# Logical Data Expiration

## *A Tutorial*

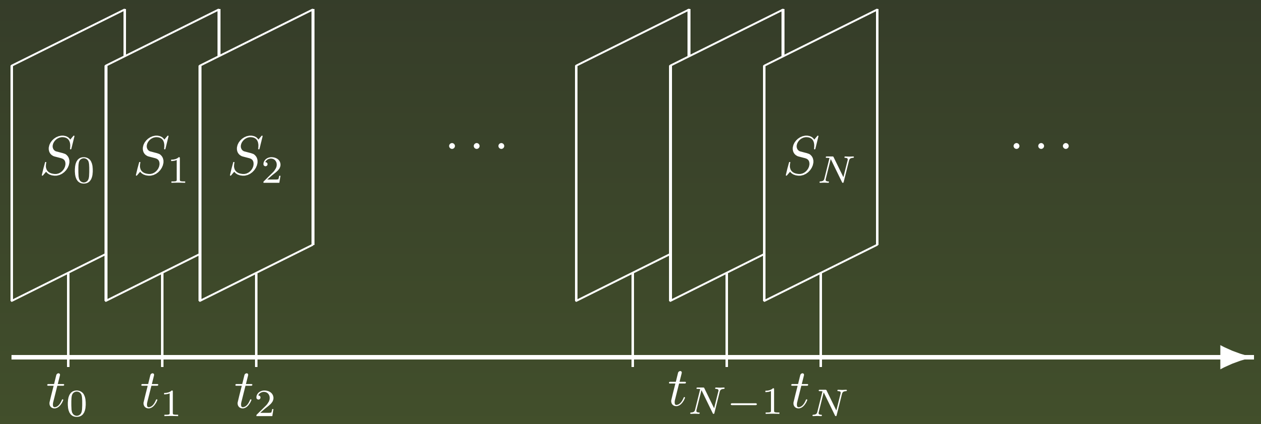
David Toman

david@uwaterloo.ca

School of Computer Science, University of Waterloo, Canada

# Data Evolution and Histories

Changes of data are captured (conceptually) by *histories*:




- states $S_i$ describe *system state*
- transitions $S_i \longrightarrow S_{i+1}$ represent system evolution

  $\Rightarrow$ append only histories (new states at the end)

# Data Access and Queries

Data is accessed using *queries*

- simple value look-ups vs. complex query languages
- current state only vs. access to *past states*
  - analysis of data warehouse evolution
  - enforcement of temporal integrity constraints
  - monitoring applications
  - . . .

# Expiration

**Question:** What data do we need to keep?

# Expiration

**Question:** What data do we need to keep?

1. *Policy*-driven expiration
2. *Query*-driven (logical) expiration

University of Waterloo

# Expiration

**Question:** What data do we need to keep?

1. *Policy*-driven expiration
2. *Query*-driven (logical) expiration

> Data to be expired is determined by
> **the (class of) queries** we are allowed to ask w.r.t. **all possible extensions** of a history

University of Waterloo

# Examples

- Record keeping/business rules:
  $\Rightarrow$ tax forms must be kept 5 years back
- Enforcing *dynamic integrity constraints:*
  $\Rightarrow$ don't hire anyone you've fired in the past
- Caching policy management:
  $\Rightarrow$ what data should be moved to backup storage?
- Moving window queries, etc. . .

University of Waterloo

# Outline of the Talk

- Temporal Database Primer
- Expiration Operators

*how good is an expiration operator?*

- Administrative Approaches to Expiration

$\Rightarrow$ materialized views and queries

- Query-driven Expiration

$\Rightarrow$ Temporal Logic and Materialized Views
$\Rightarrow$ First-order Queries and Partial Evaluation

*space limits for expiration operators*

- Infinite Extensions of Histories

$\Rightarrow$ Certain/Potential Answers

# TDB Primer

# Temporal Databases and Histories

*System states: Relational structures (fixed schema)*
*Time: discrete (integer-like)* $\{0, \ldots, N, \ldots\}$

1. Snapshot Temporal Database:

   $\Rightarrow$ time-indexed sequence of relational structures
   $\Rightarrow$ **append**-only: $H; D_{N+1}$

2. Timestamp Temporal Database:

   $\Rightarrow$ time-indexed tuples (using a *temporal attribute*)

Choices 1 and 2 equivalent [Chomicki and Toman, 1998]

University of Waterloo

# Example

Information about TA and courses by semester:

| Snapshot | | Timestamp |
|---|---|---|
| 0 | $\{(\text{John}, \text{CS448})\}$ | $\{ (0, \text{John}, \text{CS448}),$ |
| 1 | $\{(\text{John}, \text{CS448}),$ | $(1, \text{John}, \text{CS448}),$ |
| | $(\text{Sue}, \text{CS234})\}$ | $(1, \text{Sue}, \text{CS234}),$ |
| 2 | $\{(\text{John}, \text{CS448})\}$ | $(2, \text{John}, \text{CS448}),$ |
| 3 | $\{(\text{Sue}, \text{CS234})\}$ | $(3, \text{Sue}, \text{CS234}) \}$ |

# Temporal Queries

Queries: first-order formulas (over a fixed schema)

1. Temporal logic (FOTL)

   $\Rightarrow$ modal (temporal) connectives
   $\Rightarrow$ implicit references to time

2. Temporal Relational Calculus (2-FOL):

   $\Rightarrow$ temporal variables/attributes/quantifiers
   $\Rightarrow$ explicit access to time and ordering of time

**Proposition 1** *FOTL cannot express all 2-FOL queries. [Abiteboul et al., 1996, Toman and Niwinski, 1996, Toman, 2003b]*

University of Waterloo

# Examples

Students who TA'ed at least one class twice:

- in (past) FOTL:

$$\{x : \blacklozenge(\exists y.\mathrm{TA}(x,y) \wedge \bullet\blacklozenge\mathrm{TA}(x,y))\}$$

- in 2-FOL:

$$\{x : \exists t_1, t_2 . t_1 < t_2 \wedge \exists y.\mathrm{TA}(t_1,x,y) \wedge \mathrm{TA}(t_2,x,y)\}$$

University of Waterloo

# Examples

Students who TA'ed at least one class twice:

- in (past) FOTL:

$$\{x : \blacklozenge(\exists y.\mathrm{TA}(x,y) \wedge \bullet\blacklozenge\mathrm{TA}(x,y))\}$$

- in 2-FOL:

$$\{x : \exists t_1, t_2. t_1 < t_2 \wedge \exists y.\mathrm{TA}(t_1,x,y) \wedge \mathrm{TA}(t_2,x,y)\}$$

$$\exists t_1, t_2. t_1 < t_2 \wedge \forall x, y.\mathrm{TA}(t_1,x,y) \iff \mathrm{TA}(t_2,x,y)$$

cannot be expressed in FOTL

# Finite vs. Infinite Histories

Semantics of queries defined w.r.t:

- current (finite) history

  $\Rightarrow$ query evaluation on a finite temporal database

- a completion of current history

  $\Rightarrow$ hypothetical reasoning

# Data Expiration

# Expiration Operator

An *expiration operator* is a triple $(0^{\mathcal{E}}, \Delta^{\mathcal{E}}, Q^{\mathcal{E}})$ s.t.:

1. it provides an *inductive definition*

$$\begin{aligned} \mathcal{E}(\langle\ \rangle) &= 0^{\mathcal{E}} && \text{(initial state)} \\ \mathcal{E}(H; D) &= \Delta^{\mathcal{E}}(\mathcal{E}(H), D) && \text{(extension maintenance)} \end{aligned}$$

2. it maintains the following invariant:

$$Q(H) = Q^{\mathcal{E}}(\mathcal{E}(H)) \qquad \text{(answer preservation)}$$

University of Waterloo

# Examples

- the *identity* operator:

$$0^{\mathcal{E}_{\mathrm{id}}} = \langle\ \rangle \qquad Q^{\mathcal{E}_{\mathrm{id}}} = Q$$
$$\Delta^{\mathcal{E}_{\mathrm{id}}} = \lambda H \lambda S.H; S$$

- the *current* operator:

$$0^{\mathcal{E}_{\mathrm{now}}} = \langle\ \rangle \qquad Q^{\mathcal{E}_{\mathrm{now}}} = Q$$
$$\Delta^{\mathcal{E}_{\mathrm{now}}} = \lambda H \lambda S.\langle S \rangle$$

the queries *preserved* are *different*…

University of Waterloo

# Another Example

- (lossless) *compression* based operators:

$$
\begin{aligned}
0^{\mathcal{E}_{\text{compress}}} &= \text{compress}(\langle\ \rangle) \\
\Delta^{\mathcal{E}_{\text{compress}}} &= \lambda H \lambda S.\text{compress}(\text{decompress}(H); S) \\
Q^{\mathcal{E}_{\text{compress}}} &= \lambda H.Q(\text{decompress}(H))
\end{aligned}
$$

$\Rightarrow$ compress and decompress are *lossless*

… *no reduction;* $|H| \sim |\mathcal{E}_{\text{compress}}(H)|$

$\Rightarrow$ special case: *interval timestamps*.

University of Waterloo

# Expiration vs. Queries Revisited

- Given an *expiration operator*
  - for what class of queries it preserves answers?
    $\Rightarrow$ can these be characterized syntactically?
- Given a *set* of temporal queries:
  - is there an expiration operator that
    $\Rightarrow$ maintains answers to these queries?
    $\Rightarrow$ can be found algorithmically?
  - for what *query languages*?

University of Waterloo

# How Good is It ?

What is the space needed by $\mathcal{E}(H)$ in terms of

- size of the original history, $|H|$,
- length of $H$ (number of states, $|\mathbf{dom}_T|$),
- the size of the *active data domain* of $H$ (number of constants that have appeared in $H$, $|\mathbf{dom}_D|$),
- size of the answer $Q(H)$,
- size of the queries.

# How Good is It ?

What is the space needed by $\mathcal{E}(H)$ in terms of

- size of the original history, $|H|$,
- length of $H$ (number of states, $|\mathbf{dom}_T|$),
- the size of the *active data domain* of $H$ (number of constants that have appeared in $H$, $|\mathbf{dom}_D|$),
- size of the answer $Q(H)$,
- size of the queries.

**General Goal:** make $|\mathcal{E}(H)|$ independent of length of $H$. $\Rightarrow$ **bounded expiration operator**

University of Waterloo

# Example

**Proposition 2** $\mathcal{E}_{\text{now}}$ *is bounded.*

**Proposition 3** $\mathcal{E}_{\text{compress}}$ *cannot be bounded for lossless compression schemes.*

$\mathcal{E}_Q$ for a temporal query $Q$ in a language $\mathcal{L}$?

…depends on (expressive power) of $\mathcal{L}$

University of Waterloo

# Administrative Expiration Policies

# Administrative Approaches

*Query-independent expiration policies.*

- characterize queries whose answers are not affected

# Administrative Approaches

*Query-independent expiration policies.*

- characterize queries whose answers are not affected
  - expiration operator = a view of the history
    $\Rightarrow$ the view must be *self-maintainable*
  - query reformulation = query over the view
    $\Rightarrow$ *answering queries over views* problem

University of Waterloo

# Administrative Approaches

*Query-independent expiration policies.*

- characterize queries whose answers are not affected
  - expiration operator = a view of the history
    $\Rightarrow$ the view must be *self-maintainable*
  - query reformulation = query over the view
    $\Rightarrow$ *answering queries over views* problem
- detect attempts to access the *missing data*
  $\Rightarrow$ at run-time

# Cutoff Points

Common approach: *history truncation* or *cutoff point*

1. policies based on fixed *absolute cutoff point*, or
2. policies based on *now-relative cutoff point*.

$\Rightarrow$ generalization of the $\mathcal{E}^{\mathrm{id}}$ and the $\mathcal{E}^{\mathrm{now}}$ operators

University of Waterloo

# Cutoff Points

Common approach: *history truncation* or *cutoff point*

1. policies based on fixed *absolute cutoff point*, or
2. policies based on *now-relative cutoff point*.

$\Rightarrow$ generalization of the $\mathcal{E}^{\mathrm{id}}$ and the $\mathcal{E}^{\mathrm{now}}$ operators

**Example:** Vacuuming [Jensen, 1995]:

- $\rho(R) : e$ (a *remove* specification), and
- $\kappa(R) : e$ (a *keep* specification).

  $R$ is a temporal relation; $e$ a selection condition
- absolute/*now*-relative specifications

# A Query-driven Expiration: Finite Histories

# Query Driven Expiration

Expiration for queries in a temporal query language

- ⇒ Past FOTL (and variants)
- ⇒ Future FOTL
- ⇒ 2-FOL (temporal relational calculus)

# Query Driven Expiration

Expiration for queries in a temporal query language

$\Rightarrow$ Past FOTL (and variants)
$\Rightarrow$ Future FOTL
$\Rightarrow$ 2-FOL (temporal relational calculus)

- *Finite relational structures can be completely characterized by first-order queries.*

  $\Rightarrow$ best expiration operator for a fixed query $Q$.

University of Waterloo

# Query Driven Expiration

Expiration for queries in a temporal query language

$\Rightarrow$ Past FOTL (and variants)
$\Rightarrow$ Future FOTL
$\Rightarrow$ 2-FOL (temporal relational calculus)

- *Finite relational structures can be completely characterized by first-order queries.*
  $\Rightarrow$ best expiration operator for a fixed query $Q$.
- *Optimal expiration operator cannot exist.*
  $\Rightarrow$ we look for a **bounded expiration operator**.

University of Waterloo

# Query Driven Approaches

1. **Removal of "old" states (expiration)**

   $\Rightarrow$ removes a **subset** of existing states
   $\Rightarrow$ no other changes (history $\longrightarrow$ history)

2. **Auxiliary (non-temporal) view maintenance**

   $\Rightarrow$ maintains **auxiliary** relations
   $\Rightarrow$ maps a history to *a single extended state*

3. **Specialization of queries**

   $\Rightarrow$ specialize a query w.r.t. the known prefix $H$.

University of Waterloo

# Past Temporal Logic

- Syntax: First-order logic past temporal operators

$$Q ::= R(\mathbf{x}) \mid F \mid Q \wedge Q \mid \neg Q \mid \exists x.Q \mid \bullet Q \mid Q \ \textbf{since} \ Q$$

- Semantics:

$$Q(H) = \{\theta : H, \theta, n \models Q\}$$

$\Rightarrow$ queries over unbounded past: $\{x : \blacklozenge R(x)\}$

# Unfolding and Materialized Views

- Crux of the approach:

$$Q_1 \textbf{ since } Q_2 \equiv Q_1 \wedge \bullet(Q_2 \vee (Q_1 \textbf{ since } Q_2))$$

  - *auxiliary views* for temporal subformulas
    $\Rightarrow$ only the “previous” state needed

University of Waterloo

# Unfolding and Materialized Views

- Crux of the approach:

$$Q_1 \textbf{ since } Q_2 \;\equiv\; Q_1 \wedge \bullet(Q_2 \vee (Q_1 \textbf{ since } Q_2))$$

  - *auxiliary views* for temporal subformulas
    $\Rightarrow$ only the "previous" state needed
  - recurrent definitions to maintain the views.

| $\alpha$ | $R^0_\alpha$ | $R^n_\alpha$ |
|---|---|---|
| $\bullet Q$ | false | $Q^{n-1}$ |
| $Q_1 \textbf{ since } Q_2$ | false | $Q^n_1 \wedge (Q^{n-1}_2 \vee R^{n-1}_\alpha)$ |

# Example

- Query: *Students that TA'ed at least one class twice.*

$$\{x : \blacklozenge(\exists y.\mathrm{TA}(x,y) \wedge \bullet\blacklozenge\mathrm{TA}(x,y))\}$$

- Temporal subqueries:

  $\alpha_1 = \blacklozenge\mathrm{TA}(x,y)$ and

  $\alpha_2 = \bullet\blacklozenge\mathrm{TA}(x,y)$ and

  $\alpha_3 = \blacklozenge\exists y.\mathrm{TA}(x,y) \wedge \bullet\blacklozenge\mathrm{TA}(x,y)$.

# Example (cont.)

Inductive maintenance of views:

| $R_{\alpha_1}(x, y)$ | $R_{\alpha_2}(x, y)$ | $R_{\alpha_3}(x)$ |
|---|---|---|
| $\{ (\text{John}, \text{CS448}) \}$ | $\{ \}$ | $\{ \}$ |
| $\{ (\text{John}, \text{CS448}), (\text{Sue}, \text{CS234}) \}$ | $\{ (\text{John}, \text{CS448}) \}$ | $\{ \text{John} \}$ |
| $\{ (\text{John}, \text{CS448}), (\text{Sue}, \text{CS234}) \}$ | $\{ (\text{John}, \text{CS448}), (\text{Sue}, \text{CS234}) \}$ | $\{ \text{John} \}$ |
| $\{ (\text{John}, \text{CS448}), (\text{Sue}, \text{CS234}) \}$ | $\{ (\text{John}, \text{CS448}), (\text{Sue}, \text{CS234}) \}$ | $\{ \text{John}, \text{Sue} \}$ |

# Space Utilization

- $Q = \blacklozenge(p(x_1) \wedge \ldots \wedge p(x_k))$
- $H = \langle\{a_1\}, \{a_2\}, \{a_3\}, \ldots, \{a_n\}\rangle$.

For $\alpha = \blacklozenge(p(x_1) \wedge \ldots \wedge p(x_k))$:

$$|R_\alpha| = (n-1)^k$$

... the same holds for every prefix of $H$.

Full details: [Chomicki, 1995],

$\Rightarrow$ subsumes approaches based on TRA [Yang and Widom, 1998, Yang and Widom, 2000].

# Adding Fixpoints

- Syntax:

$$Q ::= R(\mathbf{x}) \mid F \mid Q \wedge Q \mid \neg Q \mid \exists x.Q \mid \bullet Q \mid \mu X.Q.$$

- Unfolding a fixpoint: $\mu X.Q \equiv Q(\mu X.Q)$
- Inductive maintenance of auxiliary relation:

| $\alpha$ | $R^0_\alpha$ | $R^n_\alpha$ |
|---|---|---|
| $\bullet Q$ | false | $Q^{n-1}$ |

$\Rightarrow$ careful definition of $Q^{n-1}$.

Full details: [Toman, 2003a]

University of Waterloo

# Metric Temporal Logic

- access to *real time* time instants

  $\Rightarrow$ a $\mathrm{clk}$ constant in each state (current *real* time)
  $\Rightarrow$ not part of the active data domain

- additional temporal operators

$$Q ::= \ldots \mid \textbf{since}_{\sim c} \mid \bullet_{\sim c}$$

  $\Rightarrow$ semantics respects $\sim c$ distances

- materialized views now contain *distance* values
  $\Rightarrow$ bounded by $c$
  $\Rightarrow$ bounded expiration if $\mathrm{clk}^i - \mathrm{clk}^{i-1} \geq \epsilon > 0$

# Future Temporal Logic

- Syntax:
  $Q ::= R(\mathbf{x}) \mid F \mid Q \wedge Q \mid \neg Q \mid \exists x.Q \mid \circ Q \mid Q \textbf{ until } Q$
- Semantics: $Q(H) = \{\theta : H, \theta, 0 \models Q\}$

  $\Rightarrow$ *still active domain semantics*

# Future Temporal Logic

- Syntax:
  $Q ::= R(\mathbf{x}) \mid F \mid Q \wedge Q \mid \neg Q \mid \exists x.Q \mid \circ Q \mid Q \textbf{ until } Q$
- Semantics: $Q(H) = \{\theta : H, \theta, 0 \models Q\}$

  $\Rightarrow$ *still active domain semantics*
- Unfolding rule (similarly to PastTL):

$$Q_1 \textbf{ until } Q_2 \equiv Q_1 \wedge (\circ Q_2 \vee \circ(Q_1 \textbf{ until } Q_2))$$

$\Rightarrow$ now we need to represent a formula with *holes* to be substituted when the history is extended.

University of Waterloo

# Biquantified Formulas

- [Lipeck and Saake, 1987, Lipeck et al., 1994]
  $\Rightarrow$ restrictions to Future FOTL syntax: 3 layers
  1. FO formulas (evaluated in a *state*),
  2. TL(FO) formulas: temporal outside (1),
  3. Universal quantifiers on top of (2)
- Automata-based approach
  $\Rightarrow$ designed in the *propositional* setting
  $\Rightarrow$ mix quantifiers and temporal connectives?

. . . bounded expiration based on an automaton for (2) implemented by *triggers*

University of Waterloo

# Two-sorted First-order Language

- Temporal Relational Calculus (2-FOL)

$$L ::= R(t, \mathbf{x}) | x = x' | t < t' | L \wedge L | L \wedge \neg L | L \vee L | \exists x.L | \exists t.L$$

for $R(t, \mathbf{x})$ true in $H$ iff $R(\mathbf{x})$ is true in $D_t$

A **bounded expiration operator** for 2-FOL?

$\Rightarrow$ conjectured that it does NOT exist

University of Waterloo

# Expiration Revisited

**Idea:** remove those states that

1. do not contribute to query answer (due to $\wedge$)
2. contribute duplicate information (due to $\exists$)

# Expiration Revisited

**Idea:** remove those states that

1. do not contribute to query answer (due to $\wedge$)
2. contribute duplicate information (due to $\exists$)

Easy for a fixed history:

$\Rightarrow$ compute answer to $Q$ bottom-up
$\Rightarrow$ propagate "back" to remove redundant data

**NOTE:** 2-FOL queries with *unbounded answers* cannot have bounded expiration operator

$\Rightarrow$ consider only *bounded queries*

# Handling History Extensions

Atomic formulas:

- $\left[{x \atop a}\right] \equiv \begin{cases} x = a & a \in \mathbf{dom}_D \\ \forall a \in \mathbf{dom}_D . x \neq a & a = \bullet \end{cases}$
- $\left[{t \atop s}\right] \equiv \begin{cases} t = s & s \in \mathbf{dom}_T \\ t > \mathrm{maxtime}(\mathbf{dom}_T) & s = \bullet \end{cases}$

Specialization of base relations *and their extensions*:

$$R(t, \mathbf{x}) \equiv \left( \bigvee_{\mathbf{a} \in R_{D_s}} \mathrm{true}\left[{t\mathbf{x} \atop s\mathbf{a}}\right] \right) \vee \left( \bigvee_{\mathbf{a} \in \mathbf{dom}_D \cup \{\bullet\}} R(t, \mathbf{x})\left[{t\mathbf{x} \atop \bullet\mathbf{a}}\right] \right)$$

$\Rightarrow$ depends **only** on the future extensions of history

University of Waterloo

# Query Specialization

$$
\mathsf{PE}_H(Q) = \begin{cases}
\{\text{true}\left[\begin{smallmatrix} t\mathbf{x} \\ s\mathbf{a} \end{smallmatrix}\right] : R(s,\mathbf{a}) \in D\} & \\
\quad \cup \{R(t,\mathbf{x})\left[\begin{smallmatrix} t\mathbf{x} \\ \bullet\mathbf{a} \end{smallmatrix}\right] : \mathbf{a} \in (\mathbf{dom}_D \cup \{\bullet\})^{|\mathbf{x}|}\} & Q \equiv R(t,\mathbf{x}) \\
\{Q'_1\left[\begin{smallmatrix} \mathbf{x} \\ \mathbf{a} \end{smallmatrix}\right] : Q'_1\left[\begin{smallmatrix} \mathbf{x} \\ \mathbf{a} \end{smallmatrix}\right] \in \mathsf{PE}_H(Q_1), \models \left[\begin{smallmatrix} \mathbf{x} \\ \mathbf{a} \end{smallmatrix}\right] \wedge F\} & Q \equiv Q_1 \wedge F \\
\{Q'_1 \wedge Q'_2\left[\begin{smallmatrix} \mathbf{xy} \\ \mathbf{ab} \end{smallmatrix}\right] : Q'_1\left[\begin{smallmatrix} \mathbf{x} \\ \mathbf{a} \end{smallmatrix}\right] \in \mathsf{PE}_H(Q_1), Q'_2\left[\begin{smallmatrix} \mathbf{y} \\ \mathbf{b} \end{smallmatrix}\right] \in \mathsf{PE}_H(Q_2), \models \left[\begin{smallmatrix} \mathbf{xy} \\ \mathbf{ab} \end{smallmatrix}\right]\} & Q \equiv Q_1 \wedge Q_2 \\
\{(\exists y. \bigvee_{Q'_1\left[\begin{smallmatrix} \mathbf{x}y \\ \mathbf{a}b \end{smallmatrix}\right] \in \mathsf{PE}_H(Q_1)} Q'_1)\left[\begin{smallmatrix} \mathbf{x} \\ \mathbf{a} \end{smallmatrix}\right] : \exists b. Q''_1\left[\begin{smallmatrix} \mathbf{x}y \\ \mathbf{a}b \end{smallmatrix}\right] \in \mathsf{PE}_H(Q_1)\} & Q \equiv \exists y. Q_1 \\
\{(\exists t. \bigvee_{Q'_1\left[\begin{smallmatrix} \mathbf{x}t \\ \mathbf{a}s \end{smallmatrix}\right] \in \mathsf{PE}_H(Q_1)} Q'_1)\left[\begin{smallmatrix} \mathbf{x} \\ \mathbf{a} \end{smallmatrix}\right] : \exists s. Q''_1\left[\begin{smallmatrix} \mathbf{x}t \\ \mathbf{a}s \end{smallmatrix}\right] \in \mathsf{PE}_H(Q_1)\} & Q \equiv \exists t. Q_1 \\
\{Q'_1 \wedge \neg Q'_2\left[\begin{smallmatrix} \mathbf{x} \\ \mathbf{a} \end{smallmatrix}\right] : Q'_1\left[\begin{smallmatrix} \mathbf{x} \\ \mathbf{a} \end{smallmatrix}\right] \in \mathsf{PE}_H(Q_1), Q'_2\left[\begin{smallmatrix} \mathbf{x} \\ \mathbf{a} \end{smallmatrix}\right] \in \mathsf{PE}_H(Q_2)\} & \\
\quad \cup \{Q'_1\left[\begin{smallmatrix} \mathbf{x} \\ \mathbf{a} \end{smallmatrix}\right] : Q'_1\left[\begin{smallmatrix} \mathbf{x} \\ \mathbf{a} \end{smallmatrix}\right] \in \mathsf{PE}_H(Q_1), Q'_2\left[\begin{smallmatrix} \mathbf{x} \\ \mathbf{a} \end{smallmatrix}\right] \not\in \mathsf{PE}_H(Q_2)\} & Q \equiv Q_1 \wedge \neg Q_2 \\
\{Q'_1 \vee Q'_2\left[\begin{smallmatrix} \mathbf{x} \\ \mathbf{a} \end{smallmatrix}\right] : Q'_1 \in \mathsf{PE}_H(Q_1)\left[\begin{smallmatrix} \mathbf{x} \\ \mathbf{a} \end{smallmatrix}\right], Q'_2\left[\begin{smallmatrix} \mathbf{x} \\ \mathbf{a} \end{smallmatrix}\right] \in \mathsf{PE}_H(Q_2)\} & \\
\quad \cup \{Q'_1\left[\begin{smallmatrix} \mathbf{x} \\ \mathbf{a} \end{smallmatrix}\right] : Q'_1\left[\begin{smallmatrix} \mathbf{x} \\ \mathbf{a} \end{smallmatrix}\right] \in \mathsf{PE}_H(Q_1), Q'_2\left[\begin{smallmatrix} \mathbf{x} \\ \mathbf{a} \end{smallmatrix}\right] \not\in \mathsf{PE}_H(Q_2)\} & \\
\quad \cup \{Q'_2\left[\begin{smallmatrix} \mathbf{x} \\ \mathbf{a} \end{smallmatrix}\right] : Q'_1\left[\begin{smallmatrix} \mathbf{x} \\ \mathbf{a} \end{smallmatrix}\right] \not\in \mathsf{PE}_H(Q_1), Q'_2\left[\begin{smallmatrix} \mathbf{x} \\ \mathbf{a} \end{smallmatrix}\right] \in \mathsf{PE}_H(Q_2)\} & Q \equiv Q_1 \vee Q_2
\end{cases}
$$

University of Waterloo

# Duplicate Information Removal

**IDEA:** Modify the $\mathsf{PE}_H$ for quantification over time:

$$(\exists t. \bigvee_{\substack{Q_1'[\begin{smallmatrix}\mathbf{x}t\\ \mathbf{a}s\end{smallmatrix}] \in \mathsf{PE}_H(Q_1) \\ s \in \mathsf{TB}_{\mathbf{a}}(t)}} Q_1')[\begin{smallmatrix}\mathbf{x}\\ \mathbf{a}\end{smallmatrix}]$$

where $Q_1''[\begin{smallmatrix}\mathbf{x}t\\ \mathbf{a}s\end{smallmatrix}] \in \mathsf{PE}_H(Q_1)$ for some $s$

# Duplicate Information Removal

**IDEA:** Modify the $\mathsf{PE}_H$ for quantification over time:

$$(\exists t. \bigvee_{\substack{Q'_1[^{\mathbf{x}t}_{\mathbf{a}s}] \in \mathsf{PE}_H(Q_1) \\ s \in \mathsf{TB}_{\mathbf{a}}(t)}} Q'_1)[^{\mathbf{x}}_{\mathbf{a}}]$$

$\longleftarrow$ what is this??

where $Q''_1[^{\mathbf{x}t}_{\mathbf{a}s}] \in \mathsf{PE}_H(Q_1)$ for some $s$

# Equivalence w.r.t. History Extensions

**Definition 1:** Let $Q_1[{}^{\mathbf{x}t}_{\mathbf{a}s_1}], Q_2[{}^{\mathbf{x}t}_{\mathbf{a}s_2}] \in \mathsf{PE}_H(Q)$ for $s_1 \neq s_2$.

We define $[{}^{\mathbf{x}t}_{\mathbf{a}s_1}] \sim^H_Q [{}^{\mathbf{x}t}_{\mathbf{a}s_2}]$ iff for any extension $H'$ of $H$

$$(\mathbf{a}, s_1) \in Q(H; H') \iff (\mathbf{a}, s_2) \in Q(H; H')$$

**Definition 2:** $\mathsf{TB}_{\mathbf{a}}(t)$ is the set of representatives of the $[{}^{\mathbf{x}t}_{\mathbf{a}s_1}] \sim^H_Q [{}^{\mathbf{x}t}_{\mathbf{a}s_2}]$ equivalence classes [e.g., min in $<$].

University of Waterloo

# Residual History Reconstruction

- *Specialization-based* expiration:

$$\begin{aligned} Q(H) &= \mathsf{PE}_H(Q)(\emptyset) \\ \mathsf{PE}_{H;H'}(Q) &\equiv \mathsf{PE}_{H'}(\mathsf{PE}_H(Q)) \end{aligned}$$

- We use $\mathsf{PE}_H(Q)$ to *construct* $\mathcal{E}(H)$

$\Rightarrow$ temporal variable $t_i \longrightarrow$ a unary relation

$$T_i(t) = \bigcup_{\mathbf{a}} \mathsf{TB}_{\mathbf{a}}(t).$$

$\Rightarrow$ each quantifier $\exists t_i.Q'$ in $Q$ is restricted to $T_i(t)$

$\Rightarrow$ a state $D_j \in H$ expires if $j \notin \bigcup_i T_i$.

# Properties

- $Q(H; H') = Q(\mathcal{E}_Q(H); H')$
  for all $H, H'$ histories and $Q$ FO query
- $|\,\mathcal{E}_Q(H)| \leq f(|\mathbf{dom}_D|, |Q|)$,
  $f$ is an exponential tower in number of nested $\neg\exists$
- $|\,\mathcal{E}_Q(H)| \leq |H| + |\mathbf{dom}_T||Q|$

…and can be implemented by FO queries/updates.

University of Waterloo

# Space: Lower Bounds

**Example:** $\exists t_1, t_2 . t_1 < t_2 \land \forall x . R(t_1, x) \iff R(t_2, x)$

- Potentially we need to keep all states for which $R$ contains distinct subsets of $\mathbf{dom}_D$

  $\Rightarrow$ potentially all subsets of $\mathbf{dom}_D$
  $\Rightarrow$ any residual history is exponential in $|\mathbf{dom}_D|$.
- *sequences of states* yield more exponents.
- non-elementary blowup even when translating monadic FO to propositional TL

# General Lower Bounds

# Limits of Bounded Encoding

Clearly, this cannot work for all possible queries:

**Example 1:** Query $\{t : R(t)\}$.

$$\text{answer} \sim |\mathbf{dom}_T H|$$

**Example 2:** Query $\{t : R(t) \wedge \forall t'.R(t') \rightarrow t \geq t'\}$.

$$\text{answer} \sim \log(|\mathbf{dom}_T H|)$$

University of Waterloo

# Counting

**Example:** "is the number of states containing $a$ greater that the number of states containing $b$?"

$\Rightarrow$ we need $\Omega(\log(|\mathbf{dom}_T|))$ space for counter(s)

$\Rightarrow \mathcal{O}(\log(|\mathbf{dom}_T|))$ is sufficient.

**Conjecture:** we can use the above technique (but remember counts of the expired values) to answer queries with counting

$\Rightarrow |\,\mathcal{E}_Q(H)| \leq POLY(\log(|\mathbf{dom}_T|))$

# Duplicates

**Example** (in SQL-style syntax):

```
( select '1'
  from   R
  where  R.x='a' ) except all ( select '1'
                                from   R
                                where  R.x='b' )
```

… is nonempty iff

the number of states containing $a$ is greater that the number of states containing $b$.

$\Rightarrow$ just like counting …

# Retroactive Updates

**Example:**

**while** $\exists t.R(t,a) \wedge \exists t.R(t,b)$ **do** { while both $a$ and $b$ exist in $R$ }

  **delete** $R(t,a)$

    **where** $\forall t'.R(t',a) \supset t' > t$; { delete (chronologically) first $a$ }

  **delete** $R(t,b)$

    **where** $\forall t'.R(t',b) \supset t' > t$; { delete (chronologically) first $b$ }

**return** $\exists t.R(t,a)$ { return true if $R$ contains an $a$ }

$\Rightarrow$ we need $\Omega(\log(|\mathbf{dom}_T|))$ space for counter(s)
$\Rightarrow$ just like for counting …

# Full Future$\mu$TL

No bounded operator can exist:

$\Rightarrow$ [Toman, 2003a] shows $\Omega(|\mathbf{dom}_T|)$ lower bound

$$\varphi = \exists x, y. \diamond(Q(x, y) \land \mu X. R(x, y) \lor \circ \exists z. X(x, z) \land X(z, y))$$

| State | Database instance |
|---|---|
| $0$ | $\{\}$ or $\{Q(a, b)\}$ |
| $\vdots$ | $\vdots$ |
| $n$ | $\{\}$ or $\{Q(a, b)\}$ |
| $n+1$ | $\{R(a, c_1), \ldots, R(c_i, c_{i+1}), \ldots, R(c_k, b)\}$ |

# Certain and Potential Answers

# Infinite Histories

**Definition 4** *Let $H$ be a finite history, $Q$ a query (in an appropriate query language), and $\theta$ a substitution.*

- *$\theta$ is a potential answer for $Q$ with respect to $H$ if there is an infinite completion $H'$ of $H$ such that $H', \theta \models Q$.*
- *$\theta$ is a certain answer for $Q$ with respect to $H$ if for all infinite completions $H'$ of $H$ we have $H', \theta \models Q$.*

*potential answer*: a direct generalization of of *potential constraint satisfaction* [Chomicki, 1995].

University of Waterloo

# Infinite Histories (cont.)

**Proposition 5 ([Gabbay et al., 1994])** *Satisfaction for two dimensional propositional temporal logic over natural numbers-based time domain is not decidable.*

**Proposition 6 ([Chomicki, 1995])** *For past formulas potential constraint satisfaction is undecidable.*

**Proposition 7 ([Chomicki and Niwinski, 1995])**
*For biquantified formulas*
$\Rightarrow$ *no internal quantifiers: EXPTIME,*
$\Rightarrow$ *a single internal quantifier: undecidable.*

…data expiration is a moot point

University of Waterloo

# Related Problems

- garbage collection in programming languages

  $\Rightarrow$ *navigational* query language (ala IMS) based on *reachability queries*

- data streams and streaming queries

| | | |
|---|---|---|
| data stream | = | history |
| synopse(is) | = | residual history |
| streaming query | = | temporal query |
| standing query | = | fixed query |
| | ... | |

University of Waterloo

# Open Problems

**FutureTL**

$\Rightarrow$ expiration operator for full FOETL

**Rich Temporal Domains** (more than linear $\leq$)

$\Rightarrow$ constraint DB techniques [Libkin et al., 2000]

**Space Bounds For Aggregate Queries**

$\Rightarrow$ a weaker bound, e.g., $|\mathcal{E}H| \in O(\log(|\mathbf{dom}_T H|)$?

**Decidable Certain/Potential Answers**

$\Rightarrow$ Decidable languages (monodic TL)

$\Rightarrow$ Optimal Expiration Operators?

University of Waterloo

# Acknowledgment

- A chapter in *Logics for Emerging Applications of Databases*, J. Chomicki, G. Saake, and R. van der Mayden (eds.), Springer 2003.

  (`http://db.uwaterloo.ca/~david/book-lead.ps`)

- Part of this research was done while at BRICS

  Centre for Basic Research in Computer Science

  funded by the Danish National Science Foundation.

- The research was supported by the National Sciences and Engineering Research Council of Canada (NSERC).

University of Waterloo

# Advertisment

Call For Papers

The 12th International Symposium on
TEMPORAL REPRESENTATION AND REASONING
(TIME 2005)

Burlington, Vermont, USA, June 23-25, 2005

URL: `http://time2005.cse.buffalo.edu/`

Papar submission: 11 pages on January 22, 2005.

# References


[Abiteboul et al., 1996] Abiteboul, S., Herr, L., and Van den Bussche, J. (1996). Temporal Versus First-Order Logic to Query Temporal Databases. In *ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 49–57.

[Chomicki, 1995] Chomicki, J. (1995). Efficient Checking of Temporal Integrity Constraints Using Bounded History Encoding. *TODS*, 20(2):149–186.

[Chomicki and Niwinski, 1995] Chomicki, J. and Niwinski, D. (1995). On the Feasibility of Checking Temporal Integrity Constraints. *Journal of Computer and System Sciences*, 51(3):523–535.

[Chomicki and Toman, 1998] Chomicki, J. and Toman, D. (1998). Temporal Logic in Information Systems. In Chomicki, J. and Saake, G., editors, *Logics for Databases and Information Systems*, pages 31–70. Kluwer.

[Gabbay et al., 1994] Gabbay, D. M., Hodkinson, I. M., and Reynolds, M. (1994). *Temporal Logic: Mathematical Foundations and Computational Aspects*. Oxford University Press.

[Jensen, 1995] Jensen, C. S. (1995). Vacuuming. In Snodgrass, R. T., editor, *The TSQL2 Temporal Query Language*, pages 447–460.

[Libkin et al., 2000] Libkin, L., Kuper, G., and Paredaens, J., editors (2000). *Constraint Databases*. Springer.

[Lipeck et al., 1994] Lipeck, U. W., Gertz, M., and Saake, G. (1994). Transitional Monitoring of Dynamic Integrity Constraints. *IEEE Data Engineering Bulletin*.

[Lipeck and Saake, 1987] Lipeck, U. W. and Saake, G. (1987). Monitoring Dynamic Integrity Constraints Based on Temporal Logic. *Information Systems*, 12(3):255–269.

[Toman, 2003a] Toman, D. (2003a). Logical Data Expiration for Fixpoint Extensions of Temporal Logics. In *International Symposium on Spatial and Temporal Databases*, page to appear. Springer LNCS.

[Toman, 2003b] Toman, D. (2003b). On Incompleteness of Multi-dimensional First-order Temporal Logics. In *International Symposium on Temporal Representation and Reasoning*, page to appear. IEEE Press.

[Toman and Niwinski, 1996] Toman, D. and Niwinski, D. (1996). First-Order Queries over Temporal Databases Inexpressible in Temporal Logic. In *Advances in Database Technology, EDBT'96*, volume 1057, pages 307–324. Springer.

[Yang and Widom, 1998] Yang, J. and Widom, J. (1998). Maintaining Temporal Views over Non-Temporal Information Sources for Data Warehousing. In *Advances in Database Technology, EDBT'98*, pages 389–403.

[Yang and Widom, 2000] Yang, J. and Widom, J. (2000). Temporal View Self-Maintenance. In *Advances in Database Technology, EDBT'00*, pages 395–412.