Logic for Communicating Automata with Parameterized Topology

Benedikt Bollig

Laboratoire Spécification et Vérification

Séminaire Vérification LIAFA
27 janvier 2014

Introduction

Objective

A Büchi-Elgot-Trakhtenbrot theorem for communicating automata with parameterized topology.

Introduction

Objective

A Büchi-Elgot-Trakhtenbrot theorem for communicating automata with parameterized topology.

Bridges the gap between high-level specifications and system models.

Introduction

Objective

A Büchi-Elgot-Trakhtenbrot theorem for communicating automata with parameterized topology.

Bridges the gap between high-level specifications and system models.

Theorem (Büchi-Elgot-Trakhtenbrot '60s)

Every MSO formula is equivalent to some finite automaton, and vice versa.

Introduction

Objective

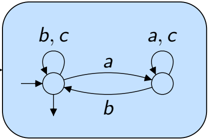
A Büchi-Elgot-Trakhtenbrot theorem for communicating automata with parameterized topology.

Bridges the gap between high-level specifications and system models.

Theorem (Büchi-Elgot-Trakhtenbrot '60s)

Every MSO formula is equivalent to some finite automaton, and vice versa.

$$\forall x(a(x) \to \exists y(x \leq y \wedge b(y)))$$



Introduction

Objective

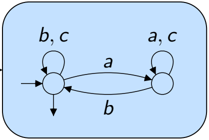
A Büchi-Elgot-Trakhtenbrot theorem for communicating automata with parameterized topology.

Bridges the gap between high-level specifications and system models.

Theorem (Büchi-Elgot-Trakhtenbrot ’60s)

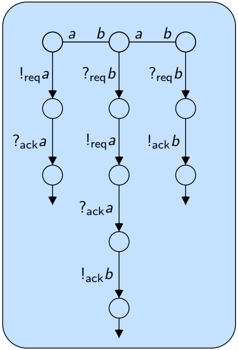
Every MSO formula is equivalent to some finite automaton, and vice versa.

$$\forall x(a(x) \rightarrow \exists y(x \leq y \wedge b(y)))$$

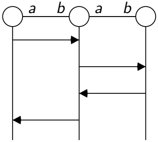


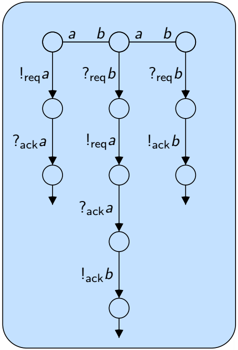
Has been extended to trees, graphs, weighted automata, ...

… and communicating automata (CA)

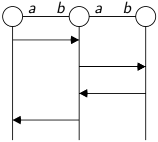


… and communicating automata (CA)

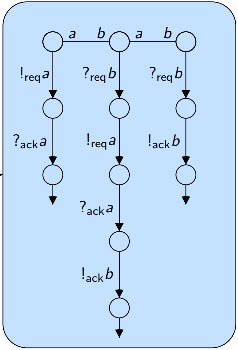




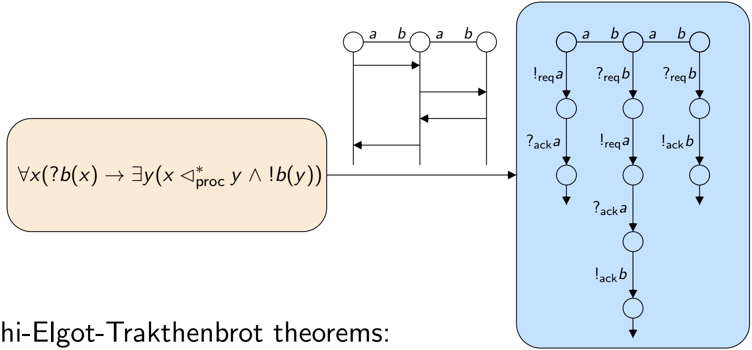
… and communicating automata (CA)



$$\forall x(?b(x) \to \exists y(x \lhd^*_{\text{proc}} y \wedge !b(y))$$



... and communicating automata (CA)

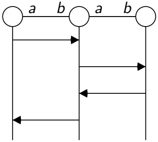


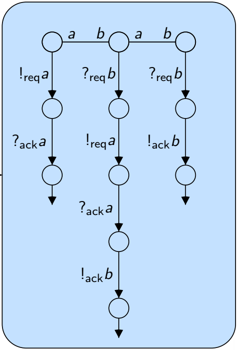
Büchi-Elgot-Trakthenbrot theorems:

- $\forall$-bounded channels [Henriksen-Mukund-Kumar-Sohoni-Thiagarajan 2000]
- $\forall$-bounded channels [Kuske 2002]
- $\exists$-bounded channels [Genest-Kuske-Muscholl 2004]
- unbounded channels (but weaker logic) [B.-Leucker 2004]

... and communicating automata (CA)

$\forall x(?b(x) \rightarrow \exists y(x \lhd^*_{\text{proc}} y \wedge !b(y))$





Büchi-Elgot-Trakthenbrot theorems:

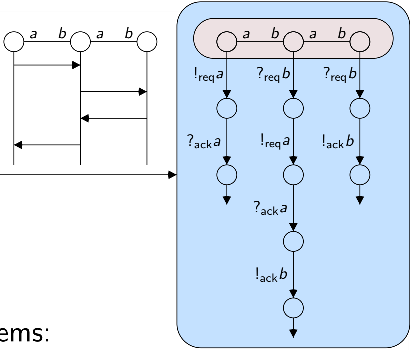
- $\forall$-bounded channels [Henriksen-Mukund-Kumar-Sohoni-Thiagarajan 2000]
- $\forall$-bounded channels [Kuske 2002]
- $\exists$-bounded channels [Genest-Kuske-Muscholl 2004]
- unbounded channels (but weaker logic) [B.-Leucker 2004]

But ...

... all results require communication topology to be fixed!

… and communicating automata (CA)

$\forall x(?b(x) \rightarrow \exists y(x \lhd^{*}_{\mathrm{proc}} y \wedge !b(y))$



Büchi-Elgot-Trakthenbrot theorems:

- ∀-bounded channels [Henriksen-Mukund-Kumar-Sohoni-Thiagarajan 2000]
- ∀-bounded channels [Kuske 2002]
- ∃-bounded channels [Genest-Kuske-Muscholl 2004]
- unbounded channels (but weaker logic) [B.-Leucker 2004]

But …

… all results require communication topology to be fixed!

Towards a parameterized version

Parameterized realizability

Let φ be a formula and $\mathfrak{T}$ be a class of topologies.

Is there a CA that is equivalent to φ on all topologies $\mathcal{T} \in \mathfrak{T}$?

Towards a parameterized version

Parameterized realizability

Let φ be a formula and $\mathfrak{T}$ be a class of topologies.

Is there a CA that is equivalent to φ on all topologies $\mathcal{T} \in \mathfrak{T}$?

More precisely:

- φ is an MSO formula over MSCs (directed acyclic graphs)
- $\mathfrak{T}$ is a class of topologies of bounded degree (such as pipelines, trees, grids, and rings)

Towards a parameterized version

Parameterized realizability

Let φ be a formula and $\mathfrak{T}$ be a class of topologies.

Is there a CA that is equivalent to φ on all topologies $\mathcal{T} \in \mathfrak{T}$?

More precisely:

- φ is an MSO formula over MSCs (directed acyclic graphs)
- $\mathfrak{T}$ is a class of topologies of bounded degree (such as pipelines, trees, grids, and rings)

⇨ Need for new notions

- Topologies (of bounded degree)
- Parameterized communicating automata (PCA)

Outline

- Topologies and MSCs

Outline

- Topologies and MSCs
- Parameterized communicating automata

Outline

- Topologies and MSCs
- Parameterized communicating automata
- MSO logic

Outline

- Topologies and MSCs
- Parameterized communicating automata
- MSO logic
- Negative results:

  There is a formula $\varphi \in \mathcal{C}$ that is not realizable for $\mathfrak{T}$.

Outline

- Topologies and MSCs
- Parameterized communicating automata
- MSO logic
- Negative results:

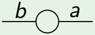
  There is a formula $\varphi \in \mathcal{C}$ that is not realizable for $\mathfrak{T}$.
- Positive results:

  All formulas $\varphi \in \mathcal{C}$ are realizable for $\mathfrak{T}$.

Topologies and MSCs

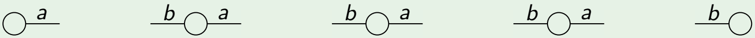
Topologies

Topology



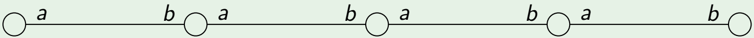
Topologies

Topology



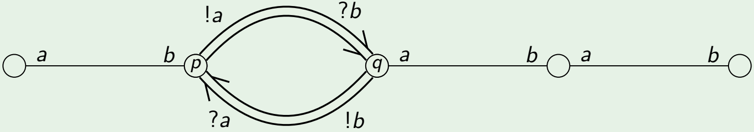
Topologies

Topology



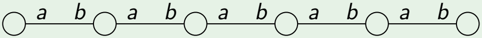
Topologies

Topology



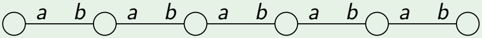
Topologies

Pipeline



Topologies

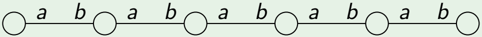
Pipeline



Fix finite set $\mathcal{N} = \{a, b, c, \ldots\}$ of interface names.

Topologies

Pipeline



Fix finite set $\mathcal{N} = \{a, b, c, \dots\}$ of interface names.

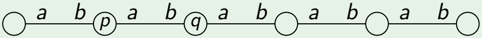
Definition

A topology over $\mathcal{N}$ is a pair $\mathcal{T} = (P, \longmapsto)$ where

- P is the nonempty finite set of processes
- $\longmapsto \subseteq P \times \mathcal{N} \times \mathcal{N} \times P$ is the edge relation

Topologies

Pipeline



Fix finite set $\mathcal{N} = \{a, b, c, \ldots\}$ of interface names.

Definition

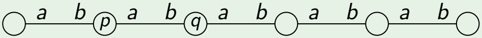
A topology over $\mathcal{N}$ is a pair $\mathcal{T} = (P, \longmapsto)$ where

- P is the nonempty finite set of processes
- $\longmapsto \subseteq P \times \mathcal{N} \times \mathcal{N} \times P$ is the edge relation

Whenever $p \overset{a \quad b}{\longmapsto} q$, the following hold:

Topologies

Pipeline



Fix finite set $\mathcal{N} = \{a, b, c, \ldots\}$ of interface names.

Definition

A topology over $\mathcal{N}$ is a pair $\mathcal{T} = (P, \longmapsto)$ where

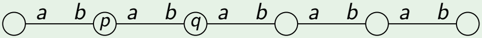
- P is the nonempty finite set of processes
- $\longmapsto \subseteq P \times \mathcal{N} \times \mathcal{N} \times P$ is the edge relation

Whenever $p \overset{a \quad b}{\longmapsto} q$, the following hold:

1. $p \neq q$

Topologies

Pipeline



Fix finite set $\mathcal{N} = \{a, b, c, \ldots\}$ of interface names.

Definition

A topology over $\mathcal{N}$ is a pair $\mathcal{T} = (P, \longmapsto)$ where

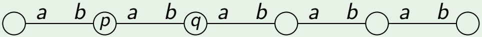
- P is the nonempty finite set of processes
- $\longmapsto \subseteq P \times \mathcal{N} \times \mathcal{N} \times P$ is the edge relation

Whenever $p \overset{a\quad b}{\longmapsto} q$, the following hold:

1. $p \neq q$
2. $q \overset{b\quad a}{\longmapsto} p$

Topologies

Pipeline



Fix finite set $\mathcal{N} = \{a, b, c, \ldots\}$ of interface names.

Definition

A topology over $\mathcal{N}$ is a pair $\mathcal{T} = (P, \longmapsto)$ where

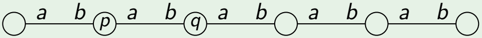
- P is the nonempty finite set of processes
- $\longmapsto \subseteq P \times \mathcal{N} \times \mathcal{N} \times P$ is the edge relation

Whenever $p \overset{a \quad b}{\longmapsto} q$, the following hold:

1. $p \neq q$
2. $q \overset{b \quad a}{\longmapsto} p$
3. $p \overset{a' \quad b'}{\longmapsto} q'$ implies $(a = a' \iff q = q')$

Topologies

Pipeline



process type $\{a\}$ $\{a,b\}$ $\{a,b\}$ $\{a,b\}$ $\{a,b\}$ $\{b\}$

Fix finite set $\mathcal{N} = \{a, b, c, \dots\}$ of interface names.

Definition

A topology over $\mathcal{N}$ is a pair $\mathcal{T} = (P, \longmapsto)$ where

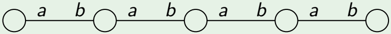
- P is the nonempty finite set of processes
- $\longmapsto \subseteq P \times \mathcal{N} \times \mathcal{N} \times P$ is the edge relation

Whenever $p \overset{a\quad b}{\longmapsto} q$, the following hold:

1. $p \neq q$
2. $q \overset{b\quad a}{\longmapsto} p$
3. $p \overset{a'\quad b'}{\longmapsto} q'$ implies $(a = a' \iff q = q')$

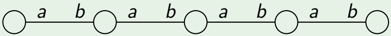
Topologies

Pipeline $\mathcal{T}^5_{\text{lin}}$

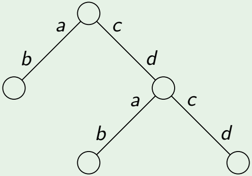


Topologies

Pipeline $\mathcal{T}^5_{\mathrm{lin}}$

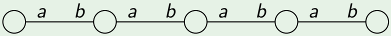


Tree

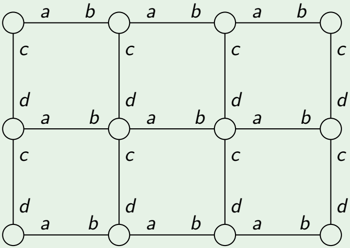


Topologies

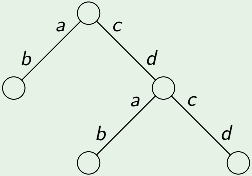
Pipeline $\mathcal{T}_{\text{lin}}^{5}$



Grid $\mathcal{T}_{\text{grid}}^{3,4}$

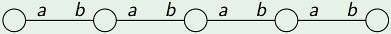


Tree

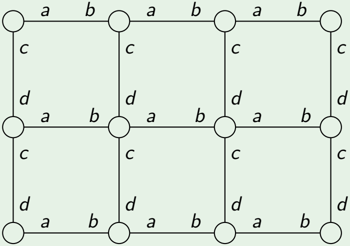


Topologies

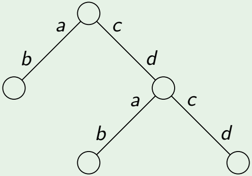
Pipeline $\mathcal{T}^5_{\mathrm{lin}}$



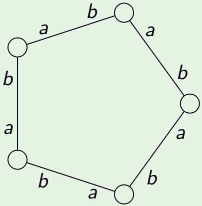
Grid $\mathcal{T}^{3,4}_{\mathrm{grid}}$



Tree

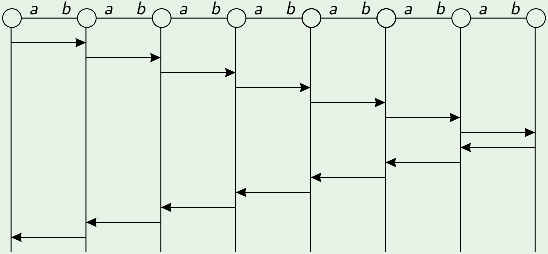


Ring $\mathcal{T}^5_{\mathrm{ring}}$



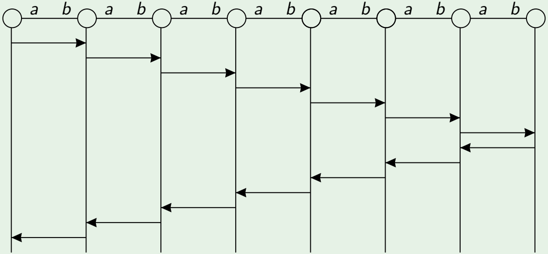
Message Sequence Charts (MSCs)

MSC



Message Sequence Charts (MSCs)

MSC

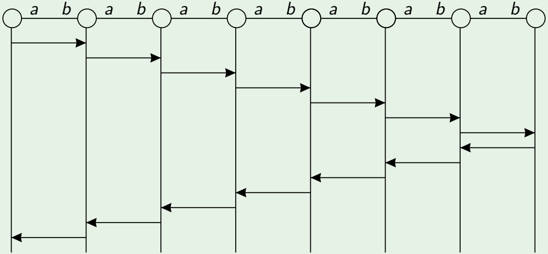


Definition

An MSC over $\mathcal{T} = (P, \longmapsto)$ is a triple $M = (E, \lhd, \ell)$ where

Message Sequence Charts (MSCs)

MSC



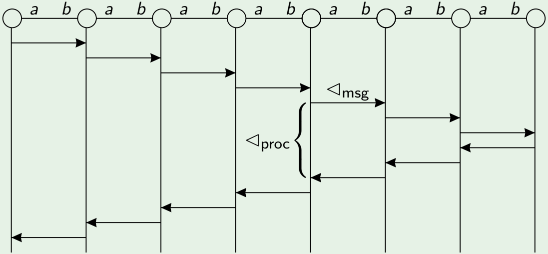
Definition

An MSC over $\mathcal{T} = (P, \longmapsto)$ is a triple $M = (E, \lhd, \ell)$ where

- E is the nonempty finite set of events

Message Sequence Charts (MSCs)

MSC



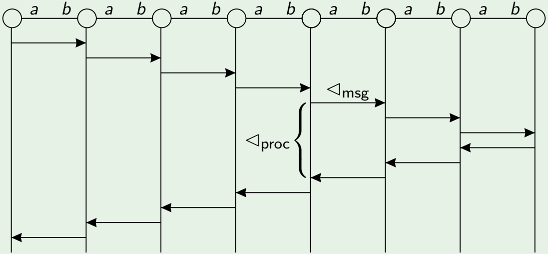
Definition

An MSC over $\mathcal{T} = (P, \longmapsto)$ is a triple $M = (E, \lhd, \ell)$ where

- E is the nonempty finite set of events
- $\lhd = \lhd_{\text{proc}} \uplus \lhd_{\text{msg}} \subseteq E \times E$ acyclic

Message Sequence Charts (MSCs)

MSC



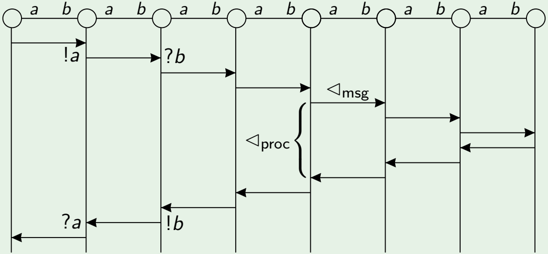
Definition

An MSC over $\mathcal{T} = (P, \longmapsto)$ is a triple $M = (E, \lhd, \ell)$ where

- E is the nonempty finite set of events
- $\lhd = \lhd_{proc} \uplus \lhd_{msg} \subseteq E \times E$ acyclic
- $\ell : E \to P$

Message Sequence Charts (MSCs)

MSC



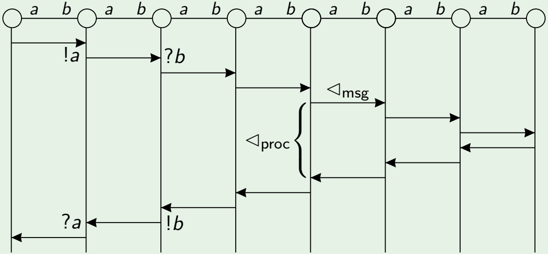
Definition

An MSC over $\mathcal{T} = (P, \longmapsto)$ is a triple $M = (E, \lhd, \ell)$ where

- E is the nonempty finite set of events
- $\lhd = \lhd_{\text{proc}} \uplus \lhd_{\text{msg}} \subseteq E \times E$ acyclic
- $\ell : E \to P$

Message Sequence Charts (MSCs)

MSC



Definition

An MSC over $\mathcal{T} = (P, \longmapsto)$ is a triple $M = (E, \lhd, \ell)$ where

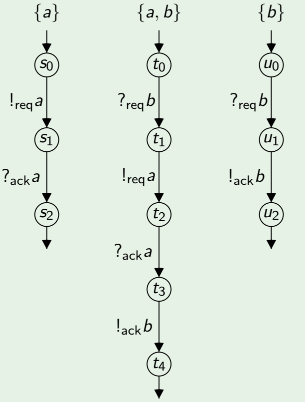
- E is the nonempty finite set of events
- $\lhd = \lhd_{\mathsf{proc}} \uplus \lhd_{\mathsf{msg}} \subseteq E \times E$ acyclic
- $\ell : E \to P$

$+$ some extra conditions

Parameterized Communicating Automata (PCA)

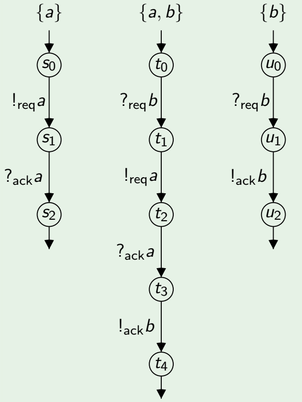
Parameterized communicating automata (PCA)

PCA $\mathcal{A}$ over $\{a, b\}$

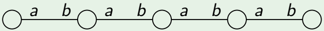


Parameterized communicating automata (PCA)

PCA $\mathcal{A}$ over $\{a, b\}$

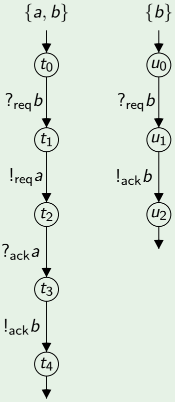


PCA $\mathcal{A}$ running on $\mathcal{T}^5_{\text{lin}}$

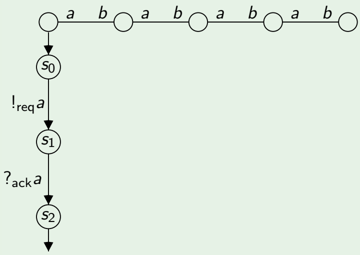


Parameterized communicating automata (PCA)

PCA $\mathcal{A}$ over $\{a, b\}$

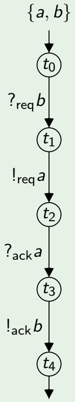


PCA $\mathcal{A}$ running on $\mathcal{T}^5_{\text{lin}}$

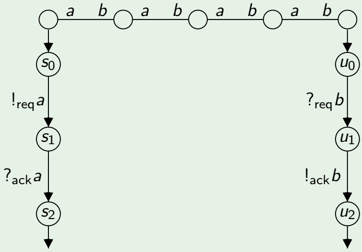


Parameterized communicating automata (PCA)

PCA $\mathcal{A}$ over $\{a, b\}$

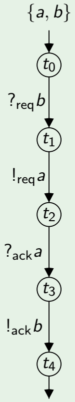


PCA $\mathcal{A}$ running on $\mathcal{T}^5_{\mathrm{lin}}$

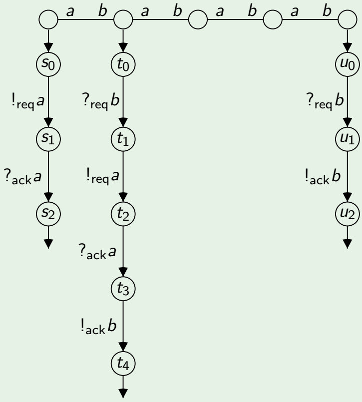


Parameterized communicating automata (PCA)

PCA $\mathcal{A}$ over $\{a, b\}$

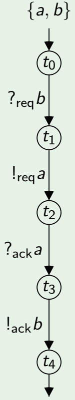


PCA $\mathcal{A}$ running on $\mathcal{T}^5_{\text{lin}}$

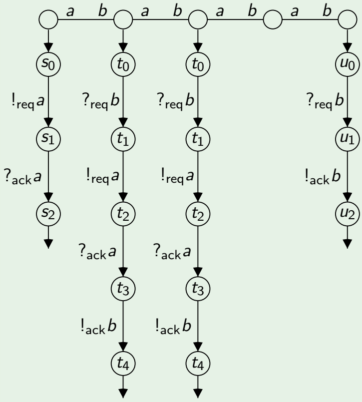


Parameterized communicating automata (PCA)

PCA $\mathcal{A}$ over $\{a, b\}$

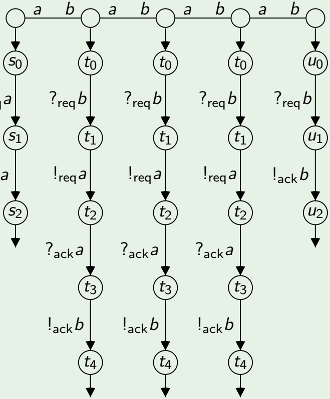


PCA $\mathcal{A}$ running on $\mathcal{T}^5_{\mathrm{lin}}$



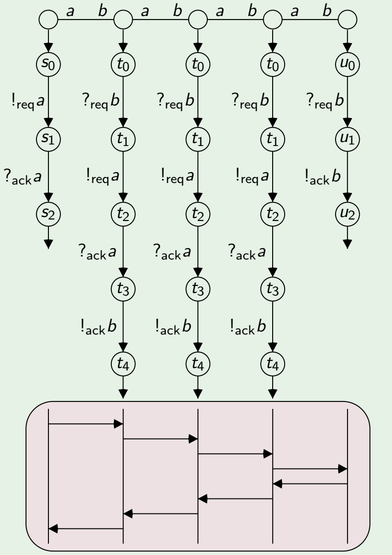
Parameterized communicating automata (PCA)

PCA $\mathcal{A}$ running on $\mathcal{T}^5_{\text{lin}}$



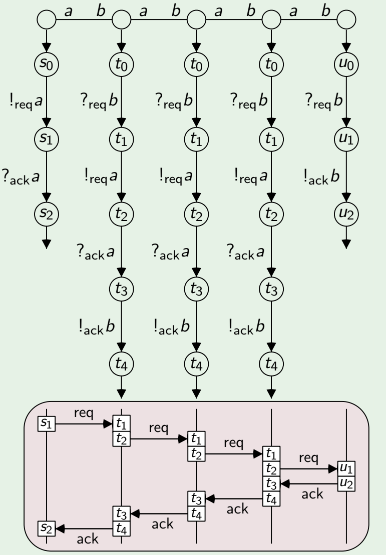
Parameterized communicating automata (PCA)

PCA $\mathcal{A}$ running on $\mathcal{T}_{\text{lin}}^5$



Parameterized communicating automata (PCA)

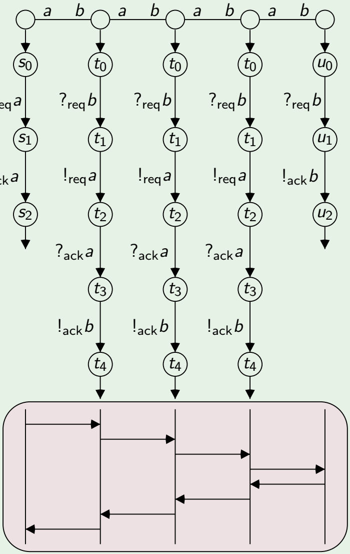
PCA $\mathcal{A}$ running on $\mathcal{T}^5_{\text{lin}}$



Parameterized communicating automata (PCA)

PCA $\mathcal{A}$ running on $\mathcal{T}_{\text{lin}}^5$

$M_{\text{lin}}^5 =$



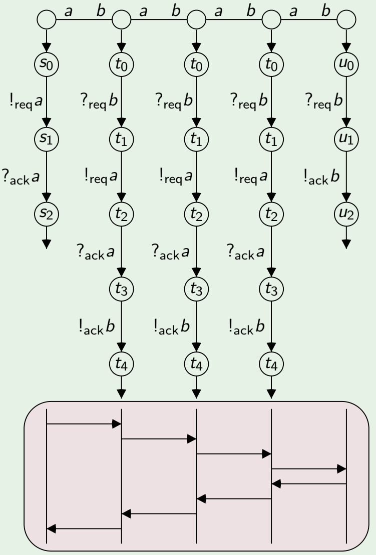
$\in L_{\mathcal{T}_{\text{lin}}^5}(\mathcal{A})$

Parameterized communicating automata (PCA)

Accepted language

$L_{\mathcal{T}^n_{\text{lin}}}(\mathcal{A}) = \{M^n_{\text{lin}}\}$
for all $n \geq 2$

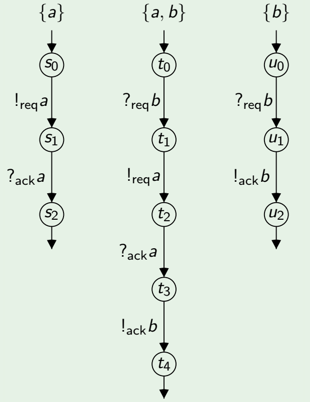
PCA $\mathcal{A}$ running on $\mathcal{T}^5_{\text{lin}}$



$M^5_{\text{lin}} =$ [diagram above] $\in L_{\mathcal{T}^5_{\text{lin}}}(\mathcal{A})$

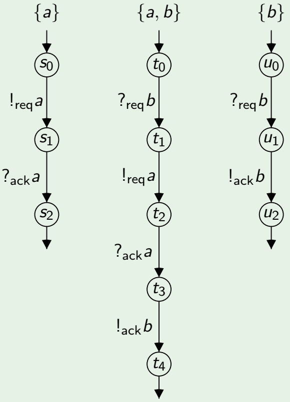
Parameterized communicating automata (PCA)

PCA $\mathcal{A}$ over $\{a, b\}$



Parameterized communicating automata (PCA)

PCA $\mathcal{A}$ over $\{a, b\}$



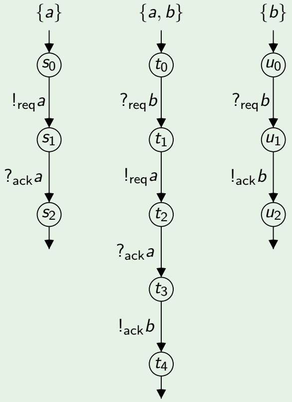
Definition

A PCA over $\mathcal{N}$ is a tuple (S, Msg, Δ, I, F):

- S finite set of states

Parameterized communicating automata (PCA)

PCA $\mathcal{A}$ over $\{a, b\}$



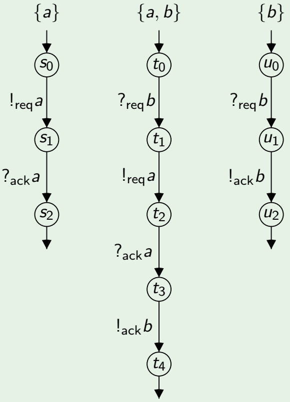
Definition

A PCA over $\mathcal{N}$ is a tuple (S, Msg, Δ, I, F):

- S finite set of states
- Msg finite set of messages

Parameterized communicating automata (PCA)

PCA $\mathcal{A}$ over $\{a, b\}$



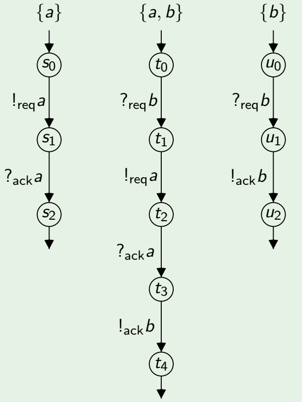
Definition

A PCA over $\mathcal{N}$ is a tuple (S, Msg, Δ, I, F):

- S finite set of states
- Msg finite set of messages
- $I : (2^{\mathcal{N}} \setminus \{\emptyset\}) \to 2^S$ initial states

Parameterized communicating automata (PCA)

PCA $\mathcal{A}$ over $\{a, b\}$



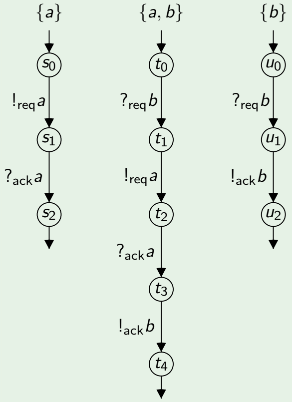
Definition

A PCA over $\mathcal{N}$ is a tuple (S, Msg, Δ, I, F):

- S finite set of states
- Msg finite set of messages
- $I : (2^{\mathcal{N}} \setminus \{\emptyset\}) \to 2^S$ initial states
- Δ the set of transitions

Parameterized communicating automata (PCA)

PCA $\mathcal{A}$ over $\{a, b\}$



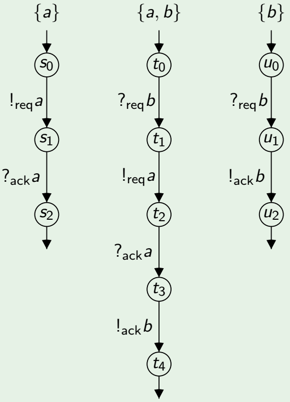
Definition

A PCA over $\mathcal{N}$ is a tuple (S, Msg, Δ, I, F):

- S finite set of states
- Msg finite set of messages
- $I : (2^{\mathcal{N}} \setminus \{\emptyset\}) \to 2^S$ initial states
- Δ the set of transitions
- F a boolean combination of statements $\langle \#(s) \geq k \rangle$ with $s \in S$ and $k \in \mathbb{N}$

Parameterized communicating automata (PCA)

PCA $\mathcal{A}$ over $\{a, b\}$



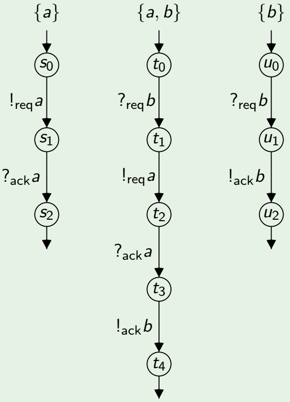
Definition

A PCA over $\mathcal{N}$ is a tuple (S, Msg, Δ, I, F):

- S finite set of states
- Msg finite set of messages
- $I : (2^{\mathcal{N}} \setminus \{\emptyset\}) \to 2^S$ initial states
- Δ the set of transitions
- F a boolean combination of statements $\langle \#(s) \geq k \rangle$ with $s \in S$ and $k \in \mathbb{N}$
  "s occurs at least k times as the terminal state of an active process"

Parameterized communicating automata (PCA)

PCA $\mathcal{A}$ over $\{a, b\}$



Definition

A PCA over $\mathcal{N}$ is a tuple (S, Msg, Δ, I, F):

- S finite set of states
- Msg finite set of messages
- $I : (2^{\mathcal{N}} \setminus \{\emptyset\}) \rightarrow 2^S$ initial states
- Δ the set of transitions
- F a boolean combination of statements $\langle \#(s) \geq k \rangle$ with $s \in S$ and $k \in \mathbb{N}$
  "s occurs at least k times as the terminal state of an active process"

$$F = \bigwedge_{s \in S \setminus \{s_2, t_4, u_2\}} \neg \langle \#(s) \geq 1 \rangle$$

Parameterized communicating automata (PCA)

✗ A PCA cannot say "the topology has at least 5 processes"

Parameterized communicating automata (PCA)

✗ A PCA cannot say "the topology has at least 5 processes"

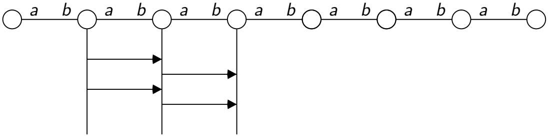
✔ A PCA can say "at least 5 processes of type $\{a, b\}$ are active"

Parameterized communicating automata (PCA)

✗ A PCA cannot say "the topology has at least 5 processes"

✓ A PCA can say "at least 5 processes of type $\{a, b\}$ are active"

✗ A PCA cannot distinguish between:

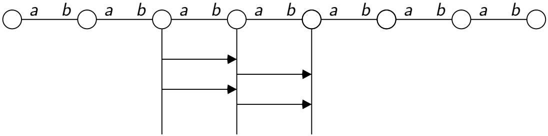


Parameterized communicating automata (PCA)

✗ A PCA cannot say "the topology has at least 5 processes"

✔ A PCA can say "at least 5 processes of type $\{a, b\}$ are active"

✗ A PCA cannot distinguish between:

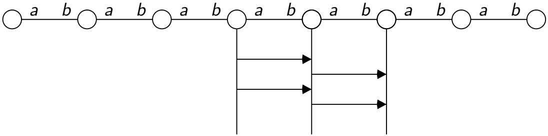


Parameterized communicating automata (PCA)

✗ A PCA cannot say "the topology has at least 5 processes"

✓ A PCA can say "at least 5 processes of type $\{a, b\}$ are active"

✗ A PCA cannot distinguish between:

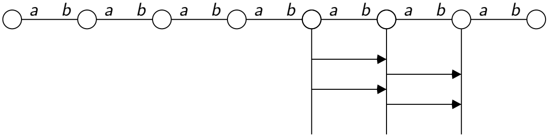


Parameterized communicating automata (PCA)

✗ A PCA cannot say "the topology has at least 5 processes"

✓ A PCA can say "at least 5 processes of type $\{a, b\}$ are active"

✗ A PCA cannot distinguish between:



MSO Logic

MSO logic

MSO logic

$$\begin{aligned}\varphi ::= \quad & !a(x) \mid ?a(x) \mid a \in \mathit{type}(x) \mid \\ & x \lhd_{\mathrm{proc}} y \mid x \lhd_{\mathrm{proc}}^{*} y \mid x \lhd_{\mathrm{msg}} y \mid x \lhd^{*} y \mid x \sim y \mid \\ & x = y \mid x \in X \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x \varphi \mid \exists X \varphi\end{aligned}$$

where $a \in \mathcal{N}$

MSO logic

MSO logic

$$\varphi ::= \; !a(x) \mid ?a(x) \mid a \in type(x) \mid$$

$$x \lhd_{\text{proc}} y \mid x \lhd_{\text{proc}}^{*} y \mid x \lhd_{\text{msg}} y \mid x \lhd^{*} y \mid x \sim y \mid$$

$$x = y \mid x \in X \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x \varphi \mid \exists X \varphi$$

where $a \in \mathcal{N}$

$x \sim y$ says that x and y are located on the same process.

MSO logic

MSO logic

$$\varphi ::= \ !a(x) \mid ?a(x) \mid a \in type(x) \mid$$
$$x \lhd_{\text{proc}} y \mid x \lhd^*_{\text{proc}} y \mid x \lhd_{\text{msg}} y \mid x \lhd^* y \mid x \sim y \mid$$
$$x = y \mid x \in X \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x\varphi \mid \exists X\varphi$$

where $a \in \mathcal{N}$

$x \sim y$ says that x and y are located on the same process.

Define fragments:

- FO: first-order logic, without $\exists X\varphi$

MSO logic

MSO logic

$$\varphi ::= \; !a(x) \mid ?a(x) \mid a \in type(x) \mid$$
$$x \lhd_{\text{proc}} y \mid x \lhd^*_{\text{proc}} y \mid x \lhd_{\text{msg}} y \mid x \lhd^* y \mid x \sim y \mid$$
$$x = y \mid x \in X \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x \varphi \mid \exists X \varphi$$

where $a \in \mathcal{N}$

$x \sim y$ says that x and y are located on the same process.

Define fragments:

- FO: first-order logic, without $\exists X \varphi$
- EMSO: formulas of the form $\exists X_1 \ldots \exists X_n \varphi$ with $\varphi \in$ FO

MSO logic

MSO logic

$$\varphi ::= \; !a(x) \mid ?a(x) \mid a \in type(x) \mid$$

$$\sigma \subseteq \quad x \lhd_{\text{proc}} y \mid x \lhd^*_{\text{proc}} y \mid x \lhd_{\text{msg}} y \mid x \lhd^* y \mid x \sim y \mid$$

$$x = y \mid x \in X \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x \varphi \mid \exists X \varphi$$

where $a \in \mathcal{N}$

$x \sim y$ says that x and y are located on the same process.

Define fragments:

- FO: first-order logic, without $\exists X \varphi$
- EMSO: formulas of the form $\exists X_1 \ldots \exists X_n \varphi$ with $\varphi \in$ FO
- FO[σ] and EMSO[σ] (e.g., FO[$\lhd_{\text{proc}}, \lhd_{\text{msg}}$])

MSO logic

MSO logic

$$\varphi ::= \ !a(x) \mid ?a(x) \mid a \in type(x) \mid$$

$$\sigma \subseteq \quad x \lhd_{\text{proc}} y \mid x \lhd_{\text{proc}}^{*} y \mid x \lhd_{\text{msg}} y \mid x \lhd^{*} y \mid x \sim y \mid$$

$$x = y \mid x \in X \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x \varphi \mid \exists X \varphi$$

where $a \in \mathcal{N}$

$x \sim y$ says that x and y are located on the same process.

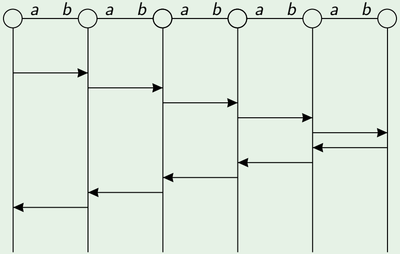
Define fragments:

- FO: first-order logic, without $\exists X \varphi$
- EMSO: formulas of the form $\exists X_1 \ldots \exists X_n \varphi$ with $\varphi \in$ FO
- FO[σ] and EMSO[σ] (e.g., FO[$\lhd_{\text{proc}}, \lhd_{\text{msg}}$])

Let $L_{\mathcal{T}}(\varphi)$ be the set of MSCs over $\mathcal{T}$ that are a model of φ.

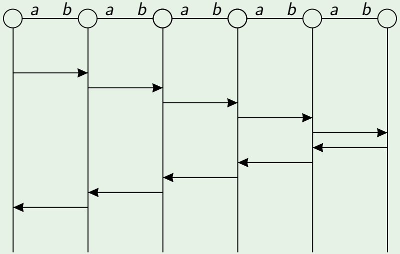
MSO logic

MSC M^6_{lin}



MSO logic

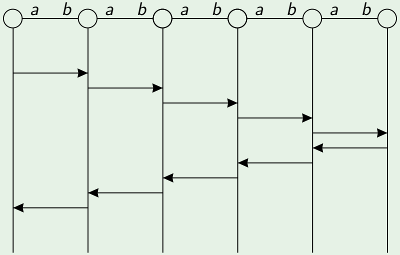
MSC M^6_{lin}



- $M^6_{\text{lin}} \models \forall x(?b(x) \;\rightarrow\; \exists y(x \lhd^*_{\text{proc}} y \wedge !b(y)))$

MSO logic

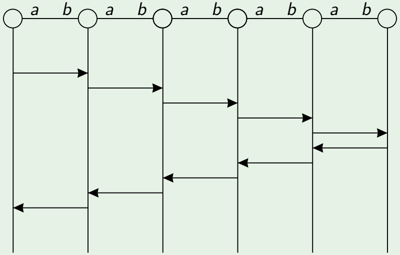
MSC M^6_{lin}



- $M^6_{\text{lin}} \models \forall x(?b(x) \;\rightarrow\; \exists y(x \lhd^*_{\text{proc}} y \wedge !b(y))) \in \text{FO}[\lhd^*_{\text{proc}}]$

MSO logic

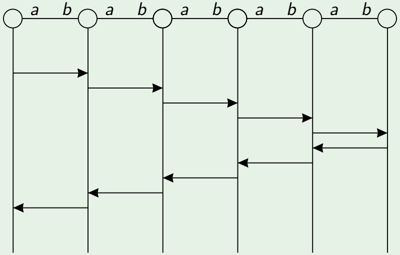
MSC M^6_{lin}



- $M^6_{\text{lin}} \models \forall x(?b(x) \ \rightarrow \ \exists y(x \lhd^*_{\text{proc}} y \wedge !b(y))) \in \text{FO}[\lhd^*_{\text{proc}}]$
- $M^6_{\text{lin}} \models \forall x \forall y(x \lhd^* y \vee y \lhd^* x)$

MSO logic

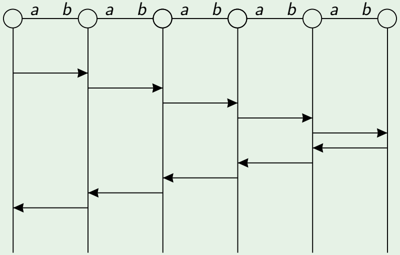
MSC M^6_{lin}



- $M^6_{\text{lin}} \models \forall x(?b(x) \;\rightarrow\; \exists y(x \lhd^*_{\text{proc}} y \wedge !b(y))) \in \text{FO}[\lhd^*_{\text{proc}}]$
- $M^6_{\text{lin}} \models \forall x \forall y(x \lhd^* y \vee y \lhd^* x)$
- $M^6_{\text{lin}} \not\models \exists x \exists y(b \notin \mathit{type}(x) \wedge a \notin \mathit{type}(y) \wedge x \lhd_{\text{msg}} y)$

MSO logic

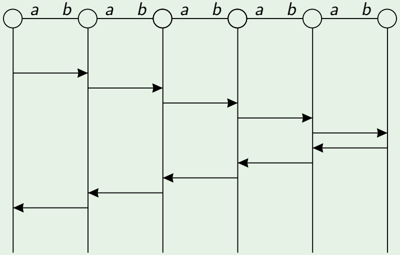
MSC M^6_{lin}



- $M^6_{\text{lin}} \models \forall x(?b(x) \ \rightarrow \ \exists y(x \lhd^*_{\text{proc}} y \wedge !b(y))) \in \text{FO}[\lhd^*_{\text{proc}}]$
- $M^6_{\text{lin}} \models \forall x \forall y(x \lhd^* y \vee y \lhd^* x)$
- $M^6_{\text{lin}} \not\models \exists x \exists y(b \notin \mathit{type}(x) \wedge a \notin \mathit{type}(y) \wedge x \lhd_{\text{msg}} y) =: \varphi$

MSO logic

MSC M^6_{lin}



- $M^6_{\text{lin}} \models \forall x(?b(x) \;\rightarrow\; \exists y(x \lhd^*_{\text{proc}} y \wedge !b(y))) \in \text{FO}[\lhd^*_{\text{proc}}]$
- $M^6_{\text{lin}} \models \forall x \forall y(x \lhd^* y \vee y \lhd^* x)$
- $M^6_{\text{lin}} \not\models \exists x \exists y(b \notin \mathit{type}(x) \wedge a \notin \mathit{type}(y) \wedge x \lhd_{\text{msg}} y) =: \varphi$
- $M^n_{\text{lin}} \models \varphi$ iff $n = 2$

Positive results

Theorem

For every PCA $\mathcal{A}$, there is a formula $\varphi \in \mathsf{EMSO}[\lhd_{\mathsf{proc}}, \lhd_{\mathsf{msg}}]$ that is equivalent to $\mathcal{A}$ on all topologies.

Positive results

Theorem

For every PCA $\mathcal{A}$, there is a formula $\varphi \in \mathrm{EMSO}[\lhd_{\mathsf{proc}}, \lhd_{\mathsf{msg}}]$ that is equivalent to $\mathcal{A}$ on all topologies.

Proof

Standard. $\square$

Negative Results

Restrictions on topologies are necessary

Theorem

There exists a sentence $\varphi \in \mathsf{FO}[\lhd_{\mathsf{proc}}, \lhd_{\mathsf{msg}}]$ over $\{a, b\}$ such that, for all PCA $\mathcal{A}$, there is a ring forest $\mathcal{T}$ with $L_{\mathcal{T}}(\mathcal{A}) \neq L_{\mathcal{T}}(\varphi)$.

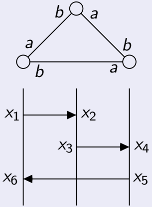
Restrictions on topologies are necessary

Theorem

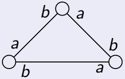
There exists a sentence $\varphi \in \mathrm{FO}[\lhd_{\mathrm{proc}}, \lhd_{\mathrm{msg}}]$ over $\{a, b\}$ such that, for all PCA $\mathcal{A}$, there is a ring forest $\mathcal{T}$ with $L_{\mathcal{T}}(\mathcal{A}) \neq L_{\mathcal{T}}(\varphi)$.

Proof

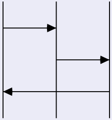
$$\varphi = \forall x \exists x_1, \ldots, x_6 (x \in \{x_1, \ldots, x_6\} \wedge \mathit{cycle}(x_1, \ldots, x_6))$$



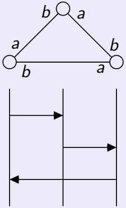
Restrictions on topologies are necessary



$M =$

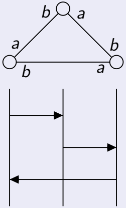


Restrictions on topologies are necessary

$M =$ 

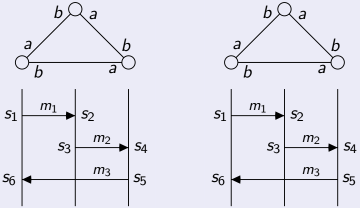
- Suppose there is $\mathcal{A}$ such that $L_{\mathcal{T}}(\mathcal{A}) = L_{\mathcal{T}}(\varphi)$ for all ring forests $\mathcal{T}$.

Restrictions on topologies are necessary

$M =$ 

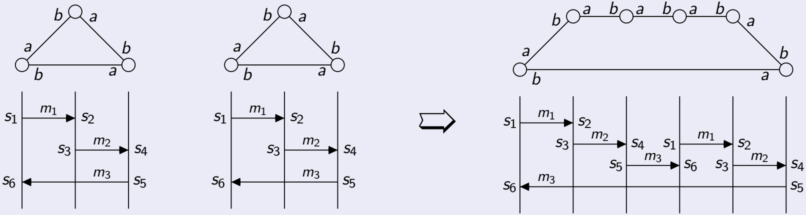
- Suppose there is $\mathcal{A}$ such that $L_{\mathcal{T}}(\mathcal{A}) = L_{\mathcal{T}}(\varphi)$ for all ring forests $\mathcal{T}$.
- We have $\underbrace{M \uplus \ldots \uplus M}_{n} \in L_{\mathcal{T}^3_{\text{ring}} \uplus \ldots \uplus \mathcal{T}^3_{\text{ring}}}(\mathcal{A})$ for all $n \geq 1$.

Restrictions on topologies are necessary

$M =$ 

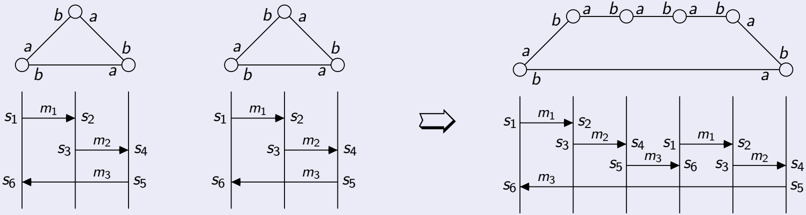
- Suppose there is $\mathcal{A}$ such that $L_{\mathcal{T}}(\mathcal{A}) = L_{\mathcal{T}}(\varphi)$ for all ring forests $\mathcal{T}$.
- We have $\underbrace{M \uplus \ldots \uplus M}_{n} \in L_{\mathcal{T}^3_{\text{ring}} \uplus \ldots \uplus \mathcal{T}^3_{\text{ring}}}(\mathcal{A})$ for all $n \geq 1$.
- For large enough n, there is a run of $\mathcal{A}$ that behaves the same on two copies of M.

Restrictions on topologies are necessary

$M =$ 

- Suppose there is $\mathcal{A}$ such that $L_{\mathcal{T}}(\mathcal{A}) = L_{\mathcal{T}}(\varphi)$ for all ring forests $\mathcal{T}$.
- We have $\underbrace{M \uplus \ldots \uplus M}_{n} \in L_{\mathcal{T}^3_{\text{ring}} \uplus \ldots \uplus \mathcal{T}^3_{\text{ring}}}(\mathcal{A})$ for all $n \geq 1$.
- For large enough n, there is a run of $\mathcal{A}$ that behaves the same on two copies of M.
- Replace the two copies with MSC over $\mathcal{T}^6_{\text{ring}}$. Contradiction. $\square$

Restrictions on topologies are necessary

$M =$ 

- Suppose there is $\mathcal{A}$ such that $L_{\mathcal{T}}(\mathcal{A}) = L_{\mathcal{T}}(\varphi)$ for all ring forests $\mathcal{T}$.
- We have $\underbrace{M \uplus \ldots \uplus M}_{n} \in L_{\mathcal{T}^3_{\text{ring}} \uplus \ldots \uplus \mathcal{T}^3_{\text{ring}}}(\mathcal{A})$ for all $n \geq 1$.
- For large enough n, there is a run of $\mathcal{A}$ that behaves the same on two copies of M.
- Replace the two copies with MSC over $\mathcal{T}^6_{\text{ring}}$. Contradiction. $\square$

Lesson learned

PCA have limited ability to "detect" cycles.

Restrictions on logic are necessary

Theorem

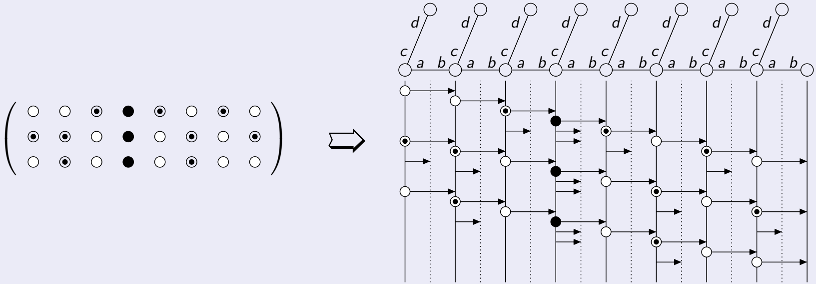
There exists a sentence $\varphi \in \mathrm{FO}[\lhd^*_{\mathrm{proc}}, \lhd_{\mathrm{msg}}, \lhd^*]$ over $\{a, b, c, d\}$ such that, for all PCA $\mathcal{A}$, there is a tree $\mathcal{T}$ with $L_{\mathcal{T}}(\mathcal{A}) \neq L_{\mathcal{T}}(\varphi)$.

Restrictions on logic are necessary

Theorem

There exists a sentence $\varphi \in \mathrm{FO}[\lhd^*_{\mathsf{proc}}, \lhd_{\mathsf{msg}}, \lhd^*]$ over $\{a, b, c, d\}$ such that, for all PCA $\mathcal{A}$, there is a tree $\mathcal{T}$ with $L_{\mathcal{T}}(\mathcal{A}) \neq L_{\mathcal{T}}(\varphi)$.

Proof (idea goes back to [Thomas 1996])

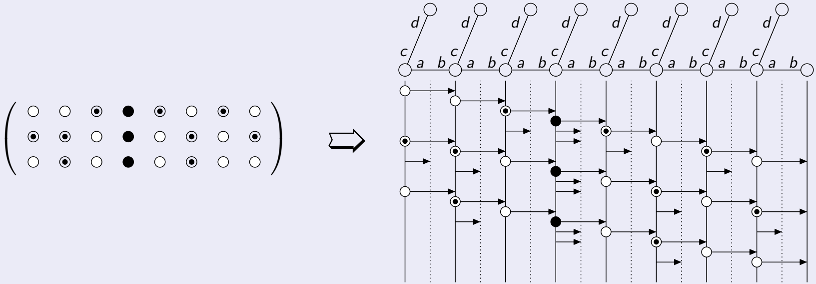


Restrictions on logic are necessary

Theorem

There exists a sentence $\varphi \in \mathrm{FO}[\lhd^*_{\mathrm{proc}}, \lhd_{\mathrm{msg}}, \lhd^*]$ over $\{a, b, c, d\}$ such that, for all PCA $\mathcal{A}$, there is a tree $\mathcal{T}$ with $L_{\mathcal{T}}(\mathcal{A}) \neq L_{\mathcal{T}}(\varphi)$.

Proof (idea goes back to [Thomas 1996])



Lesson learned

Look at more "local" logics.

Positive Results

Locality of FO logic

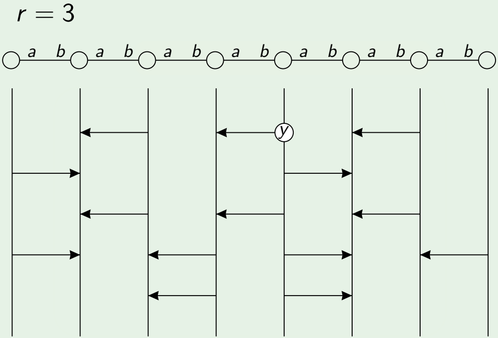
Theorem [Schwentick-Barthelmann 1999]

Every formula $\varphi \in \mathsf{FO}[\sigma]$ is equivalent to a formula of the form $\exists x_1 \ldots \exists x_n \forall y \psi \in \mathsf{FO}[\sigma]$ where ψ is *r*-local around y, for some $r \geq 1$ (quantification is restricted to elements of distance $\leq r$ from y).

Locality of FO logic

Theorem [Schwentick-Barthelmann 1999]

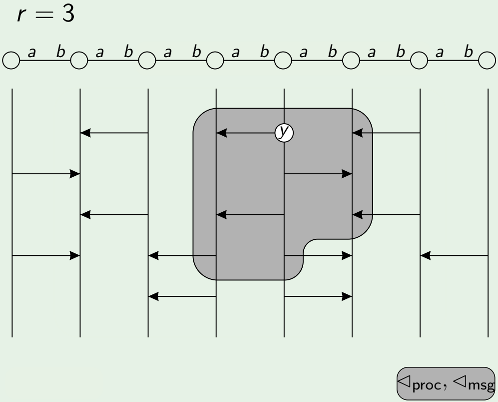
Every formula $\varphi \in \mathrm{FO}[\sigma]$ is equivalent to a formula of the form $\exists x_1 \ldots \exists x_n \forall y \psi \in \mathrm{FO}[\sigma]$ where ψ is r-local around y, for some $r \geq 1$ (quantification is restricted to elements of distance $\leq r$ from y).



Locality of FO logic

Theorem [Schwentick-Barthelmann 1999]

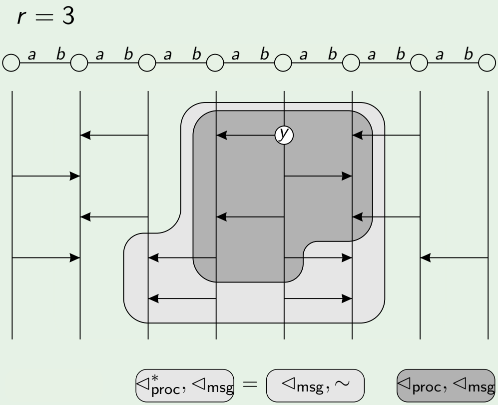
Every formula $\varphi \in \mathrm{FO}[\sigma]$ is equivalent to a formula of the form $\exists x_1 \ldots \exists x_n \forall y \psi \in \mathrm{FO}[\sigma]$ where ψ is r-local around y, for some $r \geq 1$ (quantification is restricted to elements of distance $\leq r$ from y).



Locality of FO logic

Theorem [Schwentick-Barthelmann 1999]

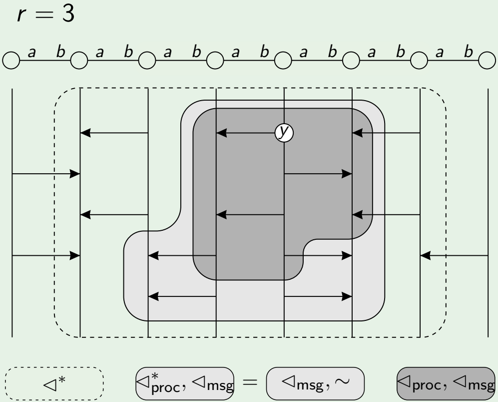
Every formula $\varphi \in \mathrm{FO}[\sigma]$ is equivalent to a formula of the form $\exists x_1 \ldots \exists x_n \forall y \psi \in \mathrm{FO}[\sigma]$ where ψ is r-local around y, for some $r \geq 1$ (quantification is restricted to elements of distance $\leq r$ from y).



Locality of FO logic

Theorem [Schwentick-Barthelmann 1999]

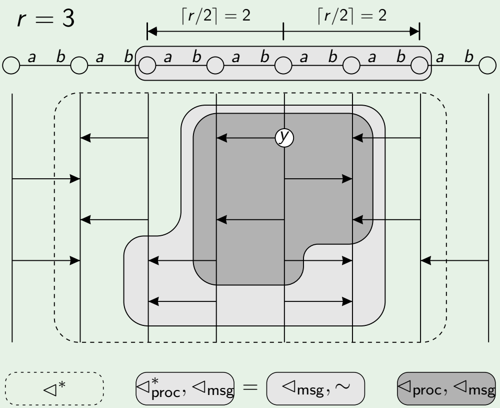
Every formula $\varphi \in \mathrm{FO}[\sigma]$ is equivalent to a formula of the form $\exists x_1 \ldots \exists x_n \forall y \psi \in \mathrm{FO}[\sigma]$ where ψ is r-local around y, for some $r \geq 1$ (quantification is restricted to elements of distance $\leq r$ from y).



Locality of FO logic

Theorem [Schwentick-Barthelmann 1999]

Every formula $\varphi \in \mathrm{FO}[\sigma]$ is equivalent to a formula of the form $\exists x_1 \ldots \exists x_n \forall y \psi \in \mathrm{FO}[\sigma]$ where ψ is r-local around y, for some $r \geq 1$ (quantification is restricted to elements of distance $\leq r$ from y).



Positive results

Theorem

Let $\varphi \in \mathsf{EMSO}[\lhd^*_{\mathsf{proc}}, \lhd_{\mathsf{msg}}]$, $B \geq 1$, and $\mathfrak{T}$ be any of the following:

Positive results

Theorem

Let $\varphi \in \mathrm{EMSO}[\lhd^*_{\mathsf{proc}}, \lhd_{\mathsf{msg}}]$, $B \geq 1$, and $\mathfrak{T}$ be any of the following:

- the set of pipeline topologies,
- the set of grid topologies,
- the set of tree topologies,
- the set of ring topologies.

Positive results

Theorem

Let $\varphi \in \mathrm{EMSO}[\lhd^*_{\mathsf{proc}}, \lhd_{\mathsf{msg}}]$, $B \geq 1$, and $\mathfrak{T}$ be any of the following:

- the set of pipeline topologies,
- the set of grid topologies,
- the set of tree topologies,
- the set of ring topologies.

There is a PCA $\mathcal{A}$ such that, for all $\mathcal{T} \in \mathfrak{T}$, we have $L^B_{\mathcal{T}}(\mathcal{A}) = L^B_{\mathcal{T}}(\varphi)$.

Positive results

Theorem

Let $\varphi \in \mathrm{EMSO}[\lhd^*_{\mathsf{proc}}, \lhd_{\mathsf{msg}}]$, $B \geq 1$, and $\mathfrak{T}$ be any of the following:

- the set of pipeline topologies,
- the set of grid topologies,
- the set of tree topologies,
- the set of ring topologies.

There is a PCA $\mathcal{A}$ such that, for all $\mathcal{T} \in \mathfrak{T}$, we have $L^B_{\mathcal{T}}(\mathcal{A}) = L^B_{\mathcal{T}}(\varphi)$.

Here, $L^B_{\mathcal{T}}(\mathcal{A})$ is the restriction of $L_{\mathcal{T}}(\mathcal{A})$ to B-bounded MSCs.

Positive results

Corollary

Let $\varphi \in \mathrm{EMSO}[\lhd^*_{\mathsf{proc}}, \lhd_{\mathsf{msg}}]$, $B \geq 1$, and $\mathfrak{T}$ be any of the following:

- the set of pipeline topologies,
- the set of grid topologies,
- the set of tree topologies,
- the set of ring topologies.

There is a PCA $\mathcal{A}$ such that, for all $\mathcal{T} \in \mathfrak{T}$, we have $L^B_{\mathcal{T}}(\mathcal{A}) = L^B_{\mathcal{T}}(\varphi)$.

Here, $L^B_{\mathcal{T}}(\mathcal{A})$ is the restriction of $L_{\mathcal{T}}(\mathcal{A})$ to B-bounded MSCs.

Positive results

Corollary

Let $\varphi \in \text{EMSO}[\lhd^*_{\text{proc}}, \lhd_{\text{msg}}]$, $B \geq 1$, and $\mathfrak{T}$ be any of the following:

- the set of pipeline topologies,
- the set of grid topologies,
- the set of tree topologies,
- the set of ring topologies.

There is a PCA $\mathcal{A}$ such that, for all $\mathcal{T} \in \mathfrak{T}$, we have $L^B_{\mathcal{T}}(\mathcal{A}) = L^B_{\mathcal{T}}(\varphi)$.

Here, $L^B_{\mathcal{T}}(\mathcal{A})$ is the restriction of $L_{\mathcal{T}}(\mathcal{A})$ to B-bounded MSCs.

Theorem

Let $\varphi \in \text{EMSO}[\lhd^*_{\text{proc}}, \lhd_{\text{msg}}]$, $B \geq 1$, and $\mathfrak{T}$ be a $(r_\varphi + 2)$-unambiguous set of topologies. There is a PCA $\mathcal{A}$ such that, for all $\mathcal{T} \in \mathfrak{T}$, we have $L^B_{\mathcal{T}}(\mathcal{A}) = L^B_{\mathcal{T}}(\varphi)$.

Here, r_φ is the radius associated with the first-order kernel of φ.

Unambiguous topology classes

Definition

Let $k \in \mathbb{N}$. A class $\mathfrak{T}$ of topologies is k-unambiguous if, for all $w \in (\mathcal{N} \times \mathcal{N})^*$ with $|w| \leq k$, all $(P, \longmapsto), (P', \longmapsto') \in \mathfrak{T}$, and all processes $p, q \in P$ and $p', q' \in P'$ such that $p \overset{w}{\longmapsto} q$ and $p' \overset{w}{\longmapsto}' q'$, we have $p = q$ iff $p' = q'$.

Unambiguous topology classes

Definition

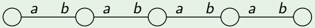
Let $k \in \mathbb{N}$. A class $\mathfrak{T}$ of topologies is *k-unambiguous* if, for all $w \in (\mathcal{N} \times \mathcal{N})^*$ with $|w| \leq k$, all $(P, \longmapsto), (P', \longmapsto') \in \mathfrak{T}$, and all processes $p, q \in P$ and $p', q' \in P'$ such that $p \overset{w}{\longmapsto} q$ and $p' \overset{w}{\longmapsto}' q'$, we have $p = q$ iff $p' = q'$.

In other words:

If w forms a cycle in a topology from $\mathfrak{T}$, then it forms a cycle anywhere, in any topology of $\mathfrak{T}$ (if it is applicable).

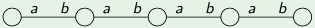
Unambiguous topology classes

Pipelines ✔ k-unambiguous for all $k \in \mathbb{N}$

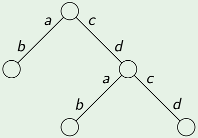


Unambiguous topology classes

Pipelines ✔ k-unambiguous for all $k \in \mathbb{N}$

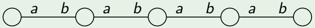


Trees ✔ k-unambiguous for all $k \in \mathbb{N}$

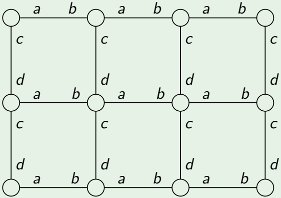


Unambiguous topology classes

Pipelines ✔ k-unambiguous for all $k \in \mathbb{N}$

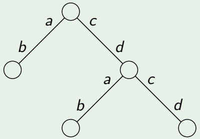


Grids ✔ k-unambiguous for all $k \in \mathbb{N}$



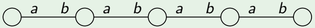
$w = (a, b)(c, d)(b, a)(d, c)$

Trees ✔ k-unambiguous for all $k \in \mathbb{N}$

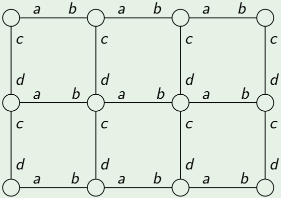


Unambiguous topology classes

Pipelines ✔ k-unambiguous for all $k \in \mathbb{N}$

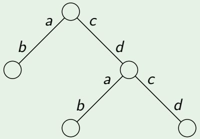


Grids ✔ k-unambiguous for all $k \in \mathbb{N}$

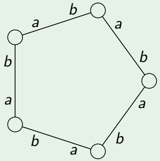


$w = (a, b)(c, d)(b, a)(d, c)$

Trees ✔ k-unambiguous for all $k \in \mathbb{N}$

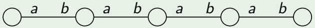


Rings ✗ not k-unambiguous for all $k \geq 3$

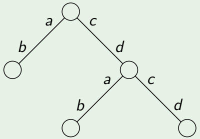


Unambiguous topology classes

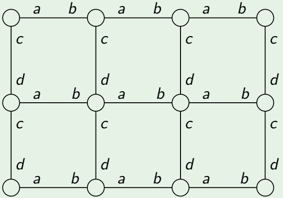
Pipelines ✔ k-unambiguous for all $k \in \mathbb{N}$



Trees ✔ k-unambiguous for all $k \in \mathbb{N}$

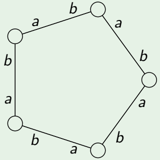


Grids ✔ k-unambiguous for all $k \in \mathbb{N}$



$w = (a, b)(c, d)(b, a)(d, c)$

Rings ✗ not k-unambiguous for all $k \geq 3$

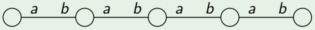


But:

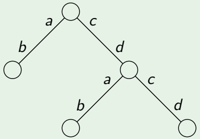
- The class of rings of size $\geq k + 1$ is k-unambiguous, for all $k \in \mathbb{N}$.

Unambiguous topology classes

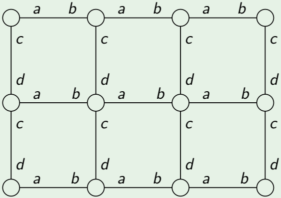
Pipelines ✔ k-unambiguous for all $k \in \mathbb{N}$



Trees ✔ k-unambiguous for all $k \in \mathbb{N}$

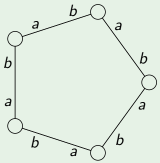


Grids ✔ k-unambiguous for all $k \in \mathbb{N}$



$w = (a, b)(c, d)(b, a)(d, c)$

Rings ✗ not k-unambiguous for all $k \geq 3$



But:

- The class of rings of size $\geq k + 1$ is k-unambiguous, for all $k \in \mathbb{N}$.
- Every single ring is k-unambiguous, for all $k \in \mathbb{N}$.

Unambiguous topology classes

Theorem

Let $\varphi \in \mathrm{EMSO}[\lhd^*_{\mathsf{proc}}, \lhd_{\mathsf{msg}}]$, $B \geq 1$, and $\mathfrak{T}$ be a $(r_\varphi + 2)$-unambiguous set of topologies. There is a PCA $\mathcal{A}$ such that, for all $\mathcal{T} \in \mathfrak{T}$, we have $L^B_{\mathcal{T}}(\mathcal{A}) = L^B_{\mathcal{T}}(\varphi)$.

Unambiguous topology classes

Theorem

Let $\varphi \in \mathrm{EMSO}[\lhd^*_{\mathrm{proc}}, \lhd_{\mathrm{msg}}]$, $B \geq 1$, and $\mathfrak{T}$ be a $(r_\varphi + 2)$-unambiguous set of topologies. There is a PCA $\mathcal{A}$ such that, for all $\mathcal{T} \in \mathfrak{T}$, we have $L^B_{\mathcal{T}}(\mathcal{A}) = L^B_{\mathcal{T}}(\varphi)$.

Proof

- Translate φ into normal form

$$\exists X_1 \ldots \exists X_m \exists x_1 \ldots \exists x_n \forall y \psi$$

where ψ is r_φ-local around y.

Unambiguous topology classes

Theorem

Let $\varphi \in \mathrm{EMSO}[\lhd^*_{\mathrm{proc}}, \lhd_{\mathrm{msg}}]$, $B \geq 1$, and $\mathfrak{T}$ be a $(r_\varphi + 2)$-unambiguous set of topologies. There is a PCA $\mathcal{A}$ such that, for all $\mathcal{T} \in \mathfrak{T}$, we have $L^B_{\mathcal{T}}(\mathcal{A}) = L^B_{\mathcal{T}}(\varphi)$.

Proof

- Translate φ into normal form

$$\exists X_1 \ldots \exists X_m \exists x_1 \ldots \exists x_n \forall y \psi$$

  where ψ is r_φ-local around y.
- Existential quantification $\Rightarrow$ projection & guessing of truth values for propositions involving only free variables of $\forall y \psi$ [Gastin-Kuske 2010].

Unambiguous topology classes

Theorem

Let $\varphi \in \text{EMSO}[\lhd^*_{\text{proc}}, \lhd_{\text{msg}}]$, $B \geq 1$, and $\mathfrak{T}$ be a $(r_\varphi + 2)$-unambiguous set of topologies. There is a PCA $\mathcal{A}$ such that, for all $\mathcal{T} \in \mathfrak{T}$, we have $L^B_{\mathcal{T}}(\mathcal{A}) = L^B_{\mathcal{T}}(\varphi)$.

Proof

- Translate φ into normal form

  $$\exists X_1 \ldots \exists X_m \exists x_1 \ldots \exists x_n \forall y \psi$$

  where ψ is r_φ-local around y.
- Existential quantification $\Rightarrow$ projection & guessing of truth values for propositions involving only free variables of $\forall y \psi$ [Gastin-Kuske 2010].
- Construct fixed-topology CA $\mathcal{A}_\theta$ for formula $\forall y \psi$ over all topology neighborhoods θ of radius $\lceil r_\varphi / 2 \rceil$ [Genest-Kuske-Muscholl 2004].

Unambiguous topology classes

Theorem

Let $\varphi \in \mathrm{EMSO}[\lhd^*_{\mathrm{proc}}, \lhd_{\mathrm{msg}}]$, $B \geq 1$, and $\mathfrak{T}$ be a $(r_\varphi + 2)$-unambiguous set of topologies. There is a PCA $\mathcal{A}$ such that, for all $\mathcal{T} \in \mathfrak{T}$, we have $L^B_{\mathcal{T}}(\mathcal{A}) = L^B_{\mathcal{T}}(\varphi)$.

Proof

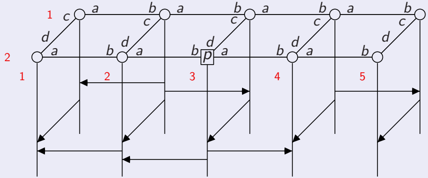
- Translate φ into normal form

$$\exists X_1 \ldots \exists X_m \exists x_1 \ldots \exists x_n \forall y \psi$$

  where ψ is r_φ-local around y.
- Existential quantification $\Rightarrow$ projection & guessing of truth values for propositions involving only free variables of $\forall y \psi$ [Gastin-Kuske 2010].
- Construct fixed-topology CA $\mathcal{A}_\theta$ for formula $\forall y \psi$ over all topology neighborhoods θ of radius $\lceil r_\varphi/2 \rceil$ [Genest-Kuske-Muscholl 2004].
- Glue fixed-topology CA together to obtain a PCA for $\forall y \psi$.

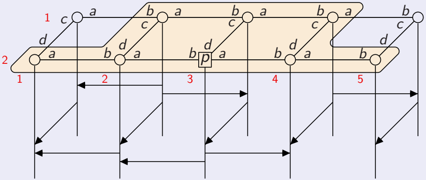
Unambiguous topology classes

Proof (cntd.) suppose $r_\varphi = 3$ so that $\lceil r_\varphi/2 \rceil = 2$



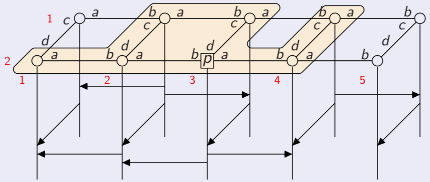
Unambiguous topology classes

Proof (cntd.) suppose $r_\varphi = 3$ so that $\lceil r_\varphi/2 \rceil = 2$



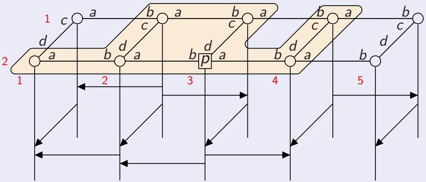
Unambiguous topology classes

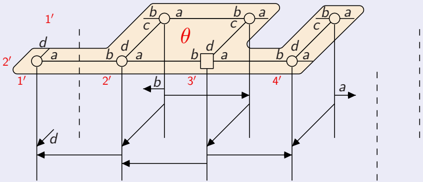
Proof (cntd.) suppose $r_\varphi = 3$ so that $\lceil r_\varphi/2 \rceil = 2$



Unambiguous topology classes

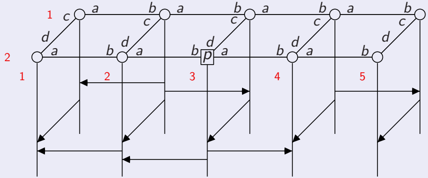
Proof (cntd.) suppose $r_\varphi = 3$ so that $\lceil r_\varphi/2 \rceil = 2$

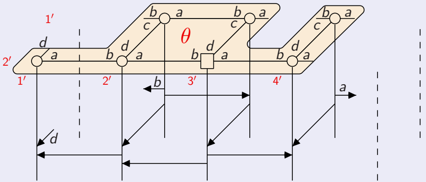




Unambiguous topology classes

Proof (cntd.) suppose $r_\varphi = 3$ so that $\lceil r_\varphi/2 \rceil = 2$

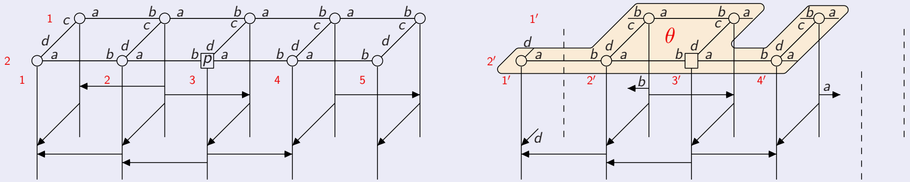




- Process $p = (2, 3)$ guesses topology neighborhood θ and simulates local automaton $\mathcal{A}_\theta[2', 3']$.

Unambiguous topology classes

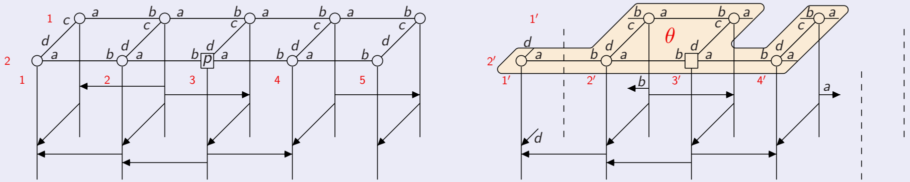
Proof (cntd.) suppose $r_\varphi = 3$ so that $\lceil r_\varphi/2 \rceil = 2$



- Process $p = (2,3)$ guesses topology neighborhood θ and simulates local automaton $\mathcal{A}_\theta[2', 3']$.
- Process p sends $(\theta, (2', 3'))$ to $(2, 2)$.

Unambiguous topology classes

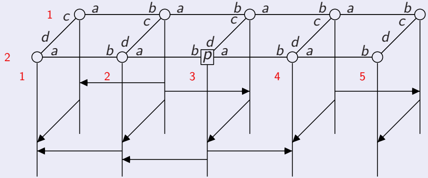
Proof (cntd.) suppose $r_\varphi = 3$ so that $\lceil r_\varphi/2 \rceil = 2$

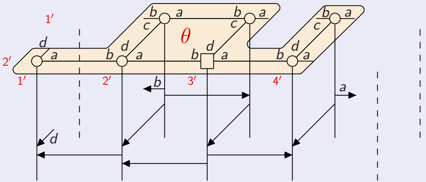


- Process $p = (2,3)$ guesses topology neighborhood θ and simulates local automaton $\mathcal{A}_\theta[2',3']$.
- Process p sends $(\theta, (2',3'))$ to $(2,2)$.
- Process $(2,2)$ receives $(\theta, (1',2'))$ from $(1,2)$ and simulates $\mathcal{A}_\theta[2',2']$.

Unambiguous topology classes

Proof (cntd.) suppose $r_\varphi = 3$ so that $\lceil r_\varphi/2 \rceil = 2$

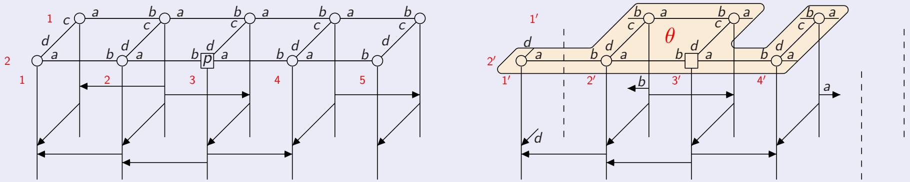




- Process $p = (2,3)$ guesses topology neighborhood θ and simulates local automaton $\mathcal{A}_\theta[2', 3']$.
- Process p sends $(\theta, (2', 3'))$ to $(2, 2)$.
- Process $(2,2)$ receives $(\theta, (1', 2'))$ from $(1,2)$ and simulates $\mathcal{A}_\theta[2', 2']$.
- ⇨ Topology admits $w = (b,a)(d,c)(a,b)(c,d)$-path from p.

Unambiguous topology classes

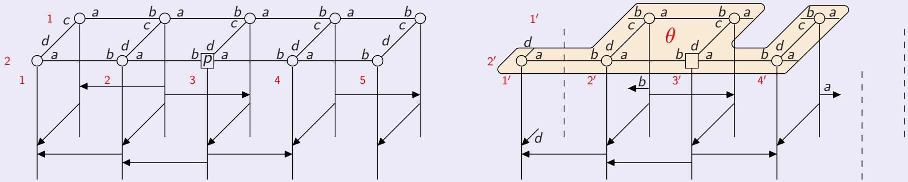
Proof (cntd.) suppose $r_\varphi = 3$ so that $\lceil r_\varphi/2 \rceil = 2$



- Process $p = (2,3)$ guesses topology neighborhood θ and simulates local automaton $\mathcal{A}_\theta[2',3']$.
- Process p sends $(\theta, (2',3'))$ to $(2,2)$.
- Process $(2,2)$ receives $(\theta, (1',2'))$ from $(1,2)$ and simulates $\mathcal{A}_\theta[2',2']$.
- ⇨ Topology admits $w = (b,a)(d,c)(a,b)(c,d)$-path from p.
- Since $|w| = 4 \leq r_\varphi + 2$, and $\mathfrak{T}$ is $(r_\varphi + 2)$-unambiguous, w forms a cycle in the topology as well.

Unambiguous topology classes

Proof (cntd.) suppose $r_\varphi = 3$ so that $\lceil r_\varphi / 2 \rceil = 2$



- Process $p = (2,3)$ guesses topology neighborhood θ and simulates local automaton $\mathcal{A}_\theta[2', 3']$.
- Process p sends $(\theta, (2', 3'))$ to $(2,2)$.
- Process $(2,2)$ receives $(\theta, (1', 2'))$ from $(1,2)$ and simulates $\mathcal{A}_\theta[2', 2']$.
- $\Rightarrow$ Topology admits $w = (b,a)(d,c)(a,b)(c,d)$-path from p.
- Since $|w| = 4 \leq r_\varphi + 2$, and $\mathfrak{T}$ is $(r_\varphi + 2)$-unambiguous, w forms a cycle in the topology as well.
- Every process has to simulate several automata.

Unambiguous topology classes

Almost the same proof works for a weaker logic without channel bound:

Theorem

Let $\varphi \in \mathrm{EMSO}[\lhd_{\mathsf{proc}}, \lhd_{\mathsf{msg}}, \sim]$, and $\mathfrak{T}$ be a $(r_\varphi + 2)$-unambiguous set of topologies. There is a PCA $\mathcal{A}$ such that, for all $\mathcal{T} \in \mathfrak{T}$, $L_{\mathcal{T}}(\mathcal{A}) = L_{\mathcal{T}}(\varphi)$.

Unambiguous topology classes

Almost the same proof works for a weaker logic without channel bound:

Theorem

Let $\varphi \in \mathrm{EMSO}[\lhd_{\mathsf{proc}}, \lhd_{\mathsf{msg}}, \sim]$, and $\mathfrak{T}$ be a $(r_\varphi + 2)$-unambiguous set of topologies. There is a PCA $\mathcal{A}$ such that, for all $\mathcal{T} \in \mathfrak{T}$, $L_{\mathcal{T}}(\mathcal{A}) = L_{\mathcal{T}}(\varphi)$.

Proof

- Translate φ into normal form

$$\exists X_1 \ldots \exists X_m \exists x_1 \ldots \exists x_n \forall y \psi$$

  where ψ is r_φ-local around y.
- Existential quantification $\Rightarrow$ projection & guessing of truth values for propositions involving only free variables of $\forall y \psi$ [Gastin-Kuske 2010].
- Construct fixed-topology CA $\mathcal{A}_\theta$ for formula $\forall y \psi$ over all topology neighborhoods θ of radius $\lceil r_\varphi / 2 \rceil$ [Genest-Kuske-Muscholl 2004].
- Glue fixed-topology CA together to obtain a PCA for $\forall y \psi$.

Unambiguous topology classes

Almost the same proof works for a weaker logic without channel bound:

Theorem

Let $\varphi \in \mathrm{EMSO}[\lhd_{\mathsf{proc}}, \lhd_{\mathsf{msg}}, \sim]$, and $\mathfrak{T}$ be a $(r_\varphi + 2)$-unambiguous set of topologies. There is a PCA $\mathcal{A}$ such that, for all $\mathcal{T} \in \mathfrak{T}$, $L_{\mathcal{T}}(\mathcal{A}) = L_{\mathcal{T}}(\varphi)$.

Proof

- Translate φ into normal form

  $$\exists X_1 \ldots \exists X_m \exists x_1 \ldots \exists x_n \forall y \psi$$

  where ψ is r_φ-local around y.
- Existential quantification $\Rightarrow$ projection & guessing of truth values for propositions involving only free variables of $\forall y \psi$ [Gastin-Kuske 2010].
- Construct fixed-topology CA $\mathcal{A}_\theta$ for formula $\forall y \psi$ over all topology neighborhoods θ of radius $\lceil r_\varphi/2 \rceil$ [B.-Leucker 2004].
- Glue fixed-topology CA together to obtain a PCA for $\forall y \psi$.

An orthogonal approach

Theorem

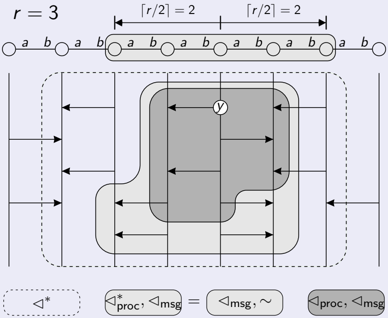
Let $\varphi \in \mathrm{EMSO}[\lhd_{\mathsf{proc}}, \lhd_{\mathsf{msg}}]$. There is a PCA $\mathcal{A}$ that is equivalent to φ on all pipelines, trees, and grids.

An orthogonal approach

Theorem

Let $\varphi \in \mathrm{EMSO}[\lhd_{\mathrm{proc}}, \lhd_{\mathrm{msg}}]$. There is a PCA $\mathcal{A}$ that is equivalent to φ on all pipelines, trees, and grids.

Proof

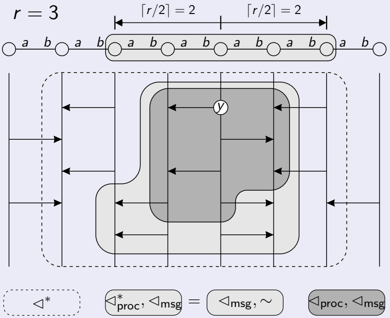


An orthogonal approach

Theorem

Let $\varphi \in \text{EMSO}[\lhd_{\text{proc}}, \lhd_{\text{msg}}]$. There is a PCA $\mathcal{A}$ that is equivalent to φ on all pipelines, trees, and grids.

Proof



Exploit sphere automaton from [B.-Leucker] to compute $\{\lhd_{\text{proc}}, \lhd_{\text{msg}}\}$-neighborhoods.

Summary of results

Negative results

- There is an $FO[\lhd_{proc}, \lhd_{msg}]$-formula that is not realizable for the class of ring forests.
- There is an FO-formula that is not realizable for the class of trees.

Summary of results

Negative results

- There is an FO[$\lessdot_{\mathsf{proc}}, \lessdot_{\mathsf{msg}}$]-formula that is not realizable for the class of ring forests.
- There is an FO-formula that is not realizable for the class of trees.

Positive results

- Under a channel bound, every FO[$\lessdot^*_{\mathsf{proc}}, \lessdot_{\mathsf{msg}}$]-formula is realizable for the classes of pipelines, trees, grids, and rings.
- Every FO[$\lessdot_{\mathsf{proc}}, \lessdot_{\mathsf{msg}}, \sim$]-formula is realizable for the classes of pipelines, trees, grids, and rings.

Summary of results

Negative results

- There is an $\mathrm{FO}[\lhd_{\mathrm{proc}}, \lhd_{\mathrm{msg}}]$-formula that is not realizable for the class of ring forests.
- There is an FO-formula that is not realizable for the class of trees.

Positive results

- Under a channel bound, every $\mathrm{FO}[\lhd^*_{\mathrm{proc}}, \lhd_{\mathrm{msg}}]$-formula is realizable for the classes of pipelines, trees, grids, and rings.
- Every $\mathrm{FO}[\lhd_{\mathrm{proc}}, \lhd_{\mathrm{msg}}, \sim]$-formula is realizable for the classes of pipelines, trees, grids, and rings.

Open problems

- Is every $\mathrm{FO}[\lhd^*_{\mathrm{proc}}, \lhd_{\mathrm{msg}}]$-formula realizable without channel bound?
- Is every $\mathrm{FO}[\lhd^*]$-formula realizable (for interesting classes of topologies)?

Related work

- Parameterized synthesis [Jacobs-Bloem 2012]
- Parameterized verification [Browne-Clarke-Grumberg 1989], [Emerson-Namjoshi 2003], [Bouajjani-Habermehl-Vojnar 2008], [Delzanno-Sangnier-Zavattaro 2010]
- Distributed algorithms [Grumbach-Wu 2010], [Chalopin-Das-Kosowski 2010]
- Automata from normal forms [Schwentick-Barthelmann 1999], [Gastin-Kuske 2010]

Conclusion

Contribution

- A notion of communicating automaton that is independent of a concrete topology
- Büchi-Elgot-Trakhtenbrot theorems for PCA

Conclusion

Contribution

- A notion of communicating automaton that is independent of a concrete topology
- Büchi-Elgot-Trakhtenbrot theorems for PCA

Future work

- Topologies of unbounded degree (unranked trees, star architectures)
- Parameterized verification

Thank You!