REDUCIBLE GRAPHS AND BISIMILARITY OF 1-FREE STAR EXPRESSIONS

Bas Laarakker & Tobias Kappé, Leiden University

TERMGRAPH

Lisbon, 19 July 2026

BISIMILARITY OF REGULAR EXPRESSIONS

BISIMILARITY OF REGULAR EXPRESSIONS

Milner (1984) studied regular expressions as denotations for processes.

BISIMILARITY OF REGULAR EXPRESSIONS

Milner (1984) studied regular expressions as denotations for processes.

Definition

The set of 1-free star expressions over a set of labels / actions A:

$$\mathrm{StExp}(A) \ni e, f ::= 0 \mid a \in A \mid e + f \mid e \cdot f \mid e^{\circledast} f$$

BISIMILARITY OF REGULAR EXPRESSIONS

Milner (1984) studied regular expressions as denotations for processes.

Definition

The set of 1-free star expressions over a set of labels / actions A:

$$\mathrm{StExp}(A) \ni e, f ::= 0 \mid a \in A \mid e + f \mid e \cdot f \mid e^{\circledast} f$$

- With binary Kleene star $e^{\circledast} f$, 1 is *not* definable.

BISIMILARITY OF REGULAR EXPRESSIONS

Milner (1984) studied regular expressions as denotations for processes.

Definition

The set of 1-free star expressions over a set of labels / actions A:

$$\mathrm{StExp}(A) \ni e, f ::= 0 \mid a \in A \mid e + f \mid e \cdot f \mid e^{\circledast} f$$

- With binary Kleene star $e^{\circledast} f$, 1 is *not* definable.
- Immediate termination is not possible in 1-free expressions.

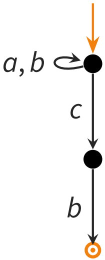
PROCESS SEMANTICS

A star expression e denotes a process, represented by a chart $C(e)$:

PROCESS SEMANTICS

A star expression e denotes a process, represented by a chart $C(e)$:

$C(e)$

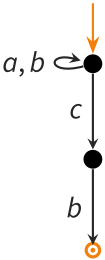


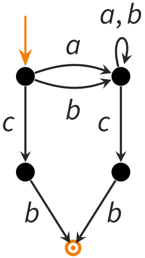
$$e = ((a + b)^{\circledast} c) \cdot b$$

PROCESS SEMANTICS

A star expression e denotes a process, represented by a chart $C(e)$:

$C(e)$





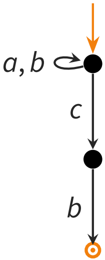
$C(f)$

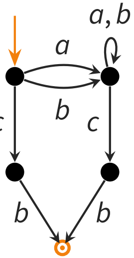
$e = ((a + b)^{\circledast} c) \cdot b$ $\qquad$ $f = (a+b)\cdot(a+b)^{\circledast}(c\cdot b)+(c\cdot b)$

PROCESS SEMANTICS

A star expression e denotes a process, represented by a chart $C(e)$:

$C(e)$





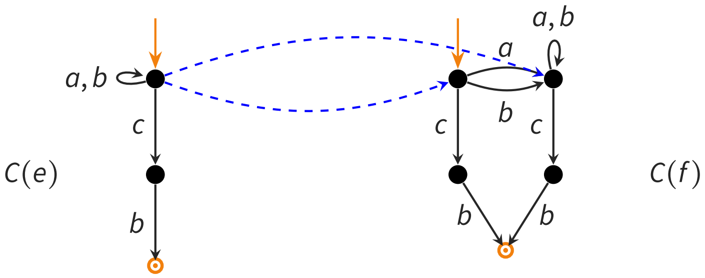
$C(f)$

$$e \;=\; ((a + b)^{\circledast} c) \cdot b \qquad f = (a+b)\cdot(a+b)^{\circledast}(c\cdot b)+(c\cdot b)$$

A bisimulation between two charts is a relation between their nodes that preserves transition structure.

PROCESS SEMANTICS

A star expression e denotes a process, represented by a chart $C(e)$:

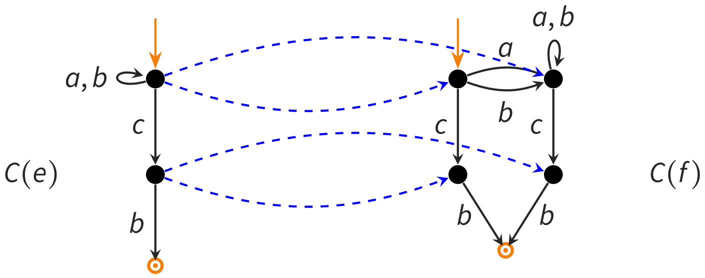


$$e \;=\; ((a + b)^{\circledast} c) \cdot b \qquad\qquad f = (a+b)\cdot(a+b)^{\circledast}(c\cdot b)+(c\cdot b)$$

A bisimulation between two charts is a relation between their nodes that preserves transition structure.

PROCESS SEMANTICS

A star expression e denotes a process, represented by a chart $C(e)$:

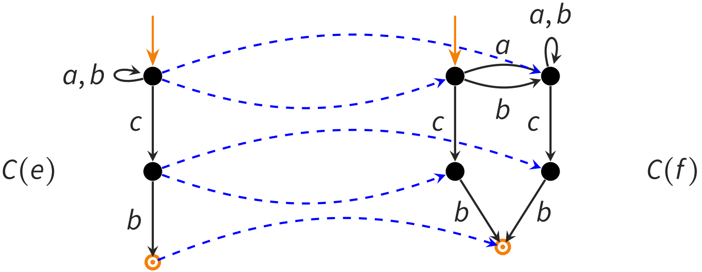


$$e = ((a+b)^{\circledast}c)\cdot b \qquad f = (a+b)\cdot(a+b)^{\circledast}(c\cdot b)+(c\cdot b)$$

A bisimulation between two charts is a relation between their nodes that preserves transition structure.

PROCESS SEMANTICS

A star expression e denotes a process, represented by a chart $C(e)$:

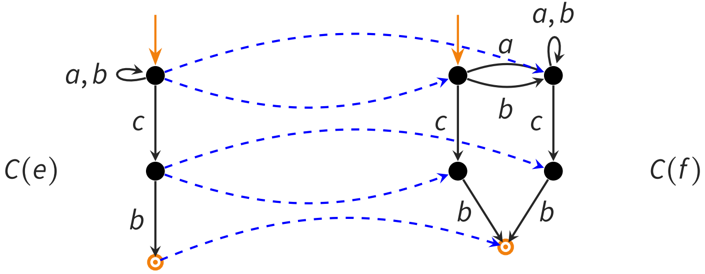


$$e \;=\; ((a + b)^{\circledast} c) \cdot b \qquad\qquad f = (a{+}b)\cdot(a{+}b)^{\circledast}(c\cdot b)+(c\cdot b)$$

A bisimulation between two charts is a relation between their nodes that preserves transition structure.

PROCESS SEMANTICS

A star expression e denotes a process, represented by a chart $C(e)$:



$$e = ((a+b)^{\circledast}c) \cdot b \qquad f = (a+b)\cdot(a+b)^{\circledast}(c\cdot b)+(c\cdot b)$$

The two charts are bisimilar: $C(e) \leftrightarroweq C(f)$.

We say that the expressions e and f are bisimilar when $C(e) \leftrightarroweq C(f)$.

BISIMILARITY VS. LANGUAGE EQUIVALENCE

BISIMILARITY VS. LANGUAGE EQUIVALENCE

Bisimilarity of expressions implies language equivalence:

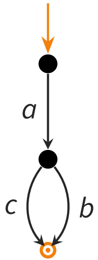
$$e \leftrightarroweq f \implies L(e) = L(f)$$

BISIMILARITY VS. LANGUAGE EQUIVALENCE

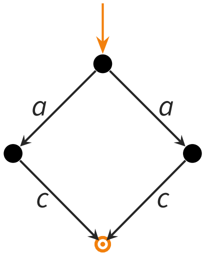
Bisimilarity of expressions implies language equivalence:

$$e \leftrightarrow f \implies L(e) = L(f)$$

The converse is not true:



$a \cdot (b + c)$



$a \cdot b + a \cdot c$

BISIMILARITY VS. LANGUAGE EQUIVALENCE

Recall that for language equivalence, the language of every automaton is expressible by a regular expression.

BISIMILARITY VS. LANGUAGE EQUIVALENCE

Recall that for language equivalence, the language of every automaton is expressible by a regular expression.

However, not every process chart is bisimilar to the interpretation of a star expression (Bosscher 1997, Milner 1984):

BISIMILARITY VS. LANGUAGE EQUIVALENCE

Recall that for language equivalence, the language of every automaton is expressible by a regular expression.

However, not every process chart is bisimilar to the interpretation of a star expression (Bosscher 1997, Milner 1984):

PROOF SYSTEM BBP FOR BISIMILARITY

Axioms for language equivalence of 1-free expressions:

(B1)	$x + y = y + x$	(B2)	$(x + y) + z = x + (y + z)$
(B3)	$x + x = x$	(B4)	$x + 0 = x$
(B5)	$(x + y) \cdot z = x \cdot z + y \cdot z$	(L1)	$x \cdot (y + z) = x \cdot y + x \cdot z$
(B6)	$(x \cdot y) \cdot z = x \cdot (y \cdot z)$	(B7)	$0 \cdot x = 0$
(L2)	$x \cdot 0 = 0$	(BKS1)	$x \cdot (x^{\circledast} y) + y = x^{\circledast} y$
(BKS2)	$(x^{\circledast} y) \cdot z = x^{\circledast}(y \cdot z)$	(RSP)	$\dfrac{x = (y \cdot x) + z}{x = y^{\circledast} z}$

PROOF SYSTEM BBP FOR BISIMILARITY

Axioms for ~~language equivalence~~ bisimilarity of 1-free expressions:

(B1)	$x + y = y + x$	(B2)	$(x + y) + z = x + (y + z)$
(B3)	$x + x = x$	(B4)	$x + 0 = x$
(B5)	$(x + y) \cdot z = x \cdot z + y \cdot z$	(L1)	$\cancel{x \cdot (y + z) = x \cdot y + x \cdot z}$
(B6)	$(x \cdot y) \cdot z = x \cdot (y \cdot z)$	(B7)	$0 \cdot x = 0$
(L2)	$\cancel{x \cdot 0 = 0}$	(BKS1)	$x \cdot (x^{\circledast} y) + y = x^{\circledast} y$
(BKS2)	$(x^{\circledast} y) \cdot z = x^{\circledast} (y \cdot z)$	(RSP)	$\dfrac{x = (y \cdot x) + z}{x = y^{\circledast} z}$

SOUNDNESS AND COMPLETENESS

SOUNDNESS AND COMPLETENESS

- (Milner 1984): The axioms are sound for bisimilarity

$$e =_{BBP} f \implies e \leftrightarrows f.$$

SOUNDNESS AND COMPLETENESS

- (Milner 1984): The axioms are sound for bisimilarity

$$e =_{BBP} f \implies e \leftrightarroweq f.$$

- (Grabmayer & Fokkink 2020): The axioms are complete for bisimilarity

$$e \leftrightarroweq f \implies e =_{BBP} f.$$

THE STRUCTURE OF COMPLETENESS PROOFS

THE STRUCTURE OF COMPLETENESS PROOFS

(Grabmayer & Fokkink 2020), (Schmid, Rot & Silva 2021)

Definition

A *(provable) solution* to a chart C, with set of nodes V, is a function $s : V \to \mathrm{StExp}(A)$ such that the following holds for all $v \in V$:

$$s(v) =_{\mathrm{BBP}} \sum_{v \xrightarrow{a} \surd} a + \sum_{v \xrightarrow{b} w} b \cdot s(w)$$

THE STRUCTURE OF COMPLETENESS PROOFS

(Grabmayer & Fokkink 2020), (Schmid, Rot & Silva 2021)

Definition

A *(provable) solution* to a chart C, with set of nodes V, is a function $s : V \to \mathrm{StExp}(A)$ such that the following holds for all $v \in V$:

$$s(v) =_{\mathrm{BBP}} \sum_{v \xrightarrow{a} \surd} a + \sum_{v \xrightarrow{b} w} b \cdot s(w)$$

To obtain a completeness proof for an axiomatisation of bisimilarity, it suffices to find a class of charts $\mathcal{C}$ such that:

THE STRUCTURE OF COMPLETENESS PROOFS

(Grabmayer & Fokkink 2020), (Schmid, Rot & Silva 2021)

Definition

A *(provable) solution* to a chart C, with set of nodes V, is a function $s : V \to \mathrm{StExp}(A)$ such that the following holds for all $v \in V$:

$$s(v) =_{\mathrm{BBP}} \sum_{v \xrightarrow{a} \surd} a + \sum_{v \xrightarrow{b} w} b \cdot s(w)$$

To obtain a completeness proof for an axiomatisation of bisimilarity, it suffices to find a class of charts $\mathcal{C}$ such that:

1. For every star expression e, the interpretation $C(e)$ belongs to $\mathcal{C}$;

THE STRUCTURE OF COMPLETENESS PROOFS

(Grabmayer & Fokkink 2020), (Schmid, Rot & Silva 2021)

Definition

A *(provable) solution* to a chart C, with set of nodes V, is a function $s : V \to \mathrm{StExp}(A)$ such that the following holds for all $v \in V$:

$$s(v) =_{\mathrm{BBP}} \sum_{v \xrightarrow{a} \surd} a + \sum_{v \xrightarrow{b} w} b \cdot s(w)$$

To obtain a completeness proof for an axiomatisation of bisimilarity, it suffices to find a class of charts $\mathcal{C}$ such that:

1. For every star expression e, the interpretation $C(e)$ belongs to $\mathcal{C}$;
2. Each chart in $\mathcal{C}$ admits a unique solution;

THE STRUCTURE OF COMPLETENESS PROOFS

(Grabmayer & Fokkink 2020), (Schmid, Rot & Silva 2021)

Definition

A *(provable) solution* to a chart C, with set of nodes V, is a function $s : V \to \mathrm{StExp}(A)$ such that the following holds for all $v \in V$:

$$s(v) =_{\mathrm{BBP}} \sum_{v \xrightarrow{a} \surd} a + \sum_{v \xrightarrow{b} w} b \cdot s(w)$$

To obtain a completeness proof for an axiomatisation of bisimilarity, it suffices to find a class of charts $\mathcal{C}$ such that:

1. For every star expression e, the interpretation $C(e)$ belongs to $\mathcal{C}$;
2. Each chart in $\mathcal{C}$ admits a unique solution;
3. The class $\mathcal{C}$ is closed under homomorphic images (quotients).

LLEE CHARTS

LLEE CHARTS

LLEE: Layered Loop Existence and Elimination (Grabmayer & Fokkink, 2020).

LLEE CHARTS

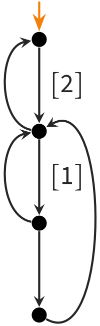
LLEE: Layered Loop Existence and Elimination (Grabmayer & Fokkink, 2020).

Idea: A chart is LLEE if the loops form a hierarchy: they are not mutually nested.

LLEE CHARTS

LLEE: Layered Loop Existence and Elimination (Grabmayer & Fokkink, 2020).

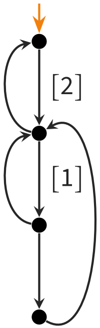
Idea: A chart is LLEE if the loops form a hierarchy: they are not mutually nested.



LLEE CHARTS

LLEE: Layered Loop Existence and Elimination (Grabmayer & Fokkink, 2020).

Idea: A chart is LLEE if the loops form a hierarchy: they are not mutually nested.



Challenge: The class of LLEE charts is closed under homomorphic images.

REDUCIBLE GRAPHS AS AN ALTERNATIVE CLASS

- We propose a new class of charts to simplify the completeness proof.

REDUCIBLE GRAPHS AS AN ALTERNATIVE CLASS

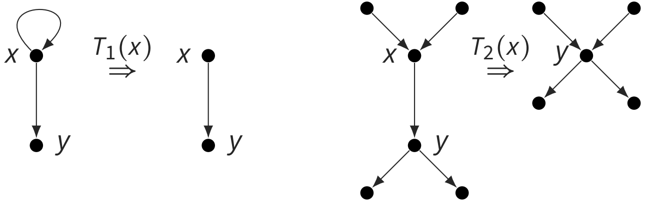
- We propose a new class of charts to simplify the completeness proof.
- Reducible graphs: an important class of flow graphs from compiler engineering (Allen, 1970).

REDUCIBLE GRAPHS AS AN ALTERNATIVE CLASS

- We propose a new class of charts to simplify the completeness proof.
- Reducible graphs: an important class of flow graphs from compiler engineering (Allen, 1970).
- We *dualise* their class of graphs:
  A graph with a terminal node is *reducible* if it reduces to a single point by the following rewrite rules:

REDUCIBLE GRAPHS AS AN ALTERNATIVE CLASS

- We propose a new class of charts to simplify the completeness proof.
- Reducible graphs: an important class of flow graphs from compiler engineering (Allen, 1970).
- We *dualise* their class of graphs:
  A graph with a terminal node is *reducible* if it reduces to a single point by the following rewrite rules:



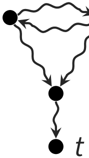
A STRUCTURAL CHARACTERISATION OF REDUCIBILITY

A STRUCTURAL CHARACTERISATION OF REDUCIBILITY

A graph G with terminal node t is not reducible iff it contains a subgraph of the following shape (Hecht & Ullman, 1972):

A STRUCTURAL CHARACTERISATION OF REDUCIBILITY

A graph G with terminal node t is not reducible iff it contains a subgraph of the following shape (Hecht & Ullman, 1972):



A STRUCTURAL CHARACTERISATION OF REDUCIBILITY

A graph G with terminal node t is not reducible iff it contains a subgraph of the following shape (Hecht & Ullman, 1972):

"Forbidden subgraph"

REDUCIBLE GRAPHS FOR STRONGLY NORMED CHARTS

- The definition of reducibility requires the graph to have a terminal node.

REDUCIBLE GRAPHS FOR STRONGLY NORMED CHARTS

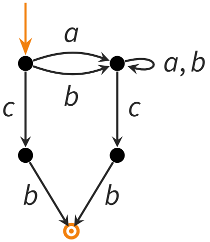
- The definition of reducibility requires the graph to have a terminal node.
- The subclass of strongly normed charts can be classified as reducible / not reducible.

REDUCIBLE GRAPHS FOR STRONGLY NORMED CHARTS

- The definition of reducibility requires the graph to have a terminal node.
- The subclass of strongly normed charts can be classified as reducible / not reducible.
  - Strongly normed: every node has a path to termination.

REDUCIBLE GRAPHS FOR STRONGLY NORMED CHARTS

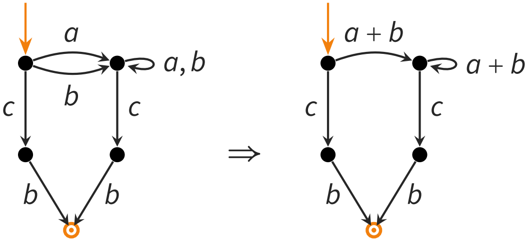
- The definition of reducibility requires the graph to have a terminal node.
- The subclass of strongly normed charts can be classified as reducible / not reducible.
  - Strongly normed: every node has a path to termination.



$\Rightarrow$

REDUCIBLE GRAPHS FOR STRONGLY NORMED CHARTS

- The definition of reducibility requires the graph to have a terminal node.
- The subclass of strongly normed charts can be classified as reducible / not reducible.
  - Strongly normed: every node has a path to termination.



SOLUTIONS TO REDUCIBLE CHARTS

2. Reducible graphs admit a unique solution

SOLUTIONS TO REDUCIBLE CHARTS

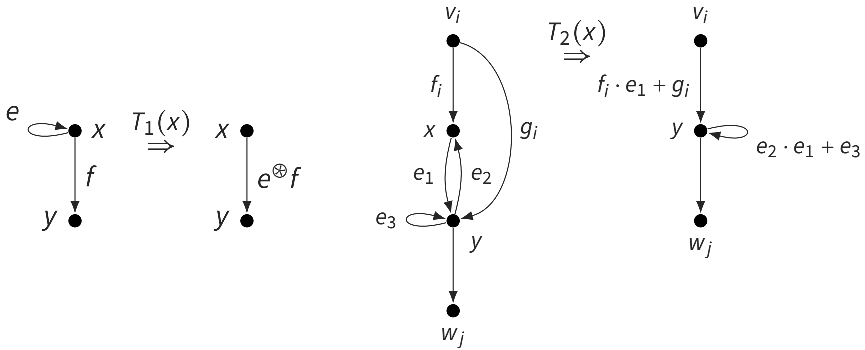
2️⃣ Reducible graphs admit a unique solution

We use the transformations T_1 and T_2 to construct solutions to reducible charts:

SOLUTIONS TO REDUCIBLE CHARTS

2 Reducible graphs admit a unique solution

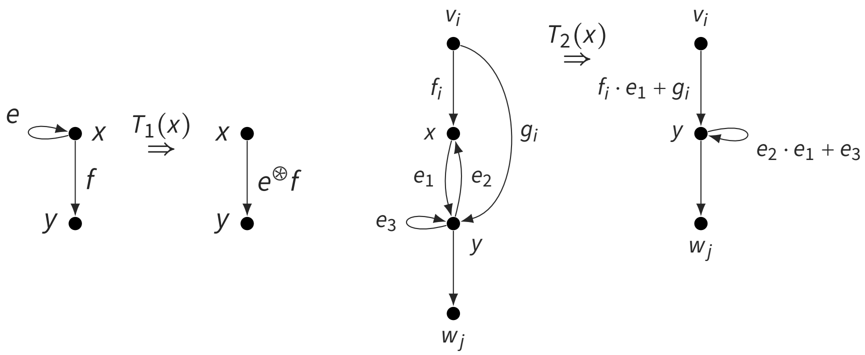
We use the transformations T_1 and T_2 to construct solutions to reducible charts:



SOLUTIONS TO REDUCIBLE CHARTS

2 Reducible graphs admit a unique solution

We use the transformations T_1 and T_2 to construct solutions to reducible charts:

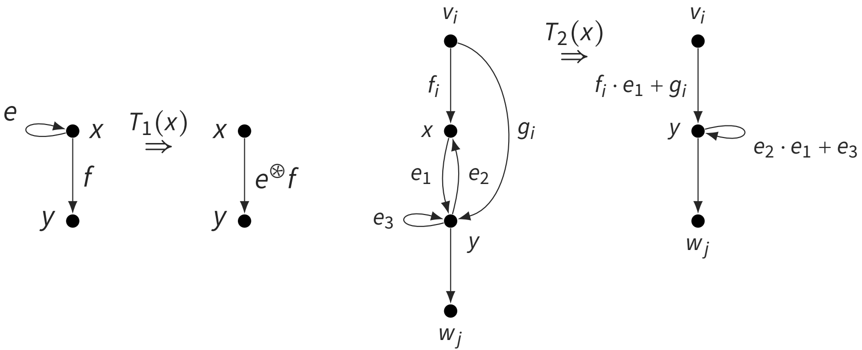


The transformations *uniquely* preserve and reflect solutions.

SOLUTIONS TO REDUCIBLE CHARTS

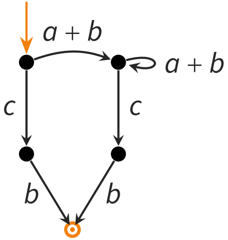
2️⃣ Reducible graphs admit a unique solution ✅

We use the transformations T_1 and T_2 to construct solutions to reducible charts:



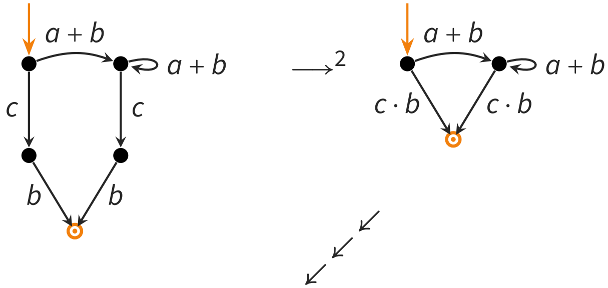
The transformations *uniquely* preserve and reflect solutions.

EXAMPLE

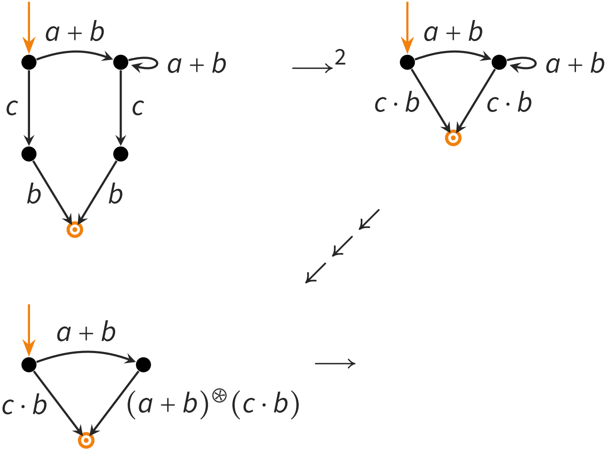


$\longrightarrow^2$

EXAMPLE

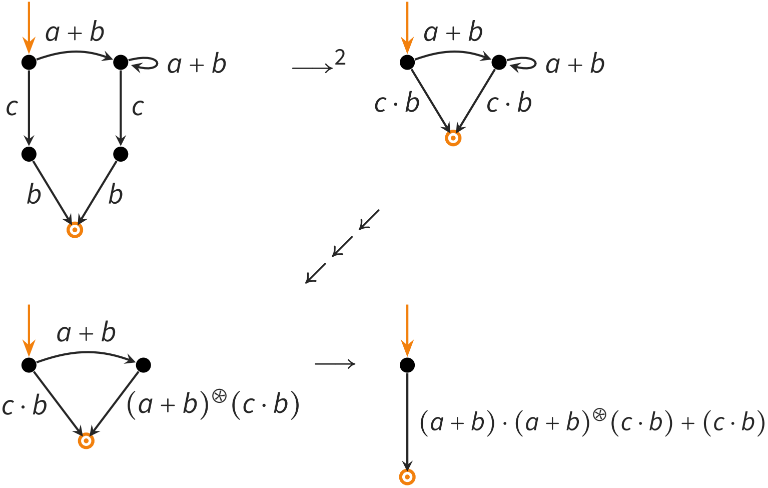


EXAMPLE



$a + b$
$a + b$
c
c
b
b
$\longrightarrow^2$
$a + b$
$a + b$
$c \cdot b$
$c \cdot b$
$a + b$
$c \cdot b$
$(a + b)^{\circledast}(c \cdot b)$
$\longrightarrow$

EXAMPLE


$a+b$
$a+b$
c
c
b
b
$\longrightarrow^2$
$a+b$
$a+b$
$c \cdot b$
$c \cdot b$
$a+b$
$c \cdot b$
$(a+b)^{\circledast}(c \cdot b)$
$\longrightarrow$
$(a+b) \cdot (a+b)^{\circledast}(c \cdot b) + (c \cdot b)$

HOMOMORPHIC IMAGES OF REDUCIBLE GRAPHS

3 Reducible charts are closed under homomorphic images

HOMOMORPHIC IMAGES OF REDUCIBLE GRAPHS

3 Reducible charts are closed under homomorphic images

Lemma

If $f : C_1 \twoheadrightarrow C_2$ is a surjective homomorphism and C_1 is reducible, then C_2 is also reducible.

HOMOMORPHIC IMAGES OF REDUCIBLE GRAPHS

3 Reducible charts are closed under homomorphic images

Lemma

If $f : C_1 \twoheadrightarrow C_2$ is a surjective homomorphism and C_1 is reducible, then C_2 is also reducible.

Proof.

Idea: If C_2 is not reducible, we pull back its forbidden subgraph along f. □

HOMOMORPHIC IMAGES OF REDUCIBLE GRAPHS

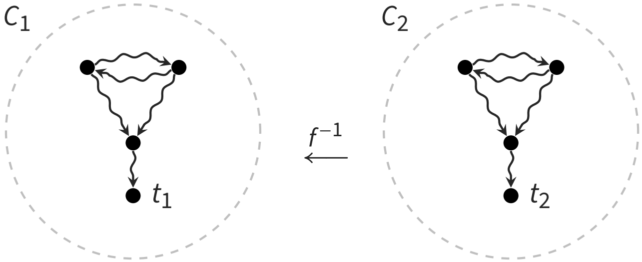
3 Reducible charts are closed under homomorphic images

Lemma

If $f : C_1 \twoheadrightarrow C_2$ is a surjective homomorphism and C_1 is reducible, then C_2 is also reducible.

Proof.

Idea: If C_2 is not reducible, we pull back its forbidden subgraph along f. □



HOMOMORPHIC IMAGES OF REDUCIBLE GRAPHS

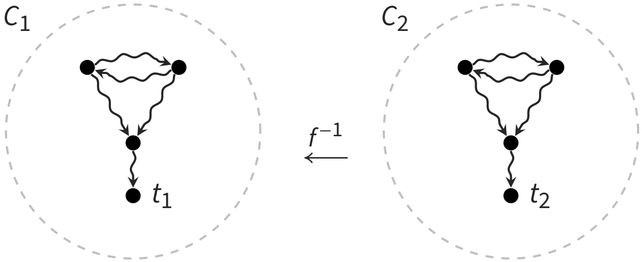
3 Reducible charts are closed under homomorphic images ✅

Lemma

If $f : C_1 \twoheadrightarrow C_2$ is a surjective homomorphism and C_1 is reducible, then C_2 is also reducible.

Proof.

Idea: If C_2 is not reducible, we pull back its forbidden subgraph along f. □



LLEE AND REDUCIBILITY

1 The interpretation of every expression is reducible

Lemma

Every strongly normed LLEE chart is a reducible chart.

LLEE AND REDUCIBILITY

1 The interpretation of every expression is reducible

Lemma

Every strongly normed LLEE chart is a reducible chart.

Proof.

If a chart is not reducible, it has a forbidden subgraph, which we cannot give an LLEE witness. $\square$

LLEE AND REDUCIBILITY

1. The interpretation of every expression is reducible

Lemma

Every strongly normed LLEE chart is a reducible chart.

Proof.

If a chart is not reducible, it has a forbidden subgraph, which we cannot give an LLEE witness. □

LLEE AND REDUCIBILITY

1 The interpretation of every strongly normed expression is reducible ✅

Lemma

Every strongly normed LLEE chart is a reducible chart.

Proof.

If a chart is not reducible, it has a forbidden subgraph, which we cannot give an LLEE witness. □

TYING IT TOGETHER

What we have obtained for reducible charts:

TYING IT TOGETHER

What we have obtained for reducible charts:

1. For every strongly normed expression e, the chart $C(e)$ is reducible;

TYING IT TOGETHER

What we have obtained for reducible charts:

1. For every strongly normed expression e, the chart $C(e)$ is reducible;
2. Every reducible chart admits a unique solution;

TYING IT TOGETHER

What we have obtained for reducible charts:

1. For every strongly normed expression e, the chart $C(e)$ is reducible;
2. Every reducible chart admits a unique solution;
3. The class of reducible charts is closed under homomorphic images.

TYING IT TOGETHER

What we have obtained for reducible charts:

1. For every strongly normed expression e, the chart $C(e)$ is reducible;
2. Every reducible chart admits a unique solution;
3. The class of reducible charts is closed under homomorphic images.

A completeness proof of BBP restricted to strongly normed expressions.

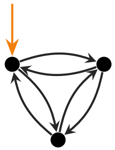
FUTURE WORK

FUTURE WORK

- A completeness proof for all (1-free) star expressions.

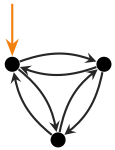
FUTURE WORK

- A completeness proof for all (1-free) star expressions.



FUTURE WORK

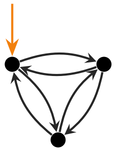
- A completeness proof for all (1-free) star expressions.



- Are LLEE and reducibility equivalent?

FUTURE WORK

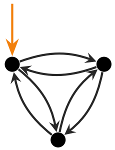
- A completeness proof for all (1-free) star expressions.



- Are LLEE and reducibility equivalent?
- Coalgebraic generalisations?

FUTURE WORK

- A completeness proof for all (1-free) star expressions.



- Are LLEE and reducibility equivalent?
- Coalgebraic generalisations?

Thank you!