Image Reflection on Process Graphs of 1-Free Regular Expressions Modulo Bisimilarity

Yuanrui Zhang[a], Xinxin Liu[b]

[a]*College of Software, Nanjing University of Aeronautics and Astronautics, Nanjing, China*
[b]*Institute of Software, Chinese Academy of Science, Beijing, China*

Abstract

We study a phenomenon called "image reflection" on a type of characterization graphs — LLEE charts — for 1-free regular expressions modulo bisimularity. This property, stating that the images of a bisimulation function from an LLEE chart actually impose a special LEE structure corresponding to the LLEE chart, is recognized by our proposed "well-structured looping-back charts" as a sub-LLEE-structure of LLEE charts. As an application, our study naturally leads to a novel proof for the completeness of the inference system **BBP** for 1-free regular expressions, due to the correspondence between 1-free regular expressions and the provable solutions of LEE/LLEE charts. Compared to the previous approach, our proof is more direct in the sense that it does not rely on a graph transformation procedure on LLEE charts in which at each step two bisimilar nodes have to be carefully selected and merged together according to selection rules. Our observation on LLEE charts is useful to understand the completeness problems of regular expressions modulo bisimilarity from a new angle, and can be also helpful for solving the completeness problems of other expressions that share similar graph structures.

Keywords: 1-Free Regular Expressions, Process Graphs, Process Semantics, Bisimulation, Axiomatization, Completeness

1. Introduction

Regular expressions, proposed by Kleene [1], are a formal language consisting of elements a (*actions*), 0 (*deadlock*), 1 (*skip*) and their compositions $e_1 + e_2$ (*non-deterministic choice*), $e_1 \cdot e_2$ (*sequence*) and e^* (*iterations*). The

original star operator was a binary one: $e_1 * e_2$ (meaning iterating e_1 for finite times and then executing e_2, or iterating e_1 infinitely). Later Copi et al. in their work [2] proposed to use the unary operator e^* instead of $e_1 * e_2$. Regular expressions have many applications in computer science. Especially, as a formalism embedded in dynamic logics [3] and Kleene algebra with tests (KAT) [4], it can be used for specification and verification of different systems. Refer to work like [5, 6, 7, 8, 9] for examples.

Axiomatisation of various types of equivalence for regular expressions has been studied for years. It is strongly related to the axiomatisation of some important theories based on regular expressions, for example, the extensions of Kleene algebra, such as KAT [4], concurrent Kleene algebra [10] and guarded KAT [11, 12]. To "axiomatize" we mean to build a finite set of inference rules (forming an *inference system*) for deriving equations that are semantically correct (*sound*). When all semantically-correct equations are derivable, an inference system is called *complete*. Salomaa [13] gave a complete inference system for deriving equations of regular expressions under the semantics of execution traces, where the behaviours of a regular expression are interpreted as a set of traces of actions. Milner was the first to study the axiomatisation of equations of regular expressions under the *process semantics* [14]. In this semantics, the behaviour of a regular expression is interpreted as a set of *bisimilar charts*, instead of execution traces. In [14], Milner built a sound inference system **Mil** for regular expressions (there he called a regular expression a "star expression"). **Mil** is obtained from Solomaa's inference system by removing the critical left associative rule (**lass**): $e_3 \cdot (e_1 + e_2) = e_3 \cdot e_1 + e_3 \cdot e_2$, which does not hold under the process semantics anymore. It was observed that the completeness problem of **Mil** could be much harder, because the Solomaa's proof approach, which relies on (**lass**), cannot be applied.

Since [14], some partial solutions for the completeness of **Mil** have been proposed. They consider various subsets of regular expressions together with their adapted inference systems from **Mil**: [15] proposed an axiomatisation for a subset of regular expressions where there are no 0 and 1, but a binary star operator $e_1 * e_2$ instead of e^*. Its completeness was proved in [16]. [17] built a complete inference system for regular expressions with 0 and 1, but with the "terminal cycle" e^ω rather than e^*. e^ω is semantically equivalent to $e^* \cdot 0$ in regular expressions.

Recently, Grabmayer and Fokkink in [18] axiomatized the so-called *1-free regular expressions*, in which there are 0 and a binary star operator $e_1 * e_2$ (denoted by $e_1 \circledast e_2$ in [18]) instead of e^*, but no 1. The inference system tai-

lored from **Mil** for 1-free regular expressions is denoted by **BBP**. To prove the completeness of **BBP**, the authors developed a characterization graph, called *LEE/LLEE charts*, for 1-free regular expressions. LEE/LLEE charts captures a special form (expressed as "LEE/LLEE properties") of equations which always has a set of 1-free regular expressions derivable from **BBP**, formally called "provable solution". Moreover, in LEE/LLEE charts, deriving the equivalence of two provable solutions from **BBP** is always possible. These capabilities are actually the critical part for the completeness proof of **BBP**. Later in [19], LEE/LLEE charts were further extended and utilized for proving the completeness of **Mil** for regular expressions.

In this paper, we study the images of a bisimulation function from an LLEE chart. A bisimulation function f is a special bisimulation relation that maps each node of an LLEE chart $\mathcal{G}$ to a (unique) node of a chart $\mathcal{H}$. The observation we present in this paper is that on f, one can obtain the structure of an LEE chart on $\mathcal{H}$ from certain "minimal pre-images" of sub-charts of $\mathcal{H}$ on $\mathcal{G}$ that we call *well-structured looping-back charts*. We call this phenomenon *image reflection*. With this we can further prove that f preserves the *LLEE property* of $\mathcal{G}$, as the main result obtained in [18]. As a direct application, we will show that a novel proof can be obtained for the completeness of **BBP**, which is more direct compared to [18], in the sense that our proof does not rely on the "connecting through-to" graph transformations on LLEE charts (cf. [18]).

To illustrate our main idea, consider a simple example as depicted in Fig. 1. Let us focus on chart $\mathcal{G}$ (i.e. $\hat{\mathcal{G}}/\mathcal{G}$) and its corresponding set of two equations on its right side. To see this correspondence relation between charts and equations, notice that nodes x and x' [1] in chart $\mathcal{G}$ correspond to expressions $s(x)$ and $s(x')$ in the equations respectively, and each transition from a node exactly corresponds to one term on the right side of the equation the node corresponds to. For example, transition $x \xrightarrow{a} x'$ of $\mathcal{G}$ corresponds to term $a \cdot s(x')$ on the right side of equation $s(x) = a \cdot s(x') + b \cdot s(x')$. $s(x)$ and $s(x')$ together are a provable solution of the equations, because the two equations can be derived according to the inference system **BBP** (will be given in Table 2). Chart $\mathcal{G}$ is an LLEE chart, which always has a provable solution (here as expressions $s(x)$ and $s(x')$).

[1] To distinguish node names from action names in the figures appearing in this paper, we underline all the node names (e.g. $\underline{X}$) in the figures.

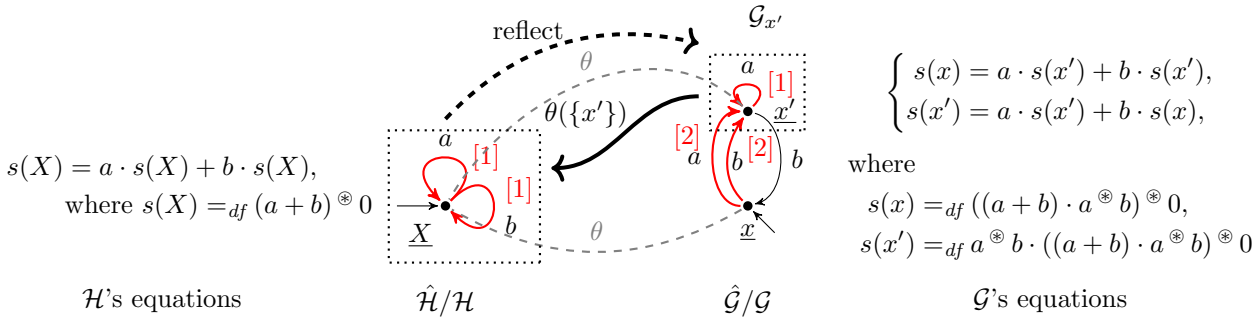

Figure 1: A simple example illustrating our idea

To prove the completeness of **BBP**, [18] adopts a so-called "minimization strategy", which will be introduced in Sect. 5, and where one critical step is to minimize an LLEE chart under bisimularity, and to prove that the obtained smallest chart bisimularly equivalent to this LLEE chart is still an LLEE chart. (The smallest chart bisimularly equivalent to a chart is also called the *bisimulation collapse* of this chart. In a bisimulation collapse, there are no distinct bisimilar nodes.) The approach taken in [18] is to divide the minimization process of an LLEE chart into simplification steps. At each step, a "connecting x_1 through-to x_2" graph transformation is performed by merging two carefully-selected bisimilar nodes x_1 and x_2. Then it is sufficient to prove that the resulting chart after each simplification step is still an LLEE chart. For example, in Fig. 1, a simplification step can be taken by merging nodes x and x' of chart $\mathcal{G}$ into node X of chart $\mathcal{H}$ (i.e. $\hat{\mathcal{H}}/\mathcal{H}$). Chart $\mathcal{H}$, with its equation shown on the left side, is also an LLEE chart.

Our work concentrates on the bisimulation function $\theta =_{df} \{x \mapsto X, x' \mapsto X\}$ from $\mathcal{G}$ to $\mathcal{H}$. By considering well-structured looping-back charts as a particular type of sub-structures in $\mathcal{H}$, we will directly prove that function θ actually preserves the LLEE structure of chart $\mathcal{G}$ using the image-reflection property, without performing the "connecting x_1 through-to x_2" transformation.

In the following sections, we will gradually clarify the above idea as we introduce our work in detail. In Sect. 2, we introduce the necessary background knowledge needed to understand this work. Sect. 3 proposes the concepts of looping-back charts and well-structuredness for LLEE charts, and provides an analysis of their properties and relationship. In Sect. 4, we introduce the image-reflection property on LLEE charts and prove the main result Theo-

rem 4.1. In Sect. 5, we show how image reflection can be utilized to prove the completeness of **BBP** and obtain the same result as in [18]. Sect. 6 concludes this paper and discusses about some possible future work.

2. Preliminaries

In this section we introduce the necessary background related to this topic. Some of the material is not directly used in our work, but is necessary for understanding the minimization strategy proposed in Sect. 5. We tag this material as "(*)".

2.1. 1-Free Regular Expressions and Charts

1-Free Regular Expressions. A 1-free regular expression $e \in RExp_{1\text{-}free}$ is defined as follows in BNF form:

$$e =_{df} a \mid 0 \mid e + e \mid e \cdot e \mid e \circledast e,$$

where $a \in Act$ is called an *action*, ranged over by a, b, c, d. The expression $e \circledast e$ is the binary star operator, $e_1 \cdot e_2$ is sometimes abbreviated as $e_1 e_2$. We use $\equiv$ to represent the syntactic identity between two 1-free regular expressions.

Chart. A *chart* is a 5-tuple $\mathcal{G} = \langle V, A, v, \rightarrow, \surd \rangle$, where V is a finite set of *nodes*, usually ranged over by x, y, z, X, Y, Z; $A \subseteq Act$ is a finite set of actions; $v \in V$ is the *initial node*; $\rightarrow : V \times A \times (V \cup \{\surd\})$ is a set of transitions; $\surd$ is a special node called *termination*. Each transition $\langle X, a, Y \rangle \in \rightarrow$ and transition $\langle X, a, \surd \rangle \in \rightarrow$ are also written as $X \xrightarrow{a} Y$ and $X \xrightarrow{a} \surd$ respectively, where $a \in A$. We also call $X \xrightarrow{a} \surd$ a *terminal transition*. A sequence of transitions is often written as: $X_1 \xrightarrow{a_1} X_2 \xrightarrow{a_2} ... \xrightarrow{a_n} X_n \xrightarrow{b} \xi$ $(n > 0)$, where ξ is a node or $\surd$, $X_1 \xrightarrow{a_1} X_2$, ..., $X_n \xrightarrow{b} \xi$ are n transitions. In this paper, all nodes in a chart are reachable from the initial node.

Sometimes we ignore symbol a and simply write a transition $X \xrightarrow{a} Y$ as $X \rightarrow Y$. We use $X \equiv Y$ to represent that nodes X and Y are the same one. Use $\cdot$ to represent an arbitrary node, e.g., a transition $X \rightarrow \cdot$. The reflexive and transitive closure of $\rightarrow$ is defined as: $X \rightarrow^* Y$ if either (i) $X \equiv Y$ or (ii) $X \rightarrow Z$ and $Z \rightarrow^* Y$ for some Z. The transitive closure $X \rightarrow^+ Y$ is defined s.t. $X \rightarrow Z$ and $Z \rightarrow^* Y$ for some Z. A *path* is a finite or infinite sequence of transitions. We use $X \xrightarrow[\not\equiv Z]{} Y$, $X \xrightarrow[\not\equiv Z]{}{}^* Y$ or $X \xrightarrow[\not\equiv Z]{}{}^+ Y$ to represent a

path where all intermediate nodes (not including X) are not Z. A *loop* is a finite path of the form: $X \rightarrow^+ X$, i.e., it starts from and returns to the same node X.

Chart Interpretation (*). The *chart interpretation* of a 1-free regular expression $e \in RExp_{1\text{-}free}$ is a chart

$$\mathcal{C}(e) =_{df} \langle V(e), A(e), e, \rightarrow(e), \surd \rangle,$$

which is obtained by expanding e according to its operational semantics given in Table 1. e is the initial node. $\rightarrow(e)$ consists of all the transitions during the expansion of e. $V(e)$ is the finite set of expressions reachable from e through transitions in $\rightarrow(e)$. $A(e)$ is the set of actions appeared in e.

$$\frac{}{a \xrightarrow{a} \surd} \qquad \frac{e_1 \xrightarrow{a} \surd}{e_1 + e_2 \xrightarrow{a} \surd} \qquad \frac{e_2 \xrightarrow{a} \surd}{e_1 + e_2 \xrightarrow{a} \surd} \qquad \frac{e_1 \xrightarrow{a} \surd}{e_1 \cdot e_2 \xrightarrow{a} e_2} \qquad \frac{e_1 \xrightarrow{a} \surd}{e_1 \circledast e_2 \xrightarrow{a} e_1 \circledast e_2}$$

$$\frac{e_2 \xrightarrow{a} \surd}{e_1 \circledast e_2 \xrightarrow{a} \surd} \qquad \frac{e_1 \xrightarrow{a} e_1'}{e_1 + e_2 \xrightarrow{a} e_1'} \qquad \frac{e_2 \xrightarrow{a} e_2'}{e_1 + e_2 \xrightarrow{a} e_2'} \qquad \frac{e_1 \xrightarrow{a} e_1'}{e_1 \cdot e_2 \xrightarrow{a} e_1' \cdot e_2}$$

$$\frac{e_1 \xrightarrow{a} e_1'}{e_1 \circledast e_2 \xrightarrow{a} e_1' \cdot (e_1 \circledast e_2)} \qquad \frac{e_2 \xrightarrow{a} e_2'}{e_1 \circledast e_2 \xrightarrow{a} e_2'}$$

Table 1: Operational semantics of 1-free regular expressions

2.2. Bisimulation Equivalence and Inference System ***BBP***

Given two charts $\mathcal{G} = \langle V_1, A_1, v_1, \rightarrow_1, \surd \rangle$ and $\mathcal{H} = \langle V_2, A_2, v_2, \rightarrow_2, \surd \rangle$, a *bisimulation relation* $\mathfrak{R} \subseteq V_1 \times V_2$ between the two charts is a binary relation such that for any $(X, Y) \in \mathfrak{R}$, the following hold:

(1) if $X \xrightarrow{a} X'$, then there is some Y'' such that $Y \xrightarrow{a} Y''$ and $(X', Y'') \in \mathfrak{R}$;

(2) if $Y \xrightarrow{a} Y'$, then there is some X'' such that $X \xrightarrow{a} X''$ and $(X'', Y') \in \mathfrak{R}$;

(3) $X \xrightarrow{a} \surd$ iff $Y \xrightarrow{a} \surd$.

We sometimes denote a bisimulation relation $\mathfrak{R} \subseteq V_1 \times V_2$ between two charts as: $\mathfrak{R} : V_1 \to V_2$, and call V_1 the *domain* and V_2 the *codomain* of relation $\mathfrak{R}$. Given $\mathfrak{R} : V_1 \to V_2$, for any set $A \subseteq V_1$ and $B \subseteq V_2$, we define $\mathfrak{R}(A) =_{df} \{X \mid X \in V_2, \exists y \in A.(y, X) \in \mathfrak{R}\}$; and define $\mathfrak{R}^{-1}(B) =_{df} \{x \mid x \in V_1, \exists Y \in B.(x, Y) \in \mathfrak{R}\}$. A bisimulation relation $\mathfrak{R} : V_1 \to V_2$ is called a *bisimulation function* if $\mathfrak{R}$ is a function from V_1 to V_2. We often use θ to represent a bisimulation function.

Given two charts $\mathcal{G} = \langle V_1, A_1, v_1, \to_1, \surd \rangle$ and $\mathcal{H} = \langle V_2, A_2, v_2, \to_2, \surd \rangle$, two nodes $X \in V_1$ and $Y \in V_2$ are *bisimilar*, denoted by $X \sim Y$, if there is a bisimulation relation $\mathfrak{R} \subseteq V_1 \times V_2$ such that $(X, Y) \in \mathfrak{R}$. Charts $\mathcal{G}$ and $\mathcal{H}$ are said to be bisimilar, denoted by $\mathcal{G} \sim \mathcal{H}$, if there is a bisimulation relation $\mathfrak{R} \subseteq V_1 \times V_2$ satisfying $(v_1, v_2) \in \mathfrak{R}$. The *bisimulation equivalence* $e_1 = e_2$ between two 1-free regular expressions $e_1, e_2 \in \mathit{RExp}_{\mathit{1\text{-}free}}$, is defined if $\mathcal{C}(e_1) \sim \mathcal{C}(e_2)$.

*Inference System **BBP** (*).* Inference system **BBP** for 1-free regular expressions modulo bisimulation equivalence $=$ is given in Table 2 as a set of rules on equations, where (A1) - (A9) are axioms, (R1) is an inference rule. Since our work in this paper will not concern this inference system, we will not explain the meaning of each rule. One can refer to [18] for more details. It is pointed out in recent work [20] that (A9) can be derived using the other rules in **BBP**. Here, however, for the sake of better understanding and comparison, we just keep the inference system the same form as proposed in [18].

Given an equation $e_1 = e_2$, we write $\vdash_{\mathit{BBP}} e_1 = e_2$ to mean that $e_1 = e_2$ can be derived by applying the rules in Table 2 in a finite number of steps. Inference system **BBP** is sound, that is, $\vdash_{\mathit{BBP}} e_1 = e_2$ implies $e_1 = e_2$ for any $e_1, e_2 \in \mathit{RExp}_{\mathit{1\text{-}free}}$. The completeness of **BBP** was first obtained as the main result in [18], which means that for every semantically-correct equation $e_1 = e_2$ with $e_1, e_2 \in \mathit{RExp}_{\mathit{1\text{-}free}}$, $\vdash_{\mathit{BBP}} e_1 = e_2$.

Chart and 1-Free-Regular-Expression Equations (*). An *evaluation* $s : V \to \mathit{RExp}_{\mathit{1\text{-}free}}$ defines a function that maps each node of V to a 1-free regular expression. Given an evaluation s, a chart $\mathcal{G} = \langle \{X_i\}_{i \in [1,n]}, A, X_1, \to, \surd \} \rangle$ corresponds to a set of equations w.r.t. s:

$$s(\mathcal{G}) =_{df} \left\{ s(X_i) = \sum_{j=1}^{n} e_{ij} \cdot s(X_j) + \sum_{k=1}^{m} f_{ik} \right\}_{i \in [1,n]},$$

$$e_1 + e_2 = e_1 + e_2 \quad (A1)$$
$$(e_1 + e_2) + e_3 = e_1 + (e_2 + e_3) \quad (A2)$$
$$e + e = e \quad (A3)$$
$$(e_1 + e_2) \cdot e_3 = e_1 \cdot e_3 + e_2 \cdot e_3 \quad (A4)$$
$$(e_1 \cdot e_2) \cdot e_3 = e_1 \cdot (e_2 \cdot e_3) \quad (A5)$$
$$e + 0 = e \quad (A6)$$
$$0 \cdot e = 0 \quad (A7)$$
$$e_1{}^{\circledast}e_2 = e_1 \cdot (e_1{}^{\circledast}e_2) + e_2 \quad (A8)$$
$$(e_1{}^{\circledast}e_2) \cdot e_3 = e_1{}^{\circledast}(e_2 \cdot e_3) \quad (A9)$$
$$\frac{e = e_1 \cdot e + e_2}{e = e_1{}^{\circledast}\, e_2} \ (R1)$$

Table 2: Inference system **BBP** for 1-free regular expressions

where in each equation $s(X_i) = \sum_{j=1}^{n} e_{ij} \cdot s(X_j) + \sum_{k=1}^{m} f_{ik}$, the term $e_{ij} \cdot s(X_j)$ ($e_{ij} \in A$) appears iff $X_i \xrightarrow{e_{ij}} X_j$ is a transition in $\rightarrow$; the term f_{ik} appears iff $X_i \xrightarrow{f_{ik}} \surd$ is a transition in $\rightarrow$. An evaluation s is a *provable solution* (sometimes simply *solution*) of $\mathcal{G}$, if each equation of $s(\mathcal{G})$ can be derived according to **BBP** in Table 2, denoted by $\vdash_{BBP} s(\mathcal{G})$. Call $s(X_1)$ the *primary solution* of $\mathcal{G}$.

2.3. Loop Charts, LEE Charts and LLEE Charts [18]

A *loop chart* $\mathcal{A} = \langle V, A, X, \rightarrow, \surd \rangle$ is a chart with the initial node X called the *start node* and satisfying the following three conditions:

(L1) There is an infinite path from X in $\mathcal{A}$;

(L2) In $\mathcal{A}$ every infinite path from X returns to X after a finite number of transitions;

(L3) $\surd$ is not in $\mathcal{A}$.

The transitions starting from X are called *loop-entry transitions*. Other transitions are called *body transitions*. Call $V \setminus \{X\}$ the *body* of $\mathcal{A}$.

For example, in Fig. 1, chart $\mathcal{H}$ is a loop chart. Chart $\mathcal{G}$ with the start node x is not a loop chart as it violates (L2).

In a chart $\mathcal{G} = \langle V, A, v, \rightarrow, \surd \rangle$, given a $X \in V$ and the set $\mathbf{E}$ of loop-entry transitions from X, a $\langle X, \mathbf{E} \rangle$-*generated chart* is a sub-chart $\mathcal{C} = \langle V_1, A, X, \rightarrow_1, \surd \rangle$ of $\mathcal{G}$ consisting of all the transitions each of which is on a path that starts from a transition $X \rightarrow Y$ in $\mathbf{E}$, and continues with other transitions of $\mathcal{G}$ until reaching X again. Formally, $\rightarrow_1 =_{df} \mathbf{E} \cup \{Y \rightarrow Z \mid X \xrightarrow[\not\equiv X]{} X_1 \xrightarrow[\not\equiv X]{}{}^{*} Y \rightarrow Z, (X \rightarrow X_1) \in \mathbf{E}\} \cup \{Y \rightarrow \surd \mid X \xrightarrow[\not\equiv X]{} X_1 \xrightarrow[\not\equiv X]{}{}^{*} Y \rightarrow \surd, (X \rightarrow X_1) \in \mathbf{E}\}$, V_1 is the set of nodes appearing in $\rightarrow_1$.

A $\langle X, \mathbf{E} \rangle$-generated chart of $\mathcal{G}$ is called a *loop sub-chart* of $\mathcal{G}$ if it is a loop chart.

In Fig. 1, chart $\mathcal{H}$, as the $\langle X, \{X \xrightarrow{a} X, X \xrightarrow{b} X\} \rangle$-generated chart, is a loop sub-chart of itself. Chart $\mathcal{G}_{x'} : \langle \{x'\}, \{a, b\}, x', \{x' \xrightarrow{a} x'\}, \surd \rangle$, as the $\langle x', \{x' \xrightarrow{a} x'\} \rangle$-generated chart, is a loop sub-chart of $\mathcal{G}$.

LEE Charts. An *elimination of a loop chart* $\mathcal{A}$ starting from X is a transformation process (namely *elim*$(\mathcal{A}, X)$) in which we first remove all of the loop-entry transitions of $\mathcal{A}$ from X, and then remove all the nodes and transitions that become unreachable (after the first step).

Definition 2.1 (LEE Property). *A chart $\mathcal{G}$ is said to satisfy the* loop existence and elimination *(LEE) property, if there exists a transformation process in which each loop sub-chart $\mathcal{A}$ of $\mathcal{G}$ (with X its start node) can be eliminated through the process elim$(\mathcal{A}, X)$, so that after the process, $\mathcal{G}$ results in a chart without an infinite path.*

A chart is called an *LEE chart* if it satisfies the LEE property.

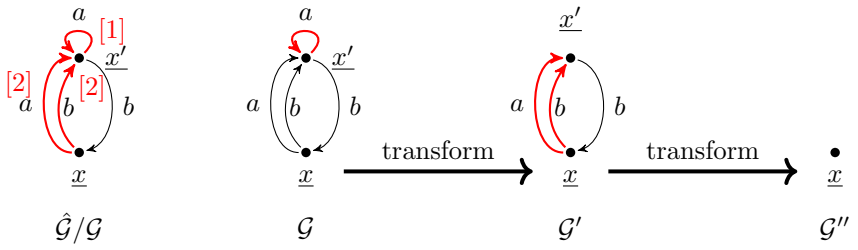

Figure 2: An example of elimination processes

Fig. 2 shows how the LEE chart $\mathcal{G}$ of Fig. 1 can be eliminated in two steps: In the first step, the loop sub-chart $\mathcal{G}_{x'}$ is eliminated through the process $elim(\mathcal{G}_{x'}, x')$, in which we remove the loop-entry transition $x' \xrightarrow{a} x'$. After the first step the resulting chart $\mathcal{G}'$ becomes a loop sub-chart of itself. In the second step, $\mathcal{G}'$ is eliminated through the process $elim(\mathcal{G}', x)$, in which we firstly remove two loop-entry transitions $x \xrightarrow{a} x'$ and $x \xrightarrow{b} x'$, and then remove node x' and transition $x' \xrightarrow{b} x$ that become unreachable. This leaves the resulting chart $\mathcal{G}''$ having no infinite paths.

As illustrated in the example above, the process of loop elimination is actually conducted in an inside-out manner, in the sense that a sub-chart $\mathcal{A}$ of a chart $\mathcal{G}$ can only become a loop sub-chart after all the loop sub-charts of $\mathcal{A}$ have been eliminated previously. This is an important intuition of our method purposed in Sect. 4.

For an elimination process on an LEE chart $\mathcal{G}$, a function $\hat{\mathcal{G}}$, called an *LEE witness*, is used to indicate the order of removing the loop-entry transitions in $\mathcal{G}$. $\hat{\mathcal{G}}$ maps the transitions of $\mathcal{G}$ to a down-closed set of natural numbers, we call *order numbers*. It maps a loop-entry transition $X \to Y$ to a number $n > 0$, denoted by $X \to_{[n]} Y$, if $X \to Y$ is removed at the nth step of the process; and maps body transitions to 0, denoted by $\cdot \to_{[0]} \cdot$ or $\cdot \to_{bo} \cdot$. Write an LEE chart $\mathcal{G}$ as $\hat{\mathcal{G}}/\mathcal{G}$.

In chart $\mathcal{G}$ of Fig. 2, the tagged numbers on arrows indicate an LEE witness $\hat{\mathcal{G}}$ of the elimination process in Fig. 2, where $\hat{\mathcal{G}}(x' \xrightarrow{a} x') =_{df} 1$, $\hat{\mathcal{G}}(x \xrightarrow{a} x') =_{df} 2$, $\hat{\mathcal{G}}(x \xrightarrow{b} x') =_{df} 2$, $\hat{\mathcal{G}}(x' \xrightarrow{b} x) =_{df} 0$.

LLEE Charts. The elimination process of an LEE chart is not necessarily unique. If in an LEE chart there exists an elimination process *elim* of a loop sub-chart in which no loop-entry transitions are removed from a node in the body of a previously eliminated loop sub-chart, then say the LEE chart satisfies the *layered loop existence and elimination* (LLEE) property. The LEE chart is then called an *LLEE chart*. A witness $\hat{\mathcal{G}}$ of an LLEE chart $\mathcal{G}$ is thus called an *LLEE witness*.

It is easy to see that chart $\mathcal{G}$ of Fig. 2 is also an LLEE chart.

Not all LEE witnesses are LLEE witnesses. Considering the chart $\hat{\mathcal{E}}_1/\mathcal{E}_1$ in Fig. 3, from its elimination process we see that $\hat{\mathcal{E}}_1$ is an LEE witness. However, it is not an LLEE witness, because when performing its elimination at step 3, loop-entry transition $X \xrightarrow{b_1}_{[3]} Z$ is from the body of the loop sub-chart $\langle \{Z, X\}, \{a_1, a_2, a_3, a_4, b_1, d_1, d_2\}, Z, \{Z \xrightarrow{a_2} X, X \xrightarrow{b_1} Z\}, \surd \rangle$ of $\mathcal{E}_1'$ after

step 1.

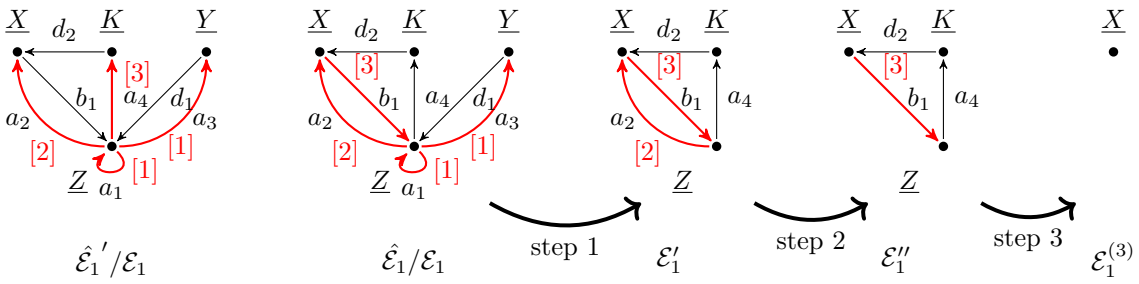

Figure 3: An example of a non-LLEE LEE witness

In an LLEE chart, a relation $X \curvearrowright Y$ is defined if there exists a path from X to Y that begins with a loop-entry transition and continues through subsequent body transitions without reaching X again. Or formally, $X \xrightarrow[\neq X]{}_{[m]} \cdot \xrightarrow[\neq X]{}^{*}_{bo} Y$ for some $m > 0$. The transitive closure $X \curvearrowright^{+} Y$ is defined s.t. (i) $X \curvearrowright Y$, or (ii) $X \curvearrowright^{+} Z$ and $Z \curvearrowright Y$ for some node Z.

The following property of $\curvearrowright$ directly comes from the LLEE property. Refer to [18] for more details.

Proposition 2.1. *The relation $\curvearrowright^{+}$ in an LLEE chart $\mathcal{G}$ is a well-founded, strict partially-ordered relation.*

Sovability of LLEE Charts (*). As stated in Sect. 1, LLEE charts characterize regular expressions modulo bisimulation equivalence $=$, and furthermore, they admit unique solutions up to BBP. We list the following results directly obtained from [18]. They form the foundation of minimization strategy introduced in Sect. 5.

Proposition 2.2 (P1). *Every LLEE chart $\hat{\mathcal{G}}/\mathcal{G}$ has a provable solution s.*

Proposition 2.3 (P2). *For every 1-free regular expression e, there is an LLEE chart $\hat{\mathcal{G}}/\mathcal{G}$ with an initial node X such that $\mathcal{G}$ has a provable solution s with $s(X) =_{df} e$.*

Proposition 2.4 (P3). *For any provable solutions s_1 and s_2 of an LLEE chart $\mathcal{G}$, $\vdash_{BBP} s_1(X) = s_2(X)$ for any node X of $\mathcal{G}$.*

Proposition 2.5 (P4). *Given two charts* $\mathcal{G} = \langle V_1, A_1, v_1, \to_1, \surd \rangle$, $\mathcal{H} = \langle V_2, A_2, v_2, \to_2, \surd \rangle$ *and a bisimulation function* $\theta : V_1 \to V_2$ *between them, if* $s : V_2 \to RExp_{1\text{-}free}$ *is a provable solution of* $\mathcal{H}$*, then* $s \circ \theta : V_1 \to RExp_{1\text{-}free}$ *is a provable solution of* $\mathcal{G}$*.*

3. Looping-back Charts and Images

In this section, we introduce the looping-back charts and well-structuredness for studying image reflection of bisimulation function further in Sect. 4. Intuitively, a looping-back chart is a sub-LLEE-chart of an LLEE chart. The well-structuredness is a special property followed by looping-back charts in an LLEE chart, which play the central role of proving the completeness of **BBP** later in Sect. 5.

Sect. 3.1 will introduce some basic notations used in the following subsections. In Sect. 3.2, we will introduce looping-back charts, while in Sect. 3.3, we will introduce the notion of looping-back-chart images and propose the notion of well-structuredness.

3.1. Some Conventions of Notations

In the rest of the paper, for convenience, we use the conventions of notations stated as follows.

Given a chart $\mathcal{G}$, we often use $V_\mathcal{G}$, $A_\mathcal{G}$, $s_\mathcal{G}$ and $\to_\mathcal{G}$ to represent its components. We usually ignore the initial state $s_\mathcal{G} \in V_\mathcal{G}$ and simply understand a chart $\langle V_\mathcal{G}, A_\mathcal{G}, s_\mathcal{G}, \to_\mathcal{G}, \surd \rangle$ as a 4-tuple: $\langle V_\mathcal{G}, A_\mathcal{G}, \to_\mathcal{G}, \surd \rangle$. This is because in our work, we do not need to distinguish the initial nodes to relate to the concepts like chart interpretation (Section 2.1) and primary solution (Section 2.2).

Given a set of nodes $N \subseteq V_\mathcal{G}$ of a chart $\mathcal{G}$, $\mathbf{G}_\mathcal{G}(N)$ (or simply $\mathbf{G}(N)$ when $\mathcal{G}$ is clear in the context) denotes the *chart of a set* N *of nodes* which consists of all nodes of N and all transitions between these nodes. Formally, $\mathbf{G}_\mathcal{G}(N) =_{df} \langle N, A_\mathcal{G}, \to, \surd \rangle$, where $\to = \{X \xrightarrow{a} Y \mid X, Y \in N, a \in A_\mathcal{G}\}$. $\mathbf{G}(\cdot)$ is often used as an abbreviation of a sub-chart by its set of nodes.

We define $\mathcal{A}_1 \cup_\mathrm{C} \mathcal{A}_2 =_{df} \mathbf{G}_\mathcal{G}(V_{\mathcal{A}_1} \cup V_{\mathcal{A}_2})$ as the chart of the union of the nodes of $\mathcal{A}_1$ and $\mathcal{A}_2$, provided that $\mathcal{A}_1$ and $\mathcal{A}_2$ are sub-charts of a chart $\mathcal{G}$. We write $\mathcal{G} \subseteq_\mathrm{C} \mathcal{H}$ (resp. $\mathcal{G} \subset_\mathrm{C} \mathcal{H}$) if $\mathcal{G}$ is a sub-chart (resp. proper sub-chart) of $\mathcal{H}$, and write $\mathcal{G} \equiv_\mathrm{C} \mathcal{H}$ if charts $\mathcal{G}$ and $\mathcal{H}$ are identical.

Given a bisimulation function $\theta : V_{\mathcal{G}} \to V_{\mathcal{H}}$ from chart $\mathcal{G}$ to $\mathcal{H}$ and a sub-chart $\mathcal{A}$ of $\mathcal{G}$, we use $\theta(\mathcal{A})$ to represent the sub-chart $\mathbf{G}_{\mathcal{H}}(\theta(V_{\mathcal{A}}))$ of the set $\theta(V_{\mathcal{A}})$ of nodes of $\mathcal{H}$.

We usually use small letters like $x, y, z, ...$ to represent nodes in a domain $V_{\mathcal{G}}$, while using capital letters like $X, Y, Z, ...$ to express nodes in a co-domain $V_{\mathcal{H}}$, of a bisimulation relation $\mathfrak{R} : V_{\mathcal{G}} \to V_{\mathcal{H}}$ from $\mathcal{G}$ to $\mathcal{H}$. And we use letters with the same name and subscript, but different cases and superscripts to express bisimilar nodes in these two graphs. For example, we can write nodes $x', x'', x^{(3)}, X, X', X'', X^{(3)}$ which are all bisimilar to node x.

3.2. Looping-back Charts

Definition 3.1 (Looping-back Charts). *The looping-back chart of an LLEE chart $\hat{\mathcal{G}}/\mathcal{G}$ w.r.t. a node x, denoted by $\mathcal{L}_x$, is a chart defined as $\mathcal{L}_x =_{df} \boldsymbol{G}_{\mathcal{G}}(\{x\} \cup \{y \mid x \curvearrowright^{+} y\})$ and such that $\mathcal{L}_x$ contains at least one loop.*

Call x the "start node" of $\mathcal{L}_x$.

Note that the stipulation that $\mathcal{L}_x$ must contain at least one loop forbids trivial charts of the form: $\langle\{x\}, A, x, \emptyset, \surd\rangle$ as looping-back charts, which contain only one node x without any transitions.

A sub-chart $\mathcal{B}$ of a looping-back chart $\mathcal{A}$ is also called a *looping-back sub-chart* if it is a looping-back chart. Call $V_{\mathcal{L}_x} \setminus \{x\}$ the *body* of looping-back chart $\mathcal{L}_x$.

For instance, in the LLEE chart $\hat{\mathcal{E}}_2/\mathcal{E}_2$ of Fig. 4, charts $\mathbf{G}(\{z', x'\})_{z'}$, $\mathbf{G}(\{z'', y\})_{z''}$, $\mathbf{G}(\{x, z'', k, y\})_x$ and the whole chart $(\mathcal{E}_2)_z$ are looping-back charts. Among them note that charts $\mathbf{G}(\{z', x'\})_{z'}$ and $\mathbf{G}(\{z'', y\})_{z''}$ are also loop sub-charts of $\mathcal{E}_2$.

The following properties about looping-back charts are straightforward from LLEE charts.

Proposition 3.1. *Given a looping-back chart $\mathcal{A}_X$ of an LLEE chart $\hat{\mathcal{G}}/\mathcal{G}$, the following propositions hold:*

(i) For any node $Y \in V_{\mathcal{A}_X} \setminus \{X\}$ and the looping-back chart $\mathcal{B}_Y$ (if there is one), $\mathcal{B}_Y \subset_{\mathrm{C}} \mathcal{A}_X$;

(ii) For each transition $Y \to Z$ such that $Y \in V_{\mathcal{A}_X} \setminus \{X\}$, $Z \in V_{\mathcal{A}_X}$.

(iii) For any node $Y \in V_{\mathcal{A}_X} \setminus \{X\}$ and any path of the form: $Y \to X_1 \to ... \to X_n \to \surd$, $n > 0$ and $X_i \equiv X$ for some $0 < i \leq n$.

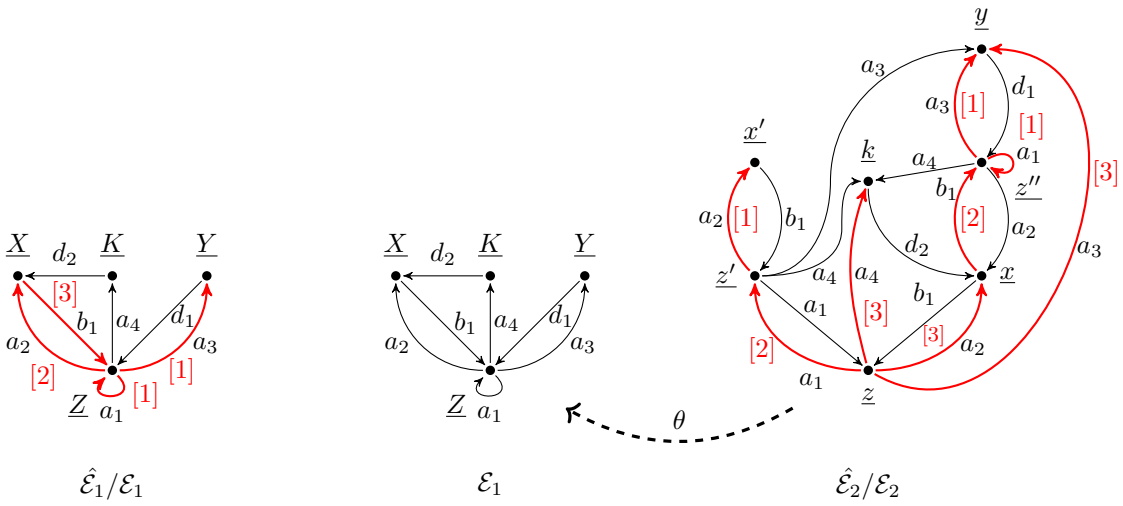

Figure 4: An example of looping-back-chart images

Intuitively, Prop. 3.1 (iii) means that any node $Y \in V_{\mathcal{A}_X} \setminus \{X\}$ does not reach $\surd$ before returning to X.

Proof (Prop. 3.1). (i): By the transitivity of $\curvearrowright^+$ indicated by Prop. 2.1, for any Z such that $Y \curvearrowright^+ Z$, since $X \curvearrowright^+ Y$, $X \curvearrowright^+ Z$. So $V_{\mathcal{B}_Y} \subseteq V_{\mathcal{A}_X}$. By Def. 3.1, $\mathcal{B}_Y \subseteq_{\mathrm{C}} \mathcal{A}_X$. If $\mathcal{B}_Y \equiv_{\mathrm{C}} \mathcal{A}_X$, since $Y \not\equiv X$, Prop. 2.1 is violated. Because we have both $X \curvearrowright^+ Y$ and $Y \curvearrowright^+ X$. Hence $\mathcal{B}_Y \subset_{\mathrm{C}} \mathcal{A}_X$.

(ii): Assume a transition $Y \to Z$ with $Y \in V_{\mathcal{A}_X} \setminus \{X\}$. Let $X \curvearrowright^+ K \curvearrowright Y$ for some node K. Then by the definition of relation $\curvearrowright$ (Sect. 2.3), depending on whether $Y \to Z$ is a loop-entry or a body transition, we have either $K \curvearrowright Y \curvearrowright Z$ or $K \curvearrowright Z$. So $X \curvearrowright^+ K \curvearrowright^+ Z$. By Def. 3.1, $Z \in V_{\mathcal{A}_X}$.

(iii): By Def. 3.1, for any node $Y \in V_{\mathcal{A}_X} \setminus \{X\}$, $X \curvearrowright^+ Y$. We proceed by induction on relation $\curvearrowright^+$.

Base case: $X \curvearrowright Y$. By the definition of $\curvearrowright$ and the LEE property (Sect. 2.3), Y is in the body of a sub-chart that is or will become a loop sub-chart starting from X during the elimination process indicated by $\hat{\mathcal{G}}$. By (L3), it has be the case that $n > 0$ and $X_i \equiv X$ for some $0 < i \leq n$.

Step case: There exists a node K such that $X \curvearrowright^+ K \curvearrowright Y$. From $K \curvearrowright Y$, by inductive hypothesis, $X_i \equiv K$ for some $0 < i \leq n$. From $X \curvearrowright^+ K$, by inductive hypothesis, we have that the path $(K \equiv X_i) \to X_{i+1} \to \ldots \to X_n \to \surd$ satisfies that $n - i > 0$ and $X_j \equiv X$ for some $i < j \leq n$. So $n > 0$ and $X_j \equiv X$ for some $0 < j \leq n$.

□

3.3. Images and Well-structured Pre-images

Definition 3.2 (Looping-back-chart Images). *Given a bisimulation function $\theta : V_{\hat{\mathcal{G}}/\mathcal{G}} \to V_{\mathcal{H}}$ from an LLEE chart $\hat{\mathcal{G}}/\mathcal{G}$ to a chart $\mathcal{H}$, a sub-chart $\mathcal{I}$ of $\mathcal{H}$ is called an "looping-back-chart image" (or simply "image"), if there exists a looping-back chart $\mathcal{L}_x$ in $\mathcal{G}$ such that $\mathcal{I} \equiv_{\mathrm{C}} \theta(\mathcal{L}_x) = \boldsymbol{G}_{\mathcal{H}}(\theta(V_{\mathcal{L}_x}))$. Call $\mathcal{I}$ "the image of $\mathcal{L}_x$" for each such $\mathcal{L}_x$, and call $\mathcal{L}_x$ a "pre-image" of $\mathcal{I}$.*

Denote by $\mathcal{I}_\theta(\mathcal{H})$ the set of all looping-back-chart images on $\mathcal{H}$ (w.r.t. θ).

Call $V_{\theta(\mathcal{L}_x)} \setminus \{\theta(x)\}$ the *body* of image $\theta(\mathcal{L}_x)$ w.r.t. a pre-image $\mathcal{L}_x$.

A looping-back-chart image can have more than one pre-image. Different images can overlap each other in the sense that they share same nodes and transitions.

For example, Fig. 4 shows the bisimulation function $\theta : V_{\hat{\mathcal{E}}_2/\mathcal{E}_2} \to V_{\mathcal{E}_1}$ from an LLEE chart $\hat{\mathcal{E}}_2/\mathcal{E}_2$ to a chart $\mathcal{E}_1$. The nodes with the same letters (e.g. x, x' and X) are bisimilar to each other. In this example, there are 3 images on $\mathcal{E}_1$: $\mathcal{E}_1$, $\mathbf{G}(\{Z, X\})$ and $\mathbf{G}(\{Z, Y\})$. Among them image $\mathcal{E}_1$ has both looping-back charts $(\mathcal{E}_2)_z$ and $\mathbf{G}(\{x, z'', k, y\})_x$ as its pre-images; image $\mathbf{G}(\{Z, X\})$ has the pre-image $\mathbf{G}(\{z', x'\})_{z'}$; image $\mathbf{G}(\{Z, Y\})$ has the pre-image $\mathbf{G}(\{z'', y\})_{z''}$. Images $\mathbf{G}(\{Z, X\})$ and $\mathbf{G}(\{Z, Y\})$ are proper sub-images of image $\mathcal{E}_1$. $\mathbf{G}(\{Z, X\})$ and $\mathbf{G}(\{Z, Y\})$ are not sub-images of each other but do overlap each other.

A bisimulation function is monotonic w.r.t. the sub-chart relation in the following sense.

Proposition 3.2. *A bisimulation function $\theta : V_{\hat{\mathcal{G}}/\mathcal{G}} \to V_{\mathcal{H}}$ satisfies that $\mathcal{A}_1 \subseteq_{\mathrm{C}} \mathcal{A}_2$ implies $\theta(\mathcal{A}_1) \subseteq_{\mathrm{C}} \theta(\mathcal{A}_2)$ for any sub-charts $\mathcal{A}_1$ and $\mathcal{A}_2$ of $\mathcal{G}$.*

Proof. $\mathcal{A}_1 \subseteq_{\mathrm{C}} \mathcal{A}_2$ means $V_{\mathcal{A}_1} \subseteq V_{\mathcal{A}_2}$. So $\theta(V_{\mathcal{A}_1}) \subseteq \theta(V_{\mathcal{A}_2})$. By the definition of $\mathbf{G}(\cdot)$ in Sect. 3.1, $\mathbf{G}(\theta(V_{\mathcal{A}_1})) \subseteq_{\mathrm{C}} \mathbf{G}(\theta(V_{\mathcal{A}_2}))$. □

Below we propose a property for looping-back-charts, called *well-structuredness*. It is the key to prove the image-reflection property (Theorem 4.1) in Section 4.

Definition 3.3 (Well-structuredness). *Given a bisimulation function $\theta : V_{\hat{\mathcal{G}}/\mathcal{G}} \to V_{\mathcal{H}}$ from an LLEE chart $\hat{\mathcal{G}}/\mathcal{G}$ to a chart $\mathcal{H}$, we say a looping-back chart $\mathcal{L}_x$ of $\hat{\mathcal{G}}/\mathcal{G}$ is "well-structured", if it satisfies that for any $\mathcal{L}_y \subset_{\mathrm{C}} \mathcal{L}_x$, $\theta(\mathcal{L}_y) \subset_{\mathrm{C}} \theta(\mathcal{L}_x)$.*

Proposition 3.3 (Well-structuredness Property). *Given a bisimulation function $\theta : V_{\hat{\mathcal{G}}/\mathcal{G}} \to V_{\mathcal{H}}$ from an LLEE chart $\hat{\mathcal{G}}/\mathcal{G}$ to a chart $\mathcal{H}$, for any image $\mathcal{I}$ of $\mathcal{I}_\theta(\mathcal{H})$, there always exists a well-structured looping-back chart $\mathcal{L}_x$ in $\hat{\mathcal{G}}/\mathcal{G}$ as one of its pre-images.*

PROOF. If a pre-image $\mathcal{L}_x$ of $\mathcal{I}$ is not well structured, which means there exists a proper looping-back sub-chart $\mathcal{L}_y$ of $\mathcal{L}_x$ such that $\theta(\mathcal{L}_y) \equiv_{\mathrm{C}} \theta(\mathcal{L}_x) \equiv_{\mathrm{C}} \mathcal{I}$, then $\mathcal{L}_y$ itself is a pre-image of $\mathcal{I}$. Since $\mathcal{G}$ is finite (thus $\subseteq_{\mathrm{C}}$ between looping-back charts is a well-founded relation), we can always obtain a well-structured pre-image for $\mathcal{I}$. □

In Fig. 4, the looping-back chart $(\mathcal{E}_2)_z$, as a pre-image of image $\mathcal{E}_1$, is not well structured, since $\mathbf{G}(\{x, z'', k, y\})_x \subset_{\mathrm{C}} (\mathcal{E}_2)_z$ but $\theta(\mathbf{G}(\{x, z'', k, y\})_x) \equiv_{\mathrm{C}} \mathcal{E}_1 \equiv_{\mathrm{C}} \theta((\mathcal{E}_2)_z)$. However, chart $\mathbf{G}(\{x, z'', k, y\})_x$ is a well-structured pre-image of image $\mathcal{E}_1$ w.r.t. θ.

Prop. 3.3 guides us to consider the relation between a looping-back-chart image and its well-structured pre-images as looping-back charts.

4. Image Reflection on LLEE Charts

In this section, we analyze bisimulation functions from LLEE charts. We discuss the relation between the looping-back-chart images of a bisimulation function and their well-structured pre-images as looping-back charts. The main result is Theorem 4.1, stated as follows.

Theorem 4.1 (Image Reflection). *For every bisimulation function $\theta : V_{\hat{\mathcal{G}}/\mathcal{G}} \to V_{\mathcal{H}}$ from an LLEE chart $\hat{\mathcal{G}}/\mathcal{G}$ to a chart $\mathcal{H}$, $\mathcal{H}$ is also an LEE chart.*

The proof of Theorem 4.1 relies on Prop. 4.1 and Lemma 4.1. Below, we first show the main idea behind our proof. Then we introduce and prove Prop. 4.1 and Lemma 4.1. Lastly, we prove Theorem 4.1.

Definition 4.1 (LEE Sub-Chart). *A sub-chart $\mathcal{A}$ of a chart $\mathcal{G}$ is called an "LEE sub-chart" if it is an LEE chart.*

In a chart, by the definition of loop sub-chart (Sect. 2.3), every loop sub-chart is an LEE sub-chart. Note that a chart that is not LEE can have an LEE sub-chart.

Fix a chart $\mathcal{H}$ of an LLEE chart $\mathcal{G}$ through a bisimulation function $\theta : V_{\hat{\mathcal{G}}/\mathcal{G}} \to V_{\mathcal{H}}$. The main idea of our proof is that through θ, the graphical shape of each image of $\mathcal{I}_\theta(\mathcal{H})$ is constrained by one of its well-structured pre-images so that it is in fact an LEE sub-chart of $\mathcal{H}$. By the sub-image relation between images, we show that there exists a transformation process through which each image of $\mathcal{H}$ will eventually become a loop sub-chart itself and thus can be eliminated further. By eliminating all the images as LEE sub-charts of $\mathcal{H}$, we show that there would be no loops left in $\mathcal{H}$ because the images of $\mathcal{I}_\theta(\mathcal{H})$ cover all the loops of $\mathcal{H}$.

Considering the following example to illustrate our idea. Given the chart $\mathcal{E}_1$ (Fig. 3, Fig. 4), the set of its images is $\mathcal{I}_\theta(\mathcal{E}_1) = \{\mathcal{E}_1, \mathbf{G}(\{Z, X\}), \mathbf{G}(\{Z, Y\})\}$. To prove that $\mathcal{E}_1$ is an LEE chart, we conduct the following two steps of reasoning in an inside-out manner w.r.t. the sub-image relation:

- Firstly, for the minimum image $\mathbf{G}(\{Z, Y\}) \equiv_{\mathrm{C}} \theta(\mathbf{G}(\{z'', y\})_{z''})$, we see that $\mathbf{G}(\{Z, Y\})$ is a loop sub-chart (so is an LEE sub-chart): As constrained by its well-structured pre-image $\mathbf{G}(\{z'', y\})_{z''}$, both loops $Z \xrightarrow{a_1} Z$ and $Z \xrightarrow{a_3} Y \xrightarrow{d_1} Z$ pass through Z and each path in $\mathbf{G}(\{Z, Y\})$ from Z stays in $\mathbf{G}(\{Z, Y\})$ before returning to Z. So $\mathbf{G}(\{Z, Y\})$ can be eliminated in $\mathcal{E}_1$ through the transformation process $\mathit{elim}(\mathbf{G}(\{Z, Y\}), Z)$. Similarly, another minimum image $\mathbf{G}(\{Z, X\})$ is also a loop sub-chart of $\mathcal{E}_1$, and can be eliminated through the process $\mathit{elim}(\mathbf{G}(\{Z, X\}), Z)$.

- Secondly, consider the image $\mathcal{E}_1$ which has images $\mathbf{G}(\{Z, Y\})$ and $\mathbf{G}(\{Z, X\})$ as its two proper sub-images. After $\mathbf{G}(\{Z, Y\})$ and $\mathbf{G}(\{Z, X\})$ are eliminated (in an order for example first $\mathbf{G}(\{Z, Y\})$ and then $\mathbf{G}(\{Z, X\})$), the remnant chart: chart $\mathcal{E}_1''$ of Fig. 3, becomes a loop sub-chart and thus can be eliminated. This is because its only loop $X \xrightarrow{b_1} Z \xrightarrow{a_4} K \xrightarrow{d_2} X$, constrained by the well-structured pre-image $\mathbf{G}(\{x, z'', k, y\})_x$, passes through X and each path from X stays in $\mathcal{E}_1''$ before returning to X. Therefore, the image $\mathcal{E}_1$ is an LEE sub-chart (of itself).

Since $\mathcal{I}_\theta(\mathcal{E}_1)$ covers all the loops of $\mathcal{E}_1$, after eliminating $\mathcal{E}_1''$, we obtain the chart $\mathcal{E}_1^{(3)}$ of Fig. 3 where there is no loops. Therefore, according to Definition 2.1 the whole chart $\mathcal{E}_1$ is an LEE chart.

Following the main idea, below we give our formal proofs.

Prop. 4.1 indicates a correspondence between the loops of an LLEE chart $\mathcal{G}$ and the loops of its a chart $\mathcal{H}$ given a bisimulation function $\theta : V_{\hat{\mathcal{G}}/\mathcal{G}} \to V_{\mathcal{H}}$.

Proposition 4.1. *Given a bisimulation function $\theta : V_{\hat{\mathcal{G}}/\mathcal{G}} \to V_{\mathcal{H}}$ from an LLEE chart $\mathcal{G}$ to a chart $\mathcal{H}$ and a loop $\mathcal{A}$ of $\mathcal{H}$, from a node x of $\mathcal{G}$ such that $\theta(x)$ is in $\mathcal{A}$, there exists a path $\mathcal{S}$ reaching a loop $\mathcal{D}$ in $\mathcal{G}$ satisfying that $\mathcal{A} \equiv_{\mathrm{C}} \theta(\mathcal{S} \cup_{\mathrm{C}} \mathcal{D}) \equiv_{\mathrm{C}} \theta(\mathcal{D})$.*

Proof. Assume loop $\mathcal{A}$ is of the form: $X \xrightarrow{a_1} X_1 \xrightarrow{a_2} ... \xrightarrow{a_n} X_n \xrightarrow{a_{n+1}} X$ ($n \geq 0$), with $X \equiv \theta(x)$. From x, by the bisimulation relation θ between $\mathcal{G}$ and $\mathcal{H}$, there is a path of the form: $x \xrightarrow{a_1} x_1 \xrightarrow{a_2} ... \xrightarrow{a_n} x_n \xrightarrow{a_{n+1}} x^{(1)} \xrightarrow{a_1} ... \xrightarrow{a_{n+1}} x^{(k)} \xrightarrow{a_1} x_1^{(k)} \xrightarrow{a_2} ... \xrightarrow{a_n} x_n^{(k)} \xrightarrow{a_{n+1}} x^{(k+1)} \xrightarrow{a_1} ...$, where $X_i \equiv \theta(x_i^{(k)})$ for all $k \geq 1$ and $1 \leq i \leq n$, and $X \equiv \theta(x^{(k)})$ for all $k \geq 1$. Since $\mathcal{G}$ is finite, we can always find the node $x^{(m)}$ with the smallest $m > 0$ such that $x^{(m)} \equiv x^{(j)}$ for some $1 \leq j < m$. So from x there is a path $\mathcal{S}$: $x \xrightarrow{a_1} x_1 \xrightarrow{a_2} ... \xrightarrow{a_n} x_n \xrightarrow{a_{n+1}} x^{(1)} \xrightarrow{a_1} ... \xrightarrow{a_{n+1}} x^{(j)}$ reaching the loop $\mathcal{D}$: $x^{(j)} \xrightarrow{a_1} ... \xrightarrow{a_{n+1}} x^{(m)} \equiv x^{(j)}$. And since $V_{\mathcal{A}} = \theta(V_{\mathcal{S}} \cup V_{\mathcal{D}}) = \theta(V_{\mathcal{D}})$, clearly $\mathcal{A} \equiv_{\mathrm{C}} \theta(\mathcal{S} \cup_{\mathrm{C}} \mathcal{D}) \equiv_{\mathrm{C}} \theta(\mathcal{D})$. □

Lemma 4.1. *Given the set of looping-back-chart images $\mathcal{I}_\theta(\mathcal{H})$ w.r.t. a bisimulation function $\theta : V_{\hat{\mathcal{G}}/\mathcal{G}} \to V_{\mathcal{H}}$ from an LLEE chart $\hat{\mathcal{G}}/\mathcal{G}$ to a chart $\mathcal{H}$, the following conditions hold:*

(1) For each loop $\mathcal{C}$ of $\mathcal{H}$, there exists an image $\mathcal{I}$ of $\mathcal{I}_\theta(\mathcal{H})$ such that $\mathcal{C} \subseteq_{\mathrm{C}} \mathcal{I}$;

(2) For each image $\theta(\mathcal{L}_x)$ of $\mathcal{I}_\theta(\mathcal{H})$ with a well-structured pre-image $\mathcal{L}_x$, in $\theta(\mathcal{L}_x)$ each loop either passes through node $\theta(x)$ or is in a proper sub-image of $\theta(\mathcal{L}_x)$.

(3) For each image $\theta(\mathcal{L}_x)$ of $\mathcal{I}_\theta(\mathcal{H})$ with a pre-image $\mathcal{L}_x$ and a node $U \in V_{\theta(\mathcal{L}_x)} \setminus \{\theta(x)\}$, $U \not\to \surd$; And for any transition $U \to W$, $W \in V_{\theta(\mathcal{L}_x)}$.

Lemma 4.1 (1) states the coverage of the loops of $\mathcal{H}$ by $\mathcal{I}_\theta(\mathcal{H})$. Lemma 4.1 (2) and (3) describe the constrained relation between the shape of an image and that of any of its well-structured pre-images. Lemma 4.1 (3) intuitively means that each path starting from node $\theta(x)$ stays in $\theta(\mathcal{L}_x)$ before returning to $\theta(x)$.

Proof (Lemma 4.1). (Proof of (1)). For each loop $\mathcal{C}$ of $\mathcal{H}$, by Prop. 4.1 there is a loop $\mathcal{D}$ in $\mathcal{G}$ such that $\mathcal{C} \equiv_{\mathrm{C}} \theta(\mathcal{D})$. Since $\mathcal{G}$ is an LLEE chart, $\mathcal{D}$ begins with a loop-entry transition in the form: $x \to_{[m]} \cdot \to^* x$, $m > 0$. By Def. 3.1, $\mathcal{D}$ is a sub-chart of the looping-back chart $\mathcal{L}_x$. By Prop. 3.2, $\mathcal{C} \equiv_{\mathrm{C}} \theta(\mathcal{D}) \subseteq_{\mathrm{C}} \theta(\mathcal{L}_x)$.

(Proof of (2)). Let $\mathcal{C}$ be a loop of $\theta(\mathcal{L}_x)$. By Prop. 4.1, starting from a node y in $\mathcal{L}_x$ such that $\theta(y)$ is in $\mathcal{C}$, there is a path $\mathcal{S}$ reaching a loop $\mathcal{D}$ such that $\mathcal{C} \equiv_{\mathrm{C}} \theta(\mathcal{S} \cup_{\mathrm{C}} \mathcal{D})$. If $\mathcal{C}$ does not pass $\theta(x)$, then chart $\mathcal{S} \cup_{\mathrm{C}} \mathcal{D}$ does not pass x, and $y \in V_{\mathcal{L}_x} \setminus \{x\}$. By Prop. 3.1 (ii) and (iii), since $y \in V_{\mathcal{L}_x} \setminus \{x\}$, loop $\mathcal{D}$ stays in $\mathcal{L}_x$ and is thus a sub-chart of $\mathcal{L}_x$. By that $\mathcal{G}$ is an LLEE chart, loop $\mathcal{D}$ starts from a loop-entry transition $k \rightarrow_{[m]} \cdot$ with $m > 0$. By Def. 3.1, $\mathcal{D}$ must be a sub-chart of the looping-back chart $\mathcal{L}_k$. So $\mathcal{D} \subseteq_{\mathrm{C}} \mathcal{L}_k \subseteq_{\mathrm{C}} \mathcal{L}_x$. Since $k \not\equiv x$, by Prop. 3.1 (i), $\mathcal{L}_k \subset_{\mathrm{C}} \mathcal{L}_x$. According to Prop. 3.3 and that $\mathcal{L}_x$ is well structured, $\theta(\mathcal{L}_k) \subset_{\mathrm{C}} \theta(\mathcal{L}_x)$. So $\mathcal{C} \equiv_{\mathrm{C}} \theta(\mathcal{S} \cup_{\mathrm{C}} \mathcal{D}) \equiv_{\mathrm{C}} \theta(\mathcal{D})$ (by Prop. 4.1) $\subseteq_{\mathrm{C}} \theta(\mathcal{L}_k)$ (by Prop. 3.2) $\subset_{\mathrm{C}} \theta(\mathcal{L}_x)$. In other words, $\mathcal{C}$ is in the proper sub-image $\theta(\mathcal{L}_k)$ of $\theta(\mathcal{L}_x)$.

(Proof of (3)). Let u be the node in $\mathcal{L}_x$ such that $\theta(u) \equiv U$. Since $U \not\equiv \theta(x)$, $u \not\equiv x$. So $u \in V_{\mathcal{L}_x} \setminus \{x\}$. By Prop. 3.1 (iii), $u \not\rightarrow \surd$. By that $u \sim U$, we have $U \not\rightarrow \surd$. For any transition $U \rightarrow W$, by $u \sim U$, there is a transition $u \rightarrow w$, moreover, $\theta(w) = W$. By Prop. 3.1 (ii), $w \in V_{\mathcal{L}_x}$. Therefore, $W \in V_{\theta(\mathcal{L}_x)}$.

□

Now we give the proof of Theorem 4.1.

Proof (Theorem 4.1). By the definition of LEE chart (Definition 2.1), we show that there is an elimination process on $\mathcal{H}$ by which $\mathcal{H}$ can be transformed into a chart $\mathcal{H}'$ without infinite paths. It is sufficient to prove that each image of $\mathcal{I}_\theta(\mathcal{H})$ is an LEE sub-chart of $\mathcal{H}$ by induction on the sub-image relations between images. The elimination process on $\mathcal{H}$ can then be naturally induced by this proof process.

Base case: Consider a minimum image $\mathcal{I}$ of $\mathcal{I}_\theta(\mathcal{H})$ (in which there are no proper sub-images). By Prop. 3.3, we can find a well-structured pre-image $\mathcal{L}_r$ such that $\mathcal{I} \equiv_{\mathrm{C}} \theta(\mathcal{L}_r)$. We show that if $\theta(\mathcal{L}_r)$ has at least one loop, it is a loop sub-chart of $\mathcal{H}$ (so is also an LEE sub-chart of $\mathcal{H}$). (L1) is obvious since $\theta(\mathcal{L}_r)$ has a loop. By Lemma 4.1 (2) and that $\theta(\mathcal{L}_r)$ is a minimum image, each loop in $\theta(\mathcal{L}_r)$ must pass through $\theta(r)$. By Lemma 4.1 (3), each path in $\theta(\mathcal{L}_r)$ starting from $\theta(r)$ cannot leave $\theta(\mathcal{L}_r)$ before returning to $\theta(r)$. So, every infinite path in $\theta(\mathcal{L}_r)$ starting from $\theta(r)$ passes through $\theta(r)$. That is, (L2) holds. (L3) is direct from Prop. 3.1 (iii) satisfied by $\mathcal{L}_r$ and the bisimulation relation θ itself.

Step case: Consider an arbitrary image $\mathcal{I}$ of $\mathcal{I}_\theta(\mathcal{H})$. By Prop. 3.3, let $\mathcal{I} \equiv_{\mathrm{C}} \theta(\mathcal{L}_s)$ for some well-structured pre-image $\mathcal{L}_s$. We show that $\theta(\mathcal{L}_s)$ is an

LEE sub-chart. By the inductive hypothesis, every proper sub-image $\mathcal{I}'$ of $\theta(\mathcal{L}_s)$ is an LEE sub-chart. So there exists an elimination process *elim* to eliminate each $\mathcal{I}'$. After eliminating all these sub-images, assume $\theta(\mathcal{L}_s)$ is transformed into a chart $(\theta(\mathcal{L}_s))'$. By Lemma 4.1 (2), all loops remaining in $(\theta(\mathcal{L}_s))'$ must pass through $\theta(s)$. If $(\theta(\mathcal{L}_s))'$ still has at least one loop, then similarly as in the base case above, we show that $(\theta(\mathcal{L}_s))'$ is a loop sub-chart:

(L1) is obvious because $(\theta(\mathcal{L}_s))'$ has a loop. By Lemma 4.1 (3), each path in $\theta(\mathcal{L}_s)$ cannot leave $\theta(\mathcal{L}_s)$ before returning to $\theta(s)$. So after the eliminations, each path in the remnant $(\theta(\mathcal{L}_s))'$ still cannot leave $(\theta(\mathcal{L}_s))'$ before returning to $\theta(s)$. By this and the fact that all loops in $(\theta(\mathcal{L}_s))'$ pass through $\theta(s)$ as obtained above, it is easy to see that (L2) holds. (L3) is direct from Prop. 3.1 (iii) satisfied by $\mathcal{L}_s$ and the bisimulation relation θ itself.

By that $(\theta(\mathcal{L}_s))'$ is a loop sub-chart, it then can be eliminated by an elimination process *elim*. So the whole image $\theta(\mathcal{L}_s)$ can be eliminated and is thus an LEE sub-chart.

Conclusion: Since each image is an LEE sub-chart of $\mathcal{H}$, as indicated in the inductive reasoning above, we obtain a process that eliminates all images in $\mathcal{I}_\theta(\mathcal{H})$ in an inside-out manner from the minimum sub-images according to the sub-image relations. Let $\mathcal{H}'$ be the chart $\mathcal{H}$ after the elimination process. By Lemma 4.1 (1), there exist no loops left in $\mathcal{H}'$. So $\mathcal{H}'$ is a chart without infinite paths. □

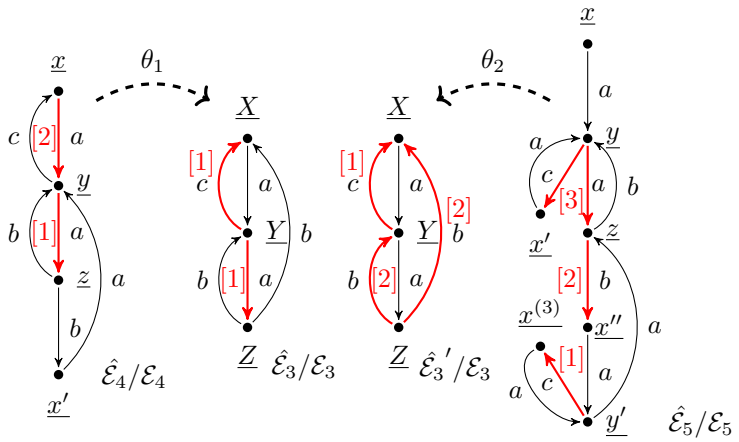

Figure 5: An example of different image reflections by a same chart

In the above proof, for each image of $\mathcal{I}_\theta(\mathcal{H})$, only one well-structured

pre-image is needed to guarantee that the image is an LEE sub-chart. The hierarchy structure of images indicated by the sub-image relations between them is the key to determine the LEE elimination process on $\mathcal{H}$.

Now let us go back to the example in Fig. 1. Chart $\mathcal{H}$ has its well-structured pre-image $\mathcal{G}_{x'}$ and obviously reflects its LEE structure through θ. In the example of Fig. 4, consider the chart $\mathcal{E}_1$. By the elimination process of its images we have discussed earlier at the beginning of this section, we obtained its LEE witness $\hat{\mathcal{E}}_1$ as also shown on the left side in Fig. 4. In this example, image $\mathcal{E}_1$ reflects the LEE structure of its well-structured pre-image $\mathbf{G}(\{x, z'', k, y\})_x$; images $\mathbf{G}(\{Z, X\})$ and $\mathbf{G}(\{Z, Y\})$ reflect the LEE structures of their well-structured pre-images $\mathbf{G}(\{z', x'\})_{z'}$ and $\mathbf{G}(\{z'', y\})_{z''}$ respectively.

In fact, from Theorem 4.1, a graph $\mathcal{H}$ is able to reflect the LEE structure of some parts of *any* LLEE chart from which there is a bisimulation function to $\mathcal{H}$. This result is related to the fact that an LLEE chart can have multiple LEE/LLEE witnesses.

Consider another example firstly proposed in [18] as shown in Fig. 5. Given two bisimulation functions $\theta_1 : V_{\hat{\mathcal{E}}_4/\mathcal{E}_4} \to V_{\mathcal{E}_3}$ and $\theta_2 : V_{\hat{\mathcal{E}}_5/\mathcal{E}_5} \to V_{\mathcal{E}_3}$ from LLEE charts $\mathcal{E}_4$ and $\mathcal{E}_5$ to their bisimulation collapse $\mathcal{E}_3$, $\mathcal{E}_3$ reflects some parts of the LEE structures of both $\mathcal{E}_4$ and $\mathcal{E}_5$: $\mathcal{E}_3$ with the left witness $\hat{\mathcal{E}}_3$ reflects the LEE structure of the looping-back chart $\mathbf{G}(\{y, z, x'\})_y$ of $\mathcal{E}_4$, whereas $\mathcal{E}_3$ with the right witness $\hat{\mathcal{E}}_3{}'$ reflects the LEE structure of the looping-back chart $\mathbf{G}(\{z, x'', y', x^{(3)}\})_z$ of $\mathcal{E}_5$. In our method, the different witnesses of $\hat{\mathcal{E}}_3$ and $\hat{\mathcal{E}}_3{}'$ come from different structures of images on $\mathcal{E}_3$ through two different bisimulation functions θ_1 and θ_2.

5. A Proof for the Completeness of BBP

As a direct application of the result in Sect. 4, this section gives a different proof for the completeness of **BBP**, which, compared to the one given in [18], is more direct, in the sense that we do not rely on a series of transformation steps but directly rely on the "image reflection" property of images using our defined structures in Sect. 3.

In the following, we introduce the minimization strategy mentioned in Sect. 1 as the "blueprint" of proving the completeness of **BBP** that was firstly proposed in [18]. Then we explain that how our result can provide an alternative proof procedure to modify the minimization strategy.

5.1. Minimization Strategy

The minimization strategy is based on Prop. 2.2 - 2.5 introduced in Sect. 2, abbreviated as (P1) - (P4) respectively in the following discussion. As illustrated in Fig. 6, given two expressions $e_1, e_2 \in \mathit{REXP}_{\mathit{1\text{-}free}}$, to prove $e_1 = e_2$, the main idea can be explained as follows in two stages:

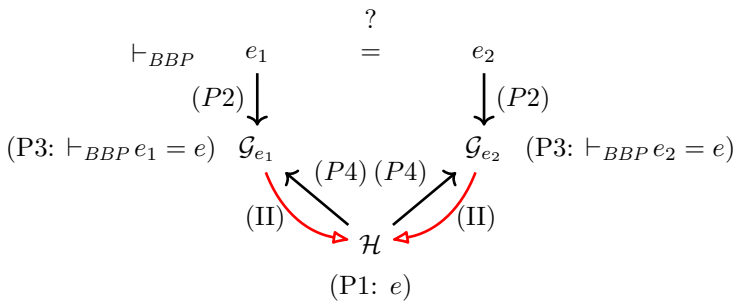

Figure 6: Minimization strategy

(I) To prove $e_1 = e_2$, by (P2), this means there are two LLEE charts, namely $\mathcal{G}_{e_1}$ and $\mathcal{G}_{e_2}$, such that $\mathcal{G}_{e_1} \sim \mathcal{G}_{e_2}$, and e_1 and e_2 are the primary solutions of $\mathcal{G}_{e_1}$ and $\mathcal{G}_{e_2}$ respectively. Since $\mathcal{G}_{e_1} \sim \mathcal{G}_{e_2}$, $\mathcal{G}_{e_1}$ and $\mathcal{G}_{e_2}$ have the common bisimulation collapse $\mathcal{H}$ through two bisimulation functions $\theta_1 : \mathcal{G}_{e_1} \to \mathcal{H}$ and $\theta_2 : \mathcal{G}_{e_2} \to \mathcal{H}$ respectively. Assume that $\mathcal{H}$ is an LLEE chart. Then by (P1) $\mathcal{H}$ has a primary solution, let us say e. By (P4), e is also a primary solution of $\mathcal{G}_{e_1}$ and $\mathcal{G}_{e_2}$. So by (P3), $e_1 = e$ and $e_2 = e$ can both be derived. Therefore, $e_1 = e_2$ can be derived simply by the symmetry and transitivity of equivalence $=$.

(II) In (I), to show that $\mathcal{H}$ is an LLEE chart, a graph transformation was carried out from either $\mathcal{G}_{e_1}$ or $\mathcal{G}_{e_2}$ into $\mathcal{H}$ step by step. During each step, the transformation merges two selected bisimular nodes in $\mathcal{G}_{e_1}$ (or $\mathcal{G}_{e_2}$), while preserving that the graph after the merge is still an LLEE chart.

We modify the minimization strategy by proposing a different stage (II') from (II):

(II') In (I), to show that $\mathcal{H}$ is an LLEE chart, we firstly prove that $\mathcal{H}$ is an LEE chart according to the bisimulation function θ_1 (or θ_2), making

use of the image reflection on $\mathcal{G}_{e_1}$ (or $\mathcal{G}_{e_2}$) through θ_1 (or θ_2); then we show that each LEE chart is also an LLEE chart by selecting a suitable LLEE witness.

5.2. Our Proof Procedure in Minimization Strategy

Provided with Theorem 4.1, to prove stage (II'), it is sufficient to show that an LEE chart is also an LLEE one. We state it as the following proposition.

Proposition 5.1. *Each LEE chart is also an LLEE chart.*

Prop. 5.1 has already been mentioned in [21] (see Remark 4.11 there), but it has not been proved there since it was not used. Here, as it plays a role in proving Theorem 4.1, we will give a proof. The key to prove Prop. 5.1, as also pointed out in [21], is to show that an LEE witness can be transformed into an LLEE witness in an LEE chart.

As an example, consider the LEE chart $\hat{\mathcal{E}_1}'/\mathcal{E}_1$ with witness $\hat{\mathcal{E}_1}'$ in Fig. 3, which is an LLEE witness. From the LEE witness $\hat{\mathcal{E}_1}$, we first turn the loop-entry transition $X \xrightarrow{b_1}_{[3]} Z$ from the body of the $\langle Z, \{Z \xrightarrow{a_2}_{[2]} X\}\rangle$-generated chart that violates the LLEE property into a body transition $X \xrightarrow{b_1}_{[0]} Z$. This causes loop $X \xrightarrow{b_1} Z \xrightarrow{a_4} K \xrightarrow{d_2} X$ become a loop without loop-entry transitions. Then we turn transition $Z \xrightarrow{a_4} K$ into a loop-entry transition $Z \xrightarrow{a_4}_{[3]} K$. By this loop-entry-transition-switching process we transform $\hat{\mathcal{E}_1}$ into an LLEE witness $\hat{\mathcal{E}_1}'$.

The formal proof is given below.

Proof (Prop. 5.1). Assume $\hat{\mathcal{G}}/\mathcal{G}$ is an LEE chart with $\hat{\mathcal{G}}$ an LEE witness. We obtain an LLEE witness $\hat{\mathcal{G}}'$ from $\hat{\mathcal{G}}$ by induction on the LEE elimination process according to $\hat{\mathcal{G}}$. In the following proof, without loss of generality, assume that every loop-entry transition has a unique order number, starting from the smallest number 1. We denote by $\mathcal{D}(X \to Y)$ a loop starting from transition $X \to Y$. We simply use $\langle X, \mathbf{E}\rangle$ to denote a $\langle X, \mathbf{E}\rangle$-generated chart of $\mathcal{G}$.

Base case: Consider the $\langle R_1, \{R_1 \to_{[1]} \cdot\}\rangle$-generated chart with $R_1 \to_{[1]} \cdot$ the smallest loop-entry transition. By LEE property, $\langle R_1, \{R_1 \to_{[1]} \cdot\}\rangle$ is a loop sub-chart. For each loop-entry transition $X \to_{[k]} \cdot$ from the body of chart $\langle R_1, \{R_1 \to_{[1]} \cdot\}\rangle$ with $k > 1$ (thus violating the LLEE property), we

turn $X \rightarrow_{[k]} \cdot$ into a body transition: $X \rightarrow_{[0]} \cdot$. For each loop $\mathcal{D}(X \rightarrow_{[0]} \cdot)$ that becomes a loop without loop-entry transitions because of the transforming from $X \rightarrow_{[k]} \cdot$ to $X \rightarrow_{[0]} \cdot$, since $X \rightarrow_{[0]} \cdot$ is from the body of $\langle R_1, \{R_1 \rightarrow_{[1]} \cdot\}\rangle$, by that $\langle R_1, \{R_1 \rightarrow_{[1]} \cdot\}\rangle$ is a loop sub-chart and (L2), $\mathcal{D}(X \rightarrow_{[0]} \cdot)$ must pass through R_1. Let $\mathcal{D}(X \rightarrow_{[0]} \cdot)$ be the form of: $X \rightarrow_{[0]} \cdot \rightarrow^* R_1 \rightarrow Z_1 \rightarrow^* X$. We then turn transition $R_1 \rightarrow Z_1$ into a loop-entry transition $R_1 \rightarrow_{[k]} Z_1$. It is not hard to see that after the loop-entry-transition-switching process described above for each $X \rightarrow_{[k]} \cdot$ with $k > 1$, all transitions $\cdot \rightarrow_{[k]} \cdot$ from the body of $\langle R_1, \{R_1 \rightarrow_{[1]} \cdot\}\rangle$ are body transitions with $k = 0 < 1$.

Step case: Consider a $\langle R_n, \{R_n \rightarrow_{[n]} \cdot\}\rangle$-generated chart starting from a loop-entry transition $R_n \rightarrow_{[n]} \cdot$ with $n > 1$. By induction hypothesis, in any $\langle R_l, \{R_l \rightarrow_{[l]} \cdot\}\rangle$-generated chart with $l < n$, any transition $\cdot \rightarrow_{[u]} \cdot$ from the body of $\langle R_l, \{R_l \rightarrow_{[l]} \cdot\}\rangle$ satisfies that $u < l$. For each loop-entry transition $X \rightarrow_{[k]} \cdot$ from the body of $\langle R_n, \{R_n \rightarrow_{[n]} \cdot\}\rangle$ satisfying $k > n$, we turn $X \rightarrow_{[k]} \cdot$ into a body transition: $X \rightarrow_{[0]} \cdot$. For each loop $\mathcal{D}(X \rightarrow_{[0]} \cdot)$ that becomes a loop without loop-entry transitions because of the transforming from $X \rightarrow_{[k]} \cdot$ to $X \rightarrow_{[0]} \cdot$, we see that loop $\mathcal{D}(X \rightarrow_{[0]} \cdot)$ must pass through R_n. Because by the LEE property, after eliminating all loop-entry transitions with an order number $l < n$, chart $\langle R_n, \{R_n \rightarrow_{[n]} \cdot\}\rangle$ becomes a loop sub-chart, denoted by $\langle R_n, \{R_n \rightarrow_{[n]} \cdot\}\rangle'$. Since $X \rightarrow_{[k]} \cdot$ is not from the body of any chart $\langle R_l, \{R_l \rightarrow_{[l]} \cdot\}\rangle$ with $l < n$ (by inductive hypothesis above), loop $\mathcal{D}(X \rightarrow_{[0]} \cdot)$ is still not eliminated in $\langle R_n, \{R_n \rightarrow_{[n]} \cdot\}\rangle'$. So by (L2), $\mathcal{D}(X \rightarrow_{[0]} \cdot)$ passes through R_n. Let $\mathcal{D}(X \rightarrow_{[0]} \cdot)$ be of the form: $X \rightarrow_{[0]} \cdot \rightarrow^* R_n \rightarrow Z_n \rightarrow^* X$, then we turn the body transition $R_n \rightarrow Z_n$ into a loop-entry transition $R_n \rightarrow_{[k]} Z_n$. After the loop-entry-transition-switching process described above for each $X \rightarrow_{[k]} \cdot$, we see that all transitions $\cdot \rightarrow_{[k]} \cdot$ from the body of $\langle R_n, \{R_n \rightarrow_{[n]} \cdot\}\rangle$ satisfy that $k < n$.

By the LLEE property, the witness $\hat{\mathcal{G}}'$ obtained after the transformations above is the LLEE witness as required.

With this additional operation stated as Prop. 5.1, based on Theorem 4.1, we obtain the same result as in stage (II) stated as follows.

Corollary 5.1 ([18]). *For every bisimulation function $\theta : V_{\hat{\mathcal{G}}/\mathcal{G}} \rightarrow V_{\mathcal{H}}$ from an LLEE chart $\hat{\mathcal{G}}/\mathcal{G}$ to a bisimulation collapse $\mathcal{H}$, $\mathcal{H}$ is an LLEE chart.*

6. Conclusion and Future Work

In this paper, we study a property called "image reflection" on a type of characterization graphs — LLEE charts. By introducing well-structured looping-back charts on LLEE charts, we show that the looping-back images of a bisimulation function $f : V_{\mathcal{G}} \to V_{\mathcal{H}}$ from an LLEE chart $\mathcal{G}$ to a chart $\mathcal{H}$ actually impose an LEE structure related to $\mathcal{G}$. This leads us to a different approach for proving the completeness of **BBP**, which is more direct in the sense that we skip any graph transformations that rely on careful selections of bisimilar-node pairs.

Previous work has extended LEE/LLEE charts to characterize regular expressions modulo bisimulation equivalence by introducing a silent (without doing any action) *1-transition* $X \xrightarrow{1} Y$ (cf. [22, 19]). The LEE/LLEE charts with additional 1-transitions, called *LEE/LLEE 1-charts*, are defined the same as shown in Sect. 2.3. With 1-transitions, a similar correspondence between regular expressions and provable solutions of LLEE 1-charts can be built, analog to Prop. 2.3 and 2.2 for LLEE charts. The connection of LLEE 1-charts to the completeness of **Mil** were fully explored in recent work [19].

The extension of looping-back charts and well-structuredness to the case of LEE/LLEE 1-charts is natural. Because as indicated in our proofs (, which was also observed in [23]), LEE/LLEE properties are irrelevant to the transition types. The significance of the idea in our proof for the completeness of **BBP** is that it can be tailored to simplify the graph transformations proposed in [19], which are technically complex and difficult. To do this, we need to investigate how to find a proper 1-chart $\mathcal{H}$ and a bisimulation function $\theta : V_{\mathcal{G}} \to V_{\mathcal{H}}$ from an LLEE 1-chart $\mathcal{G}$ to $\mathcal{H}$ so that $\mathcal{H}$ satisfies the "crystallization conditions" in [19]. Our future work will focus on this aspect.

Another possible usage of our work is for the completeness proofs of process algebras that share similar structures with charts as their process semantics. One example is skip-free star algebra [20], a process algebra more general than 1-free regular expression. Its process semantics is described by a more abstract model "M-systems", which is extended from charts by generalizing the notion of transitions. The special class of M-systems needed for proving the completeness of skip-free star algebra, called "well-layered M-systems", actually share the same shape with LLEE charts (, but provided with more structures on transitions). Combined with the irrelevance of LEE/LLEE properties to the transition types, this evidence makes us believe that our method can also be tailored for proving the completeness of

skip-free star algebra.

Acknowledgment. We thank the reviewers for their careful reviewing of this work. Especially, we thank the 2nd reviewer, for his/her inspiring ideas to help improve this work.

This work is partially supported by the New Faculty Start-up Foundation of NUAA (No. 90YAT24003), the Youth Project of National Science Foundation of China (No. 62102329), and the Project of National Science Foundation of China (No. 62272397).

References

[1] S. C. Kleene, Representation of Events in Nerve Nets and Finite Automata, Princeton University Press, Princeton, 1956, pp. 3–42 [cited 2023-10-08]. doi:doi:10.1515/9781400882618-002.
URL https://doi.org/10.1515/9781400882618-002

[2] I. M. Copi, C. C. Elgot, J. B. Wright, Realization of Events by Logical Nets, J. ACM 5 (2) (1958) 181–196. doi:10.1145/320924.320931.
URL https://doi.org/10.1145/320924.320931

[3] D. Harel, D. Kozen, J. Tiuryn, Dynamic Logic, MIT Press, 2000.

[4] D. Kozen, Kleene Algebra with Tests, ACM Trans. Program. Lang. Syst. 19 (3) (1997) 427–443. doi:10.1145/256167.256195.
URL https://doi.org/10.1145/256167.256195

[5] S. Smolka, N. Foster, J. Hsu, T. Kappé, D. Kozen, A. Silva, Guarded Kleene Algebra with Tests: Verification of Uninterpreted Programs in Nearly Linear Time, Proc. ACM Program. Lang. 4 (POPL) (2019). doi:10.1145/3371129.
URL https://doi.org/10.1145/3371129

[6] B. Beckert, V. Klebanov, B. Weiß, Dynamic Logic for Java, Springer International Publishing, Cham, 2016, pp. 49–106. doi:10.1007/978-3-319-49812-6_3.
URL https://doi.org/10.1007/978-3-319-49812-6_3

[7] A. Platzer, Logical Foundations of Cyber-Physical Systems, 1st Edition, Springer Publishing Company, Incorporated, 2018.

[8] Y. Zhang, H. Wu, Y. Chen, F. Mallet, A Clock-based Dynamic Logic for the Verification of CCSL Specifications in Synchronous Systems, Science of Computer Programming 203 (2021) 102591. doi:https://doi.org/10.1016/j.scico.2020.102591.
URL https://www.sciencedirect.com/science/article/pii/S0167642320301994

[9] Y. Zhang, F. Mallet, Z. Liu, A dynamic logic for verification of synchronous models based on theorem proving, Frontiers of Computer Science 16 (2022) 164407.

[10] C. A. R. T. Hoare, B. Möller, G. Struth, I. Wehrman, Concurrent kleene algebra, in: M. Bravetti, G. Zavattaro (Eds.), CONCUR 2009 - Concurrency Theory, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 399–414.

[11] N. Foster, D. Kozen, K. Mamouras, M. Reitblatt, A. Silva, Probabilistic netkat, in: P. Thiemann (Ed.), Programming Languages and Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2016, pp. 282–309.

[12] T. Schmid, T. Kappé, D. Kozen, A. Silva, Guarded Kleene Algebra with Tests: Coequations, Coinduction, and Completeness, in: N. Bansal, E. Merelli, J. Worrell (Eds.), 48th International Colloquium on Automata, Languages, and Programming (ICALP 2021), Vol. 198 of Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2021, pp. 142:1–142:14. doi:10.4230/LIPIcs.ICALP.2021.142.
URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ICALP.2021.142

[13] A. Salomaa, Two Complete Axiom Systems for the Algebra of Regular Events, J. ACM 13 (1) (1966) 158–169. doi:10.1145/321312.321326.
URL https://doi.org/10.1145/321312.321326

[14] R. Milner, A Complete Inference System for a Class of Regular Behaviours, Journal of Computer and System Sciences 28 (3) (1984) 439–466. doi:https://doi.org/10.1016/0022-0000(84)90023-0.
URL https://www.sciencedirect.com/science/article/pii/0022000084900230

[15] J. A. Bergstra, I. Bethke, A. Ponse, Process Algebra with Iteration and Nesting, The Computer Journal 37 (4) (1994) 243–258. arXiv:https://academic.oup.com/comjnl/article-pdf/37/4/243/1067027/370243.pdf, doi:10.1093/comjnl/37.4.243.
URL https://doi.org/10.1093/comjnl/37.4.243

[16] W. Fokkink, On the Completeness of the Equations for the Kleene Star in Bisimulation, in: M. Wirsing, M. Nivat (Eds.), Algebraic Methodology and Software Technology, Springer Berlin Heidelberg, Berlin, Heidelberg, 1996, pp. 180–194.

[17] W. J. Fokkink, An Axiomatization for the Terminal Cycle, Nederlands Archief Voor Kerkgeschiedenis (1996).
URL https://api.semanticscholar.org/CorpusID:6310338

[18] C. Grabmayer, W. Fokkink, A complete proof system for 1-free regular expressions modulo bisimilarity, in: Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 465–478. doi:10.1145/3373718.3394744.
URL https://doi.org/10.1145/3373718.3394744

[19] C. A. Grabmayer, Milner's Proof System for Regular Expressions Modulo Bisimilarity is Complete: Crystallization: Near-Collapsing Process Graph Interpretations of Regular Expressions, in: Proceedings of the 37th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '22, Association for Computing Machinery, New York, NY, USA, 2022. doi:10.1145/3531130.3532430.
URL https://doi.org/10.1145/3531130.3532430

[20] T. Kappé, T. Schmid, A general completeness theorem for skip-free star algebras, in: P. A. Abdulla, D. Kesner (Eds.), Foundations of Software Science and Computation Structures, Springer Nature Switzerland, Cham, 2025, pp. 265–286.

[21] C. Grabmayer, A Coinductive Reformulation of Milner's Proof System for Regular Expressions Modulo Bisimilarity, Logical Methods in Computer Science 19 (2) (2023).

[22] C. Grabmayer, Structure-Constrained Process Graphs for the Process Semantics of Regular Expressions, in: TERMGRAPH@FSCD, 2020, pp. 29–45.

[23] T. Schmid, J. Rot, A. Silva, On star expressions and completeness theorems, Electronic Proceedings in Theoretical Computer Science 351 (2021) 242–259. doi:10.4204/eptcs.351.15.
URL http://dx.doi.org/10.4204/EPTCS.351.15