Forms of Graph Sharing, and Expressibility of Process Graphs by Regular Expressions

Clemens Grabmayer

Computer Science group
Gran Sasso Science Institute
L'Aquila

Formal Methods Reading Group
CS@GSSI
December 14, 2022

Overview

Forms of sharing (in term graphs / rooted directed graphs)

- horizontal, vertical, and twisted sharing
- twisted ⇒ only vertical sharing: exponential size increase possible

Expressibility of process graphs by regular expressions

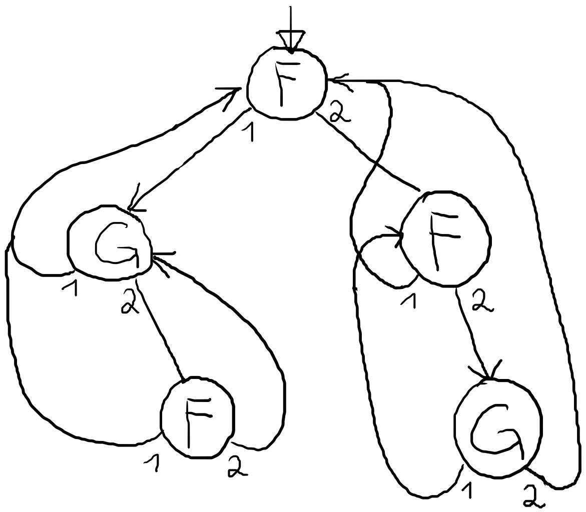
- loop-elimination property LEE
  - guarantees expressibility by a regular expression
- process interpretations of reg. expressions do not always satisfy LEE
  - but can be 1-transition refined to guarantee LEE
- expressibility problem of process graphs by regular expressions
- steps to solve the expressibility problem (perhaps) efficiently

Questions

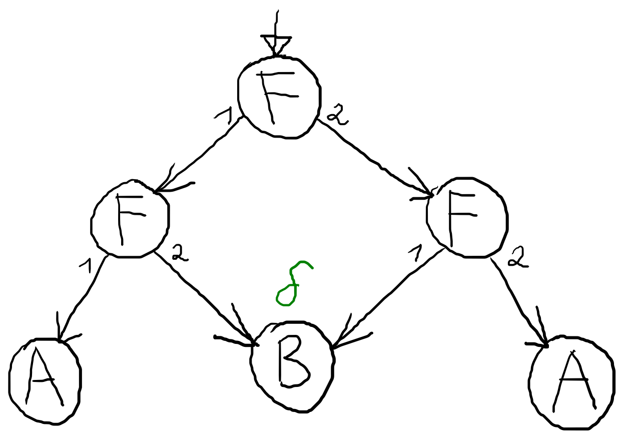
Forms of Sharing

Forms of Sharing (Stefan Blom, 2001) in term graphs and rooted directed graphs

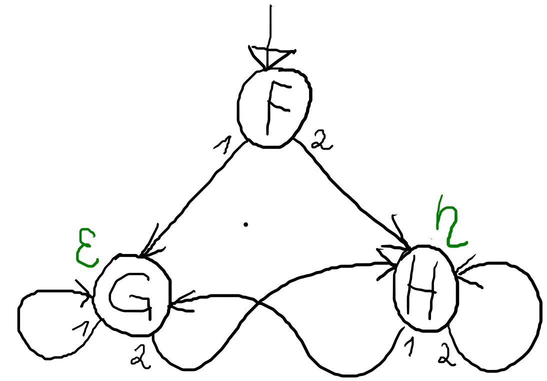
Vertical

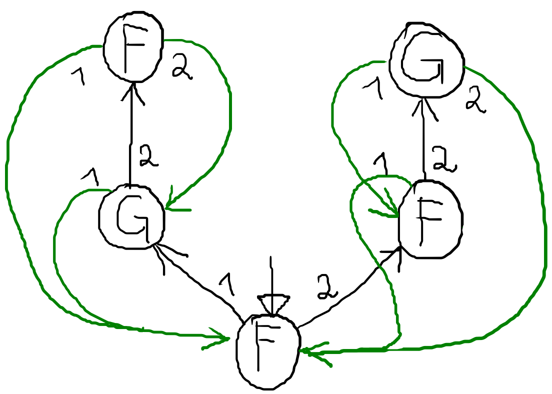


Horizontal



Twisted

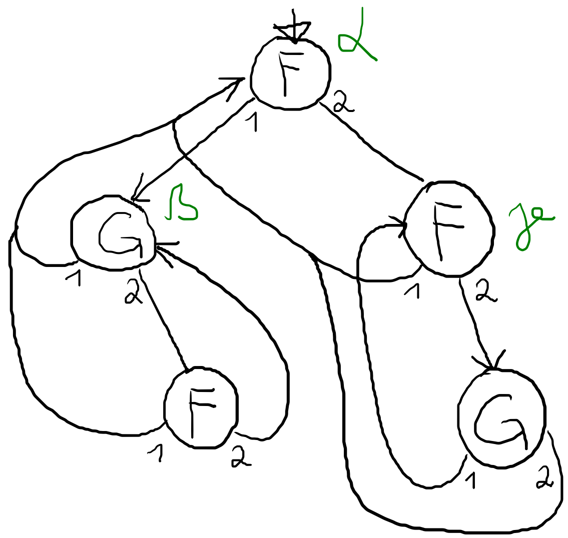




palm tree (Tarjan)

Forms of Sharing in term graphs and rooted directed graphs (Blom, 2001)

Vertical



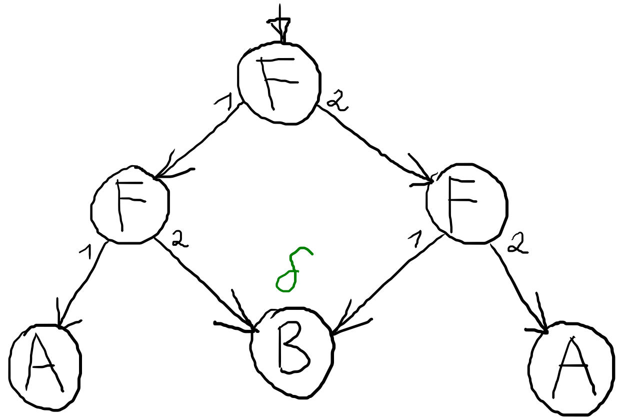
 is embeddable

$\mu\alpha.\, F(\mu\beta.\, G(\alpha, F(\alpha,\beta)),\ \mu\gamma.\, F(\alpha, G(\gamma,\alpha)))$

μ-expressible

ar(ity) (F) = ar (G) = ar(H) = 2 ar (A) = Ar (B) = 0

Horizontal



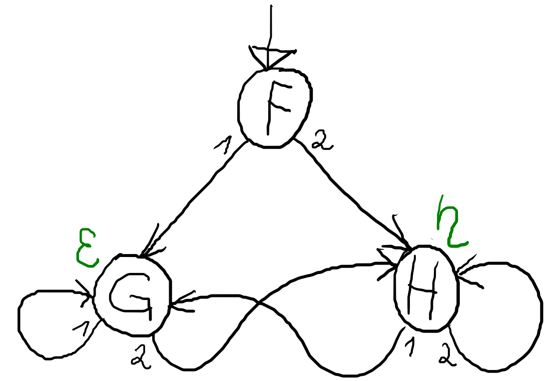
 is embeddable

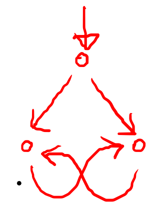
<u>Let</u> $\delta = B$

<u>in</u> $F(F(A,\delta),$
$F(\delta, A))$

Let - expressible

Twisted



 is embeddable

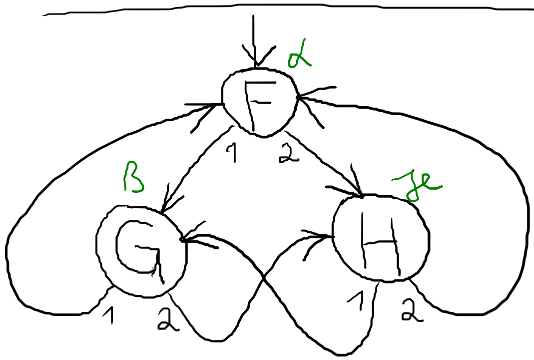
<u>Letrec</u> $\varepsilon = G(\varepsilon, \eta),$
$\eta = H(\varepsilon, \eta)$

<u>in</u> $F(\varepsilon, \eta)$

letrec - expressible

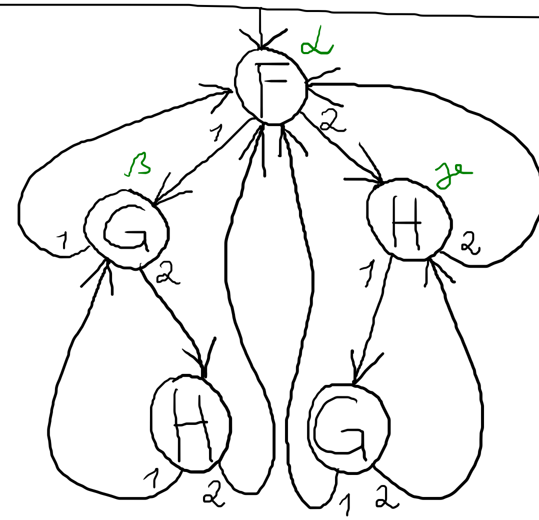
Exponential size increase from twisted to vertical-sharing representations:

ar(F) = ar(G) = ar(H) = 2



twisted sharing

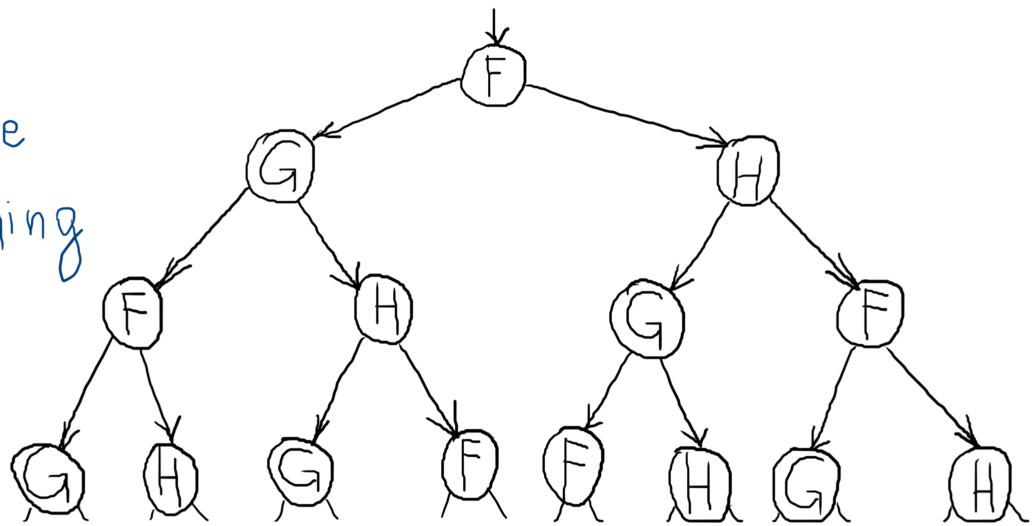
3 vertices, 6+1=7 edges



only vertical sharing

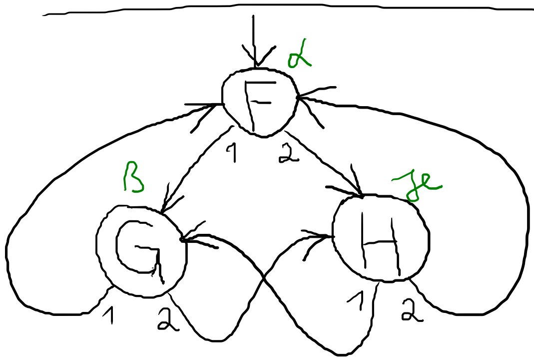
5 vertices
10+1 edges
vertical paths include: FGH
FHG
2!

tree unfolding



Exponential size increase from twisted to vertical-sharing representations:

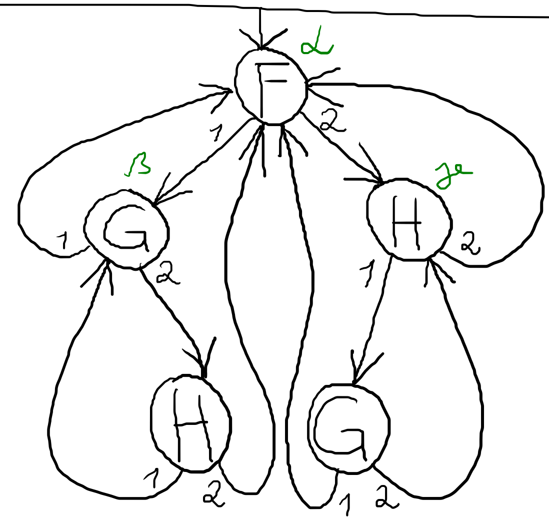
$ar(F) = ar(G) = ar(H) = 2$



twisted sharing

3 vertices, 6+1=7 edges

<u>Letrec</u> $\alpha = F(\beta, \gamma)$, $\beta = G(\alpha, \gamma)$, $\gamma = H(\beta, \alpha)$
<u>in</u> α



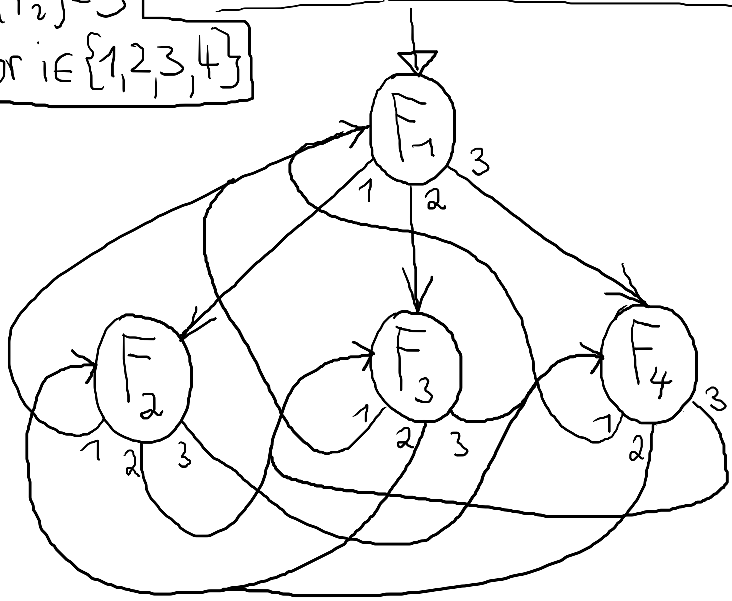
only vertical sharing

5 vertices
10+1 edges
vertical paths include: FGH
FHG
2!

$\mu\alpha.F(\mu\beta.G(\alpha, H(\beta, \alpha)), \mu\gamma.H(G(\alpha, \gamma), \alpha))$

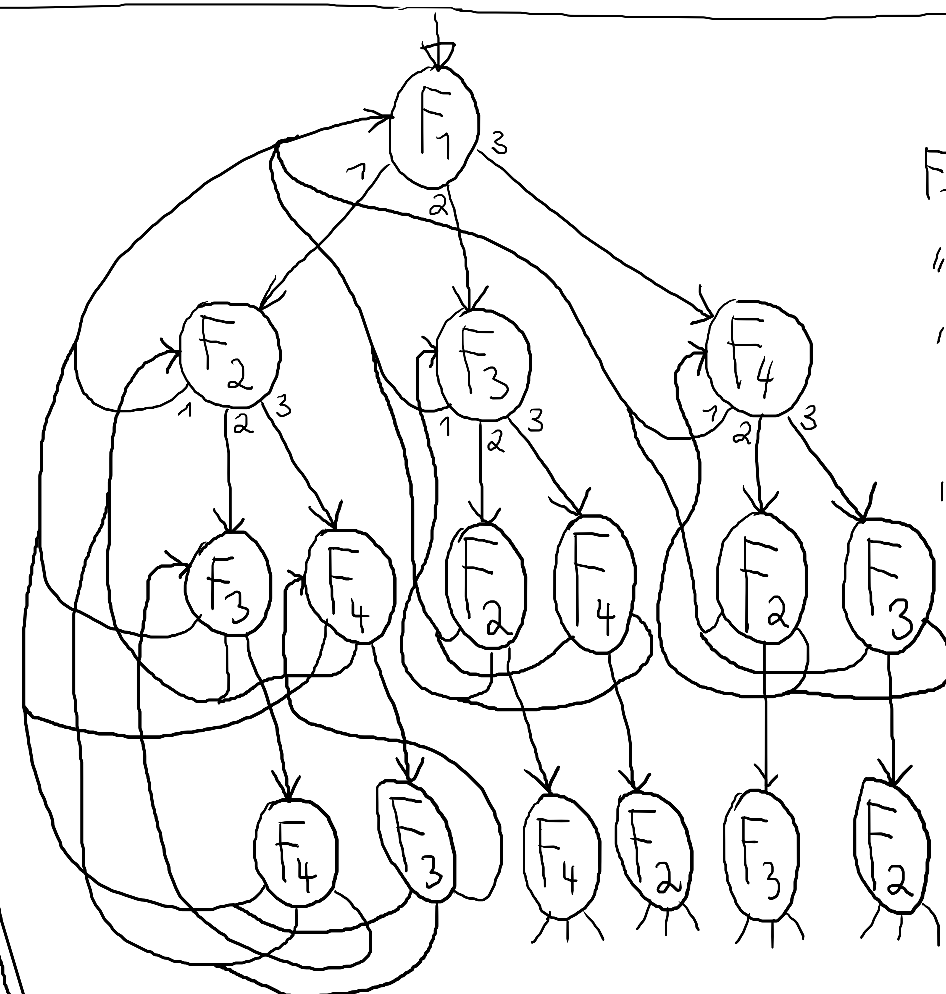
Exponential size increase from twisted to vertical-sharing representations

$ar(F_i)=3$ for $i \in \{1,2,3,4\}$



4 vertices
12+1 edges

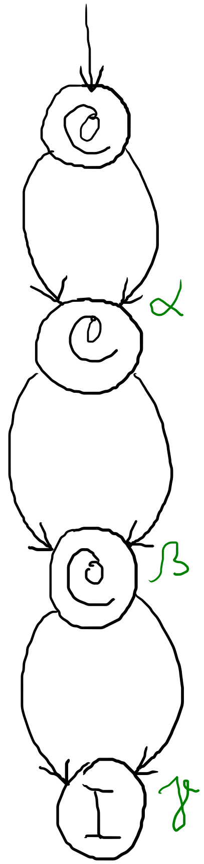
$\Rightarrow$ for n vertices in twisted sharing
$\underbrace{(n-1)!}_{\in \Omega(2^n)}$ vertices in vertical-sharing representation

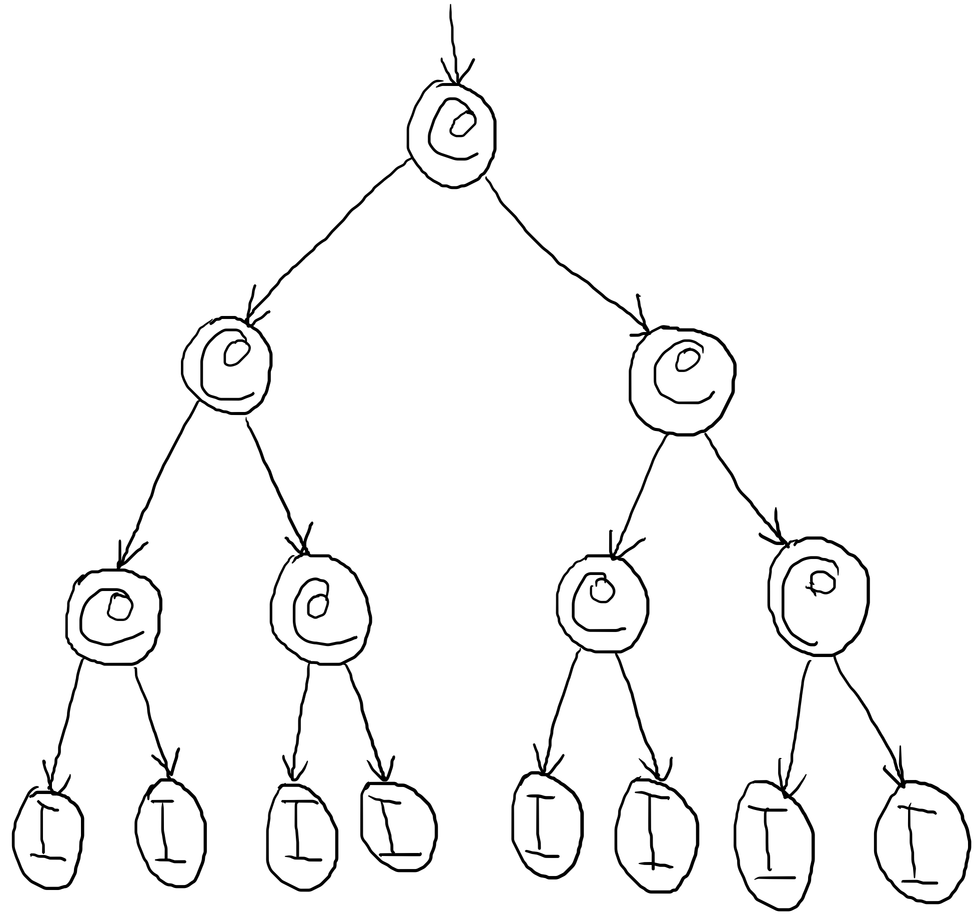


$F_1\ F_2\ F_3\ F_4$
" " $F_4\ F_3$
" $F_3\ F_2\ F_4$
" " $F_4\ F_2$
" $F_4\ F_2\ F_3$
" " $F_3\ F_2$

$3! = 6$ vertices
18+1 edges

horizontal sharing





Let $\alpha = @(\beta, \beta)$

$\beta = @(\gamma, \gamma)$

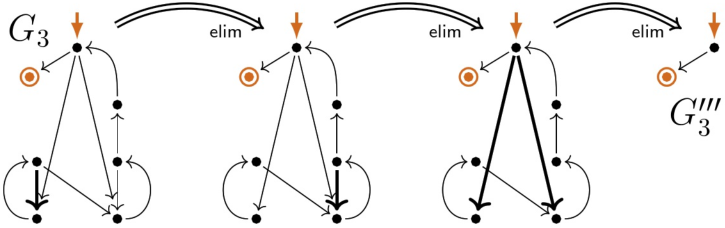
$\gamma = I$

in $@(\alpha, \alpha)$

$@(@(@(I, I), @(I, I)),$
$@(@(I, I), @(I, I)))$

Expressibility of process graphs by regular expressions

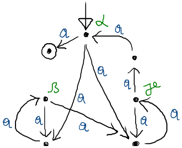
Loop Existence and Elimination (LEE)



LEE

vertical, horizontal
but not twisted sharing

(Milner, 1984)



all transitions are actions a

"Behaviour" as:

regular expressions: $(a \cdot a \cdot (a \cdot a)^{*} \cdot a + a) \cdot a \cdot (a \cdot a)^{*} \cdot a \cdot a)^{*} \cdot a$

regular behaviour in μ-term notation: $\mu\alpha.\big((a \cdot a \cdot \mu\beta(a \cdot a \cdot \beta + a) + a) \cdot a \cdot \mu\gamma(a \cdot a \cdot \gamma + a \cdot a \cdot \alpha) + a\big)$

Loop charts (interpretations of innermost iterations)

Definition

A chart is a loop chart if:

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

(L2) Every infinite path from the start vertex returns to it.

(L3) Termination is only possible at the start vertex.

Loop charts (interpretations of innermost iterations)

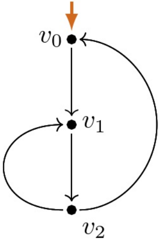
Definition

A chart is a loop chart if:

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

(L2) Every infinite path from the start vertex returns to it.

(L3) Termination is only possible at the start vertex.



Loop charts (interpretations of innermost iterations)

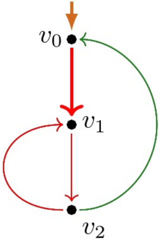
Definition

A chart is a loop chart if:

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

(L2) Every infinite path from the start vertex returns to it.

(L3) Termination is only possible at the start vertex.



(L1), ~~(L2)~~

Loop charts (interpretations of innermost iterations)

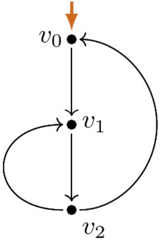
Definition

A chart is a loop chart if:

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

(L2) Every infinite path from the start vertex returns to it.

(L3) Termination is only possible at the start vertex.



(L1), ~~(L2)~~

Loop charts (interpretations of innermost iterations)

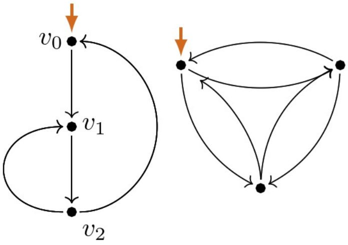
Definition

A chart is a loop chart if:

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

(L2) Every infinite path from the start vertex returns to it.

(L3) Termination is only possible at the start vertex.



(L1), ~~(L2)~~

Loop charts (interpretations of innermost iterations)

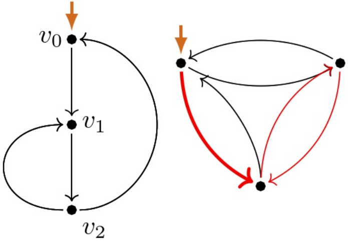
Definition

A chart is a loop chart if:

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

(L2) Every infinite path from the start vertex returns to it.

(L3) Termination is only possible at the start vertex.



(L1), ~~(L2)~~

Loop charts (interpretations of innermost iterations)

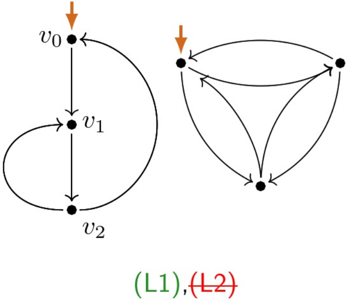
Definition

A chart is a loop chart if:

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

(L2) Every infinite path from the start vertex returns to it.

(L3) Termination is only possible at the start vertex.



Loop charts (interpretations of innermost iterations)

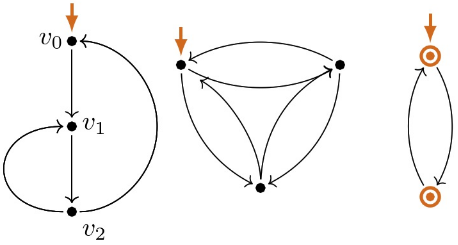
Definition

A chart is a loop chart if:

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

(L2) Every infinite path from the start vertex returns to it.

(L3) Termination is only possible at the start vertex.



(L1), ~~(L2)~~

Loop charts (interpretations of innermost iterations)

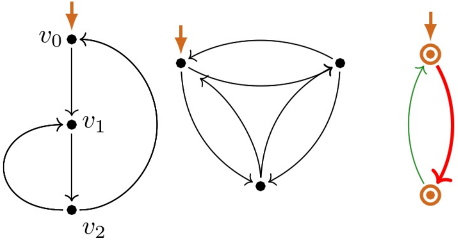
Definition

A chart is a loop chart if:

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

(L2) Every infinite path from the start vertex returns to it.

(L3) Termination is only possible at the start vertex.



(L1),~~(L2)~~

Loop charts (interpretations of innermost iterations)

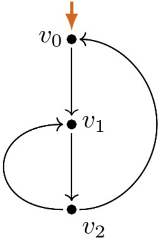
Definition

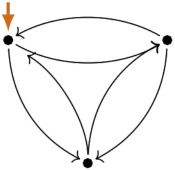
A chart is a loop chart if:

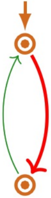
(L1) There is an infinite path from the start vertex.

(L2) Every infinite path from the start vertex returns to it.

(L3) Termination is only possible at the start vertex.







(L1),~~(L2)~~ (L1),(L2),~~(L3)~~

Loop charts (interpretations of innermost iterations)

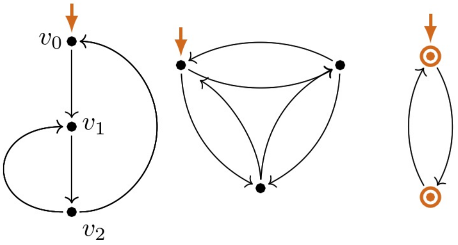
Definition

A chart is a loop chart if:

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

(L2) Every infinite path from the start vertex returns to it.

(L3) Termination is only possible at the start vertex.



(L1),~~(L2)~~ (L1),(L2),~~(L3)~~

Loop charts (interpretations of innermost iterations)

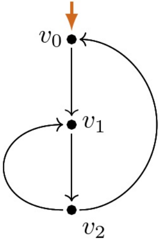
Definition

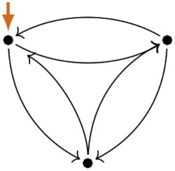
A chart is a loop chart if:

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

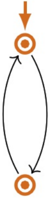
(L2) Every infinite path from the start vertex returns to it.

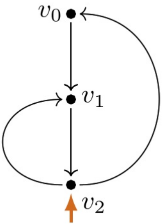
(L3) Termination is only possible at the start vertex.





(L1),~~(L2)~~





(L1),(L2),~~(L3)~~

Loop charts (interpretations of innermost iterations)

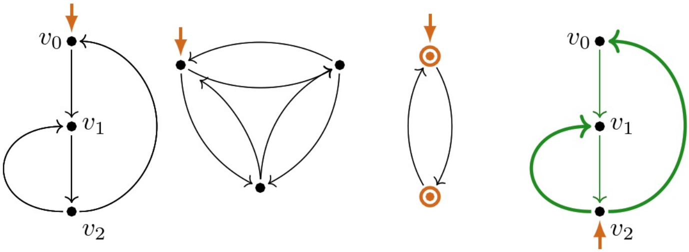
Definition

A chart is a loop chart if:

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

(L2) Every infinite path from the start vertex returns to it.

(L3) Termination is only possible at the start vertex.



(L1),~~(L2)~~ (L1),(L2),~~(L3)~~

Loop charts (interpretations of innermost iterations)

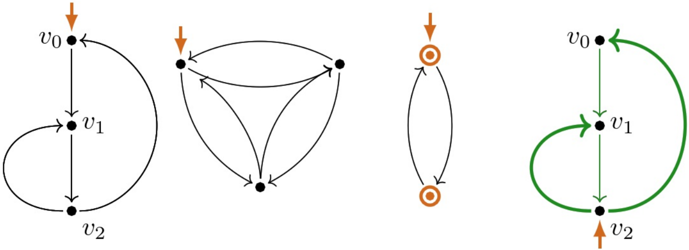
Definition

A chart is a loop chart if:

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

(L2) Every infinite path from the start vertex returns to it.

(L3) Termination is only possible at the start vertex.



(L1),~~(L2)~~ (L1),(L2),~~(L3)~~ loop chart

Loop charts (interpretations of innermost iterations)

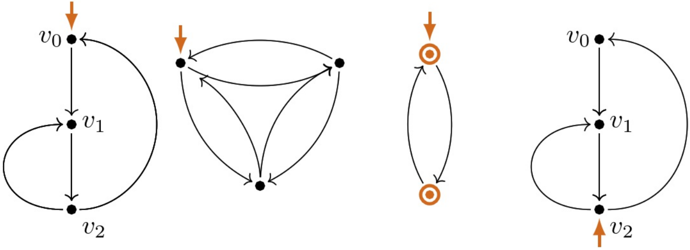
Definition

A chart is a loop chart if:

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

(L2) Every infinite path from the start vertex returns to it.

(L3) Termination is only possible at the start vertex.



(L1),~~(L2)~~ (L1),(L2),~~(L3)~~ loop chart

Loop charts (interpretations of innermost iterations)

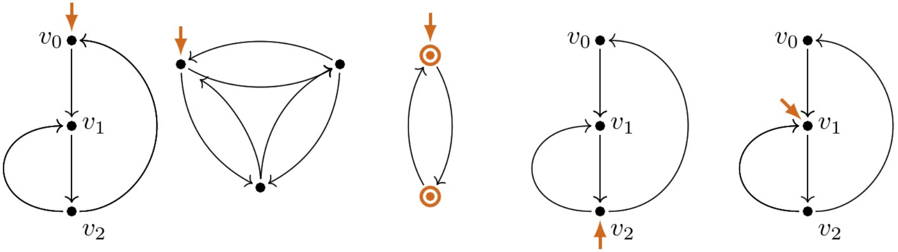
Definition

A chart is a loop chart if:

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

(L2) Every infinite path from the start vertex returns to it.

(L3) Termination is only possible at the start vertex.



(L1),~~(L2)~~ (L1),(L2),~~(L3)~~ loop chart

Loop charts (interpretations of innermost iterations)

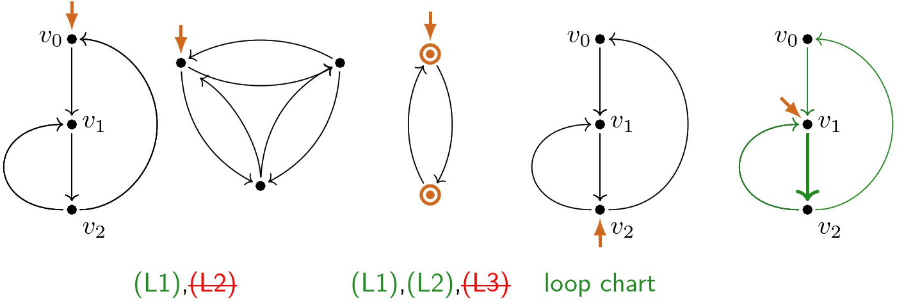
Definition

A chart is a loop chart if:

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

(L2) Every infinite path from the start vertex returns to it.

(L3) Termination is only possible at the start vertex.



Loop charts (interpretations of innermost iterations)

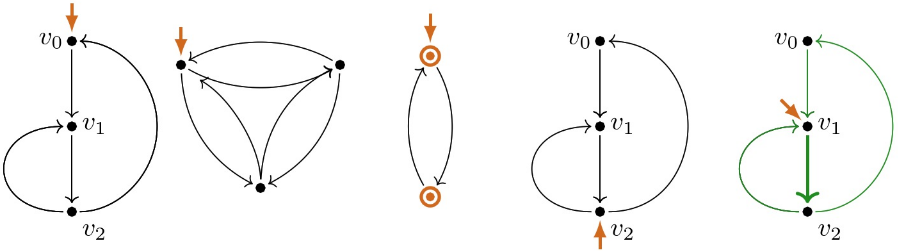
Definition

A chart is a loop chart if:

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

(L2) Every infinite path from the start vertex returns to it.

(L3) Termination is only possible at the start vertex.



(L1),~~(L2)~~ (L1),(L2),~~(L3)~~ loop chart loop chart

Loop charts (interpretations of innermost iterations)

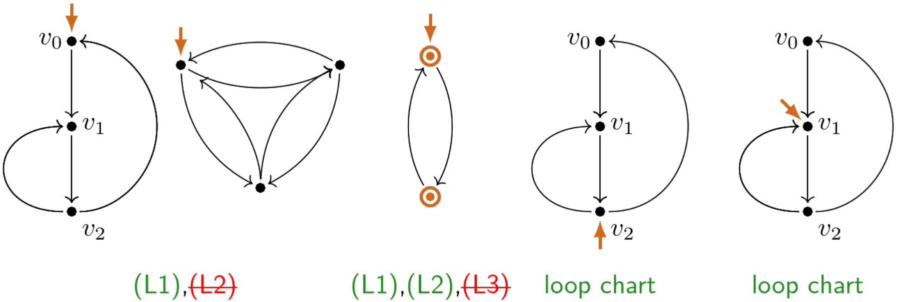
Definition

A chart is a loop chart if:

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

(L2) Every infinite path from the start vertex returns to it.

(L3) Termination is only possible at the start vertex.



Loop charts (interpretations of innermost iterations)

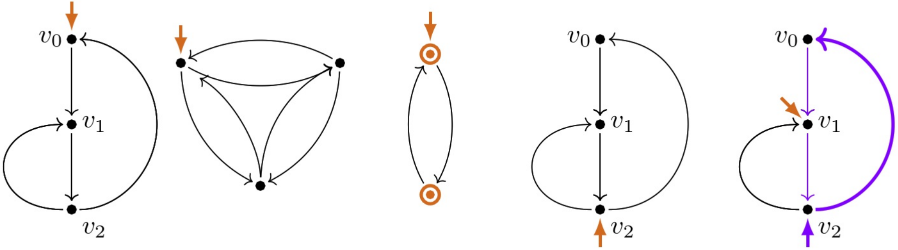
Definition

A chart is a loop chart if:

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

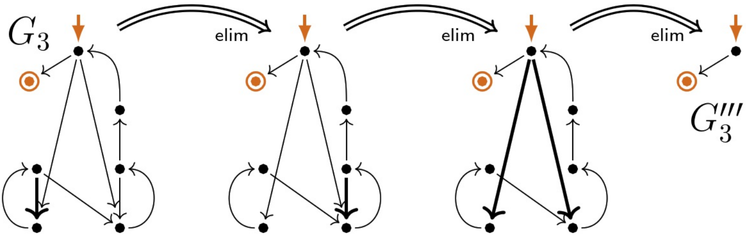
(L2) Every infinite path from the start vertex returns to it.

(L3) Termination is only possible at the start vertex.



(L1),~~(L2)~~ (L1),(L2),~~(L3)~~ loop chart loop subchart

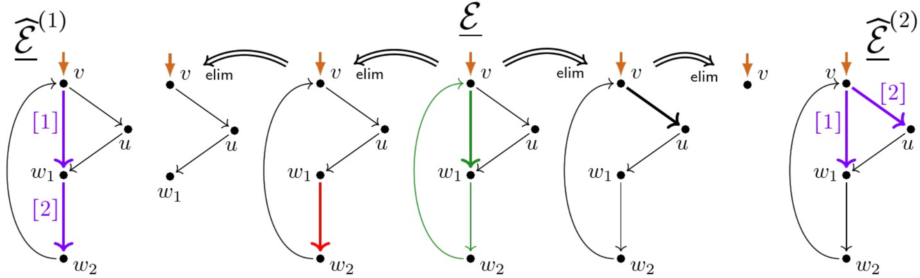
Layered Loop Existence and Elimination (LLEE)



LLEE-chart

LLEE: loop subcharts not eliminated
from bodies of previously eliminated loop subcharts

LEE-witness / layered LEE-witness



LEE-witness

LLEE-witness
layered LEE-witness

Deciding (L)LEE

Proposition

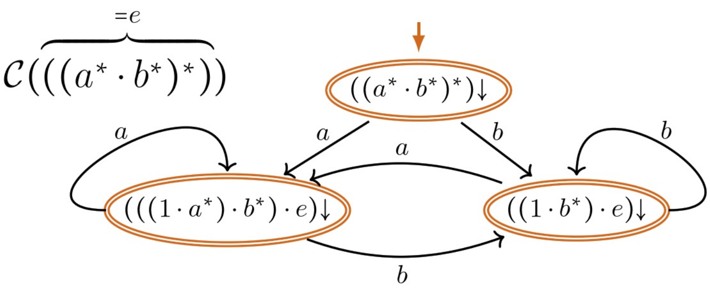
A 1-chart $\underline{\mathcal{C}}$ satisfies LEE if and only if it satisfies LLEE.

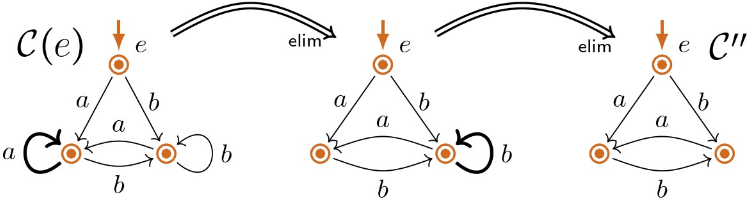
Deciding-(L)LEE
Instance: A 1-chart $\underline{\mathcal{C}}$.
Question: Does $\underline{\mathcal{C}}$ satisfy LLEE?

Proposition

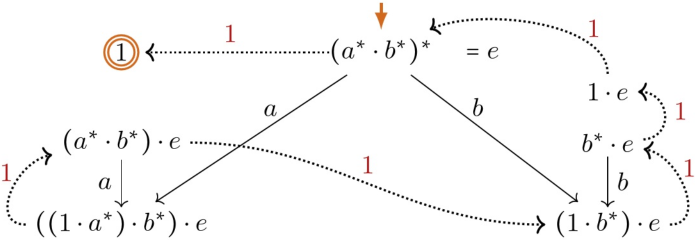
Deciding-(L)LEE $\in$ P.

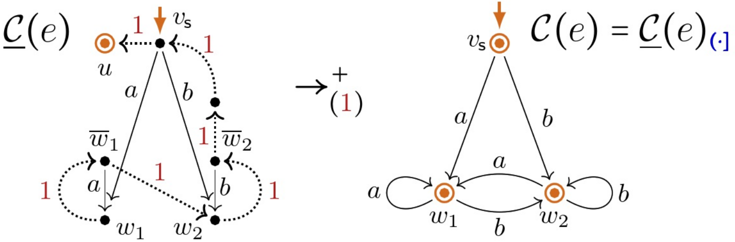
Process interpretations do not always satisfy LEE



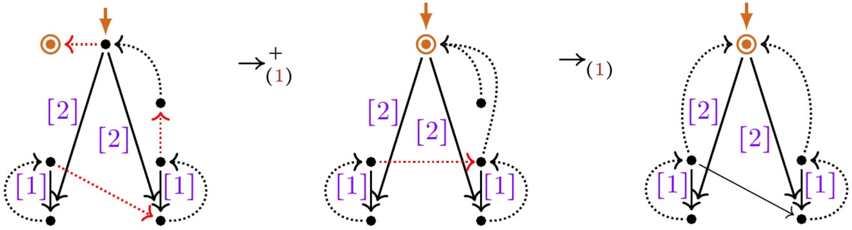


Process interpretations can be refined into LLEE-1-charts





1-Transition reduced LLEE-witnesses

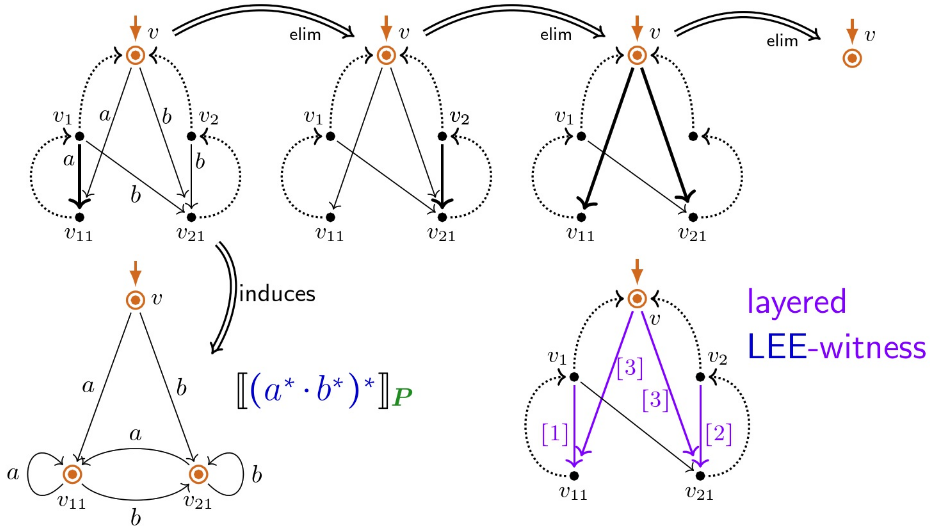


this LLEE-witness
is 1-transition reduced:
only backlinks
are 1-transitions

Lemma

Every LLEE-1-chart $\underline{\mathcal{C}}$ 1-transition refines a LLEE-1-chart $\underline{\mathcal{C}}_r$ that is 1-transition reduced, and it holds $\underline{\mathcal{C}} \to^{*}_{(1)} \underline{\mathcal{C}}_r$.

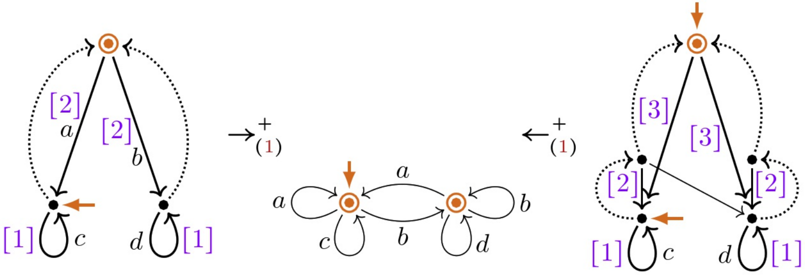
LEE, and LLEE-witness, induced process graph



Deciding refinability into a LLEE-1-chart

A 1-chart $\underline{\mathcal{C}}$ is 1-transition refinable into a 1-chart $\underline{\mathcal{C}}'$
if $\underline{\mathcal{C}}' \to^{*}_{(1)} \underline{\mathcal{C}}$ (that is, $\underline{\mathcal{C}}$ arises by 1-transition elimination steps from $\underline{\mathcal{C}}'$).

REFINABILITY-INTO-LLEE-1-CHART
Instance: A 1-chart $\underline{\mathcal{C}}$.
Question: Can $\underline{\mathcal{C}}$ be 1-transition refined into a 1-chart with LLEE?



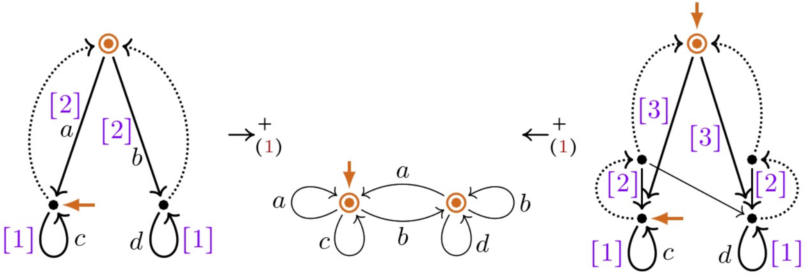
Deciding refinability into a LLEE-1-chart

A 1-chart $\underline{\mathcal{C}}$ is 1-transition refinable into a 1-chart $\underline{\mathcal{C}}'$
if $\underline{\mathcal{C}}' \rightarrow^{*}_{(1)} \underline{\mathcal{C}}$ (that is, $\underline{\mathcal{C}}$ arises by 1-transition elimination steps from $\underline{\mathcal{C}}'$).

REFINABILITY-INTO-LLEE-1-CHART

Instance: A 1-chart $\underline{\mathcal{C}}$.

Question: Can $\underline{\mathcal{C}}$ be 1-transition refined into a 1-chart with LLEE?



Proposition

REFINABILITY-INTO-LLEE-1-CHART $\in$ P.

Expressibility problem

A chart $\mathcal{C}$ is called expressible by a regular expression modulo bisimilarity if $\mathcal{C}$ is bisimilar to the process interpretation of a regular expression.

EXPRESSIBILITY-MODULO-BISIMILARITY

Instance: A chart $\mathcal{C}$ (finite process graph).

Question: Is $\mathcal{C}$ expressible by a regular expression modulo bisimilarity?

Expressibility problem

A chart $\mathcal{C}$ is called expressible by a regular expression modulo bisimilarity
if $\mathcal{C}$ is bisimilar to the process interpretation of a regular expression.

EXPRESSIBILITY-MODULO-BISIMILARITY
Instance: A chart $\mathcal{C}$ (finite process graph).
Question: Is $\mathcal{C}$ expressible by a regular expression modulo bisimilarity?

Lemma

If a chart $\mathcal{C}$ is refinable into a LLEE-1-chart,

$\Longrightarrow$

$\mathcal{C}$ is expressible by a regular expression modulo bisimilarity.

Expressibility problem

A chart $\mathcal{C}$ is called expressible by a regular expression modulo bisimilarity
if $\mathcal{C}$ is bisimilar to the process interpretation of a regular expression.

EXPRESSIBILITY-MODULO-BISIMILARITY
Instance: A chart $\mathcal{C}$ (finite process graph).
Question: Is $\mathcal{C}$ expressible by a regular expression modulo bisimilarity?

Lemma

If a chart $\mathcal{C}$ is refinable into a LLEE-1-chart,

$\Longrightarrow$

$\mathcal{C}$ is expressible by a regular expression modulo bisimilarity.

Theorem (Baeten–Corradini–G, 2007)

EXPRESSIBILITY-MODULO-BISIMILARITY *is decidable*
(yet by a (highly) super-exponential decision procedure).

Expandability into crystallized LLEE-1-chart

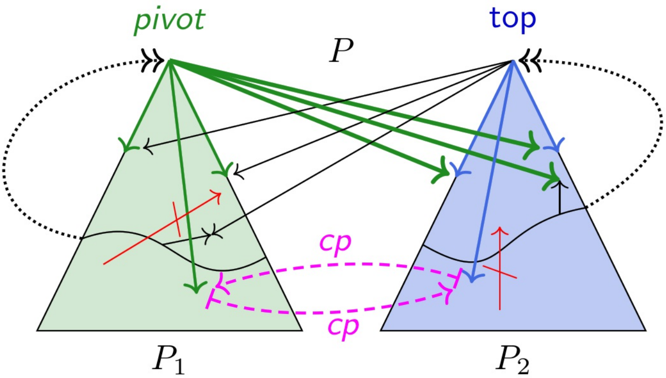
Theorem (follows from LICS'22 article)

A chart $\mathcal{C}$ is expressible by a regular expression modulo bisimilarity

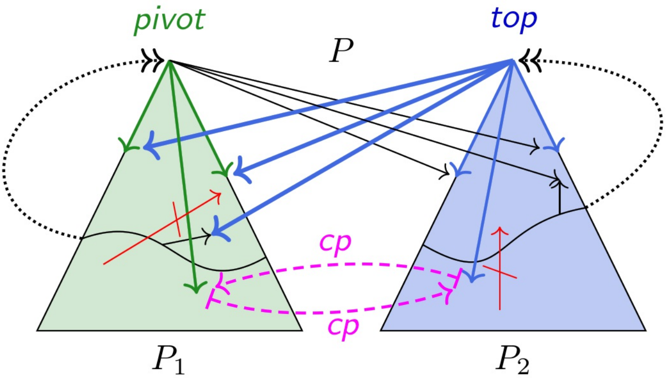
$\Longleftrightarrow$

the bisimulation collapse $\underline{\mathcal{C}_0}$ of $\underline{\mathcal{C}}$
can be expanded into a crystallized LLEE-1-chart $\mathcal{C}_{0,\mathsf{ref}}$
($\mathcal{C}_0$ results from $\mathcal{C}_{0,\mathsf{ref}}$ by 'connect-through' and 1-transition elim. steps).

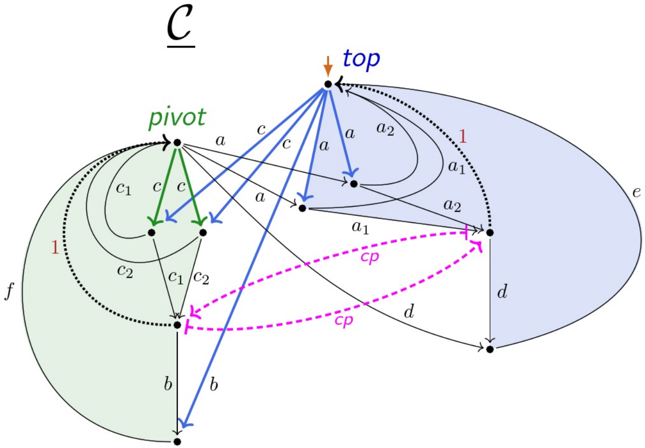
Twin-Crystal



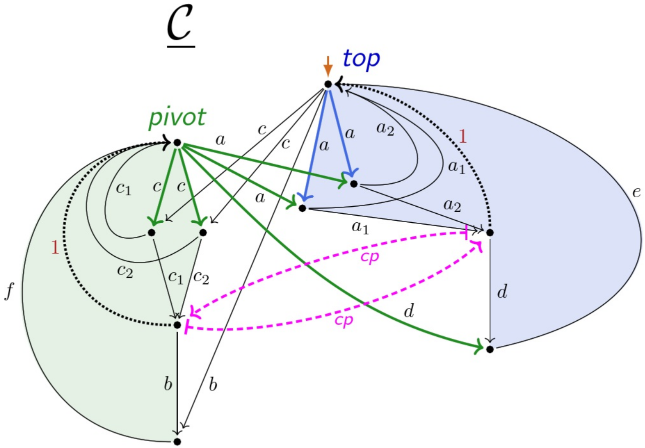
Twin-Crystal



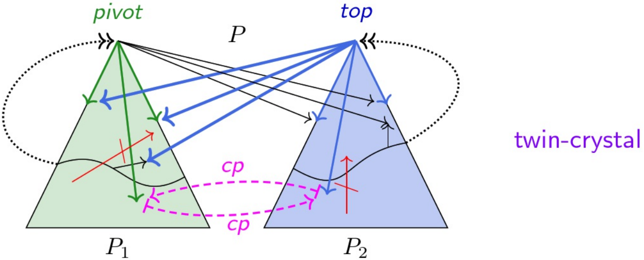
Twin-Crystal



Twin-Crystal



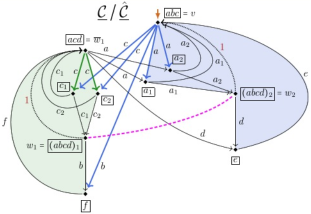
Crystallization



Crystallized 1-charts = LLEE-1-charts that are collapsed apart from some scc's that are twin-crystals.

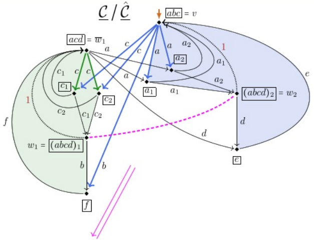
(CR) Crystallization: Every LLEE-1-chart can be reduced under bisimilarity to a bisimilar crystallized 1-chart.

1-Collapses and Bisimulation Collapse of Twin-Crystal

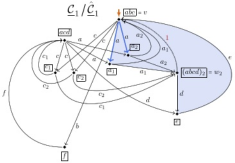


Twin-Crystal

1-Collapses and Bisimulation Collapse of Twin-Crystal

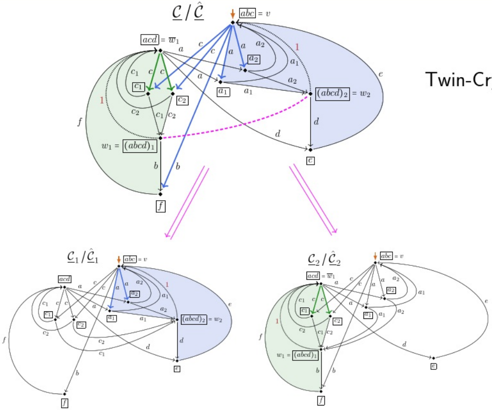


Twin-Crystal



1-Collapses

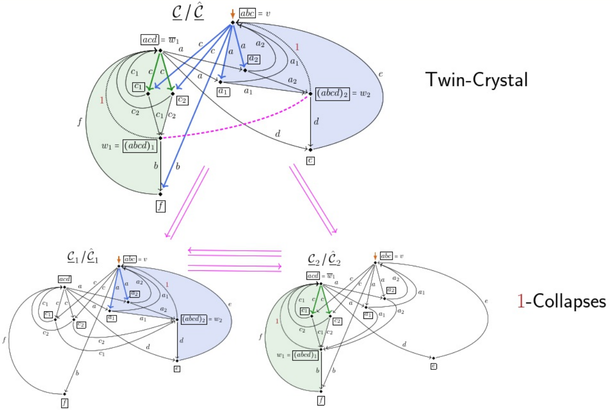
1-Collapses and Bisimulation Collapse of Twin-Crystal



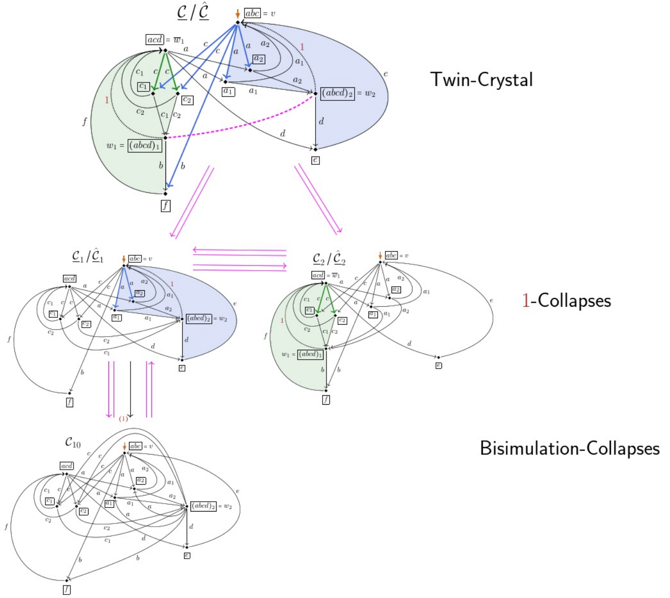
Twin-Crystal

1-Collapses

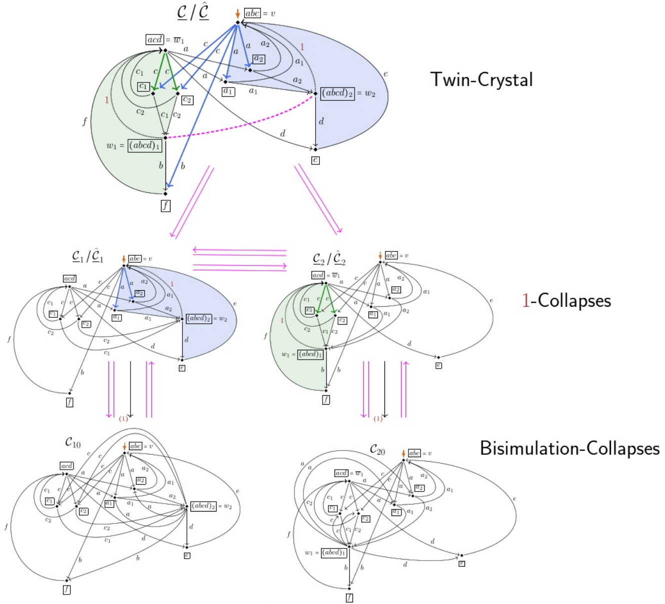
1-Collapses and Bisimulation Collapse of Twin-Crystal



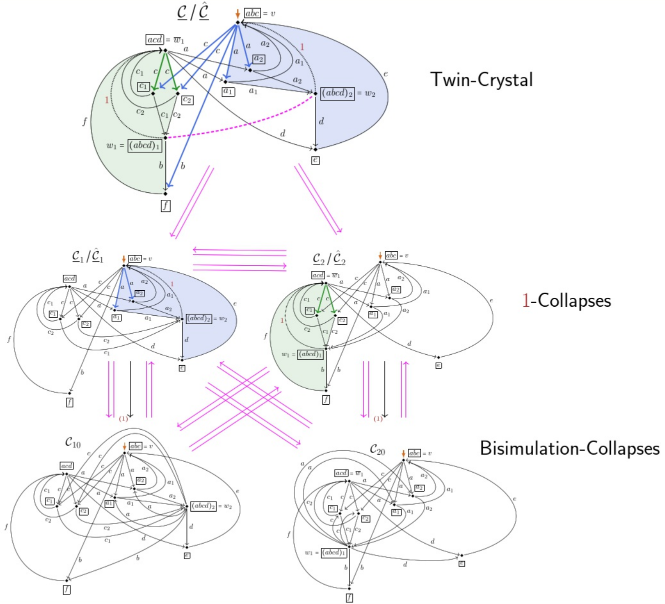
1-Collapses and Bisimulation Collapse of Twin-Crystal



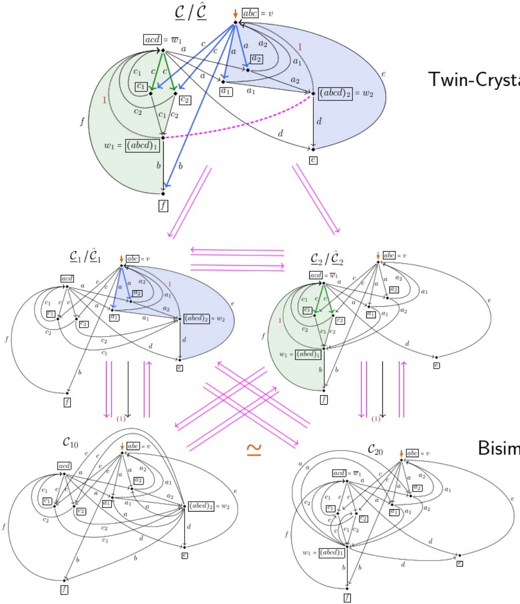
1-Collapses and Bisimulation Collapse of Twin-Crystal



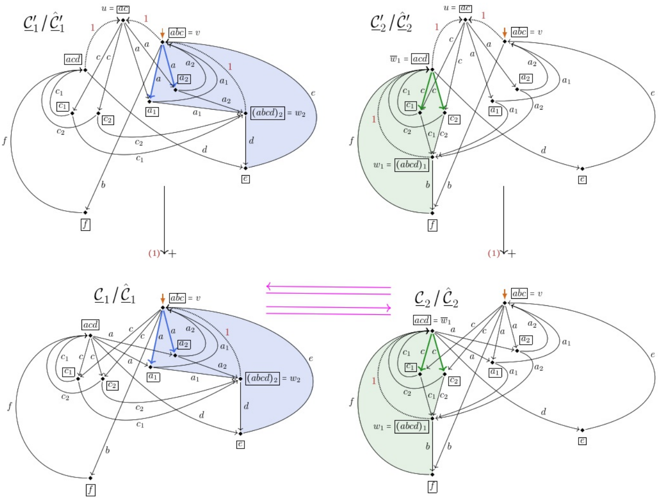
1-Collapses and Bisimulation Collapse of Twin-Crystal



1-Collapses and Bisimulation Collapse of Twin-Crystal



Not 1-transition refinable into LLEE-1-chart



Expandability into crystallized LLEE-1-chart

Theorem (follows from LICS'22 article)

A chart $\mathcal{C}$ is expressible by a regular expression modulo bisimilarity

$\Longleftrightarrow$

the bisimulation collapse $\underline{\mathcal{C}_0}$ of $\underline{\mathcal{C}}$

can be expanded into a crystallized LLEE-1-chart $\mathcal{C}_{0,\mathrm{ref}}$

($\mathcal{C}_0$ results from $\mathcal{C}_{0,\mathrm{ref}}$ by 'connect-through' and 1-transition elim. steps).

Expandability into crystallized LLEE-1-chart

Theorem (follows from LICS'22 article)

A chart $\mathcal{C}$ is expressible by a regular expression modulo bisimilarity

$\Longleftrightarrow$

the bisimulation collapse $\underline{\mathcal{C}_0}$ of $\underline{\mathcal{C}}$
can be expanded into a crystallized LLEE-1-chart $\mathcal{C}_{0,\mathsf{ref}}$
($\mathcal{C}_0$ results from $\mathcal{C}_{0,\mathsf{ref}}$ by 'connect-through' and 1-transition elim. steps).

EXPANDABILITY-INTO-CRYSTALLIZED-LLEE-1-CHART
Instance: A bisimulation-collapsed chart $\mathcal{C}$.
Question: Can $\mathcal{C}$ be expanded into a crystallized LLEE-1-chart?

Expandability into crystallized LLEE-1-chart

Theorem (follows from LICS'22 article)

A chart $\mathcal{C}$ is expressible by a regular expression modulo bisimilarity

$\Longleftrightarrow$

the bisimulation collapse $\underline{\mathcal{C}}_0$ of $\underline{\mathcal{C}}$
can be expanded into a crystallized LLEE-1-chart $\mathcal{C}_{0,\text{ref}}$
($\mathcal{C}_0$ results from $\mathcal{C}_{0,\text{ref}}$ by 'connect-through' and 1-transition elim. steps).

EXPANDABILITY-INTO-CRYSTALLIZED-LLEE-1-CHART

Instance: A bisimulation-collapsed chart $\mathcal{C}$.

Question: Can $\mathcal{C}$ be expanded into a crystallized LLEE-1-chart?

Question

EXPANDABILITY-INTO-CRYSTALLIZED-LLEE-1-CHART $\in$ P ?

Expandability into crystallized LLEE-1-chart

Theorem (follows from LICS'22 article)

A chart $\mathcal{C}$ is expressible by a regular expression modulo bisimilarity

$\Longleftrightarrow$

the bisimulation collapse $\underline{\mathcal{C}_0}$ of $\underline{\mathcal{C}}$
can be expanded into a crystallized LLEE-1-chart $\mathcal{C}_{0,\text{ref}}$
($\mathcal{C}_0$ results from $\mathcal{C}_{0,\text{ref}}$ by 'connect-through' and 1-transition elim. steps).

EXPANDABILITY-INTO-CRYSTALLIZED-LLEE-1-CHART
Instance: A bisimulation-collapsed chart $\mathcal{C}$.
Question: Can $\mathcal{C}$ be expanded into a crystallized LLEE-1-chart?

Question

EXPANDABILITY-INTO-CRYSTALLIZED-LLEE-1-CHART $\in$ P ?

Conjecture

p-EXPANDABILITY-INTO-CRYSTALLIZED-LLEE-1-CHART $\in$ FPT,
with the maximum outdegree of vertices of $\underline{\mathcal{C}}$ as parameter.

Aims and questions

Articles

- motivation of crystallization
- crystallization procedure

Tool implementation

- first step: efficiently deciding refinability into a LLEE-1-chart
- second step (envisaged):
  - deciding expandability
    of a given collapsed process graph into a crystallized LLEE-1-chart

Questions

- relation with attribute grammars?
- examples, where efficient local manipulation or evaluation of process graphs with twisted sharing is used/would be advantageous?

Summary

Forms of sharing (in term graphs / rooted directed graphs)

- horizontal, vertical, and twisted sharing
- twisted ⇒ only vertical sharing: exponential size increase possible

Expressibility of process graphs by regular expressions

- loop-elimination property LEE
  - guarantees expressibility by a regular expression
- process interpretations of reg. expressions do not always satisfy LEE
  - but can be 1-transition refined to guarantee LEE
- expressibility problem of process graphs by regular expressions
- steps to solve the expressibility problem (perhaps) efficiently

Questions