Milner’s Proof System for Regular Expressions
Modulo Bisimilarity is Complete

Crystallization: Near-Collapsing Process Graph Interpretations of Regular Expressions

Clemens Grabmayer
clemens.grabmayer@gssi.it
Gran Sasso Science Institute
L’Aquila, Abruzzo, Italy

Abstract
Milner (1984) defined a process semantics for regular expressions. He formulated a sound proof system for bisimilarity of process interpretations of regular expressions, and asked whether this system is complete.

We report conceptually on a proof that shows that Milner’s system is complete, by motivating and describing all of its main steps. We substantially refine the completeness proof by Grabmayer and Fokkink (2020) for the restriction of Milner’s system to ‘1-free’ regular expressions. As a crucial complication we recognize that process graphs with empty-step transitions that satisfy the layered loop-existence and elimination property LLEE are not closed under bisimulation collapse (unlike process graphs with LLEE that only have proper-step transitions). We circumnavigate this obstacle by defining a LLEE-preserving ‘crystallization procedure’ for such process graphs. By that we obtain ‘near-collapsed’ process graphs with LLEE whose strongly connected components are either collapsed or of ‘twin-crystal’ shape. Such near-collapsed process graphs guarantee provable solutions for bisimulation collapses of process interpretations of regular expressions.

1 Introduction
Kleene [15] (1951) introduced regular expressions, which are widely studied in formal language theory. In a typical formulation, they are constructed from constants 0, 1, letters a from some alphabet (interpreted as the formal languages Ø, {e}, and {a}, where e is the empty word) and binary operators + and ·, and the unary Kleene star * (which are interpreted as language union, concatenation, and iteration).

Milner [16] (1984) introduced a process semantics for regular expressions. He defined an interpretation C(e) of regular expressions e as charts (finite process graphs): the interpretation of 0 is deadlock, of 1 is successful termination, letters a are atomic actions, the operators + and · stand for choice and concatenation of processes, and (unary) Kleene star (·)∗ represents iteration with the option to terminate successfully before each execution of the iteration body. He then defined the process semantics of ‘star expressions’ (regular expressions in this context) e as ‘star behaviors’ [e]P := [C(e)][∞], that is, as equivalence classes of chart interpretations with respect to bisimilarity ≅. Milner was interested in an axiomatization of equality of ‘star behaviors’. For this purpose he adapted Salomaa’s complete proof system [17] for language equivalence on regular expressions to a system Mil (see Def. 3.6) that is sound for equality of denoted star behaviors. Recognizing that Salomaa’s proof strategy cannot be followed directly, he left completeness as an open question.

Over the past 38 years, completeness results have been obtained for restrictions of Milner’s system to the following subclasses of star expressions: (a) without 0 and 1, but with binary star iteration e1 ∗ e2 instead of unary star 2, (b) with 0, with iterations restricted to exit-less ones (·)∗ · 0, without 1 [6] and with 1 [5], (c) without 0, and with only restricted occurrences of 1 [4], and (d) ‘1-free’ expressions formed with 0, without 1, but with binary instead of unary iteration [14]. By refining concepts developed in [14] for the proof of (d) we can finally establish completeness of Mil.

The aim of this article. We provide an outline of the completeness proof for Mil. Hereby our focus is on the main new concepts and results. While details are sometimes only hinted at in this article, we think that the crystallization technique we present opens up a wide space for other applications (we suggest one in Sect. 10). We want to communicate this technique in summarized form to the community in order to stimulate its further development.

2 Motivation for the chosen proof strategy
We explain the main obstacle we encountered for developing our proof strategy through explaining shortcomings of existing approaches. Finally we describe crucial new concepts that we use for adapting the collapse strategy from [13, 14].

Obstacle for the ‘bisimulation chart’ proof strategy. Milner [16] recognized that completeness of the proof system
Mil cannot be established along the lines of Salomaa’s completeness proof for his proof system $F_{0}$ of language equivalence of regular expressions [17]. The reason is as follows.

Adopting Salomaa’s proof strategy would mean (i) to link given bisimilar chart interpretations $C(e_{1})$ and $C(e_{2})$ of star expressions $e_{1}$ and $e_{2}$ via a chart $B$ that represents a bisimulation between $C(e_{1})$ and $C(e_{2})$, (ii) to use this link via functional bisimulations from $B$ to $C(e_{1})$ and $C(e_{2})$ to prove equal in Mil the provable solutions $e_{1}$ of $C(e_{1})$, and $e_{2}$ of $C(e_{2})$, (iii) to extract from $B$ a star expression $e$ that provably solves $B$, and then is provably equal to $e_{1}$ and $e_{2}$. Here a ‘provable solution’ of a chart $C$ is a function $s$ from the set of vertices of $C$ to star expressions such that the value $s(v)$ at a vertex $v$ can be reconstructed, provably in Mil, from the transitions to, and the values of $s$ at, the immediate successor vertices of $v$ in $C$, and (non-)termination at $v$. By the ‘principal value’ of a provable solution we mean its value at the start vertex. In pictures we write ‘$e$ is solution’ for ‘$e$ is the principal value of a solution’.

$$
B \xrightarrow{e_{1} \text{ is solution of } B} C(e_{1}) \xrightarrow{(\text{assm})} e_{2} \text{ is solution of } C(e_{2}) \xrightarrow{?} e_{1} =_{\text{Mil}} e_{2}
$$

First by (i) star expressions $e_{1}$ and $e_{2}$ can be shown to be the principal values of provable solutions of their chart interpretations $C(e_{1})$ and $C(e_{2})$, respectively. These solutions can be transferred backwards by (ii) over the functional bisimulations from the bisimulation chart $B$ to $C(e_{1})$ and $C(e_{2})$, respectively. It follows that $e_{1}$ and $e_{2}$ are the principal values of two provable solutions of $B$. However, now the obstacle appears, because the extraction procedure in (iii) of a proof of $e_{1} =_{\text{Mil}} e_{2}$ in Mil cannot work, like Salomaa’s, for all charts $B$ irrespective of the actions of its transitions. An example that demonstrates that is the chart $C_{12}$ in Ex. 4.1 in [13, 14]. This is because some charts are unsolvable (“In contrast with the case for languages—an arbitrary system of guarded equations in [star]-behaviours cannot in general be solved in star expressions” [16]), but turn into a solvable one if all actions in it are replaced by a single one. The reason for the failure of Salomaa’s extraction procedure is then that the absence in Mil of the left-distributivity law $x \cdot (y + z) = x \cdot y + x \cdot z$ (it is not sound under bisimilarity) frequently prevents applications of the fixed-point rule $\text{RSP}^{*}$ in Mil unlike for the system $F_{0}$ that Salomaa proved complete. We conclude that such a bisimulation-chart proof strategy, inspired by Salomaa [17], is not expedient for showing completeness of Mil.

However, if the fixed-point rule $\text{RSP}^{*}$ in Milner’s system is replaced in Mil by a general unique-solvability rule scheme USP for guarded systems of equations, then a proof system arises to which the bisimulation-chart proof strategy is applicable. That system can therefore be shown to be complete comparatively easily (as noted in [10]).

**Loop existence and elimination.** A sufficient structural condition for solvability of a chart, and correspondingly of a linear system of recursion equations, by a regular expression modulo bisimilarity was given by Grabmayer and Fokkink in [14]: the ‘loop existence and elimination’ condition LEE, and its ‘layered’ specialization LLEE, which is independent of the specific actions in a chart. These properties are refinements for graphs of ‘well-behaved specifications’ due to Baeten and Corradini in [2] that single out a class of ‘palm trees’ (trees with back-links) that specify star expressions under the process interpretation. For showing that the tailored restriction BBP of Milner’s system Mil to ‘1-free’ star expressions (without 1, but with binary instead of unary star iteration) is complete, the following properties were established in [14]: (L) Chart interpretations of 1-free star expressions are LLEE-charts. (S) Every 1-free star expression $e$ is the principal value of a provable solution of its chart interpretation $C(e)$. (E) From every LLEE-chart $C$ a provable solution of $C$ (by 1-free star expressions) can be extracted. (S) All provable solutions of a LLEE-chart are provably equal. (L) Every provable solution can be transferred from the target to the source chart of a functional bisimulation to obtain a provable solution of the source chart. (C) The bisimulation collapse of a LLEE-chart is again a LLEE-chart.

As a consequence of these properties, a finite chart $C$ is expressible by a 1-free star expression modulo bisimilarity if and only if the bisimulation collapse of $C$ satisfies LEE.

**The ‘bisimulation collapse’ proof strategy for BBP ([14]).** For the completeness proof of the tailored restriction BBP of Milner’s system Mil to 1-free star expressions, Grabmayer and Fokkink in [14] linked bisimilar chart interpretations $C(e_{1})$ and $C(e_{2})$ of 1-free star expressions $e_{1}$ and $e_{2}$ via the joint bisimulation collapse $C_{0}$. That argument, which we recapitulate below, can be illustrated as follows:

$$
\begin{align*}
\text{LLEE} & \iff C(e_{1}) \xrightarrow{(\text{assm})} C(e_{2}) \\
\text{LLEE} & \iff \text{LLEE} \\
\text{LLEE} & \iff \text{LLEE} \\
\text{LLEE} & \iff \text{LLEE} \\
\text{LLEE} & \iff \text{LLEE} \\
\end{align*}
$$

By (S), the star expressions $e_{1}$ and $e_{2}$ are the principal values of provable solutions $s_{1}$ and $s_{2}$ of their chart interpretations $C(e_{1})$ and $C(e_{2})$, respectively. Furthermore, by (L) the chart interpretations of the 1-free star expressions $C(e_{1})$ and $C(e_{2})$ have the property LEE. Since LEE is preserved under the operation of bisimulation collapse due to (C), the joint bisimulation collapse $C_{0}$ of $C(e_{1})$ and $C(e_{2})$ is again a LLEE-chart. Therefore a provable solution $s_{0}$ can be extracted from $C_{0}$ due to (E). Let $e_{0}$ be its principal value. The solution $s_{0}$ can be transferred from $C_{0}$ backwards over the functional bisimulations from $C(e_{1})$ and $C(e_{2})$ to $C_{0}$ due to (L), and thereby defines provable solutions $s_{1}$ of $C(e_{1})$ and $s_{2}$ of $C(e_{2})$, both with $e_{0}$ as the principal value. Now having
provable solutions \( s_1 \) and \( \bar{s}_1 \) of the LEE-chart \( C(e_1) \), these solutions are BBP-provably equal by \((S\subseteq)\), and hence also their principal values \( e_1 \) and \( e_0 \), that is \( e_1 =_{\text{BBP}} e_0 \). Analogously \( e_1 =_{\text{BBP}} e_0 \) can be established. Then \( e_1 =_{\text{BBP}} e_2 \) follows by applying symmetry and transitivity proof rules of equational logic.

**Obstacles for a 'bisimulation collapse' strategy for Mil.** A generalization of this argument for arbitrary star expressions runs into two problems that can be illustrated as:

\[
\begin{align*}
\text{(1)} & \quad C(e) \xrightarrow{\text{LEE}} C(e) \xrightarrow{\text{LLEE}} C_0 \\
\text{any 1-bisimulation collapse} & \quad \xrightarrow{\text{LLEE}} \text{of } C
\end{align*}
\]

First, see (1), there are star expressions \( e \) whose chart interpretation \( C(e) \) satisfies neither LEE nor LLEE, as was noted in [12]. In order to still be able to utilize LEE, in [12] a variant chart interpretation \( C(e) \) was defined for star expressions \( e \) such that \( C(e) \) is a 'LEE-1-chart', that is, a chart with '1-transitions' (explicit empty-step transitions) that satisfies LEE, and \( C(e) \) is '1-bisimilar' to the chart interpretation \( C(e) \). Hereby '1-bisimulations' and '1-bisimilarity' are adaptations of bisimulations and bisimilarity to 1-charts.

However, use of the variant chart interpretation encounters the second obstacle (C) as illustrated above. A part of it that was also observed in [12] is that LEE-1-charts are not closed under bisimulation collapse, unlike LEE-charts. While this renders the bisimulation collapse proof strategy unusable, we here show that an adaptation to a '1-bisimulation collapse' strategy is not possible, either, if it is based on '1-bisimulation collapsed' 1-charts in which none of its vertices are 1-bisimilar. In particular we show (C) as the first of our key observations and concepts as listed below:

1. LEE-1-charts are not in general collapsible to (collapsed) LEE-1-charts. Nor do 1-bisimilar LEE-1-charts always have a joint (1-bisimilarity) minimization. We demonstrate this by an example (see Fig. 4).

The second part of (1) prevents a change from the '1-bisimilarity collapse strategy' to a 'joint minimization strategy' that would use the weaker statement that any two 1-bisimilar LEE-1-charts were always jointly minimizable under 1-bisimilarity (which is also wrong due to (1)). We want to credit Schmid, Rot, and Silva’s careful coinductive analysis in [18] of the proof in [14], which helped us to realize that this possibility needs to be excluded as well.

**How we recover the collapse proof strategy for Mil.** We define 'crystallized' approximations with LEE of collapsed LEE-1-charts in order to show that bisimulation collapses of LEE-1-charts have provable solutions. For this purpose we combine the following concepts and their properties:

2. **Twin-Crystals**: These are 1-charts with a single strongly connected component (scc) that exhibit a self-inverse symmetry function that links 1-bisimilar vertices. Twin-Crystals abstract our example that demonstrates (1).

3. **Near-Collapsed 1-charts**: These are 1-charts in which 1-bisimilar vertices appear as pairs that are linked by a self-inverse function that induces a 'grounded 1-bisimulation slice'. Twin-Crystals are near-collapsed LEE-1-charts.

4. **Crystallization**: By this we understand a process of stepwise minimization of LEE-1-charts under 1-bisimilarity that produces 1-bisimilar 'crystallized' LEE-1-charts in which all strongly connected components are collapsed or of twin-crystal shape. This process uses the connective operation from [14] for 1-bisimilar vertices. We show that crystallized 1-charts are near-collapsed.

5. **Complete Mil-provable solution of a 1-chart \( C \)**: This is a Mil-provable solution of \( C \) with the property that its values for 1-bisimilar vertices of \( C \) are Mil-provably equal. Any complete Mil-provable solution of a 1-chart \( C \) yields a Mil-provable solution of the bisimulation collapse of \( C \).

6. **Elevation of vertex sets above 1-charts**: This is a concept of partially unfolding 1-charts that facilitates us to show that near-collapsed weakly guarded LEE-1-charts have complete Mil-provable solutions.

With these conceptual tools we will be able to recover the collapse proof strategy for LEE-1-charts. The idea is to establish, for given 1-bisimilar LEE-1-charts \( C_1 \) and \( C_2 \), a link via which solutions of \( C_1 \) and \( C_2 \) with the same principal value can be obtained. We create such a link via the crystallized LEE-1-chart \( C_{10} \) of (one of them, say) \( C_1 \) and the joint bisimulation collapse \( C_0 \) of \( C_1 \) and \( C_2 \). Then a solution of \( C_{10} \) can obtained, transferred first to \( C_0 \), and then to \( C_2 \) and to \( C_2 \). More precisely, the central part of our completeness proof for Mil can be illustrated as follows:

Here we start from the assumption of 1-bisimilar 1-charts \( C_1 \) and \( C_2 \) of which at least \( C_1 \) satisfies LEE. Then the LEE-1-chart \( C_1 \) can be crystallized with as result a 1-bisimilar LEE-1-chart \( C_{10} \). Due to LEE, a provable solution \( s_{10} \) can be extracted from \( C_{10} \). Let \( e_{10} \) be the principal value of \( s_{10} \). Since crystallized 1-charts are near-collapsed LEE-1-charts, the solution \( s_{10} \) of \( C_{10} \) is complete. As such it defines a provable solution with principal value \( e_{10} \) on the joint bisimulation collapse \( C_0 \) of \( C_0 \), \( C_1 \), and \( C_2 \). This solution
can be pulled back conversely over the functional 1-bisimulations from $C_0 \to C_1$ and to $C_2$. In this way we obtain provable solutions $s_1$ of $C_1$ and $s_2$ of $C_2$, both with the same principal value $e_0$.

This is the central argument of our completeness proof for Mil that we give in Sect. 5 based on lemmas. Its illustration appears in the illustration of the completeness proof in Fig. 3.

3 Preliminaries

Let $A$ be a set whose members we call actions. The set $StExp(A)$ of star expressions over actions in $A$ is defined by the following grammar, where $a \in A$:

$$e, e_1, e_2 :: = 0 \mid 1 \mid a \mid e_1 + e_2 \mid e_1 \cdot e_2 \mid e^*$$

**Definition 3.1** (1-charts and 1-LTSs). A 1-chart is a 6-tuple $\langle V, A, 1, s_o, streamline, 1 \rangle$ with $V$ a finite set of vertices, $A$ a finite set of (proper) action labels, $1 \notin A$ the specified empty step label, $s_o \in V$ the start vertex (hence $V \neq \emptyset$), $\rightarrow \subseteq V \times A \times V$ the labeled transition relation, where $A := A \cup \{1\}$ is the set of action labels including 1, and $\downarrow \subseteq V$ a set of vertices with immediate termination (or terminating vertices). In such a 1-chart, we call a transition in $\rightarrow \cap (V \times A \times V)$ (labeled by a proper action in $A$) a proper transition, and a transition in $\rightarrow \cap (V \times \{1\} \times V)$ (labeled by 1) a 1-transition. Reserving non-underlined action labels like $a, b, \ldots$ for proper actions, we use highlighting underlined action label symbols like $\hat{a}$ for actions in the set $A$ that includes the label 1.

A 1-LTS (a labeled transition system with 1-transitions and immediate termination) is a 5-tuple $\langle V, A, 1, s_o, streamline \rangle$ with concepts as explained above but without a start vertex. For every 1-chart $C = \langle V, A, 1, s_o, streamline \rangle$, we denote by $\mathcal{L}(C) = \langle V, A, 1, streamline \rangle$ the 1-LTS underlying $C$ (or that underlies $C$).

We say that a 1-chart $C$ (a 1-LTS $\mathcal{L}$) is weakly guarded (w.g.) if $\mathcal{L}$ (resp. $\mathcal{L}$) does not have an infinite path of 1-transitions.

By an induced a-transition $a \rightarrow w$, for a proper action $a \in A$, in a 1-chart $C$ (in a 1-LTS $\mathcal{L}$) we mean a path of the form $v \xrightarrow{\hat{a}} \cdots \xrightarrow{\hat{a}} \xrightarrow{\hat{1}} w$ in $C$ (in $\mathcal{L}$) that consists of a finite number of 1-transitions that ends with a proper a-transition. By induced termination $\{\hat{1}\}$, for $v \in V$ we mean that there is a path $v \xrightarrow{\hat{1}} \cdots \xrightarrow{\hat{1}} \hat{1}$ with $\hat{1}$ in $C$ (in $\mathcal{L}$).

By a chart (a LTS) we mean a 1-transition free 1-chart (a 1-transition free LTS). Let $C = \langle V, A, 1, s_o, streamline \rangle$ be a 1-chart. We define by $C_{\downarrow} := \langle V, A, 1, s_o, \{\hat{1}\}, \{\hat{1}\} \rangle$ the induced chart of $C$ whose transitions are the induced transitions of $C$, and whose terminating vertices are the vertices of $C$ with induced termination. Note that $C_{\downarrow}$ is 1-transition free. Also, for every vertex $w \in V$ we denote by $C_w := \langle V, A, 1, w, streamline \rangle$ the generated sub(1)-chart of $C$ at $w$.

**Definition 3.2** (1-bisimulating slices, 1-bisimulations for 1-LTSs). Let $\mathcal{L}_i = \langle V_i, A, 1, streamline, 1 \rangle$ for $i \in \{1, 2\}$ be 1-LTSs.

A 1-bisimulating slice between $\mathcal{L}_1$ and $\mathcal{L}_2$ is a binary relation $B \subseteq V_1 \times V_2$, with active domain $W_i := \text{dom}_{\text{act}}(B) = \pi_1(B)$, and active codomain $W_2 := \text{cod}_{\text{act}}(B) = \pi_2(B)$, where $\pi_i : V_1 \times V_2 \to V_i, \pi_i((v_1, v_2)) = v_i$ for $i \in \{1, 2\}$, such that $B \neq \emptyset$, and for all $(v_1, v_2) \in B$ the three conditions hold:

**Forth** $a \in A \forall v_1' \in V_1 (v_1 \xrightarrow{a} v_1' \land v_1' \in W_1)$

$$\Rightarrow \exists v_2' \in V_2 (v_2 \xrightarrow{a} v_2' \land \langle v_1', v_2' \rangle \in B)$$

**Back** $a \in A \forall v_2' \in V_2 (v_2 \xrightarrow{a} v_2' \land v_2' \in W_2)$

$$\Rightarrow \exists v_1' \in V_1 (v_1 \xrightarrow{a} v_1' \land \langle v_1', v_2' \rangle \in B)$$

(termination) $v_{1+1}^{(1)} \xrightarrow{a} v_{2+1}^{(2)}$.

Here (forth), entails $v_2' \in W_2$, and (back), entails $v_1' \in W_1$.

A 1-bisimulation between $\mathcal{L}_1$ and $\mathcal{L}_2$ is a 1-bisimulating slice $B$ between $\mathcal{L}_1$ and $\mathcal{L}_2$, such that the active domain of $B$, and the active codomain of $B$, are transition-closed (that is, closed under $\rightarrow_1$ and $\rightarrow_2$, respectively), or equivalently, a non-empty relation $B \subseteq V_1 \times V_2$ such that for every $(w_1, w_2) \in B$ the conditions (forth), (back), (termination) hold, where (forth), and (back) result from (forth), and (back), by dropping the underlined conjuncts.

By a 1-bisimulating slice between $\mathcal{L}_1$ and $\mathcal{L}_2$, we mean a 1-bisimulating slice between $\mathcal{L}_1$ and $\mathcal{L}_2$ that is contained in a 1-bisimulation between $\mathcal{L}_1$ and $\mathcal{L}_2$.

A 1-bisimulating slice (a 1-bisimulation slice, a 1-bisimulation) on a 1-LTS $\mathcal{L}$ is a 1-bisimulating slice (and resp., a 1-bisimulation slice, a 1-bisimulation) between $\mathcal{L}$ and $\mathcal{L}$.

**Definition 3.3** ((funct.) 1-bisimulation between 1-charts). We consider 1-charts $C_i = \langle V_i, A, 1, s_o, streamline \rangle$ for $i \in \{1, 2\}$.

A 1-bisimulation between 1-charts $C_1$ and $C_2$ is a 1-bisimulation $B \subseteq V_1 \times V_2$ between the 1-LTSs $\mathcal{L}(C_1)$ and $\mathcal{L}(C_2)$, respectively, such that additionally:

- (start) $(s_o, s_o) \in B$ (relates start vertices of $C_1$ and $C_2$)
- (cond) $B$ must satisfy the conditions (forth), (back), (termination) from Def. 3.2.

By a functional 1-bisimulation from $C_1$ to $C_2$, we mean a 1-bisimulation between $C_1$ and $C_2$, that is the graph of a partial function from $V_1$ to $V_2$. By $C_1 \xrightarrow{\text{fbs}} C_2$ (by $C_1 \xrightarrow{\text{fbs}, \text{fbs}} C_2$), we denote that there is a 1-bisimulation between $C_1$ and $C_2$, respectively, a functional 1-bisimulation from $C_1$ to $C_2$.

**Definition 3.4.** Let $C_i = \langle V_i, A, 1, s_o, streamline \rangle$ be a 1-chart.

By $\text{fbs}_C$ we denote 1-bisimilarity on $C_i$ the largest 1-bisimulation (which is the union of all 1-bisimulations) between $C$ and itself. If $w_1 \xrightarrow{\text{fbs}_C} w_2$ holds for vertices $w_1, w_2 \in V$, then we say that $w_1$ and $w_2$ are 1-bisimilar in $C_i$.

We call $C_i$ 1-collapsed, and a 1-bisimulation collapse, if $\text{fbs}_C = \text{id}_C$ holds, that is, if 1-bisimilar vertices of $C_i$ are identical. If, additionally, $C_i$ does not contain any 1-transitions, then we call $C_i$ collapsed, and a bisimulation collapse.

Let $w_1, w_2 \in V$. We say that $w_1$ is a substate of $w_2$, denoted by $w_1 \subseteq w_2$, if the pair $(w_1, w_2)$ forth-progresses to 1-bisimilarity on $C_i$ in the sense of the following conditions:

4
(prog-forth) \( \forall w_1' \in V_1 \forall a \in A \{ w_1 \xrightarrow{(a)} w_1' \) \implies \exists w_2' \in V_2 \{ w_2 \xrightarrow{(a)} w_1' \land w_2' \xrightarrow{a} w_2 \} \),
(prog-termination) \( w_1^+[10] \implies w_2^+[10] \).

**Definition 3.5.** The chart interpretation of a star expression \( e \in StExp(A) \) is the \((1,\text{-}tr.)\) chart \( C(e) = \langle V(e), A, 1, e, \rightarrow \cap \nabla (V(e) \times A \times V(e)), \nabla \cap V(e) \rangle \) where \( V(e) \) consists of all star expressions that are reachable from \( e \) via transitions of the labeled transition relation \( \rightarrow \subseteq StExp(A) \times A \times StExp(A) \), which is defined, together with the imm.-termination relation \( \downarrow \subseteq StExp(A) \), by derivability in the transition system specification (TSS) \( S \langle A \rangle \), where \( a \in A, e, e_1, e_2, e' \in StExp(A) \):

\[
\begin{array}{c|c|c|c}
\backslash & e_1 & (e_1 \cdot e_2) & (e^*) \\
\hline
\rightarrow & e_1 \xrightarrow{a} e'_1 & (e_1 \cdot e_2) \xrightarrow{a} e'_2 & (e^*) \xrightarrow{a} e'_3 \\
\hline
a & \rightarrow 1 & e_1 + e_2 \xrightarrow{a} e'_1 & e^* \xrightarrow{a} e' \cdot e^* \\
\hline
& e_1 \cdot e_2 \xrightarrow{a} e'_1 \cdot e_2 & e_1 \cdot e_2 \xrightarrow{a} e'_2 & e_1 \cdot e_2 \xrightarrow{a} e'_3 \\
\end{array}
\]

**Definition 3.6.** Milner’s proof system Mil on star expressions has the following axioms (here numbered differently):

(A1) \( e + (f + g) = (e + f) + g \) (A7) \( e = 1 \cdot e \) (A2) \( e + 0 = e \) (A8) \( e = e \cdot 1 \) (A3) \( e + f = f + e \) (A9) \( 0 = 0 \cdot e \) (A4) \( e + e = e \) (A10) \( e^* = 1 + e \cdot e^* \) (A5) \( e \cdot (f + g) = (e \cdot f) + g \) (A11) \( e^* = (1 + e)^* \) (A6) \( (e + f) \cdot g = e \cdot g + f \cdot g \)

The rules of Mil are the basic inference rules of equational logic (reflexivity, symmetry, transitivity of =, compatibility of = with +, \cdot, (\cdot)^*) as well as the fixed-point rule RSP*:

\[
e = f \cdot e + g \iff e = f^* \cdot g \quad \text{RSP* if } f \neq \emptyset
\]

By \( e_1 =_{\text{Mil}} e_2 \) we denote that \( e_1 = e_2 \) is derivable in Mil.

By \( \text{Mil}^- \) we denote the purely equational part of Mil that results by dropping the rule scheme RSP* from Mil.

**Definition 3.7.** While we formulate the stipulations below for 1-LTSS, we will use them also for 1-charts. So we let \( L^\langle \rangle = \langle V, A, 1, \rightarrow, \downarrow \rangle \) be a 1-LTS, and we let \( S \in \{ \text{Mil}, \text{Mil}^- \} \).

By a star expression function on \( L^\langle \rangle \) we mean a function \( s : V \rightarrow StExp(A) \) on the vertices of \( L^\langle \rangle \). Now we let \( v \in V \). We say that such a star expression function \( s \) on \( L^\langle \rangle \) is an \( S \)-provably solution of \( L^\langle \rangle \) at \( v \) if it holds that:

\[
s(v) = S \tau_L^\langle \rangle (v) + \sum_{i=1}^{n} a_i \cdot s(u_i)
\]

given that \( \tau_L^\langle \rangle (v) = \{ v \xrightarrow{a} u_i \mid i \in \{1, \ldots, n\} \} \) is a (possibly redundant) list representation of transitions from \( v \) in \( L^\langle \rangle \) and where \( \tau_L^\langle \rangle (v) \) is the termination constant \( \tau_L^\langle \rangle (v) \) of \( L^\langle \rangle \) at \( v \)

defined as \( 0 \) if \( v \downarrow \), and as \( 1 \) if \( v \downarrow \). This definition does not depend on the specifically chosen list representation of \( \tau_L^\langle \rangle (v) \), because \( S \) contains the associativity, commutativity, and idempotency axioms for +.

By an \( S \)-provably solution of \( L^\langle \rangle \) (with principal value \( s(v) \)) at the start vertex \( v_0 \) we mean a star expression function \( s \) on \( L^\langle \rangle \) that is an \( S \)-provably solution of \( L^\langle \rangle \) at every vertex of \( L^\langle \rangle \).

We say that an \( S \)-provably solution \( s \) of \( L^\langle \rangle \) is \( S \)-complete if:

\[
w_1 \xrightarrow{a} w_2 = S(w_1) = S(w_2) \quad \text{(concept (5))}
\]

to hold for all \( w_1, w_2 \in V \), that is, if values of the solution \( s \) at 1-bisimilar vertices of \( L^\langle \rangle \) are \( S \)-provably equal.

The following lemma gathers preservation statements of (complete) provable solutions under (functional) 1-bisimilarity that are crucial for the completeness proof.

**Lemma 3.8.** On weakly guarded 1-charts, the following statements hold for all star expressions \( e \in StExp \):

(i) Mil-Provable solvability with principal value \( e \) is preserved under converse functional 1-bisimilarity.

(ii) Mil-Complete Mil-provable solvability with principal value \( e \) of a w.g. 1-chart \( C \) implies Mil-provable solvability with principal value \( e \) of the bisimulation collapse of \( C \). (See (5).)

(iii) Mil-Complete Mil-provable solvability with principal value \( e \) is preserved under 1-bisimilarity.

### 4 LLEE-1-Charts

We use the adaptation of the ‘loop existence and elimination property’ LEE from [14] to 1-charts as described in [8, 10]. Here we only briefly explain the concept by examples, and refer to [8, 10] and to the appendix for the definitions. Crucially, we gather statements from [8, 10] that we need for the proof.

LEE is defined by a stepwise elimination procedure of ‘loop sub-1-charts’ from a given 1-chart \( C \). A run of this procedure is illustrated in Fig. 1. A 1-chart \( \overline{C} \) of \( C \) implies Mil-provable solvability with principal value \( e \) of the bisimulation collapse of \( C \). (See (5).)

(L1) There is an infinite path from the start vertex \( v_0 \).

(L2) Every infinite path from \( v_0 \) returns to \( v_0 \) after a positive number of transitions.

(L3) Immediate termination is only permitted at the start vertex, that is, \( \downarrow \subseteq \{ v_0 \} \).

A loop sub-1-chart of a 1-chart \( C \) is a loop 1-chart \( \overline{C} \) that is a sub-1-chart of \( C \) with some vertex \( v \in V \) of \( C \) as start vertex, such that \( \overline{C} \) is constructed, for a nonempty set \( U \) of transitions of \( C \) from \( v \), by all paths that start with a transition in \( U \) and continue onward until \( v \) is reached again (so the transitions in \( U \) are the loop-entry transitions of \( C \)).

Eliminating a loop sub-1-chart \( \overline{C} \) from a 1-chart \( C \) consists of removing all loop-entry transitions of \( \overline{C} \) from \( C \) and then also removing all vertices and transitions that become unreachable. Fig. 1 shows a successful three-step run of the
loop elimination procedure. A 1-chart \( \hat{C} \) has the loop existence and elimination property (LEE) if the procedure, started on \( \hat{C} \), of repeated eliminations of loop sub-1-charts results in a 1-chart without an infinite path. If, in a successful elimination process from a 1-chart \( \hat{C} \), loop-entry transitions are never removed from the body of a previously eliminated loop sub-1-chart, then we say that \( \hat{C} \) satisfies layered LEE (LEE), and is a LLEE-1-chart. LLEE-1-LTSs are 1-LTSs that are defined analogously. While the property LLEE leads to a formally easier concept of ‘witness’, it is equivalent to LEE. Since the resulting 1-chart \( \hat{C}'' \) in Fig. 1 does not have an infinite path, and no loop-entry transitions have been removed from a previously eliminated loop sub-1-chart, we conclude that the initial 1-chart \( \hat{C} \) satisfies LLEE as well as LEE.

A LLEE-witness \( \hat{C} \) of a 1-chart \( C \) is the recording of a successful run of the loop elimination procedure by attaching to a transition \( \tau \) of \( C \) the marking label \( n \) for \( n \in \mathbb{N}^+ \) (in pictures indicated as \( [n] \), in steps as \( \tau \rightarrow [n] \)) forming a loop-entry transition if \( \tau \) is eliminated in the \( n \)-th step, and by attaching marking label \( 0 \) to all other transitions of \( C \) (in pictures neglected, in steps indicated as \( \tau \rightarrow \)) forming a body transition. Formally, LLEE-witnesses arise as entry/body-labelings from 1-charts, and are charts in which the transition labels are pairs of action labels over \( A \), and marking labels in \( \mathbb{N} \).

The LLEE-witness \( \hat{C} \) in Fig. 2 arises from the run of the loop elimination procedure in Fig. 1. The LLEE-witnesses \( \hat{C}_1 \), and \( \hat{C}_3 \) of \( C \) in Fig. 2 record the two successful runs of the loop elimination procedure of length 4 and 2, respectively, where for \( \hat{C}_3 \) we have permitted to eliminate two loop sub-charts at different vertices together in the first step.

**Definition 4.1** (1-charts with LLEE-1-lim). Let \( C \) be a 1-chart.

Let \( \hat{C} \) be a LLEE-witness of \( C \). We say that \( \hat{C} \) is 1-transition limited if every 1-transition of \( \hat{C} \) lifts to a backlink (a transition from the body of a loop sub-1-chart back to its start) in \( \hat{C} \). Fixing also a weaker property, we say that \( \hat{C} \) is guarded if all of its loop-entry transitions are proper transitions.

We say that \( C \) satisfies LLEE-1-lim, and is 1-transition limited, if \( C \) has a 1-transition limited LLEE-witness. We say that \( \hat{C} \) is guarded if \( \hat{C} \) has a guarded LLEE-witness.

We note that 1-transition limited LLEE-1-charts are guarded (since loop-entry transition are not backlinks), and guarded LLEE-1-charts are weakly guarded (since in a guarded LLEE-witness every 1-transition path is a body transition path, which as in every LLEE-witness is guaranteed to be finite).

**Lemma 4.2.** Every weakly guarded LLEE-1-chart is 1-bisimilar to a (guarded) 1-chart with LLEE-1-lim.

Two crucial properties of LLEE-1-charts that motivate their use, like for LLEE-charts earlier in [13, 14], are their provable solvability and unique solvability modulo provability in Mil. The two lemmas below that express these properties are generalizations to LLEE-1-charts of Prop. 5.5 and Prop. 5.8 in [13, 14], and have been proved in [9, 10].

**Lemma 4.3.** From every guarded LLEE-witness \( \hat{C} \) of a (weakly guarded) LLEE-1-chart \( C \) a Mil- provable solution \( \hat{C}_1 \) of \( C \) can be extracted effectively.

**Lemma 4.4.** For every guarded LLEE-1-chart \( C \) it holds that any two Mil-provable solutions of \( C \) are Mil-provably equal.

In Sect. 9 we will need a consequence of Lem. 4.4, namely provable invariance of provable solutions under ‘transfer functions’, which define functional 1-bisimulations. While this statement holds also for LLEE-1-charts, we have to formulate it for LLEE-1-LTSs for use later in Sect. 9.

**Definition 4.5.** A transfer (partial) function between 1-LTSs \( L_1 \) and \( L_2 \), for \( L_2 \), is a partial function \( \phi : V_1 \rightarrow V_2 \) whose graph \( \{ (v, \phi(v)) \mid v \in V_1 \} \) is a 1-bisimulation between \( L_1 \) and \( L_2 \).

**Lemma 4.6.** Let \( \phi : V_1 \rightarrow V_2 \) be a transfer function between LLEE-1-LTSs \( L_1 \) and \( L_2 \), for \( i \in \{1, 2\} \). Then for all Mil-provable solutions \( s_1 \) of \( L_1 \), and \( s_2 \) of \( L_2 \) it holds that \( s_1 \) coincides Mil-provably with the precomposition \( s_2 \circ \phi \) of \( s_2 \) and \( \phi \):

\[
\text{s}_1(w) = \text{Mil}_\phi \text{s}_2(\phi(w)), \quad \text{for all } w \in \text{dom}(\phi).
\]
A substantial obstacle for the use of LLEE-1-charts was already recognized in [14]: the chart interpretation of star expressions does not in general define LLEE-charts. This obstacle can, however, be navigated successfully by using the result from [8, 12] that a variant chart interpretation can be defined that produces 1-bisimilar LLEE-1-charts instead.

**Lemma 4.7.** For every star expression \( e \), there is a 1-chart interpretation \( C(e) \) of \( e \) that has the following properties:

(i) \( C(e) \) is a 1-transition limited (guarded) LLEE-1-chart,

(ii) \( C(e) \) is complete with respect to process semantics equality of regular expressions.

(iii) \( e \) is the principal value of a Mil-provable solution of \( C(e) \).  

**5 Completeness proof based on lemmas**

We anticipate the completeness proof for Milner's system by basing it on the following lemmas, which are faithful abbreviations of statements as formulated in other sections. The chosen acronyms for these lemmas stem from the letters that are typeset in boldface italics in their statements:

**IV** For every star expression \( e \), there is a 1-chart interpretation (variant) \( C(e) \) of \( e \) with the properties (i), (ii), and (iii) in Lem. 4.7 (see above).

**T** Provables solutions can be transferred backwards over a transfer function between weakly guarded 1-charts. (See Lem. 3.8, (i)).

**E** From every guarded LLEE-1-chart \( C_0 \) a provable solution of \( C \) can be extracted. (See Lem. 4.3.)

**CR** Every guarded LLEE-1-chart can be transformed into a 1-bisimilar (guarded) crystallized (LLEE-)1-chart. (See Thm. 8.9, based on Def. 8.5.)

**CN** Every crystallized 1-chart is near-collapsed. (Lem. 8.8.)

**NC** Solutions extracted from near-collapsed guarded LLEE-1-charts are complete provable solutions. (See Lem. 9.2.)

**CC** If a weakly guarded 1-chart \( C \) has a complete provable solution with principal value \( e \), then also the (1-transition free) bisimulation collapse of \( C \) has a provable solution with principal value \( e \). (See Lem. 3.8, (ii).)

**SE** All provable solutions of a guarded LLEE-1-chart are provably equal. (See Lem. 4.4.)

**Theorem 5.1.** Milner’s proof system Mil is complete with respect to process semantics equality of regular expressions.

Proof. (See Fig. 3 for an illustration.) Let \( e_1, e_2 \in StExp(A) \) be star expressions such that \( \llbracket e_1 \rrbracket_p = \llbracket e_2 \rrbracket_p \), holds, that is, their process interpretations coincide. This means that the behaviors \( \llbracket C(e_1) \rrbracket_\mathcal{P} \) and \( \llbracket C(e_2) \rrbracket_\mathcal{P} \) of the chart interpretations \( C(e_1) \) of \( e_1 \) and \( C(e_2) \) of \( e_2 \) coincide. Therefore \( C(e_1) \) and \( C(e_2) \) are bisimilar. We have to show \( e_1 =_{Mil} e_2 \), that is, that \( e_1 \) and \( e_2 \) are provably equal in Milner’s system Mil. By "f is (complete) solution of \( C \)" we here mean that "f is the principal value of a (complete) provable solution of \( C \)." The indicated lemmas are explained in Sect. 5.
Now let \( C_0 \) be the (1-transition free) bisimulation collapse of \( C_{10} \). By applying (CC) to the complete provable solution \( s_{10} \) of \( C_{10} \), which is weakly guarded since it is guarded, we obtain a provable solution \( s_0 \) of \( C_0 \) that also has the principal value \( e_{10} \). Due to \( C_{i}(e_1) \sim C_{i}(e_2) \), \( C_0 \) is the joint bisimulation collapse of \( C_{i}(e_1) \) and \( C_{i}(e_2) \). It follows that there are functional 1-bisimulations from \( C_{i}(e_1) \) and from \( C_{i}(e_2) \) to \( C_0 \), that is, \( C_{i}(e_1) \sim C_0 \sim C_{i}(e_2) \). Now we can use (T) to transfer the provable solution \( s_0 \) from \( C_0 \) to \( C_{i}(e_1) \) and to \( C_{i}(e_2) \). We obtain provable solutions \( s_1 \) of \( C_{i}(e_1) \), and \( s_2 \) of \( C_{i}(e_2) \), both of which have \( e_{10} \) as their principal value.

Since both \( s_1 \) and \( s_2 \) are provable solutions of the guarded LLEE-1-chart \( C_{i}(e_1) \), we can apply (SE) to find that \( s_1 \) and \( s_2 \) are provably equal. In particular, the principal values \( e_1 \) of \( s_1 \) and \( e_10 \) of \( s_2 \) are provably equal. That is, \( e_1 =_{\text{Mil}} e_{10} \) holds. Analogously, as both \( s_2 \) and \( s_2 \) are provable solutions of the guarded LLEE-1-chart \( C_{i}(e_2) \), (SE) also entails that \( s_2 \) and \( s_2 \) are provably equal. Therefore also the principal values \( e_2 \) of \( s_2 \) and \( e_10 \) of \( s_2 \) are provably equal. That is, \( e_2 =_{\text{Mil}} e_{10} \) holds.

From \( e_1 =_{\text{Mil}} e_{10} \) and \( e_2 =_{\text{Mil}} e_{10} \) we obtain \( e_1 =_{\text{Mil}} e_2 \), and hence that \( e_1 = e_2 \) is provable in Milner’s system Mil.

\[ \square \]

6 Failure of LLEE-preserving 1-collapse

Here we expand on observation (1), due to which we have realized in Sect. 2 that the bisimulation collapse strategy in [14] cannot be extended directly to a 1-bisimulation collapse strategy for showing completeness of Mil. We define the properties ‘collapsible’, ‘1-collapsible’, and ‘jointly minimizable’ formally, formulate their failure for LLEE-1-charts, and suggestively explain the reason by means of an example.

Let \( C \) be a LLEE-1-chart. We say that \( C \) is LLEE-preservingly 1-collapsible (LLEE-preservingly collapsible) if \( C \) has a 1-bisimulation collapse that is a LLEE-1-chart (and respectively, the bisimulation collapse of \( C \) is a LLEE-1-chart).

We say that two LLEE-1-charts \( C_1 \) and \( C_2 \), that are 1-bisimilar (that is, with \( C_1 \sim C_2 \) are LLEE-preservingly jointly minimizable (under functional 1-bisimilarity \( \sim \)) if there is a LLEE-1-chart \( C \) such that \( C_1 \sim C_0 \sim C_2 \).

**Proposition 6.1** (1). The following two statements hold:

(i) W.g. LLEE-1-charts are not in general LLEE-preservingly 1-collapsible, hence not in general LLEE-pres. collapsible.

(ii) Two 1-bisimilar w.g. LLEE-1-charts are not in general LLEE-preservingly jointly minimizable under \( \sim \).

Statement (i) is witnessed by the LLEE-1-chart \( C \) in Fig. 4. Statement (ii) is witnessed by the 1-bisimilar generated sub-LLEE-1-charts \( C^{\text{10}}_1 \) and \( C^{\text{10}}_2 \) of the LLEE-1-chart \( C \) in Fig. 4.

That the LLEE-1-chart \( C \) in Fig. 4 is not 1-collapsible is suggested there by exhibiting a natural 1-bisimulation collapse that is not a LLEE-1-chart: the 1-bisimilar 1-chart \( C \), that results by ‘connecting through’ all incoming transitions at \( w_1 \) in \( C \) over to \( w_2 \). From \( C \), it is easy to intuit that the bisimulation collapse of \( C \) cannot be a LLEE-1-chart, either, and hence that \( C \) is not LLEE-preservingly collapsible. While that is only part of the (remaining) proof of Prop. 6.1, (i), it is also easy to check that the connect-\( w_2 \)-through-to-\( w_1 \) 1-chart \( C_j \) of \( C \) cannot be a LLEE-1-chart, either. For the proof that \( C^{\text{10}}_1 \) and \( C^{\text{10}}_2 \) are not LLEE-preservingly jointly minimizable it is crucial to realize that a function that maps \( w_1 \) to \( w_2 \), and \( w_2 \) to \( w_1 \) cannot be extended into a transfer function on \( C \). (That function, however, defines a ‘grounded’ 1-bisimulation slice on \( C \), see Def. 7.1 later, and Fig. 5).

For motivating concepts in the next sections we will use the simplified version \( C_5 \) in Fig. 5 of the LLEE-1-chart \( C \) in Fig. 4. We emphasize, however, that although \( C_5 \) is not collapsible, it is 1-collapsible. Hence \( C_5 \) does not witness statement (i), nor can it be used for showing (ii) in Prop. 6.1.

7 Twin-Crystals

The 1-chart \( C \) in Fig. 4 and Prop. 6.1 will turn out to be prototypical for strongly connected components in LLEE-1-charts that are ‘nearly collapsed’ but not LLEE-preservingly collapsible any further. By isolating a number of its properties from \( C \), and from its simplified version \( C_5 \), in Fig. 5, we define the central concept of ‘twin-crystal’. Each of the LLEE-1-charts \( C \) and \( C_5 \) consists of a single sce that is of
We write $w \looparrowright v$, $w \text{ loops-back-to } v$, if $v \xrightarrow{w} v'$ and $w \looparrowright v'$ holds with $n \in \mathbb{N}^+$ and such that $v$ is only encountered again at the end. We fix $\mathcal{C}^* w := \{ w \mid w \text{ loops-back-to } v \}$, the loops-back-to part of $v$.

The structure of the LLEE-1-charts $\mathcal{C}$ and $\mathcal{C}_t$ above can be described by the illustration in Fig. 6, together with the following eight properties that define when a vertex set $P$ in a LLEE-1-chart $\mathcal{C} = \langle V, A, 1, \rightarrow, \downarrow \rangle$ is the carrier of a twin-crystal (and that the sub-LTS induced by $P$ in $\mathcal{C}$ is a twin-crystal) with respect to vertices $\text{top, pivot } v \in P \subseteq V$, sets $P_1, P_2 \subseteq P$, a LLEE-witness $\mathcal{C}^* w$ of $\mathcal{C}$ with binary loops-back-to relation $\mathcal{C}_t$, and a non-empty set $E_2$ of transitions from $\text{top}$:

- (forth)$_R$ $\forall a \in A \forall w' \in V_1$ $\left( \begin{array}{c} w_1 \xrightarrow{a} w'_1 \land w'_1 \notin W_1 \\ w_2 \xrightarrow{a} w'_2 \land w'_2 \notin W_2 \end{array} \right)$,

where $W_1 := \text{dom}_a(B)$, and $W_2 := \text{cod}_a(B)$ are the active domain, and the active codomain of $B$, respectively.

**Lemma 7.2.** For every grounded bisimulation slice $B \subseteq V \times V$ on a 1-LTS $\mathcal{L} = \langle V, A, 1, \rightarrow, \downarrow \rangle$, the relation $\mathcal{L}^- := B \cup -$ is a 1-bisimulation on $\mathcal{L}$.

**Definition 7.3.** A local-transfer function on a 1-LTS $\mathcal{L} = \langle V, A, 1, \rightarrow, \downarrow \rangle$ is a partial function $\phi : V \rightarrow V$ whose graph $\langle \phi, \phi(a) \mid a \in V \rangle$ is a grounded 1-bisimulation slice on $\mathcal{L}$.

**Example 7.4.** Both of the functions $c_{\mathcal{L}}$ on the 1-charts $\mathcal{C}$ in Fig. 4, and $c_{\mathcal{L}}$ on $\mathcal{C}_t$ in Fig. 5 are local transfer functions: in particular in Fig. 5 it can be checked easily that $c_{\mathcal{L}}$ defines a grounded 1-bisimulation slice. Neither of these local transfer functions can be extended into a transfer function. For $\mathcal{C}_t$ this can be checked in Fig. 4: all pairs of the identity function have to be added in order to extend graph($c_{\mathcal{L}}$) into a 1-bisimulation, thereby violating functionality of the relation.

We will say that a 1-chart $\mathcal{C}$ (and respectively, a 1-LTS $\mathcal{L}$) is ‘near-collapsed’ if 1-bisimilarity on $\mathcal{C}$ (on $\mathcal{L}$) is the reflexive-symmetrical closure of the union of the graphs of finitely many local transfer functions on $\mathcal{L}$.

**Definition 7.5 (**) Let $\mathcal{L}$ be a 1-LTS with state set $V$, and let $P \subseteq V$ be a subset of the vertices of $\mathcal{L}$. We say that $\mathcal{L}$ is locally near-collapsed for $P$ if there are local transfer functions $\phi_1, \ldots, \phi_n : V \rightarrow V$ on $\mathcal{L}$ such that:

$$\mathcal{L} \cap (P \times P) \subseteq \mathcal{L}^-_{\mathcal{L}}, \quad \text{for } R = \bigcup_{i=1}^n \text{graph}($$ $\phi_i)$,

where $\mathcal{L}^-_{\mathcal{L}}$ means the reflexive-symmetric closure of $R$. We say that $\mathcal{L}$ is near-collapsed if $\mathcal{L}$ is near-collapsed for $V$.

A 1-chart $\mathcal{C}$ is (locally for $P$) near-collapsed if the 1-LTS $\mathcal{L}(\mathcal{C})$ underlying $\mathcal{C}$ is (locally for $P$) near-collapsed.

**Example 7.1.** Each of the 1-charts $\mathcal{C}$ in Fig. 4, and $\mathcal{C}_t$ in Fig. 5 contains only one pair of distinct 1-similar vertices. Since these vertices are related, respectively, by the appertaining transfer function $c_{\mathcal{L}}$, it follows that $\mathcal{C}$ and $\mathcal{C}_t$ are locally near-collapsed for the sets $P$ of all their vertices, respectively. Therefore $\mathcal{C}$ and $\mathcal{C}_t$ are near-collapsed.
suggests the symmetric nature of contains a 

is locally near-collapsed for 

(R1.2) (R3.4) (R3.4) (R1)

\[ \varnothing \]

with domain (with subkinds (dis-

\( \) with part subkinds (R1,1), (R1.2), (R2)), or (R3) (with subkinds (R3.1), (R3.2), (R3.3), (R3.4)) that are illustrated in Fig. 7. Note that

\[ cp_p(w) := \begin{cases} w & \text{if } w \in Am, \text{ and } w \text{ the 1-bisimilar counterpart of } w \text{ in } P, \\ \text{undefined} & \text{if } w \notin Am. \end{cases} \]

\[ \]

**Example 7.6.** Each of the 1-transition limited LLEE-1-charts \( C \) in Fig. 4 and \( C_2 \) in Fig. 5 consists of a single scc that is carrier of a twin-crystal. In the appertaining figures, the parts \( P_1 \) and \( P_2 \) are colored blue and green, respectively.

The picture in Fig. 6 suggests the symmetric nature of twin-crystals: if in the underlying LLEE-witness the loop-entry transitions from \( P_2 \) to \( P_1 \) (maximum level \( m \)) are relabeled into body transitions, and the body transitions from \( P_1 \) to \( P_2 \) into loop-entry transitions (level \( m \)), then a twin-crystal with permuted roles of \( piv \) and \( top \) arises. But for the completeness proof only the properties of twin-crystal shaped scc’s that are formulated by the two lemmas below are crucial.

\[ \]

**Lemma 7.7.** The counterpart function on the carrier \( P \) of a twin-crystal in a LLEE-1-chart \( C \) is a local transfer function.

**Lemma 7.8.** If \( P \) is the carrier of a twin-crystal in a LLEE-1-chart \( C \), then \( C \) is locally near-collapsed for \( P \).

8 Crystallization of LLEE-1-charts

For this central part of the proof we sketch how every weakly guarded LLEE-1-chart can be minimized under 1-bisimilarity far enough to obtain a LLEE-1-chart in ‘crystallized’ form. By that we mean that the resulting 1-bisimilar LLEE-1-chart is 1-collapsed apart from that some of its scc’s may be twin-crystals. A ‘groundedness’ clause in the definition will ensure that every crystalized 1-chart is also near-collapsed.

The minimization process we describe here is a refinement of the process for LLEE-charts (without 1-transitions) that was defined in \([13, 14] \). We first find that if a LLEE-1-chart \( C \) is not 1-collapsed, then \( C \) contains a ‘1-bisimilarity redundancy’ from one of three ‘reduced kinds’ (with subkinds). This is stated below by Lem. 8.1, which is a generalization of Prop. 6.4 in \([13, 14] \). Second, we argue that every reduced 1-bisimilarity redundancy can be eliminated LLEE-preservingly except if it belongs to a subkind which can be found in the not 1-collapsible LLEE-1-chart in Fig. 4. Third, we define a few further transformations for cutting scc’s into twin-crystals.

**Lemma 8.1** (kinds of reduced 1-bisimilarity redundancies). Let \( C \) be a 1-chart, and let \( \tilde{C} \) a 1-transition limited LLEE-witness of \( C \). Suppose that \( \tilde{C} \) is not a bisimulation collapse.

Then \( \tilde{C} \) contains a 1-bisimilarity redundancy \( \langle w_1, w_2 \rangle \) (distinct 1-bisimilar vertices \( w_1 \) and \( w_2 \) in \( C \)) that satisfies, with respect to \( \tilde{C} \), one of the position conditions (kinds) (R1) (with subkinds (R1.1), (R1.2)) (R2), or (R3) (with subkinds (R3.1), (R3.2), (R3.3), (R3.4)) that are illustrated in Fig. 7. Note that

**Figure 6.** Structure schema of a twin-crystal with carrier \( P \), with part \( P_1 \) of pivot vertex \( piv \), and part \( P_2 \) of top vertex \( top \), where \( P_2 \) is generated by the top entry transitions in \( E_2 \).
The vertices \( w_1 \) and \( w_2 \) are in the same scc for position kinds (R2) and (R3), but in different scc’s for position kind (R1).

**Definition 8.2.** We consider a 1-bisimilarity redundancy \( \langle w_1, w_2 \rangle \) in \( C \) under the assumptions of Lem. 8.1 on \( C \) and \( C \).

We say that \( \langle w_1, w_2 \rangle \) is reduced (with respect to \( C \)) if it is of one of the kinds (R1)–(R3) in Fig. 7. We say that \( \langle w_1, w_2 \rangle \) is simple if it is of (sub-)kind (R1), (R2), (R3.1), (R3.2), or (R3.3) in Fig. 7. We say that \( \langle w_1, w_2 \rangle \) is precrystalline if it is of subkind (R3.4) in Fig. 7. We say that \( \langle w_1, w_2 \rangle \) is crystalline if it is precrystalline, but neither of subkind (R3.4.1) nor (R3.4.2) in Fig. 8.

**Figure 7.** Reduced 1-bisimilarity redundancies, see Lem. 8.1. Upward dotted arrows: body 1-transition backlinks. Upward dashed double arrows: body transition paths of (direct) loops-back-to links. Dashed double arrows: paths of body transitions. Struck out red arrows: prohibited body-transitions and body-tr.-paths. Dashed links, bottom: assumed 1-bisimilarity.

**Example 8.3.** In the LLEE-1-charts \( C \) in Fig. 4, and \( C \) in Fig. 5, the pair \( \langle \text{abcd}, \text{abcd} \rangle \) of vertices forms a crystalline reduced 1-bisimilarity redundancy, since it is of kind (R3.4), hence precrystalline, but not of kind (R3.4.1) nor (R3.4.2).

From a LLEE-1-chart, every reduced 1-bisimilarity redundancy that is not crystalline can be eliminated LLEE-preservingly by which we mean that the result is a 1-bisimilar LLEE-1-chart. The transformations needed are adaptations of the connect-w1-through-to-w2 operation from [13, 14] in which the incoming transitions at vertex \( w_1 \) are redirected to a 1-bisimilar vertex \( w_2 \). Here this operation typically requires an unraveling step in which loop levels above \( w_1 \) that are reachable by 1-transitions are removed by similar transition redirections. In an example we illustrate the elimination of a reduced 1-bisimilarity redundancy of kind (R3.2) in Fig. 9.

**Lemma 8.4.** Every reduced 1-bisimilarity redundancy \( \langle w_1, w_2 \rangle \) can be eliminated LLEE-preservingly from a 1-transition limited LLEE-1-chart provided it is of either of the following kinds:

(i) \( \langle w_1, w_2 \rangle \) is simple, or
(ii) \( \langle w_1, w_2 \rangle \) is precrystalline, but not crystalline.

**Figure 8.** A precrystalline 1-bisim. redundancy \( \langle w_1, w_2 \rangle \) is crystalline if it is neither of form (R3.4.1) nor of form (R3.4.2).

**Figure 9.** Example for the LLEE-preserving elimination of a red. 1-bisim. red. of kind (R3.2) by redirecting transitions to 1-bisimilar targets. Here all proper action labels are the same.
In order to cut twin-crystals in scc’s we also need to safeguard that the joining loop vertices of crystalline reduced 1-bisimilarity redundancies are ‘parsimoniously insulated’ from above. The top vertices \( v \) of 1-bisimilarity redundancies of kind \((R3.3)\) and \((R3.4)\) are, due to the occurring 1-transitions, substrates of \( w_1 \) and \( w_2 \). Therefore any such vertex \( v \) can be insulated from above, that is, turned into a \( \ominus \)-maximal vertex, by redirecting all induced transitions from \( v \) into the loops-back-to part of \( v \) or below. In the example in Fig. 10, where all proper transitions have the same action label, in an insulation step (first step) the transition from \( v \) to \( u \) is redirected to the \( \ominus \)-bisimilar target \( \overrightarrow{w_2} \). In the arising 1-chart \( v \) is not yet parsimonious, because the loop-entry transition from \( v \) to \( \overrightarrow{w_1} \) can be redirected to \( \ominus \)-bisimilar target \( \overrightarrow{w_2} \) (second step), thereby making less use of the loops-back-to part of \( v \), and eliminating it (and permitting further minimization).

We now define ‘crystallized’ (LLEE-1)-charts as follows.

**Definition 8.5** (crystallized 1-chart (4)). Let \( C \) be a 1-chart.

We say that \( C \) is crystallized if there is a LLEE-witness \( \hat{C} \) of \( C \) such that the following four conditions hold:

- (cr-1) \( \hat{C} \) is a (finite) 1-chart with LLEE-1-lim, and specifically, \( C \) is 1-transition limited with respect to \( \hat{C} \).
- (cr-2) Every \( \hat{C} \)-reduced 1-bisim. redundancy is crystalline.
- (cr-3) Every crystalline \( \hat{C} \)-reduced 1-bisimilarity redundancy in \( C \) is parsimoniously insulated from above.
- (cr-4) Every carrier of an scc in \( C \) is grounded in \( \hat{C} \).

Then we also say that \( C \) is crystallized with respect to \( \hat{C} \).

The lemma below gathers properties of crystallized 1-charts. The subsequent proposition justifies the term ‘crystallized 1-chart’ by explaining the connection with twin-crystals.

**Lemma 8.6.** Every 1-chart \( C \) that is crystallized with respect to a LLEE-witness \( \hat{C} \) with loops-back-to relation \( \ominus \), satisfies:

(i) \( C \) is 1-collapsed apart from within scc’s, i.e. 1-collapsed for loops-back-to part of \( \ominus \)-maximal loop vertices of \( \hat{C} \).
(ii) \( C \) is 1-collapsed for every loops-back-to part of a loop vertex of \( \hat{C} \) that is not \( \ominus \)-maximal.

**Proposition 8.7** (crystallized \( \Rightarrow \) 1-coll./twin-crystal scc’s).

For every carrier \( P \) of a scc in a 1-chart \( C \), that is crystallized, either \( C \) is 1-collapsed for \( P \), or \( P \) is the carrier of a twin-crystal.

**Lemma 8.8** (4). Every crystallized 1-chart is near-collapsed.

By combining LLEE-preserving eliminations of precrystalline 1-bisimilarity redundancies, of parsimonious insulation of crystalline 1-bisimilarity redundancies, and of grounding of scc’s we are able to prove our main auxiliary statement.

**Theorem 8.9** (crystallization, near-collapse, (3),(4)). Every weakly guarded LLEE-1-chart \( \hat{C} \) can be transformed, together with a LLEE-witness \( \hat{C} \) of \( C \) into a 1-bisimilar LLEE-1-chart \( C' \) with 1-transition limited LLEE-witness \( \hat{C}' \) such that \( C \) is crystallized with respect to \( \hat{C}' \), and \( C' \) is near-collapsed.

### 9 Near-collapsed LLEE-1-charts have complete solutions

We show that near-collapsed LLEE-1-charts have complete solutions by linking local transfer functions, as in the definition of ‘near-collapsed’, to transfer functions in order to be able to use the transfer-property (T) for provable solutions.

Local-Transfer functions can be linked to transfer functions via the concept of the 1-LTS' \( E_W(C) \) that is the ‘elevation of a set \( W \) of vertices above’ a 1-LTS \( \tau = (V, A, 1, \rightarrow, 1) \), which is constructed as follows. The set of vertices of \( E_W(C) \) consists of two copies of its set \( V \) of vertices, the ‘ground
floor’ \( V \times \{0\} \), and the ‘first floor’ \( V \times \{1\} \). These two copies of the set of vertices of \( \mathcal{L} \) are linked by copies of the corresponding transitions of \( \mathcal{L} \) with the exception that proper \( \langle v_1, a, v_2 \rangle \) of \( \mathcal{L} \) do not give rise to a proper transition of the form \( \langle v_1, 1 \rangle, a, \langle v_2, 1 \rangle \) on the first floor if the vertex \( v_2 \) is not contained in \( W \). Those transitions get redirected as transitions \( \langle \langle v_1, 1 \rangle, a, \langle v_2, 0 \rangle \rangle \) to target the corresponding copy \( \langle v_2, 0 \rangle \) of \( v_2 \) on the ground floor. Note that such redirections from the first floor to the ground floor do not happen for 1-transitions. The sub-1-LTS of \( E_W(\mathcal{L}) \) that consists of all transitions between vertices on the ground floor is an exact copy of the original 1-LTS \( \mathcal{L} \). Yet within the elevation \( E_W(\mathcal{L}) \) of \( W \) above \( \mathcal{L} \), a number of vertices on the ground floor will have additional incoming proper-action transitions from vertices on the first floor.

**Example 9.1.** Fig. 11 contains two copies of the elevation \( E_W(\mathcal{C}_z) \) of the set \( W := \{abed, abed\} \) of vertices of the 1-chart \( \mathcal{C}_z \), in Fig. 5 above \( \mathcal{C}_q \).

Indeed, Fig. 11 also shows how the local transfer function \( cp_p \) on 1-chart \( \mathcal{C}_z \), in Fig. 5 can be lifted to a transfer function \( \varphi^{(1)} \) on the elevation \( E_W(\mathcal{C}_z) \) of \( W \) above \( \mathcal{C}_z \): namely by defining \( \varphi^{(1)} \) as \( cp_p \) on the first floor, and as the identity function on the ground floor of \( E_W(\mathcal{C}_z) \).

In general the following statement holds, which is a generalization to 1-charts of Prop. 2.4 in [11]. Every local transfer function \( \phi : V \to V \) on a 1-LTS \( \mathcal{L} \) with \( W := \text{dom}(\phi) \cap \text{ran}(\phi) \) lifts to a transfer function \( \varphi^{(1)} \) on the elevation \( E_W(\mathcal{L}) \) of \( W \) over \( \mathcal{L} \) via the projection transfer function \( \pi_1 \) from \( E_W(\mathcal{L}) \) to \( \mathcal{L} \) such that the diagram below commutes for vertices on the first floor of \( E_W(\mathcal{L}) \):

\[
\begin{align*}
\text{L} & \xrightarrow{\phi} \text{L} \quad \text{for all } w \in \text{dom}(\phi): \\
\pi_1 & \quad \pi_1 \quad (\pi_1 \circ \phi)(\langle w, 1 \rangle) \\
E_W(\mathcal{L}) & \xrightarrow{\varphi^{(1)}} E_W(\mathcal{L}) \\
\end{align*}
\]

(1)

Together with preservation of LLEE for elevations, the possibility to lift local transfer functions to transfer functions on elevations facilitates us to use provable solutions under transfer functions between LLEE-1-LTSs for proving invariance of provable solutions under local transfer functions on LLEE-1-LTSs. Then we can use this fact to show complete solvability of near-collapsed LLEE-1-charts.

**Lemma 9.1.** Let \( \phi : V \to V \) be a local transfer function on a w.g. LLEE-1-LTS \( \mathcal{L} = \langle V, A, 1, \rightarrow, \downarrow \rangle \). Then every Mil-provable solution \( s \) of \( \mathcal{L} \) is Mil-provably invariant under \( \phi \):

\[
s(w) =_{\text{Mil}} s(\phi(w)) \quad (\text{for all } w \in \text{dom}(\phi)) .
\]

(2)

**Proof (Sketch).** We use the diagram in (1), and that \( E_W(\mathcal{L}) \) is also a LLEE-1-LTS. Since \( \pi_1 \) and \( \varphi^{(1)} \circ \pi_1 \) are transfer functions, by Lem. 4.6 \( s \circ \pi_1 \) and \( s \circ \pi_1 \circ \varphi^{(1)} \) are Mil-provably equal. Then by using we diagram commutativity on the first floor in (1) we get: \( s(w) = (s \circ \pi_1)(\langle w, 1 \rangle) =_{\text{Mil}} (s \circ \pi_1 \circ \varphi^{(1)})(\langle w, 1 \rangle) = (s \circ \phi \circ \pi_1)(\langle w, 1 \rangle) = s(\phi(w)) \), for all \( w \in \text{dom}(\phi) \). In this way we have obtained (2).

**Lemma 9.2 (6).** Every w.g. LLEE-1-chart that is near-collapsed has a Mil-complete Mil-lee-chart that is bisimilar to a crystallized, and hence to a near-collapsed, LLEE-1-chart.

Since the size of a crystallized 1-chart is bounded by at most double the size of an 1-bisimulation collapse (as every vertex is 1-bisimilar to at most one other vertex in twin-crystals, and crystallized 1-charts), this characterization raises the hopes for a polynomial algorithm for recognizing expressibility of finite process graphs. Such a recognition algorithm would substantially improve on the superexponential algorithm for deciding expressibility in [1].

**Acknowledgments**

Wan Fokkink introduced me to Milner’s questions from [16] in 2005. This facilitated my work on them, and led me to the decision result [1] with Jos Baeten and Flavio Corradini. In 2015, Wan suggested to tackle the problem together by looking for minimization strategies for star expressions that he had pioneered in [5–7]. For that it led me from the structure constraints of LEE and layered LEE for process graphs (generalizing ‘well-behaved specifications’ of processes by Flavio Corradini) to an idea for tackling Milner’s axiomatization question in full generality, weekly meetings with Wan in 2015–2018 were crucial. Together with Wan’s research visit to GSSI in 2019, these meetings led us to the completeness result [14] for the tailored restriction BBP of Milner’s system to ‘1-free star expressions’. While I continued further on this path by myself for the past two years, my work with Wan has formed the firm basis of this work.

I am very thankful to Luca Aceto for comments on parts of this submission and previous ones, and for giving me the chance to complete this work within the PRIN project IT MATTERS - Methods and Tools for Trustworthy Smart Systems (project ID: 2017FTXR75_005).

I want to thank Emilio Tuosto very much for discussing aspects of my work with him, for the chance to speak about it in the FM/SE meetings he organized at GSSI, and also for quick help, repeatedly.
For their help during the TAPS-typesetting process of the LICS conference article, I want to thank Vincent van Oostrom for crucial advice about how to proceed in order to overcome an impasse, and Alessandro Aloisio for an idea that helped me locate an unspecific, wrongly located typesetting error.

Finally I want to thank the anonymous reviewers of the LICS submission for their interest and thorough reading, and a conceptualizing summary. Their comments signaled the need to make details in Sect. 5 better accessible, and provided me with an idea to visually improve Fig. 3 by using a ‘pyramid’ structure.

References

[1] Jos Baeten, Flavio Corradini, and Clemens Grabmayer. 2007. A Characterization of Regular Expressions Under Bisimulation. J. ACM 54, 2 (2007), 1–28. https://doi.org/10.1145/1219952.1219994
[2] J. C. M. Baeten and F. Corradini. 2005. Regular Expressions in Process Algebra. In Proceedings of the 20th Annual IEEE Symposium on Logic in Computer Science (LICS ’05). IEEE Computer Society, USA, 12–19. https://doi.org/10.1109/LICS.2005.43
[3] Doeko Bosscher. 1997. Grammars Modulo Bisimulation. Ph. D. Dissertation. University of Amsterdam.
[4] Flavio Corradini, Rocco De Nicola, and Anna Labella. 2002. An Equational Axiomatization of Bisimulation over Regular Expressions. Journal of Logic and Computation 12, 2 (2002), 301–320. https://doi.org/10.1093/logcom/12.2.301
[5] Wan Fokkink. 1996. An Axiomatization for the Terminal Cycle. Technical Report. Logic Group Preprint Series. Vol. 167, Utrecht University.
[6] Wan Fokkink. 1997. Axiomatizations for the Perpetual Loop in Process Algebra. In Proc. ICALP’97 (LNCS, Vol. 1256). Springer, Berlin, Heidelberg, 571–581. https://doi.org/10.1007/3-540-63165-8_212
[7] Wan Fokkink and Hans Zantema. 1994. Basic Process Algebra with Iteration: Completeness of its Equational Axioms. Comput. J. 37, 4 (1994), 259–267. https://doi.org/10.1093/comjnl/37.4.259
[8] Clemens Grabmayer. 2020. Structure-Constrained Process Graphs for the Process Semantics of Regular Expressions. Technical Report. arxiv.org. arXiv:2012.10869 [cs.LO] Extended report for [12].
[9] Clemens Grabmayer. 2021. A Coinductive Version of Milner’s Proof System for Regular Expressions Modulo Bisimilarity. Technical Report arXiv:2108.13104. arxiv.org. arXiv:2108.13104 [cs.LO] Extended report for [10].
[10] Clemens Grabmayer. 2021. A Coinductive Version of Milner’s Proof System for Regular Expressions Modulo Bisimilarity. In 9th Conference on Algebra and Coalgebra in Computer Science (CALCO 2021) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 211). Fabio Gadducci and Alexandra Silva (Eds.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 16:1-16:23. https://doi.org/10.4230/LIPIcs.CALCO.2021.16 Extended report see [9].
[11] Clemens Grabmayer. 2021. Bisimulation Slices and Transfer Functions. Technical report, Reykjavik University. Abstract for the 32nd Nordic Workshop on Programming Theory (NWPT 2021), http://icetcs.ru/nwpt21/abstracts/paper5.pdf.
[12] Clemens Grabmayer. 2021. Structure-Constrained Process Graphs for the Process Semantics of Regular Expressions. In Proceedings 11th International Workshop on Computing with Terms and Graphs, Online, 5th July 2020 (Electronic Proceedings in Theoretical Computer Science, Vol. 334). Patrick Bahr (Ed.). Open Publishing Association, 111 Cooper Street, Waterloo, NSW 2017, Australia, 29–45. https://doi.org/10.4204/EPTCS.334.3 Extended report see [8].
A Appendix

A.1 Supplements for Section 2

A.1.1 Obstacle for the bisimulation-chart proof strategy (more detail).

In Sect. 2, starting on page 2, we argued that a bisimulation-chart proof strategy that would operate in analogy with Salomaa’s proof in [17] of completeness of $F_1$ for language equivalence of regular expressions does not work for showing completeness of Milner’s system with respect to the process interpretation. In particular, we argued that an extraction procedure $EP$ of a solution from any given guarded linear system of recursion equations as in Salomaa’s proof is not possible, not even from systems that are solvable. The reason is as follows. An extraction procedure $EP$ that were completely analogous to the one used by Salomaa would have the following two properties:

- (EP-1) $EP$ extracts a solution from any guarded linear system $S$ of recursion equations.
- (EP-2) $EP$ proceeds ‘data-obliviously’ by mechanically combining the actions in the equations of a recursion system $S$ according to a fixed way of traversing it that, while copying and transporting specific letters from the equations over to the solution, never compares different letters and never takes decisions on the basis of the specific letter at hand. In particular $EP$ also never takes a decision that the system would not be solvable, but always produces a result.

Now property (EP-1) cannot be obtained, because as Milner showed, see Example A.1 below, there are guarded linear systems of recursion equations that are unsolvable by star expressions in the process semantics. While this shows that an extraction procedure $EP$ with (EP-1) and (EP-2) is impossible, it leaves open the possibility of an extraction procedure $EP'$ that satisfies (EP-2) except for that it may sometimes not terminate (see (EP’-2) below), and the restriction (EP’-1) of (EP-1) to solvable recursion equations (see (EP’-1), (a)) together with a soundness condition (see (EP’-1), (b)): 

- (EP’-1) (a) $EP'$ extracts a solution from any guarded linear system $S$ of recursion equations that is solvable.
- (b) Whenever $EP'$ obtains a result for a guarded linear system $S$ of recursion equations, then that is a solution of $S$.

(EP’-2) $EP'$ proceeds data-obliviously in the same way as (EP-2) requires for $EP$. However, the extraction process is not required to be terminating.

It turns out that such a restricted extraction procedure $EP'$ is not possible, either. The reason is that there are unsolvable specifications $S_{uo}$ that are ‘data-obliviously’ the same (that is, the underlying process graph has the same structure when action names are ignored) with a (respectively) corresponding solvable specification $S_{so}$. Then, in order to safeguard solvability for $S_{uo}$, see (EP’-1), (a), $EP'$ would need to produce a result for both $S_{uo}$ and $S_{so}$, but this then leads to a contradiction with (EP’-1), (b), because the result extracted from $S_{uo}$ cannot be a solution, as $S_{uo}$ is unsolvable. We give two examples of such pairs of unsolvable and solvable specifications that are data-obliviously the same below in Example A.1 and Example A.2, where the specifications in Example A.2 are solvable counterparts of unsolvable specifications in Example A.1.

Such pairs of unsolvable and solvable process graphs that are ‘data-obliviously’ the same are not only artificial counterexamples, but can also arise from bisimulation charts that link bisimilar expressible graphs, see Example 4.1 in [13, 14]. (The such a soluble bisimulation chart is paired with a chart that arises from it by relabeling each transition by a different action.)

Example A.1 (not expressible process graphs, unsolvable recursive specifications). As mentioned in Sect. 2 on page 2, Milner noticed that guarded systems of recursion equations cannot always be solved by star expressions under the process semantics:

"[In] contrast with the case for languages—an arbitrary system of guarded equations in [star]-behaviours cannot in general be solved in star expressions" [16].

In fact, Milner showed in [16] that the linear specification $S(G_1)$ defined by the process graph $G_1$ below does not have a star expression solution modulo bisimilarity. He conjectured that that holds also for the easier specification $S(G_2)$ defined by the process graph $G_2$ below. That that is indeed the case was confirmed and proved later by Boscher [3].

![Example A.1 Diagram](image)

Here the start vertex of a process graph is again highlighted by a brown arrow $\rightarrow$, and a vertex $v$ with immediate termination is emphasized in brown as $\emptyset$ including a boldface ring.
G_1 and G_2 are finite process graphs that are not bisimilar to the process interpretation of any star expression. In this sense, G_1 and G_2 are not expressible by a regular expression (under the process semantics). This sets the process semantics of regular expressions apart from the standard language semantics, with respect to which every language that is accepted by a finite-state automaton is the interpretation of some regular expression.

Furthermore it is easy to see that both of G_1 and G_2 do not satisfy LEE. Namely, both of these process graphs do not contain loop subcharts, but each of them represents an infinite behavior. Therefore the loop elimination procedure stops immediately on either of them, unsuccessfully.

**Example A.2.** While the specifications S(G_1) and S(G_2) in Example A.1 are not solvable by star expressions in the process semantics, this situation changes drastically if all actions in S(G_1) and S(G_2) are replaced by a single action a. Then we obtain the following process graphs G_1^{(a)} and G_2^{(a)} with appertaining specifications S(G_1^{(a)}) and S(G_2^{(a)}):

![Process Graphs G_1^{(a)} and G_2^{(a)}]

These specifications are solvable by setting X_1 := X_2 := X_3 := a^* \cdot 0 in S(G_1^{(a)}), and by setting Y_1 := Y_2 := a in S(G_2^{(a)}), because the following identities are provable in Milner’s system Mil (which in any case is sound for the process semantics):

\[
\begin{align*}
 a^* \cdot 0 &= \text{Mil} \cdot 1 + a \cdot a^* \cdot 0 = \text{Mil} \cdot 0 + a \cdot (a^* \cdot 0), \\
 a^* &= \text{Mil} \cdot 1 + a \cdot a^*
\end{align*}
\]

From these Mil-provable identities the Mil-provable correctness conditions for these settings follow directly, for example for X_1 in S(G_1^{(a)}), and for Y_1 in S(G_2^{(a)}):

\[
\begin{align*}
 X_1 &= a^* \cdot 0 = \text{Mil} \cdot a \cdot (a^* \cdot 0) + a \cdot (a^* \cdot 0) = a \cdot X_2 + a \cdot X_3, \\
 Y_1 &= a^* = \text{Mil} \cdot 1 + a \cdot a^* = 1 + a \cdot Y_2.
\end{align*}
\]

These Mil-provable identities show, together with the analogous ones for X_2, X_3, and Y_2 that the settings X_1 := X_2 := X_3 := a^* \cdot 0 and Y_1 := Y_2 := a^* define solutions of S(G_1^{(a)}), and S(G_2^{(a)}), respectively, because Mil-provable identities are also identical with respect to the process semantics (as Milner’s system is sound for the process semantics).

The crucial reason why we have obtained solvable specifications S(G_1^{(a)}) and S(G_2^{(a)}) from unsolvable specifications S(G_1) and S(G_2), respectively, is that the underlying process graphs G_1^{(a)} and G_2^{(a)} are not bisimulation collapses, in contrast with the process graphs G_1 and G_2 from which they arose by renaming all actions to a single one. Indeed the bisimulation collapses G_10 of G_1, and G_20 of G_2 are of particularly easy form that are obviously solvable by the star expressions a^* \cdot 0, and a^*, respectively. These solutions can be transferred backwards over the functional bisimulations G_1^{(a)} \sqsupseteq G_1^{(a)} \sqsupseteq G_10 \sqsupseteq G_10^{(a)} (indicated by magenta links in the pictures below) according to Lem. 3.8, (i).

![Functional Bisimulations G_1^{(a)} \sqsupseteq G_1^{(a)} \sqsupseteq G_10 \sqsupseteq G_10^{(a)}]

in order to obtain the Mil-provable solutions described above.

This example witnesses the result from [13, 14] that a process graph G without 1-transitions is expressible by a 1-free star expression in the process semantics if the bisimulation collapse of G has the property LEE. Here the bisimulation collapses G_10^{(a)} and G_20^{(a)} satisfy LEE, although the process graphs from which they arise by collapse, G_1^{(a)} and G_2^{(a)}, do not satisfy LEE.