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

Crystallization: Near-Collapsing Process Graph Interpretations of Regular Expressions

Clemens Grabmayer (Department of Computer Science, Gran Sasso Science Institute, Viale F. Crispi, 7, 67100 L'Aquila AQ, Italy)

Abstract

We report on a lengthy completeness proof for Robin Milner's proof system Mil (1984) for bisimilarity of regular expressions in the process semantics. Central for our proof are the recognitions:

1. Process graphs with 1-transitions (1-charts) and the loop existence/elimination property LLEE are **not** closed under bisimilation collapse,
2. Such process graphs can be '**crystallized**' to 'near-collapsed' 1-charts with some strongly connected components of 'twin-crystal' form.

The Process Semantics of Regular Expressions

Milner (1984) introduced a process semantics for regular expressions: the interpretation of 0 is *deadlock*, of 1 is an *empty step to termination*, letters a are *atomic actions*, the operators $+$ and $\cdot$ stand for *choice* and *concatenation* of processes, and unary Kleene star $(\cdot)^*$ represents *(unbounded) iteration*. Formally, Milner defined chart (finite process graph) interpretations $\mathcal{C}(e)$ of regular expressions e.

Milner's Proof System

As axiomatization of the relation $e_1 =_{\boldsymbol{P}} e_2$ on regular expressions e_1 and e_2 defined by $\mathcal{C}(e_1) \leftrightarrow \mathcal{C}(e_2)$ (as bisimilarity $\leftrightarrow$ of chart interpretations), Milner asked whether the following system Mil is complete:

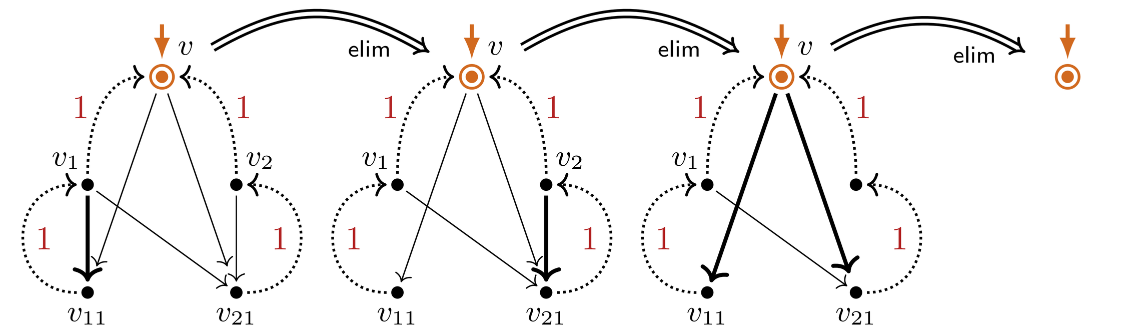
(A1) $e + (f + g) = (e + f) + g$	(A7) $e = 1 \cdot e$
(A2) $e + 0 = e$	(A8) $e = e \cdot 1$
(A3) $e + f = f + e$	(A9) $0 = 0 \cdot e$
(A4) $e + e = e$	(A10) $e^* = 1 + e \cdot e^*$
(A5) $e \cdot (f \cdot g) = (e \cdot f) \cdot g$	(A11) $e^* = (1 + e)^*$
(A6) $(e + f) \cdot g = e \cdot g + f \cdot g$	

$$\frac{e = f \cdot e + g}{e = f^* \cdot g}\,\text{RSP}^* \quad \text{(if } f \text{ does not terminate immediately)}$$

This system is a variation of Salomaa's complete axiom system (1966) for language equality of regular expressions, missing left-distributivity $e \cdot (f + g) = e \cdot f + e \cdot g$ and $e \cdot 0 = 0$, which are unsound here.

Loop Existence and Elimination

The process semantics is incomplete: not every finite process graph is *expressible by* (=bisimilar to the interpretation of) a regular expression. A sufficient condition for expressibility is the *(layered) loop existence and elimination property* LLEE. It is defined via elimination of 'loops' (loop subcharts):



LLEE holds if a graph without infinite behavior can be obtained. Important features of LLEE:

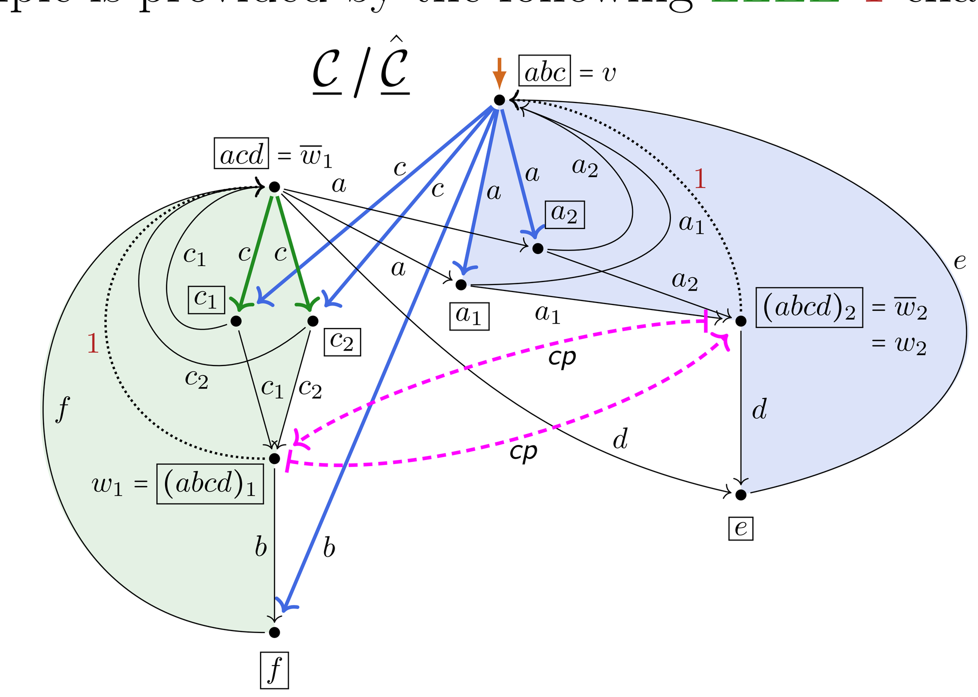
(US) Every guarded LLEE-1-chart (chart, maybe 1-transitions, with LLEE) is uniquely Mil-provably solvable modulo provability in Mil (CALCO 2021).

(IV) The chart interpretation $\mathcal{C}(e)$ of a regular expression e can always be expanded under bisimilarity to a LLEE-1-chart $\mathcal{C}(e)$ (TERMGRAPH 2020).

($\mathrm{C}_{\not 1}$) LLEE-charts (without 1-transitions) are preserved by bisimulation collapse (G/Fokkink, LICS'20).

LLEE-preserving Collapse Fails

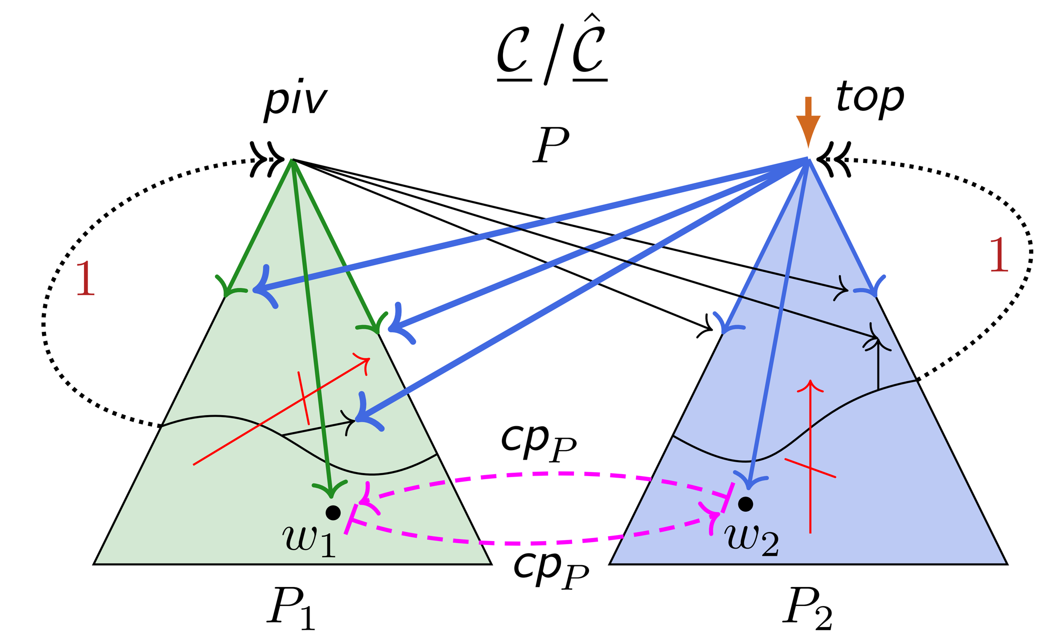
LLEE-1-charts with 1-transitions, however, are **not** preserved under bisimulation collapse. A counterexample is provided by the following LLEE-1-chart $\mathcal{C}$:



Identifying the bisimilar vertices w_1 and w_2 yields a chart for which LLEE fails. Also, the subcharts of $\mathcal{C}$ that are rooted at w_1 and w_2 are **not** *LLEE-preservingly jointly minimizable under bisimilarity.*

Twin-Crystals

The counterexample to LLEE-preserving collapse is symmetric, and its structure can be abstracted as:



It is a LLEE-1-chart with a single scc (strongly connected component) P that consists of a *pivot part* P_1 below *pivot vertex* **piv**, and a *top part* P_2 below *top vertex* **top**. P_1 and P_2 are connected only via transitions from **piv** and from **top**. While both P_1 and P_2 are collapsed, P contains *bisimilarity redundancies* (= distinct bisimilar vertices) such as $\{w_1, w_2\}$ that are linked by a self-inverse counterpart function cp_P. We call such an scc a *twin-crystal*. We have:

(CC) Every Mil-provable solution of a twin-crystal gives rise to a Mil-provable solution of its bisimulation collapse (which often is not a LLEE-1-chart).

Crystallization of LLEE-1-charts

By *crystallization* of a LLEE-1-chart $\mathcal{C}$ we mean:

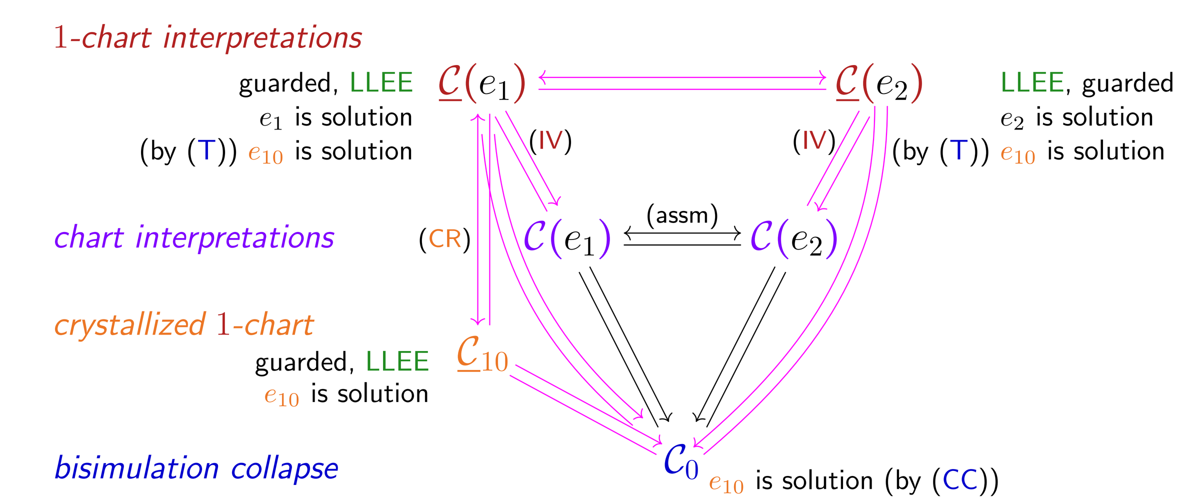
- a process of minimization of $\mathcal{C}$ under bisimilarity by steps that **eliminate most** (all but crystalline) bisimilarity redundancies $\{w_1, w_2\}$, roughly by redirecting transitions that target w_1 over to w_2;
- hereby only '**reduced**' bisimilarity redundancies can be eliminated LLEE-preservingly, which exist whenever a LLEE-1-chart is **not** collapsed;
- the result is a *crystallized* LLEE-1-chart that is bisimilar to $\mathcal{C}$, and collapsed **apart from** within some its scc's that are twin-crystals.

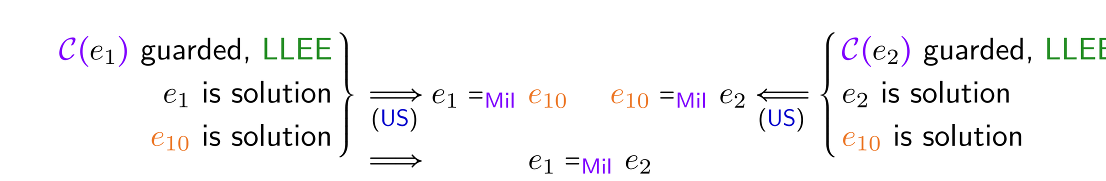
The crystallization process facilitates to show:

(CR) From every LLEE-1-chart a bisimilar crystallized LLEE-1-chart can be obtained.

Completeness Proof

Let $\mathcal{C}(e_1) \leftrightarrow \mathcal{C}(e_2)$ be bisimilar chart interpretations of regular expressions e_1 and e_2. To secure LLEE, $\mathcal{C}(e_1)$ and $\mathcal{C}(e_2)$ are expanded to their 1-chart interpretations $\mathcal{C}(e_1)$ and $\mathcal{C}(e_2)$. One of them, say $\mathcal{C}(e_1)$, is crystallized to $\mathcal{C}_{10}$. All (1-)charts are linked by (1-)bisimulations to their bisimulation collapse $\mathcal{C}_0$.





From $\mathcal{C}_{10}$ a provable solution e_{10} can be extracted due to LLEE, transferred (T) to the collapse $\mathcal{C}_0$, and then to $\mathcal{C}(e_1)$ and $\mathcal{C}(e_2)$. On the LLEE-1-charts $\mathcal{C}(e_1)$ and $\mathcal{C}(e_2)$, e_{10} can be proved equal to the solutions e_1 and e_2 there, respectively. By transitivity, $e_1 =_{\mathsf{Mil}} e_2$ (provability of $e_1 = e_2$ in Mil) follows.

Theorem. *Milner's system* Mil *is complete:* $e_1 =_{\boldsymbol{P}} e_2$ *implies* $e_1 =_{\mathsf{Mil}} e_2$*, for reg. expr's* e_1, e_2*.*

Next Steps and Projects

- Monograph project: proof in fine-grained details.
- Build an animation tool for crystallization.
- Apply crystallization to find an efficient algorithm for expressibility of finite process graphs by a regular expression modulo bisimilarity.

Contact

clemens.grabmayer@gssi.it

GRAN SASSO SCIENCE INSTITUTE
SCHOOL OF ADVANCED STUDIES
Scuola Universitaria Superiore