Coequational Logic for Finitary Functors

Daniel Schwencke

Department of Theoretical Computer Science
Institute of Technology, Braunschweig

5th April 2008

Motivation

Goal

Deduce properties of (state-based) systems from known properties of these systems.

Motivation

Goal

Deduce properties of (state-based) systems from known properties of these systems.

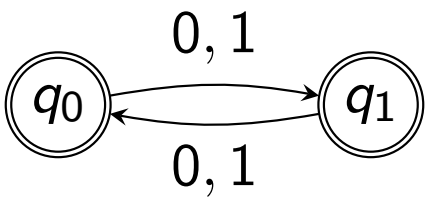
concept	**model as**	**example**
system	coalgebra	deterministic automaton
system property	coequation	each state accepts the empty word
deduction of system properties	coequational logic	each state accepts the empty word $\Rightarrow$ each state accepts every word

Motivation

Goal

Deduce properties of (state-based) systems from known properties of these systems.

concept	**model as**	**example**
system	coalgebra	deterministic automaton
system property	coequation	each state accepts the empty word
deduction of system properties	coequational logic	each state accepts the empty word $\Rightarrow$ each state accepts every word



Outline

Outline

Definitions

Definition (P. Gumm)

Given a cofree coalgebra Q, a coequation is a coatomic set $Q \setminus q$ for some $q \in Q$.

- completely determined by q
- notation $\boxtimes q$ (read: avoid q)

Definitions

Definition (P. Gumm)

Given a cofree coalgebra Q, a coequation is a coatomic set $Q \setminus q$ for some $q \in Q$.

- completely determined by q
- notation $\boxtimes q$ (read: avoid q)

Definition

Given a cofree H-coalgebra (Q, α_Q, γ) on C, an H-coalgebra (S, α_S) satisfies $\boxtimes q$ if $c^*(s) \neq q$ for all colourings $c : S \to C$ and for all $s \in S$.

- $c^* : S \to Q$ denotes the unique H-coalgebra homomorphism with $c = \gamma \circ c^*$

Coequations for deterministic automata

Example

Deterministic automata with binary input alphabet $\{0, 1\}$

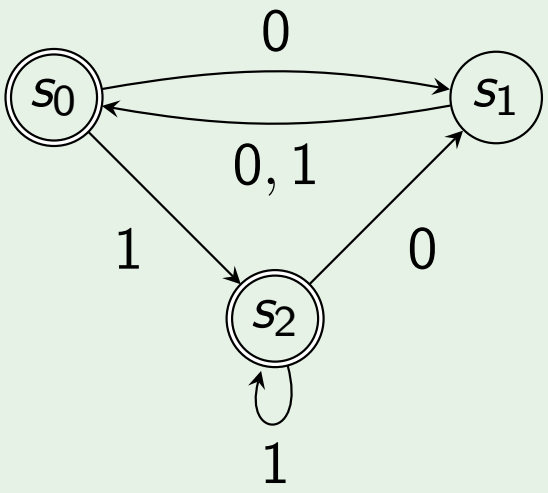
- coalgebras of the polynomial functor $\{tt, ff\} \times (-)^{\{0,1\}}$
- $Q =$ all coloured ordered binary trees without leaves and with node labels from $\{tt, ff\}$
- given $c : S \to C$, $c^* : S \to Q$ maps $s \in S$ to the tree of its outgoing transitions

Coequations for deterministic automata

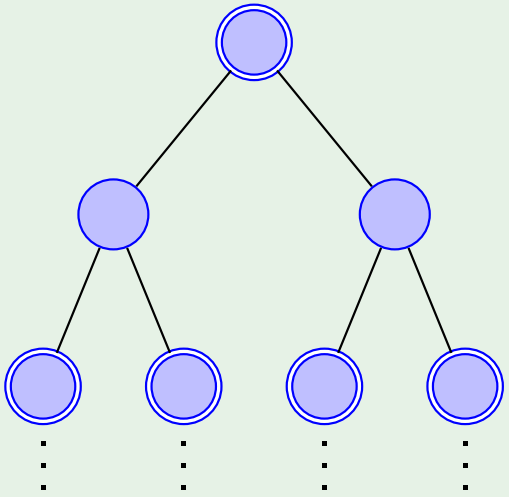
Example

Deterministic automata with binary input alphabet $\{0, 1\}$

- coalgebras of the polynomial functor $\{tt, ff\} \times (-)^{\{0,1\}}$
- Q = all coloured ordered binary trees without leaves and with node labels from $\{tt, ff\}$
- given $c : S \to C$, $c^* : S \to Q$ maps $s \in S$ to the tree of its outgoing transitions



satisfies $\boxtimes$

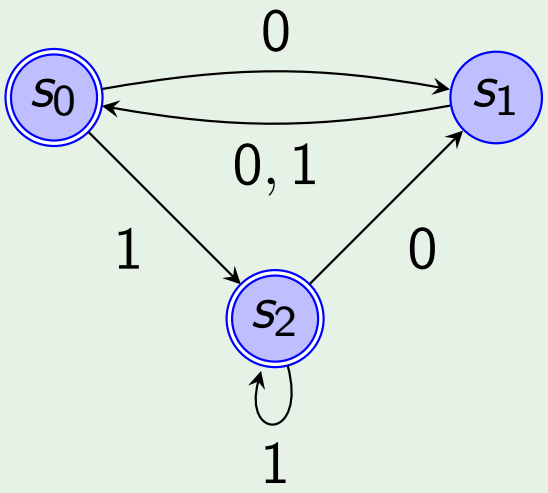


Coequations for deterministic automata

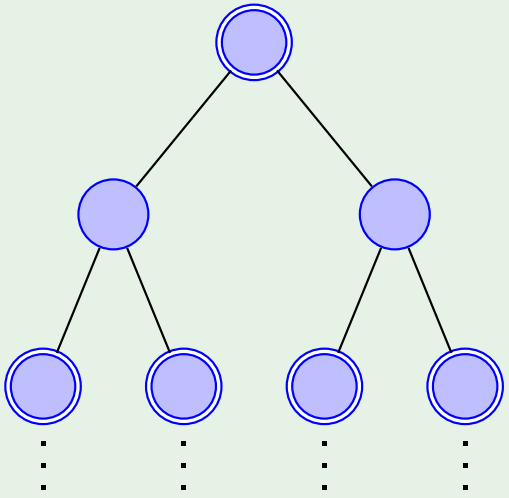
Example

Deterministic automata with binary input alphabet $\{0, 1\}$

- coalgebras of the polynomial functor $\{tt, ff\} \times (-)^{\{0,1\}}$
- Q = all coloured ordered binary trees without leaves and with node labels from $\{tt, ff\}$
- given $c : S \to C$, $c^* : S \to Q$ maps $s \in S$ to the tree of its outgoing transitions



satisfies $\boxtimes$

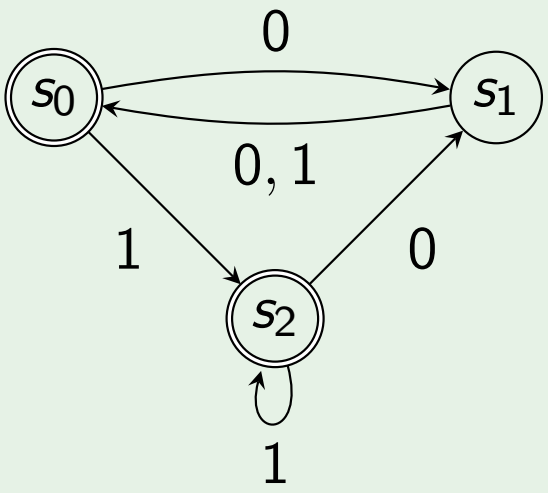


Coequations for deterministic automata

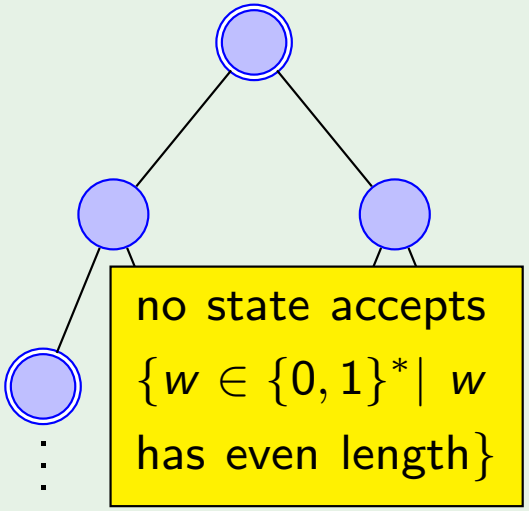
Example

Deterministic automata with binary input alphabet $\{0, 1\}$

- coalgebras of the polynomial functor $\{tt, ff\} \times (-)^{\{0,1\}}$
- Q = all coloured ordered binary trees without leaves and with node labels from $\{tt, ff\}$
- given $c : S \to C$, $c^* : S \to Q$ maps $s \in S$ to the tree of its outgoing transitions



satisfies $\boxtimes$



Semantical consequences

Definition

$\boxtimes q$ is a semantical consequence of $\boxtimes q'$ if every coalgebra satisfying $\boxtimes q'$ also satisfies $\boxtimes q$.

- notation $\boxtimes q' \models \boxtimes q$

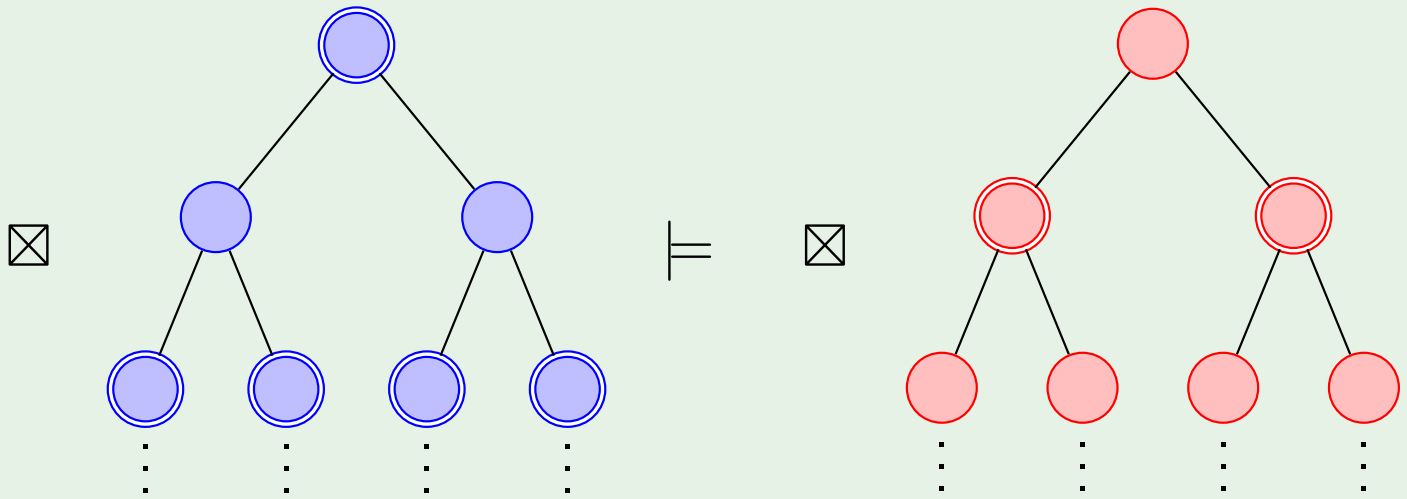
Semantical consequences

Definition

$\boxtimes q$ is a semantical consequence of $\boxtimes q'$ if every coalgebra satisfying $\boxtimes q'$ also satisfies $\boxtimes q$.

- notation $\boxtimes q' \models \boxtimes q$

Example



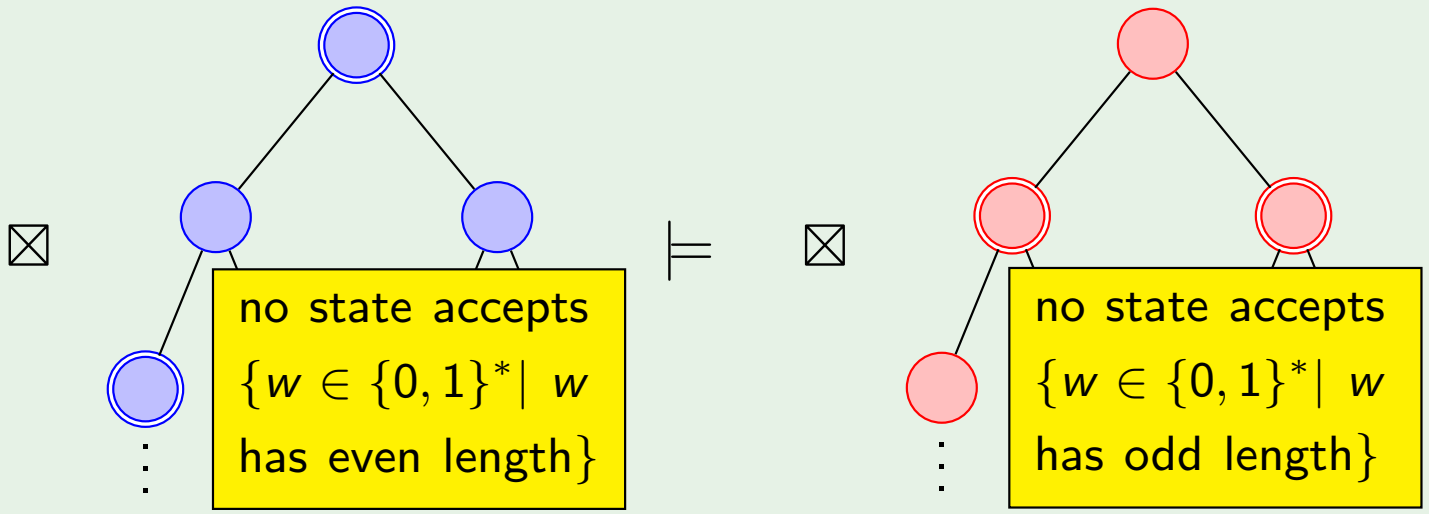
Semantical consequences

Definition

$\boxtimes q$ is a semantical consequence of $\boxtimes q'$ if every coalgebra satisfying $\boxtimes q'$ also satisfies $\boxtimes q$.

- notation $\boxtimes q' \models \boxtimes q$

Example



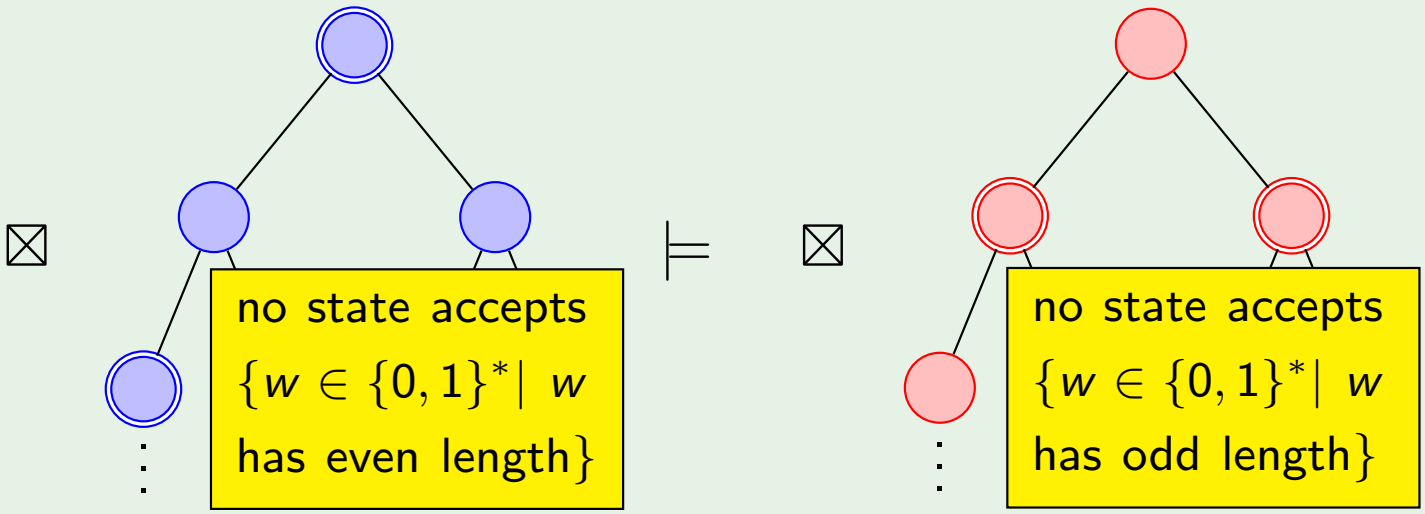
Semantical consequences

Definition

$\boxtimes q$ is a semantical consequence of $\boxtimes q'$ if every coalgebra satisfying $\boxtimes q'$ also satisfies $\boxtimes q$.

- notation $\boxtimes q' \models \boxtimes q$

Example



Is this true?

Outline

The relation $\sqsubseteq$

Definition

q' is a recolouring of q if there exists a homomorphism $h : Q \to Q$ with $h(q) = q'$.

- or equivalently: q and q' have the same shape and equivalent subtrees in q are equivalent in q' again
- write $q' \sqsubseteq q$ if q' is a recolouring of a subtree of q

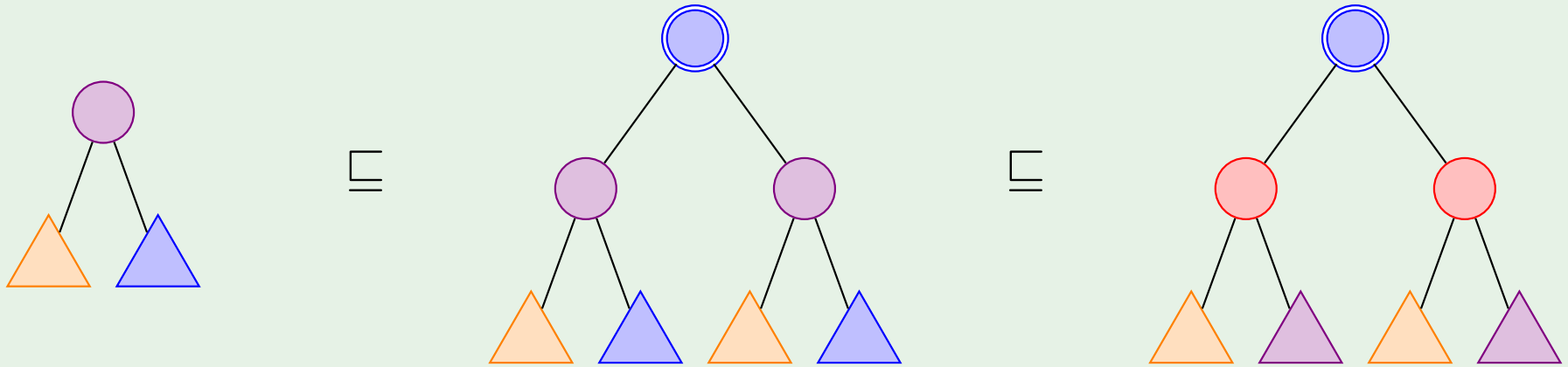
The relation $\sqsubseteq$

Definition

q' is a recolouring of q if there exists a homomorphism $h : Q \to Q$ with $h(q) = q'$.

- or equivalently: q and q' have the same shape and equivalent subtrees in q are equivalent in q' again
- write $q' \sqsubseteq q$ if q' is a recolouring of a subtree of q

Example



Logic for polynomial functors

Theorem (J. Adámek)

$$\{\boxtimes q_i \mid i \in I\} \models \boxtimes q \quad \Longleftrightarrow \quad \exists i \in I : q_i \sqsubseteq q$$

Logic for polynomial functors

Theorem (J. Adámek)

$\{\boxtimes q_i | i \in I\} \models \boxtimes q \quad \Longleftrightarrow \quad \exists i \in I : q_i \sqsubseteq q$

Corollary

Deduction with the following rules is sound and complete.

Child rule:

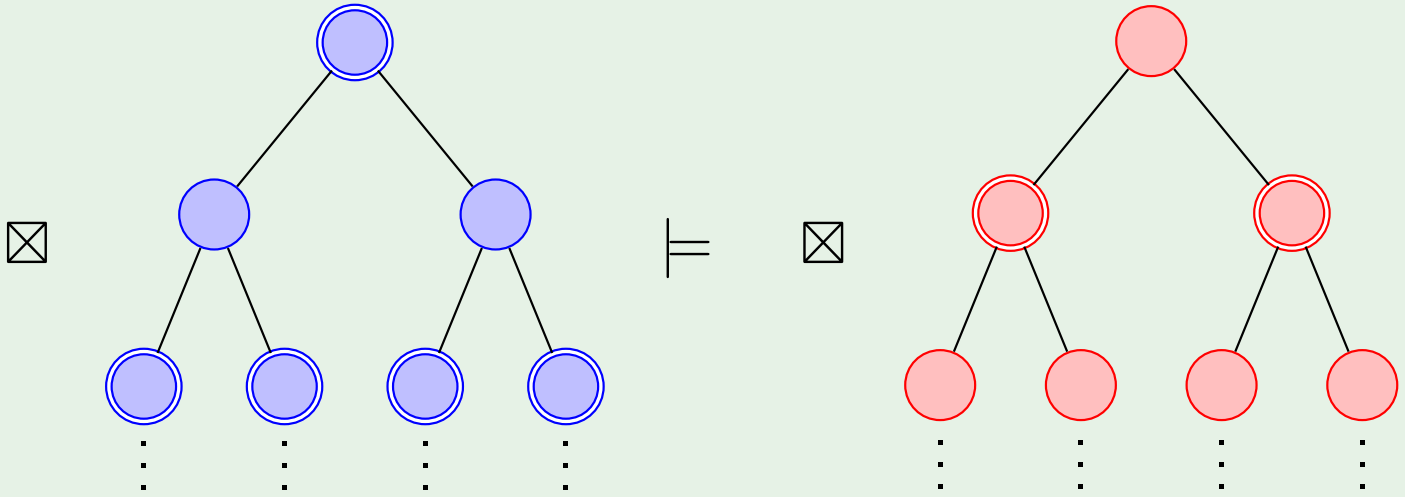
$$\frac{\boxtimes q'}{\boxtimes q} \quad \textit{where } q' \textit{ is a child tree of } q$$

Recolouring rule:

$$\frac{\boxtimes q'}{\boxtimes q} \quad \textit{where } q' \textit{ is recolouring of } q$$

An answer to our question

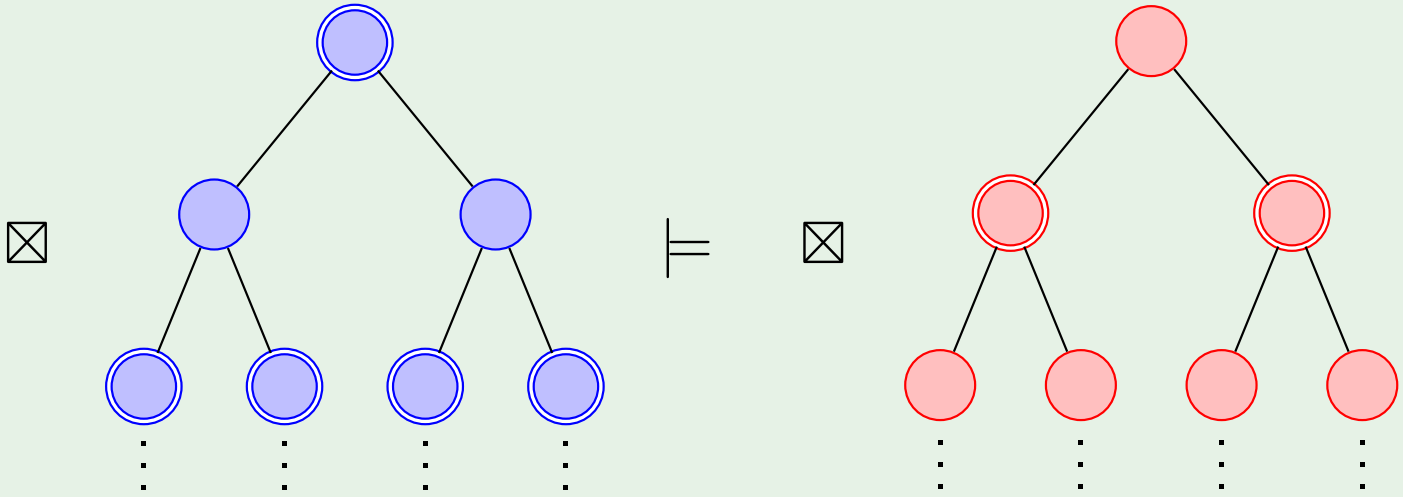
Example



Yes, this is true!

An answer to our question

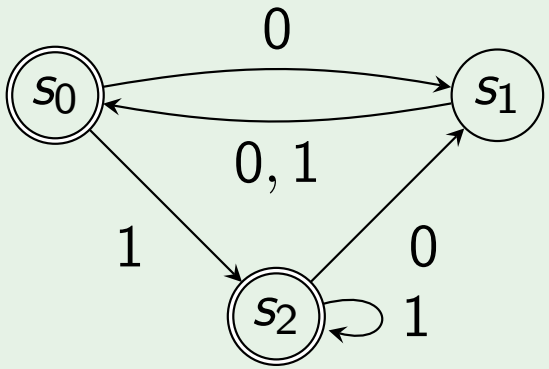
Example



Yes, this is true!

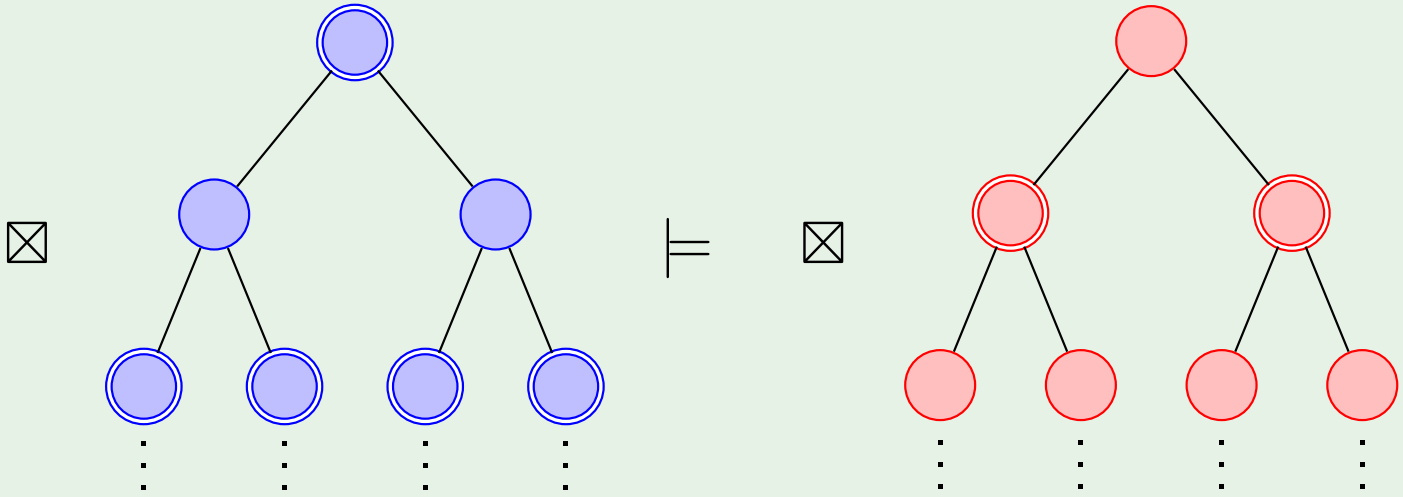
Check the example automaton:

Example



An answer to our question

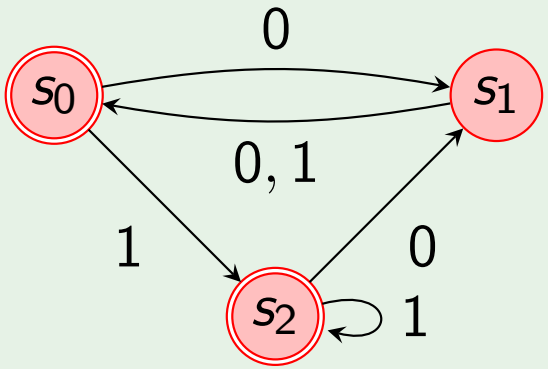
Example



Yes, this is true!

Check the example automaton:

Example



Logic for accessible functors

Accessible functor H

- quotient of some polynomial functor H_Σ
  - via surjective natural transformation $\epsilon_S : H_\Sigma S \twoheadrightarrow HS$
- cofree coalgebra Q quotient of Q_Σ
  - via surjective H-homomorphism $\gamma^*_\Sigma : (Q_\Sigma, \epsilon_{Q_\Sigma} \circ \alpha_{Q_\Sigma}) \twoheadrightarrow (Q, \alpha_Q)$
- notation $[t]$ for the equivalence class of $t \in Q_\Sigma$

Logic for accessible functors

Accessible functor H

- quotient of some polynomial functor H_Σ
  - via surjective natural transformation $\epsilon_S : H_\Sigma S \twoheadrightarrow HS$
- cofree coalgebra Q quotient of Q_Σ
  - via surjective H-homomorphism $\gamma^*_\Sigma : (Q_\Sigma, \epsilon_{Q_\Sigma} \circ \alpha_{Q_\Sigma}) \twoheadrightarrow (Q, \alpha_Q)$
- notation $[t]$ for the equivalence class of $t \in Q_\Sigma$

Theorem (J. Adámek)

$$\{\boxtimes q_i | i \in I\} \models \boxtimes q \quad \Longleftrightarrow \quad \forall t.[t] = q \; \exists i \in I \; \exists r.[r] = q_i : r \sqsubseteq t$$

- no simple deduction rules

Outline

Labelled transition systems

Example

Finitely branching LTS with labels from A

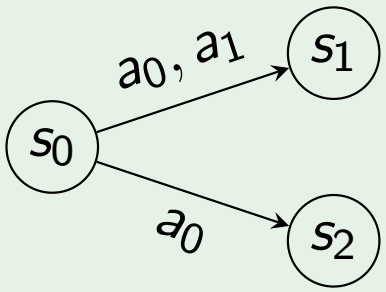
- coalgebras of the functor $\mathcal{P}_f(A \times -)$
- logic for accessible functors applies, but:
- cofree coalgebra has direct description (J. Worrell)
  - Q = coloured unordered strongly extensional trees with edge labels from A
  - notation: t^q represents the equivalence class q

Labelled transition systems

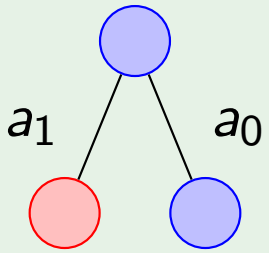
Example

Finitely branching LTS with labels from A

- coalgebras of the functor $\mathcal{P}_f(A \times -)$
- logic for accessible functors applies, but:
- cofree coalgebra has direct description (J. Worrell)
  - Q = coloured unordered strongly extensional trees with edge labels from A
  - notation: t^q represents the equivalence class q



satisfies $\boxtimes$

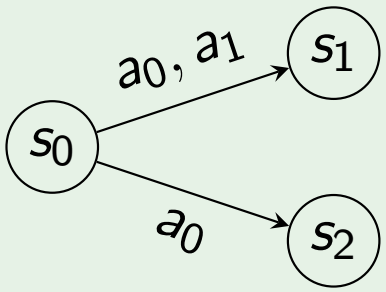


Labelled transition systems

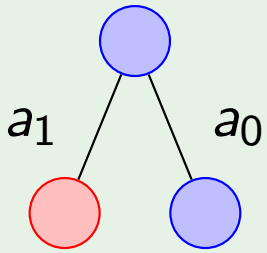
Example

Finitely branching LTS with labels from A

- coalgebras of the functor $\mathcal{P}_f(A \times -)$
- logic for accessible functors applies, but:
- cofree coalgebra has direct description (J. Worrell)
  - Q = coloured unordered strongly extensional trees with edge labels from A
  - notation: t^q represents the equivalence class q



satisfies $\boxtimes$



each state with only a_0-/a_1-transitions to halting states has a common a_0-/a_1-successor

Logic for LTS

Theorem

$$\{\boxtimes q_i \mid i \in I\} \models \boxtimes q \quad \Longleftrightarrow \quad \exists i \in I \; \exists r.[r] = q_i : r \sqsubseteq t^q$$

Logic for LTS

Theorem

$$\{\boxtimes q_i | i \in I\} \models \boxtimes q \quad \Longleftrightarrow \quad \exists i \in I \; \exists r.[r] = q_i : r \sqsubseteq t^q$$

Corollary

Deduction with the following rules is sound and complete.

Child rule:

$$\frac{\boxtimes q'}{\boxtimes q} \quad \textit{where } t^{q'} \textit{ is a child tree of } t^q$$

Recolouring rule:

$$\frac{\boxtimes q'}{\boxtimes q} \quad \textit{where } [r] = q' \textit{ for a recolouring } r \textit{ of } t^q$$

Finitary functors preserving preimages

Finitary functor H

- quotient of some polynomial functor H_Σ with finitary signature Σ
  - Q is quotient of Q_Σ via H-homomorphism γ^*_Σ
- example: the LTS-functor $\mathcal{P}_f(A \times -)$

Finitary functors preserving preimages

Finitary functor H

- quotient of some polynomial functor H_Σ with finitary signature Σ
  - Q is quotient of Q_Σ via H-homomorphism γ^*_Σ
- example: the LTS-functor $\mathcal{P}_f(A \times -)$

Preimages

- pullbacks along monos
- if preserved by H, Q is a "well-behaved" quotient of Q_Σ
- preserved by $\mathcal{P}_f(A \times -)$

Finitary functors preserving preimages

Finitary functor H

- quotient of some polynomial functor H_Σ with finitary signature Σ
  - Q is quotient of Q_Σ via H-homomorphism γ^*_Σ
- example: the LTS-functor $\mathcal{P}_f(A \times -)$

Preimages

- pullbacks along monos
- if preserved by H, Q is a "well-behaved" quotient of Q_Σ
- preserved by $\mathcal{P}_f(A \times -)$

Logic as for LTS

- find a set of representative trees t^q (closed under taking subtrees)
- prove compatibility of equivalence classes with subtrees and recolourings

Equivalence classes and subtrees

Proposition

Let H preserve preimages, then $[t] = [t'] \implies \forall s \exists s' : [s] = [s']$ *(where s, s' are subtrees of t and t' respectively).*

Equivalence classes and subtrees

Proposition

Let H preserve preimages, then $[t] = [t'] \implies \forall s \exists s' : [s] = [s']$ *(where s, s' are subtrees of t and t' respectively).*

Proof.

It suffices to show this for child trees s, s' of t, t'.

$$\begin{tikzcd}
Q_\Sigma \arrow[r, two heads, "\gamma^*_\Sigma"] \arrow[d, "\alpha_{Q_\Sigma}"'] & Q \arrow[dd, "\alpha_Q"] \\
H_\Sigma Q_\Sigma \arrow[d, two heads, "\epsilon_{Q_\Sigma}"'] & \\
HQ_\Sigma \arrow[r, two heads, "H\gamma^*_\Sigma"'] & HQ
\end{tikzcd}$$

□

Equivalence classes and subtrees

Proposition

Let H preserve preimages, then $[t] = [t'] \implies \forall s \exists s' : [s] = [s']$ *(where s, s' are subtrees of t and t' respectively).*

Proof.

It suffices to show this for child trees s, s' of t, t'.

$$\begin{tikzcd}
Q_\Sigma \arrow[r, two heads, "\gamma^*_\Sigma"] \arrow[d, "\alpha_{Q_\Sigma}"'] & Q \arrow[dd, "\alpha_Q"] \\
H_\Sigma Q_\Sigma \arrow[d, two heads, "\epsilon_{Q_\Sigma}"'] & \\
HQ_\Sigma \arrow[r, two heads, "H\gamma^*_\Sigma"'] & HQ
\end{tikzcd}$$

- α_{Q_Σ} and α_Q are isomorphisms

□

Equivalence classes and subtrees

Proposition

Let H preserve preimages, then $[t] = [t'] \implies \forall s \exists s' : [s] = [s']$ *(where s, s' are subtrees of t and t' respectively).*

Proof.

It suffices to show this for child trees s, s' of t, t'.

$$\begin{tikzcd}
Q_\Sigma \arrow[r, two heads, "\gamma^*_\Sigma"] \arrow[d, "\alpha_{Q_\Sigma}"'] & Q \arrow[dd, "\alpha_Q"] \\
H_\Sigma Q_\Sigma \arrow[d, two heads, "\epsilon_{Q_\Sigma}"'] & \\
HQ_\Sigma \arrow[r, two heads, "H\gamma^*_\Sigma"'] & HQ
\end{tikzcd}$$

- α_{Q_Σ} and α_Q are isomorphisms
- α_{Q_Σ} presents $t \in Q_\Sigma$ by its root labels and child trees $c/\sigma(t_1, \ldots, t_n)$

$\square$

Equivalence classes and subtrees

Proposition

Let H preserve preimages, then $[t] = [t'] \implies \forall s \exists s' : [s] = [s']$ *(where s, s' are subtrees of t and t' respectively).*

Proof.

It suffices to show this for child trees s, s' of t, t'.

$$\begin{tikzcd}
Q_\Sigma \arrow[r, two heads, "\gamma^*_\Sigma"] \arrow[d, "\alpha_{Q_\Sigma}"'] & Q \arrow[dd, "\alpha_Q"] \\
H_\Sigma Q_\Sigma \arrow[d, two heads, "\epsilon_{Q_\Sigma}"'] & \\
HQ_\Sigma \arrow[r, two heads, "H\gamma^*_\Sigma"'] & HQ
\end{tikzcd}$$

- α_{Q_Σ} and α_Q are isomorphisms
- α_{Q_Σ} presents $t \in Q_\Sigma$ by its root labels and child trees $c/\sigma(t_1, \ldots, t_n)$
- $ker(H\gamma^*_\Sigma \circ \epsilon_{Q_\Sigma} \circ \alpha_{Q_\Sigma}) = (ker(\epsilon_{Q_\Sigma} \circ \alpha_{Q_\Sigma}) \cup M)^*$ where $M := \{(t, t') \mid \alpha_{Q_\Sigma}(t) = c/\sigma(t_1, \ldots, t_n), \alpha_{Q_\Sigma}(t') = c/\sigma(t'_1, \ldots, t'_n), [t_i] = [t'_i]\}$

$\square$

Equivalence classes and subtrees

Proposition

Let H preserve preimages, then $[t] = [t'] \implies \forall s \exists s' : [s] = [s']$ *(where s, s' are subtrees of t and t' respectively).*

Proof.

It suffices to show this for child trees s, s' of t, t'.

$$\begin{tikzcd}
Q_\Sigma \arrow[r, two heads, "\gamma^*_\Sigma"] \arrow[d, "\alpha_{Q_\Sigma}"'] & Q \arrow[dd, "\alpha_Q"] \\
H_\Sigma Q_\Sigma \arrow[d, two heads, "\epsilon_{Q_\Sigma}"'] & \\
HQ_\Sigma \arrow[r, two heads, "H\gamma^*_\Sigma"'] & HQ
\end{tikzcd}$$

- α_{Q_Σ} and α_Q are isomorphisms
- α_{Q_Σ} presents $t \in Q_\Sigma$ by its root labels and child trees $c/\sigma(t_1, \ldots, t_n)$
- $ker(H\gamma^*_\Sigma \circ \epsilon_{Q_\Sigma} \circ \alpha_{Q_\Sigma}) = (ker(\epsilon_{Q_\Sigma} \circ \alpha_{Q_\Sigma}) \cup M)^*$ where $M := \{(t, t') \mid \alpha_{Q_\Sigma}(t) = c/\sigma(t_1, \ldots, t_n), \alpha_{Q_\Sigma}(t') = c/\sigma(t'_1, \ldots, t'_n), [t_i] = [t'_i]\}$
- pairs in $ker(\epsilon_{Q_\Sigma} \circ \alpha_{Q_\Sigma})$ and M have same child tree equivalence classes, preserved by $\cup$ and *

$\square$

Equivalence classes and recolourings

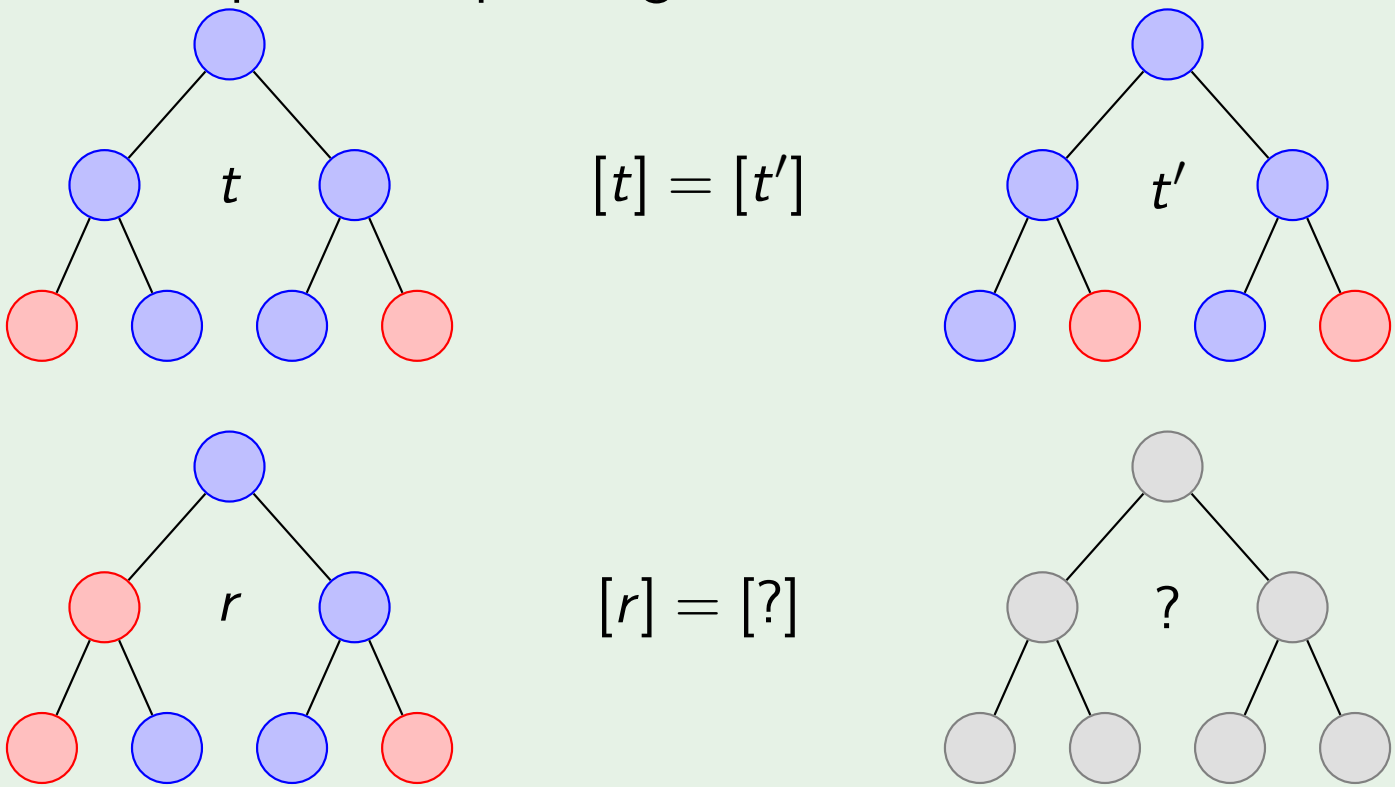
We try: *"Let H preserve preimages, then* $[t] = [t'] \Longrightarrow \forall r \exists r' : [r] = [r']$ *(where r, r' are recolourings of t and t' respectively)."*

Equivalence classes and recolourings

We try: *"Let H preserve preimages, then $[t] = [t'] \Longrightarrow \forall r \exists r' : [r] = [r']$ (where r, r' are recolourings of t and t' respectively)."* But:

Example

Let H be the quotient of $H_\Sigma = (-)^2 + 1$ modulo the order of the binary operation, then H preserves preimages.



Representative trees

Given H as a quotient of H_Σ (Σ finitary), it holds that

- T_Σ consists of finitely branching Σ-trees
- a linear order on Σ induces a linear order on T_Σ
  - lexicographic order: compare trees by breadth-first search
- similarly, an order on $\Sigma \times C$ induces one on Q_Σ

Representative trees

Given H as a quotient of H_Σ (Σ finitary), it holds that

- T_Σ consists of finitely branching Σ-trees
- a linear order on Σ induces a linear order on T_Σ
  - lexicographic order: compare trees by breadth-first search
- similarly, an order on $\Sigma \times C$ induces one on Q_Σ
- for each $q \in Q$ the first tree of that class in the order can be used as the representative tree t^q
- subtrees of these representative trees are representative trees again
  - suppose the opposite and derive a contradiction

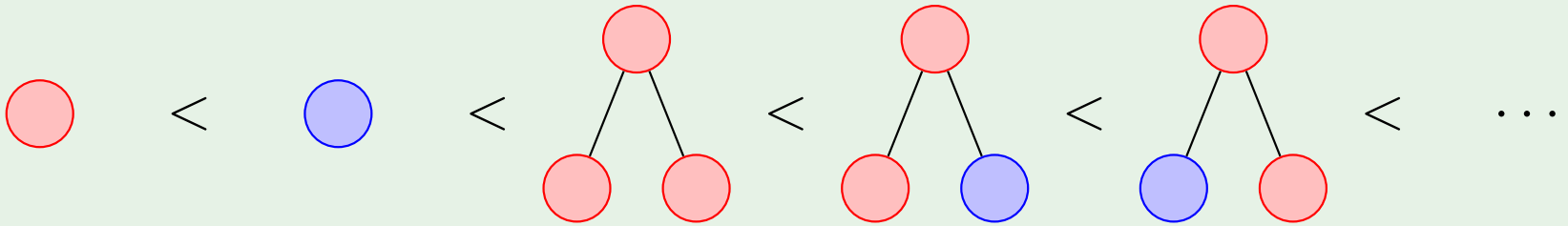
Representative trees

Given H as a quotient of H_Σ (Σ finitary), it holds that

- T_Σ consists of finitely branching Σ-trees
- a linear order on Σ induces a linear order on T_Σ
  - lexicographic order: compare trees by breadth-first search
- similarly, an order on $\Sigma \times C$ induces one on Q_Σ
- for each $q \in Q$ the first tree of that class in the order can be used as the representative tree t^q
- subtrees of these representative trees are representative trees again
  - suppose the opposite and derive a contradiction

Example

Let again $H_\Sigma = (-)^2 + 1$, i. e. $\Sigma = \{c, \sigma\}$ and $C = \{b, r\}$. Order $\Sigma \times C$ by $(c, r) < (c, b) < (\sigma, r) < (\sigma, b)$. Obtain



Equivalence classes and recolourings, continued

Proposition

Let H preserve preimages, then $[t^q] = [t'] \implies \forall r \exists r' : [r] = [r']$ *(where r, r' are recolourings of t^q and t' respectively).*

Equivalence classes and recolourings, continued

Proposition

Let H preserve preimages, then $[t^q] = [t'] \implies \forall r \exists r' : [r] = [r']$ *(where r, r' are recolourings of t^q and t' respectively).*

Proof.

- r induces a map $f' : \mathit{subtrees}(t^q) \to C$
- extend f' to $f : Q_\Sigma \to C$ with respect to equivalence classes
- obtain unique homomorphism $f^* : Q_\Sigma \to Q_\Sigma$ with $f = \gamma_\Sigma \circ f^*$
- obviously $r = f^*(t^q)$, choose $r' := f^*(t')$ and prove $[r] = [r']$

□

Logic for preimage preserving finitary functors

Theorem

$$\{\boxtimes q_i \mid i \in I\} \models \boxtimes q \quad \Longleftrightarrow \quad \exists i \in I \; \exists r.[r] = q_i : r \sqsubseteq t^q$$

Logic for preimage preserving finitary functors

Theorem

$\{\boxtimes q_i | i \in I\} \models \boxtimes q \quad \Longleftrightarrow \quad \exists i \in I \; \exists r.[r] = q_i : r \sqsubseteq t^q$

Proof.

Show equivalence of the above rhs and
$\forall t'.[t'] = q \; \exists i \in I \; \exists r'.[r'] = q_i : r' \sqsubseteq t'$ (logic for accessible functors).

Logic for preimage preserving finitary functors

Theorem

$\{\boxtimes q_i | i \in I\} \models \boxtimes q \quad \Longleftrightarrow \quad \exists i \in I \; \exists r.[r] = q_i : r \sqsubseteq t^q$

Proof.

Show equivalence of the above rhs and
$\forall t'.[t'] = q \; \exists i \in I \; \exists r'.[r'] = q_i : r' \sqsubseteq t'$ (logic for accessible functors).

$\Rightarrow$ Suppose there is r with $[r] = q_i$ for some $i \in I$ and $r \sqsubseteq t^q$

- i. e. r is recolouring of some subtree s of t^q

Logic for preimage preserving finitary functors

Theorem

$\{\boxtimes q_i | i \in I\} \models \boxtimes q \quad \Longleftrightarrow \quad \exists i \in I \; \exists r.[r] = q_i : r \sqsubseteq t^q$

Proof.

Show equivalence of the above rhs and
$\forall t'.[t'] = q \; \exists i \in I \; \exists r'.[r'] = q_i : r' \sqsubseteq t'$ (logic for accessible functors).

$\Rightarrow$ Suppose there is r with $[r] = q_i$ for some $i \in I$ and $r \sqsubseteq t^q$

- i. e. r is recolouring of some subtree s of t^q
- for every tree t' with $[t'] = [t^q]$ we can find a subtree s' of t' with $[s'] = [s]$
- s is a representantive tree because t^q is

Logic for preimage preserving finitary functors

Theorem

$\{\boxtimes q_i | i \in I\} \models \boxtimes q \quad \Longleftrightarrow \quad \exists i \in I \; \exists r.[r] = q_i : r \sqsubseteq t^q$

Proof.

Show equivalence of the above rhs and
$\forall t'.[t'] = q \; \exists i \in I \; \exists r'.[r'] = q_i : r' \sqsubseteq t'$ (logic for accessible functors).

$\Rightarrow$ Suppose there is r with $[r] = q_i$ for some $i \in I$ and $r \sqsubseteq t^q$

- i. e. r is recolouring of some subtree s of t^q
- for every tree t' with $[t'] = [t^q]$ we can find a subtree s' of t' with $[s'] = [s]$
- s is a representantive tree because t^q is
- for s' with $[s'] = [s]$ we can find a recolouring r' of s' with $[r'] = [r]$
- together: $r' \sqsubseteq t'$ and $[r'] = [r] = q_i$

Logic for preimage preserving finitary functors

Theorem

$\{\boxtimes q_i | i \in I\} \models \boxtimes q \quad \Longleftrightarrow \quad \exists i \in I \; \exists r.[r] = q_i : r \sqsubseteq t^q$

Proof.

Show equivalence of the above rhs and
$\forall t'.[t'] = q \; \exists i \in I \; \exists r'.[r'] = q_i : r' \sqsubseteq t'$ (logic for accessible functors).

$\Rightarrow$ Suppose there is r with $[r] = q_i$ for some $i \in I$ and $r \sqsubseteq t^q$

- i. e. r is recolouring of some subtree s of t^q
- for every tree t' with $[t'] = [t^q]$ we can find a subtree s' of t' with $[s'] = [s]$
- s is a representantive tree because t^q is
- for s' with $[s'] = [s]$ we can find a recolouring r' of s' with $[r'] = [r]$
- together: $r' \sqsubseteq t'$ and $[r'] = [r] = q_i$

$\Leftarrow$ Suppose the formula for accessible functors to be true

- apply it to the representative tree t^q ($[t^q] = q$)

$\square$

Outline

Finitary functors not preserving preimages

Theorem

A finitary functor H preserves preimages if and only if for every linear order of $\Sigma \times C$ we have

$$\{\boxtimes q_i | i \in I\} \models \boxtimes q \quad \Longleftrightarrow \quad \exists i \in I \; \exists r.[r] = q_i : r \sqsubseteq t^q.$$

Finitary functors not preserving preimages

Theorem

A finitary functor H preserves preimages if and only if for every linear order of $\Sigma \times C$ we have

$$\{\boxtimes q_i | i \in I\} \models \boxtimes q \quad \Longleftrightarrow \quad \exists i \in I \; \exists r.[r] = q_i : r \sqsubseteq t^q.$$

Nevertheless, there are finitary functors not preserving preimages to which the simple logic still applies in case of a special order.

Example

Let H be the quotient of $H_\Sigma = (-)^2 + 1$ ($\Sigma = \{c, \sigma\}$) modulo the equation $\sigma(x, x) = c$. Then H does not preserve preimages, but given an order of $\Sigma \times C$ where $(c, k) < (\sigma, k)$ for all $k \in C$, the simple logic applies.

Summary

- coequations as system properties
- coequational logic for
  - polynomial functors
  - accessible functors
- finitary functors preserving preimages
  - example: LTS
  - how to find a simpler logic

Summary / Literature

- coequations as system properties
- coequational logic for
  - polynomial functors
  - accessible functors
- finitary functors preserving preimages
  - example: LTS
  - how to find a simpler logic

- J. Adámek.
  A Logic of Coequations.
  In: L. Ong, editor, *Computer Science Logic – Proceedings of the 19th International Workshop, LNCS 3634* (2005), pp. 70–86.
- D. Schwencke.
  Coequational Logic for Finitary Functors.
  To appear in ENTCS (CMCS'08 volume).

Thank you...

...for your attention!

schwencke@iti.cs.tu-bs.de

Properties presentable by coequations

Definition

A covariety is a class of coalgebras that is closed under sums, subcoalgebras and homomorphic images.

Properties presentable by coequations

Definition

A covariety is a class of coalgebras that is closed under sums, subcoalgebras and homomorphic images.

Theorem (Co-Birkhoff-Theorem, J. Rutten)

Covarieties are exactly the classes of coalgebras defined by coequations (where the latter means that the class contains all coalgebras satisfying the coequations).

Properties presentable by coequations

Definition

A covariety is a class of coalgebras that is closed under sums, subcoalgebras and homomorphic images.

Theorem (Co-Birkhoff-Theorem, J. Rutten)

Covarieties are exactly the classes of coalgebras defined by coequations (where the latter means that the class contains all coalgebras satisfying the coequations).

Corollary

The system properties coequations stand for are exactly the ones preserved by sums, subcoalgebras and homomorphic images of coalgebras.

A presentation of finitary functors

Definition

Let X be a set of variables. For two operations $\sigma, \tau \in \Sigma$ the expression

$$\sigma(x_1, \ldots, x_n) = \tau(y_1, \ldots, y_m) \qquad x_i, y_j \in X$$

is called an ϵ-equation if $\epsilon_X : H_\Sigma X \twoheadrightarrow HX$ merges the two sides.

A presentation of finitary functors

Definition

Let X be a set of variables. For two operations $\sigma, \tau \in \Sigma$ the expression

$$\sigma(x_1, \ldots, x_n) = \tau(y_1, \ldots, y_m) \qquad x_i, y_j \in X$$

is called an ϵ-equation if $\epsilon_X : H_\Sigma X \twoheadrightarrow HX$ merges the two sides.

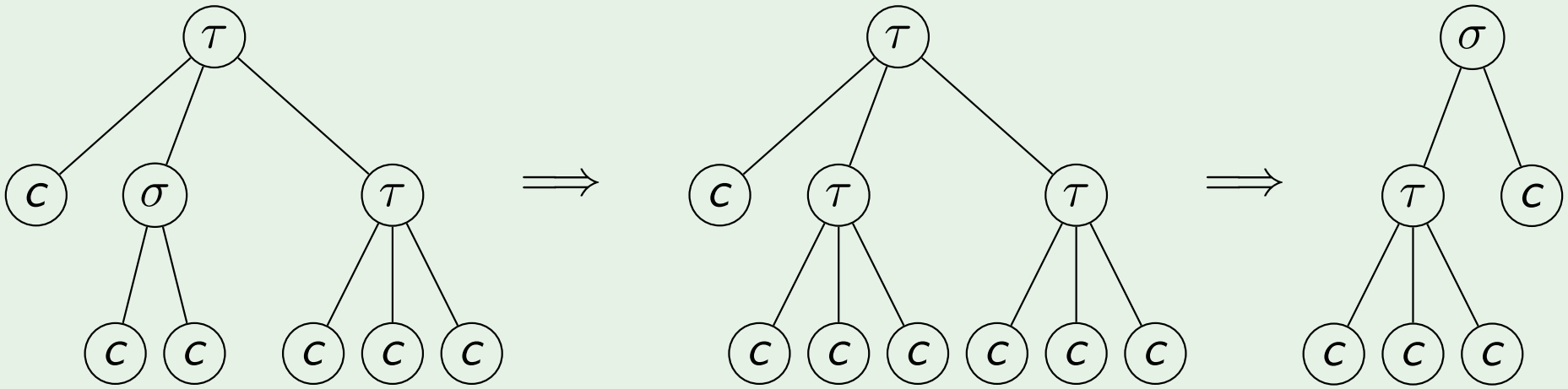
Application of an ϵ-equation (from left to right) to $t \in T_\Sigma$ at node v

- possible if
  - v is labelled by σ and
  - two of its children at positions k, l root the same tree whenever $x_k = x_l$
- it means
  - relabelling v by τ and
  - rearranging the child trees of v according to the rhs of the equation (for new variables arbitrary trees are chosen)

Application of ϵ-equations

Example

Let $\Sigma = \{c, \sigma, \tau\}$ (c constant, σ binary and τ ternary). We apply the ϵ-equation $\sigma(x, y) = \tau(y, x, x)$.



A presentation of finitary functors (continued)

Definition

We say t' can be obtained from t by (possibly infinitely many) applications of ϵ-equations if for every $k \in \mathbb{N}$ the tree $\partial_k t'$ can be obtained from $\partial_k t$ by (finitely many) applications of ϵ-equations.

- $\partial_k t$ denotes the cutting of t at level k

A presentation of finitary functors (continued)

Definition

We say t' can be obtained from t by (possibly infinitely many) applications of ϵ-equations if for every $k \in \mathbb{N}$ the tree $\partial_k t'$ can be obtained from $\partial_k t$ by (finitely many) applications of ϵ-equations.

- $\partial_k t$ denotes the cutting of t at level k

Theorem (J. Adámek, S. Milius)

The terminal H-coalgebra T is the quotient of T_Σ modulo the congruence of application of ϵ-equations.

- similar for cofree coalgebras

ϵ-equations and preimages

Definition

An ϵ-equation is called regular if the sets of variables on both sides are the same.

ϵ-equations and preimages

Definition

An ϵ-equation is called regular if the sets of variables on both sides are the same.

Theorem (J. Adámek, D. Lücke, S. Milius)

A finitary functor H preserves preimages if and only if there is a presentation of H as a quotient of some H_Σ via regular ϵ-equations.