Expressivity of first-order logic, star-free propositional dynamic logic and communicating automata

Marie Fortin

PhD Defense – November 27, 2020

LSV, ENS Paris-Saclay, Université Paris-Saclay

Introduction

Verification

Verification

Requirements

System

Verification

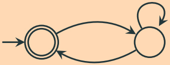
Requirements

Formal specification φ

$\neg(\exists x.\mathrm{error}(x))$

System

System model $\mathcal{A}$



Verification

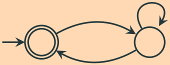
Requirements

System

Formal specification φ

$\neg(\exists x.\mathrm{error}(x))$

System model $\mathcal{A}$



Model-checking

$\mathcal{A} \models \varphi$?

Verification

Requirements

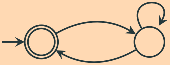
System

Formal specification φ

$\neg(\exists x.\mathsf{error}(x))$

$\xrightarrow{\text{Synthesis}}$

System model $\mathcal{A}$



Specification: many choices!

Example: “every request is eventually granted”

Specification: many choices!

Example: “every request is eventually granted”

$$\mathsf{G}(\text{request} \implies \mathsf{F}\,\text{grant})$$

Linear temporal logic (LTL)

Specification: many choices!

Example: "every request is eventually granted"

$$\mathsf{G}(\mathsf{request} \implies \mathsf{F}\,\mathsf{grant})$$

Linear temporal logic (LTL)

$$\forall x.(\mathsf{request}(x) \implies \exists y. x \leq y \wedge \mathsf{grant}(y))$$

First-order logic (FO)

Specification: many choices!

Example: "every request is eventually granted"

$\mathsf{G}(\mathsf{request} \implies \mathsf{F}\,\mathsf{grant})$

Linear temporal logic (LTL)

$\Sigma^* \cdot (\mathsf{request} \cdot \Sigma^* \cdot \mathsf{grant})^*$

Regular expression

$\forall x.(\mathsf{request}(x) \implies \exists y. x \leq y \land \mathsf{grant}(y))$

First-order logic (FO)

Specification: many choices!

Example: "every request is eventually granted"

$$\mathsf{G}(\mathsf{request} \implies \mathsf{F}\,\mathsf{grant})$$

Linear temporal logic (LTL)

$$\Sigma^* \cdot (\mathsf{request} \cdot \Sigma^* \cdot \mathsf{grant})^*$$

Regular expression

$$\forall x.(\mathsf{request}(x) \implies \exists y. x \leq y \land \mathsf{grant}(y))$$

First-order logic (FO)

Comparing specification languages

Specification: many choices!

Example: "every request is eventually granted"

$$\mathsf{G}(\mathsf{request} \implies \mathsf{F}\,\mathsf{grant})$$

Linear temporal logic (LTL)

$$\Sigma^* \cdot (\mathsf{request} \cdot \Sigma^* \cdot \mathsf{grant})^*$$

Regular expression

$$\forall x.(\mathsf{request}(x) \implies \exists y. x \leq y \wedge \mathsf{grant}(y))$$

First-order logic (FO)

Comparing specification languages

- Expressive power

Specification: many choices!

Example: "every request is eventually granted"

$$\mathsf{G}(\mathsf{request} \implies \mathsf{F}\,\mathsf{grant})$$

Linear temporal logic (LTL)

$$\Sigma^* \cdot (\mathsf{request} \cdot \Sigma^* \cdot \mathsf{grant})^*$$

Regular expression

$$\forall x.(\mathsf{request}(x) \implies \exists y. x \leq y \wedge \mathsf{grant}(y))$$

First-order logic (FO)

Comparing specification languages

- Expressive power
- Complexity/Decidability

Specification: many choices!

Example: "every request is eventually granted"

$$\mathsf{G}(\mathsf{request} \implies \mathsf{F}\,\mathsf{grant})$$

Linear temporal logic (LTL)

$$\Sigma^* \cdot (\mathsf{request} \cdot \Sigma^* \cdot \mathsf{grant})^*$$

Regular expression

$$\forall x.(\mathsf{request}(x) \implies \exists y. x \leq y \wedge \mathsf{grant}(y))$$

First-order logic (FO)

Comparing specification languages

- Expressive power
- Complexity/Decidability
- Succinctness

Specification: many choices!

Example: "every request is eventually granted"

$$\mathsf{G}(\mathsf{request} \implies \mathsf{F}\,\mathsf{grant})$$

Linear temporal logic (LTL)

$$\Sigma^* \cdot (\mathsf{request} \cdot \Sigma^* \cdot \mathsf{grant})^*$$

Regular expression

$$\forall x.(\mathsf{request}(x) \implies \exists y.x \leq y \wedge \mathsf{grant}(y))$$

First-order logic (FO)

Comparing specification languages

- Expressive power
- Complexity/Decidability
- Succinctness
- Convenience

Specification: many choices!

Example: "every request is eventually granted"

$$\mathsf{G}(\text{request} \implies \mathsf{F}\,\text{grant})$$

Linear temporal logic (LTL)

$$\Sigma^* \cdot (\text{request} \cdot \Sigma^* \cdot \text{grant})^*$$

Regular expression

$$\forall x.(\text{request}(x) \implies \exists y. x \leq y \wedge \text{grant}(y))$$

First-order logic (FO)

Comparing specification languages

- Expressive power
- Complexity/Decidability
- Succinctness
- Convenience

Specification: many choices!

Example: "every request is eventually granted"

$\mathsf{G}(\text{request} \implies \mathsf{F}\,\text{grant})$

Linear temporal logic (LTL)

$\Sigma^* \cdot (\text{request} \cdot \Sigma^* \cdot \text{grant})^*$

Regular expression

$\forall x.(\text{request}(x) \implies \exists y. x \leq y \wedge \text{grant}(y))$

First-order logic (FO)

Comparing specification languages

- Expressive power
- Complexity/Decidability
- Succinctness
- Convenience

Comparisons with automata

Specification: many choices!

Example: "every request is eventually granted"

$$\mathsf{G}(\mathsf{request} \implies \mathsf{F}\,\mathsf{grant})$$

Linear temporal logic (LTL)

$$\Sigma^* \cdot (\mathsf{request} \cdot \Sigma^* \cdot \mathsf{grant})^*$$

Regular expression

$$\forall x.(\mathsf{request}(x) \implies \exists y. x \leq y \wedge \mathsf{grant}(y))$$

First-order logic (FO)

Comparing specification languages

- Expressive power
- Complexity/Decidability
- Succinctness
- Convenience

Comparisons with automata

Given a specification, can we always construct an equivalent automaton?

Some expressivity results over finite words

Some expressivity results over finite words

Theorem [Büchi 1960, Elgot 1961, Trakhtenbrot 1962]

Finite automata and MSO have the same expressive power.

Some expressivity results over finite words

Theorem [Büchi 1960, Elgot 1961, Trakhtenbrot 1962]

Finite automata and MSO have the same expressive power.

Theorem

Over words, FO$[<]$ defines the same class of languages as

- LTL [Kamp 1968]
- FO$^3[<]$ [Kamp 1968]
- Star-free expressions [McNaughton, Papert 1971]
- Counter-free automata [McNaughton, Papert 1971]
- Aperiodic monoids [Schützenberger 1965]

Some expressivity results over finite words

Theorem [Büchi 1960, Elgot 1961, Trakhtenbrot 1962]

Finite automata and MSO have the same expressive power.

Theorem

Over words, $\mathrm{FO}[<]$ defines the same class of languages as

- LTL [Kamp 1968]
- $\mathrm{FO}^3[<]$ [Kamp 1968]
- Star-free expressions [McNaughton, Papert 1971]
- Counter-free automata [McNaughton, Papert 1971]
- Aperiodic monoids [Schützenberger 1965]

What about more complex structures?

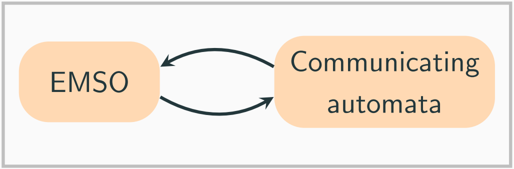
Outline

Outline

Logic-automata connections

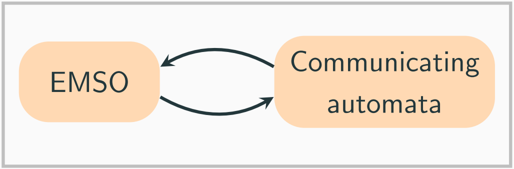
Outline

Logic-automata connections



Outline

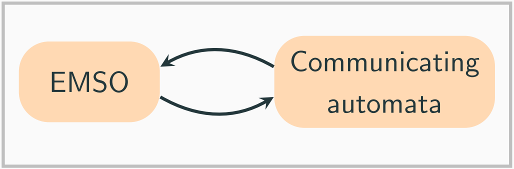
Logic-automata connections

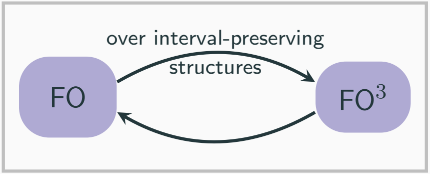


The 3-variable property

Outline

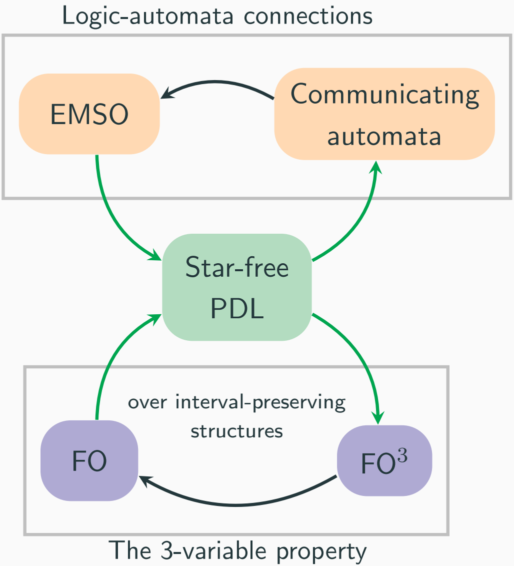
Logic-automata connections



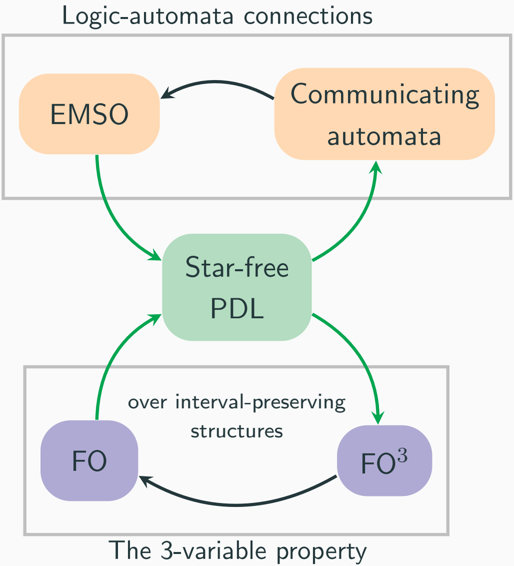


The 3-variable property

Outline


Logic-automata connections
EMSO
Communicating automata
Star-free PDL
over interval-preserving structures
FO
FO^3
The 3-variable property

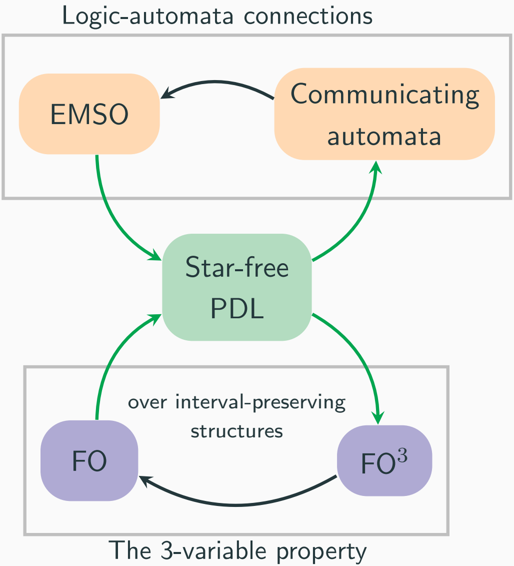
Outline



Part I

Star-free PDL

Outline



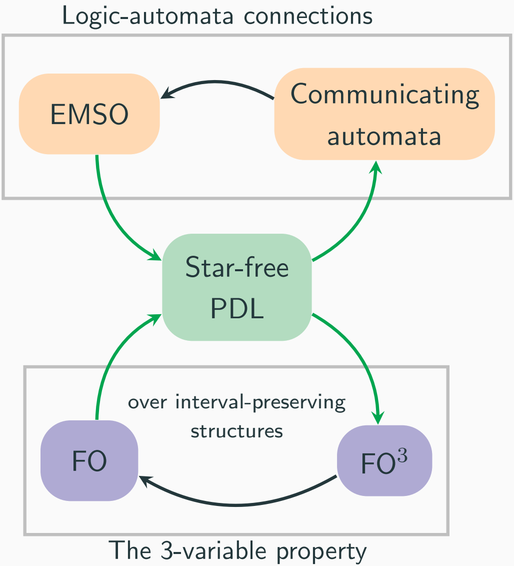
Part II

A Büchi theorem for message-passing systems

Part I

Star-free PDL

Outline



Part II

A Büchi theorem for message-passing systems

Part I

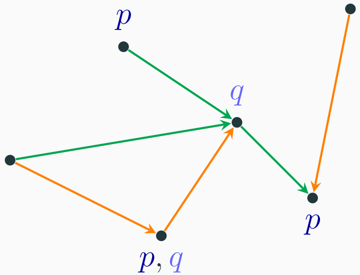
Star-free PDL

Part III

Sufficient conditions for the 3-variable property

Star-free PDL

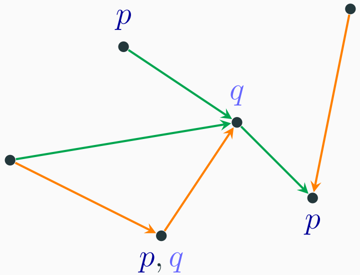
Examples



- Binary relations $\rightarrow$ and $\rightarrow$
- Unary predicates p, q

Examples

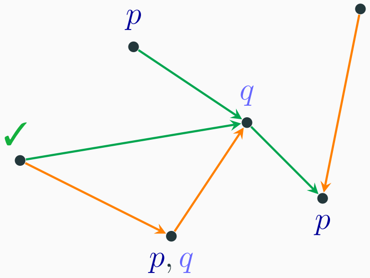
- $p \vee \langle \textcolor{orange}{\rightarrow} \rangle\, q$



- Binary relations $\textcolor{orange}{\rightarrow}$ and $\textcolor{green}{\rightarrow}$
- Unary predicates p, q

Examples

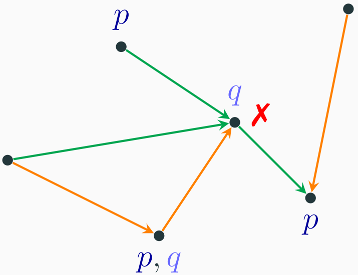
- $p \lor \langle \rightarrow \rangle\, q$



- Binary relations $\rightarrow$ and $\rightarrow$
- Unary predicates p, q

Examples

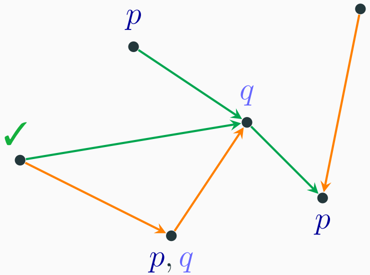
- $p \vee \langle \rightarrow \rangle q$



- Binary relations $\rightarrow$ and $\rightarrow$
- Unary predicates p, q

Examples

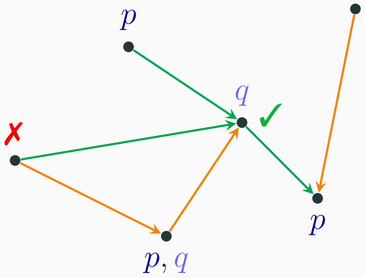
- $p \vee \langle \rightarrow \rangle\, q$
- $\langle \rightarrow \cdot \rightarrow^{-1} \rangle\, (p \vee q)$



- Binary relations $\rightarrow$ and $\rightarrow$
- Unary predicates p, q

Examples

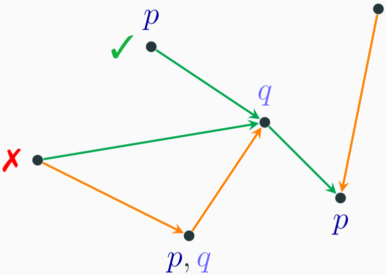
- $p \vee \langle \rightarrow \rangle\, q$
- $\langle \rightarrow \cdot \rightarrow^{-1} \rangle\, (p \vee q)$
- $\langle \rightarrow \cdot \{p\}? \cdot \rightarrow^{-1} \rangle\, \mathit{true}$



- Binary relations $\rightarrow$ and $\rightarrow$
- Unary predicates p, q

Examples

- $p \vee \langle \rightarrow \rangle\, q$
- $\langle \rightarrow \cdot \rightarrow^{-1} \rangle\, (p \vee q)$
- $\langle \rightarrow \cdot \{p\}? \cdot \rightarrow^{-1} \rangle\, true$
- $\langle \rightarrow \cap (\rightarrow \cdot \rightarrow)^{\mathsf{c}} \rangle\, q$



- Binary relations $\rightarrow$ and $\rightarrow$
- Unary predicates p, q

Example: MTL modalities

Over structures with

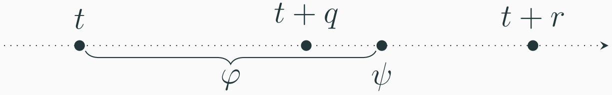
- domain $\mathbb{R}$
- binary relations $<$ and $+q$ for every $q \in \mathbb{Q}_+$
- atomic propositions $P, Q, \ldots$

Example: MTL modalities

Over structures with

- domain $\mathbb{R}$
- binary relations $<$ and $+q$ for every $q \in \mathbb{Q}_+$
- atomic propositions $P, Q, \ldots$

$\varphi \, \mathsf{U}_{(q,r)} \, \psi \equiv$

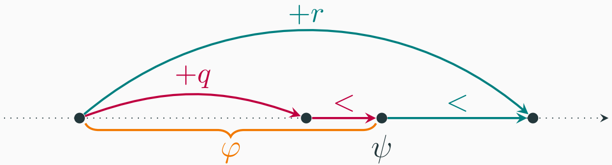


Example: MTL modalities

Over structures with

- domain $\mathbb{R}$
- binary relations $<$ and $+q$ for every $q \in \mathbb{Q}_+$
- atomic propositions $P, Q, \ldots$

$$\varphi \,\mathsf{U}_{(q,r)}\, \psi \equiv \left\langle (+q \cdot <) \cap (+r \cdot <^{-1}) \cap (< \cdot \{\neg\varphi\}? \cdot <)^{\mathsf{c}} \right\rangle \psi$$



Syntax of Star-free Propositional Dynamic Logic

State formulas:

$$\varphi ::= p \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\pi\rangle\,\varphi$$

Path formulas:

$$\pi ::= R \mid \{\varphi\}? \mid \pi^{-1} \mid \pi \cdot \pi \mid \pi \cup \pi \mid \pi^{\mathsf{c}}$$

$\mathsf{PDL}_{\mathsf{sf}}$

Syntax of Star-free Propositional Dynamic Logic

State formulas:

$$\varphi ::= p \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\pi\rangle\,\varphi$$

Path formulas:

$$\pi ::= R \mid \{\varphi\}? \mid \pi^{-1} \mid \pi \cdot \pi \mid \pi \cup \pi \mid \pi^{\mathsf{c}}$$

$\mathsf{PDL}_{\mathsf{sf}}$

Combines features from

- Propositional Dynamic Logic [Fisher, Ladner 1979]
- Star-free regular expressions
- The calculus of relations

Syntax of Star-free Propositional Dynamic Logic

State formulas:

$$\varphi ::= p \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\pi\rangle\,\varphi$$

Path formulas:

$$\pi ::= R \mid \{\varphi\}? \mid \pi^{-1} \mid \pi \cdot \pi \mid \pi \cup \pi \mid \pi^{\mathsf{c}}$$

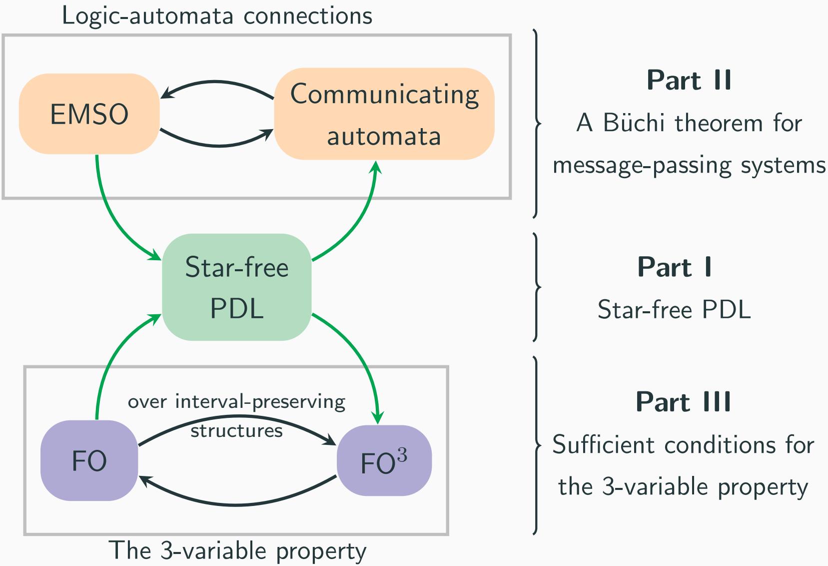
$\mathsf{PDL}_{\mathsf{sf}}$

Combines features from

- Propositional Dynamic Logic [Fisher, Ladner 1979]
- Star-free regular expressions
- The calculus of relations

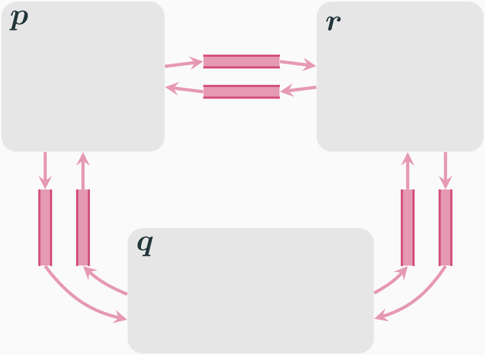
Theorem [Tarski, Givant 1987] (calculus of relations)

$\mathsf{PDL}_{\mathsf{sf}}$ and FO^3 are expressively equivalent.


Logic-automata connections
EMSO
Communicating automata
Star-free PDL
over interval-preserving structures
FO
FO^3
The 3-variable property
Part II
A Büchi theorem for message-passing systems
Part I
Star-free PDL
Part III
Sufficient conditions for the 3-variable property

A Büchi theorem for message-passing systems

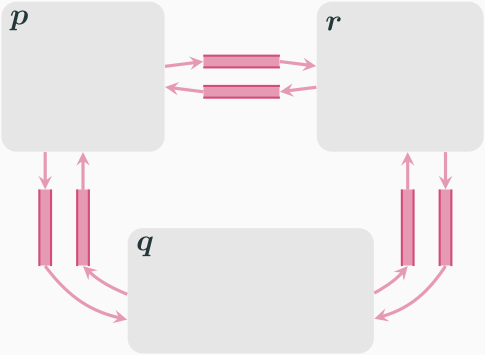
Communicating finite-state machines (CFMs)[1]



[1] [Brand, Zafiropulo 1983]

Communicating finite-state machines (CFMs)[1]

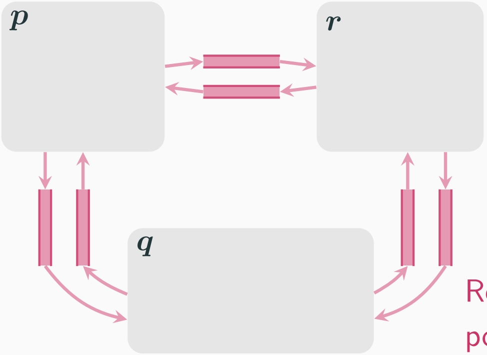
Fixed, finite set of processes, e.g. $\{p, q, r\}$



[1] [Brand, Zafiropulo 1983]

Communicating finite-state machines (CFMs)[1]

Fixed, finite set of processes, e.g. $\{p, q, r\}$

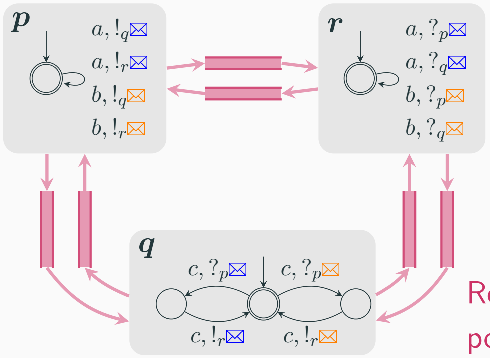


Reliable unbounded point-to-point FIFO channels

[1] [Brand, Zafiropulo 1983]

Communicating finite-state machines (CFMs)[1]

Fixed, finite set of processes, e.g. $\{p, q, r\}$



One finite automaton for each process

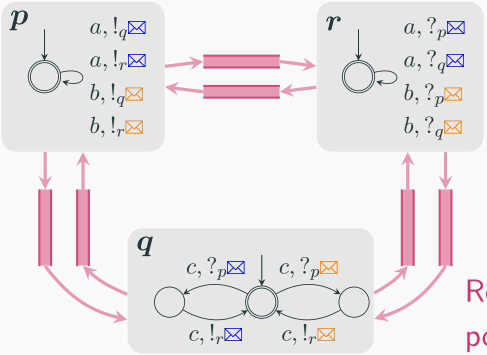
- Finite input alphabet, e.g. $\Sigma = \{a, b, c\}$
- Sends/receives from a finite message alphabet, e.g. {✉, ✉}

Reliable unbounded point-to-point FIFO channels

[1] [Brand, Zafiropulo 1983]

Communicating finite-state machines (CFMs)[1]

Fixed, finite set of processes, e.g. $\{p, q, r\}$



Global acceptance condition

One finite automaton for each process

- Finite input alphabet, e.g. $\Sigma = \{a, b, c\}$
- Sends/receives from a finite message alphabet, e.g. {✉, ✉}

Reliable unbounded point-to-point FIFO channels

[1] [Brand, Zafiropulo 1983]

Message Sequence Charts (MSC)

The language of a CFM is a set of MSCs.

Message Sequence Charts (MSC)

The language of a CFM is a set of MSCs.

$$\boldsymbol{p} \quad a \longrightarrow a \longrightarrow c \longrightarrow a \longrightarrow a \longrightarrow a \longrightarrow a \longrightarrow a$$

$$\boldsymbol{q} \quad a \longrightarrow a \longrightarrow a \longrightarrow a \longrightarrow a \longrightarrow a \longrightarrow a \longrightarrow a \longrightarrow a \longrightarrow a$$

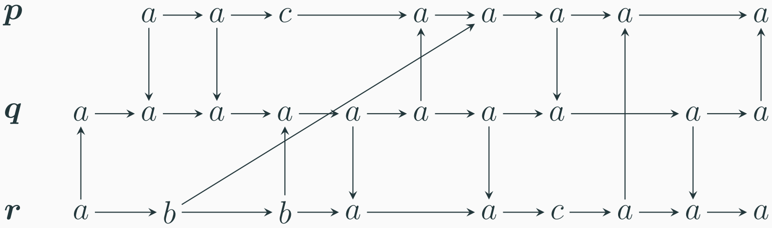
$$\boldsymbol{r} \quad a \longrightarrow b \longrightarrow b \longrightarrow a \longrightarrow a \longrightarrow c \longrightarrow a \longrightarrow a \longrightarrow a$$

Partial order consisting of

- One sequence of events for each process

Message Sequence Charts (MSC)

The language of a CFM is a set of MSCs.

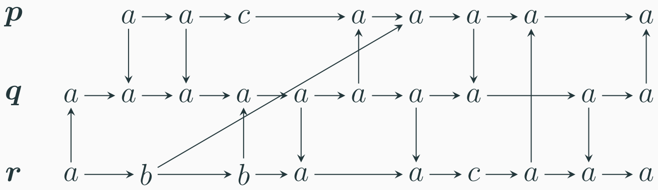


Partial order consisting of

- One sequence of events for each process
- Message relation connecting matching sends and receives

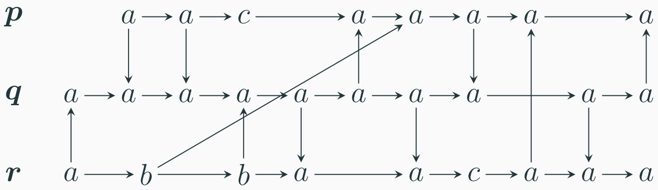
Monadic Second-Order logic (MSO) over MSCs

$\varphi ::=$



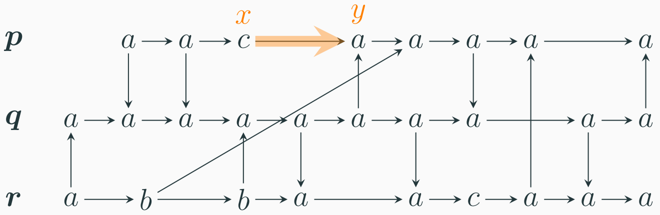
Monadic Second-Order logic (MSO) over MSCs

$\varphi ::= \quad a(x) \mid p(x)$ label/process of event x



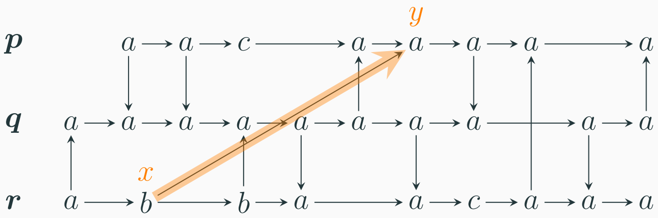
Monadic Second-Order logic (MSO) over MSCs

$\varphi ::= \quad a(x) \mid p(x)$	label/process of event x
$\quad\quad \mid x \to y$	process successor



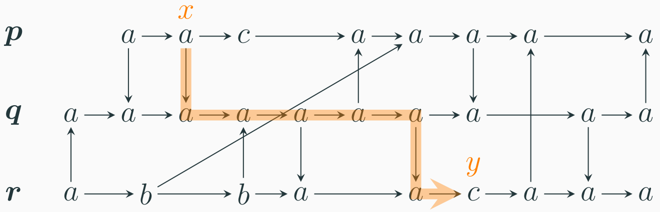
Monadic Second-Order logic (MSO) over MSCs

$\varphi ::=$	$a(x) \mid p(x)$	label/process of event x
	$\mid x \rightarrow y$	process successor
	$\mid x \lhd y$	message relation



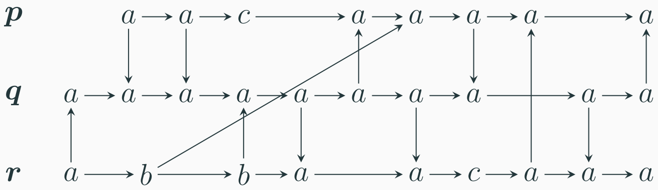
Monadic Second-Order logic (MSO) over MSCs

$\varphi ::=$	$a(x) \mid p(x)$	label/process of event x
	$\mid x \rightarrow y$	process successor
	$\mid x \lhd y$	message relation
	$\mid x \leq y$	happened-before [Lamport 1978]



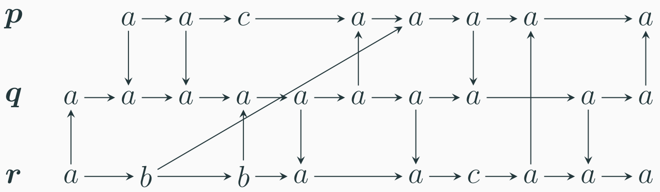
Monadic Second-Order logic (MSO) over MSCs

$\varphi ::= \; a(x) \mid p(x)$	label/process of event x
$\quad \mid x \to y$	process successor
$\quad \mid x \lhd y$	message relation
$\quad \mid x \leq y$	happened-before [Lamport 1978]
$\quad \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x.\ \varphi \mid \exists X.\ \varphi \mid x \in X$	



Monadic Second-Order logic (MSO) over MSCs

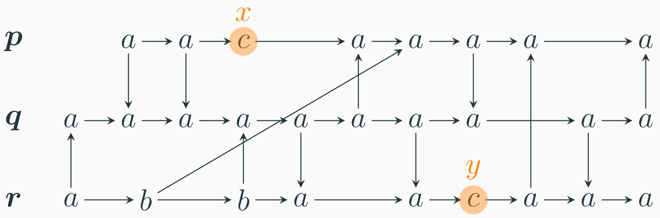
$$
\begin{array}{lll}
\varphi ::= & a(x) \mid p(x) & \text{label/process of event } x \\
& \mid x \rightarrow y & \text{process successor} \\
& \mid x \lhd y & \text{message relation} \\
& \mid x \leq y & \text{happened-before [Lamport 1978]} \\
& \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x.\ \varphi \mid \exists X.\ \varphi \mid x \in X &
\end{array}
$$



Example: mutual exclusion $\neg(\exists x.\ \exists y.\ c(x) \wedge c(y) \wedge\ x \parallel y\ )$

Monadic Second-Order logic (MSO) over MSCs

$\varphi ::= \quad a(x) \mid p(x)$	label/process of event x
$\mid x \rightarrow y$	process successor
$\mid x \lhd y$	message relation
$\mid x \leq y$	happened-before [Lamport 1978]
$\mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x.\ \varphi \mid \exists X.\ \varphi \mid x \in X$	



Example: mutual exclusion $\neg(\exists x.\ \exists y.\ c(x) \wedge c(y) \wedge\ x \parallel y\ )$

$\neg(x \leq y) \wedge \neg(y \leq x)$

Expressive power of CFMs

Theorem [Büchi 1960, Elgot 1961, Trakhtenbrot 1962]

If there is only one process,
CFMs ($=$ finite automata) $=$ MSO$[\lessdot, \leq]$.

Expressive power of CFMs

Theorem [Büchi 1960, Elgot 1961, Trakhtenbrot 1962]

If there is only one process,
CFMs ($=$ finite automata) $=$ MSO$[\lessdot, \leq]$.

With ≥ 2 processes, CFMs are not closed under complement.

[Bollig, Leucker 2006]

Expressive power of CFMs

Theorem [Büchi 1960, Elgot 1961, Trakhtenbrot 1962]

If there is only one process,
CFMs ($=$ finite automata) $=$ MSO$[\lhd, \leq]$.

With ≥ 2 processes, CFMs are not closed under complement.

[Bollig, Leucker 2006]

But:

Theorem [Bollig, Leucker 2006]

Over finite MSCs, CFM $=$ EMSO$[\lhd, \rightarrow]$.

Expressive power of CFMs

Theorem [Büchi 1960, Elgot 1961, Trakhtenbrot 1962]

If there is only one process,
CFMs (= finite automata) = MSO$[\lhd, \leq]$.

With ≥ 2 processes, CFMs are not closed under complement.

[Bollig, Leucker 2006]

But:

Theorem [Bollig, Leucker 2006]

Over finite MSCs, CFM = EMSO$[\lhd, \rightarrow]$.

$\exists X_0 \ldots \exists X_n . \varphi$ with $\varphi \in$ FO$[\lhd, \rightarrow]$

Expressive power of CFMs

Theorem [Büchi 1960, Elgot 1961, Trakhtenbrot 1962]

If there is only one process,
CFMs ($=$ finite automata) $=$ MSO$[\lhd, \leq]$.

With ≥ 2 processes, CFMs are not closed under complement.

[Bollig, Leucker 2006]

But:

Theorem [Bollig, Leucker 2006]

Over finite MSCs, CFM $=$ EMSO$[\lhd, \rightarrow]$.

Theorem [Bollig, Kuske 2008]

Over infinite MSCs, CFM $=$ EMSO$^\infty[\lhd, \rightarrow]$.

Expressive power of CFMs

Theorem [Büchi 1960, Elgot 1961, Trakhtenbrot 1962]

If there is only one process,
CFMs (= finite automata) = MSO$[\lhd, \leq]$.

With ≥ 2 processes, CFMs are not closed under complement.

[Bollig, Leucker 2006]

But:

Theorem [Bollig, Leucker 2006]

Over finite MSCs, CFM = EMSO$[\lhd, \rightarrow]$.

Theorem [Bollig, Kuske 2008]

Over infinite MSCs, CFM = EMSO$^\infty[\lhd, \rightarrow]$.

Note: here the happened-before relation $\leq$ is not included.

$\rightarrow$ Harder to formalize concurrency properties such as mutual exclusion.

Expressive power of CFMs

The equivalence with $\mathrm{MSO}[\lhd, \leq]$ is recovered if we assume that the channels are of bounded size.

Theorem

$\mathrm{CFM} = \mathrm{MSO}[\lhd, \leq]$ over

- finite, universally bounded MSCs [Henriksen,Mukund,Narayan Kumar,Sohoni,Thiagarajan 2005]
- infinite, universally bounded MSCs [Kuske 2003]
- finite, existentially bounded MSCs [Genest, Kuske, Muscholl 2006]

Expressive power of CFMs

The equivalence with $\mathrm{MSO}[\lessdot, \leq]$ is recovered if we assume that the channels are of bounded size.

Theorem

$\mathrm{CFM} = \mathrm{MSO}[\lessdot, \leq]$ over

- finite, universally bounded MSCs [Henriksen,Mukund,Narayan Kumar,Sohoni,Thiagarajan 2005]
- infinite, universally bounded MSCs [Kuske 2003]
- finite, existentially bounded MSCs [Genest, Kuske, Muscholl 2006]

Remark: model-checking is undecidable in general, but decidable when restricted to bounded MSCs.

Expressive power of CFMs

What happens for formulas with the happened-before relation, over unbounded channels?

Expressive power of CFMs

What happens for formulas with the happened-before relation, over unbounded channels?

Theorem [Bollig,F.,Gastin 2018 & 2020]

Over finite and infinite MSCs, CFM $=$ EMSO$[\lhd, \leq]$.

Expressive power of CFMs

What happens for formulas with the happened-before relation, over unbounded channels?

Theorem [Bollig,F.,Gastin 2018 & 2020]

Over finite and infinite MSCs, CFM $=$ EMSO$[\lhd, \leq]$.

Other consequence: new, direct proof for the bounded case:

Theorem [Bollig,F.,Gastin 2020]

Over finite and infinite **existentially bounded** MSCs, CFM $=$ MSO$[\lhd, \leq]$.

Expressive power of CFMs

What happens for formulas with the happened-before relation, over unbounded channels?

Theorem [Bollig,F.,Gastin 2018 & 2020]

Over finite and infinite MSCs, CFM $=$ EMSO$[\lhd, \leq]$.

Other consequence: new, direct proof for the bounded case:

Theorem [Bollig,F.,Gastin 2020]

Over finite and infinite **existentially bounded** MSCs, CFM $=$ MSO$[\lhd, \leq]$.

Some proof ideas

Remarks

Some proof ideas

Remarks

- CFM $\subseteq$ EMSO$[\lhd, \leq]$ already known

Some proof ideas

Remarks

- CFM $\subseteq$ EMSO$[\lhd, \leq]$ already known
- EMSO$[\lhd, \leq] \subseteq$ CFM follows from FO$[\lhd, \leq] \subseteq$ CFM and closure under projection

Some proof ideas

Remarks

- CFM $\subseteq$ EMSO$[\lessdot, \leq]$ already known
- EMSO$[\lessdot, \leq] \subseteq$ CFM follows from FO$[\lessdot, \leq] \subseteq$ CFM and closure under projection
- FO$[\lessdot, \leq]$ = FO$[\lessdot, \leq_{\mathsf{proc}}]$

Some proof ideas

Remarks

- CFM $\subseteq$ EMSO$[\lessdot, \leq]$ already known
- EMSO$[\lessdot, \leq] \subseteq$ CFM follows from FO$[\lessdot, \leq] \subseteq$ CFM and closure under projection
- FO$[\lessdot, \leq]$ = FO$[\lessdot, \leq_{\mathsf{proc}}]$

Goal: FO$[\lessdot, \leq_{\mathsf{proc}}] \subseteq$ CFM

Some proof ideas

Remarks

- CFM $\subseteq$ EMSO$[\lessdot, \leq]$ already known
- EMSO$[\lessdot, \leq] \subseteq$ CFM follows from FO$[\lessdot, \leq] \subseteq$ CFM and closure under projection
- FO$[\lessdot, \leq] =$ FO$[\lessdot, \leq_{\mathsf{proc}}]$

Goal: FO$[\lessdot, \leq_{\mathsf{proc}}] \subseteq$ CFM

Difficulty: CFMs are not closed under complement.

Some proof ideas

Remarks

- CFM $\subseteq$ EMSO$[\lessdot, \leq]$ already known
- EMSO$[\lessdot, \leq] \subseteq$ CFM follows from FO$[\lessdot, \leq] \subseteq$ CFM and closure under projection
- FO$[\lessdot, \leq] =$ FO$[\lessdot, \leq_{\mathsf{proc}}]$

Goal: FO$[\lessdot, \leq_{\mathsf{proc}}] \subseteq$ CFM

Difficulty: CFMs are not closed under complement.

Idea: Use fragments of star-free PDL as intermediate steps.

A fragment of $\mathrm{PDL}_{\mathsf{sf}}$ without explicit complements

State formulas:

$$\varphi ::= P \mid p \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\pi\rangle\,\varphi$$

Path formulas:

$$\pi ::= \rightarrow \mid \lhd_{p,q} \mid \{\varphi\}? \mid \pi^{-1} \mid \pi \cdot \pi \mid \;\pi \cup \pi\; \mid \;\pi^{\mathsf{c}}$$

A fragment of $\mathrm{PDL}_{\mathrm{sf}}$ without explicit complements

State formulas:

$$\varphi ::= P \mid p \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\pi\rangle\,\varphi$$

Path formulas:

$$\pi ::= \rightarrow \mid \lhd_{p,q} \mid \{\varphi\}? \mid \pi^{-1} \mid \pi \cdot \pi \mid \cancel{\pi \cup \pi} \mid \cancel{\pi^{\mathsf{c}}}$$

A fragment of $\mathrm{PDL}_{\mathrm{sf}}$ without explicit complements

State formulas:

$$\varphi ::= P \mid p \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\pi\rangle\,\varphi \mid \mathsf{Loop}(\pi)$$

Path formulas:

$$\pi ::= \rightarrow \mid \lhd_{p,q} \mid \{\varphi\}? \mid \pi^{-1} \mid \pi\cdot\pi \mid \xcancel{\pi\cup\pi} \mid \xcancel{\pi^{c}}$$

A fragment of $\mathrm{PDL}_{\mathrm{sf}}$ without explicit complements

State formulas:

$$\varphi ::= P \mid p \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\pi\rangle\,\varphi \mid \mathsf{Loop}(\pi)$$

Path formulas:

$$\pi ::= \rightarrow \mid \lhd_{p,q} \mid \{\varphi\}? \mid \pi^{-1} \mid \pi \cdot \pi \mid \cancel{\pi \cup \pi} \mid \cancel{\pi^{c}} \mid \top \mid \xrightarrow{\varphi}$$

A fragment of PDL$_{\mathsf{sf}}$ without explicit complements

State formulas:

$$\varphi ::= P \mid p \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\pi\rangle\,\varphi \mid \mathsf{Loop}(\pi)$$

Path formulas:

$$\pi ::= \rightarrow \mid \lhd_{p,q} \mid \{\varphi\}? \mid \pi^{-1} \mid \pi\cdot\pi \mid \cancel{\pi\cup\pi} \mid \cancel{\pi^{c}} \mid \top \mid \xrightarrow{\varphi}$$

- $\mathsf{Loop}(\pi)$: $e \in [\![\mathsf{Loop}(\pi)]\!]$ if $(e,e) \in [\![\pi]\!]$

A fragment of $\mathrm{PDL}_{\mathsf{sf}}$ without explicit complements

State formulas:

$$\varphi ::= P \mid p \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\pi\rangle\,\varphi \mid \mathsf{Loop}(\pi)$$

Path formulas:

$$\pi ::= \rightarrow \mid \lhd_{p,q} \mid \{\varphi\}? \mid \pi^{-1} \mid \pi \cdot \pi \mid \cancel{\pi \cup \pi} \mid \cancel{\pi^{c}} \mid \top \mid \xrightarrow{\varphi}$$

- $\mathsf{Loop}(\pi)$: $e \in [\![\mathsf{Loop}(\pi)]\!]$ if $(e, e) \in [\![\pi]\!]$
- $\top$ is the universal relation

A fragment of $\mathsf{PDL}_{\mathsf{sf}}$ without explicit complements

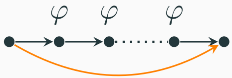
State formulas:

$$\varphi ::= P \mid p \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle \pi \rangle\, \varphi \mid \mathsf{Loop}(\pi)$$

Path formulas:

$$\pi ::= \rightarrow \mid \lhd_{p,q} \mid \{\varphi\}? \mid \pi^{-1} \mid \pi \cdot \pi \mid \cancel{\pi \cup \pi} \mid \cancel{\pi^{c}} \mid \top \mid \xrightarrow{\varphi}$$

- $\mathsf{Loop}(\pi)$: $e \in [\![\mathsf{Loop}(\pi)]\!]$ if $(e, e) \in [\![\pi]\!]$
- $\top$ is the universal relation
- $\xrightarrow{\varphi}$: similar to Until



A fragment of $\mathsf{PDL}_{\mathsf{sf}}$ without explicit complements

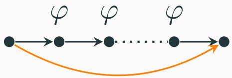
$\mathsf{PDL}_{\mathsf{sf}}[\xrightarrow{\varphi}, \mathsf{Loop}]$

State formulas:

$$\varphi ::= P \mid p \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\pi\rangle\,\varphi \mid \mathsf{Loop}(\pi)$$

Path formulas:

$$\pi ::= \rightarrow \mid \lhd_{p,q} \mid \{\varphi\}? \mid \pi^{-1} \mid \pi \cdot \pi \mid \cancel{\pi \cup \pi} \mid \cancel{\pi^{c}} \mid \top \mid \xrightarrow{\varphi}$$

- $\mathsf{Loop}(\pi)$: $e \in [\![\mathsf{Loop}(\pi)]\!]$ if $(e, e) \in [\![\pi]\!]$
- $\top$ is the universal relation
- $\xrightarrow{\varphi}$: similar to Until 

A fragment of $\mathsf{PDL}_{\mathsf{sf}}$ without explicit complements

$\mathsf{PDL}_{\mathsf{sf}}[\xrightarrow{\varphi}, \mathsf{Loop}]$

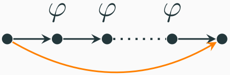
State formulas:

$$\varphi ::= P \mid p \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\pi\rangle\,\varphi \mid \mathsf{Loop}(\pi)$$

Path formulas:

$$\pi ::= \to \mid \lhd_{p,q} \mid \{\varphi\}? \mid \pi^{-1} \mid \pi \cdot \pi \mid \cancel{\pi \cup \pi} \mid \cancel{\pi^{c}} \mid \top \mid \xrightarrow{\varphi}$$

- $\mathsf{Loop}(\pi)$: $e \in [\![\mathsf{Loop}(\pi)]\!]$ if $(e, e) \in [\![\pi]\!]$
- $\top$ is the universal relation
- $\xrightarrow{\varphi}$: similar to Until



Negation occurs only at the level of state formulas (as in LTL)

Translation from first-order logic to CFMs

$\varphi \in \mathsf{FO}[\lhd, \leq]$

CFM $\mathcal{A}$ with $L(\mathcal{A}) = L(\varphi)$

Translation from first-order logic to CFMs

$\varphi \in \mathsf{FO}[\lhd, \leq]$

translation into star-free PDL

(non-elementary)

$\varphi' \in \mathsf{PDL_{sf}}[\xrightarrow{\varphi}, \mathsf{Loop}]$ with $\varphi \equiv \varphi'$

CFM $\mathcal{A}$ with $L(\mathcal{A}) = L(\varphi)$

Translation from first-order logic to CFMs

$\varphi \in \mathsf{FO}[\lessdot, \leq]$

translation into star-free PDL

(non-elementary)

$\varphi' \in \mathsf{PDL}_{\mathsf{sf}}[\xrightarrow{\varphi}, \mathsf{Loop}]$ with $\varphi \equiv \varphi'$

removal of Loop formulas

($\textsc{Ptime}$)

$\varphi'' \in \mathsf{PDL}_{\mathsf{sf}}[\xrightarrow{\varphi}]$ with $\varphi'' \equiv \varphi'$ up to projection

CFM $\mathcal{A}$ with $L(\mathcal{A}) = L(\varphi)$

Translation from first-order logic to CFMs

$\varphi \in \mathsf{FO}[\lhd, \leq]$

$\downarrow$ translation into star-free PDL (non-elementary)

$\varphi' \in \mathsf{PDL_{sf}}[\xrightarrow{\varphi}, \mathsf{Loop}]$ with $\varphi \equiv \varphi'$

$\downarrow$ removal of Loop formulas (PTIME)

$\varphi'' \in \mathsf{PDL_{sf}}[\xrightarrow{\varphi}]$ with $\varphi'' \equiv \varphi'$ up to projection

$\downarrow$ similar to LTL $\rightarrow$ finite automata (PSPACE)

CFM $\mathcal{A}$ with $L(\mathcal{A}) = L(\varphi)$

Translation from first-order logic to CFMs

$\psi = \exists X_1 \ldots \exists X_n . \varphi$
$\in \mathsf{EMSO}[\lhd, \leq]$

$\varphi \in \mathsf{FO}[\lhd, \leq]$

translation into star-free PDL
(non-elementary)

$\varphi' \in \mathsf{PDL}_{\mathsf{sf}}[\xrightarrow{\varphi}, \mathsf{Loop}]$ with $\varphi \equiv \varphi'$

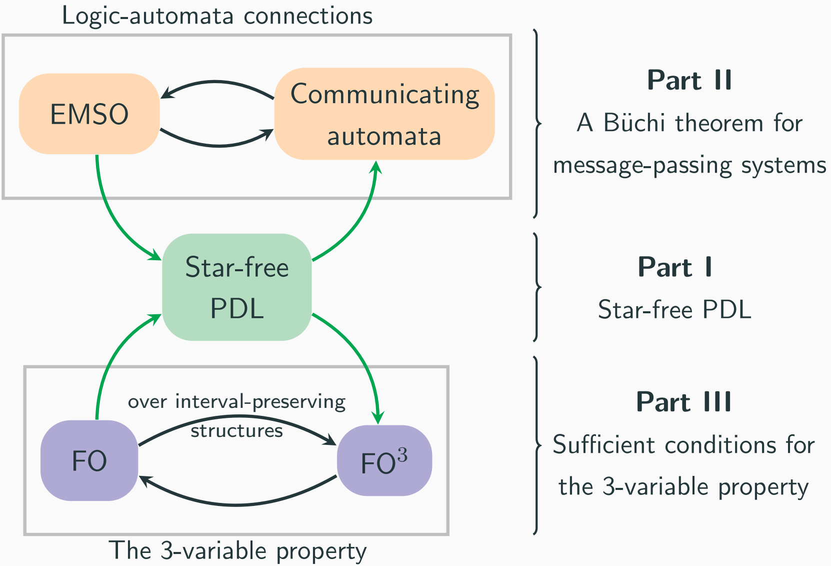
removal of Loop formulas
(Ptime)

$\varphi'' \in \mathsf{PDL}_{\mathsf{sf}}[\xrightarrow{\varphi}]$ with $\varphi'' \equiv \varphi'$ up to projection

similar to LTL $\rightarrow$ finite automata
(Pspace)

CFM $\mathcal{A}$ with $L(\mathcal{A}) = L(\varphi)$

projection


Logic-automata connections
EMSO
Communicating automata
Star-free PDL
over interval-preserving structures
FO
FO^3
The 3-variable property
Part II
A Büchi theorem for message-passing systems
Part I
Star-free PDL
Part III
Sufficient conditions for the 3-variable property

The 3-variable property for interval-preserving structures

Reminder

Theorem

Over MSCs, FO and $\mathsf{PDL}_{\mathsf{sf}}$ have the same expressive power.

Reminder

Theorem

Over MSCs, FO and $\mathsf{PDL}_{\mathsf{sf}}$ have the same expressive power.

Theorem

$\mathsf{PDL}_{\mathsf{sf}}$ and FO^3 have the same expressive power.

Reminder

Theorem

Over MSCs, FO and $\mathsf{PDL}_{\mathsf{sf}}$ have the same expressive power.

Theorem

$\mathsf{PDL}_{\mathsf{sf}}$ and FO^3 have the same expressive power.

formulas with at most 3 variables

Reminder

Theorem

Over MSCs, FO and $\mathrm{PDL}_{\mathsf{sf}}$ have the same expressive power.

Theorem

$\mathrm{PDL}_{\mathsf{sf}}$ and FO^3 have the same expressive power.

formulas with at most 3 variables

$\rightarrow$ **MSCs have the 3-variable property.**

The k-variable property

- Over arbitrary structures, strict hierarchy

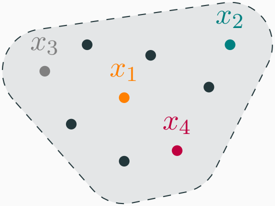
$$\mathsf{FO}^1 \subsetneq \mathsf{FO}^2 \subsetneq \mathsf{FO}^3 \subsetneq \mathsf{FO}^4 \subsetneq \cdots$$

The k-variable property

- Over arbitrary structures, strict hierarchy

$$\mathsf{FO}^1 \subsetneq \mathsf{FO}^2 \subsetneq \mathsf{FO}^3 \subsetneq \mathsf{FO}^4 \subsetneq \cdots$$

$$\exists x_1.\exists x_2.\exists x_3.\exists x_4.\ \bigwedge_{1\le i<j\le 4} x_i \neq x_j$$

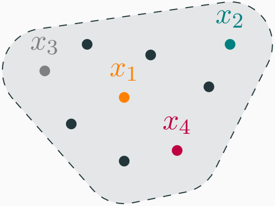


The k-variable property

- Over arbitrary structures, strict hierarchy

$$\mathsf{FO}^1 \subsetneq \mathsf{FO}^2 \subsetneq \mathsf{FO}^3 \subsetneq \mathsf{FO}^4 \subsetneq \cdots$$

$$\exists x_1. \exists x_2. \exists x_3. \exists x_4. \bigwedge_{1 \le i < j \le 4} x_i \neq x_j$$



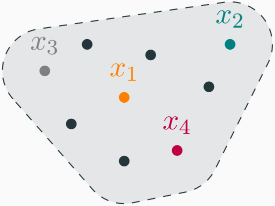
- For some classes of models, the hierarchy collapses:

The k-variable property

- Over arbitrary structures, strict hierarchy

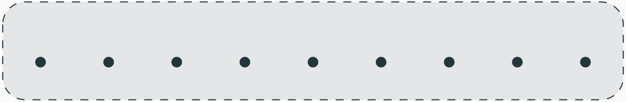
$$\mathsf{FO}^1 \subsetneq \mathsf{FO}^2 \subsetneq \mathsf{FO}^3 \subsetneq \mathsf{FO}^4 \subsetneq \cdots$$

$\exists x_1 . \exists x_2 . \exists x_3 . \exists x_4 . \ \bigwedge_{1 \le i < j \le 4} x_i \neq x_j$



- For some classes of models, the hierarchy collapses:

$\exists x . \exists y . \big(x < y \wedge \exists x . \, (y < x \wedge \exists y . \, x < y) \big)$

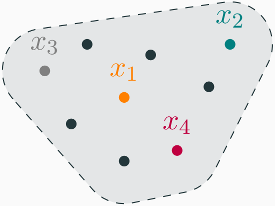


The k-variable property

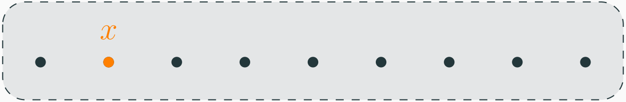
- Over arbitrary structures, strict hierarchy

$$\mathsf{FO}^1 \subsetneq \mathsf{FO}^2 \subsetneq \mathsf{FO}^3 \subsetneq \mathsf{FO}^4 \subsetneq \cdots$$

$$\exists x_1.\exists x_2.\exists x_3.\exists x_4.\ \bigwedge_{1\leq i<j\leq 4} x_i \neq x_j$$



- For some classes of models, the hierarchy collapses:
  $\exists x.\exists y.\big(x < y \wedge \exists x.\,(y < x \wedge \exists y.\, x < y)\big)$

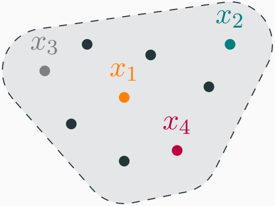


The k-variable property

- Over arbitrary structures, strict hierarchy

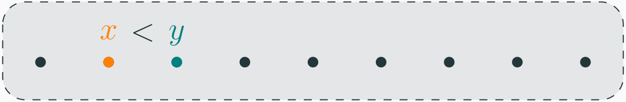
$$\mathsf{FO}^1 \subsetneq \mathsf{FO}^2 \subsetneq \mathsf{FO}^3 \subsetneq \mathsf{FO}^4 \subsetneq \cdots$$

$$\exists x_1.\exists x_2.\exists x_3.\exists x_4.\ \bigwedge_{1\le i<j\le 4} x_i \neq x_j$$



- For some classes of models, the hierarchy collapses:

$\exists x.\exists y.\big(x < y \wedge \exists x.\,(y < x \wedge \exists y.\, x < y)\big)$

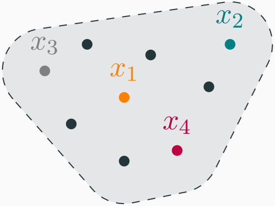


The k-variable property

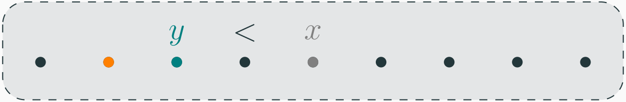
- Over arbitrary structures, strict hierarchy

$$\mathsf{FO}^1 \subsetneq \mathsf{FO}^2 \subsetneq \mathsf{FO}^3 \subsetneq \mathsf{FO}^4 \subsetneq \cdots$$

$\exists x_1.\exists x_2.\exists x_3.\exists x_4.\ \bigwedge_{1\leq i<j\leq 4} x_i \neq x_j$



- For some classes of models, the hierarchy collapses:
  $\exists x.\exists y.\big(x < y \wedge \exists x.\,(y < x \wedge \exists y.\, x < y)\big)$

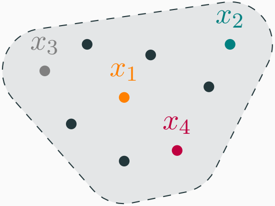


The k-variable property

- Over arbitrary structures, strict hierarchy

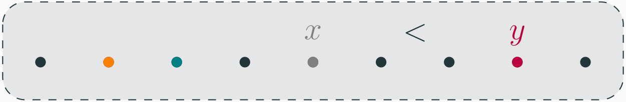
$$\mathsf{FO}^1 \subsetneq \mathsf{FO}^2 \subsetneq \mathsf{FO}^3 \subsetneq \mathsf{FO}^4 \subsetneq \cdots$$

$$\exists x_1.\exists x_2.\exists x_3.\exists x_4.\ \bigwedge_{1\leq i<j\leq 4} x_i \neq x_j$$



- For some classes of models, the hierarchy collapses:

$$\exists x.\exists y.\big(x < y \wedge \exists x.\,(y < x \wedge \exists y.\, x < y)\big)$$

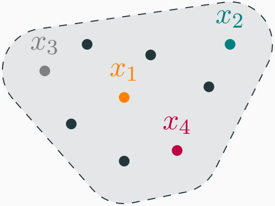


The k-variable property

- Over arbitrary structures, strict hierarchy

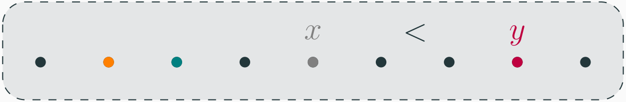
$$\mathsf{FO}^1 \subsetneq \mathsf{FO}^2 \subsetneq \mathsf{FO}^3 \subsetneq \mathsf{FO}^4 \subsetneq \cdots$$

$$\exists x_1.\exists x_2.\exists x_3.\exists x_4.\ \bigwedge_{1\leq i<j\leq 4} x_i \neq x_j$$



- For some classes of models, the hierarchy collapses:

$$\exists x.\exists y.\big(x < y \wedge \exists x.\,(y < x \wedge \exists y.\, x < y)\big)$$



Over linear orders, FO = FO^3.

Examples

$\boxed{\begin{array}{c}\text{Over words,}\\ \mathsf{FO} = \mathsf{FO}^3\\ \text{[Kamp'68]}\end{array}}$ ✓

$\boxed{\begin{array}{c}\text{Over linear orders,}\\ \mathsf{FO} = \mathsf{FO}^3\\ \text{[Immerman-Kozen'89]}\end{array}}$ ✓

Examples

Over words,
$$\mathsf{FO} = \mathsf{FO}^3$$
[Kamp'68]

✓

Over linear orders,
$$\mathsf{FO} = \mathsf{FO}^3$$
[Immerman-Kozen'89]

✓

What happens if we have additional binary relations?

Examples

Over words,

$\mathsf{FO} = \mathsf{FO}^3$

[Kamp'68]

✓

Over linear orders,

$\mathsf{FO} = \mathsf{FO}^3$

[Immerman-Kozen'89]

✓

What happens if we have additional binary relations?

Over ordered graphs,

$\forall k, \mathsf{FO} \neq \mathsf{FO}^k$

[Rossman'08]

✗

Examples

Over words,
FO $=$ FO^3
[Kamp'68] ✓

Over linear orders,
FO $=$ FO^3
[Immerman-Kozen'89] ✓

What happens if we have additional binary relations?

Over ordered graphs,
$\forall k,$ FO $\neq$ FO^k
[Rossman'08] ✗

Over MSCs,
FO $=$ FO^3
[Bollig-F.-Gastin'18] ✓

Examples

Over words,
$FO = FO^3$
[Kamp'68] ✓

Over linear orders,
$FO = FO^3$
[Immerman-Kozen'89] ✓

What happens if we have additional binary relations?

Over ordered graphs,
$\forall k, FO \neq FO^k$
[Rossman'08] ✗

Over MSCs,
$FO = FO^3$
[Bollig-F.-Gastin'18] ✓

Over Mazurkiewicz traces,
$FO = FO^3$
[Gastin-Mukund'02] ✓

Examples

Over words,
$\mathsf{FO} = \mathsf{FO}^3$
[Kamp'68]
✓

Over linear orders,
$\mathsf{FO} = \mathsf{FO}^3$
[Immerman-Kozen'89]
✓

What happens if we have additional binary relations?

Over ordered graphs,
$\forall k, \mathsf{FO} \neq \mathsf{FO}^k$
[Rossman'08]
✗

Over $(\mathbb{R}, <, +1)$,
$\mathsf{FO} = \mathsf{FO}^3$
[AHRW'15]
✓

Over MSCs,
$\mathsf{FO} = \mathsf{FO}^3$
[Bollig-F.-Gastin'18]
✓

Over Mazurkiewicz traces,
$\mathsf{FO} = \mathsf{FO}^3$
[Gastin-Mukund'02]
✓

Examples

Over words,
FO $=$ FO^3
[Kamp'68] ✓

Over linear orders,
FO $=$ FO^3
[Immerman-Kozen'89] ✓

What happens if we have additional binary relations?

Over ordered graphs,
$\forall k$, FO $\neq$ FO^k
[Rossman'08] ✗

Over $(\mathbb{R}, <, +1)$,
FO $=$ FO^3
[AHRW'15] ✓

Over MSCs,
FO $=$ FO^3
[Bollig-F.-Gastin'18] ✓

Over Mazurkiewicz traces,
FO $=$ FO^3
[Gastin-Mukund'02] ✓

What do the positive results have in common?

Generalisation

Theorem [F. 2019]

$\mathsf{FO} = \mathsf{PDL}_{\mathsf{sf}} = \mathsf{FO}^3$ over structures with

- one linear order $\leq$
- relations $R_1, R_2 \ldots$ defined by monotone partial functions
- arbitrary unary predicates $p, q, \ldots$

Applications

Generalisation

Theorem [F. 2019]

$\mathsf{FO} = \mathsf{PDL}_{\mathsf{sf}} = \mathsf{FO}^3$ over structures with

- one linear order $\leq$
- relations $R_1, R_2 \ldots$ defined by monotone partial functions
- arbitrary unary predicates $p, q, \ldots$

Applications

1. Linear orders: finite or infinite words, $\mathbb{R}$, $\mathbb{Q}$, ordinals...

Generalisation

Theorem [F. 2019]

$\mathsf{FO} = \mathsf{PDL_{sf}} = \mathsf{FO}^3$ over structures with

- one linear order $\leq$
- relations $R_1, R_2 \ldots$ defined by monotone partial functions
- arbitrary unary predicates $p, q, \ldots$

Applications

1. Linear orders: finite or infinite words, $\mathbb{R}$, $\mathbb{Q}$, ordinals...
2. $(\mathbb{R}, \leq, +1)$, $(\mathbb{R}, \leq, (+q)_{q \in \mathbb{Q}})$ ...

Generalisation

Theorem [F. 2019]

$\mathsf{FO} = \mathsf{PDL}_{\mathsf{sf}} = \mathsf{FO}^3$ over structures with

- one linear order $\leq$
- relations $R_1, R_2 \ldots$ defined by monotone partial functions
- arbitrary unary predicates $p, q, \ldots$

Applications

1. Linear orders: finite or infinite words, $\mathbb{R}$, $\mathbb{Q}$, ordinals...
2. $(\mathbb{R}, \leq, +1)$, $(\mathbb{R}, \leq, (+q)_{q\in\mathbb{Q}})$ ...
3. $(\mathbb{R}, \leq) +$ polynomial functions (new)

Generalisation

Theorem [F. 2019]

FO $=$ PDL$_{\mathsf{sf}}$ $=$ FO^3 over structures with

- one linear order $\leq$
- relations $R_1, R_2 \ldots$ defined by monotone partial functions
- arbitrary unary predicates $p, q, \ldots$

Applications

1. Linear orders: finite or infinite words, $\mathbb{R}$, $\mathbb{Q}$, ordinals...
2. $(\mathbb{R}, \leq, +1)$, $(\mathbb{R}, \leq, (+q)_{q \in \mathbb{Q}})$ $\ldots$
3. $(\mathbb{R}, \leq)$ $+$ polynomial functions (new)
4. MSCs (new)

Generalisation

Theorem [F. 2019]

$\mathsf{FO} = \mathsf{PDL}_{\mathsf{sf}} = \mathsf{FO}^3$ over structures with

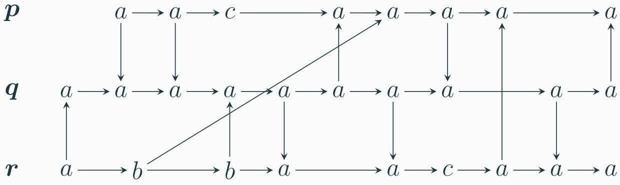
- one linear order $\leq$
- relations $R_1, R_2 \ldots$ defined by monotone partial functions
- arbitrary unary predicates $p, q, \ldots$

Applications

1. Linear orders: finite or infinite words, $\mathbb{R}$, $\mathbb{Q}$, ordinals...
2. $(\mathbb{R}, \leq, +1)$, $(\mathbb{R}, \leq, (+q)_{q\in\mathbb{Q}})$ ...
3. $(\mathbb{R}, \leq)$ + polynomial functions (new)
4. MSCs (new)
5. Mazurkiewicz traces, pomsets without auto-concurrency

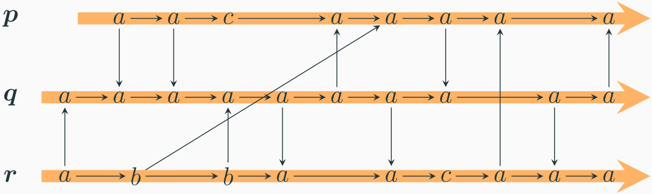
Application to MSCs

Application to MSCs



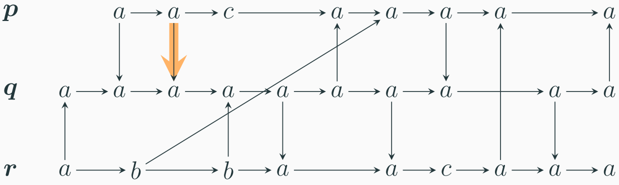
- Process order $\leq_{\mathsf{proc}}$
- Message relations $\lhd_{p,q}$

Application to MSCs



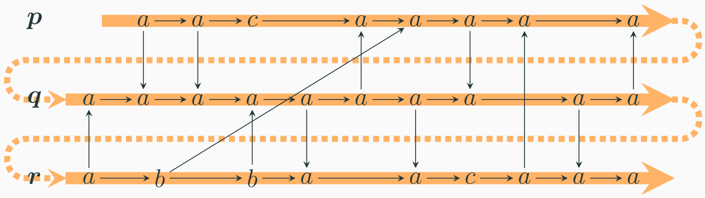
- Process order $\leq_{\mathsf{proc}}$
- Message relations $\lhd_{p,q}$

Application to MSCs



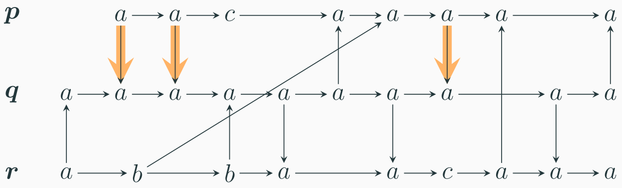
- Process order $\leq_{\mathsf{proc}}$
- Message relations $\lhd_{p,q}$

Application to MSCs



- Process order $\leq_{\text{proc}}$ can be extended to a linear order $\sqsubseteq$
- Message relations $\lhd_{p,q}$

Application to MSCs



- Process order $\leq_{\text{proc}}$ can be extended to a linear order $\sqsubseteq$
- Message relations $\lhd_{p,q}$ FIFO $\rightarrow$ monotone

Generalisation

Theorem [F. 2019]

$\mathsf{FO} = \mathsf{PDL}_{\mathsf{sf}} = \mathsf{FO}^3$ over structures with

- one linear order $\leq$
- relations $R_1, R_2 \dots$ defined by monotone partial functions
- arbitrary unary predicates $p, q, \dots$

Generalisation

Theorem [F. 2019]

$\mathsf{FO} = \mathsf{PDL}_{\mathsf{sf}} = \mathsf{FO}^3$ over structures with

- one linear order $\leq$
- interval-preserving binary relations $R_1, R_2, \ldots$
- arbitrary unary predicates $p, q, \ldots$

Generalisation

Theorem [F. 2019]

$\mathsf{FO} = \mathsf{PDL}_{\mathsf{sf}} = \mathsf{FO}^3$ over structures with

- one linear order $\leq$
- interval-preserving binary relations $R_1, R_2, \ldots$
- arbitrary unary predicates $p, q, \ldots$

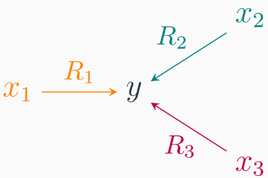
Applications

1. Linear orders with partial monotone functions (new)
2. Linear orders: finite or infinite words, $\mathbb{R}$, $\mathbb{Q}$, ordinals...
3. $(\mathbb{R}, \leq, +1)$, $(\mathbb{R}, \leq, (+q)_{q \in \mathbb{Q}})$ ...
4. $(\mathbb{R}, \leq)$ + polynomial functions (new)
5. MSCs (new)
6. Mazurkiewicz traces, pomsets without auto-concurrency

How does the interval-preserving assumption help?

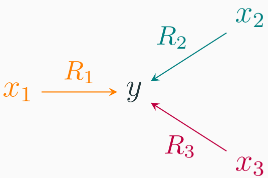
How does the interval-preserving assumption help?

$$\varphi(x_1, x_2, x_3) = \exists y.\, R_1(x_1, y) \wedge R_2(x_2, y) \wedge R_3(x_3, y)$$



How does the interval-preserving assumption help?

$$\varphi(x_1, x_2, x_3) = \exists y.\, R_1(x_1, y) \wedge R_2(x_2, y) \wedge R_3(x_3, y)$$



$\boxed{\text{Equivalent FO}^3\text{ formula?}}$

How does the interval-preserving assumption help?

$$\varphi(x_1, x_2, x_3) = \exists y.\, R_1(x_1, y) \wedge R_2(x_2, y) \wedge R_3(x_3, y)$$

Equivalent FO^3 formula?

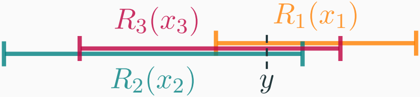
How does the interval-preserving assumption help?

$$\varphi(x_1, x_2, x_3) = \exists y.\, R_1(x_1, y) \wedge R_2(x_2, y) \wedge R_3(x_3, y)$$

Equivalent FO^3 formula?

How does the interval-preserving assumption help?

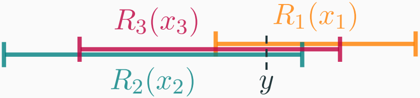
$$
\begin{aligned}
\varphi(x_1, x_2, x_3) &= \exists y.\, R_1(x_1, y) \wedge R_2(x_2, y) \wedge R_3(x_3, y) \\
&\equiv \Big(\exists y.\, R_1(x_1, y) \wedge R_2(x_2, y) \wedge \qquad\qquad \Big) \wedge \\
&\quad\; \Big(\exists y.\, R_1(x_1, y) \wedge R_3(x_3, y) \wedge \qquad\qquad \Big) \wedge \\
&\quad\; \Big(\exists y.\, R_2(x_2, y) \wedge R_2(x_3, y) \wedge \qquad\qquad \Big)
\end{aligned}
$$



$\boxed{\text{Equivalent FO}^3\text{ formula?}}$

How does the interval-preserving assumption help?

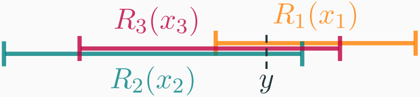
$$\begin{aligned}
\varphi(x_1, x_2, x_3) &= \exists y.\, R_1(x_1, y) \wedge R_2(x_2, y) \wedge R_3(x_3, y) \\
&\equiv \Big(\exists y.\, R_1(x_1, y) \wedge R_2(x_2, y) \wedge \exists x.\, R_3(x, y)\Big) \wedge \\
&\quad\; \Big(\exists y.\, R_1(x_1, y) \wedge R_3(x_3, y) \wedge \exists x.\, R_2(x, y)\Big) \wedge \\
&\quad\; \Big(\exists y.\, R_2(x_2, y) \wedge R_2(x_3, y) \wedge \exists x.\, R_1(x, y)\Big)
\end{aligned}$$



$\boxed{\text{Equivalent FO}^3\text{ formula?}}$

How does the interval-preserving assumption help?

$$\begin{aligned}
\varphi(x_1, x_2, x_3) &= \exists y.\, R_1(x_1, y) \land R_2(x_2, y) \land R_3(x_3, y) \\
&\equiv \Big(\exists x_3.\, R_1(x_1, x_3) \land R_2(x_2, x_3) \land \exists x_1.\, R_3(x_1, x_3)\Big) \land \\
&\quad\ \Big(\exists x_2.\, R_1(x_1, x_2) \land R_3(x_3, x_2) \land \exists x_1.\, R_2(x_1, x_2)\Big) \land \\
&\quad\ \Big(\exists x_1.\, R_2(x_2, x_1) \land R_2(x_3, x_1) \land \exists x_2.\, R_1(x_2, x_1)\Big)
\end{aligned}$$



Equivalent FO^3 formula?

Interval-preserving fragment of star-free PDL

Goal: inductive translation from FO to $\mathsf{PDL_{sf}} \equiv \mathsf{FO}^3$.

Interval-preserving fragment of star-free PDL

Goal: inductive translation from FO to $\mathsf{PDL}_{\mathsf{sf}} \equiv \mathsf{FO}^3$.

Invariant: use only interval-preserving relations

Interval-preserving fragment of star-free PDL

Goal: inductive translation from FO to $\mathrm{PDL}_{\mathrm{sf}} \equiv \mathrm{FO}^3$.

Invariant: use only interval-preserving relations

State formulas:

$$\varphi ::= P \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\pi\rangle\,\varphi$$

Path formulas:

$$\pi ::= \leq \mid R \mid \{\varphi\}? \mid \pi^{-1} \mid \pi \cdot \pi \mid \pi \cup \pi \mid \pi^{\mathsf{c}}$$

$\mathrm{PDL}_{\mathrm{sf}}$

$$\begin{aligned}\pi ::= {} & \leq \mid R \mid \{\varphi\}? \mid \pi^{-1} \mid \pi \cdot \pi \mid \pi \cap \pi \mid \\ & (\leq \cdot \pi \cdot \leq)^{\mathsf{c}} \mid (\leq \cdot \pi \cdot \geq)^{\mathsf{c}} \mid \\ & (\geq \cdot \pi \cdot \leq)^{\mathsf{c}} \mid (\geq \cdot \pi \cdot \geq)^{\mathsf{c}}\end{aligned}$$

$\mathrm{PDL}_{\mathrm{sf}}^{\mathrm{int}}$

Interval-preserving fragment of star-free PDL

Goal: inductive translation from FO to $\mathsf{PDL_{sf}} \equiv \mathsf{FO}^3$.

Invariant: use only interval-preserving relations

State formulas:

$$\varphi ::= P \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle \pi \rangle \varphi$$

Path formulas:

$$\pi ::= \leq \mid R \mid \{\varphi\}? \mid \pi^{-1} \mid \pi \cdot \pi \mid \pi \cup \pi \mid \pi^{\mathsf{c}}$$

$\mathsf{PDL_{sf}}$

$$\pi ::= \leq \mid R \mid \{\varphi\}? \mid \pi^{-1} \mid \pi \cdot \pi \mid \pi \cap \pi \mid (\leq \cdot \pi \cdot \leq)^{\mathsf{c}} \mid (\leq \cdot \pi \cdot \geq)^{\mathsf{c}} \mid (\geq \cdot \pi \cdot \leq)^{\mathsf{c}} \mid (\geq \cdot \pi \cdot \geq)^{\mathsf{c}}$$

$\mathsf{PDL_{sf}^{int}}$

Lemma: $\forall \pi \in \mathsf{PDL_{sf}^{int}}$, $[\![\pi]\!]$ is interval-preserving.

Equivalences over interval-preserving structures

$$\begin{tikzcd}
\mathsf{FO} \arrow[r] & \mathsf{PDL}_{\mathsf{sf}}^{\mathsf{int}} \arrow[d] \\
\mathsf{FO}^3 \arrow[u] & \mathsf{PDL}_{\mathsf{sf}} \arrow[l]
\end{tikzcd}$$

Equivalences over interval-preserving structures

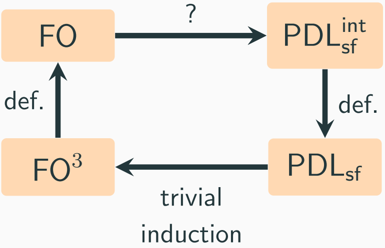
$$\begin{tikzcd}
\mathsf{FO} \arrow[r] & \mathsf{PDL}^{\mathsf{int}}_{\mathsf{sf}} \arrow[d, "\text{def.}"] \\
\mathsf{FO}^3 \arrow[u, "\text{def.}"] & \mathsf{PDL}_{\mathsf{sf}} \arrow[l]
\end{tikzcd}$$

Equivalences over interval-preserving structures

$$
\begin{tikzcd}
\mathsf{FO} \arrow[r] & \mathsf{PDL}_{\mathsf{sf}}^{\mathsf{int}} \arrow[d, "\text{def.}"] \\
\mathsf{FO}^3 \arrow[u, "\text{def.}"] & \mathsf{PDL}_{\mathsf{sf}} \arrow[l, "\substack{\text{trivial}\\ \text{induction}}"]
\end{tikzcd}
$$

- State formula $\varphi \in \mathsf{PDL}_{\mathsf{sf}} \quad \rightsquigarrow \quad \varphi^{\mathsf{FO}}(x) \in \mathsf{FO}$
- Path formula $\pi \in \mathsf{PDL}_{\mathsf{sf}} \quad \rightsquigarrow \quad \pi^{\mathsf{FO}}(x, y) \in \mathsf{FO}$

Equivalences over interval-preserving structures



Equivalences over interval-preserving structures

$$\mathsf{FO} \xrightarrow{?} \mathsf{PDL}^{\mathsf{int}}_{\mathsf{sf}}$$

(Induction) Any FO formula $\Phi(x_1, \ldots, x_n)$ is equivalent to a finite positive boolean combination of formulas of the form $\pi^{\mathsf{FO}}(x_i, x_j)$, where $\pi \in \mathsf{PDL}^{\mathsf{int}}_{\mathsf{sf}}$.

Equivalences over interval-preserving structures

$$\mathsf{FO} \xrightarrow{?} \mathsf{PDL}^{\mathsf{int}}_{\mathsf{sf}}$$

(Induction) Any FO formula $\Phi(x_1, \ldots, x_n)$ is equivalent to a finite positive boolean combination of formulas of the form $\pi^{\mathsf{FO}}(x_i, x_j)$, where $\pi \in \mathsf{PDL}^{\mathsf{int}}_{\mathsf{sf}}$.

- Atomic formulas, disjunction: easy

Equivalences over interval-preserving structures

$$\mathsf{FO} \xrightarrow{?} \mathsf{PDL}^{\mathsf{int}}_{\mathsf{sf}}$$

(Induction) Any FO formula $\Phi(x_1, \ldots, x_n)$ is equivalent to a finite positive boolean combination of formulas of the form $\pi^{\mathsf{FO}}(x_i, x_j)$, where $\pi \in \mathsf{PDL}^{\mathsf{int}}_{\mathsf{sf}}$.

- Negation: Express π^{c} using $(\leq \cdot \pi \cdot \leq)^{\mathsf{c}}$, $(\leq \cdot \pi \cdot \geq)^{\mathsf{c}}$, $(\geq \cdot \pi \cdot \leq)^{\mathsf{c}}$, $(\geq \cdot \pi \cdot \geq)^{\mathsf{c}}$.

Equivalences over interval-preserving structures

$$\mathsf{FO} \xrightarrow{?} \mathsf{PDL}^{\mathsf{int}}_{\mathsf{sf}}$$

(Induction) Any FO formula $\Phi(x_1, \ldots, x_n)$ is equivalent to a finite positive boolean combination of formulas of the form $\pi^{\mathsf{FO}}(x_i, x_j)$, where $\pi \in \mathsf{PDL}^{\mathsf{int}}_{\mathsf{sf}}$.

- Existential quantification: Similar to the example before.

Equivalences over interval-preserving structures

$$\mathsf{FO} \xrightarrow{?} \mathsf{PDL}^{\mathsf{int}}_{\mathsf{sf}}$$

(Induction) Any FO formula $\Phi(x_1, \ldots, x_n)$ is equivalent to a finite positive boolean combination of formulas of the form $\pi^{\mathsf{FO}}(x_i, x_j)$, where $\pi \in \mathsf{PDL}^{\mathsf{int}}_{\mathsf{sf}}$.

- Existential quantification: Similar to the example before.

  $\exists x. \bigwedge_i \pi_i^{\mathsf{FO}}(x_i, x)$

Equivalences over interval-preserving structures

$$\mathsf{FO} \xrightarrow{?} \mathsf{PDL}_{\mathsf{sf}}^{\mathsf{int}}$$

(Induction) Any FO formula $\Phi(x_1, \dots, x_n)$ is equivalent to a finite positive boolean combination of formulas of the form $\pi^{\mathsf{FO}}(x_i, x_j)$, where $\pi \in \mathsf{PDL}_{\mathsf{sf}}^{\mathsf{int}}$.

- Existential quantification: Similar to the example before.

$$\underbrace{\exists x. \bigwedge_i \pi_i^{\mathsf{FO}}(x_i, x)}_{\text{intersection of } n \text{ intervals}}$$

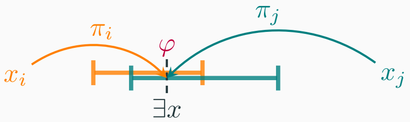
Equivalences over interval-preserving structures

$$\mathsf{FO} \xrightarrow{?} \mathsf{PDL}_{\mathsf{sf}}^{\mathsf{int}}$$

(Induction) Any FO formula $\Phi(x_1, \ldots, x_n)$ is equivalent to a finite positive boolean combination of formulas of the form $\pi^{\mathsf{FO}}(x_i, x_j)$, where $\pi \in \mathsf{PDL}_{\mathsf{sf}}^{\mathsf{int}}$.

- Existential quantification: Similar to the example before.

$$\underbrace{\exists x. \bigwedge_i \pi_i^{\mathsf{FO}}(x_i, x)}_{\text{intersection of } n \text{ intervals}}$$



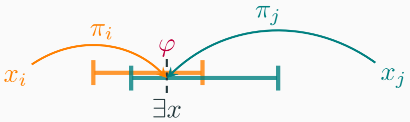
Equivalences over interval-preserving structures

$$\mathsf{FO} \xrightarrow{?} \mathsf{PDL}^{\mathsf{int}}_{\mathsf{sf}}$$

(Induction) Any FO formula $\Phi(x_1, \ldots, x_n)$ is equivalent to a finite positive boolean combination of formulas of the form $\pi^{\mathsf{FO}}(x_i, x_j)$, where $\pi \in \mathsf{PDL}^{\mathsf{int}}_{\mathsf{sf}}$.

- Existential quantification: Similar to the example before.

$$\underbrace{\exists x. \bigwedge_i \pi_i^{\mathsf{FO}}(x_i, x)}_{\text{intersection of } n \text{ intervals}} \equiv \underbrace{\bigwedge_{i,j} (\pi_i \cdot \{\varphi\}? \cdot \pi_j^{-1})^{\mathsf{FO}}(x_i, x_j)}_{\text{pairwise intersections}}$$



Conclusion

Contributions

Translation from first-order logic to CFMs.

Contributions

Translation from first-order logic to CFMs.

- A new Büchi theorem: $\mathsf{CFM} = \mathsf{EMSO}[\leq, \lhd]$, even with unbounded channels.

Contributions

Translation from first-order logic to CFMs.

- A new Büchi theorem: $\mathsf{CFM} = \mathsf{EMSO}[\leq, \lhd]$, even with unbounded channels.
- New proof that over existentially-bounded MSCs, $\mathsf{CFM} = \mathsf{MSO}[\leq, \lhd]$.

Contributions

Translation from first-order logic to CFMs.

- A new Büchi theorem: $\mathsf{CFM} = \mathsf{EMSO}[\leq, \lhd]$, even with unbounded channels.
- New proof that over existentially-bounded MSCs, $\mathsf{CFM} = \mathsf{MSO}[\leq, \lhd]$.

$\mathsf{FO} = \mathsf{FO}^3$ over interval-preserving structures.

Contributions

Translation from first-order logic to CFMs.

- A new Büchi theorem: CFM $=$ EMSO$[\leq, \lhd]$, even with unbounded channels.
- New proof that over existentially-bounded MSCs, CFM $=$ MSO$[\leq, \lhd]$.

FO $=$ FO^3 over interval-preserving structures.

- New, unifying proof of several known results, including linear orders, $(\mathbb{R}, <, +1)$, and Mazurkiewicz traces.

Contributions

Translation from first-order logic to CFMs.

- A new Büchi theorem: CFM $=$ EMSO$[\leq, \lhd]$, even with unbounded channels.
- New proof that over existentially-bounded MSCs, CFM $=$ MSO$[\leq, \lhd]$.

FO $=$ FO^3 over interval-preserving structures.

- New, unifying proof of several known results, including linear orders, $(\mathbb{R}, <, +1)$, and Mazurkiewicz traces.
- New applications: polynomial functions, linear orders with monotone functions, MSCs, ...

Contributions

Star-free PDL: an interesting specification language and a key technical tool.

Contributions

Star-free PDL: an interesting specification language and a key technical tool.

- Natural variant of PDL, equivalent to FO^3.

Contributions

Star-free PDL: an interesting specification language and a key technical tool.

- Natural variant of PDL, equivalent to FO^3.
- 2-dimensional temporal logic expressively complete over interval-preserving structures.

Contributions

Star-free PDL: an interesting specification language and a key technical tool.

- Natural variant of PDL, equivalent to FO^3.
- 2-dimensional temporal logic expressively complete over interval-preserving structures.
- (Fragments of) PDL_{sf} serve as intermediate steps to go
    - from FO to FO^3 over interval-preserving structures.
    - from FO to CFMs.

Contributions

Star-free PDL: an interesting specification language and a key technical tool.

- Natural variant of PDL, equivalent to FO^3.
- 2-dimensional temporal logic expressively complete over interval-preserving structures.
- (Fragments of) $\mathrm{PDL}_{\mathsf{sf}}$ serve as intermediate steps to go
    - from FO to FO^3 over interval-preserving structures.
    - from FO to CFMs.
- Over MSCs: expressively complete fragment closer to LTL and with a PSPACE translation into CFMs.

Perspectives

Continue a unified approach to expressivity problems: what makes a result work, and how can we extend it?

Perspectives

Continue a unified approach to expressivity problems: what makes a result work, and how can we extend it?

- Can we find a sufficient condition for the 3-variable property generalizing both interval-preserving structures and classes of trees for which similar results are known?

Perspectives

Continue a unified approach to expressivity problems: what makes a result work, and how can we extend it?

- Can we find a sufficient condition for the 3-variable property generalizing both interval-preserving structures and classes of trees for which similar results are known?
- What are necessary conditions for the 3-variable property?

Perspectives

Continue a unified approach to expressivity problems: what makes a result work, and how can we extend it?

- Can we find a sufficient condition for the 3-variable property generalizing both interval-preserving structures and classes of trees for which similar results are known?
- What are necessary conditions for the 3-variable property?
- Over which classes of structures are EMSO and automata equivalent?

Perspectives

Continue a unified approach to expressivity problems: what makes a result work, and how can we extend it?

- Can we find a sufficient condition for the 3-variable property generalizing both interval-preserving structures and classes of trees for which similar results are known?
- What are necessary conditions for the 3-variable property?
- Over which classes of structures are EMSO and automata equivalent?
- What are sufficient conditions for the existence of an expressively complete (1-dimensional) temporal logic?

Perspectives

Specific questions for MSCs.

Perspectives

Specific questions for MSCs.

- Can every formula of CPDL, with operations $(\cdot, *, +, ^{-1})$, be translated into an equivalent CFM?
  Known: YES with $(\cdot, \mathsf{c}, +, ^{-1})$ or $(\cdot, *, +)$, NO with $(\cdot, *, +, \cap, ^{-1})$
  [Bollig, Kuske, Meinecke 2010]

Perspectives

Specific questions for MSCs.

- Can every formula of CPDL, with operations $(\cdot, *, +, ^{-1})$, be translated into an equivalent CFM?
  Known: YES with $(\cdot, \mathsf{c}, +, ^{-1})$ or $(\cdot, *, +)$, NO with $(\cdot, *, +, \cap, ^{-1})$
  [Bollig, Kuske, Meinecke 2010]
- Is there a 1-dimensional temporal logic over MSC that is expressively complete for first-order logic?