$FO = FO^3$ for Linear Orders with Monotone Binary Relations

Marie Fortin

University of Liverpool

YR-OWLS, June 16, 2020
The $k$-variable property

How many variables are needed in first-order logic?
The $k$-variable property

How many variables are needed in first-order logic?

- Some properties require unboundedly many variables

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

... but not in every class of models:
The $k$-variable property

How many variables are needed in first-order logic?

- Some properties require unboundedly many variables:

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

- ... but not in every class of models:
The $k$-variable property

How many variables are needed in first-order logic?

- Some properties require unboundedly many variables

\[ \exists x_1. \exists x_2. \exists x_3. \exists x_4. \wedge_{1 \leq i < j \leq 4} x_i \neq x_j \]

- ... but not in every class of models:

\[ \exists x. \exists y. (x < y \land \exists x. (y < x \land \exists y. x < y)) \]
The $k$-variable property

How many variables are needed in first-order logic?

- Some properties require unboundedly many variables

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

- ...but not in every class of models:

\[ \exists x. \exists y. (x < y \land \exists x. (y < x \land \exists y. x < y)) \]
The $k$-variable property

How many variables are needed in first-order logic?

- Some properties require unboundedly many variables

\[ \exists x_1. \exists x_2. \exists x_3. \exists x_4. \wedge_{1 \leq i < j \leq 4} x_i \neq x_j \]

- But not in every class of models:
\[ \exists x. \exists y. (x < y \land \exists x. (y < x \land \exists y. x < y)) \]
The $k$-variable property

How many variables are needed in first-order logic?

- Some properties require unboundedly many variables

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

- ... but not in every class of models:

$$\exists x. \exists y. (x < y \land \exists x. (y < x \land \exists y. x < y))$$
The $k$-variable property

How many variables are needed in first-order logic?

- Some properties require unboundedly many variables

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

- ... but not in every class of models:

\[ \exists x. \exists y. (x < y \land \exists x. (y < x \land \exists y. x < y)) \]
The $k$-variable property

How many variables are needed in first-order logic?

- Some properties require unboundedly many variables

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

- ... but not in every class of models:

$$\exists x. \exists y. (x < y \land \exists x. (y < x \land \exists y. x < y))$$

Over linear orders, $\text{FO} = \text{FO}^3$. 
Bounded variable logics

Why do we care about the number of variables?
Bounded variable logics

Why do we care about the number of variables?

- (Descriptive) complexity
Bounded variable logics

Why do we care about the number of variables?

- (Descriptive) complexity
- Temporal logics
Bounded variable logics

Why do we care about the number of variables?

- (Descriptive) complexity
- Temporal logics

[Gabbay 1981] In any class of time flows, TFAE:

- There exists an expressively complete finite set of FO-definable (multi-dimensional) temporal connectives
- There exists $k$ such that every first-order sentence is equivalent to one with at most $k$ variables
Example

Over linear orders, $\text{FO} = \text{FO}^3$. 
Example

Over **linear orders**, $\text{FO} = \text{FO}^3$.

Two classical techniques to prove $\text{FO} = \text{FO}^k$ (over a class $C$)
Example

Over **linear orders**, $\text{FO} = \text{FO}^3$.

Two classical techniques to prove $\text{FO} = \text{FO}^k$ (over a class $C$)

1. Corollary of expressive completeness of a temporal logic
Example

Over **linear orders**, \( \text{FO} = \text{FO}^3 \).

Two classical techniques to prove \( \text{FO} = \text{FO}^k \) (over a class \( C \))

1. Corollary of expressive completeness of a temporal logic

   **Example:** Over complete linear orders, 
   \[
   \text{FO}^3 \subseteq \text{FO} = \text{LTL} \subseteq \text{FO}^3
   \]
   [Kamp 1968]
Example

Over linear orders, \( FO = FO^3 \).

Two classical techniques to prove \( FO = FO^k \) (over a class \( C \))

1. Corollary of expressive completeness of a temporal logic

**Example:** Over complete linear orders,

\[
FO^3 \subseteq FO = LTL \subseteq FO^3
\]

[Kamp 1968]

Over (arbitrary) linear orders,

\[
FO^3 \subseteq FO = LTL \text{ with Stavi connectives} \subseteq FO^3
\]

[Gabbay, Hodkinson, Reynolds 1993]
Example

Over linear orders, $\text{FO} = \text{FO}^3$.

Two classical techniques to prove $\text{FO} = \text{FO}^k$ (over a class $C$)

1. Corollary of expressive completeness of a temporal logic
2. Ehrenfeucht-Fraïssé games with $k$ pebbles
Example

Over **linear orders**, $\mathrm{FO} = \mathrm{FO}^3$.

Two classical techniques to prove $\mathrm{FO} = \mathrm{FO}^k$ (over a class $C$)

1. Corollary of expressive completeness of a temporal logic
2. Ehrenfeucht-Fraïssé games with $k$ pebbles

**Example:** Over complete linear orders,

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

[Immerman, Kozen 1989]
Example

Over **linear orders**, \( \text{FO} = \text{FO}^3 \).

Two classical techniques to prove \( \text{FO} = \text{FO}^k \) (over a class \( C \))

1. Corollary of expressive completeness of a temporal logic
   0 or 1 free variables

2. Ehrenfeucht-Fraïssé games with \( k \) pebbles
   up to \( k \) free variables
Known results (non-exhaustive)

Over linear orders,

\[ \text{FO} = \text{FO}^3 \]

[Immerman-Kozen’89]
Known results (non-exhaustive)

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

What happens if we have additional binary relations?
Known results (non-exhaustive)

Over linear orders,
\[ \text{FO} = \text{FO}^3 \]
[Immerman-Kozen’89] ✓

What happens if we have additional binary relations?

Over ordered graphs,
\[ \forall k, \text{FO} \neq \text{FO}^k \]
[Rossman’08] ✗
Known results (non-exhaustive)

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

What happens if we have additional binary relations?

Over ordered graphs, 
$$\forall k, \text{FO} \neq \text{FO}^k$$
[Rossman’08]

Over $$(\mathbb{R}, <, +1)$$, 
$$\text{FO} = \text{FO}^3$$
[AHRW’15]
Known results (non-exhaustive)

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 Mazurkiewicz traces, \( FO = FO^3 \)
[Gastin-Mukund’02]

Over MSCs, \( FO = FO^3 \)
[Bollig-F.-Gastin’18]
Known results (non-exhaustive)

Over linear orders,
\[ \text{FO} = \text{FO}^3 \]
[Immerman-Kozen’89]

What happens if we have additional binary relations?

Over ordered graphs,
\[ \forall k, \text{FO} \neq \text{FO}^k \]
[Rossman’08] ×

Over (\(\mathbb{R}, <, +1\)),
\[ \text{FO} = \text{FO}^3 \]
[AHRW’15] √

Over Mazurkiewicz traces,
\[ \text{FO} = \text{FO}^3 \]
[Gastin-Mukund’02] √

Over MSCs,
\[ \text{FO} = \text{FO}^3 \]
[Bollig-F.-Gastin’18] √

What do these 4 positive results have in common?
Generalisation \[F.'19\]

\[ \text{FO} = \text{FO}^3 \] over structures with

- one linear order \( \leq \),
- “interval-preserving” binary relations \( R_1, R_2, \ldots \),
- arbitrary unary predicates \( p, q, \ldots \)
Generalisation [F.’19]

$\text{FO} = \text{FO}^3$ over structures with

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

$R$ is interval-preserving if for all intervals $I$,

- $R(I)$ is an interval of $(\text{Im}(R), \leq)$
- $R^{-1}(I)$ is an interval of $(\text{dom}(R), \leq)$
Generalisation [F.’19]

$\text{FO} = \text{FO}^3$ over structures with

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

$R$ is \textit{interval-preserving} if for all intervals $I$,

- $R(I)$ is an interval of $(\text{Im}(R), \leq)$
- $R^{-1}(I)$ is an interval of $(\text{dom}(R), \leq)$
Generalisation [F.’19]

$\text{FO} = \text{FO}^3$ over structures with

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

\[ R \text{ is interval-preserving if for all intervals } I, \]
- $R(I)$ is an interval of $(\text{Im}(R), \leq)$
- $R^{-1}(I)$ is an interval of $(\text{dom}(R), \leq)$
Generalisation [F.’19]

$\text{FO} = \text{FO}^3$ over structures with
- one linear order $\leq$,
- “interval-preserving” binary relations $R_1, R_2, \ldots$,
- arbitrary unary predicates $p, q, \ldots$

$R$ is interval-preserving if for all intervals $I$,
- $R(I)$ is an interval of $(\text{Im}(R), \leq)$
- $R^{-1}(I)$ is an interval of $(\text{dom}(R), \leq)$
Generalisation [F.’19]

\[ \text{FO} = \text{FO}^3 \] over structures with

- one linear order \( \leq \),
- “interval-preserving” binary relations \( R_1, R_2, \ldots \),
- arbitrary unary predicates \( p, q, \ldots \)

\[ R \] is interval-preserving if for all intervals \( I \),

- \( R(I) \) is an interval of \( \text{Im}(R), \leq \)
- \( R^{-1}(I) \) is an interval of \( \text{dom}(R), \leq \)
Generalisation \[F.'19\]

$$\text{FO} = \text{FO}^3$$ over structures with

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

\[R\] is interval-preserving if for all intervals $$I$$,

- $$R(I)$$ is an interval of $$(\text{Im}(R), \leq)$$
- $$R^{-1}(I)$$ is an interval of $$(\text{dom}(R), \leq)$$
A special case: monotone partial functions

Any relation $R$ corresponding to a monotone partial function is interval-preserving.
A special case: monotone partial functions

Any relation $R$ corresponding to a monotone partial function is interval-preserving.

- $R(I)$ is an interval of $(\text{Im}(R), \leq)$
A special case: monotone partial functions

Any relation $R$ corresponding to a monotone partial function is interval-preserving.

- $R(I)$ is an interval of $(\text{Im}(R), \leq)$

\[
\begin{align*}
I
\end{align*}
\]
A special case: monotone partial functions

Any relation $R$ corresponding to a monotone partial function is interval-preserving.

- $R(I)$ is an interval of $(\text{Im}(R), \leq)$
A special case: monotone partial functions

Any relation \( R \) corresponding to a monotone partial function is interval-preserving.

- \( R(I) \) is an interval of \((\text{Im}(R), \leq)\)
A special case: monotone partial functions

Any relation $R$ corresponding to a monotone partial function is interval-preserving.

- $R(I)$ is an interval of $(\text{Im}(R), \leq)$
A special case: monotone partial functions

Any relation $R$ corresponding to a monotone partial function is interval-preserving.

- $R(I)$ is an interval of $(\text{Im}(R), \leq)$
- $R^{-1}(I)$ is an interval of $(\text{dom}(R), \leq)$
A special case: monotone partial functions

Any relation $R$ corresponding to a monotone partial function is interval-preserving.

- $R(I)$ is an interval of $(\text{Im}(R), \leq)$
- $R^{-1}(I)$ is an interval of $(\text{dom}(R), \leq)$
A special case: monotone partial functions

Any relation $R$ corresponding to a monotone partial function is interval-preserving.

- $R(I)$ is an interval of $(\text{Im}(R), \leq)$
- $R^{-1}(I)$ is an interval of $(\text{dom}(R), \leq)$
A special case: monotone partial functions

Any relation $R$ corresponding to a monotone partial function is interval-preserving.

- $R(I)$ is an interval of $(\text{Im}(R), \leq)$
- $R^{-1}(I)$ is an interval of $(\text{dom}(R), \leq)$
Applications

$$FO = FO^3$$ over

1. Linear orders with partial non-decreasing or non-increasing functions (**new**).
Applications

\[ \text{FO} = \text{FO}^3 \text{ over} \]

1. Linear orders with partial non-decreasing or non-increasing functions \( \text{(new)} \)

2. Linear orders: finite or infinite words, \( \mathbb{R}, \mathbb{Q}, \) ordinals...
Applications

$FO = FO^3$ over

1. Linear orders with partial non-decreasing or non-increasing functions (new)
2. Linear orders: finite or infinite words, $\mathbb{R}$, $\mathbb{Q}$, ordinals...
3. $(\mathbb{R}, \leq, +1)$, $(\mathbb{R}, \leq, (+q)_{q \in \mathbb{Q}})$ …
Applications

$\text{FO} = \text{FO}^3$ over

1. Linear orders with partial non-decreasing or non-increasing 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)
Applications

5. Message sequence charts (MSCs)
5. Message sequence charts (MSCs)

\[ p \quad a \rightarrow a \rightarrow c \rightarrow a \rightarrow a \rightarrow a \rightarrow a \rightarrow a \rightarrow a \\
q \quad a \rightarrow a \rightarrow a \rightarrow a \rightarrow a \rightarrow a \rightarrow a \rightarrow a \rightarrow a \\
r \quad a \rightarrow b \rightarrow b \rightarrow a \rightarrow a \rightarrow c \rightarrow a \rightarrow a \rightarrow a \rightarrow a \]
Applications

5. Message sequence charts (MSCs)

 executions of message-passing systems
Applications

5. Message sequence charts (MSCs)

Executions of **message-passing systems**

- Fixed, finite set of processes
Applications

5. Message sequence charts (MSCs)

Executions of *message-passing systems*

- Fixed, finite set of processes
- Process order $\leq_{\text{proc}}$
Applications

5. Message sequence charts (MSCs)

Executions of message-passing systems

- Fixed, finite set of processes
- Process order $\leq_{\text{proc}}$
- Message relations $\triangleleft_{p,q}$
Applications

5. Message sequence charts (MSCs)

Executions of \textbf{message-passing systems}

- Fixed, finite set of processes
- Process order $\leq_{\text{proc}}$
- Message relations $\prec_{p,q}$
Applications

5. Message sequence charts (MSCs)

Executions of message-passing systems

- Fixed, finite set of processes
- Process order $\leq_{\text{proc}}$
- Message relations $\triangleleft_{p,q}$

Extended to a linear order
Applications

5. Message sequence charts (MSCs)

Executions of **message-passing systems**

- Fixed, finite set of processes
- Process order $\leq_{\text{proc}}$
- Message relations $\triangleleft_{p,q}$

Extended to a linear order $\text{FIFO} \rightarrow$ monotone
Applications

5. Message sequence charts (MSCs)

Executions of message-passing systems

- Fixed, finite set of processes
- Process order $\leq_{\text{proc}}$
- Message relations $\triangleleft_{p,q}$

Extended to a linear order
FIFO $\rightarrow$ monotone
$\rightarrow$ Interval-preserving structure
Applications

$FO = FO^3$ over structures with

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

1. Linear orders with partial non-decreasing or non-increasing 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
6. Mazurkiewicz traces
How does the interval-preserving assumption help?
How does the interval-preserving assumption help?

\[ \varphi(x_1, x_2, x_3) = \exists y. R_1(x_1, y) \land R_2(x_2, y) \land R_3(x_3, y) \]
How does the interval-preserving assumption help?

\[ \varphi(x_1, x_2, x_3) = \exists y. R_1(x_1, y) \land R_2(x_2, y) \land R_3(x_3, y) \]

Equivalent FO³ formula?
How does the interval-preserving assumption help?

\[ \varphi(x_1, x_2, x_3) = \exists y. R_1(x_1, y) \land R_2(x_2, y) \land R_3(x_3, y) \]
How does the interval-preserving assumption help?

\[ \varphi(x_1, x_2, x_3) = \exists y. R_1(x_1, y) \land R_2(x_2, y) \land R_3(x_3, y) \]

Equivalent $\mathsf{FO}^3$ formula?
How does the interval-preserving assumption help?

\[
\varphi(x_1, x_2, x_3) = \exists y. R_1(x_1, y) \land R_2(x_2, y) \land R_3(x_3, y)
\]

\[
\equiv \left( \exists y. R_1(x_1, y) \land R_2(x_2, y) \land \left( \exists y. R_1(x_1, y) \land R_3(x_3, y) \land \left( \exists y. R_2(x_2, y) \land R_2(x_3, y) \land \right) \right) \right) \land
\]

Equivalent FO³ formula?
How does the interval-preserving assumption help?

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

$$\equiv \left( \exists y. R_1(x_1, y) \land R_2(x_2, y) \land \exists x. R_3(x, y) \right) \land$$

$$\left( \exists y. R_1(x_1, y) \land R_3(x_3, y) \land \exists x. R_2(x, y) \right) \land$$

$$\left( \exists y. R_2(x_2, y) \land R_2(x_3, y) \land \exists x. R_1(x, y) \right)$$

Equivalent FO$^3$ formula?
How does the interval-preserving assumption help?

\[ \varphi(x_1, x_2, x_3) = \exists y. R_1(x_1, y) \land R_2(x_2, y) \land R_3(x_3, y) \]

\[ \equiv \left( \exists x_3. R_1(x_1, x_3) \land R_2(x_2, x_3) \land \exists x_1. R_3(x_1, x_3) \right) \land \]

\[ \left( \exists x_2. R_1(x_1, x_2) \land R_3(x_3, x_2) \land \exists x_1. R_2(x_1, x_2) \right) \land \]

\[ \left( \exists x_1. R_2(x_2, x_1) \land R_3(x_3, x_1) \land \exists x_2. R_1(x_2, x_1) \right) \]

Equivalent FO$^3$ formula?
The proof

\[ \text{FO} = \text{FO}^3 \text{ over structures with} \]

- one linear order \( \leq \),
- “interval-preserving” binary relations \( R_1, R_2, \ldots \),
- arbitrary unary predicates \( p, q, \ldots \)
The proof

$\text{FO} = \text{FO}^3$ over structures with

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

**Key idea:** Go through an intermediate language: Star-free Propositional Dynamic Logic.
Star-free Propositional Dynamic Logic

Examples
Star-free Propositional Dynamic Logic

Examples

\[\langle R \rangle q \lor \langle \leq \cdot R - 1 \rangle q \lor \langle \leq \cdot \{\langle R \rangle q\} \cdot \leq \rangle p \lor \langle R \cap \leq \rangle (p \land q)\]
Star-free Propositional Dynamic Logic

Examples

\[(p \land \neg q) \lor (q \land \neg p)\]

\[\langle R \rangle q \;
\[\langle \leq \cdot R \cdot \leq -1 \rangle q \]

\[\langle \leq \cdot \{ \langle R \rangle q \} \cdot \leq \rangle p \]

\[\langle R \cap \leq \rangle (p \land q)\]
Star-free Propositional Dynamic Logic

Examples

\[(p \land \neg q) \lor (q \land \neg p)\]  
\[\langle R \rangle q\]
Star-free Propositional Dynamic Logic

Examples

\[(p \land \neg q) \lor (q \land \neg p)\]  
\[\langle R \rangle q\]  
\[\langle \leq \cdot R^{-1} \rangle q\]
Star-free Propositional Dynamic Logic

Examples

\[(p \land \neg q) \lor (q \land \neg p)\]

\[\langle R \rangle q\]

\[\langle \leq \cdot R^{-1} \rangle q\]

\[\langle \leq \cdot \{\langle R \rangle q\} \? \cdot \leq \rangle p\]
Star-free Propositional Dynamic Logic

Examples

\[(p \land \neg q) \lor (q \land \neg p)\]

\[\langle R \rangle q\]

\[\langle \leq \cdot R^{-1} \rangle q\]

\[\langle \leq \cdot \{\langle R \rangle q\} \cdot \leq \rangle p\]

\[\langle R^c \cap \leq \rangle (p \land q)\]
Star-free Propositional Dynamic Logic

Examples

Over \((\mathbb{R}, <, \{+q \mid q \in \mathbb{Q}_+\})\),

\[
\varphi U_{(q,r)} \psi \equiv
\]

\[
\begin{array}{c}
  t \\
  \varnothing
\end{array}
\]

\[
\begin{array}{c}
  t + q \\
  \varphi
\end{array}
\]

\[
\begin{array}{c}
  t + r \\
  \psi
\end{array}
\]
Star-free Propositional Dynamic Logic

Examples

Over \((\mathbb{R}, <, \{+q \mid q \in \mathbb{Q}_+\})\),

\[\varphi \mathbf{U}_{(q,r)} \psi \equiv \langle (\cdot <) \cap (+r \cdot <^{-1}) \cap (\cdot \{\neg \varphi\}? \cdot <^c) \rangle \psi\]
Star-free Propositional Dynamic Logic

Syntax

**State formulas:**
\[ \varphi ::= P \mid \varphi \lor \varphi \mid \neg \varphi \mid \langle \pi \rangle \varphi \]

**Path formulas:**
\[ \pi ::= \leq \mid R \mid \{ \varphi \}? \mid \pi^{-1} \mid \pi \cdot \pi \mid \pi \cup \pi \mid \pi^c \]
Star-free Propositional Dynamic Logic

Syntax

State formulas:

\[ \varphi ::= P \mid \varphi \lor \varphi \mid \neg \varphi \mid \langle \pi \rangle \varphi \]

Path formulas:

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

Combines features from

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

Syntax

State formulas:
$$\varphi ::= P \mid \varphi \lor \varphi \mid \neg \varphi \mid \langle \pi \rangle \varphi$$

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

Combines features from

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

Theorem: [Tarski-Givant 1987 (calculus of relations)]
$\text{PDL}_{sf}$ and $\text{FO}^3$ are expressively equivalent
A fragment of Star-free PDL

State formulas:
\[ \varphi ::= P | \varphi \lor \varphi | \neg \varphi | \langle \pi \rangle \varphi \]

Path formulas:
\[ \pi ::= \leq | R | \{ \varphi \} ? | \pi^{-1} | \pi \cdot \pi | \pi \cup \pi | \pi_c \]

Lemma:
\[ \forall \pi \in \text{PDL}^{sf}, J \pi K \text{ is interval-preserving} \]
A fragment of Star-free PDL

State formulas:
\[ \varphi ::= P \mid \varphi \lor \varphi \mid \neg \varphi \mid \langle \pi \rangle \varphi \]

Path formulas:
\[ \pi ::= \leq \mid R \mid \{ \varphi \}? \mid \pi^{-1} \mid \pi \cdot \pi \mid \pi \cup \pi \mid \pi^c \]
A fragment of Star-free PDL

**State formulas:**
\[ \phi ::= P \mid \phi \lor \phi \mid \neg \phi \mid \langle \pi \rangle \phi \]

**Path formulas:**
\[ \pi ::= \leq \mid R \mid \{ \phi \}? \mid \pi^{-1} \mid \pi \cdot \pi \mid \pi \cup \pi \mid \pi^c \]

---

**PDL\textsubscript{sf}**

**PDL\textsubscript{sf}\textsuperscript{int}**

**Lemma:** \( \forall \pi \in \text{PDL}_{\text{sf}}^{\text{int}}, \llbracket \pi \rrbracket \) is interval-preserving
Equivalences over interval-preserving structures
Equivalences over interval-preserving structures
Equivalences over interval-preserving structures

- $\text{FO}$
- $\text{FO}^3$
- $\text{PDL}^\text{int}_{sf}$
- $\text{PDL}_{sf}$

Definitions:
- def.

Trivial induction:
- trivial induction
Equivalences over interval-preserving structures

- State formula $\varphi \in \text{PDL}_{sf} \rightsquigarrow \varphi^{\text{FO}}(x) \in \text{FO}$

- Path formula $\pi \in \text{PDL}_{sf} \rightsquigarrow \pi^{\text{FO}}(x, y) \in \text{FO}$
Equivalences over interval-preserving structures

- **State formula** $\varphi \in PDL_{sf} \iff \varphi^{FO}(x) \in FO$

  $\langle \pi \rangle \varphi \iff \exists y. \pi^{FO}(x, y) \land \varphi^{FO}(y)$

- **Path formula** $\pi \in PDL_{sf} \iff \pi^{FO}(x, y) \in FO$

  $\pi_1 \cdot \pi_2 \iff \exists z. \pi_1^{FO}(x, z) \land \pi_2^{FO}(z, y)$
Any FO formula $\Phi(x_1, \ldots, x_n)$ is equivalent to a finite positive boolean combination of formulas of the form $\pi^{FO}(x_i, x_j)$, where $\pi \in PDL_{sf}^{int}$. 
Any FO formula $\Phi(x_1, \ldots, x_n)$ is equivalent to a finite positive boolean combination of formulas of the form $\pi^{FO}(x_i, x_j)$, where $\pi \in PDL^{int}_{sf}$.

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

**Proof:** by induction on $\Phi$.

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

**Proof:** by induction on $\Phi$.

- **Negation:** Express $\pi^c$ using
  
  $$(\leq \cdot \pi \cdot \leq)^c, (\leq \cdot \pi \cdot \geq)^c, (\geq \cdot \pi \cdot \leq)^c, (\geq \cdot \pi \cdot \geq)^c.$$
Equivalences over interval-preserving structures

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

**Proof:** by induction on $\Phi$.

- **Existential quantification:** Similar to the example before.
Any FO formula $\Phi(x_1, \ldots, x_n)$ is equivalent to a finite positive boolean combination of formulas of the form $\pi^{FO}(x_i, x_j)$, where $\pi \in PDL_{sf}^{int}$.

**Proof:** by induction on $\Phi$.

- **Existential quantification:** Similar to the example before.

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

**Proof:** by induction on $\Phi$.

- **Existential quantification:** Similar to the example before.

$$\exists x. \bigwedge_i \pi^{\text{FO}}_i(x_i, x)$$

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

**Proof:** by induction on $\Phi$.

- **Existential quantification:** Similar to the example before.

\[
\exists x. \bigwedge_i \pi_i^{\text{FO}}(x_i, x)
\]

intersection of $n$ intervals
Equivalences over interval-preserving structures

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

**Proof:** by induction on $\Phi$.

- **Existential quantification:** Similar to the example before.
  $$\exists x. \bigwedge_i \pi_i^{\text{FO}}(x_i, x) \equiv \bigwedge_{i,j} (\pi_i \cdot \{\varphi\} \cdot \pi_j^{-1})^{\text{FO}}(x_i, x_j)$$

intersection of $n$ intervals

pairwise intersections
Conclusion

- Over linearly ordered structures with interval-preserving binary relations,

\[ \text{FO} = \text{PDL}_{sf} = \text{FO}^3 \]
Conclusion

▶ Over linearly ordered structures with interval-preserving binary relations,

\[
\text{FO} = \text{PDL}_{sf} = \text{FO}^3
\]

▶ Covers many classical classes of structures: linear orders, real-time signals, MSCs, . . .
Conclusion

- Over linearly ordered structures with interval-preserving binary relations,

\[ \text{FO} = \text{PDL}_{sf} = \text{FO}^3 \]

- Covers many classical classes of structures: linear orders, real-time signals, MSCs, ... 
- Star-free PDL is a useful technical tool, but also an interesting logic on its own.
Conclusion

- Over linearly ordered structures with interval-preserving binary relations,

\[ \text{FO} = \text{PDL}_{sf} = \text{FO}^3 \]

- Covers many classical classes of structures: linear orders, real-time signals, MSCs, . . .
- Star-free PDL is a useful technical tool, but also an interesting logic on its own.

Further directions:

- Generalizations to other types of orders (trees . . .), relations of arity \( \geq 2 \)?
Conclusion

- Over linearly ordered structures with interval-preserving binary relations,

\[ \text{FO} = \text{PDL}_{sf} = \text{FO}^3 \]

- Covers many classical classes of structures: linear orders, real-time signals, MSCs, . . .

- Star-free PDL is a useful technical tool, but also an interesting logic on its own.

Further directions:

- Generalizations to other types of orders (trees . . .), relations of arity > 2?

- Uniform approach for proving completeness of temporal logics?
Conclusion

- Over linearly ordered structures with interval-preserving binary relations,
  \[ \text{FO} = \text{PDL}_{sf} = \text{FO}^3 \]

- Covers many classical classes of structures: linear orders, real-time signals, MSCs, . . .
- Star-free PDL is a useful technical tool, but also an interesting logic on its own.

Further directions:
- Generalizations to other types of orders (trees. . . ), relations of arity \( \geq 2 \)?
- Uniform approach for proving completeness of temporal logics?
Conclusion

- Over linearly ordered structures with interval-preserving binary relations,

\[ \text{FO} = \text{PDL}_{sf} = \text{FO}^3 \]

- Covers many classical classes of structures: linear orders, real-time signals, MSCs, . . .
- Star-free PDL is a useful technical tool, but also an interesting logic on its own.

Further directions:
- Generalizations to other types of orders (trees . . .), relations of arity > 2?
- Uniform approach for proving completeness of temporal logics?

Thank you!