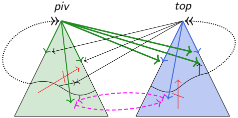
Milner's Proof System for Regular Expressions Modulo Bisimilarity is Complete

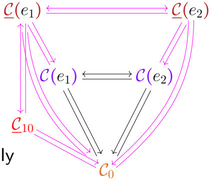
Crystallization: Near-Collapsing Process Graph Interpretations of Regular Expressions



Clemens Grabmayer

GRAN SASSO SCIENCE INSTITUTE
SCHOOL OF ADVANCED STUDIES
Scuola Universitaria Superiore

Department of Computer Science, GSSI, L'Aquila, Italy



Computer Science Seminar

GSSI

July 27, 2022

Process semantics of regular expressions $[\![\cdot]\!]_P$ *(Milner, 1984)*

$\mathbf{0} \xmapsto{[\![\cdot]\!]_P}$ deadlock δ, no termination

$\mathbf{1} \xmapsto{[\![\cdot]\!]_P}$ empty-step process ϵ, then terminate

$a \xmapsto{[\![\cdot]\!]_P}$ atomic action a, then terminate

$e + f \xmapsto{[\![\cdot]\!]_P}$ (*choice*) execute $[\![e]\!]_P$ or $[\![f]\!]_P$

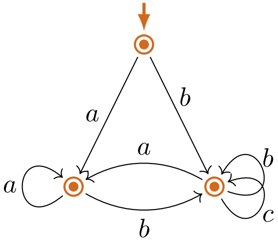
$e \cdot f \xmapsto{[\![\cdot]\!]_P}$ (*sequentialization*) execute $[\![e]\!]_P$, then $[\![f]\!]_P$

$e^* \xmapsto{[\![\cdot]\!]_P}$ (*iteration*) repeat (terminate or execute $[\![e]\!]_P$)

$[\![e]\!]_P := [\mathcal{C}(e)]_{\leftrightarrow}$ (bisimilarity equivalence class of chart $\mathcal{C}(e)$)

Process semantics $[\![\cdot]\!]_P$ (examples, bisimulation collapse)

chart (Milner)



$\mathcal{C}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$

Process semantics $[\![\cdot]\!]_P$ (examples, bisimulation collapse)

1-chart

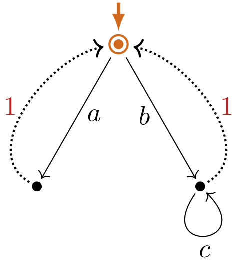
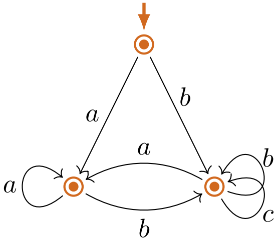


chart (Milner)



$\underline{\mathcal{C}}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$ $\mathcal{C}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$

Process semantics $[\![\cdot]\!]_P$ (examples, bisimulation collapse)

1-chart

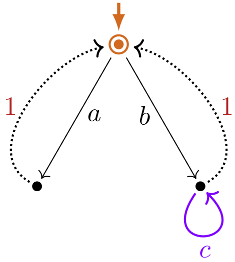
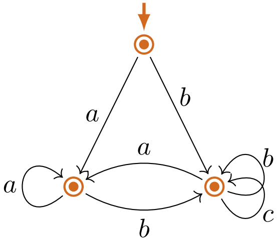


chart (Milner)



$\underline{\mathcal{C}}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$ $\mathcal{C}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$

Process semantics $[\![\cdot]\!]_P$ (examples, bisimulation collapse)

1-chart

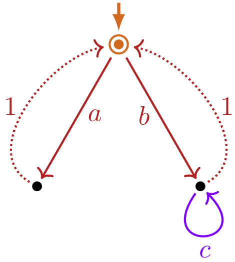
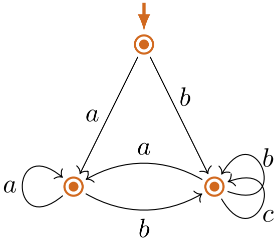


chart (Milner)



$\underline{\mathcal{C}}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$ $\mathcal{C}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$

Process semantics $[\![\cdot]\!]_P$ (examples, bisimulation collapse)

1-chart

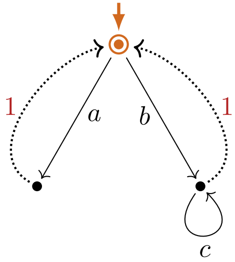
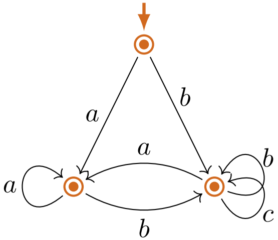
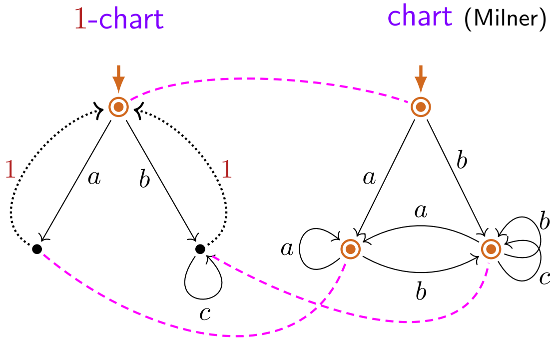


chart (Milner)



$\underline{\mathcal{C}}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$ $\mathcal{C}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$

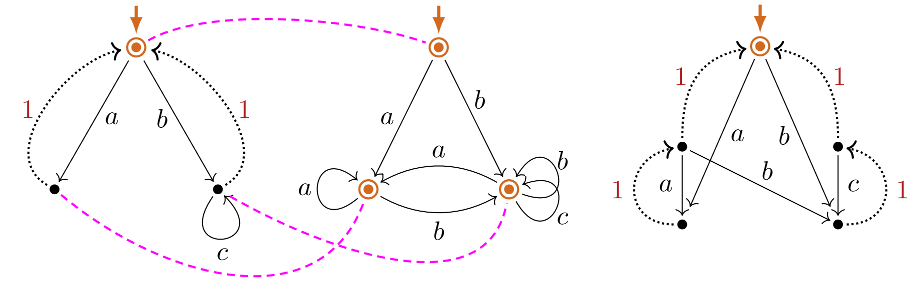
Process semantics $[\![\cdot]\!]_P$ (examples, bisimulation collapse)



$\underline{\mathcal{C}}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$ $\mathcal{C}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$

Process semantics $[\![\cdot]\!]_P$ (examples, bisimulation collapse)

1-chart | chart (Milner) | 1-chart

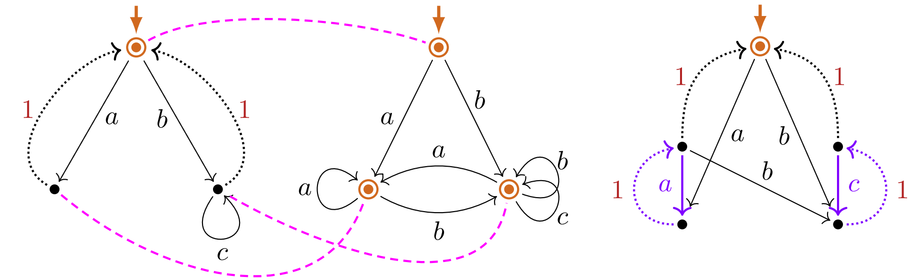


$\underline{\mathcal{C}}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$ $\mathcal{C}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$

$$\underline{\mathcal{C}}\Big(\big(a \cdot (1 \cdot ((a \cdot 1)^* \cdot 1)) \cdot b \cdot ((1 \cdot (c \cdot 1)^*)) \cdot 1 + b \cdot ((1 \cdot (c \cdot 1)^*)) \cdot 1\big)^*\Big)$$

Process semantics $[\![\cdot]\!]_P$ (examples, bisimulation collapse)

1-chart | chart (Milner) | 1-chart

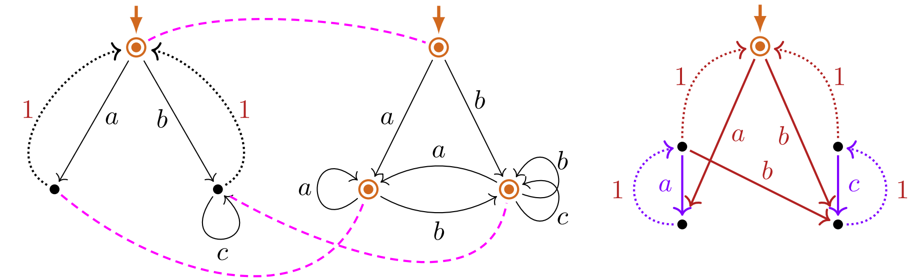


$\underline{\mathcal{C}}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$ $\mathcal{C}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$

$$\underline{\mathcal{C}}\Big(\big(a \cdot (1 \cdot ((a \cdot 1)^* \cdot 1)) \cdot b \cdot ((1 \cdot (c \cdot 1)^*)) \cdot 1 + b \cdot ((1 \cdot (c \cdot 1)^*)) \cdot 1\big)^*\Big)$$

Process semantics $[\![\cdot]\!]_P$ (examples, bisimulation collapse)

1-chart | chart (Milner) | 1-chart

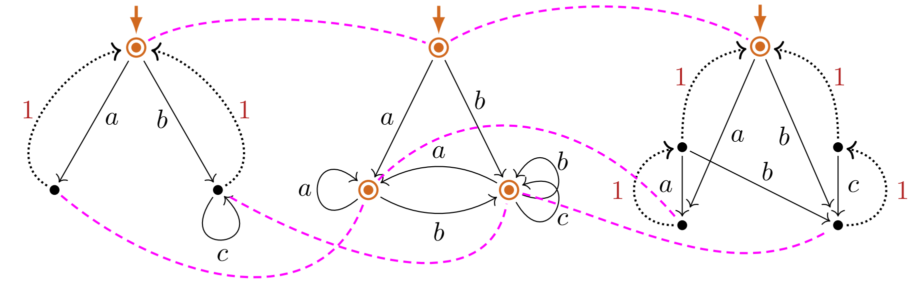


$\underline{\mathcal{C}}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$

$\mathcal{C}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$

$$\underline{\mathcal{C}}\Big(\big(a \cdot (1 \cdot ((a \cdot 1)^* \cdot 1)) \cdot b \cdot ((1 \cdot (c \cdot 1)^*)) \cdot 1 + b \cdot ((1 \cdot (c \cdot 1)^*)) \cdot 1\big)^*\Big)$$

Process semantics $[\![\cdot]\!]_P$ (examples, bisimulation collapse)



$\underline{\mathcal{C}}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$ $\mathcal{C}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$ $\underline{\mathcal{C}}\Big(\big(a \cdot (1 \cdot ((a \cdot 1)^* \cdot 1)) \cdot b \cdot ((1 \cdot (c \cdot 1)^*)) \cdot 1 + b \cdot ((1 \cdot (c \cdot 1)^*)) \cdot 1\big)^*\Big)$

Process semantics $[\![\cdot]\!]_P$ (examples, bisimulation collapse)

1-chart

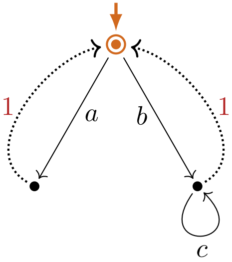
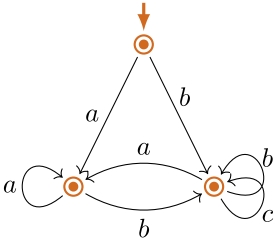
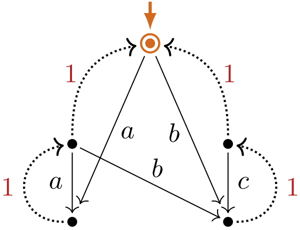


chart (Milner)



1-chart



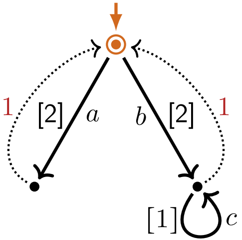
$\underline{\mathcal{C}}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$

$\mathcal{C}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$

$$\underline{\mathcal{C}}\Big(\big(a \cdot (1 \cdot ((a \cdot 1)^* \cdot 1)) \cdot b \cdot ((1 \cdot (c \cdot 1)^*)) \cdot 1 + b \cdot ((1 \cdot (c \cdot 1)^*)) \cdot 1\big)^*\Big)$$

Process semantics $[\![\cdot]\!]_P$ (examples, bisimulation collapse)

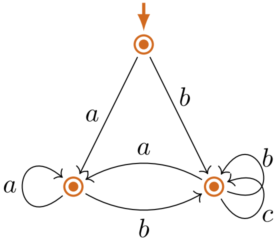
1-chart



LLEE

$\underline{\mathcal{C}}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$

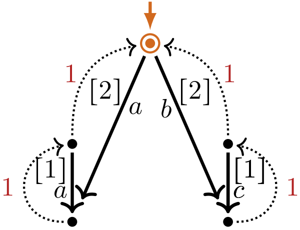
chart (Milner)



~~LLEE~~

$\mathcal{C}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$

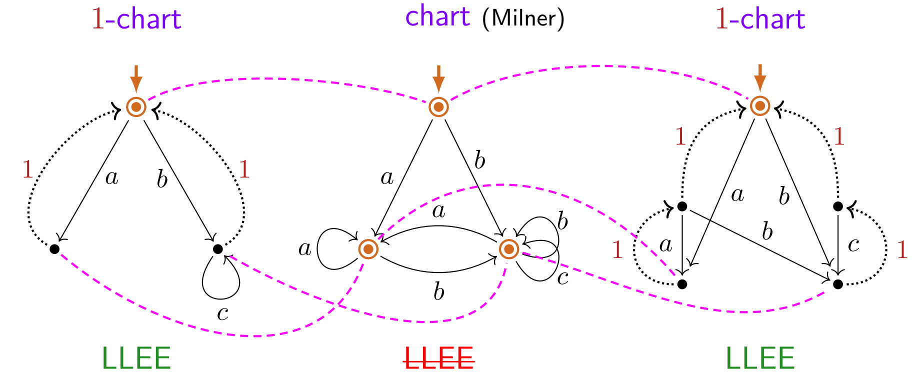
1-chart



LLEE

$$\underline{\mathcal{C}}\Big(\big(a \cdot (1 \cdot ((a \cdot 1)^* \cdot 1)) \cdot b \cdot ((1 \cdot (c \cdot 1)^*)) \cdot 1 + b \cdot ((1 \cdot (c \cdot 1)^*)) \cdot 1\big)^*\Big)$$

Process semantics $[\![\cdot]\!]_P$ (examples, bisimulation collapse)

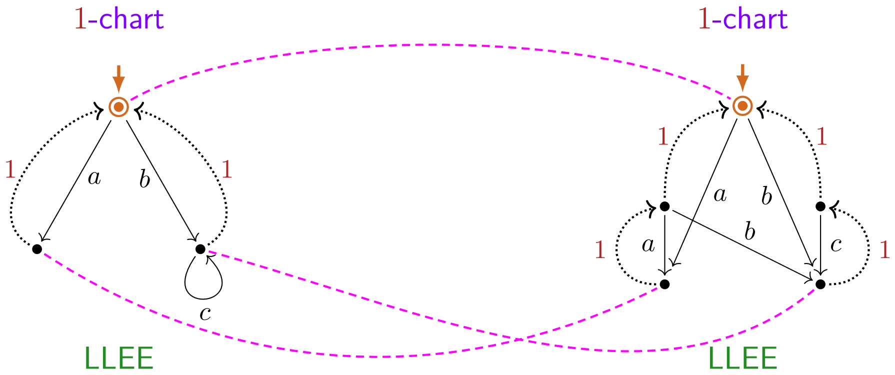


$\underline{\mathcal{C}}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$

$\mathcal{C}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$

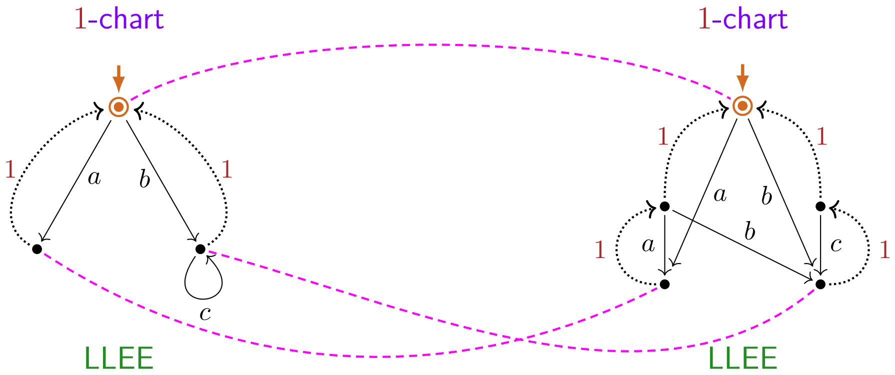
$\underline{\mathcal{C}}\Big(\big(a \cdot (1 \cdot ((a \cdot 1)^* \cdot 1)) \cdot b \cdot ((1 \cdot (c \cdot 1)^*)) \cdot 1 + b \cdot ((1 \cdot (c \cdot 1)^*)) \cdot 1\big)^*\Big)$

Process semantics $[\![\cdot]\!]_P$ (examples, bisimulation collapse)



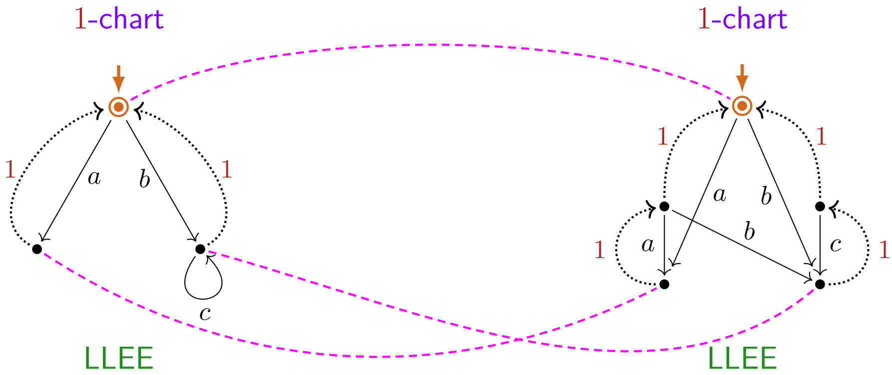
$\underline{\mathcal{C}}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$ $\underline{\leftrightarrow}$ $\underline{\mathcal{C}}\Big(\big(a \cdot (1 \cdot ((a \cdot 1)^* \cdot 1)) \cdot b \cdot ((1 \cdot (c \cdot 1)^*)) \cdot 1 + b \cdot ((1 \cdot (c \cdot 1)^*)) \cdot 1\big)^*\Big)$

Process semantics $[\![\cdot]\!]_P$ (examples, bisimulation collapse)



$$[\![(a \cdot 1 + b \cdot (c^* \cdot 1))^*]\!]_P = [\![\big(a \cdot (1 \cdot ((a \cdot 1)^* \cdot 1)) \cdot b \cdot ((1 \cdot (c \cdot 1)^*)) \cdot 1 + b \cdot ((1 \cdot (c \cdot 1)^*)) \cdot 1\big)^*]\!]_P$$

Process semantics $[\![\cdot]\!]_{\mathbf{P}}$ (examples, bisimulation collapse)



$(a\cdot 1 + b\cdot(c^*\cdot 1))^*$ $\equiv_{\mathbf{P}}$ $\big(a\cdot(1\cdot((a\cdot 1)^*\cdot 1)) \cdot b\cdot((1\cdot(c\cdot 1)^*))\cdot 1 + b\cdot((1\cdot(c\cdot 1)^*))\cdot 1\big)^*$

Milner's proof system Mil

Axioms :

(A1) $e + (f + g) = (e + f) + g$

(A2) $e + 0 = e$

(A3) $e + f = f + e$

(A4) $e + e = e$

(A5) $e \cdot (f \cdot g) = (e \cdot f) \cdot g$

(A6) $(e + f) \cdot g = e \cdot g + f \cdot g$

(A7) $e = 1 \cdot e$

(A8) $e = e \cdot 1$

(A9) $0 = 0 \cdot e$

(A10) $e^* = 1 + e \cdot e^*$

(A11) $e^* = (1 + e)^*$

But: $e \cdot (f + g) \neq e \cdot f + e \cdot g$

But: $e \cdot 0 \neq 0$

Inference rules : rules of equational logic *plus*

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

Milner's question (1984)

Is Mil complete with respect to $\equiv_{\mathbf{P}}$? (Does $\equiv_{\mathbf{P}} \subseteq =_{\mathsf{Mil}}$ hold?)

Bisimulation collapse proof strategy (G/Fokkink, LICS'20)

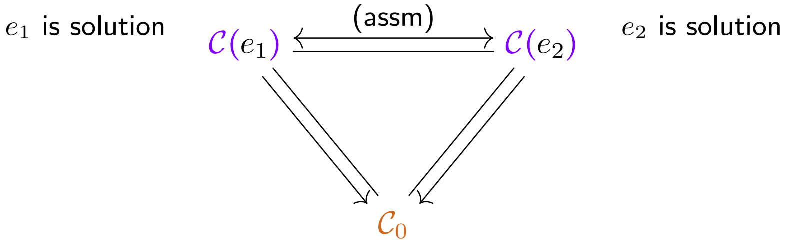
For 1-free regular expressions e_1 and e_2:

e_1 is solution $\quad \mathcal{C}(e_1) \overset{\text{(assm)}}{\longleftrightarrow} \mathcal{C}(e_2) \quad$ e_2 is solution

$e_1 =_{\mathsf{Mil}} e_2$

Bisimulation collapse proof strategy (G/Fokkink, LICS'20)

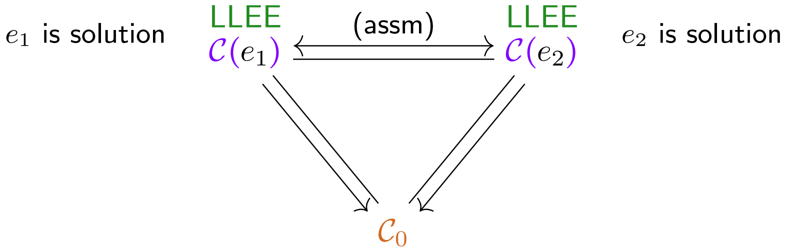
For 1-free regular expressions e_1 and e_2:



$e_1 =_{\mathsf{Mil}} e_2$

Bisimulation collapse proof strategy (G/Fokkink, LICS'20)

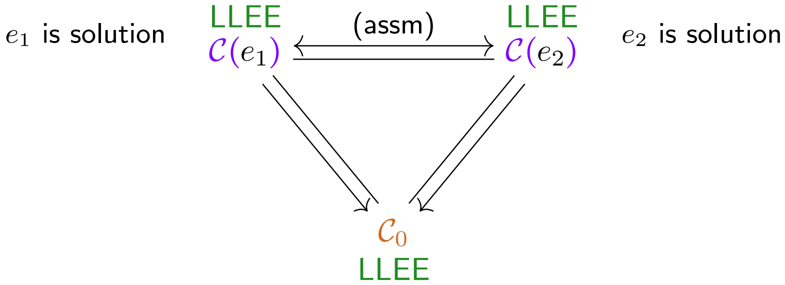
For 1-free regular expressions e_1 and e_2:



$e_1 =_{\mathsf{Mil}} e_2$

Bisimulation collapse proof strategy (G/Fokkink, LICS'20)

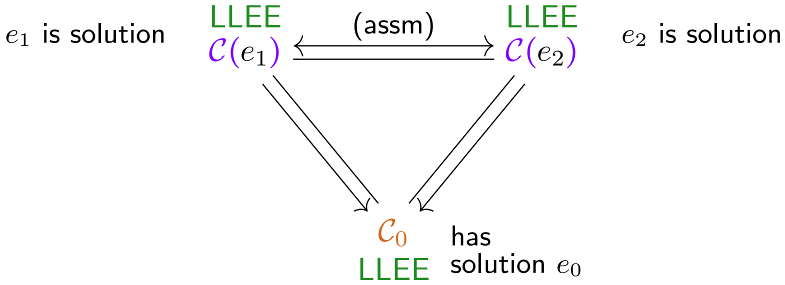
For 1-free regular expressions e_1 and e_2:



$e_1 =_{\text{Mil}} e_2$

Bisimulation collapse proof strategy (G/Fokkink, LICS'20)

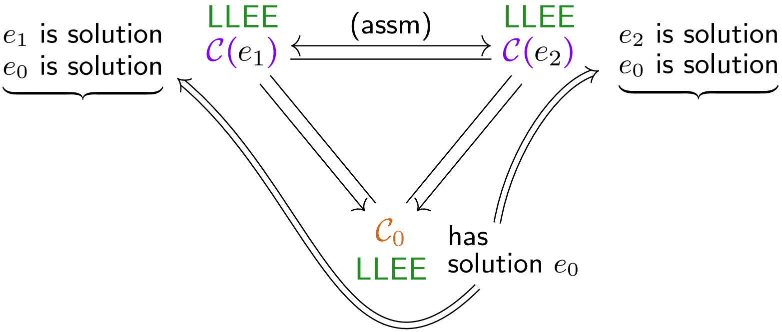
For 1-free regular expressions e_1 and e_2:



$e_1 =_{\text{Mil}} e_2$

Bisimulation collapse proof strategy (G/Fokkink, LICS'20)

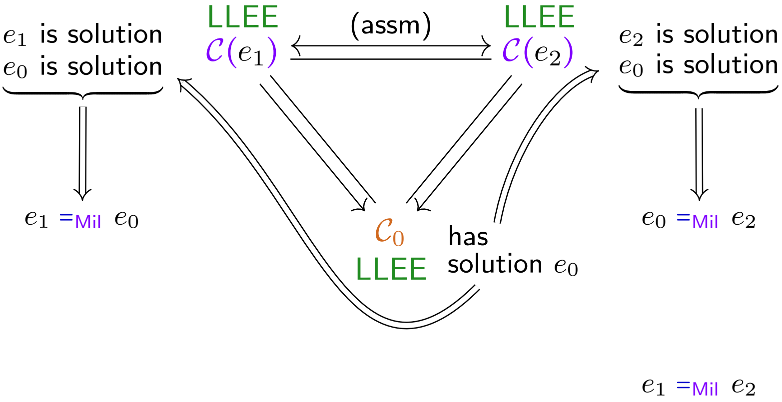
For 1-free regular expressions e_1 and e_2:



$e_1 =_{\mathsf{Mil}} e_2$

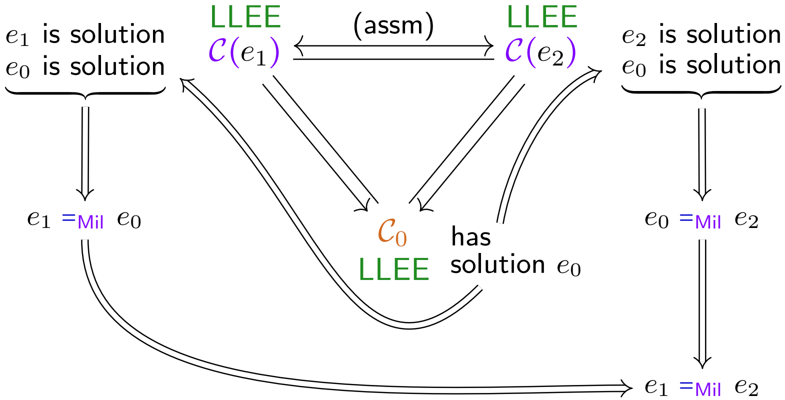
Bisimulation collapse proof strategy (G/Fokkink, LICS'20)

For 1-free regular expressions e_1 and e_2:



Bisimulation collapse proof strategy (G/Fokkink, LICS'20)

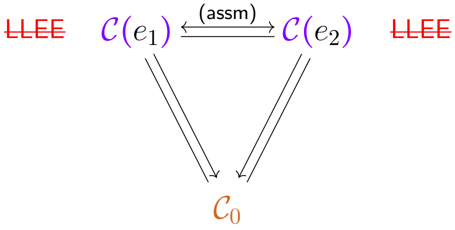
For 1-free regular expressions e_1 and e_2:



Bisimulation collapse proof strategy (general case)

Problem 1

chart interpretations

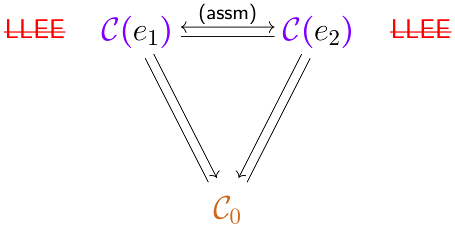


bisimulation collapse

Bisimulation collapse proof strategy (general case)

Remedy for Problem 1 *(G, TERMGRAPH 2020)*

chart interpretations



bisimulation collapse

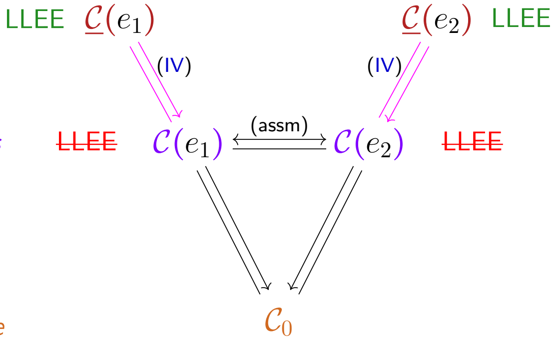
Bisimulation collapse proof strategy (general case)

Remedy for Problem 1 *(G, TERMGRAPH 2020)*

1-*chart interpretations*

chart interpretations

bisimulation collapse



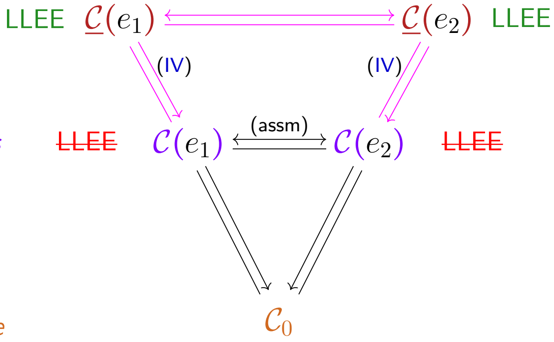
Bisimulation collapse proof strategy (general case)

Remedy for Problem 1 *(G, TERMGRAPH 2020)*

1-*chart interpretations*

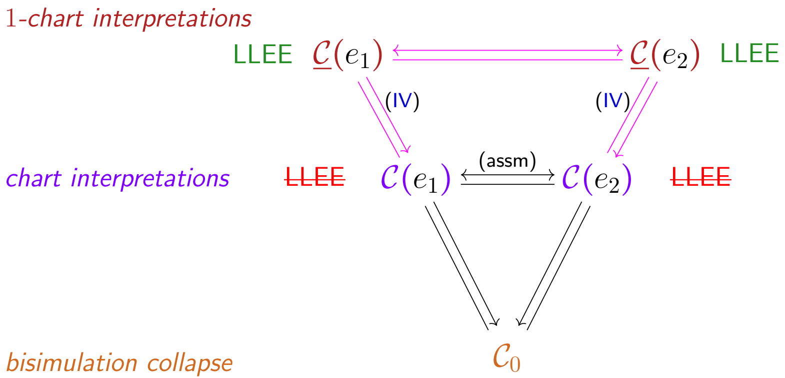
chart interpretations

bisimulation collapse



Bisimulation collapse proof strategy (general case)

Remedy for Problem 1 *(G, TERMGRAPH 2020)*



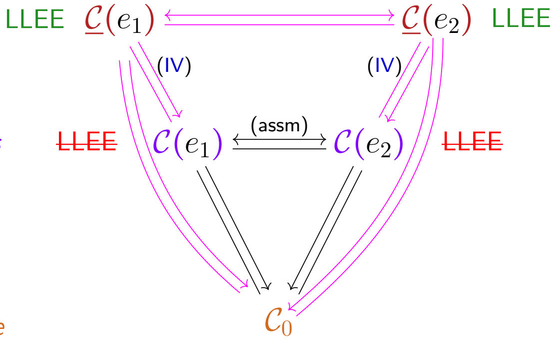
Bisimulation collapse proof strategy (general case)

Remedy for Problem 1 *(G, TERMGRAPH 2020)*

1-*chart interpretations*

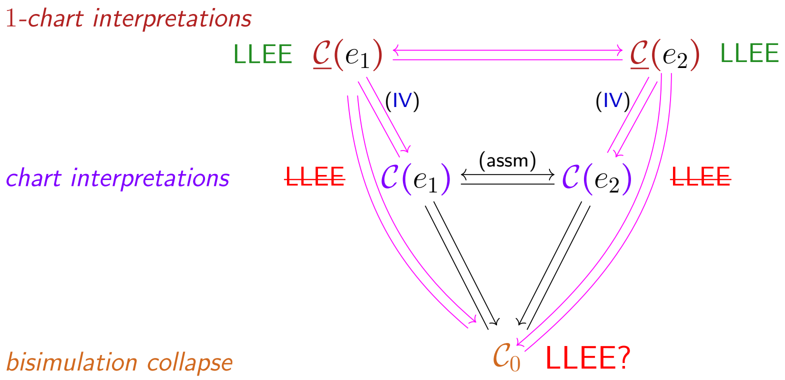
chart interpretations

bisimulation collapse



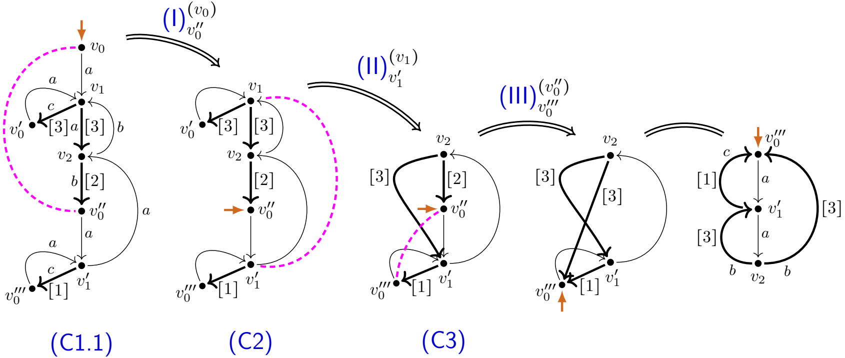
Bisimulation collapse proof strategy (general case)

Remedy for Problem 1 *(G, TERMGRAPH 2020)*



LLEE-preserving collapse of LLEE-charts (G/Fokkink, LICS'20)

(no 1-transitions!)



Lemma

The bisimulation collapse of a LLEE-chart is again a LLEE-chart.

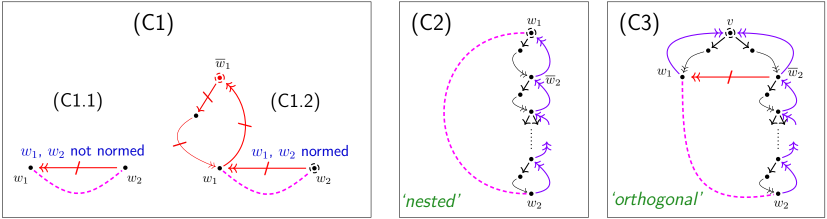
Reduced bisimilarity redundancies in LLEE-charts (no 1-trans.!)
(G/Fokkink, LICS'20)

Lemma

Every not collapsed LLEE-chart contains bisimilar vertices $w_1 \neq w_2$ of kind (C1), (C2), or (C3) (a reduced bisimilarity redundancy $\langle w_1, w_2 \rangle$):

w_1, w_2 in different scc's

w_1, w_2 in the same scc



Lemma

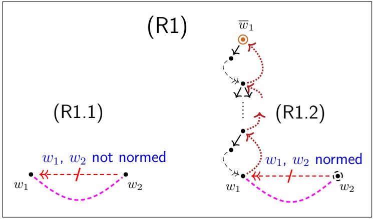
Every reduced bisimilarity redundancy in a LLEE-chart can be eliminated LLEE-preservingly.

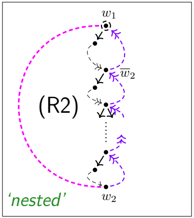
Reduced 1-bisimilarity redundancies in LLEE-1-charts

Lemma

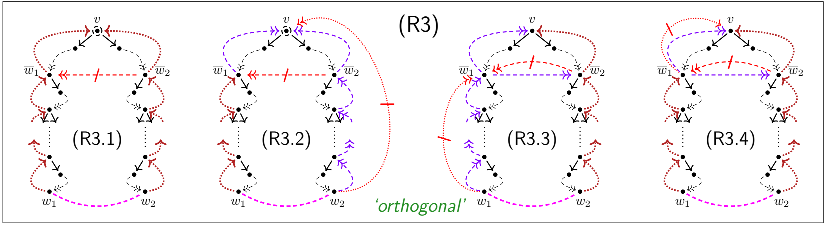
Every not collapsed LLEE-1-chart contains 1-bisimilar vertices $w_1 \neq w_2$ (a reduced 1-bisimilarity redundancy $\langle w_1, w_2 \rangle$) of kind (R1), (R2), (R3):

w_1, w_2 in different scc's





w_1, w_2 in the same scc

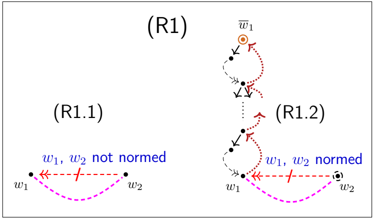


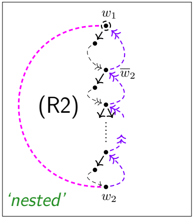
Reduced 1-bisimilarity redundancies in LLEE-1-charts

Lemma

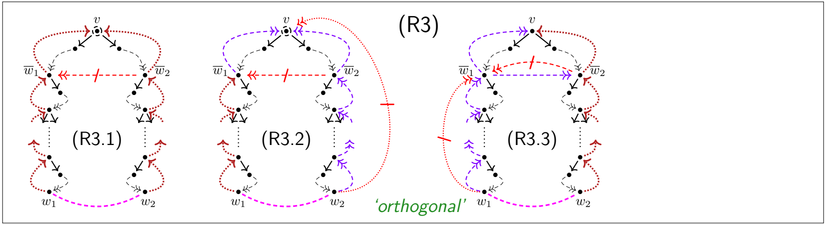
Every simple reduced 1-bisimilarity redundancies in a LLEE-1-chart can be eliminated LLEE-preservingly.

w_1, w_2 in different scc's





w_1, w_2 in the same scc



Reduced 1-bisimilarity redundancies in LLEE-1-charts

Stumbling Block

How to LLEE-preservingly eliminate
reduced 1-bisimilarity redundancies of kind (R3.4)?

w_1, w_2 in different scc's

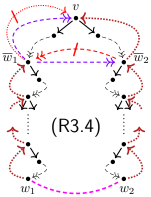
(R1)

(R2)

'nested'

w_1, w_2 in the same scc

(R3)



'orthogonal'

Reduced 1-bisimilarity redundancies in LLEE-1-charts

Stumbling Block

How to LLEE-preservingly eliminate
precrystalline reduced 1-bisimilarity redundancies?

w_1, w_2 in different scc's

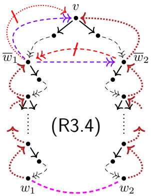
(R1)

(R2)

'nested'

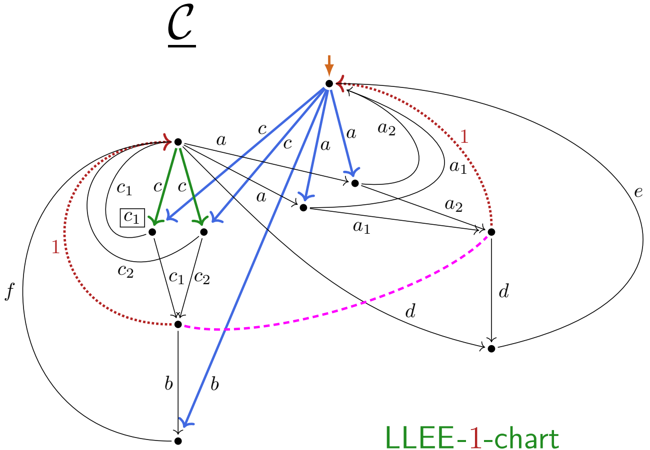
w_1, w_2 in the same scc

(R3)



'orthogonal'

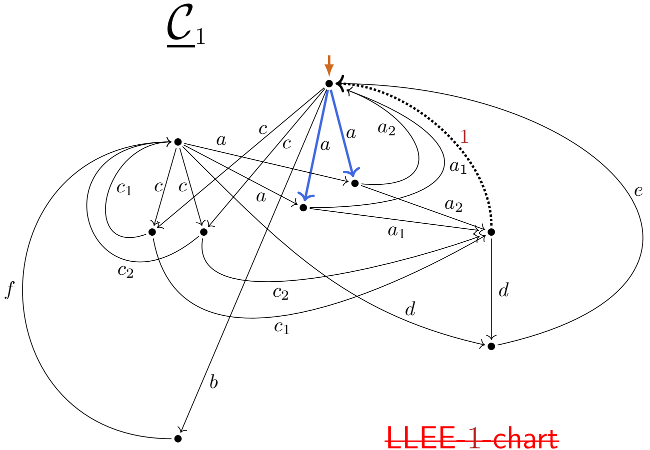
Counterexample LLEE-preserving collapse



LLEE-1-chart

$\langle w_1, w_2 \rangle$ is a reduced
1-bisimilarity redundancy of kind (R3.4)

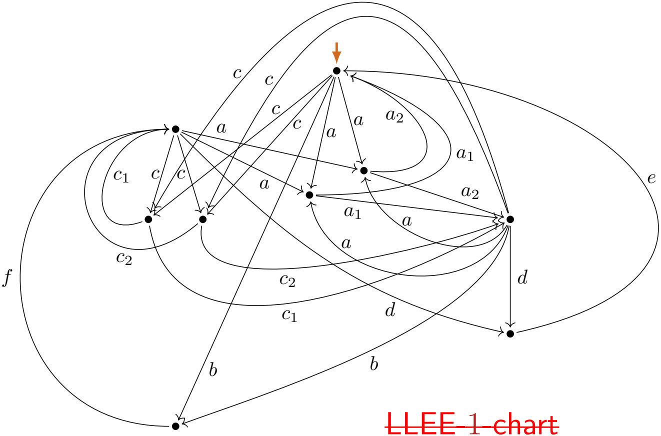
Counterexample LLEE-preserving collapse



~~LLEE-1-chart~~

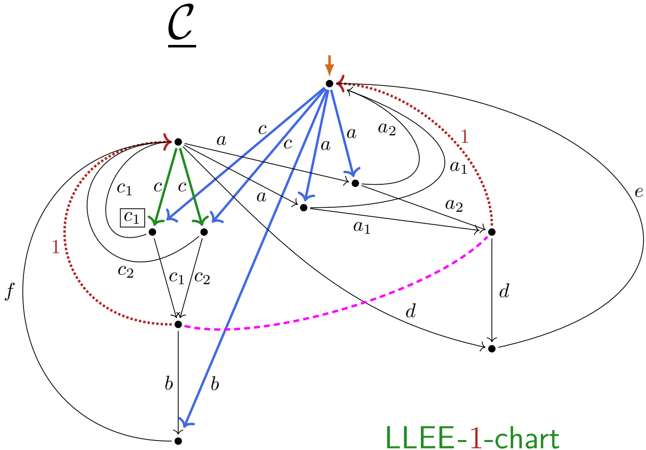
1-bisimulation collapse

Counterexample LLEE-preserving collapse



~~LLEE-1-chart~~
bisimulation collapse

Counterexample LLEE-preserving collapse

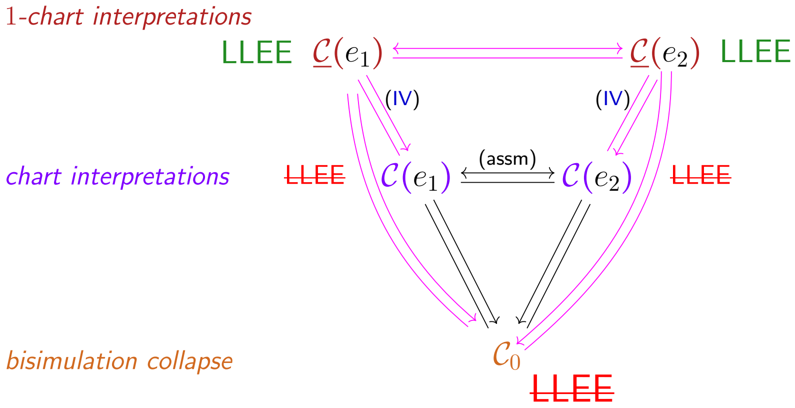


LLEE-1-chart

$\langle w_1, w_2 \rangle$ is a reduced
1-bisimilarity redundancy of kind (R3.4)

Bisimulation collapse proof strategy (general case)

Problem 2: There are regular expressions e_1 and e_2 such that:

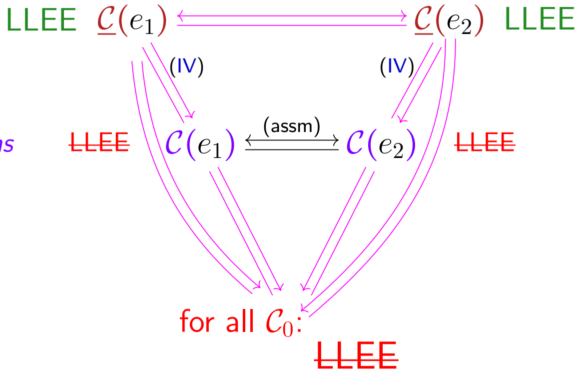


Bisimulation collapse proof strategy (general case)

Problem 2: There are regular expressions e_1 and e_2 such that:

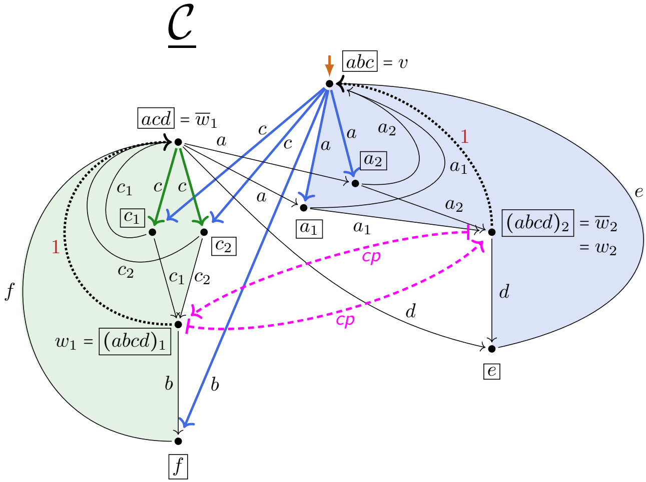
1-*chart interpretations*

chart interpretations

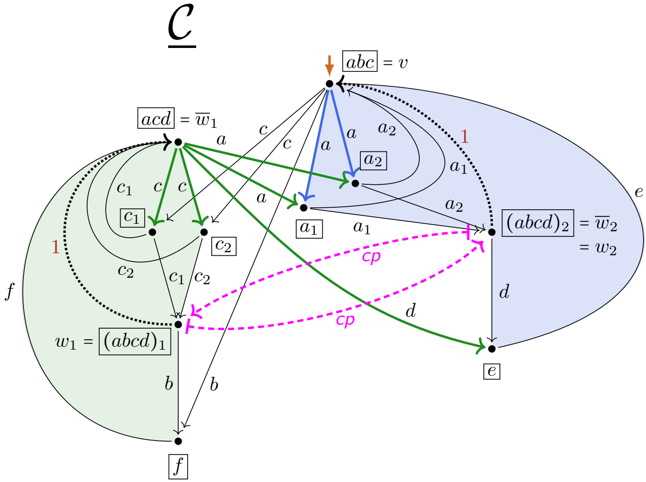


$\mathcal{C}(e_1)$, $\mathcal{C}(e_2)$, $\underline{\mathcal{C}}(e_1)$ and $\underline{\mathcal{C}}(e_2)$ are **not** LLEE-preservingly jointly minimizable under bisimilarity.

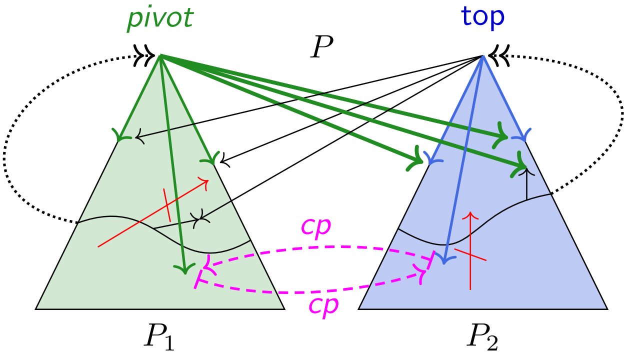
Twin-Crystal



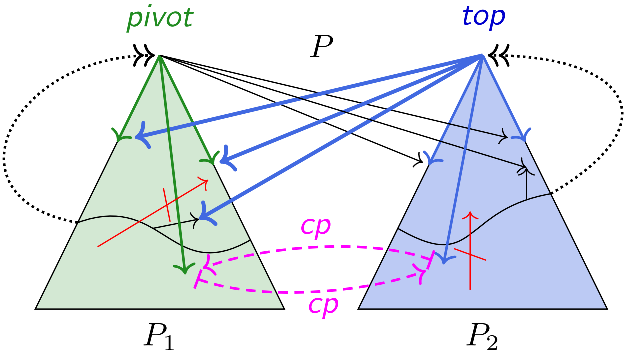
Twin-Crystal



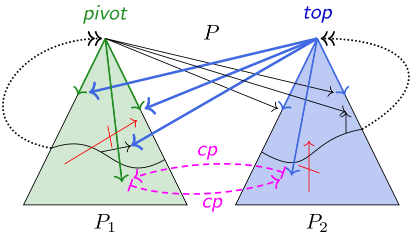
Twin-Crystal



Twin-Crystal



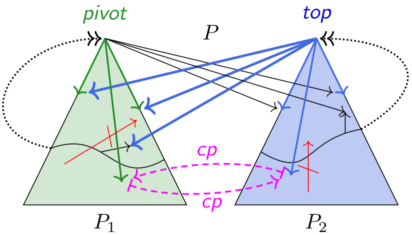
Crystallization



twin-crystal

Crystallized 1-charts = LLEE-1-charts that are collapsed
apart from strongly connected components of twin-crystal form.

Crystallization

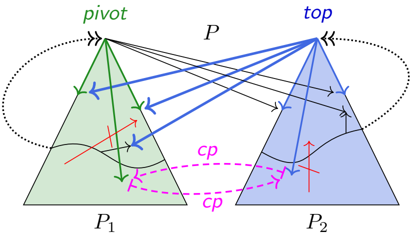


twin-crystal

Crystallized 1-charts = LLEE-1-charts that are collapsed apart from strongly connected components of twin-crystal form.

(CR) Crystallization: Every LLEE-1-chart can be reduced under bisimilarity to a 1-bisimilar crystallized 1-chart.

Crystallization



twin-crystal

Crystallized 1-charts = LLEE-1-charts that are collapsed
apart from strongly connected components of twin-crystal form.

(CR) Crystallization: Every LLEE-1-chart can be reduced under bisimilarity to a 1-bisimilar crystallized 1-chart.

(CC) Every Mil-provable solution of a crystallized 1-chart give rise to Mil-provable solution on the bisimulation collapse.

Completeness proof of Mil (structure)

chart interpretations $\mathcal{C}(e_1) \xleftrightarrow{\text{(assm)}} \mathcal{C}(e_2)$

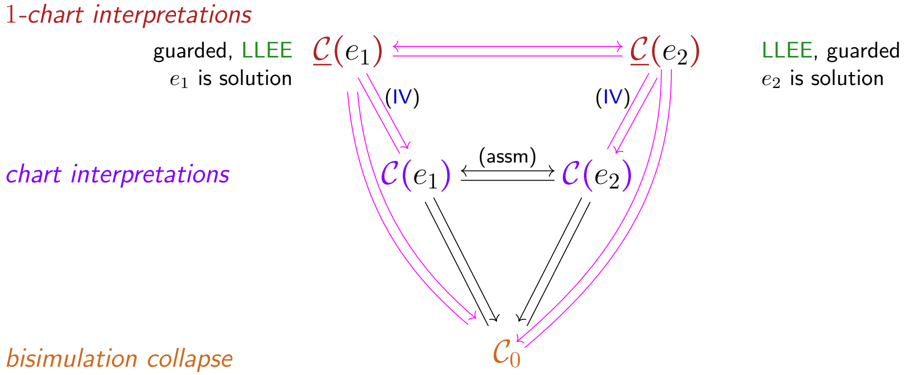
$\overset{?}{\Longrightarrow}$ $e_1 =_{\mathsf{Mil}} e_2$

Completeness proof of Mil (structure)

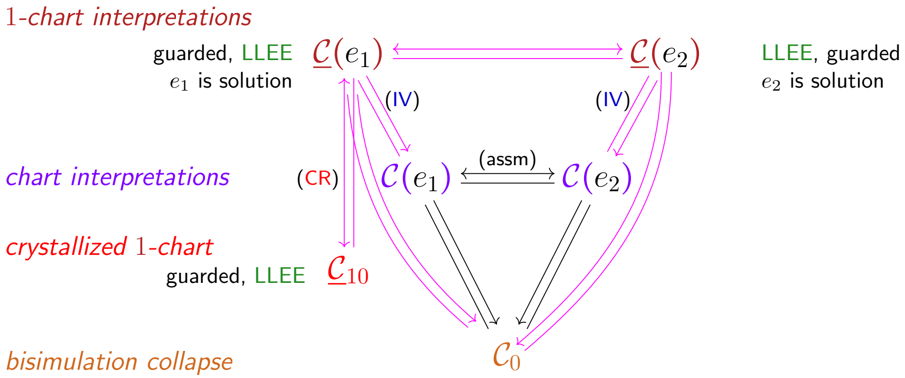
chart interpretations

$$\mathcal{C}(e_1) \xleftrightarrow{\text{(assm)}} \mathcal{C}(e_2)$$

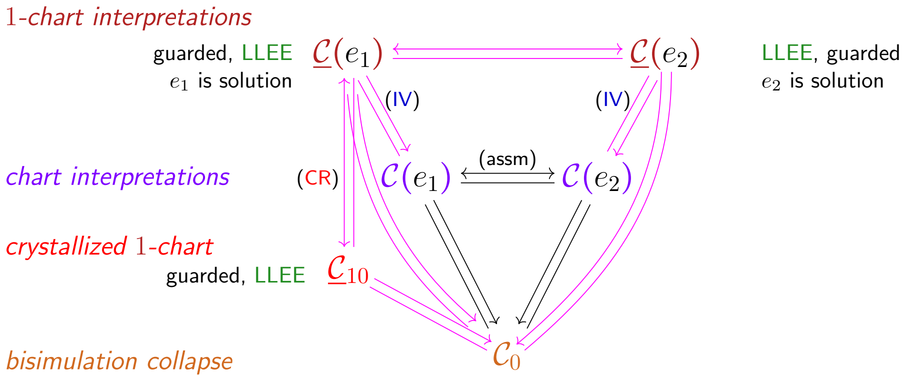
Completeness proof of Mil (structure)



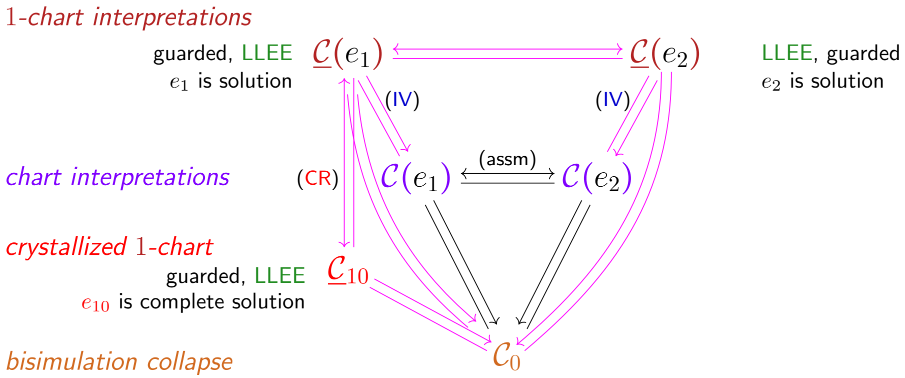
Completeness proof of Mil (structure)



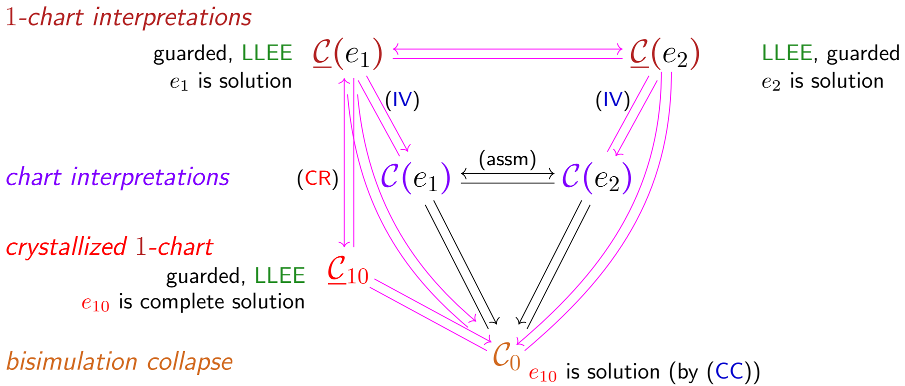
Completeness proof of Mil (structure)



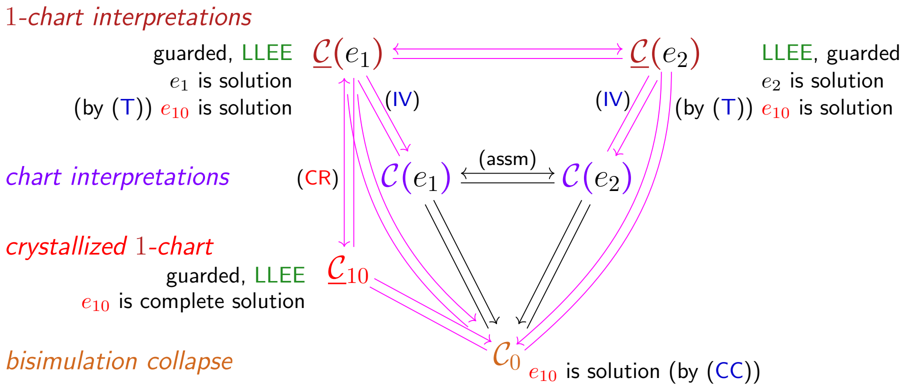
Completeness proof of Mil (structure)



Completeness proof of Mil (structure)



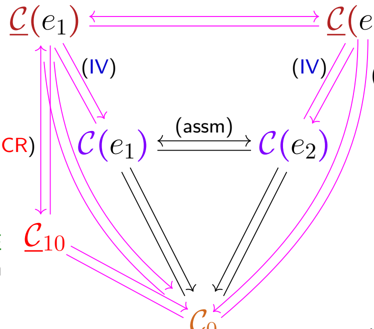
Completeness proof of Mil (structure)



Completeness proof of Mil (structure)

1-chart interpretations

guarded, LLEE
e_1 is solution
(by (T)) e_{10} is solution



LLEE, guarded
e_2 is solution
(by (T)) e_{10} is solution

chart interpretations

crystallized 1-chart

guarded, LLEE
e_{10} is complete solution

bisimulation collapse

e_{10} is solution (by (CC))

$\mathcal{C}(e_1)$ guarded, LLEE
e_1 is solution
e_{10} is solution

$\mathcal{C}(e_2)$ guarded, LLEE
e_2 is solution
e_{10} is solution

Completeness proof of Mil (structure)

1-chart interpretations

guarded, LLEE
e_1 is solution
(by (T)) e_{10} is solution

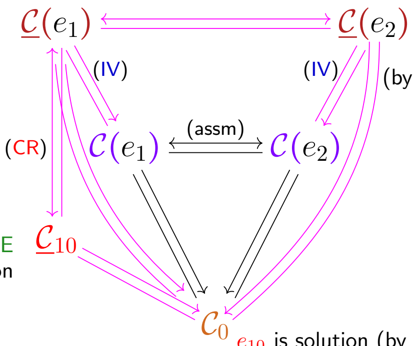
LLEE, guarded
e_2 is solution
(by (T)) e_{10} is solution

chart interpretations

crystallized 1-chart

guarded, LLEE
e_{10} is complete solution

bisimulation collapse



e_{10} is solution (by (CC))

$\left.\begin{array}{r}\mathcal{C}(e_1) \text{ guarded, LLEE} \\ e_1 \text{ is solution} \\ e_{10} \text{ is solution}\end{array}\right\} \underset{\text{(SE)}}{\Longrightarrow} e_1 =_{\text{Mil}} e_{10}$

$\mathcal{C}(e_2)$ guarded, LLEE
e_2 is solution
e_{10} is solution

Completeness proof of Mil (structure)

1-chart interpretations

guarded, LLEE
e_1 is solution
(by (T)) e_{10} is solution

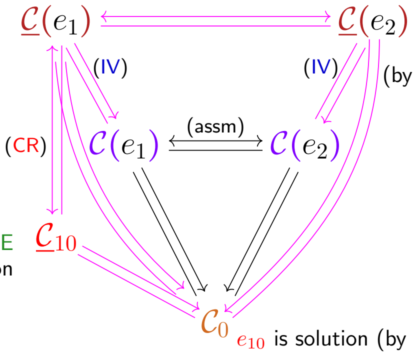
LLEE, guarded
e_2 is solution
(by (T)) e_{10} is solution

chart interpretations

crystallized 1-chart

guarded, LLEE
e_{10} is complete solution

bisimulation collapse

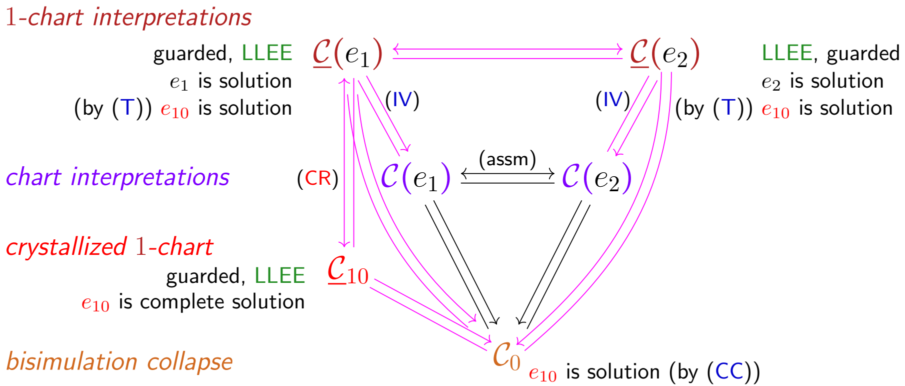


e_{10} is solution (by (CC))

$\left.\begin{array}{r}\mathcal{C}(e_1) \text{ guarded, LLEE} \\ e_1 \text{ is solution} \\ e_{10} \text{ is solution}\end{array}\right\} \overset{}{\underset{\text{(SE)}}{\Longrightarrow}} e_1 =_{\text{Mil}} e_{10}$

$e_{10} =_{\text{Mil}} e_2 \underset{\text{(SE)}}{\Longleftarrow} \left\{\begin{array}{l}\mathcal{C}(e_2) \text{ guarded, LLEE} \\ e_2 \text{ is solution} \\ e_{10} \text{ is solution}\end{array}\right.$

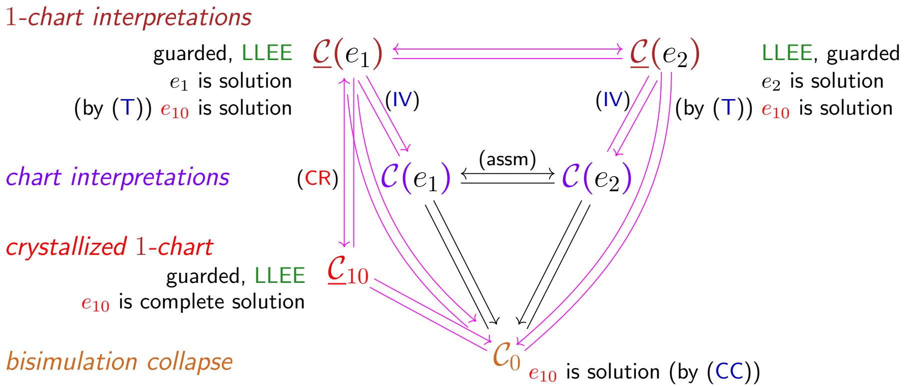
Completeness proof of Mil (structure)



$$\left.\begin{array}{r}\mathcal{C}(e_1) \text{ guarded, LLEE}\\ e_1 \text{ is solution}\\ e_{10} \text{ is solution}\end{array}\right\} \underset{\text{(SE)}}{\Longrightarrow} e_1 =_{\mathsf{Mil}} e_{10} \qquad e_{10} =_{\mathsf{Mil}} e_2 \underset{\text{(SE)}}{\Longleftarrow} \left\{\begin{array}{l}\mathcal{C}(e_2) \text{ guarded, LLEE}\\ e_2 \text{ is solution}\\ e_{10} \text{ is solution}\end{array}\right.$$

$$\Longrightarrow \qquad e_1 =_{\mathsf{Mil}} e_2$$

Completeness proof of Mil (structure)



Theorem

Milner's proof system Mil is complete for process semantics equivalence $\equiv_{\mathbf{P}}$ of regular expressions.

Since: $e_1 \equiv_{\mathbf{P}} e_2 \implies [\![e_1]\!]_{\mathbf{P}} = [\![e_2]\!]_{\mathbf{P}} \implies \mathcal{C}(e_1) \leftrightarrows \mathcal{C}(e_1) \implies e_1 =_{\mathsf{Mil}} e_2$.

Outlook

poster presentation

- tomorrow, 10–10.30

next steps and projects

- monograph project: proof in fine-grained detail
- computation/animation tool for crystallization
- use crystallization for recognition problem

resources on Github:

- `https://github.com/clegra/crystallization/blob/main`
  - article (after rebuttal): `/cryst-article.pdf`
  - poster: /poster-lics2022.pdf
  - presentation: /presentation-lics2022.pdf

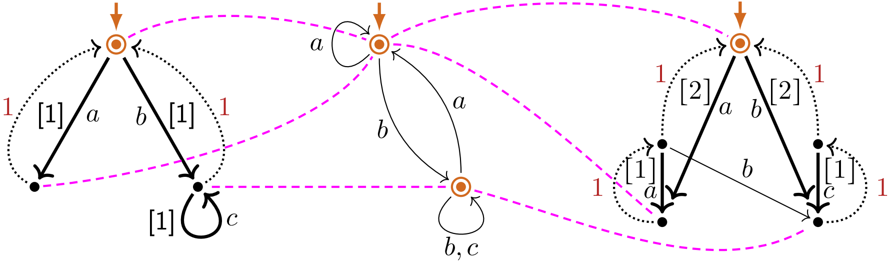
acknowledgment & thanks to:

- Wan Fokkink (for long collaboration)

Thank you for your attention!

Process semantics $[\![\cdot]\!]_P$ (examples, bisimulation collapse)

1-chart | bisimulation collapse chart | 1-chart



LLEE | ~~LLEE~~ | LLEE

$\underline{\mathcal{C}}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$

$\mathcal{C}((a \cdot 1 + b \cdot (c^* \cdot 1))^*)$

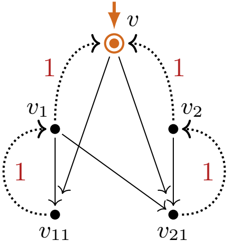
$\underline{\mathcal{C}}\big((a \cdot (1 \cdot ((a \cdot 1)^* \cdot 1)) \cdot b \cdot ((1 \cdot (c \cdot 1)^*)) \cdot 1 + b \cdot ((1 \cdot (c \cdot 1)^*)) \cdot 1\big)^*\big)$

$\underline{\mathcal{C}}((a + b \cdot c^*)^*)$

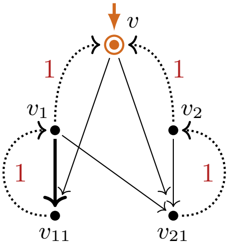
$\mathcal{C}((a + b \cdot c^*)^*)$

$\underline{\mathcal{C}}((a^* \cdot (1 + b \cdot c^*))^*)$

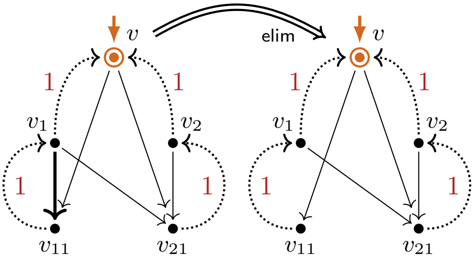
Layered loop existence/elimination and LLEE-witnesses



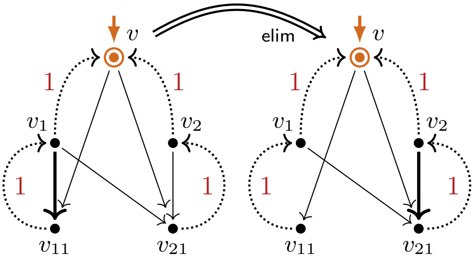
Layered loop existence/elimination and LLEE-witnesses



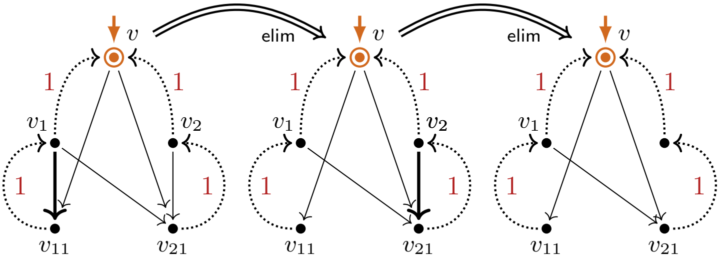
Layered loop existence/elimination and LLEE-witnesses



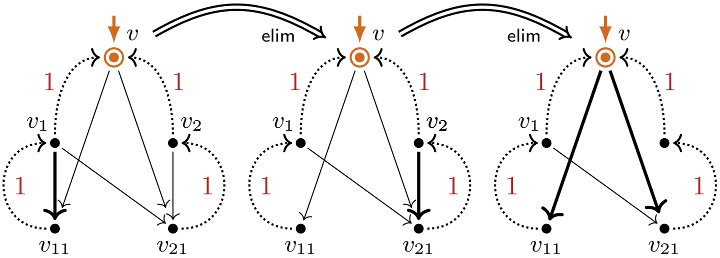
Layered loop existence/elimination and LLEE-witnesses



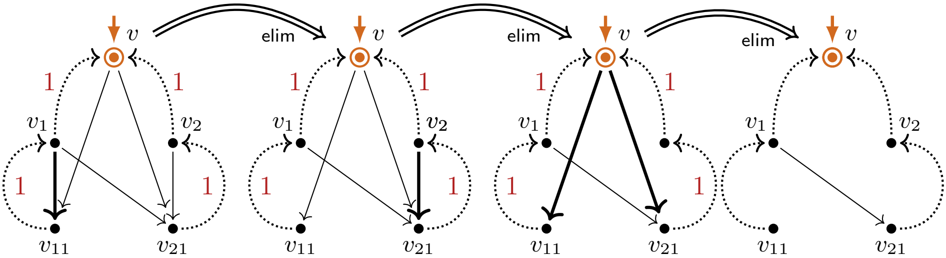
Layered loop existence/elimination and LLEE-witnesses



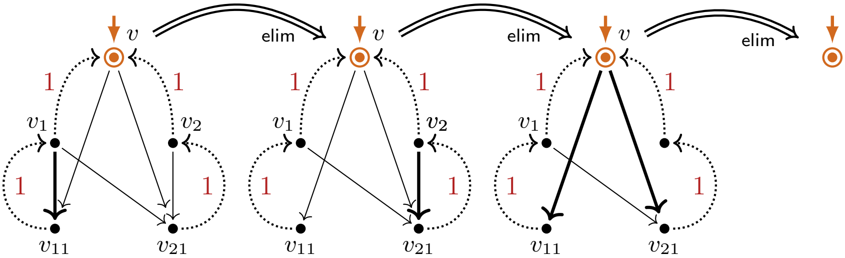
Layered loop existence/elimination and LLEE-witnesses



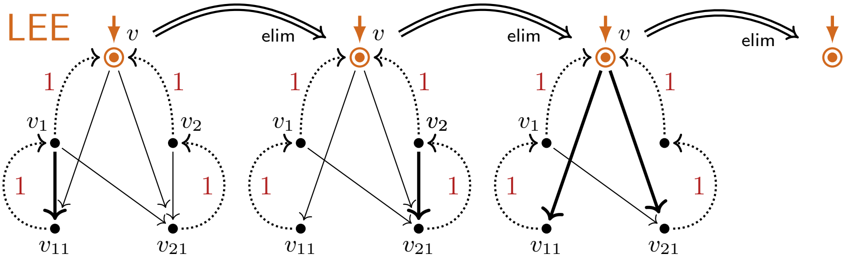
Layered loop existence/elimination and LLEE-witnesses



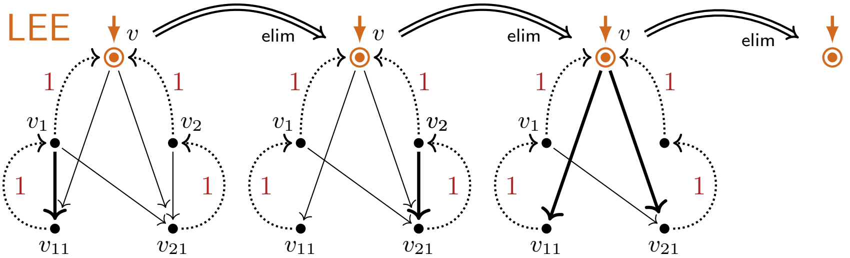
Layered loop existence/elimination and LLEE-witnesses

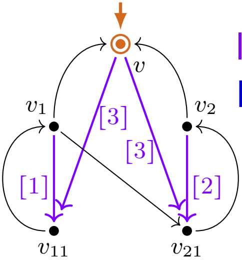


Layered loop existence/elimination and LLEE-witnesses



Layered loop existence/elimination and LLEE-witnesses

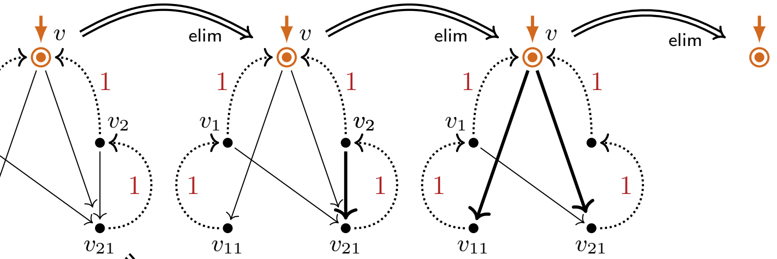




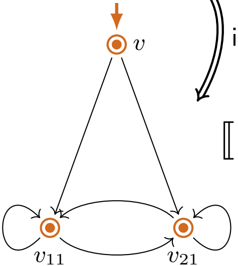
layered
LEE-witness

Layered loop existence/elimination and LLEE-witnesses

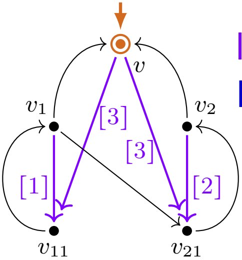
LEE



induces

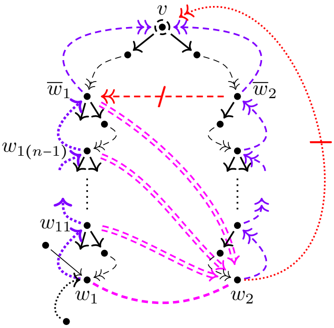


$[\![(a^* \cdot b^*)^*]\!]_{\mathsf{P}}$



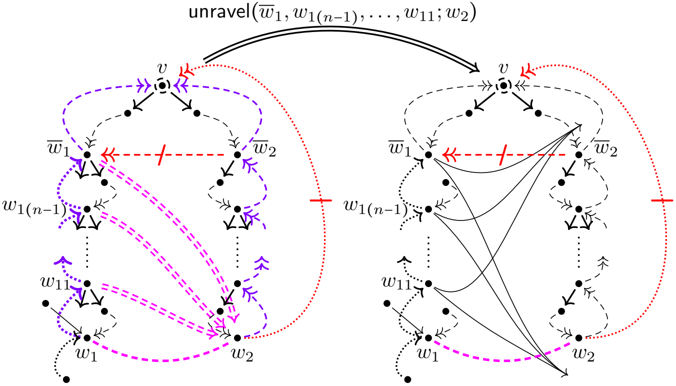
layered LEE-witness

Eliminating reduced 1-bisimilarity redundancies (example)



reduced 1-bisimilarity
redundancy of kind (R3.2)

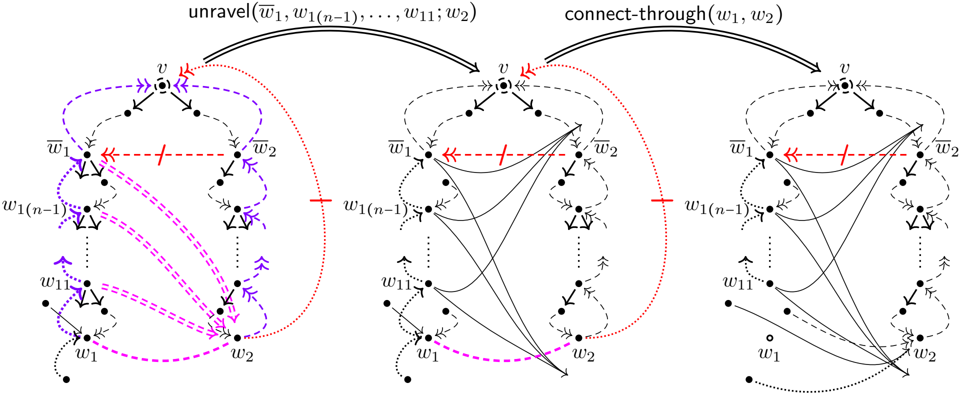
Eliminating reduced 1-bisimilarity redundancies (example)



reduced 1-bisimilarity redundancy of kind (R3.2)

result of unravelling loop vertices $\overline{w}_1, w_{1(n-1)}, \ldots, w_{11}$

Eliminating reduced 1-bisimilarity redundancies (example)



reduced 1-bisimilarity redundancy of kind (R3.2)

result of unravelling loop vertices $\overline{w}_1, w_{1(n-1)}, \ldots, w_{11}$

result of connecting through w_1 to w_2