilde{G}_1) \parallel \tilde{G}_2) \cup (\tilde{G}_1 \parallel X_a^{l_1,l_2}(\tilde{G}_2))$.
(2) $X_a^{l_1,l_2}(\tilde{G}_1 \cdot G_2)$ is the smallest set such that
  a) $X_a^{l_1,l_2}(\tilde{G}_1) \cdot \{G_2\} \subseteq X_a^{l_1,l_2}(\tilde{G}_1 \cdot G_2)$;
  b) if $E_{l_i}(\tilde{G}_1)$, $X_a^{l_1,l_2}(G_2^{l_i}) \subseteq X_a^{l_1,l_2}(\tilde{G}_1 \cdot G_2)$, for $i = 1, 2$, where $\tilde{G}_2^{l_i} = (V^{G_2}, \{R_{\dot{a}}^{G_2}\}_{\dot{a}\in\dot{A}_1}, \{(l_i, v_s^{G_2})\}, v_t^{G_2})$.

Theorem V.5. $\tilde{\mathcal{G}}(D_a^{l_1,l_2}(\tilde{t})) = X_a^{l_1,l_2}(\tilde{\mathcal{G}}(\tilde{t}))$.

Proof. It is proved by induction on the length of $\tilde{t}$ using Proposition V.4. □

Next we extend the derivatives from edges to graphs.

Definition V.6 (Derivatives on graphs)**.** Let e be an event and let D be a partial function from $\tilde{\mathcal{T}}$ to $\wp(\tilde{\mathcal{T}})$. Then $e(D)$ is a partial function from $\tilde{\mathcal{T}}$ to $\wp(\tilde{\mathcal{T}})$, defined as follows:

- if $e = \mathtt{A}(l_1)$ and $l_1 \notin L(D)$,
  - $e(D)(\tilde{t}_1)$ is undefined if $\tilde{t}_1 \notin \tilde{\mathcal{T}}_{L(D)\cup\{l_1\}}$,
  - $e(D)(\tilde{t}_1) := D(\tilde{t}_1)$ if $\tilde{t}_1 \in \tilde{\mathcal{T}}_{L(D)}$,
  - $e(D)(\tilde{t}_1 \cap \tilde{t}_2) := e(D)(\tilde{t}_1) \Cap e(D)(\tilde{t}_2)$,
  - $e(D)(@l_1.t) = \{@l_1.t\}$,
  - $e(D)(\tilde{t}_1 \cdot t_2) := e(D)(\tilde{t}_1) \cdot \{t_2\}$;
- if $e = \mathtt{C}(l_1, l_2, a)$ and $l_1, l_2 \in L(D)$,
  $e(D) := (D \circ D_a^{l_1,l_2})^* \circ D$;
- if $e = \mathtt{F}(l_1)$ and $l_1 \in L(D)$,
  $e(D) := D \cap (\tilde{\mathcal{T}}_{L(D)\setminus\{l_1\}} \times \tilde{\mathcal{T}}_{L(D)\setminus\{l_1\}})$;
- otherwise, $e(D)$ is undefined.

($\bullet^*$ in the definition of the case $\rho = \rho'\mathtt{C}(l_1, l_2, a)$ is the reflexive transitive closure viewed as a relation.)

The *derivative* with respect to a SGCP ρ of a labelled term $\tilde{t}$, written $D_\rho(\tilde{t})$, is defined inductively as follows:

- if $\rho = \mathtt{A}(l_1)$, $D_\rho := \{\tilde{t} \mapsto \{\tilde{t}\} \mid \tilde{t} \in \tilde{\mathcal{T}}_{\{l_1\}}\}$;
- if $\rho = \rho' e$, $D_\rho := e(D_{\rho'})$.

$D_\rho(\tilde{T})$ denotes the set of terms $\bigcup_{\tilde{t}\in\tilde{T}} D_\rho(\tilde{t})$.

Example V.7. We list a few examples of derivatives on graphs.

1) $D_{\mathtt{A}(l_0)\mathtt{A}(l_1)\mathtt{C}(l_0,l_1,a)}(@l_0.a) = \{@l_0.a, @l_1.1\}$.
2) $D_{\mathtt{A}(l_0)\mathtt{A}(l_1)\mathtt{C}(l_0,l_1,a)\mathtt{C}(l_1,l_0,b)}(@l_0.ab \cap 1)$
   $= \{@l_0.ab \cap 1, @l_1.b \cap @l_0.1, @l_0.1 \cap @l_0.1\}$.
3) $D_{\mathtt{A}(l_0)\mathtt{A}(l_1)\mathtt{C}(l_0,l_1,a)}(@l_0.aa^{\smile}a)$
   $= \{@l_0.aa^{\smile}a, @l_1.a^{\smile}a, @l_0.a, @l_1.1\}$.

Proposition V.8. *Let $\rho = \rho'\mathtt{A}(l)$ be a SGCP. For any simple labelled graphs $\tilde{G}_1$ and $\tilde{G}_2$,*
(1) $\rho^{-1}(\tilde{G}_1 \parallel \tilde{G}_2) = \rho^{-1}(\tilde{G}_1) \parallel \rho^{-1}(\tilde{G}_2)$.
(2) if $l \in L(\tilde{G}_1)$, $\rho^{-1}(\tilde{G}_1 \cdot G_2) = \rho^{-1}(\tilde{G}_1) \cdot G_2$.
(3) if $L(\tilde{G}_1) = \{l\}$, $\rho^{-1}(\tilde{G}_1) = \{\tilde{G}_1\}$.

Sketch. These are proved by considering the range under the influence of ${\rho'}^{-1}$. □

Theorem V.9. $\tilde{\mathcal{G}}(D_\rho(\tilde{t})) = \rho^{-1}(\tilde{\mathcal{G}}(\tilde{t}))$.

Sketch. It is proved by induction on the pair (the length of ρ, the length of $\tilde{t}$) using Theorem V.5 for $\mathtt{C}(l_1, l_2, a)$ and using Proposition V.8 for $\mathtt{A}(l_1)$. □

Theorem V.10. *The following are equivalent:*
(1) $E_{l_t}(D_\rho(\tilde{t}))$;
(2) $\tilde{G}(\rho) \lhd \tilde{\mathcal{G}}(\tilde{t})^{l_t}$.

Proof. By Proposition V.2 and Theorem IV.11 and V.9. □

Summarizing the above discussion, the inclusion problem for Kleene allegories can be characterized by the derivatives on graphs as follows: REL $\models t_1 \leq t_2 \Leftrightarrow$ REL$^{\sim} \models @l_s.t_1 \leq @l_s.t_2$ (Proposition II.2) $\Leftrightarrow \forall l_t{\in}L.\ \tilde{\mathcal{G}}(@l_s.t_1)^{l_t} \lhd \tilde{\mathcal{G}}(@l_s.t_2)^{l_t}$ (Theorem II.9 and II.11) $\Leftrightarrow \forall l_t{\in}L.\forall\rho.\ \tilde{G}(\rho) \lhd \tilde{\mathcal{G}}(@l_s.t_1)^{l_t} \rightarrow \tilde{G}(\rho) \lhd \tilde{\mathcal{G}}(@l_s.t_2)^{l_t}$ ($\lhd$ is transitive and any labelled graph in $\tilde{\mathcal{G}}(@l_s.t_1)^{l_t}$ can be expressed by a SGCP by Corollary III.11) $\Leftrightarrow \forall l_t{\in}L.\forall\rho.\ E_{l_t}(D_\rho(@l_s.t_1)) \rightarrow E_{l_t}(D_\rho(@l_s.t_2))$ (Theorem V.10). Algorithm 2 is for deciding the above formula, where the domain of ρ is restricted to $(|t_1|+2)$-SGCPs by Corollary III.11 and cl^L is defined in the next subsection.

Algorithm 2 The inclusion problem for Kleene Allegories

```
Ensure: REL ⊨ t1 ≤ t2?
  Let L be a set of size |t1| + 2 and let ls, lt ∈ L.
  CL ⇐ cl^L(@ls.t1) ∪ cl^L(@ls.t2)
  (D, d) ⇐ ({t̃ ↦ {t̃} | t̃ ∈ CL}, 0)
  while d < 2^(#CL^2) do
    if E_lt(D(@ls.t1)) ∧ ¬E_lt(D(@ls.t2)) then
      return false
    end if
    pickup an event e nondeterministically
    (D, d) ⇐ (e(D), d + 1)
  end while
  return true
```

A. Complexity

Definition V.11. The *closure* of a labelled term $\tilde{t}$ with respect to a set of labels L, $\mathrm{cl}^L(\tilde{t})$, is defined inductively as follows. (For convenience, we also define $\mathrm{cl}^L(t)$ for term t.)
$\mathrm{cl}^L(0) := \bigcup_{l' \in L}\{@l'.0\}$; $\mathrm{cl}^L(1) := \bigcup_{l' \in L}\{@l'.1\}$;
$\mathrm{cl}^L(a) := \bigcup_{l' \in L}\{@l'.a, @l'.1\}$; $\mathrm{cl}^L(\breve{a}) := \bigcup_{l' \in L}\{@l'.\breve{a}, @l'.1\}$;
$\mathrm{cl}^L(t_1 \cup t_2) := \bigcup_{l' \in L}\{@l'.t_1 \cup t_2\} \cup \mathrm{cl}^L(t_1) \cup \mathrm{cl}^L(t_2)$;
$\mathrm{cl}^L(t_1 \cdot t_2) := \bigcup_{l' \in L}\{@l'.t_1 \cdot t_2\} \cup \mathrm{cl}^L(t_1) \cdot \{t_2\} \cup \mathrm{cl}^L(t_2)$;
$\mathrm{cl}^L(t_1^*) := \bigcup_{l' \in L}\{@l'.t_1^*\} \cup \mathrm{cl}^L(t_1) \cdot \{t_1^*\}$;
$\mathrm{cl}^L(t_1 \cap t_2) := \bigcup_{l' \in L}\{@l'.t_1 \cap t_2\} \cup (\mathrm{cl}^L(\tilde{t}_1) \Cap \mathrm{cl}^L(\tilde{t}_2))$;
$\mathrm{cl}^L(@l.t_1) := \mathrm{cl}^L(t_1)$; $\mathrm{cl}^L(\tilde{t}_1 \cdot t_2) := \mathrm{cl}^L(\tilde{t}_1) \cdot \{t_2\} \cup \mathrm{cl}^L(t_2)$;
$\mathrm{cl}^L(\tilde{t}_1 \cap \tilde{t}_2) := \mathrm{cl}^L(\tilde{t}_1) \Cap \mathrm{cl}^L(\tilde{t}_2)$.

$\mathrm{cl}^L(\tilde{T})$ denotes $\bigcup_{\tilde{t} \in \tilde{T}} \mathrm{cl}^L(\tilde{t})$ and cl^L is a closure operator.

Theorem V.12.
(1) $D_a^{l_1,l_2}(\tilde{t}) \subseteq \mathrm{cl}^L(\tilde{t})$, where $\{l_1, l_2\} \cup L(\tilde{t}) \subseteq L$.
(2) $D_\rho(\tilde{t}) \subseteq \mathrm{cl}^L(\tilde{t})$, where $L(\rho) \cup L(\tilde{t}) \subseteq L$.
(3) $\# \mathrm{cl}^L(t) \leq (2 \times |t| \times \#L)^{\mathrm{iw}(t)}$.
(4) $\# \mathrm{cl}^L(\tilde{t}) \leq (2 \times |\tilde{t}| \times \#L)^{\mathrm{iw}(\tilde{t})}$.

Proof. (1) is proved by induction on the length of $\tilde{t}$. (2) is proved by induction on ρ using (1).

(3) is proved by induction on the length of t. (4) is proved by induction on the length of $\tilde{t}$ using (3). For example, when $\tilde{t} = \tilde{t}_1 \cap \tilde{t}_2$, it is proved as follows:

$$\begin{aligned}
\# \mathrm{cl}^L(\tilde{t}) &\leq \# \mathrm{cl}^L(\tilde{t}_1) \times \# \mathrm{cl}^L(\tilde{t}_2) \\
&\leq (2 \times |\tilde{t}_1| \times \#L)^{\mathrm{iw}(\tilde{t}_1)} \times (2 \times |\tilde{t}_2| \times \#L)^{\mathrm{iw}(\tilde{t}_2)} \\
&\leq (2 \times |\tilde{t}| \times \#L)^{\mathrm{iw}(\tilde{t}_1)} \times (2 \times |\tilde{t}| \times \#L)^{\mathrm{iw}(\tilde{t}_2)} \\
&\leq (2 \times |\tilde{t}| \times \#L)^{\mathrm{iw}(\tilde{t}_1) + \mathrm{iw}(\tilde{t}_2)} \\
&\leq (2 \times |\tilde{t}| \times \#L)^{\mathrm{iw}(\tilde{t})}
\end{aligned}$$

□

By $\#L \leq |\tilde{t}_1| + 2$, $\mathrm{iw}(\tilde{t}) \leq |\tilde{t}|$, and Theorem V.12, $\#CL$ is an exponential of $(|t_1| + |t_2|)$. Therefore, the space complexity of Algorithm 2 is an exponential of the sum of the length of t_1 and t_2. Thus, the next theorem is derived in the same way as Theorem I.9.

Theorem V.13. *The inclusion problem for Kleene allegories is in EXPSPACE.*

B. A finite automaton construction

The derivatives on graphs derive an algorithm turning labelled terms into finite automata like that derivatives on strings derive algorithm regular expressions into finite automata.

A deterministic finite automaton (DFA) $\mathcal{A}$ is a five tuple (Q, A, δ, q_I, F), where Q is a set of states, A is an alphabet, $\delta : Q \times A \to Q$ is a transition function, $q_I \in Q$ is a initial state, and $F \subseteq Q$ is a set of acceptance states. For a string s, $\delta(q, s)$ denotes that, $\delta(q, s) = \{q\}$ if $s = \epsilon$, and $\delta(q, s) = \delta(\delta(q, s'), a)$ if $s = s'a$. we call that $\mathcal{A}$ accepts a string s if $\delta(q_I, s) \in F$.

Definition V.14 (DFA of a labelled term)**.** Let l_t be a label and let L be a set of labels such that $\{l_t\} \cup L(\tilde{t}) \subseteq L$. Then the DFA of a labelled term $\mathcal{A}(\tilde{t}, L, l_t) = (Q, A, \delta, q_I, F)$ is defined as follows: (a) $Q = \wp(\mathrm{cl}^L(\tilde{t}))^{\mathrm{cl}^L(\tilde{t})} \cup \{\bot\}$; (b) $A = \{\mathtt{A}(l_1) \mid l_1 \in L\} \cup \{\mathtt{C}(l_1, l_2, a) \mid l_1, l_2 \in L \text{ and } a \in A\} \cup \{\mathtt{F}(l_1) \mid l_1 \in L\}$; (c) $\delta(D, e) = e(D) \restriction \mathrm{cl}^L(\tilde{t})$ if $e(D)$ is defined, and $\delta(D, e) = \bot$ otherwise; (d) $q_I = \Delta(\tilde{\mathcal{T}}_{\{l_t\}}) \restriction \mathrm{cl}^L(\tilde{t})$; (e) $F = \{D \mid E_{l_t}(D(\tilde{t})) \text{ is } \mathit{true}\}$.

Proposition V.15. *Let ρ be any SGCP in the form of $\mathtt{A}(l_t)\rho'$.*

$$\delta^{\mathcal{A}(\tilde{t}, L, l_t)}(q_I^{\mathcal{A}(\tilde{t}, L, l_t)}, \rho') = D_\rho \restriction \mathrm{cl}^L(\tilde{t}).$$

Proof. It is proved by induction on the length of ρ using Theorem V.12 (2). □

Theorem V.16. *Let ρ be any SGCP in the form of $\mathtt{A}(l_t)\rho'$. The following are equivalent:*
(1) $E_{l_t}(D_\rho(\tilde{t}))$ is true;
(2) $\mathcal{A}(\tilde{t}, L, l_t)$ accepts ρ'.

Remark. The automata can be regarded as an extension of the automata for Kleene algebra with converse (Kleene allegories without intersection) in [6, Theorem 5.14] from path graphs to general graphs, when G in [6] is regarded as a path graph.

VI. Complexity: Language and Relations

In the previous section, it is shown that the inclusion problem for Kleene allegories is in EXPSPACE. In this section, we show that the inclusion problem for Kleene allegories is EXPSPACE-complete. In the language model, a lowerbound of the universality problem is shown by the next theorem.

Theorem VI.1 ([14, Theorem 2])**.** *The language universality problem for terms without $\bullet^{\smile}$ is EXPSPACE-complete.*

Also the next proposition shows that the equational theory over relation models is smaller than the equational theory on the language model.

Theorem VI.2 ([2])**.** *Let t_1 and t_2 be any terms without $\bullet^{\smile}$.*

$$\mathrm{REL} \models t_1 \leq t_2 \Longrightarrow \mathcal{L}(t_1) \subseteq \mathcal{L}(t_2).$$

We now show the next theorem, that is a bit stronger claim of Proposition I.12.

Theorem VI.3. *Let t_1 and t_2 be any terms without $\bullet^{\smile}$ such that $\cap$ does not occur in t_1.*

$$\mathrm{REL} \models t_1 \leq t_2 \iff \mathcal{L}(t_1) \subseteq \mathcal{L}(t_2).$$

Definition VI.4 (Tree unwound model (see e.g., [16, p.132]))**.** The *tree unwound model* of a relation model $M = (V, \{R_a\}_{a \in A})$, written M^t, is the relation model $(V \cdot (A \cdot V)^*, \{\{(\mathbf{v}v, \mathbf{v}vav') \mid (v, v') \in R_a\}\}_{a \in A})$.

The string expressed by $\mathbf{v}$, written $s(\mathbf{v})$, is inductively defined as follows: $s(\mathbf{v}) = \epsilon$ if $\mathbf{v} = \epsilon$, $s(\mathbf{v}) = s(\mathbf{v}')a$ if $\mathbf{v} = \mathbf{v}'a$, and $s(\mathbf{v}) = s(\mathbf{v}')$ if $\mathbf{v} = \mathbf{v}'v$.

For tree unwound models, the next propositions hold.

Proposition VI.5. *Let t be any term without $\bullet^{\smile}$.*
(1) $(\mathbf{v}v, \mathbf{v}v\mathbf{v}') \in R_t^{M^t} \iff (v, v\mathbf{v}') \in R_t^{M^t}$.
(2) If $(\mathbf{v}, \mathbf{v}') \in R_t^{M^t}$, there exists $\mathbf{v}''$ such that $\mathbf{v}' = \mathbf{v}\mathbf{v}''$.

Lemma VI.6.

(1) Let t be any term without $\bullet^{\smile}$ nor $\cap$.

$$(v, v') \in R_t^M \Longrightarrow \exists \mathbf{v} \in (V \cdot A)^*.(v, \mathbf{v}v') \in R_t^{M^t}.$$

(2) Let t be any term without $\bullet^{\smile}$.

$$\exists \mathbf{v} \in (V \cdot A)^*.(v, \mathbf{v}v') \in R_t^{M^t} \Longrightarrow (v, v') \in R_t^M.$$

(3) Let t be any term without $\bullet^{\smile}$.

$$(v, \mathbf{v}) \in R_t^{M^t} \iff (v, \mathbf{v}) \in R_{A^*}^{M^t} \wedge s(\mathbf{v}) \in \mathcal{L}(t).$$

Proof. These are proved by simple induction on the length of t using Proposition VI.5. ☐

Proof of Theorem VI.3. ($\Rightarrow$): By Theorem VI.2. ($\Leftarrow$): We show the contraposition. Let M be a counter relation model and let (v, v') be a pair of nodes in $R_{t_1}^M \setminus R_{t_2}^M$. By Lemma VI.6 (1) (2), there exists $\mathbf{v} \in (V \cdot A)^*$ such that $(v, \mathbf{v}v') \in R_{t_1}^{M^t}$ and $(v, \mathbf{v}v') \notin R_{t_2}^{M^t}$. By Lemma VI.6 (3), $(v, \mathbf{v}v') \in R_{A^*}^{M^t}$, $s(\mathbf{v}v') \in \mathcal{L}(t_1)$, and $s(\mathbf{v}v') \notin \mathcal{L}(t_2)$ hold. Therefore $s(\mathbf{v}v')$ is a counter string of $\mathcal{L}(t_1) \subseteq \mathcal{L}(t_2)$. Thus $\mathcal{L}(t_1) \not\subseteq \mathcal{L}(t_2)$. ☐

Corollary VI.7. *Let $A = \{a_1, \ldots, a_n\}$ and let t be any term without $\bullet^{\smile}$. The following are equivalent:*

(1) $\mathcal{L}(A^*) = \mathcal{L}(t)$;

(2) $\mathrm{REL} \models A^* \leq t$.

Proof. It is proved by letting t_1 be $(a_1 \cup \cdots \cup a_n)^*$ and letting t_2 be t in Theorem VI.3. Note that $\mathcal{L}(A^*) = \mathcal{L}(t)$ and $\mathcal{L}(A^*) \subseteq \mathcal{L}(t)$ are equivalent. ☐

Theorem VI.8.

(1) The universality problem for Kleene allegories is EXPSPACE-complete.

(2) The equational theory of Kleene allegories is EXPSPACE-complete.

Proof. (1): By Corollary VI.7 and Theorem VI.1, the problem is EXPSPACE-hard. From this and Theorem V.13, the problem is EXPSPACE-complete. (2): By (1) and Theorem V.13. ☐

VII. Future work

The derivatives on graphs allow us to obtain a finite automata construction algorithm for Kleene allegories. There are some natural directions for future work.

For instance, it would be interesting to apply the derivatives on graphs and the SGCPs to modal logics. The SGCPs in this paper would may be applicable to modal logics when the pathwidths of theirs models are restricted by a parameter. It would be desired some extensions of SGCPs related with *treewidth*. Also it would be interesting to extend with the *negation operator* in (representable) relation algebra. The following are related work in regard to extend with negation. (a) The undecidability of Kleene allegories with the negation operator is an immediate consequence of the undecidability of the relation algebra that has the operations, negation, union, and composition [23]. (b) The fragment of the relation algebra that the negation operator is only applied to characters is decidable [20]. (c) PDL (Propositional Dynamic Logic) with intersection and negation of characters is undecidable [15]. To the best our knowledge it is open whether Kleene allegories with negation of characters is decidable.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Number 16J08119.

References

[1] Hajnal Andréka and Dimitry A Bredikhin. The equational theory of union-free algebras of relations. *Algebra Universalis*, 33(4):516–532, 1995.

[2] Hajnal Andréka, Szabolcs Mikulás, and István Németi. The equational theory of kleene lattices. *Theoretical Computer Science*, 412(52):7099–7108, 2011.

[3] Valentin Antimirov. Partial derivatives of regular expressions and finite automaton constructions. *Theoretical Computer Science*, 155(2):291–319, 1996.

[4] 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 *International Workshop on Descriptional Complexity of Formal Systems*, pages 45–59, 2016.

[5] Patrick Blackburn, Johan van Benthem, and Frank Wolter. *Handbook of modal logic*, volume 3. Elsevier, 2006.

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

[7] Hans L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1):1 – 45, 1998.

[8] Mikołaj Bojańczyk and Michał Pilipczuk. Definability equals recognizability for graphs of bounded treewidth. In *31st Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 407–416, 2016.

[9] Paul Brunet and Damien Pous. Petri Automata for Kleene Allegories. In *2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 68–79, 2015.

[10] Janusz A. Brzozowski. Derivatives of Regular Expressions. *Journal of the ACM*, 11(4):481–494, 1964.

[11] John Horton Conway. *Regular algebra and finite machines*. Chapman and Hall, 1971.

[12] David Eppstein. Parallel recognition of series-parallel graphs. *Information and Computation*, 98(1):41–55, 1992.

[13] Peter J Freyd and Andre Scedrov. *Categories, allegories*, volume 39. Elsevier, 1990.

[14] Martin Fürer. The complexity of the inquivalence problem for regular expressions with intersection. In *International Colloquium on Automata, Languages, and Programming*, pages 234–245, 1980.

[15] Stefan Göller, Markus Lohrey, and Carsten Lutz. PDL with intersection and converse: satisfiability and infinite-state model checking. *The Journal of Symbolic Logic*, 74(01):279–314, 2009.

[16] David Harel, Dexter Kozen, and Jerzy Tiuryn. *Dynamic Logic*. MIT Press, 2000.

[17] Harry B Hunt, Daniel J Rosenkrantz, and Thomas G Szymanski. On the equivalence, containment, and covering problems for the regular and context-free languages. *Journal of Computer and System Sciences*, 12(2):222–268, 1976.

[18] Dexter Kozen. A completeness theorem for kleene algebras and the algebra of regular events. In *Sixth Annual IEEE Symposium on Logic in Computer Science*, pages 214–225, 1991.

[19] Dexter Kozen and Frederick Smith. Kleene algebra with tests: Completeness and decidability. In *International Workshop on Computer Science Logic*, pages 244–259. Springer, 1996.

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

[21] Neil Robertson and Paul D Seymour. Graph minors. I. Excluding a forest. *Journal of Combinatorial Theory, Series B*, 35(1):39–61, 1983.

[22] Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177 – 192, 1970.

[23] Alfred Tarski and Steven R Givant. *A formalization of set theory without variables*, volume 41. American Mathematical Soc., 1987.