# Point vs. Interval based Query Languages for Temporal Databases

David Toman

Department of Computer Science

University of Toronto

# Temporal Databases as FO structures

- Data Domain $D = \langle \mathrm{D}, = \rangle$

- Database schema $r_1, \ldots r_k$

- Instance of a schema $\mathbf{R}_1, \ldots, \mathbf{R}_k$

  where $\mathbf{R}_i \in \quad \mathrm{D}^{\mathrm{arity}(r_i)}$

- Database

  a structure $\quad D \cup \{\mathbf{R}_1, \ldots, \mathbf{R}_k\}$

- Semantics

  $\mathbf{x}$ are $r_i$-related $\quad$ iff $\quad \mathbf{x} \in \mathbf{R}_i$

- Temporal Domain $T_P = \langle \mathrm{T}, \leq \rangle$

$\mathrm{T}\times$

$T_P \cup$

at $t \in \mathrm{T}$ $\quad t,$

- Temporal Domain $T_I = \langle I(\mathrm{T}), \leq_{++}, \leq_{+-}, \leq_{-+}, \leq_{--} \rangle$
  where $I(\mathrm{T}) = \{(a, b) \in \mathrm{T} \times \mathrm{T} : a \leq b\}$
  and $[a, b] \leq_{++} [c, d] \iff b \leq d$

$I(\mathrm{T})\times$

$T_P\cup$

at $t \in \mathrm{T}$ $I$, and $t \in I$

# Semantics of Intervals

- Intervals as (encodings of) sets of points

```
king("Charles IV",  "Czech Kingdom", [1347, 1378])
king("Casimir III", "Poland",        [1333, 1370])
```

  * allows set operations UNION, INTERSECTION, ...
  * is equivalent to a single-dimensional time

- Intervals as entities

```
electricity("Jones A.", 40, 05/15/96, 06/15/96)
electricity("Smith J.", 35, 05/01/96, 06/01/96)
```

  * NO set operations
  * is equivalent to a two-dimensional time
  * created by *aggregation* over time periods

# First-order Temporal Query Languages

- Point-based Language $\mathbf{L_P}$

$$\begin{array}{rlll} L ::= & R_i(t, \mathbf{x}) & & \\ \mid & L \vee L & \mid & \neg L \\ \mid & \exists x.L & \mid & x_1 = x_2 \\ \mid & \exists t.L & \mid & t_1 \leq t_2 \end{array}$$

- Interval-based Language $\mathbf{L_I}$

$$\begin{array}{rlll} L ::= & R_i(I, \mathbf{x}) & & \\ \mid & L \vee L & \mid & \neg L \\ \mid & \exists x.L & \mid & x_1 = x_2 \\ \mid & \exists I.L & \mid & I_1^* \leq I_2^* \end{array}$$

  where $I_1^* \leq I_2^*$ stands for $I_1 \leq_{**} I_2$, $* \in \{+, -\}$.

- Tarskian semantics

- Queries have exactly one free temporal variable

- Allen's algebra based queries can be simulated by $\mathbf{L_I}$

# Examples

Salary: employee names and salaries. Query:

*Find names of everyone whose salary has decreased*:

$$\begin{aligned}\{t, x : \ & \exists t_1, t_2 . t_1 > t_2 \wedge \\ & \exists s_1, s_2 . s_1 < s_2 \wedge \\ & \text{Salary}(t_1, x, s_1) \wedge \text{Salary}(t_2, x, s_2)\}\end{aligned}$$

Easy enough to formulate in $\mathbf{L_I}$ (by hand):

$$\begin{aligned}\{[-\infty, \infty], x : \ & \exists I_1, I_2 . I_1^+ > I_2^- \wedge \\ & \exists s_1, s_2 . s_1 < s_2 \wedge \\ & \text{Salary}(I_1, x, s_1) \wedge \text{Salary}(I_2, x, s_2)\}\end{aligned}$$

# Examples (cont.)

Indep: names and periods of independence. Query:

*Find all periods when no country was independent*:

$$\{t : \neg\exists x.\, \text{Indep}(t, x)\}$$

Result in $\mathbf{L_I}$:

$$\begin{aligned}\{I :& \exists I'(\psi(I') \wedge I^+ = I'^- - 1 \vee I^+ = \infty) \wedge \\ & \exists I'(\psi(I') \wedge I^- = I'^+ + 1 \vee I^- = -\infty) \wedge \\ & \forall I'(\psi(I') \supset I \cap I' = \emptyset)\}\end{aligned}$$

where $\psi(I)$ is a shorthand for $\exists x.\, \text{Indep}(I, x)$.

# Mapping of $\mathbf{L_P}$ to $\mathbf{L_I}$

- Separate bindings of data variables $\mathbf{x}$ from bindings of temporal ones $\mathbf{t}$:

$$\varphi(\mathbf{t},\mathbf{x}) \;\mapsto\; \exists \mathbf{I}.\varphi^I(\mathbf{I},\mathbf{x}) \wedge \varphi^<(\mathbf{I},\mathbf{t})$$

- By structural induction:

$$
\begin{array}{lll}
F(\varphi) = \text{case } \varphi \text{ of} & & \\
\quad t_i < t_j: & \text{true} \wedge t_i < t_j & \\
\quad x_i = x_j: & x_i = x_j \wedge \text{true} & \\
\quad R(t,\mathbf{x}): & \exists I_t.R(I_t,\mathbf{x}) \wedge t \in I_t & \\
 & \qquad \text{for } R \text{ atomic and } I_t \text{ fresh} & \\
\quad \varphi_1 \vee \varphi_2: & \bigvee \exists \mathbf{I}_1,\mathbf{I}_2.\psi_i^j & \\
 & \qquad \text{for } F(\varphi_1) = \bigvee \exists \mathbf{I}.\psi_i^1 & \\
 & \qquad \text{and } F(\varphi_2) = \bigvee \exists \mathbf{I}.\psi_i^2 & \\
\quad \exists x.\varphi: & \bigvee \exists \mathbf{I}.(\exists x.\psi_i^I) \wedge \psi_i^< & \\
 & \qquad \text{for } F(\varphi) = \bigvee \exists \mathbf{I}.\psi_i^I \wedge \psi_i^< & \\
\quad \exists t.\varphi: & \bigvee \exists \mathbf{I}.\psi_i^I \wedge \mathrm{QE}_t(\psi_i^<) & \\
 & \qquad \text{for } F(\varphi) = \bigvee \exists \mathbf{I}.\psi_i^I \wedge \psi_i^< & \\
\quad \neg\varphi: & \mathrm{NOT}(F(\varphi)) &
\end{array}
$$

## Mapping of $\mathbf{L_P}$ to $\mathbf{L_I}$ (cont.)

- FO complementation for linear order

  In single dimension, for a fixed set of data constants $\mathbf{a}$:

  1. $D \models \varphi(t, \mathbf{a})$ for $t \in [a_i, b_i]$
  2. $L = \{-\infty, b_i + 1\}$, $H = \{a_i - 1, \infty\}$
  3. $R = \{[c_i, d_j] : \; c_i \in L, d_j \in H, c_i \leq d_j, \; \neg\exists [a_k, b_k].[c_i, d_j] \cap [a_k, b_k] \neq \emptyset$
  4. $D \models \neg\varphi(t, \mathbf{a})$ for $t \in I \in R$

- For dim $> 1$ we use hypercubes instead of intervals

- For dense domain we distinguish open/closed intervals

## Range restricted Queries

- Data variables:

  range restrictedness can be tested similarly to CALC

- Temporal variables:

  always range restricted
  (due to $\neg$-closure of the temporal domain)

- Can be tested before $F$ is applied

# Optimization of the Translation

The resulting formula can be exponential with respect to the source formula.

Two sources:

- Negation: repeated references to the same subquery

  $*$ recomputation avoided by materialized views

  $*$ result is bounded by a polynomial

- Distribution of conjunctions into disjunctions

  $*$ can NOT be avoided in general

  $*$ all CALC to RALG translators have similar problem

# Genericity and Encoding

- A single database can be encoded in different ways:

  $\mathbf{R}^{D_1} = \{([0,3], a)\}$
  $\mathbf{R}^{D_2} = \{([0,2], a), ([1,3], a)\}$

  both represent $\mathbf{R} = \{(0,a), (1,a), (2,a), (3,a)\}$.

- Canonical encoding: *flat* database

  smallest w.r.t. cardinality

- A *generic* query is invariant w.r.t. the encoding.

- Not all $\mathbf{L_I}$ queries are generic:

$$\exists I, J. \exists x (R(I,x) \wedge R(J,x) \wedge I \neq J)$$

- All generic queries can be expressed in $\mathbf{L_P}$.

# Other Encodings of Time

Intervals are quantifier free formulas in $TH(<_{\mathrm{Lin}})$ with exactly one free variable. Can we use different theories to encode sets of time instants?

- the theory must admit quantifier elimination and complementation.

- The QE and NOT have to be *value-independent*

  $$Z - [a, b] = [-\infty, a - 1] \cup [b + 1, \infty]$$

  (value independent)

  $$Z - \{a : a = 0(mod3)\} = \{a : a = 1(mod3)\} \cup \{a : a = 2(mod3)\}$$

  (not value independent)

# Conclusion

- Related work:
  - ∗ [BCST96] mapping FOTL to TSQL2
  - ∗ [VGT91], [AHV95] mapping CALC to RALG
- Future work:
  - ∗ Canonical encoding for dimension $> 1$
  - ∗ Complementation using non-FO features of $\mathbf{L_I}$
  - ∗ Multiple temporal dimensions in temporal database