Closing the Image of the Process Interpretation of 1-Free Regular Expressions under Bisim. Collapse

Clemens Grabmayer

https://clegra.github.io

Department of Computer Science

GRAN SASSO SCIENCE INSTITUTE

SCHOOL OF ADVANCED STUDIES

Scuola Universitaria Superiore

L'Aquila, Italy

Computer Science Seminar

University of Novi Sad

March 2026

Overview

- 1-free regular expressions (with unary/binary star)
- process interpretation / semantics of regular expressions
  - expressible/not expressible process graphs
- loop existence and elimination property (LEE) (G/Fokkink, 2020)
  - interpretation/extraction correspondences with 1-free reg. expr's
  - LEE is preserved under bisimulation collapse
- Q: Image of process interpretation of 1-free regular expressions closed under bisimulation collapse?
  - A1: No. — But ...
- compact process interpretation
- refined expression extraction
  - A2: compact process interpretation is image-closed under collapse
- outlook: consequences

Regular Expressions

Definition (~ *Copi–Elgot–Wright, 1958*)

Regular expressions over alphabet A with unary Kleene star:

$$e, e_1, e_2 \ ::= \ \mathbf{0} \ \mid \ \boldsymbol{a} \ \mid \ e_1 + e_2 \ \mid \ e_1 \cdot e_2 \ \mid \ e^{*} \qquad \text{(for } \boldsymbol{a} \in A\text{).}$$

Regular Expressions

Definition (~ *Copi–Elgot–Wright, 1958*)

Regular expressions over alphabet A with unary Kleene star:

$$e, e_1, e_2 \;::=\; \mathbf{0} \;\mid\; a \;\mid\; e_1 + e_2 \;\mid\; e_1 \cdot e_2 \;\mid\; e^{*} \qquad (\text{for } a \in A).$$

- symbol $\mathbf{0}$ instead of $\varnothing$, symbol $\mathbf{1}$ instead of $\{\epsilon\}$
- with unary star $*$: $\mathbf{1}$ is definable as $\mathbf{0}^{*}$

Regular Expressions

Definition (~ *Kleene, 1951*, ~ *Copi–Elgot–Wright, 1958*)

Regular expressions over alphabet A with unary / binary Kleene star:

$$e, e_1, e_2 \;::=\; \mathbf{0} \mid \boldsymbol{a} \mid e_1 + e_2 \mid e_1 \cdot e_2 \mid e^{*} \qquad (\text{for } \boldsymbol{a} \in A).$$

$$e, e_1, e_2 \;::=\; \mathbf{0} \mid \mathbf{1} \mid \boldsymbol{a} \mid e_1 + e_2 \mid e_1 \cdot e_2 \mid e_1{}^{\circledast} e_2 \qquad (\text{for } \boldsymbol{a} \in A).$$

- symbol $\mathbf{0}$ instead of $\varnothing$, symbol $\mathbf{1}$ instead of $\{\epsilon\}$
- with unary star *: $\mathbf{1}$ is definable as $\mathbf{0}^{*}$
- with binary star $^{\circledast}$: $\mathbf{1}$ is not definable (in its absence)

Regular Expressions

Definition (~ *Kleene, 1951*, ~ *Copi–Elgot–Wright, 1958*)

Regular expressions over alphabet A with unary / binary Kleene star:

$$e, e_1, e_2 ::= \mathbf{0} \mid \mathbf{1} \mid \boldsymbol{a} \mid e_1 + e_2 \mid e_1 \cdot e_2 \mid e^{*} \qquad \text{(for } \boldsymbol{a} \in A\text{).}$$

$$e, e_1, e_2 ::= \mathbf{0} \mid \mathbf{1} \mid \boldsymbol{a} \mid e_1 + e_2 \mid e_1 \cdot e_2 \mid e_1{}^{\circledast} e_2 \qquad \text{(for } \boldsymbol{a} \in A\text{).}$$

- symbol $\mathbf{0}$ instead of $\emptyset$, symbol $\mathbf{1}$ instead of $\{\epsilon\}$
- with unary star $*$: $\mathbf{1}$ is definable as $\mathbf{0}^{*}$
- with binary star $\circledast$: $\mathbf{1}$ is not definable (in its absence)

Regular Expressions

Definition (~ *Kleene, 1951*, ~ *Copi–Elgot–Wright, 1958*)

Regular expressions over alphabet A with unary / binary Kleene star:

$$e, e_1, e_2 \ ::= \ \mathbf{0} \mid \mathbf{1} \mid \boldsymbol{a} \mid e_1 + e_2 \mid e_1 \cdot e_2 \mid e^{*} \qquad (\text{for } \boldsymbol{a} \in A).$$

$$e, e_1, e_2 \ ::= \ \mathbf{0} \mid \mathbf{1} \mid \boldsymbol{a} \mid e_1 + e_2 \mid e_1 \cdot e_2 \mid e_1{}^{\circledast} e_2 \qquad (\text{for } \boldsymbol{a} \in A).$$

- symbol $\mathbf{0}$ instead of $\varnothing$, symbol $\mathbf{1}$ instead of $\{\epsilon\}$
- with unary star *: $\mathbf{1}$ is definable as $\mathbf{0}^{*}$
- with binary star ${}^{\circledast}$: $\mathbf{1}$ is not definable (in its absence)

Definition

1-free regular expressions over alphabet A with binary Kleene star:

$$f, f_1, f_2 \ ::= \ \mathbf{0} \mid \boldsymbol{a} \mid f_1 + f_2 \mid f_1 \cdot f_2 \mid f_1{}^{\circledast} f_2 \qquad (\text{for } \boldsymbol{a} \in A).$$

Regular Expressions

Definition (~ *Kleene, 1951*, ~ *Copi–Elgot–Wright, 1958*)

Regular expressions over alphabet A with unary / binary Kleene star:

$$e, e_1, e_2 \;::=\; \mathbf{0} \mid \mathbf{1} \mid \boldsymbol{a} \mid e_1 + e_2 \mid e_1 \cdot e_2 \mid e^{*} \qquad (\text{for } \boldsymbol{a} \in A).$$

$$e, e_1, e_2 \;::=\; \mathbf{0} \mid \mathbf{1} \mid \boldsymbol{a} \mid e_1 + e_2 \mid e_1 \cdot e_2 \mid e_1{}^{\circledast} e_2 \qquad (\text{for } \boldsymbol{a} \in A).$$

- symbol $\mathbf{0}$ instead of $\varnothing$, symbol $\mathbf{1}$ instead of $\{\epsilon\}$
- with unary star *: $\mathbf{1}$ is definable as $\mathbf{0}^{*}$
- with binary star $^{\circledast}$: $\mathbf{1}$ is not definable (in its absence)

Definition

1-free regular expressions over alphabet A with unary/binary Kleene star:

$$f, f_1, f_2 \;::=\; \mathbf{0} \mid \boldsymbol{a} \mid f_1 + f_2 \mid f_1 \cdot f_2 \mid (f_1{}^{*}) \cdot f_2 \qquad (\text{for } \boldsymbol{a} \in A),$$

$$f, f_1, f_2 \;::=\; \mathbf{0} \mid \boldsymbol{a} \mid f_1 + f_2 \mid f_1 \cdot f_2 \mid f_1{}^{\circledast} f_2 \qquad (\text{for } \boldsymbol{a} \in A).$$

Under-Star-/1-Free regular expressions

Definition

The set $\mathit{RExp}^{(\cancel{1})}(A)$ of 1-free regular expressions over A is defined by:

$$f,\, f_1,\, f_2 \;::=\; 0 \mid a \mid f_1 + f_2 \mid f_1 \cdot f_2 \mid {f_1}^{\star} \cdot f_2 \qquad \text{(for } a \in A\text{)},$$

the set $\mathit{RExp}^{(\cancel{1}\backslash *)}(A)$ of under-star-1-free regular expressions over A by:

$$uf,\, uf_1,\, uf_2 \;::=\; 0 \mid 1 \mid a \mid uf_1 + uf_2 \mid uf_1 \cdot uf_2 \mid f^{\star} \qquad \text{(for } a \in A\text{)}.$$

Under-Star-/1-Free regular expressions

Definition

The set $RExp^{(\neq 1)}(A)$ of 1-free regular expressions over A is defined by:

$$f, f_1, f_2 ::= 0 \mid a \mid f_1 + f_2 \mid f_1 \cdot f_2 \mid {f_1}^{\star} \cdot f_2 \qquad \text{(for } a \in A\text{)},$$

the set $RExp^{(\neq 1 \backslash *)}(A)$ of under-star-1-free regular expressions over A by:

$$uf, uf_1, uf_2 ::= 0 \mid 1 \mid a \mid uf_1 + uf_2 \mid uf_1 \cdot uf_2 \mid f^{\star} \qquad \text{(for } a \in A\text{)}.$$

Under the language interpretation, subclasses of minor relevance:

- 1-free regular expressions denote all regular languages without ϵ.
- Under-star-1-free regular expressions denote all regular languages.

Process interpretation $P(\cdot)$ of regular expressions *(Milner, 1984)*

$\mathbf{0} \xrightarrow{P}$ deadlock $\boldsymbol{\delta}$, no termination

$\mathbf{1} \xrightarrow{P}$ empty-step process $\boldsymbol{\epsilon}$, then terminate

$a \xrightarrow{P}$ atomic action a, then terminate

Process interpretation $P(\cdot)$ of regular expressions *(Milner, 1984)*

$$\mathbf{0} \xmapsto{P} \text{deadlock } \boldsymbol{\delta}\text{, no termination}$$

$$\mathbf{1} \xmapsto{P} \text{empty-step process } \boldsymbol{\epsilon}\text{, then terminate}$$

$$a \xmapsto{P} \text{atomic action } a\text{, then terminate}$$

$$e_1 + e_2 \xmapsto{P} (\textit{choice}) \text{ execute } P(e_1) \text{ or } P(e_2)$$

$$e_1 \cdot e_2 \xmapsto{P} (\textit{sequentialization}) \text{ execute } P(e_1)\text{, then } P(e_2)$$

$$e^* \xmapsto{P} (\textit{iteration}) \text{ repeat (terminate or execute } P(e)\text{)}$$

Process interpretation $P(\cdot)$ of regular expressions *(Milner, 1984)*

$\mathbf{0} \xmapsto{P}$ deadlock $\boldsymbol{\delta}$, no termination

$\mathbf{1} \xmapsto{P}$ empty-step process $\boldsymbol{\epsilon}$, then terminate

$a \xmapsto{P}$ atomic action a, then terminate

$e_1 + e_2 \xmapsto{P}$ (*choice*) execute $P(e_1)$ or $P(e_2)$

$e_1 \cdot e_2 \xmapsto{P}$ (*sequentialization*) execute $P(e_1)$, then $P(e_2)$

$e^* \xmapsto{P}$ (*iteration*) repeat (terminate or execute $P(e)$)

$e_1{}^{\circledast} e_2 \xmapsto{P}$ (*iteration-exit*) repeat (terminate or execute $P(e_1)$), then $P(e_2)$

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

$\mathbf{0} \overset{P}{\longmapsto}$ deadlock $\boldsymbol{\delta}$, no termination

$\mathbf{1} \overset{P}{\longmapsto}$ empty-step process ϵ, then terminate

$a \overset{P}{\longmapsto}$ atomic action a, then terminate

$e_1 + e_2 \overset{P}{\longmapsto}$ (*choice*) execute $P(e_1)$ or $P(e_2)$

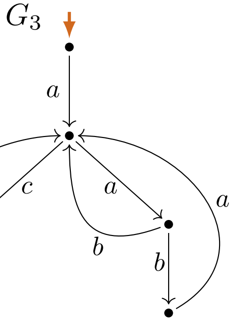
$e_1 \cdot e_2 \overset{P}{\longmapsto}$ (*sequentialization*) execute $P(e_1)$, then $P(e_2)$

$e^* \overset{P}{\longmapsto}$ (*iteration*) repeat (terminate or execute $P(e)$)

$e_1{}^{\circledast} e_2 \overset{P}{\longmapsto}$ (*iteration-exit*) repeat (terminate or execute $P(e_1)$), then $P(e_2)$

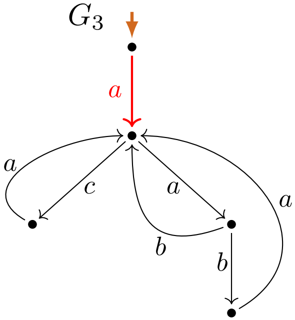
$[\![e]\!]_P \;:=\; [P(e)]_{\underline{\leftrightarrow}}$ (bisimilarity equivalence class of process $P(e)$)

P-expressibility and $[\![\cdot]\!]_P$-expressibility (examples)



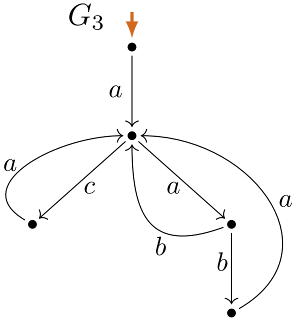
$$P\Big(\overbrace{(a\cdot(c\cdot a + a\cdot(b + b\cdot a))^*)\cdot 0}^{f}\Big)$$

P-expressibility and $[\![\cdot]\!]_P$-expressibility (examples)



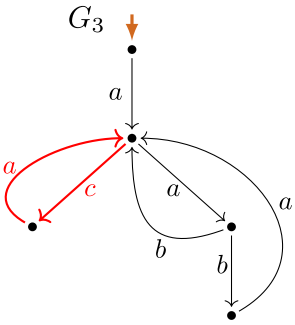
$$P\Big(\overbrace{(a \cdot (c \cdot a + a \cdot (b + b \cdot a))^*) \cdot 0}^{f}\Big)$$

P-expressibility and $[\![\cdot]\!]_P$-expressibility (examples)



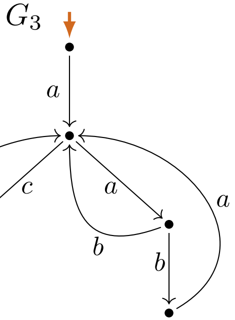
$$P\big(\overbrace{(a\cdot(c\cdot a+a\cdot(b+b\cdot a))^*)\cdot 0}^{f}\big)$$

P-expressibility and $[\![\cdot]\!]_P$-expressibility (examples)



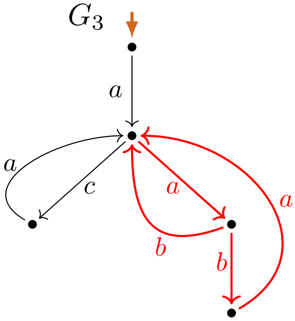
$$P\big(\overbrace{(a\cdot(c\cdot a + a\cdot(b + b\cdot a))^*)\cdot 0}^{f}\big)$$

P-expressibility and $[\![\cdot]\!]_P$-expressibility (examples)



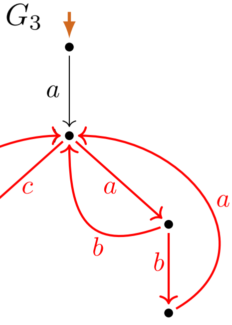
$$P\big(\overbrace{(a\cdot(c\cdot a + a\cdot(b + b\cdot a))^*)\cdot 0}^{f}\big)$$

P-expressibility and $[\![\cdot]\!]_P$-expressibility (examples)



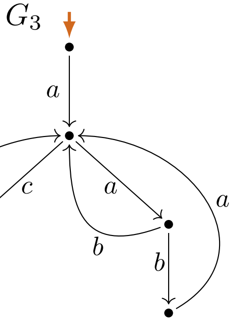
$$P(\overbrace{(a\cdot(c\cdot a + a\cdot(b + b\cdot a))^{*})\cdot 0}^{f})$$

P-expressibility and $[\![\cdot]\!]_P$-expressibility (examples)



$$P(\overbrace{(a \cdot (c \cdot a + a \cdot (b + b \cdot a))^*) \cdot 0}^{f})$$

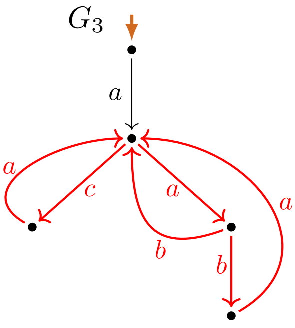
P-expressibility and $[\![\cdot]\!]_P$-expressibility (examples)



$$P\Big(\overbrace{(a\cdot(c\cdot a+a\cdot(b+b\cdot a))^*)\cdot 0}^{f}\Big)$$

$$P\Big(a\cdot(c\cdot a+a\cdot(b+b\cdot a))^{\circledast}0\Big)$$

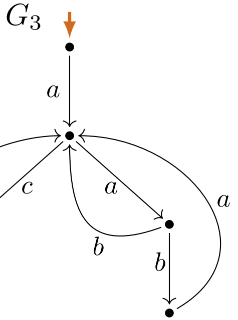
P-expressibility and $[\![\cdot]\!]_P$-expressibility (examples)



$$P\Big(\overbrace{(a\cdot(c\cdot a + a\cdot(b + b\cdot a))^{*})\cdot 0}^{f}\Big)$$

$$P\Big(a\cdot(c\cdot a + a\cdot(b + b\cdot a))^{\circledast}0\Big)$$

P-expressibility and $[\![\cdot]\!]_P$-expressibility (examples)

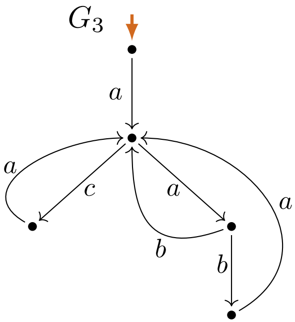


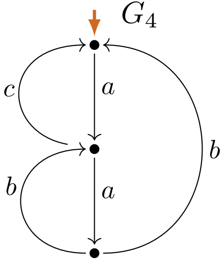
$$P\Big(\overbrace{(a\cdot(c\cdot a+a\cdot(b+b\cdot a))^*)\cdot 0}^{f}\Big)$$

$$P\Big(a\cdot(c\cdot a+a\cdot(b+b\cdot a))^{\circledast}0\Big)$$

$$G_3 \in [\![f]\!]_P$$

P-expressibility and $[\![\cdot]\!]_P$-expressibility (examples)



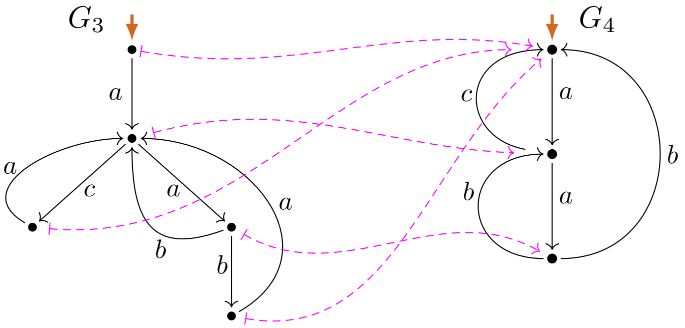


$$P\Big(\overbrace{(a\cdot(c\cdot a + a\cdot(b + b\cdot a))^*)\cdot 0}^{f}\Big)$$

$$P\Big(a\cdot(c\cdot a + a\cdot(b + b\cdot a))^{\circledast}0\Big)$$

$$G_3 \in [\![f]\!]_P$$

P-expressibility and $[\![\cdot]\!]_P$-expressibility (examples)

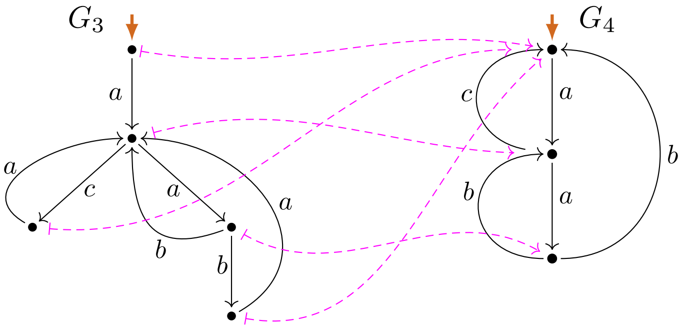


$$P\Big(\overbrace{(a \cdot (c \cdot a + a \cdot (b + b \cdot a))^*) \cdot 0}^{f}\Big)$$

$$P\Big(a \cdot (c \cdot a + a \cdot (b + b \cdot a))^{\circledast} 0\Big)$$

$$G_3 \in [\![f]\!]_P$$

P-expressibility and $[\![\cdot]\!]_P$-expressibility (examples)



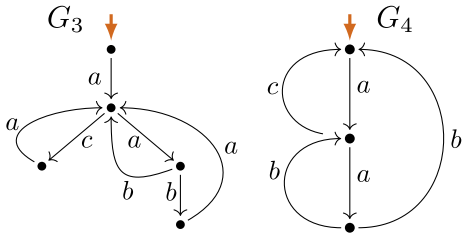
$$P\Big(\overbrace{(a\cdot(c\cdot a+a\cdot(b+b\cdot a))^{*})\cdot 0}^{f}\Big)$$

$$P\Big(a\cdot(c\cdot a+a\cdot(b+b\cdot a))^{\circledast}0\Big)$$

$$G_3 \in [\![f]\!]_P$$

$$G_4 \in [\![f]\!]_P$$

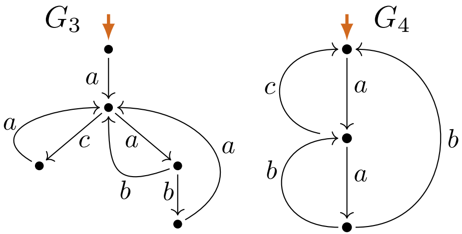
P-expressibility and $[\![\cdot]\!]_P$-expressibility (examples)



P-expressible

$[\![\cdot]\!]_P$-expressible $\quad$ $[\![\cdot]\!]_P$-expressible

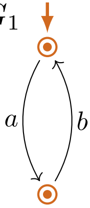
P-expressibility and $[\![\cdot]\!]_P$-expressibility (examples)



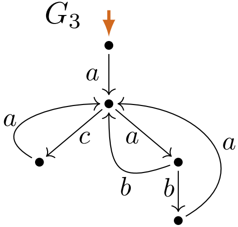
P-expressible	**?**
$[\![\cdot]\!]_P$-expressible	$[\![\cdot]\!]_P$-expressible

P-expressibility and $[\![\cdot]\!]_P$-expressibility (examples)

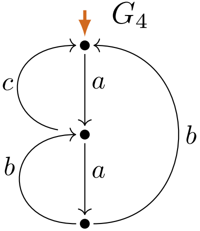
G_1



G_3



G_4



not P-expressible

not $[\![\cdot]\!]_P$-expressible

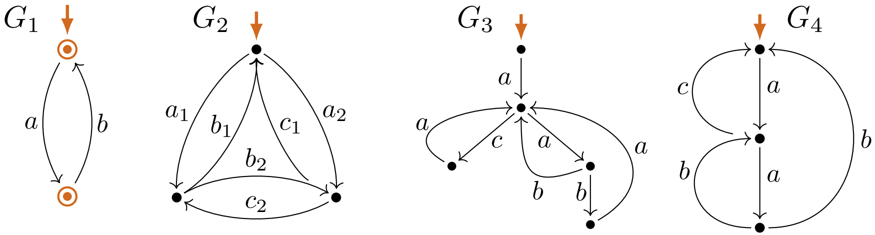
P-expressible

$[\![\cdot]\!]_P$-expressible

?

$[\![\cdot]\!]_P$-expressible

P-expressibility and $[\![\cdot]\!]_P$-expressibility (examples)



not P-expressible
not $[\![\cdot]\!]_P$-expressible

P-expressible
$[\![\cdot]\!]_P$-expressible

?
$[\![\cdot]\!]_P$-expressible

Process interpretation P (formal definition)

Definition (Transition system specification $\mathcal{T}$)

$$\frac{}{a \xrightarrow{a} 1} \qquad \frac{e_i \xrightarrow{a} e_i'}{e_1 + e_2 \xrightarrow{a} e_i'} \ (i \in \{1,2\})$$

Process interpretation P (formal definition)

Definition (Transition system specification $\mathcal{T}$)

$$\frac{}{a \xrightarrow{a} 1} \qquad \frac{e_i \xrightarrow{a} e_i'}{e_1 + e_2 \xrightarrow{a} e_i'} \; (i \in \{1,2\})$$

$$\frac{e \xrightarrow{a} e'}{e^* \xrightarrow{a} e' \cdot e^*}$$

Process interpretation P (formal definition)

Definition (Transition system specification $\mathcal{T}$)

$$\frac{}{\mathbf{1}{\Downarrow}} \qquad \frac{e_i{\Downarrow}}{(e_1 + e_2){\Downarrow}}\ (i \in \{1,2\}) \qquad \frac{e_1{\Downarrow} \qquad e_2{\Downarrow}}{(e_1 \cdot e_2){\Downarrow}} \qquad \frac{}{(e^*){\Downarrow}}$$

$$\frac{}{a \xrightarrow{a} \mathbf{1}} \qquad \frac{e_i \xrightarrow{a} e_i'}{e_1 + e_2 \xrightarrow{a} e_i'}\ (i \in \{1,2\})$$

$$\frac{e \xrightarrow{a} e'}{e^* \xrightarrow{a} e' \cdot e^*}$$

Process interpretation P (formal definition)

Definition (Transition system specification $\mathcal{T}$)

$$\frac{}{\mathbf{1}{\Downarrow}} \qquad \frac{e_i{\Downarrow}}{(e_1 + e_2){\Downarrow}}\,(i \in \{1,2\}) \qquad \frac{e_1{\Downarrow} \quad e_2{\Downarrow}}{(e_1 \cdot e_2){\Downarrow}} \qquad \frac{}{(e^*){\Downarrow}}$$

$$\frac{}{a \xrightarrow{a} \mathbf{1}} \qquad \frac{e_i \xrightarrow{a} e_i'}{e_1 + e_2 \xrightarrow{a} e_i'}\,(i \in \{1,2\})$$

$$\frac{e_1 \xrightarrow{a} e_1'}{e_1 \cdot e_2 \xrightarrow{a} e_1' \cdot e_2} \qquad \frac{e_1{\Downarrow} \quad e_2 \xrightarrow{a} e_2'}{e_1 \cdot e_2 \xrightarrow{a} e_2'} \qquad \frac{e \xrightarrow{a} e'}{e^* \xrightarrow{a} e' \cdot e^*}$$

Process interpretation P (formal definition)

Definition (Transition system specification $\mathcal{T}$)

$$\frac{}{\mathbf{1}\Downarrow} \qquad \frac{e_i\Downarrow}{(e_1 + e_2)\Downarrow}\,(i \in \{1,2\}) \qquad \frac{e_1\Downarrow \quad e_2\Downarrow}{(e_1 \cdot e_2)\Downarrow} \qquad \frac{}{(e^*)\Downarrow}$$

$$\frac{}{a \xrightarrow{a} \mathbf{1}} \qquad \frac{e_i \xrightarrow{a} e_i'}{e_1 + e_2 \xrightarrow{a} e_i'}\,(i \in \{1,2\})$$

$$\frac{e_1 \xrightarrow{a} e_1'}{e_1 \cdot e_2 \xrightarrow{a} e_1' \cdot e_2} \qquad \frac{e_1\Downarrow \quad e_2 \xrightarrow{a} e_2'}{e_1 \cdot e_2 \xrightarrow{a} e_2'} \qquad \frac{e \xrightarrow{a} e'}{e^* \xrightarrow{a} e' \cdot e^*}$$

Definition

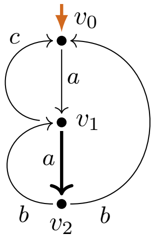
The process (graph) interpretation $P(e)$ of a regular expression e:

$P(e)$:= labeled transition graph generated by e by derivations in $\mathcal{T}$.

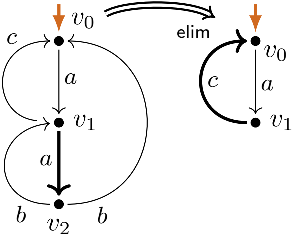
Overview

- 1-free regular expressions (with unary/binary star)
- process interpretation/semantics of regular expressions
  - expressible/not expressible process graphs
- loop existence and elimination property (LEE) (G/Fokkink, 2020)
  - interpretation/extraction correspondences with 1-free reg. expr's
  - LEE is preserved under bisimulation collapse
- Q: Image of process interpretation of 1-free regular expressions closed under bisimulation collapse?
  - A1: No. — But ...
- compact process interpretation
- refined expression extraction
  - A2: compact process interpretation is image-closed under collapse
- outlook: consequences

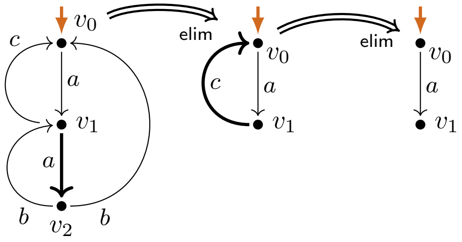
Loop existence and elimination



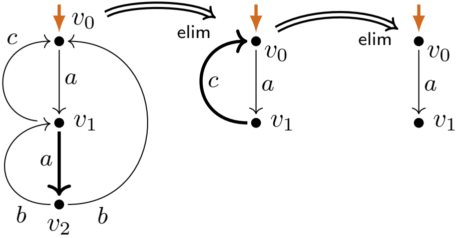
Loop existence and elimination

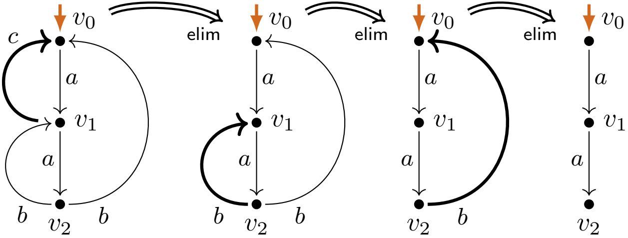


Loop existence and elimination



Loop existence and elimination





LEE

Definition

A chart $\mathcal{C}$ satisfies LEE (*loop existence and elimination*) if:

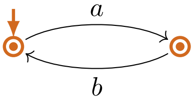
$$\exists \mathcal{C}_0 \left(\mathcal{C} \longrightarrow^{*}_{\mathsf{elim}} \mathcal{C}_0 \not\longrightarrow_{\mathsf{elim}} \right.$$
$$\left. \wedge\ \mathcal{C}_0 \text{ permits no infinite path} \right).$$

LEE

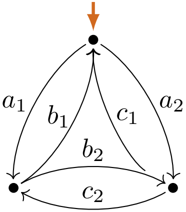
Definition

A chart $\mathcal{C}$ satisfies LEE (*loop existence and elimination*) if:

$$\exists \mathcal{C}_0 \Big(\mathcal{C} \longrightarrow^{*}_{\text{elim}} \mathcal{C}_0 \not\longrightarrow_{\text{elim}} \\ \wedge\ \mathcal{C}_0 \text{ permits no infinite path} \Big).$$

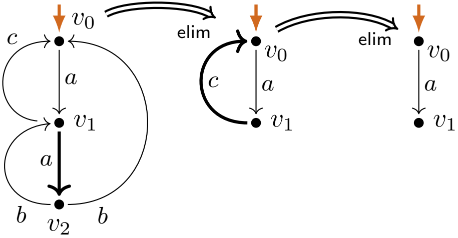


~~LEE~~



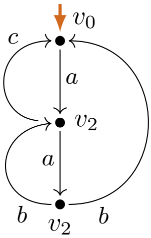
~~LEE~~

LEE

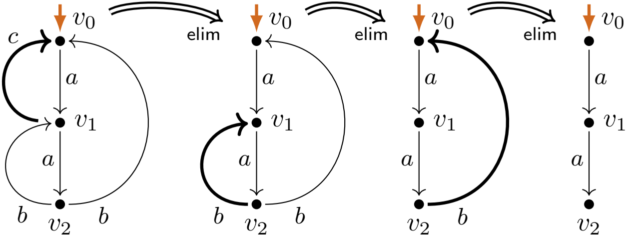


v_0
c
a
v_1
a
b
v_2
b
elim
v_0
c
a
v_1
elim
v_0
a
v_1

LEE

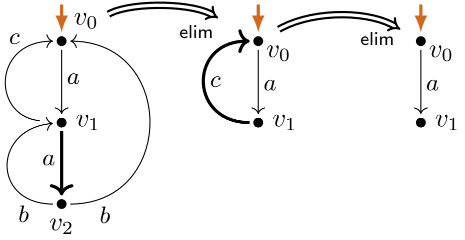


v_0
c
a
v_2
a
b
v_2
b

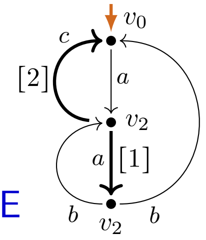


v_0
c
a
v_1
a
b
v_2
b
elim
v_0
a
v_1
a
b
v_2
b
elim
v_0
a
v_1
a
v_2
b
elim
v_0
a
v_1
a
v_2

LEE

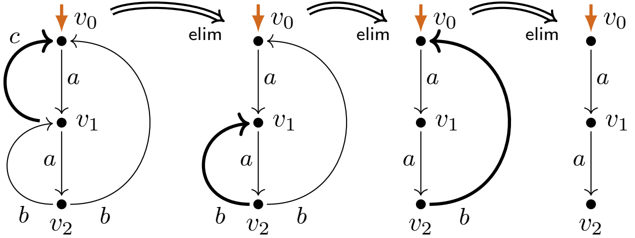


v_0
v_1
v_2
a
b
c
elim
elim

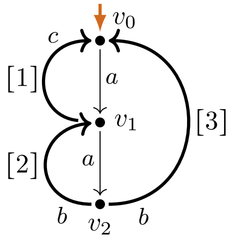


v_0
v_2
a
b
c
[1]
[2]

LEE

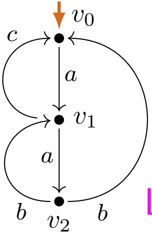


v_0
v_1
v_2
a
b
c
elim
elim
elim



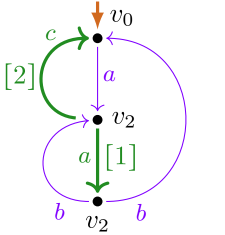
v_0
v_1
v_2
a
b
c
[1]
[2]
[3]

LEE witness and LEE-charts

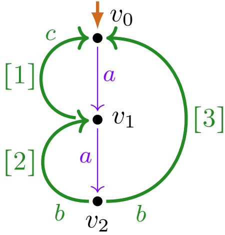


LEE-chart

LEE-witness



LEE-witness



Properties of LEE-charts

Theorem ($\Leftarrow$ G/Fokkink, 2020)

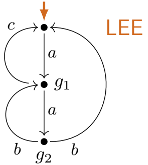
A process graph G
is $[\![\cdot]\!]_P$-expressible by an under-star-1-free regular expression
(i.e. P-expressible modulo bisimilarity by an $(\cancel{1}\backslash *)$ reg. expr.)
if and only if
the bisimulation collapse of G satisfies LEE.

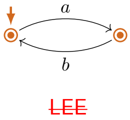
Properties of LEE-charts

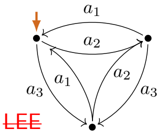
Theorem (⇐ G/Fokkink, 2020)

A process graph G
is $[\![\cdot]\!]_P$-expressible by an under-star-1-free regular expression
(i.e. P-expressible modulo bisimilarity by an $(\cancel{1}\backslash *)$ reg. expr.)
if and only if
the bisimulation collapse of G satisfies LEE.

Hence $[\![\cdot]\!]_P$-expressible | **not** $[\![\cdot]\!]_P$-expressible by 1-free regular expressions:







Interpretation/extraction correspondences with LEE

($\Leftarrow$ G/Fokkink 2020, G 2021)

(Int)$_P^{(\cancel{1}\backslash *)}$: $P^{\bullet}$-$(\cancel{1}\backslash *)$*-expressible graphs have structural property* LEE

Process **i**nterpretations $P(e)$
of under-star-1-free regular expressions e
are finite process graphs that satisfy LEE.

Interpretation/extraction correspondences with LEE

($\Leftarrow$ G/Fokkink 2020, G 2021)

(Int)$_{P}^{(\cancel{1}\backslash *)}$: *$P^{\bullet}$-$(\cancel{1}\backslash *)$-expressible graphs have structural property* LEE

Process **i**nterpretations $P(e)$
of under-star-1-free regular expressions e
are finite process graphs that satisfy LEE.

(Extr)$_{P}$: LEE *implies* $[\![\cdot]\!]_{P}$*-expressibility*

From every finite process graph G with LEE
a regular expression e can be **e**xtracted
such that $G \underline{\leftrightarrow} P(e)$.

Interpretation/extraction correspondences with LEE

($\Leftarrow$ G/Fokkink 2020, G 2021)

(Int)$_{P}^{(\cancel{1}\backslash *)}$: *$P^{\bullet}$-$(\cancel{1}\backslash *)$-expressible graphs have structural property LEE*

Process **i**nterpretations $P(e)$
of under-star-1-free regular expressions e
are finite process graphs that satisfy LEE.

(Extr)$_{P}$: LEE *implies* $[\![\cdot]\!]_{P}$*-expressibility*

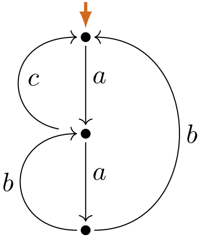
From every finite process graph G with LEE
a regular expression e can be **e**xtracted
such that $G \leftrightarrow P(e)$.

(Coll): LEE *is preserved under collapse*

The class of finite process graphs with LEE
is closed under bisimulation **c**ollapse.

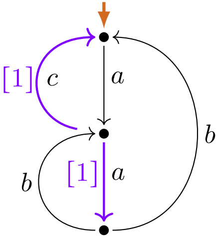
Expression extraction using LEE (G/Fokkink 2020, G 2021/22)

G_4



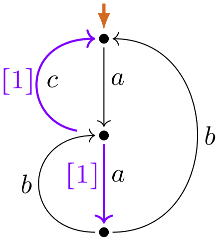
Expression extraction using LEE (G/Fokkink 2020, G 2021/22)

$\widehat{G_4}$



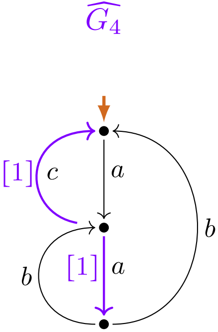
Expression extraction using LEE (G/Fokkink 2020, G 2021/22)

$\widehat{G_4}$



$$\Big(\qquad\qquad\qquad\qquad \Big)^{*} \cdot 0$$

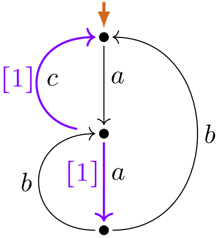
Expression extraction using LEE (G/Fokkink 2020, G 2021/22)



$$\Big(\quad \overset{\vdots\, a}{\downarrow} \quad \Big)^* \cdot 0$$

Expression extraction using LEE (G/Fokkink 2020, G 2021/22)

$\widehat{G_4}$



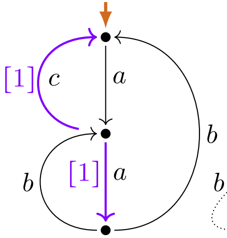
$(a \cdot 1) \cdot ((\qquad)^* \cdot 0)$

$\downarrow a$

$(\qquad)^* \cdot 0$

Expression extraction using LEE (G/Fokkink 2020, G 2021/22)

$\widehat{G_4}$



$(a \cdot 1) \cdot ((\qquad)^{*} \cdot 0)$

a

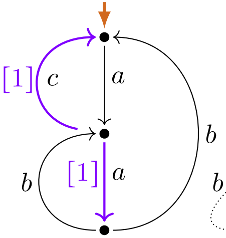
$(\qquad)^{*} \cdot 0$

b

b

Expression extraction using LEE (G/Fokkink 2020, G 2021/22)

$\widehat{G_4}$



$(a \cdot 1) \cdot ((\qquad)^* \cdot 0)$

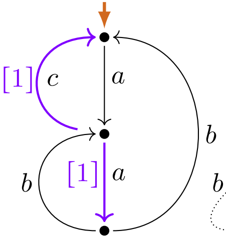
$\downarrow a$

$(\qquad)^* \cdot 0$

$(b \cdot 1 + b \cdot (a \cdot 1)) \cdot ((\qquad)^*) \cdot 0)$

Expression extraction using LEE (G/Fokkink 2020, G 2021/22)

$\widehat{G_4}$



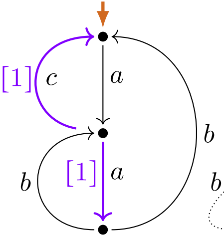
$(a \cdot 1) \cdot ((\qquad)^* \cdot 0)$

$(\qquad)^* \cdot 0$

$(b \cdot 1 + b \cdot (a \cdot 1)) \cdot ((\qquad)^*) \cdot 0)$

Expression extraction using LEE (G/Fokkink 2020, G 2021/22)

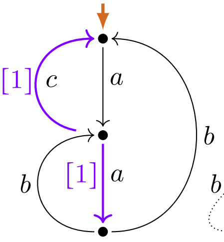
$\widehat{G_4}$



$$
\begin{tikzcd}
(a\cdot 1)\cdot\big(\big(\qquad \big)^*\cdot 0\big) \arrow[d, dotted, "a"] \\
\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^*\cdot 0 \arrow[u, dotted, bend left, "c"] \arrow[d, dotted, "a"] \\
(b\cdot 1+b\cdot(a\cdot 1))\cdot\big(\big(\qquad \big)^*\big)\cdot 0\big) \arrow[u, dotted, bend left, "b"] \arrow[uu, dotted, bend right, "b"']
\end{tikzcd}
$$

Expression extraction using LEE (G/Fokkink 2020, G 2021/22)

$\widehat{G_4}$



$$
\begin{array}{c}
(a \cdot 1) \cdot \Big(\big(c \cdot (a \cdot 1) + a \cdot (b \cdot 1 + b \cdot (a \cdot 1)) \big)^{*} \cdot 0 \Big) \\
\Big\downarrow a \\
\big(c \cdot (a \cdot 1) + a \cdot (b \cdot 1 + b \cdot (a \cdot 1)) \big)^{*} \cdot 0 \\
\Big\downarrow a \\
(b \cdot 1 + b \cdot (a \cdot 1)) \cdot \Big(\big(c \cdot (a \cdot 1) + a \cdot (b \cdot 1 + b \cdot (a \cdot 1)) \big)^{*}) \cdot 0 \Big)
\end{array}
$$

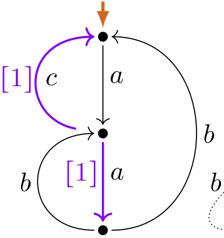
c

b

b

Expression extraction using LEE (G/Fokkink 2020, G 2021/22)

$\widehat{G_4}$



$$\overbrace{(a\cdot 1)\cdot\Big(\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\cdot 0\Big)}^{e}$$

$$\downarrow a$$

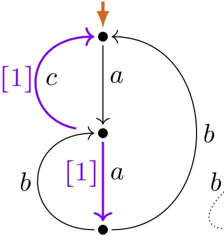
$$\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\cdot 0$$

$$\downarrow a$$

$$(b\cdot 1+b\cdot(a\cdot 1))\cdot\Big(\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\Big)\cdot 0\Big)$$

Expression extraction using LEE (G/Fokkink 2020, G 2021/22)

$\widehat{G_4}$



$$P(e) \rightarrow G_4$$

$$\overbrace{(a \cdot 1) \cdot \big(\big(c \cdot (a \cdot 1) + a \cdot (b \cdot 1 + b \cdot (a \cdot 1)) \big)^{*} \cdot 0 \big)}^{e}$$

$$\downarrow a$$

$$\big(c \cdot (a \cdot 1) + a \cdot (b \cdot 1 + b \cdot (a \cdot 1)) \big)^{*} \cdot 0$$

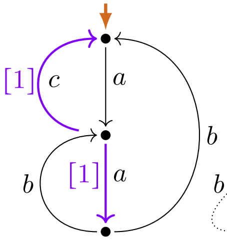
$$\downarrow a$$

$$(b \cdot 1 + b \cdot (a \cdot 1)) \cdot \big(\big(c \cdot (a \cdot 1) + a \cdot (b \cdot 1 + b \cdot (a \cdot 1)) \big)^{*}) \cdot 0 \big)$$

c b b

Expression extraction using LEE (G/Fokkink 2020, G 2021/22)

$\widehat{G_4}$



$P(e) \underline{\to} G_4$

$$\overbrace{(a\cdot 1)\cdot\Big(\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\cdot 0\Big)}^{e}$$

$$\cancel{\downarrow}\; a$$

$$\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\cdot 0$$

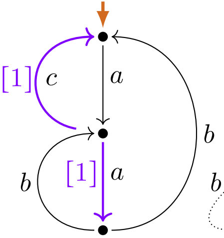
$$\downarrow a$$

$$(b\cdot 1+b\cdot(a\cdot 1))\cdot\Big(\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*})\cdot 0\Big)$$

$c \quad b \quad b$

Expression extraction using LEE (G/Fokkink 2020, G 2021/22)

$\widehat{G_4}$



$P(e) \;\underline{\to}\; G_4$

$$\overbrace{(a\cdot 1)\cdot\Big(\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\cdot 0\Big)}^{e}$$

$$\downarrow a$$

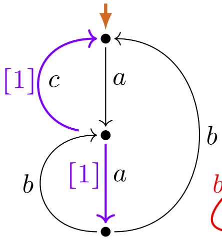
$$\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\cdot 0$$

$$\downarrow a$$

$$(b\cdot 1+b\cdot(a\cdot 1))\cdot\Big(\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\big)\cdot 0\Big)$$

Expression extraction using LEE (G/Fokkink 2020, G 2021/22)

$\widehat{G_4}$



$\mathsf{P}(e) \mathrel{\underline{\rightarrow}} G_4$

$$\overbrace{(a \cdot 1) \cdot \Big(\big(c \cdot (a \cdot 1) + a \cdot (b \cdot 1 + b \cdot (a \cdot 1)) \big)^{*} \cdot 0 \Big)}^{e}$$

$$\downarrow a$$

$$\big(c \cdot (a \cdot 1) + a \cdot (b \cdot 1 + b \cdot (a \cdot 1)) \big)^{*} \cdot 0$$

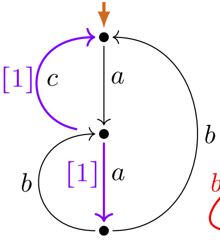
$$\downarrow a$$

$$(b \cdot 1 + b \cdot (a \cdot 1)) \cdot \Big(\big(c \cdot (a \cdot 1) + a \cdot (b \cdot 1 + b \cdot (a \cdot 1)) \big)^{*} \Big) \cdot 0 \Big)$$

c b b

Expression extraction using LEE (G/Fokkink 2020, G 2021/22)

$\widehat{G_4}$



$P(e) \;\underline{\rightarrow}\; G_4 \not\simeq P(e)$

$$\overbrace{(a\cdot 1)\cdot\big(\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\cdot 0\big)}^{e}$$

$$\Big\downarrow a$$

$$\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\cdot 0$$

$$\Big\downarrow a$$

$$(b\cdot 1+b\cdot(a\cdot 1))\cdot\big(\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\big)\cdot 0\big)$$

Interpretation of extracted expression

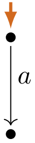
G_5

$P(e) = G_5$

$$\overbrace{(a\cdot 1)\cdot\Big(\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\cdot 0\Big)}^{e}$$

Interpretation of extracted expression

G_5



$$P(e) = G_5$$

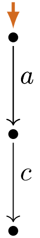
$$\overbrace{(a \cdot 1) \cdot \Big(\big(c \cdot (a \cdot 1) + a \cdot (b \cdot 1 + b \cdot (a \cdot 1))\big)^{*} \cdot 0\Big)}^{e}$$

$$\Big\downarrow a$$

$$(1 \cdot 1) \cdot \Big(\big(c \cdot (a \cdot 1) + a \cdot (b \cdot 1 + b \cdot (a \cdot 1))\big)^{*} \cdot 0\Big)$$

Interpretation of extracted expression

G_5



$P(e) = G_5$

$$\overbrace{(a\cdot 1)\cdot\Big(\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\cdot 0\Big)}^{e}$$

$$\downarrow a$$

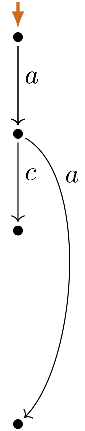
$$(1\cdot 1)\cdot\Big(\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\cdot 0\Big)$$

$$\downarrow c$$

$$\Big((1\cdot(a\cdot 1))\cdot\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\Big)\cdot 0$$

Interpretation of extracted expression

G_5



$P(e) = G_5$

$$\overbrace{(a \cdot 1) \cdot \big(\big(c \cdot (a \cdot 1) + a \cdot (b \cdot 1 + b \cdot (a \cdot 1)) \big)^{*} \cdot 0 \big)}^{e}$$

$$\downarrow a$$

$$(1 \cdot 1) \cdot \big(\big(c \cdot (a \cdot 1) + a \cdot (b \cdot 1 + b \cdot (a \cdot 1)) \big)^{*} \cdot 0 \big)$$

$$\downarrow c$$

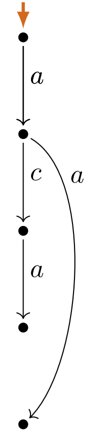
$$\big((1 \cdot (a \cdot 1)) \cdot \big(c \cdot (a \cdot 1) + a \cdot (b \cdot 1 + b \cdot (a \cdot 1)) \big)^{*} \big) \cdot 0$$

$$\xrightarrow{a}$$

$$\big((1 \cdot (b \cdot 1 + b \cdot (a \cdot 1))) \cdot \big(c \cdot (a \cdot 1) + a \cdot (b \cdot 1 + b \cdot (a \cdot 1)) \big)^{*} \big) \cdot 0$$

Interpretation of extracted expression

G_5



$P(e) = G_5$

$$\overbrace{(a\cdot 1)\cdot\big(\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\cdot 0\big)}^{e}$$

$$\downarrow a$$

$$(1\cdot 1)\cdot\big(\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\cdot 0\big)$$

$$\downarrow c$$

$$\big((1\cdot(a\cdot 1))\cdot\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\big)\cdot 0$$

$$\downarrow a$$

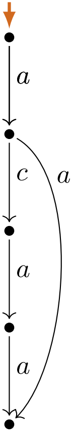
$$\big((1\cdot 1)\cdot\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\big)\cdot 0\big)$$

$$a$$

$$\big((1\cdot(b\cdot 1+b\cdot(a\cdot 1)))\cdot\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\big)\cdot 0$$

Interpretation of extracted expression

G_5



$P(e) = G_5$

$$\overbrace{(a\cdot 1)\cdot\big((c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1)))^{*}\cdot 0\big)}^{e}$$

$$\downarrow a$$

$$(1\cdot 1)\cdot\big((c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1)))^{*}\cdot 0\big)$$

$$\downarrow c$$

$$\big((1\cdot(a\cdot 1))\cdot\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\big)\cdot 0$$

$$\downarrow a$$

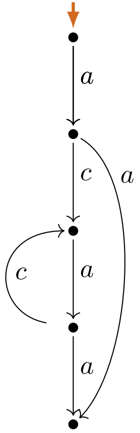
$$\big((1\cdot 1)\cdot\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\big)\cdot 0\big)$$

$$\downarrow a$$

$$\big((1\cdot(b\cdot 1+b\cdot(a\cdot 1)))\cdot\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\big)\cdot 0$$

Interpretation of extracted expression

G_5



$P(e) = G_5$

$$\overbrace{(a\cdot 1)\cdot\big((c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1)))^{*}\cdot 0\big)}^{e}$$

$$\downarrow a$$

$$(1\cdot 1)\cdot\big((c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1)))^{*}\cdot 0\big)$$

$$\downarrow c$$

$$\big((1\cdot(a\cdot 1))\cdot(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1)))^{*}\big)\cdot 0$$

$$\downarrow a$$

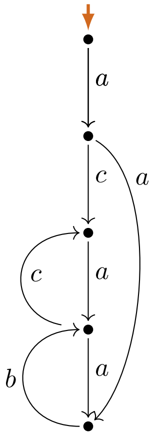
$$\big((1\cdot 1)\cdot(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1)))^{*}\big)\cdot 0\big)$$

$$\downarrow a$$

$$\big((1\cdot(b\cdot 1+b\cdot(a\cdot 1)))\cdot(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1)))^{*}\big)\cdot 0$$

Interpretation of extracted expression

G_5



$P(e) = G_5$

$$\overbrace{(a\cdot 1)\cdot\Big(\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\cdot 0\Big)}^{e}$$

$$\Big\downarrow a$$

$$(1\cdot 1)\cdot\Big(\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\cdot 0\Big)$$

$$\Big\downarrow c$$

$$\Big((1\cdot(a\cdot 1))\cdot\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\Big)\cdot 0$$

$$\Big\downarrow a$$

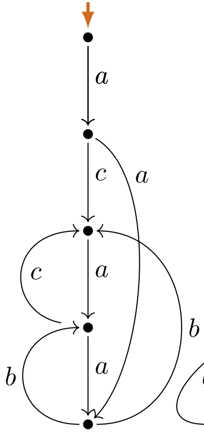
$$\Big((1\cdot 1)\cdot\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\Big)\cdot 0\Big)$$

$$\Big\downarrow a$$

$$\Big((1\cdot(b\cdot 1+b\cdot(a\cdot 1)))\cdot\big(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1))\big)^{*}\Big)\cdot 0$$

Interpretation of extracted expression

G_5



$P(e) = G_5$

$$\overbrace{(a \cdot 1) \cdot \big(\big(c \cdot (a \cdot 1) + a \cdot (b \cdot 1 + b \cdot (a \cdot 1)) \big)^{*} \cdot 0 \big)}^{e}$$

$$\downarrow a$$

$$(1 \cdot 1) \cdot \big(\big(c \cdot (a \cdot 1) + a \cdot (b \cdot 1 + b \cdot (a \cdot 1)) \big)^{*} \cdot 0 \big)$$

$$\downarrow c$$

$$\big((1 \cdot (a \cdot 1)) \cdot \big(c \cdot (a \cdot 1) + a \cdot (b \cdot 1 + b \cdot (a \cdot 1)) \big)^{*} \big) \cdot 0$$

$$\downarrow a$$

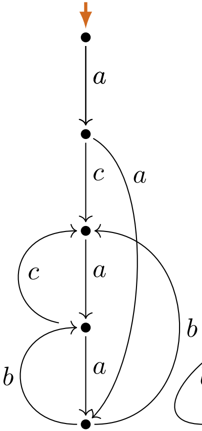
$$\big((1 \cdot 1) \cdot \big(c \cdot (a \cdot 1) + a \cdot (b \cdot 1 + b \cdot (a \cdot 1)) \big)^{*} \big) \cdot 0 \big)$$

$$\downarrow a$$

$$\big((1 \cdot (b \cdot 1 + b \cdot (a \cdot 1))) \cdot \big(c \cdot (a \cdot 1) + a \cdot (b \cdot 1 + b \cdot (a \cdot 1)) \big)^{*} \big) \cdot 0$$

Interpretation of extracted expression

G_5



$$P(e) = G_5 \rightarrow G_4$$

$$\overbrace{(a \cdot 1) \cdot \Big(\big(c \cdot (a \cdot 1) + a \cdot (b \cdot 1 + b \cdot (a \cdot 1)) \big)^* \cdot 0 \Big)}^{e}$$

$$\downarrow a$$

$$(1 \cdot 1) \cdot \Big(\big(c \cdot (a \cdot 1) + a \cdot (b \cdot 1 + b \cdot (a \cdot 1)) \big)^* \cdot 0 \Big)$$

$$\downarrow c$$

$$\Big((1 \cdot (a \cdot 1)) \cdot \big(c \cdot (a \cdot 1) + a \cdot (b \cdot 1 + b \cdot (a \cdot 1)) \big)^* \Big) \cdot 0$$

$$\downarrow a$$

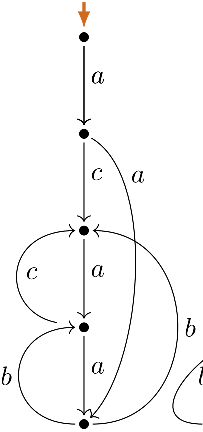
$$\Big((1 \cdot 1) \cdot \big(c \cdot (a \cdot 1) + a \cdot (b \cdot 1 + b \cdot (a \cdot 1)) \big)^* \Big) \cdot 0$$

$$\downarrow a$$

$$\Big((1 \cdot (b \cdot 1 + b \cdot (a \cdot 1))) \cdot \big(c \cdot (a \cdot 1) + a \cdot (b \cdot 1 + b \cdot (a \cdot 1)) \big)^* \Big) \cdot 0$$

Interpretation of extracted expression

G_5



$$P(e) = G_5 \Rightarrow G_4 \not\underline{\leftrightarrow} G_5$$

$$\overbrace{(a\cdot 1)\cdot\big((c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1)))^{*}\cdot 0\big)}^{e}$$

$$\downarrow a$$

$$(1\cdot 1)\cdot\big((c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1)))^{*}\cdot 0\big)$$

$$\downarrow c$$

$$\big((1\cdot(a\cdot 1))\cdot(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1)))^{*}\big)\cdot 0$$

$$\downarrow a$$

$$\big((1\cdot 1)\cdot(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1)))^{*}\big)\cdot 0\big)$$

$$\downarrow a$$

$$\big((1\cdot(b\cdot 1+b\cdot(a\cdot 1)))\cdot(c\cdot(a\cdot 1)+a\cdot(b\cdot 1+b\cdot(a\cdot 1)))^{*}\big)\cdot 0$$

Overview

- 1-free regular expressions (with unary/binary star)
- process interpretation/semantics of regular expressions
  - expressible/not expressible process graphs
- loop existence and elimination property (LEE) (G/Fokkink, 2020)
  - interpretation/extraction correspondences with 1-free reg. expr's
  - LEE is preserved under bisimulation collapse
- Q: Image of process interpretation of 1-free regular expressions closed under bisimulation collapse?
  - A1: No. — But ...
- compact process interpretation
- refined expression extraction
  - A2: compact process interpretation is image-closed under collapse
- outlook: consequences

Image of *P* is **not** closed under bisimulation collapse

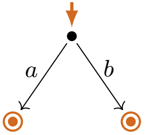
$P(uf)$

$P(uf)$

$$uf := a \cdot \overbrace{(a \cdot (a + a \cdot 0))^*}^{uf_a} + b \cdot \overbrace{(b \cdot (b + b \cdot 0))^*}^{uf_b}$$

Image of P is **not** closed under bisimulation collapse

$P(uf)$



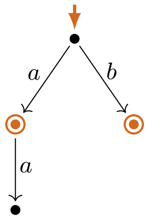
$P(uf)$

$$uf := a \cdot \overbrace{(a \cdot (a + a \cdot 0))^*}^{uf_a} + b \cdot \overbrace{(b \cdot (b + b \cdot 0))^*}^{uf_b}$$

$$1 \cdot uf_a \Downarrow \xleftarrow{a} \quad \xrightarrow{b} \Downarrow 1 \cdot uf_b$$

Image of P is **not** closed under bisimulation collapse

$P(uf)$



$P(uf)$

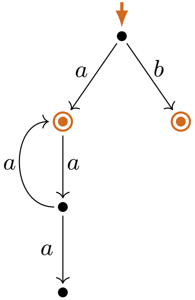
$$uf := a \cdot \overbrace{(a \cdot (a + a \cdot 0))^*}^{uf_a} + b \cdot \overbrace{(b \cdot (b + b \cdot 0))^*}^{uf_b}$$

$$uf \xrightarrow{a} 1 \cdot uf_a \Downarrow \qquad uf \xrightarrow{b} \Downarrow 1 \cdot uf_b$$

$$1 \cdot uf_a \xrightarrow{a} (1 \cdot (a + a \cdot 0)) \cdot uf_a$$

Image of P is **not** closed under bisimulation collapse

$P(uf)$



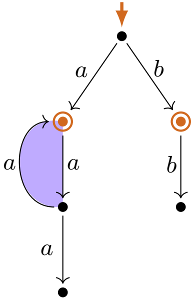
$P(uf)$

$$uf := a \cdot \overbrace{(a \cdot (a + a \cdot 0))^*}^{uf_a} + b \cdot \overbrace{(b \cdot (b + b \cdot 0))^*}^{uf_b}$$

$$\begin{tikzcd}
 & uf \arrow[dl, "a"'] \arrow[dr, "b"] & \\
1 \cdot uf_a \Downarrow \arrow[d, "a"] & & \Downarrow 1 \cdot uf_b \\
(1 \cdot (a + a \cdot 0)) \cdot uf_a \arrow[u, bend left, "a"] \arrow[d, "a"] & & \\
(1 \cdot 0) \cdot uf_a & &
\end{tikzcd}$$

Image of P is **not** closed under bisimulation collapse

$P(uf)$



$P(uf)$

$$uf := a \cdot \overbrace{(a \cdot (a + a \cdot 0))^*}^{uf_a} + b \cdot \overbrace{(b \cdot (b + b \cdot 0))^*}^{uf_b}$$

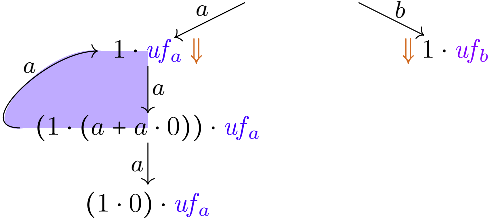
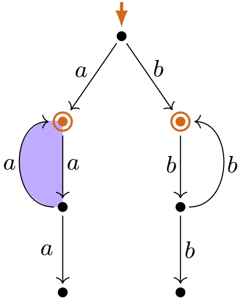


Image of *P* is **not** closed under bisimulation collapse

$P(uf)$



$P(uf)$

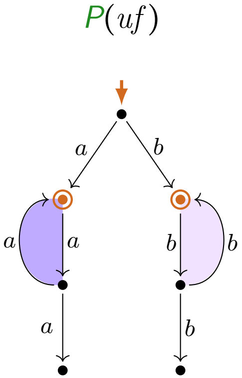
$$uf := a \cdot \overbrace{(a \cdot (a + a \cdot 0))^*}^{uf_a} + b \cdot \overbrace{(b \cdot (b + b \cdot 0))^*}^{uf_b}$$

$$uf \xrightarrow{a} 1 \cdot uf_a \Downarrow \qquad uf \xrightarrow{b} \Downarrow 1 \cdot uf_b$$

$$1 \cdot uf_a \xrightarrow{a} (1 \cdot (a + a \cdot 0)) \cdot uf_a \xrightarrow{a} (1 \cdot 0) \cdot uf_a \qquad (1 \cdot (a + a \cdot 0)) \cdot uf_a \xrightarrow{a} 1 \cdot uf_a$$

$$1 \cdot uf_b \xrightarrow{b} (1 \cdot (b + b \cdot 0)) \cdot uf_b$$

Image of P is **not** closed under bisimulation collapse



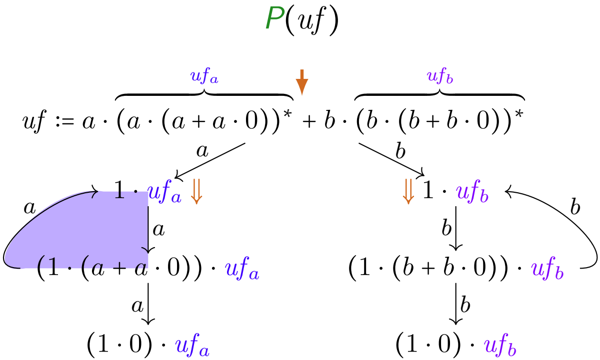
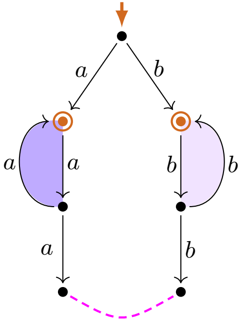


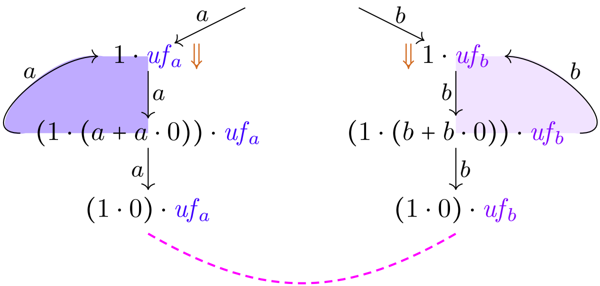
Image of P is **not** closed under bisimulation collapse

$P(uf)$



$P(uf)$

$$uf := a \cdot \overbrace{(a \cdot (a + a \cdot 0))^*}^{uf_a} + b \cdot \overbrace{(b \cdot (b + b \cdot 0))^*}^{uf_b}$$



Compact process interpretation $P^{\bullet}$

Definition (Transition system specification $\mathcal{T}$)

$$\frac{}{\mathbf{1}{\Downarrow}} \qquad \frac{e_i{\Downarrow}}{(e_1 + e_2){\Downarrow}}\;(i \in \{1,2\}) \qquad \frac{e_1{\Downarrow} \quad e_2{\Downarrow}}{(e_1 \cdot e_2){\Downarrow}} \qquad \frac{}{(e^{*}){\Downarrow}}$$

$$\frac{}{a \xrightarrow{a} \mathbf{1}} \qquad \frac{e_i \xrightarrow{a} e_i'}{e_1 + e_2 \xrightarrow{a} e_i'}\;(i \in \{1,2\})$$

$$\frac{e_1 \xrightarrow{a} e_1'}{e_1 \cdot e_2 \xrightarrow{a} e_1' \cdot e_2} \qquad \frac{e_1{\Downarrow} \quad e_2 \xrightarrow{a} e_2'}{e_1 \cdot e_2 \xrightarrow{a} e_2'} \qquad \frac{e \xrightarrow{a} e'}{e^{*} \xrightarrow{a} e' \cdot e^{*}}$$

Compact process interpretation $P^{\bullet}$

Definition (Transition system specification $\mathcal{T}$)

$$\frac{e_1 \xrightarrow{a} e_1'}{e_1 \cdot e_2 \xrightarrow{a} e_1' \cdot e_2}$$

$$\frac{e \xrightarrow{a} e'}{e^* \xrightarrow{a} e' \cdot e^*}$$

Compact process interpretation $P^{\bullet}$

Definition (Transition system specification $\mathcal{T}^{\bullet}$, changed rules w.r.t. $\mathcal{T}$)

$$\frac{e_1 \xrightarrow{a} e_1'}{e_1 \cdot e_2 \xrightarrow{a} e_1' \cdot e_2} \text{ (if } e_1' \text{ is normed)}$$

$$\frac{e \xrightarrow{a} e'}{e^* \xrightarrow{a} e' \cdot e^*} \text{ (if } e' \text{ is normed)}$$

Compact process interpretation $P^\bullet$

Definition (Transition system specification $\mathcal{T}^\bullet$, changed rules w.r.t. $\mathcal{T}$)

$$\frac{e_1 \xrightarrow{a} e_1'}{e_1 \cdot e_2 \xrightarrow{a} e_1' \cdot e_2} \text{ (if } e_1' \text{ is normed)} \qquad \frac{e_1 \xrightarrow{a} e_1'}{e_1 \cdot e_2 \xrightarrow{a} e_1'} \text{ (if } e_1' \text{ is not normed)}$$

$$\frac{e \xrightarrow{a} e'}{e^* \xrightarrow{a} e' \cdot e^*} \text{ (if } e' \text{ is normed)} \qquad \frac{e \xrightarrow{a} e'}{e^* \xrightarrow{a} e'} \text{ (if } e' \text{ is not normed)}$$

Compact process interpretation $P^{\bullet}$

Definition (Transition system specification $\mathcal{T}^{\bullet}$, changed rules w.r.t. $\mathcal{T}$)

$$\frac{e_1 \xrightarrow{a} e_1'}{e_1 \cdot e_2 \xrightarrow{a} e_1' \cdot e_2} \text{ (if } e_1' \text{ is normed)} \qquad \frac{e_1 \xrightarrow{a} e_1'}{e_1 \cdot e_2 \xrightarrow{a} e_1'} \text{ (if } e_1' \text{ is not normed)}$$

$$\frac{e \xrightarrow{a} e'}{e^{*} \xrightarrow{a} e' \cdot e^{*}} \text{ (if } e' \text{ is normed)} \qquad \frac{e \xrightarrow{a} e'}{e^{*} \xrightarrow{a} e'} \text{ (if } e' \text{ is not normed)}$$

Definition

The compact process (graph) interpretation $P^{\bullet}(e)$ of a reg. expr's e:

$P^{\bullet}(e) :=$ labeled transition graph generated by e by derivations in $\mathcal{T}^{\bullet}$.

Compact process interpretation $P^{\bullet}$

Definition (Transition system specification $\mathcal{T}^{\bullet}$, changed rules w.r.t. $\mathcal{T}$)

$$\frac{e_1 \xrightarrow{a} e_1'}{e_1 \cdot e_2 \xrightarrow{a} e_1' \cdot e_2} \text{ (if } e_1' \text{ is normed)} \qquad \frac{e_1 \xrightarrow{a} e_1'}{e_1 \cdot e_2 \xrightarrow{a} e_1'} \text{ (if } e_1' \text{ is not normed)}$$

$$\frac{e \xrightarrow{a} e'}{e^* \xrightarrow{a} e' \cdot e^*} \text{ (if } e' \text{ is normed)} \qquad \frac{e \xrightarrow{a} e'}{e^* \xrightarrow{a} e'} \text{ (if } e' \text{ is not normed)}$$

Definition

The compact process (graph) interpretation $P^{\bullet}(e)$ of a reg. expr's e:

$P^{\bullet}(e)$:= labeled transition graph generated by e by derivations in $\mathcal{T}^{\bullet}$.

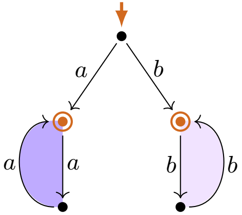
Lemma ($P^{\bullet}$ increases sharing; $P^{\bullet}$, P have same bisimulation semantics)

(i) $P(e) \mathrel{\underline{\to}} P^{\bullet}(e)$ *for all regular expressions* e.

(ii) (G *is* $[\![\cdot]\!]_{P^{\bullet}}$*-expressible* $\iff$ G *is* $[\![\cdot]\!]_{P}$*-expressible) for all graphs* G.

Image of $P^{\bullet}$ under bisimulation collapse ...

$P^{\bullet}(uf)$



$P^{\bullet}(uf)$

$$uf := a \cdot \overbrace{(a \cdot (a + a \cdot 0))^*}^{uf_a} + b \cdot \overbrace{(b \cdot (b + b \cdot 0))^*}^{uf_b}$$

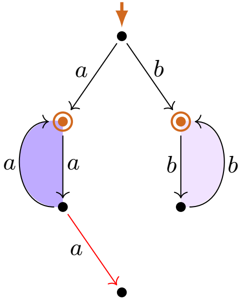
$$uf \xrightarrow{a} 1 \cdot uf_a \Downarrow \qquad uf \xrightarrow{b} \Downarrow 1 \cdot uf_b$$

$$1 \cdot uf_a \xrightarrow{a} (1 \cdot (a + a \cdot 0)) \cdot uf_a \xrightarrow{a} 1 \cdot uf_a$$

$$1 \cdot uf_b \xrightarrow{b} (1 \cdot (b + b \cdot 0)) \cdot uf_b \xrightarrow{b} 1 \cdot uf_b$$

Image of $P^{\bullet}$ under bisimulation collapse . . .

$P^{\bullet}(uf)$



$P^{\bullet}(uf)$

$$uf := a \cdot \overbrace{(a \cdot (a + a \cdot 0))^*}^{uf_a} + b \cdot \overbrace{(b \cdot (b + b \cdot 0))^*}^{uf_b}$$

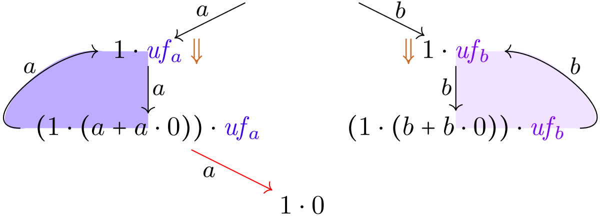
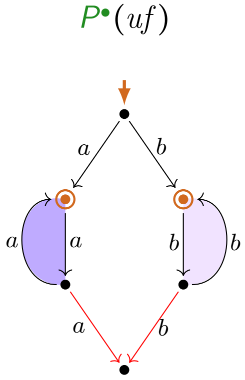
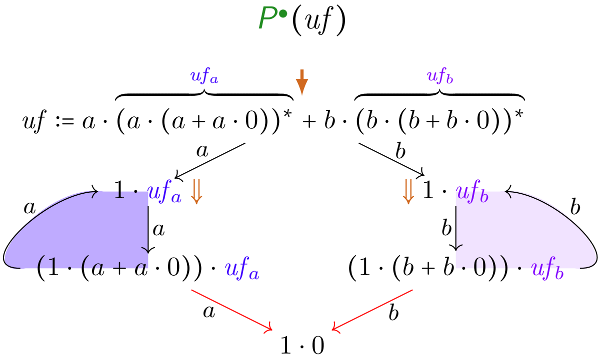


Image of $P^{\bullet}$ under bisimulation collapse . . .





Interpretation correspondence of $P^{\bullet}$ with LEE

(Int)$_{P^{\bullet}}^{(\cancel{1}\backslash *)}$: *By under-star-1-free expressions $P^{\bullet}$-expressible graphs satisfy* LEE:

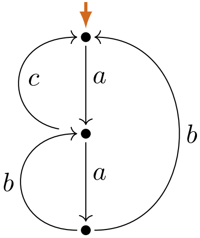
Compact process **i**nterpretations $P^{\bullet}(uf)$
of under-star-1-free regular expressions uf
are finite process graphs that satisfy LEE.

(Extr)$_{P^{\bullet}}^{(\cancel{1}\backslash *)}$: LEE *implies $[\![\cdot]\!]_P$-expressibility by under-star-1-free reg. expr's*:

From every finite process graph G with LEE
an under-star-1-free regular expression uf can be **e**xtracted
such that $G \mathrel{\underline{\rightarrow}} P(uf)$.

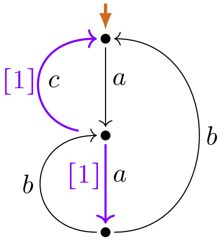
Refined extraction expression (example)

G_4

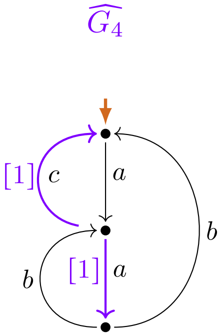


Refined extraction expression (example)

$\widehat{G_4}$



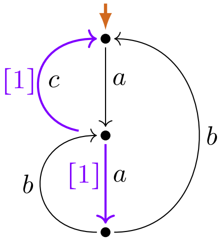
Refined extraction expression (example)



$$(1 \cdot (\qquad\qquad\qquad)^{*}) \cdot 0$$

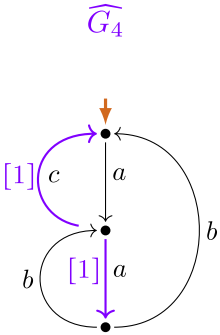
Refined extraction expression (example)

$\widehat{G_4}$



$$(1 \cdot (\qquad \overset{a}{\downarrow} \qquad)^*) \cdot 0$$

Refined extraction expression (example)



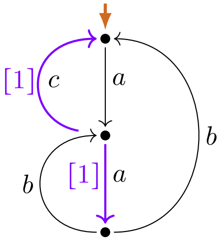
$$((1 \cdot a) \cdot (\qquad\qquad\qquad)^*) \cdot 0$$

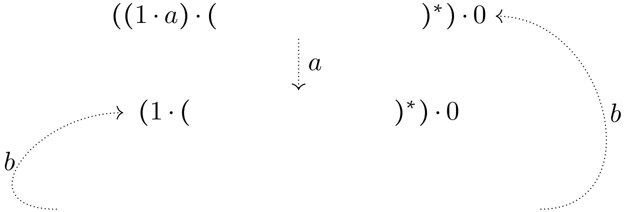
$$\Big\downarrow a$$

$$(1 \cdot (\qquad\qquad\qquad)^*) \cdot 0$$

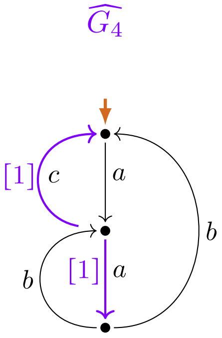
Refined extraction expression (example)

$\widehat{G_4}$





Refined extraction expression (example)



$((1 \cdot a) \cdot ($ $)^{*}) \cdot 0$

$\downarrow a$ $\quad b$

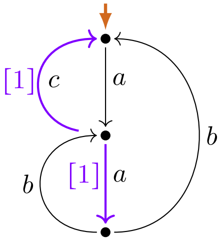
$(1 \cdot ($ $)^{*}) \cdot 0$

b

$((1 \cdot (b + b \cdot a)) \cdot ($ $)^{*}) \cdot 0$

Refined extraction expression (example)

$\widehat{G_4}$



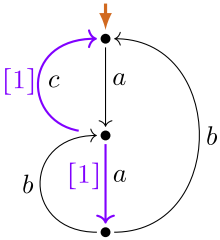
$$((1 \cdot a) \cdot (\qquad)^*) \cdot 0$$

$$(1 \cdot (\qquad)^*) \cdot 0$$

$$((1 \cdot (b + b \cdot a)) \cdot (\qquad)^*) \cdot 0$$

Refined extraction expression (example)

$\widehat{G_4}$



$$((1\cdot a)\cdot(\qquad\qquad\qquad\qquad)^{*})\cdot 0$$

$$\downarrow a$$

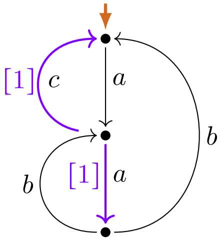
$$(1\cdot(c\cdot a + a\cdot(b + b\cdot a))^{*})\cdot 0$$

$$\downarrow a$$

$$((1\cdot(b + b\cdot a))\cdot(\qquad\qquad\qquad\qquad)^{*})\cdot 0$$

Refined extraction expression (example)

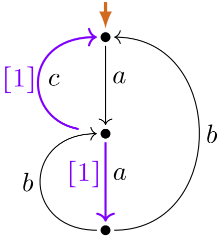
$\widehat{G_4}$



$$
\begin{tikzcd}
((1 \cdot a) \cdot (c \cdot a + a \cdot (b + b \cdot a))^{*}) \cdot 0 \arrow[d, dotted, "a"] \\
(1 \cdot (c \cdot a + a \cdot (b + b \cdot a))^{*}) \cdot 0 \arrow[d, dotted, "a"] \arrow[u, dotted, bend left, "c"] \\
((1 \cdot (b + b \cdot a)) \cdot (c \cdot a + a \cdot (b + b \cdot a))^{*}) \cdot 0 \arrow[u, dotted, bend left, "b"] \arrow[uu, dotted, bend right, "b"']
\end{tikzcd}
$$

Refined extraction expression (example)

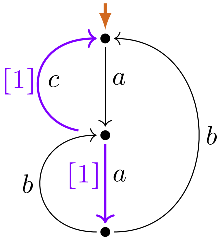
$\widehat{G_4}$



$$
\begin{tikzcd}
((1 \cdot a) \cdot (c \cdot a + a \cdot (b + b \cdot a))^{*}) \cdot 0 \arrow[d, "a"] \\
(1 \cdot (c \cdot a + a \cdot (b + b \cdot a))^{*}) \cdot 0 \arrow[d, "a"] \arrow[u, bend left, "c"] \\
((1 \cdot (b + b \cdot a)) \cdot (c \cdot a + a \cdot (b + b \cdot a))^{*}) \cdot 0 \arrow[u, bend left, "b"] \arrow[uu, bend right, "b"']
\end{tikzcd}
$$

Refined extraction expression (example)

$\widehat{G_4}$



$$P^{\bullet}(uf) = P(uf) \simeq G_4$$

$$\begin{tikzcd}
\overbrace{((1\cdot a)\cdot(c\cdot a + a\cdot(b + b\cdot a))^{*})\cdot 0}^{uf} \arrow[d, "a"] \\
(1\cdot(c\cdot a + a\cdot(b + b\cdot a))^{*})\cdot 0 \arrow[u, bend left, "c"] \arrow[d, "a"] \\
((1\cdot(b + b\cdot a))\cdot(c\cdot a + a\cdot(b + b\cdot a))^{*})\cdot 0 \arrow[u, bend left, "b"] \arrow[uu, bend right, "b"']
\end{tikzcd}$$

Interpretation/extraction correspondences of $P^{\bullet}$ with LEE

(Int)$_{P^{\bullet}}^{(\cancel{1}\backslash *)}$: *By under-star-1-free expressions $P^{\bullet}$-expressible graphs satisfy* LEE:

Compact process **i**nterpretations $P^{\bullet}(uf)$
of under-star-1-free regular expressions uf
are finite process graphs that satisfy LEE.

(Extr)$_{P^{\bullet}}^{(\cancel{1}\backslash *)}$: LEE *implies $[\![\cdot]\!]_P$-expressibility by under-star-1-free reg. expr's*:

From every finite process graph G with LEE
an under-star-1-free regular expression uf can be **e**xtracted
such that $G \rightrightarrows P(uf)$.

From every finite collapsed process graph G with LEE
an under-star-1-free regular expression uf can be **e**xtracted
such that $G \simeq P(uf)$.

Interpretation/extraction correspondences of $P^{\bullet}$ with LEE

(Int)$^{(\cancel{1}\backslash *)}_{P^{\bullet}}$: *By under-star-1-free expressions $P^{\bullet}$-expressible graphs satisfy LEE:*

Compact process **i**nterpretations $P^{\bullet}(uf)$
of under-star-1-free regular expressions uf
are finite process graphs that satisfy LEE.

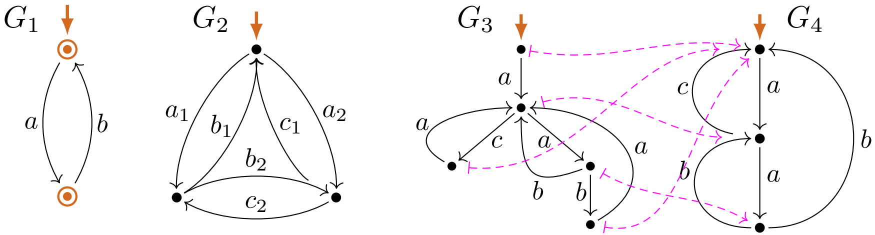
(Extr)$^{(\cancel{1}\backslash *)}_{P^{\bullet}}$: LEE *implies* $[\![\cdot]\!]_P$*-expressibility by under-star-1-free reg. expr's*:

From every finite process graph G with LEE
an under-star-1-free regular expression uf can be **e**xtracted
such that $G \mathrel{\underline{\to}} P(uf)$.

From every finite collapsed process graph G with LEE
an under-star-1-free regular expression uf can be **e**xtracted
such that $G \mathrel{\underline{\leftrightarrow}} P(uf)$.

(ImColl)$^{(\cancel{1}\backslash *)}_{P^{\bullet}}$: The image of $P^{\bullet}$,
restricted to under-star-1-free regular expressions,
is closed under bisimulation **c**ollapse.

P-/$P^{\bullet}$-expressibility and $[\![\cdot]\!]_P$-expressibility (examples)



not P-expressible

not $[\![\cdot]\!]_P$-expressible

P-/$P^{\bullet}$-expressible $\quad$ $P^{\bullet}$-expressible

$[\![\cdot]\!]_P$-expressible $\quad$ $[\![\cdot]\!]_P$-expressible

Summary and outlook

- 1-free/under-star-1-free $(\cancel{1}\backslash *)$ reg. expr's defined (also) with unary star
- image of $(\cancel{1}\backslash *)$ regular expressions under the process interpretation P
  is **not** closed under bisimulation collapse

Summary and outlook

- 1-free/under-star-1-free $(\not{1}\backslash *)$ reg. expr's defined (also) with unary star
- image of $(\not{1}\backslash *)$ regular expressions under the process interpretation P
  is **not** closed under bisimulation collapse
- compact process interpretation $P^{\bullet}$
- refined expression extraction from process graphs with LEE
- image of $(\not{1}\backslash *)$ reg. expr's under $P^{\bullet}$ **is closed** under collapse

Summary and outlook

- 1-free/under-star-1-free ($\underline{1}\backslash *$) reg. expr's defined (also) with unary star
- image of ($\underline{1}\backslash *$) regular expressions under the process interpretation P
  is **not** closed under bisimulation collapse
- compact process interpretation $P^{\bullet}$
- refined expression extraction from process graphs with LEE
- image of ($\underline{1}\backslash *$) reg. expr's under $P^{\bullet}$ **is closed** under collapse
- A finite process graph G is $[\![\cdot]\!]_P$-expressible by a ($\underline{1}\backslash *$) regular expression
  $\Longleftrightarrow$ the bisimulation collapse of G satisfies LEE (G/Fokkink 2020).

Summary and outlook

- 1-free/under-star-1-free $(\underline{1}\backslash *)$ reg. expr's defined (also) with unary star
- image of $(\underline{1}\backslash *)$ regular expressions under the process interpretation P
  is **not** closed under bisimulation collapse
- compact process interpretation $P^{\bullet}$
- refined expression extraction from process graphs with LEE
- image of $(\underline{1}\backslash *)$ reg. expr's under $P^{\bullet}$ **is closed** under collapse
- A finite process graph G is $[\![\cdot]\!]_P$-expressible by a $(\underline{1}\backslash *)$ regular expression
  $\Longleftrightarrow$ the bisim. collapse of G is $P^{\bullet}$-expressible by a $(\underline{1}\backslash *)$ reg. expr..

Summary and outlook

- 1-free/under-star-1-free $(\underline{1}\backslash *)$ reg. expr's defined (also) with unary star
- image of $(\underline{1}\backslash *)$ regular expressions under the process interpretation P
  is **not** closed under bisimulation collapse
- compact process interpretation $P^{\bullet}$
- refined expression extraction from process graphs with LEE
- image of $(\underline{1}\backslash *)$ reg. expr's under $P^{\bullet}$ **is closed** under collapse
- A finite process graph G is $[\![\cdot]\!]_P$-expressible by a $(\underline{1}\backslash *)$ regular expression
  $\Longleftrightarrow$ the bisim. collapse of G is $P^{\bullet}$-expressible by a $(\underline{1}\backslash *)$ reg. expr..

Outlook on an extension:

- image of $(\underline{1}\backslash *)$ reg. expr's under $P^{\bullet}$ $=$ finite process graphs with LEE.

Summary and outlook

- 1-free/under-star-1-free $(\mathit{1}\backslash *)$ reg. expr's defined (also) with unary star
- image of $(\mathit{1}\backslash *)$ regular expressions under the process interpretation P
  is **not** closed under bisimulation collapse
- compact process interpretation $P^{\bullet}$
- refined expression extraction from process graphs with LEE
- image of $(\mathit{1}\backslash *)$ reg. expr's under $P^{\bullet}$ **is closed** under collapse
- A finite process graph G is $[\![\cdot]\!]_P$-expressible by a $(\mathit{1}\backslash *)$ regular expression
  $\Longleftrightarrow$ the bisim. collapse of G is $P^{\bullet}$-expressible by a $(\mathit{1}\backslash *)$ reg. expr..

Outlook on an extension:

- image of $(\mathit{1}\backslash *)$ reg. expr's under $P^{\bullet}$ **=** finite process graphs with LEE.

  A finite process graph G is $P^{\bullet}$-expressible by a $(\mathit{1}\backslash *)$ regular expression
  $\Longleftrightarrow$ G satisfies LEE.

Summary and outlook

- 1-free/under-star-1-free ($\underline{1}\backslash *$) reg. expr's defined (also) with unary star
- image of ($\underline{1}\backslash *$) regular expressions under the process interpretation P
  is **not** closed under bisimulation collapse
- compact process interpretation $P^{\bullet}$
- refined expression extraction from process graphs with LEE
- image of ($\underline{1}\backslash *$) reg. expr's under $P^{\bullet}$ **is closed** under collapse
- A finite process graph G is $[\![\cdot]\!]_P$-expressible by a ($\underline{1}\backslash *$) regular expression
  $\iff$ the bisimulation collapse of G satisfies LEE (G/Fokkink 2020).

Outlook on an extension:

- image of ($\underline{1}\backslash *$) reg. expr's under $P^{\bullet}$ $=$ finite process graphs with LEE.

  A finite process graph G is $P^{\bullet}$-expressible by a ($\underline{1}\backslash *$) regular expression
  $\iff$ G satisfies LEE.

Resources

- Slides/extended abstract on `clegra.github.io`
  - slides: `.../lf/TG-2024.pdf`
  - extended abstract: `.../lf/closing-bs-i-pi-us1f.pdf`

- CG, Wan Fokkink: A Complete Proof System for 1-Free Regular Expressions Modulo Bisimilarity,
  - LICS 2020, arXiv:2004.12740, video on youtube.
- CG: Modeling Terms by Graphs with Structure Constraints,
  - TERMGRAPH 2018, EPTCS 288, arXiv:1902.02010.
- CG: The Image of the Process Interpretation of Regular Expressions is Not Closed under Bisimulation Collapse,
  - arXiv:2303.08553.
- CG: Milner's Proof System for Regular Expressions Modulo Bisimilarity is Complete,
  - LICS 2022, arXiv:2209.12188, poster.

Language semantics $[\![\cdot]\!]_L$ of reg. expr's *(Copi–Elgot–Wright, 1958)*

$$\mathbf{0} \xmapsto{L} \text{empty language } \varnothing$$

$$\mathbf{1} \xmapsto{L} \{\epsilon\} \qquad (\epsilon \text{ the empty word})$$

$$a \xmapsto{L} \{a\}$$

Language semantics $[\![\cdot]\!]_L$ of reg. expr's *(Copi–Elgot–Wright, 1958)*

$\mathbf{0} \stackrel{L}{\longmapsto}$ empty language $\varnothing$

$\mathbf{1} \stackrel{L}{\longmapsto} \{\epsilon\}$ (ϵ the empty word)

$a \stackrel{L}{\longmapsto} \{a\}$

$e_1 + e_2 \stackrel{L}{\longmapsto}$ union of $L(e_1)$ and $L(e_2)$

$e_1 \cdot e_2 \stackrel{L}{\longmapsto}$ element-wise concatenation of $L(e_1)$ and $L(e_2)$

$e^* \stackrel{L}{\longmapsto}$ set of words formed by concatenating words in $L(e)$, and adding the empty word ϵ

Language semantics $[\![\cdot]\!]_L$ of reg. expr's *(Copi–Elgot–Wright, 1958)*

$\mathbf{0} \xrightarrow{L}$ empty language $\varnothing$

$\mathbf{1} \xrightarrow{L} \{\epsilon\}$ (ϵ the empty word)

$a \xrightarrow{L} \{a\}$

$e_1 + e_2 \xrightarrow{L}$ union of $L(e_1)$ and $L(e_2)$

$e_1 \cdot e_2 \xrightarrow{L}$ element-wise concatenation of $L(e_1)$ and $L(e_2)$

$e^* \xrightarrow{L}$ set of words formed by concatenating words in $L(e)$,
and adding the empty word ϵ

$[\![e]\!]_L := L(e)$ (language defined by e)

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)

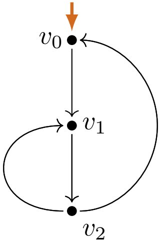
Definition

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)

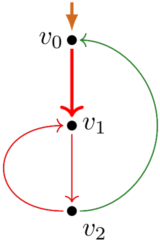
Definition

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)

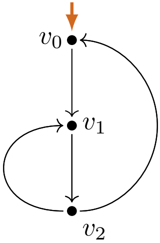
Definition

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)

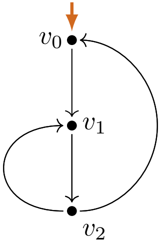
Definition

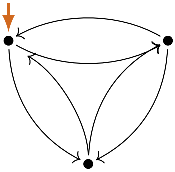
A chart is a loop chart if:

(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)

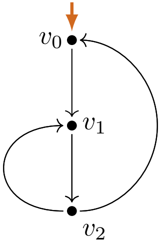
Definition

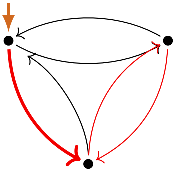
A chart is a loop chart if:

(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)

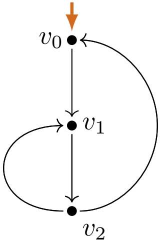
Definition

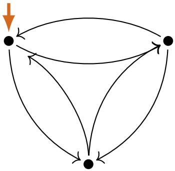
A chart is a loop chart if:

(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.





(L1), ~~(L2)~~

Loop charts (interpretations of innermost iterations)

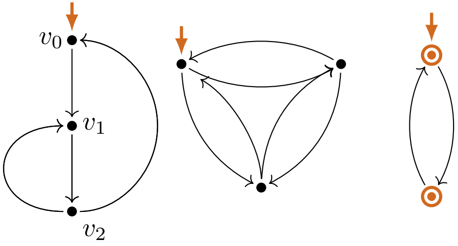
Definition

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)

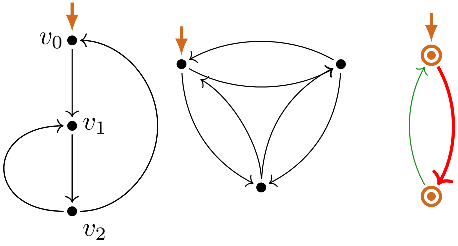
Definition

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)

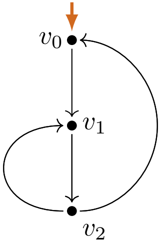
Definition

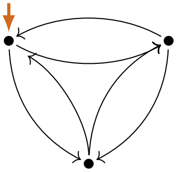
A chart is a loop chart if:

(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)~~

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

Loop charts (interpretations of innermost iterations)

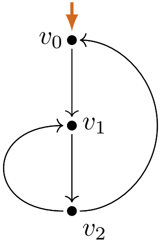
Definition

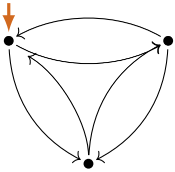
A chart is a loop chart if:

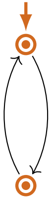
(L1) There is an infinite path from the start vertex.

(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)

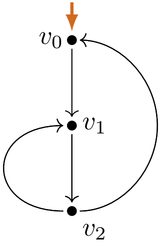
Definition

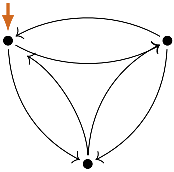
A chart is a loop chart if:

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

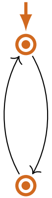
(L2) Every infinite path from the start vertex returns to it.

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

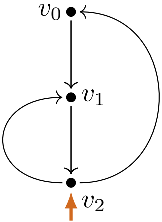




(L1),~~(L2)~~



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



Loop charts (interpretations of innermost iterations)

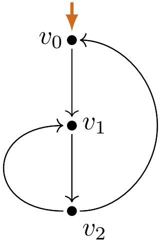
Definition

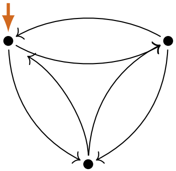
A chart is a loop chart if:

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

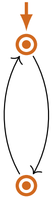
(L2) Every infinite path from the start vertex returns to it.

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

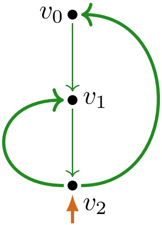




(L1),~~(L2)~~



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



Loop charts (interpretations of innermost iterations)

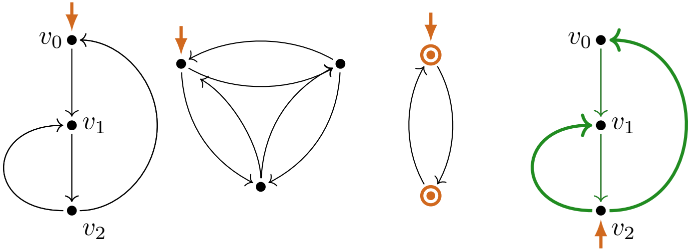
Definition

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)

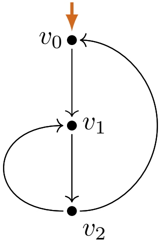
Definition

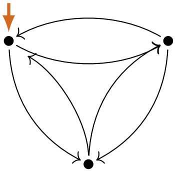
A chart is a loop chart if:

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

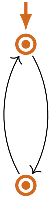
(L2) Every infinite path from the start vertex returns to it.

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

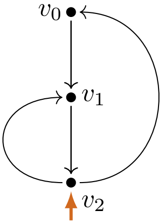




(L1),~~(L2)~~



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



loop chart

Loop charts (interpretations of innermost iterations)

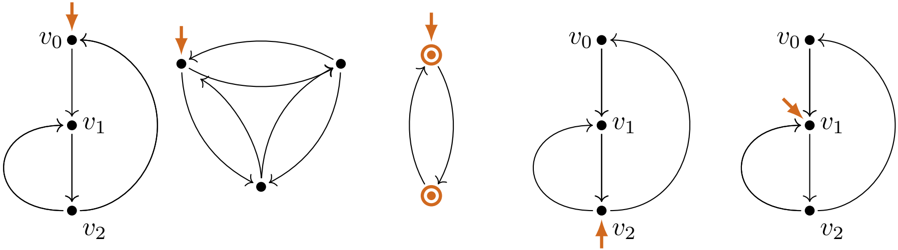
Definition

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)

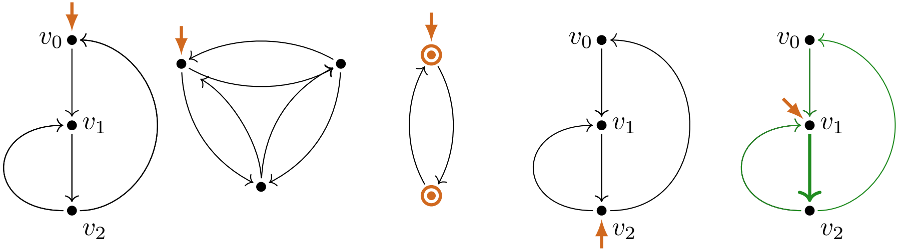
Definition

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.



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

Loop charts (interpretations of innermost iterations)

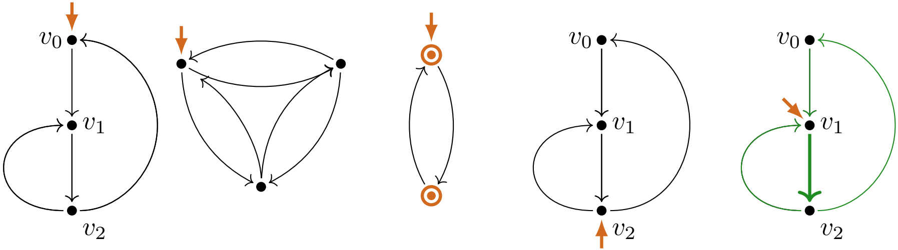
Definition

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 chart

Loop charts (interpretations of innermost iterations)

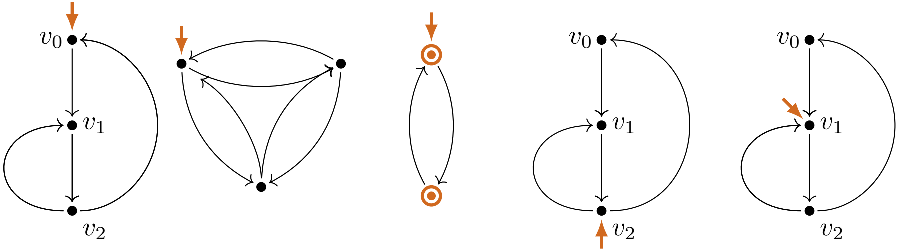
Definition

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 chart

Loop charts (interpretations of innermost iterations)

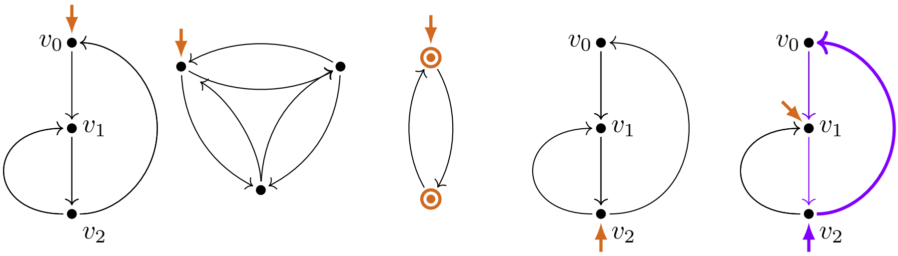
Definition

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 subchart