Scribe Notes: 1/9/2013
Presenter: Dr. Berthe Y. Choueiry
Scribe: Robert Woodward

Topic: Temporal Reasoning (Chapter 12 of Dechter book)

The discussion focused on the content of Chapter 12 of Dechter's textbook. However, the Handbook of CP too does a fantastic, comprehensive summary of the research of CP on temporal reasoning. Another good survey on temporal CSPs, focused on only *metric constraints,* is the paper by Planken (2007).

Refresher

A refresher from 421/821.

Minimality: Zion answered:

- Every variable-value-pair appears in some solution (domains are minimal).
- Every tuple appears in some solution (constraints are minimal). Removing any tuple eliminates a solution or more.

Path consistency: 4 loops (k, the middle point, must always be in the outer loop)

repeat until quiescence
… ∀ k = 1 to n
…… ∀ i = 1 to n
……… ∀ j = 1 to n
………… $R_{i,j} \leftarrow R_{i,j} \cap R_{i,k} \cdot R_{k,j}$

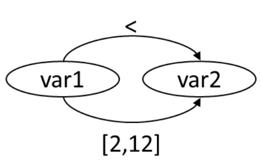
If composition distributes over intersection, or convexity property, then

1. we can get rid of outer loop and
2. PC guarantees decomposability and minimality.

Overview

Temporal and spatial reasoning problems share a fair amount of similarities. We often hear of combination: temporal-spatial problems. Often, researchers work in both areas.

In temporal constraint problems, the constraints between variables restrict the times that the variable could occur at. For example,



in the figure above, there are two constraints between the variables *var1* and *var2*:

- The first constraint is a *qualitative constraint* (<), and
- The second constraint is a *metric* constraint. The interval indicates: [2,12] ≡ $2 \leq (var2 - var1) \leq 12$. This specific constraint is also called a *constraint of bounded difference*. In Constraint DB, it is called a *gap constraint*.

Temporal Reasoning is important because of its many applications, such as:

- Planning, scheduling, resource allocation. In AI, we usually make the following assumptions:
    - Planning focused on the *sequence* of actions

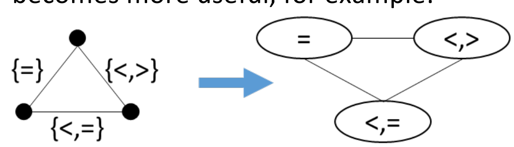
    - Scheduling focused on *time-point* assignments to actions
    - In resource allocation, we generally assume that we know the sequence and time execution of actions, and we are concerned about assigning resources. For example, in assigning TAs to courses, we assume the course schedule is fixed.
- Plan recognition
    - Understand story telling. (E.g., in the story Little Red Ridding Hood, there are many of actions. Understanding the sequence of actions allows understanding the intrigue of the story and the drama between wolf and the girl.)
- Verification: In hardware verification and digital logic, temporal logics are usually used to simulate the operation of the signals in time and verify the implemented functionalities (e.g., ITL, HNL, CTL, LTL, MITL, STL).

Temporal Constraint Networks

Syntax: time objects and relations (constraints)
Semantics: The basic objects are

1. Time entities: they can be *time points* or *time intervals*.
    - They represent events w/ or w/o duration. Example of event: switched light on. There is some time it takes to turn the light on, what happens in this in-between time?
    - One can imagine also various time granularities (intervals in one granularity can be a point in another) and algebra for translating between them.
2. Constraints:
    - Can be specified as *qualitative* (general relations, w/o numbers) or *quantitative* (specificying constraints with numbers)

    Notice, the domains of variables are continuous. Often, we do not care about the domains, but usually we care only about relationship between domains. *Emphasis is on the relations*. When the emphasis is on the relations, Tony suggested that the dual graph representation becomes more useful, for example:



Constraint Graph   Dual Graph

    When there is no edge between two variables, have a **universal** constraint (not a global constraint).

Proof Theory: Algorithms to reason about the objects and their relations.

a. In CP, search + propagation
b. In temporal reasoning, lots of propagation, path consistency, and a little search.
c. In temporal logic, theorem proving (inference rules) or model checking (search).

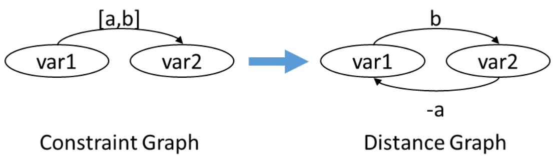
Approaches in AI for Temporal Reasoning: Logic and constraint.

- *Temporal Logics* (purely theoretical): Very sophisticated, (e.g., see International Symposium on Temporal Representation and Reasoning: And International Conference on Temporal Logic. TIME-ICTL).

- *Constraint (Temporal) Networks* (using CSPs): The focus of this talk.

Types of temporal Networks

1. Qualitative relations:
    - *Interval Algebra*: 13 relations – before, meets, overlaps, during, starts, finishes, their inverses, and equal. Discussion: No inverse of equals because its inverse is the same. Saying *x equals y* and *y equals x* is the same. However, *x before y* and *y before x* means two different things.
    - *Point Algebra*: <, =, >. Adding ≠ (disjunction) makes the problem intractable.
2. Quantitative: Difference of points. Two different representations, as shown below:



In the Simple Temporal Problem, for example, all constraints are constraints of bounded differences.

- We can convert the STP to a distance graph, use Floyd-Warshal Algorithm (from 310) to determine decomposability and minimality. It can thus determine whether or not the network is consistent. (We can also use Bellman-Floyd.)
- Therefore, path-consistency can solve the problem.

Qualitative Temporal Networks

Interval Algebra (IA)/Allen Algebra

- Question: (Daniel) Can there be both a `Before` and a `During`? Yes, because {b,d} is a *disjunction*, therefore it could be `Before` or `During`. One of them must hold (not necessarily both).

Interval Algebra Constraint Networks formally defined (Definition 12.1, page 336)

- **Variables**: temporal intervals
- **Domain**: set of ordered pairs of real numbers (however do not typically care about domain)
- **Constraints:** are subsets of 13 relations (Always a binary CSP)
    - Zion states there are 8191 = 2^{13}-1 distinct relations
- **Solution**: assignment of a pair of numbers to each variable such that no constraint is violated (but we do not really care about the numbers assigned to the time points, but care about the minimality of the constraints)

Cannot search on the variables (continuous domains), therefore, search on the dual of the network (discrete domains).

Nate noticed an error on the example on slide 11. The statement must be "when light went off" and the arrow switch and room should be reversed (both on slide 11 and slide 12).

Homework for Monday: compute how to filter the edges to get the minimal network of the example of Figure 12.1. Use Table 12.2 (page 339) for computing the composition in the interval algebra.

	b	***s***	***d***	***o***	***m***
b	*b*	*b*	*bomds*	*b*	*b*
s	*b*	*s*	*d*	*bom*	*b*
d	*b*	*d*	*d*	*bomds*	*b*
o	*b*	*o*	*ods*	*bom*	*b*
m	*b*	*m*	*ods*	*b*	*b*

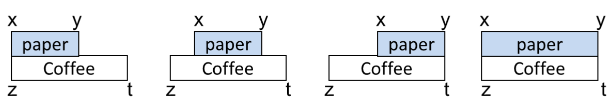
Notice from the table: composing with *o* or *d* yields disjunction (does not tighten). Such disjunctions can be the source of intractability and we may have to use backtrack search to enumerate consistent possibilities.

Point algebra (PA)

- **Variables:** time points
- **Domains:** reals
- **Constraints**: express relative positions of 2 points (Always a binary CSP)
    - Three basic relations, <, =, >. (2^3 -1 = 7 possible relations)

Point algebra is cheaper than interval algebra. In PA, 3/4-consistency guarantees minimal network.

Example:



IA: Paper {s,d,f,=} Coffee

PA: Paper: [x,y], Coffee: [z,t]

Constraints: x < y, z < t

From first: x = z, y < t

From second: x < t, y > z

From third: z < x, y = t

From fourth: x = z, y = t

Simplifying the constraints yields the conjunction of constraints: x < y, z < t, x < t, x ≥ z, y ≤ t, y > z

Conversion from IA to PA not always possible. ☹ Consider x {b,a} y in the example above. We cannot express this disjunction in PA (PA can only express conjunction).

<u>Out of time</u>: for next lecture prepare more questions/answer other questions.