Moving Object Languages

query languages for moving objects databases
Definition

Data abstraction

moving point - only time-dependent position
- eg. cars, trucks, airplanes, ships, mobile phone users

moving region - also time-dependent shape and extent
- eg. forest fires, oil spills in the sea, armies, epidemic diseases, and hurricanes

Database perspectives

Location management perspective
- maintain locations; query the current and expected near future positions and relations

Spatio-temporal data perspective
- the complete histories of movements are represented
Modeling and Querying Current Movement
– the MOST Model and FTL Language

Strategies:
1. Store a motion vector - a position as a linear function of time
2. Database is updated if the deviation exceeds the threshold
3. Assumptions on object classes and spatial attributes
4. Contains the object - Time, yields the current time at every instant

Dynamic Attributes
A standard data type but changes its value automatically over time
\[ \text{eg. } \text{value}(A, t) = A.\text{value} + A.\text{function}(t - A.\text{datetime}). \quad t \geq A.\text{datetime} \]

Representing Object Positions
\[ \text{loc}(\text{route}, \text{startlocation}, \text{starttime}, \text{direction}, \text{velocity}, \text{uncertainty}) \]

Semantics of Queries, Query Types
Database state: A mapping (associates each object class with a set of objects of appropriate types, and the Time object)
Database history: An infinite sequence of states
\[ \text{eg. } Q(Ht, t). \quad (\text{instantaneous query}) \]
\[ Q(Ht, t), Q(Ht+1, t + 1), Q(Ht+2, t + 2), \ldots \text{(continuous query)} \]

The Language FTL
\[ \text{RETRIEVE <target-list> FROM <object classes>} \]
\[ \text{WHERE <FTL-formula>} \quad \text{until, nexttime, always, eventually_within_c...} \]
Modeling and Querying History Movement
– Spatio-temporal Data Types(1)

Important Abstractions:
Geometries in spatial databases change continuously over time
Point(position relevant), Line(connections), Region(extents relevant),
Partition(subdivisions), Network(graph structure)

Spatio-temporal Data Types:
moving point and moving region visualized (2D/3D + time)

Example Operations and Queries
Trajectory, traversed, deftime, intersection, atinstant, val, duration
eg. SELECT count(*) FROM cars AS c, weather AS w WHERE duration(deftime(intersection(c.trip, w.area))) > 1800

Goals in the Design of Types and Operations
Closure of type system; Genericity; Consistency between
nontemporal and temporal types; Consistency between nontemporal and temporal operations

Abstract and Discrete Model
Abstract models: mathematically simple, elegant, and uniform, but
not directly implementable.
Discrete models: more complex and heterogeneous, but can be implemented (a finite representation)
Modeling and Querying History Movement
– Spatio-temporal Data Types(2)

**Operations:**
1. Design operations for nontemporal types.
2. Use lifting make them all time dependent in a way consistent with the static definition.
3. Add specialized operations for the temporal types

**Implementation:**
- sliced representation
- temporal function
- simple function
- unit

**The Structure of The Type System**

```
    instant
    \downarrow
    \begin{array}{c}
    \text{int} \\
    \text{real} \\
    \text{string} \\
    \text{bool} \\
    \text{point} \\
    \text{points} \\
    \text{line} \\
    \text{region}
    \end{array}
    \begin{array}{c}
    \text{moving(int)} \\
    \text{moving(real)} \\
    \text{moving(string)} \\
    \text{moving(bool)}
    \end{array}
    \begin{array}{c}
    \text{all temporal types for these} \\
    \text{all projections to the range}
    \end{array}
    \begin{array}{c}
    \text{range(int)} \\
    \text{range(real)} \\
    \text{range(string)} \\
    \text{range(bool)}
    \end{array}
    \begin{array}{c}
    \text{moving(point)} \\
    \text{moving(points)} \\
    \text{moving(line)} \\
    \text{moving(region)}
    \end{array}
    \begin{array}{c}
    \text{all projections to the domain}
    \end{array}
    \begin{array}{c}
    \text{periods}
    \end{array}
```

*eg.*
- change position in discrete steps --> set of points
- move continuously --> a curve line value
Further Work and Applications

1. Moving Objects in Networks
   A network is modeled as a set of routes and junctions between routes
   $gpoint, gline, moving(gpoint)$ and $moving(gline)$

2. Spatio-temporal Predicates and Developments
   The framework first allows one to obtain basic spatio-temporal
   predicates by aggregating a static topological relationship over all
   instants of a time interval
   $Cross := Disjoint --> meet --> Inside --> meet --> Disjoint$

3. Uncertain Trajectories
   $PossiblySometimeInside$ \quad $SometimePossiblyInside$
   $PossiblyAlwaysInside$ \quad $AlwaysPossiblyInside$

Key Applications

Query languages (first kind) - current and near future movement
   gas stations, hotels, parcel delivery services, air traffic control

Query languages (second kind) - history of movement
   movements of animals, deforestation of the Amazon rain forest