On formal semantics of STSQL

Tomáš Richta

Department of Intelligent Systems
Faculty of Information Technology
Brno University of Technology

2.12.2009

Outline

1. Motivation
2. E3VPC
   - The structure of E3VPC Expressions
   - Interpretation of an E3VPC Expression
   - SQL to E3VPC Translation
3. An Algebra of Spatio-Temporal Predicates
   - Spatial and temporal data types and operations
   - Spatial Predicates
   - Temporally Lifted Spatial Predicates
4. Querying Developments in STSQL

Motivation

- decision made based on spatial realtionship recorded within some time domain
- spatio-temporal data storage and retrieval operations
- many SQL extensions
- ... the question is - which is the right one?

```
SELECT f.id
FROM FLIGHT f, WEATHER w
WHERE f.Route Disjoint >> meet >> Inside w.Extent
```

Solution

- formal semantics of SQL[1]
- an algebra of spatio-temporal predicates[2]
- formal semantics of STSQL

Formal semantics of SQL

- Negri defines semantics of SQL as mapping from the set of input tables to one output table
- technically, he translates SQL query to E3VPC expression
- then he transforms E3VPC expression to Cannonical Form, which could be manipulated as predicate expression
- the query result is set of n-tuples, which satisfy defined predicate

Outline

1. Motivation
2. E3VPC
    - The structure of E3VPC Expressions
    - Interpretation of an E3VPC Expression
    - SQL to E3VPC Translation
3. An Algebra of Spatio-Temporal Predicates
    - Spatial and temporal data types and operations
    - Spatial Predicates
    - Temporally Lifted Spatial Predicates
4. Querying Developments in STSQL

E3VPC Expression

Definition

An E3VPC expression has the structure

$$\{t(v_1, \ldots, v_n) : \| \; P(v_1, \ldots, v_n) \; \|^{\alpha}\}$$

where

- $v_1, \ldots, v_n$ are tuple variables
- $t(v_1, \ldots, v_n)$ is the target list of the expression
- $P(v_1, \ldots, v_n)$ is predicate formula
- $\| \; \ldots \; \|^{\alpha}$ is the interpretation operator, where α assume one of the two values, T (true) and F (false)

Outline

1. Motivation
2. E3VPC
   - The structure of E3VPC Expressions
   - Interpretation of an E3VPC Expression
   - SQL to E3VPC Translation
3. An Algebra of Spatio-Temporal Predicates
   - Spatial and temporal data types and operations
   - Spatial Predicates
   - Temporally Lifted Spatial Predicates
4. Querying Developments in STSQL

Interpretation of an E3VPC Expression

The meaning of an expression

$$\{xinR : P(x)\}$$

is undefined, because it is not defined whether a tuple x', such that $P(x') = U$ belongs to it or not.

Interpretation operator

Definition

Interpretation operator. The interpretation of unknowns is defined as follows: let $P(x)$ be a three-valued predicate formula and $Q(x)$ be a two-valued predicate formula, then

1. $Q(x)$ is a true-interpreted two-valued equivalent of $P(x)$ if

$$
\begin{aligned}
P(x) = T &\Rightarrow Q(x) = T \\
P(x) = F &\Rightarrow Q(x) = F \\
P(x) = U &\Rightarrow Q(x) = T
\end{aligned}
$$

The notation $\parallel P(x) \parallel^T$ means a predicate that is a true-interpreted two-valued equivalent of $P(x)$.

Interpretation operator

Definition

2. $Q(x)$ is a false-interpreted two-valued equivalent of $P(x)$ if

$$\begin{aligned} P(x) = T &\Rightarrow Q(x) = T \\ P(x) = F &\Rightarrow Q(x) = F \\ P(x) = U &\Rightarrow Q(x) = F \end{aligned}$$

The notation $\| \ P(x) \ \|^F$ means a predicate which is a false-interpreted two-valued equivalent of $P(x)$.

Outline

1. Motivation
2. E3VPC
   - The structure of E3VPC Expressions
   - Interpretation of an E3VPC Expression
   - SQL to E3VPC Translation
3. An Algebra of Spatio-Temporal Predicates
   - Spatial and temporal data types and operations
   - Spatial Predicates
   - Temporally Lifted Spatial Predicates
4. Querying Developments in STSQL

General structure of the translation

$$\{TR\langle FRCLAUSE\rangle : \| TR\langle WHCLAUSE\rangle \wedge TR\langle HCLAUSE\rangle \|^{F}\}$$

Example of translation

DEPT(d#, nofemp, location, manager)
EMP(e#,d#, residence)

```
SELECT d.manager
FROM DEPT d
WHERE d.location = ALL
  SELECT e.residence
  FROM EMP e
  WHERE e.d# = d.d#
GROUP BY d.manager
HAVING AVG(d.nofemp) > 500
```

Example of translation - step (a)

$S12 \quad \langle FRCLAUSE \rangle ::= FROM \langle TABLE\ REFERENCE \rangle (.\langle TABLE\ REFERENCE \rangle)$

$S13 \quad \langle TABLE\ REFERENCE \rangle ::= \langle table\ name \rangle \langle correlation\ name \rangle$

$S14 \quad \langle WHCLAUSE \rangle ::= WHERE \langle WHERE\ SEARCH\ COND \rangle$

$S16 \quad \langle HCLAUSE \rangle ::= HAVING \langle HAVING\ SEARCH\ COND \rangle$

$T1 \quad \{ TR \langle FRCLAUSE \rangle : \| TR \langle WHCLAUSE \rangle \wedge TR \langle HCLAUSE \rangle \|^{F} \}$

$T13 \quad \langle correlation\ name \rangle in \langle table\ name \rangle$

$(a) \quad \{ d\ in\ DEPT : \| TR \langle WHERE\ SEARCH\ COND \rangle \wedge TR \langle HAVING\ SEARCH\ COND \rangle \|^{F} \}$

Example of translation - step (b)

$S17$ $\langle WHERE\ SEARCH\ COND\rangle ::= Boolean\ expression\ of \langle WPRED\rangle$

$S18$ $\langle WPRED\rangle ::= \langle SIMPLE\ PRED\rangle | \langle COMPLEX\ PRED\rangle$

$S20$ $\langle COMPLEX\ PRED\rangle ::= \langle SOME\ QUANTIFIED\ PRED\rangle | \langle SOME\ QUANTIFIED\ AFPRED\rangle |$
$\langle ALL\ QUANTIFIED\ PRED\rangle | \langle ALL\ QUANTIFIED\ AFPRED\rangle | \langle COMPLEX\ IN\ PRED\rangle |$
$\langle COMPLEX\ IN\ AFPRED\rangle | \langle COMPLEX\ NOT\ IN\ PRED\rangle | \langle COMPLEX\ NOT\ IN\ AFPRED\rangle |$
$\langle EXISTS\ PRED\rangle | \langle COMPLEX\ COMP\ PRED\rangle | \langle COMPLEX\ COMP\ AFPRED\rangle$

$S23$ $\langle ALL\ QUANTIFIED\ PRED\rangle ::= \langle COL\ OR\ VAL\rangle \langle comp\ op\rangle ALL \langle SUBQ\rangle$

$S49$ $\langle SUBQ\rangle ::= SELECT (ALL | DISTINCT) \langle COL\ OR\ VAL\rangle \langle FRCLAUSE\rangle (\langle WHCLAUSE\rangle)$
$(\langle GBCLAUSE\rangle)(\langle HCLAUSE\rangle)$

$S4$ $\langle COL\ OR\ VAL\rangle ::= \langle correlation\ name\rangle \langle column\ name\rangle | \langle literal\rangle$

$T23$ $\forall TR1 \langle SUBQ\rangle \langle COL\ OR\ VAL\rangle \langle comp\ op\rangle TR2 \langle SUBQ\rangle$

(b) $\forall TR1 \langle SUBQ\rangle d.location = TR2 \langle SUBQ\rangle$

Example of translation - steps (c), and (d)

$S49$ $\langle SUBQ\rangle ::= SELECT(ALL|DISTINCT)\langle COL\ OR\ VAL\rangle\langle FRCLAUSE\rangle$
$(\langle WHCLAUSE\rangle)(\langle GBCLAUSE\rangle)(\langle HCLAUSE\rangle)$

$T49.1$ $\{TR\langle FRCLAUSE\rangle \| TR\langle WHCLAUSE\rangle \wedge \langle HCLAUSE\rangle \|^{F}\}$
$T49.2$ $\langle COL\ OR\ VAL\rangle$

(c) $\{e\ in\ EMP : \|e.d\# = d.d\#\|^{F}\}$
(d) $e.residence$

Example of translation - step (e)

$$
\begin{array}{ll}
(b) & \forall TR1\langle SUBQ\rangle d.location = TR2\langle SUBQ\rangle \\
(c) & \{e \; in \; EMP : \|e.d\# = d.d\#\|^{F}\} \\
(d) & e.residence \\
 & \\
(e) & \forall\{e \; in \; EMP : \|e.d\# = d.d\#\|^{F}\}d.location = e.residence
\end{array}
$$

Example of translation - steps (f), and (g)

$S32$ $\langle HAVING\ SEARCH\ COND \rangle ::= Boolean\ Expression\ of \langle HPRED \rangle$

$S33$ $\langle HPRED \rangle ::= \langle HSIMPLE\ PRED \rangle | \langle HCOMPLEX\ PRED \rangle |$
$\langle HACOL\ PRED \rangle | \langle HAFUN\ PRED \rangle | \langle HACOMPLEX\ PRED \rangle$

$S36$ $\langle HACOL\ PRED \rangle ::= \langle FUNC\ SPEC \rangle \langle comp\ op \rangle \langle COL\ OR\ VAL \rangle$

$S5$ $\langle FUNC\ SPEC \rangle ::= \langle COUNT\ FUNCTION\ SPEC \rangle | \langle AGGR\ FUNCTION\ SPEC \rangle$

$S7$ $\langle AGGR\ FUNCTION\ SPEC \rangle ::= \langle DISTINCT\ AGGR\ FUNCTION \rangle | \langle ALL\ AGGR\ FUNCTION \rangle$

$S10$ $\langle ALL\ AGGR\ FUNCTION \rangle ::= \langle AGGR\ FUNCTION\ NAME \rangle ((ALL) \langle correlation\ name \rangle \langle column\ name \rangle)$

$S11$ $\langle AGGR\ FUNCTION\ NAME \rangle ::= AVG | MAX | MIN | SUM$

$T36$ $TR1 \langle FUNC\ SPEC \rangle \{ TR \langle FRCLAUSE \rangle \| TR \langle WHCLAUSE \rangle \wedge TR \langle GBCLAUSE \rangle \|^{F} \langle comp\ op \rangle \langle COL\ OR\ VAL \rangle$

(f) $AVG(d.nofemp) \{ TR \langle FRCLAUSE \rangle : \| TR \langle WHCLAUSE \rangle \wedge TR \langle GBCLAUSE \rangle \|^{F} \} > 500$

(g) $AVG(d.nofemp) \{ d\ in\ DEPT : \| \forall \{ e\ in\ EMP : \| e.d\# = d.d\# \|^{F} \} d.location = e.residence \wedge$
$TR \langle GBCLAUSE \rangle \|^{F} \} > 500$

Example of translation - step (h)

$S15 \quad \langle \mathit{GBCLAUSE} \rangle ::= \mathit{GROUP\ BY} \langle \mathit{correlation\ name} \rangle \langle \mathit{column\ name} \rangle (\langle \mathit{correlationname} \rangle \langle \mathit{columnname} \rangle)$

$T15 \quad \langle \mathit{correlation\ name} \rangle \langle \mathit{column\ name} \rangle \overset{\omega}{=} \langle \mathit{correlation\ name} \rangle \uparrow .\langle \mathit{column\ name} \rangle$

$(\wedge \langle \mathit{correlation\ name} \rangle \langle \mathit{column\ name} \rangle \overset{\omega}{=} \langle \mathit{correlation\ name} \rangle \uparrow .\langle \mathit{column\ name} \rangle)$

$(h) \quad d.\mathit{manager} \overset{\omega}{=} d \uparrow .\mathit{manager}$

Example of translation - step (i)

$$
\begin{aligned}
(i)\qquad & \{d \textit{ in } DEPT : \|\forall\{e \textit{ in } EMP : \|e.d\# = d.d\#\|^{F}\}d.location = e.residence \wedge \\
& \wedge AVG(d.nofemp)\{d \textit{ in } DEPT : \|\forall\{e \textit{ in } EMP : \|e.d\# = d.d\#\|^{F}\}d.location = e.residence \\
& \wedge d.manager \overset{\omega}{=} d \uparrow .manager\|^{F}\} > 500\|^{F}\}
\end{aligned}
$$

Outline

1. Motivation
2. E3VPC
   - The structure of E3VPC Expressions
   - Interpretation of an E3VPC Expression
   - SQL to E3VPC Translation
3. An Algebra of Spatio-Temporal Predicates
   - Spatial and temporal data types and operations
   - Spatial Predicates
   - Temporally Lifted Spatial Predicates
4. Querying Developments in STSQL

Temporal data types and operations

$$\textit{mregion} : \textit{instant} \rightarrow \textit{region}$$
$$\textbf{defitme:}\textit{moving}(\alpha) \rightarrow \textit{periods}$$

where

- *region* is abstraction for an entity having an extent in the 2D space
- *instant* is a particular chronon or mark on the timeline
- *periods* is set of disjoint anchored intervals on the timeline (*temporal element*)

Outline

1. Motivation
2. E3VPC
   - The structure of E3VPC Expressions
   - Interpretation of an E3VPC Expression
   - SQL to E3VPC Translation
3. An Algebra of Spatio-Temporal Predicates
   - Spatial and temporal data types and operations
   - Spatial Predicates
   - Temporally Lifted Spatial Predicates
4. Querying Developments in STSQL

Topological relationships

$$\begin{bmatrix} \partial A \cap \partial B \neq \varnothing & \partial A \cap B^{\circ} \neq \varnothing & \partial A \cap B^{-} \neq \varnothing \\ A^{\circ} \cap \partial B \neq \varnothing & A^{\circ} \cap B^{\circ} \neq \varnothing & A^{\circ} \cap B^{-} \neq \varnothing \\ A^{-} \cap \partial B \neq \varnothing & A^{-} \cap B^{\circ} \neq \varnothing & A^{-} \cap B^{-} \neq \varnothing \end{bmatrix}$$

- $A, B \ldots$ regions
- $\partial A, \partial B \ldots$ boundary
- $A^{\circ}, B^{\circ} \ldots$ interior
- $A^{-}, B^{-} \ldots$ exterior

Topological predicates

$$\underset{disjoint}{\begin{bmatrix} F & F & T \\ F & F & T \\ T & T & T \end{bmatrix}} \quad \underset{meet}{\begin{bmatrix} T & F & T \\ F & F & T \\ T & T & T \end{bmatrix}} \quad \underset{overlap}{\begin{bmatrix} T & T & T \\ T & T & T \\ T & T & T \end{bmatrix}} \quad \underset{equal}{\begin{bmatrix} T & F & F \\ F & T & F \\ F & F & T \end{bmatrix}}$$

$$\underset{inside}{\begin{bmatrix} F & T & F \\ F & T & F \\ T & T & T \end{bmatrix}} \quad \underset{contains}{\begin{bmatrix} F & F & T \\ T & T & T \\ F & F & T \end{bmatrix}} \quad \underset{covers}{\begin{bmatrix} T & F & T \\ T & T & T \\ F & F & T \end{bmatrix}} \quad \underset{coveredBy}{\begin{bmatrix} T & T & F \\ F & T & F \\ T & T & T \end{bmatrix}}$$

Outline

1 Motivation

2 E3VPC
- The structure of E3VPC Expressions
- Interpretation of an E3VPC Expression
- SQL to E3VPC Translation

3 An Algebra of Spatio-Temporal Predicates
- Spatial and temporal data types and operations
- Spatial Predicates
- Temporally Lifted Spatial Predicates

4 Querying Developments in STSQL

Temporally lifted topological predicates

$$\textbf{inside:}\ \ \textit{region} \times \textit{region} \to \textit{bool}$$

$$\textbf{inside:}\ \ \textit{moving}(\textit{region}) \times \textit{moving}(\textit{region}) \to \textit{moving}(\textit{bool})$$

Definition

A *spatio-temporal predicate* is a function of type $\textit{moving}(\alpha) \times \textit{moving}(\beta) \to \textit{bool} \setminus \{\perp\}$ for $\alpha, \beta \in \{\textit{point}, \textit{region}\}$.

Temporal aggregation

$$\exists p := \lambda(S_1, S_2).\exists t : p(S_1(t), S_2(t))$$

$$\forall_\gamma p := \lambda(S_1, S_2).\forall t \in \gamma(dom(S_1), dom(S_2)) : p(S_1(t), S_2(t))$$

- $S_1, S_2 \in \{moving(\alpha), moving(\beta)\}$
- $\gamma \in \{instant, \cup, \cap, \pi_1, \pi_2\}$
- $\pi_i(x_1, \ldots, x_i, \ldots, x_n) = x_i$

Basic spatio-temporal predicates

- **Disjoint** $:= \forall_{\cap}$ **disjoint**
- **Meet** $:= \forall_{\cup}$ **meet**
- **Overlap** $:= \forall_{\cup}$ **overlap**
- **Equal** $:= \forall_{\cup}$ **equal**
- **Covers** $:= \forall_{\pi_2}$ **covers**
- **Contains** $:= \forall_{\pi_2}$ **contains**
- **CoveredBy** $:= \forall_{\pi_1}$ **coveredBy**
- **Inside** $:= \forall_{\pi_1}$ **inside**

Temporal composition

Definition

Temporal composition. Let p be a spatial predicate, and let P and Q be a spatio-temporal predicates. Then:

$$p \textbf{ then } P := \lambda(S_1, S_2).\exists t : p(S_1(t), S_2(t)) \land P_{>t}(S_1, S_2)$$

$$P \textbf{ until } p := \lambda(S_1, S_2).\exists t : p(S_1(t), S_2(t)) \land P_{<t}(S_1, S_2)$$

P **until** p **then** $Q :=$

$$\lambda(S_1, S_2).\exists t : p(S_1(t), S_2(t)) \land P_{<t}(S_1, S_2) \land Q_{>t}(S_1, S_2).$$

Developments

Disjoint *until* meet *then* (Inside *until* meet *then* Disjoint)

Disjoint $\triangleright$ meet $\triangleright$ Inside $\triangleright$ meet $\triangleright$ Disjoint

Enter := Disjoint $\triangleright$ meet $\triangleright$ Overlap $\triangleright$ coveredBy $\triangleright$ Inside

Leave := Enter$^{\leftarrow}$

Cross := Enter $\triangleright$ Leave

SQL syntax extension

$S19 \quad \langle \mathit{SIMPLE\ PRED} \rangle ::= \langle \mathit{COL\ OR\ VAL} \rangle \langle \mathit{COMP\ OP} \rangle \langle \mathit{COL\ OR\ VAL} \rangle$

$S19.1 \quad \langle \mathit{COMP\ OP} \rangle ::= \langle \mathit{DEVELOPMENT} \rangle | \langle \mathit{STPRED} \rangle | \langle \mathit{SPRED} \rangle | \langle \mathit{comp\ op} \rangle$

$S19.2 \quad \langle \mathit{DEVELOPMENT} \rangle ::= \langle \mathit{STSPRED} \rangle (>> \langle \mathit{STPRED} \rangle)|(. >> \langle \mathit{STSPRED} \rangle)$

$S19.3 \quad \langle \mathit{STSPRED} \rangle ::= \langle \mathit{STPRED} \rangle >> \langle \mathit{SPRED} \rangle$

$S19.4 \quad \langle \mathit{STPRED} \rangle ::= \mathit{Disjoint} | \mathit{Meet} | \mathit{Overlap} | \mathit{Equal} | \mathit{Inside} | \mathit{Contains} | \mathit{Covers} | \mathit{CoveredBy}$

$S19.5 \quad \langle \mathit{SPRED} \rangle ::= \mathit{disjoint} | \mathit{meet} | \mathit{overlap} | \mathit{equal} | \mathit{inside} | \mathit{contains} | \mathit{covers} | \mathit{coveredBy}$

Development query

FLIGHT(id: string, Route: mpoint)
WEATHER(kind: string, Extent: mregion)

```
SELECT f.id
FROM FLIGHT f, WEATHER w
WHERE f.Route Disjoint >> meet >> Inside w.Extent
```

For Further Reading I

M. Negri.
Formal Semantics of SQL Queries.
ACM Transactions on Database Systems, Vol. 17, No. 3, 1991.

R. H. Güting, M. Schneider.
Moving Objects Databases.
Elsevier, 2005.