# First-Order Queries Inexpressible in Temporal Logic

David Toman

Department of Computer Science

University of Toronto

and

Damian Niwiński

Institute of Informatics

Warsaw University

# Temporal Databases

- Relational Database

  data domain $\langle D; =^D \rangle$

  database schema $r_1, \ldots, r_m$

- Temporal Domain $\langle T; <^T \rangle$

- (uniform) temporal databases

$$\mathcal{D} = \langle T, <^T; D, =^D; R_1^{\mathcal{D}}, \ldots, R_m^{\mathcal{D}} \rangle$$

Intended interpretation:

$$r_i(c_1, \ldots, c_k) \text{ holds at } t \text{ iff } \mathcal{D} \models R_i(t, c_1, \ldots, c_k)$$

# First-order Query Languages

- Two-sorted Relational Calculus (2-FOL):

$$
\begin{array}{lll}
M ::= & R_i & \text{– extended database schema} \\
& \left.\begin{array}{l} M \wedge M \\ \neg M \end{array}\right\} & \text{propositional connectives} \\
& \left.\begin{array}{l} x_i = x_j \\ \exists x_i . M \end{array}\right\} & \text{data variables} \\
& \left.\begin{array}{l} t_i < t_j \\ \exists t_i . M \end{array}\right\} & \text{temporal variables}
\end{array}
$$

- Example:
*personal ID of everyone whose salary has decreased*

$$
\begin{aligned}
\exists y_1, y_2 . y_1 < y_2 \wedge \exists t_1 ( & t_1 < t_0 \wedge \text{Salary}(t_1, x, y_1) \\
& \wedge \exists t_2 (t_2 < t_1 \wedge \text{Salary}(t_2, x, y_2)))
\end{aligned}
$$

# First-order Query Languages II

- Temporal Connectives (FOTL):

$$
\begin{array}{lll}
O ::= & X_i & \text{– placeholders} \\
& \left.\begin{array}{l} O \wedge O \\ \neg O \end{array}\right\} & \text{propositional connectives} \\
& \left.\begin{array}{l} t_i < t_j \\ \exists t_i . O \end{array}\right\} & \text{temporal variables}
\end{array}
$$

  Let $\Omega$ be a finite set of (names of) $O$-formulas, each of which has exactly one free variable $t_0$.

- Example of temporal connectives

$$X_1 \textbf{ since } X_2 = \exists t_2 < t_0 . X_2 \wedge \forall t_1 . t_2 < t_1 < t_0 \supset X_1$$

$$X_1 \textbf{ until } X_2 = \exists t_2 > t_0 . X_2 \wedge \forall t_1 . t_2 > t_1 > t_0 \supset X_1$$

# First-order Query Languages III

- First-order Temporal Logic (FOTL):

$$
\begin{array}{lll}
F ::= & r_i & \text{– database schema} \\
& F \wedge F & \multirow{2}{*}{\} propositional connectives} \\
& \neg F & \\
& x_i = x_j & \multirow{2}{*}{\} data variables} \\
& \exists x_i . F & \\
& \omega(F_1, \ldots, F_k) & \text{– temporal connectives}
\end{array}
$$

where $\omega \in \Omega$.

- Example:
*personal ID of everyone whose salary has decreased*

$$\exists y_1, y_2 . y_1 < y_2 \wedge \blacklozenge(\text{salary}(x, y_1) \wedge \blacklozenge \text{salary}(x, y_2))$$

# Semantics of FOTL

$$\text{Embed} : \text{FOTL} \longrightarrow \text{2-FOL}$$

$$\begin{aligned}
\text{Embed}(r_i(x_1, \ldots, x_{v_i})) &= R_i(t_0, x_1, \ldots, x_{v_i}) \\
\text{Embed}(F_1 \wedge F_2) &= \text{Embed}(F_1) \wedge \text{Embed}(F_2) \\
\text{Embed}(\neg F) &= \neg\, \text{Embed}(F) \\
\text{Embed}(x_i = x_j) &= x_i = x_j \\
\text{Embed}(\exists x. F) &= \exists x.\, \text{Embed}(F) \\
\text{Embed}(\omega(F_1, \ldots, F_k)) &=
\end{aligned}$$

$$\omega^*(\text{Embed}(F_1)[t_0/t_1], \ldots, \text{Embed}(F_k)[t_0/t_k])$$

# Kamp's Theorem

**Theorem [Kamp, 1968]**

Propositional TL(**since**, **until**)
is expressively equivalent to monadic FOL($<$)
over complete linear orders.

**Theorem [Kamp, 1971]**

FOTL(**since**, **until**)
is strictly contained in FOTL(**since**, **until**, **now**)
over dense linear orders without endpoints.

# Our Results

1. FOTL is strictly less expressive than 2-FOL
   for arbitrary finite set of connectives
   over dense linear order.

   $$\exists t_1 < t < t_2 . \forall x (P(t_1, x) \iff R(t_2, x))$$

2. Future FOTL is less expressive than FOTL
   for dense linear order with left endpoint

   $$\exists t > 0 . \forall x (P(0, x) \iff R(t, x))$$

3. Multi-dimensional first-order temporal logics form a proper hierarchy (w.r.t. their expressive power):

   $$k\text{-FOTL} \subset (k+1)\text{-FOTL}$$

4. FOTL = 2-FOL
   if the data domain $D$ has fixed size

# Consequences

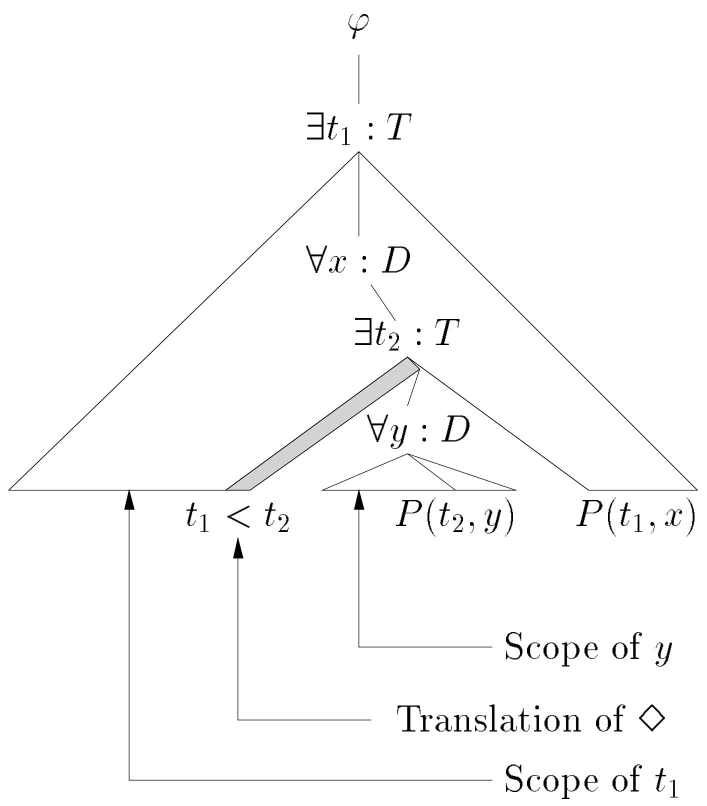
- There is no FO-complete relational algebra closed over uniform temporal databases.
    1. weaker query languages
    2. non-uniform temporal databases
    3. not closed query languages

- Nested queries can *not* be decomposed into a sequence of (not nested) views.

# How do we prove it

- we formulate a syntactic restriction on variables names that can occur together in the leaves of FOTL formulas,
- we modify the Ehrenfeucht-Fraïssé Game to capture exactly this restriction, and
- we show structures (temporal databases) that are distinguishable by a 2-FOL formula but equivalent w.r.t. the Ehrenfeucht-Fraïssé Game.

# Scopes of Variables in FOTL



# Ehrenfeucht-Fraïssé Game

- Game on structures $\mathcal{A}$ and $\mathcal{B}$
- Players Player I and Player IIplay $n$ moves

$$\begin{array}{ccccccccccc} t_1 & a_2 & a_3 & \ldots & t_i & \ldots & a_k & \ldots & t_j & \ldots & a_n \\ s_1 & b_2 & b_3 & \ldots & s_i & \ldots & b_k & \ldots & s_j & \ldots & b_n \end{array}$$

- Winning condition for Player II

$$(\forall R) \;\; R^{\mathcal{A}}(a_{i_1}, \ldots, a_{i_k}) \iff R^{\mathcal{B}}(b_{i_1}, \ldots, b_{i_k})$$

- Winning strategy for Player II defines $\mathcal{A} \sim_n \mathcal{B}$

**Theorem**

$$\mathcal{A}; a \sim_n \mathcal{B}; b \quad \text{iff} \quad \forall \varphi \in \mathcal{L}_n . \mathcal{A}; a \models \varphi \iff \mathcal{B}; b \models \varphi$$

**Method**

$\mathcal{A}_n, \mathcal{B}_n$ a sequence of pairs of structures such that:

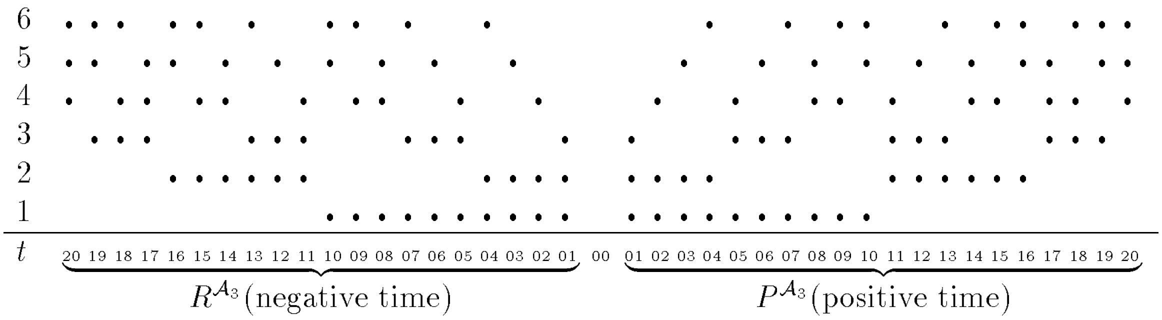
(1) $\mathcal{A}_n \in P$ and $\mathcal{B}_n \notin P$
(2) $\mathcal{A}_n \sim_n \mathcal{B}_n$

for all $n > 0$. Then $P$ is not definable.

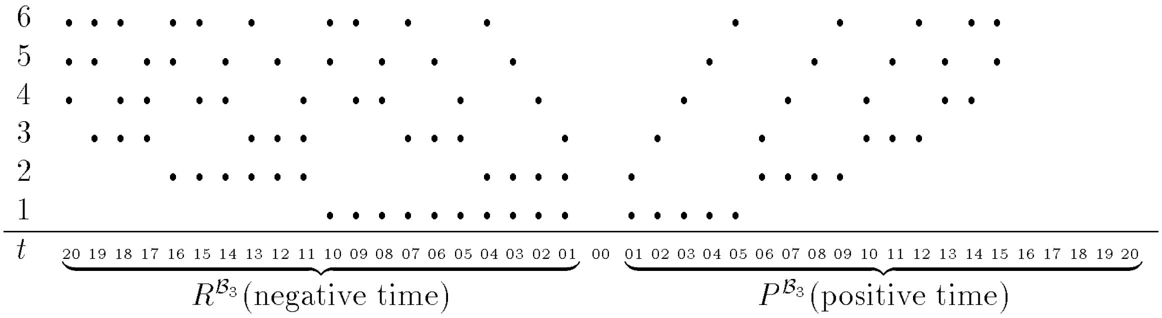
# Structures Indistinguishable by FOTL

Database schema $\langle p(x), r(x) \rangle$

$\mathcal{A}_3$:




$\mathcal{B}_3$:




(for an empty signature of the temporal domain)

# Conclusion and Related work

- Further research:

  Implementation techniques for FO-complete TQL

  Advantages of using TL (easier implementation)

  Relationship between sets of temporal connectives

  More complex structure of time

- Other proof methods:

  Counting/communication complexity [Abiteboul *et al.*]