Maintaining Knowledge about Temporal Intervals

James F. Allen, University of Rochester

Speaker: Maurizio Atzori, University of Pisa

Temporal Knowledge

- Needed to temporal reasoning problems
- Outdated data can be important:
  - “Which employees worked for us last year?”
- We need to represent temporal information

Examples from AI

- Planning
  - robot's actions on the world must be taken into account to ensure that a plan will be effective
- Natural language processing
  - Temporal information in sentences are extracted from sentences to be able to answer queries

Temporal Intervals

- Temporal interval is a primitive notion
  - Relationships between temporal intervals
    - Based on hierarchy
    - Constraint propagation techniques
  - Temporal knowledge is relative
    - Cannot be described by a date

Contents

- Interval-based systems vs. point-based systems
- Interval-based deduction technique based on constraint propagation

Temporal representation chararacteristics

- Should allow imprecision
  - A before B
- Uncertainty of information
  - we don’t know the exact relationship between A and B but some constrains on how they could be related are known
- Grain of reasoning
  - Days, years… nanoseconds!
- Persistence
  - Default reasoning: “I parked my car this morning, it should be there now”

Temporal systems approaches

- State space approaches
- Date line systems
- Before/after chaining
- Formal models

State space approaches

- Only for simple problem solving tasks
- A state is a description of the world
  - Actions are functions mapping between states
- Previous states:
  - Can be retained (better representation)
  - Can be deleted: present state only (efficiency)
- Do provide a notion of persistence

Database systems

- Each fact is indexed by a date
  - Date: representation of a time such that the temporal ordering between 2 dates can be computed by simple operations
    - Integers, calendar dates
- Unfortunately we cannot assign a precise date for every event
  - Fuzzy extensions
- A and B not together
  - Time intervals make the harder

Before/After chaining

- Capture relative temporal information quite directly
- Computational problems:
  - When the amount of temporal information grows
    - Search problems (long chains)
    - Space problems (if all possible relationships are precomputed)
- Allan's work is based on B/A chaining

Formal models

- From philosophy
- From AI
  - Situation calculus
  - McDermott work
- All of them are point-based theories
  - Time intervals are created from time points

Time points vs. time intervals: sentences

Precise point in time at which the finding event occurred

- We found the letter at twelve noon

- Time interval
  - We found the letter yesterday
- Temporal relation between time intervals
  - We found the letter while John was away
  - We found the letter after we made the decision

Time points vs. time intervals: decomposition

- “finding the letter”
  - “looking at spot X where the letter was”
  - “realizing that it was the letter you were looking at”
- And then decompose “realizing…”
- The times we consider are times of events
  - We can decompose events
    - Time point seems to be not useful
  - Time points can be useful if they are “ very small intervals”

Time points vs. time intervals: endpoints

- A situation where a light is turned on
- Two interval times: light off, light on
  - (off) (on)
  - [off] [on]
- An "artificial" solution
  - [off) [on)
- Time based on points does not correspond to our intuitive notion of time
  - intervals as primitive!

Time points vs. time intervals: intervals as primitive

- Definition of intervals using endpoints
  - An interval **t** is an ordered pair of points with the first point less than the second
    - (t-, t+)
    - t- < t+

Interval Relations defined by endpoints

Interval relation	*Equivalent Relations on Endpoints*
t < s	t+ < s-
t = s	(t- =s-) & (t+ = s+)
t **overlaps** s	(t- < s-) & (t+ > s-) & (t+ < s+)
t **meets** s	t+ = s-
t **during** s	((t- > s-) & (t+ =< s+)) or ((t- >= s-) & (t+ < s+))

13 relations

Relation	Symbol	Sym inverse	Pict Example
X **before** Y	<	>	XXX YYY
X **equal** Y	=	=	XXX YYY
X **meets** Y	m	mi	XXXYYY
X **overlaps** Y	o	oi	XXX YYY
X **during** Y	d	di	XXX YYYYYY
X **starts** Y	s	si	XXX YYYYYY
X **finishes** Y	f	fi	XXX YYYYYY

Network representation

- Nodes represent individual intervals
- Arcs are labeled to indicate the possible relationship between two intervals (their nodes)
- Uncertainty → all possible cases are entered on the arc
- Note: 13 possibilities are mutually exclusive
  - No ambiguity in this notation

Representing knowledge of temporal relations in a Network

■ i during j	Ni –(d)→ Nj
■ i during j or i before j or j during i	Ni –(< d di)→ Nj
■ (i < j) or (i > j) or i meets j or j meets i	Ni –(< > m mi) → Nj

Basic algorithm

Giver R1, R2 arc labels (two sets of relations)

Constraints(R1, R2)

C ← ε

for each r1 **in** R1

for each r2 **in** R2

C ← C **union** T(r1, r2);

return C;

Basic algorithm (2)

```
To Add R(i,j)
    Add <i,j> to queue ToDo
    While ToDo is not empty do
        Get next <i,j> from ToDo
        N(i,j) ← R(i,j)
        For each node k such that Comparable(k,j) do
            R(k,j) ← N(k,j) ∩ Constraints(N(k,i),R(i,j))
            If R(k,i) contained in N(k,i)
                then add <k,i> to ToDo
        For each node k such that Comparable(i,k) do
            R(i,k) ← N(i,k) ∩ Constraints(N(i,k),R(i,k))
            If R(i,k) contained in N(k,i)
                then add <i,k> to ToDo
    end while
```

An example

- "John was not in the room when I touched the switch to turn on the light"
  - S is the time of touching the switch
  - L is the time the light was on
  - R is the time that John was in the room
- The network:
  - $R \leftarrow (< m \; mi >) - S - (o \; m) \rightarrow L$

An example (2)

- When we insert the second fact (R ← S) we compute:

    T(oi, <) = (< o m di fi)

    T(oi, m)

  …

    T(mi, >) = (>)

  R ← (< > o oi m di s si fi =) – L

An example (3)

- And then: “But John was in the room later while the light went out”

L overlaps, starts or during R

L –(o s d) → R

An example (4)

- We eliminate the impossible relationship d (by intersection)
- We propagate the effects of the constraints through the network
  - A new constraint between S and R can be calculated using the path:

    S –(om) → L –(os) → R

  We find S –(< o m) →R and intersecting S –(< m)→R

An example (5)

- Finally we have:

R ← (< m) – S –(o m) → L

R ← ——— (o s) —————— L

R ← (< m) – S

(John entered the room either after I touched the switch or at the same time that I finished touching the switch)

Conclusion

- Good balance between expressive power and efficiency of its deductive engine
- High efficiency in time
- High space requirement solved by some extentions to the basic algorithm
  - Reference Intervals (Hierarchy among time intervals)
    - Cluster of intervals for which the temporal constraints between each pair of intervals is fully computed