Planning & Scheduling

Roman Barták
Department of Theoretical Computer Science and Mathematical Logic

Temporal Reasoning

Planning and time

The conceptual model of planning assumes **implicit time**:

- actions and events are instantaneous (no duration)
- goals are verified at the end of the plan

This restricted view of planning is appropriate for studying the "logic" behind planning (situation calculus) and for formal complexity studies.

- **In practice** the situation is slightly **different**:
  - actions take some time (**duration**) to be executed
  - action **preconditions** may be required also during action duration (not just at the beginning)
  - action **effects** may happen before the end of the action, they may be true during action duration, or even they may become true sometime later
  - **effects** of more actions may be **combined**
  - **goals** may be required **during execution** of plans

Basic principles

What is time?

The core mathematical structure for describing time is a **set with transitive and asymmetric ordering** relation.

The set can be continuous (real numbers) or discrete (integer numbers).

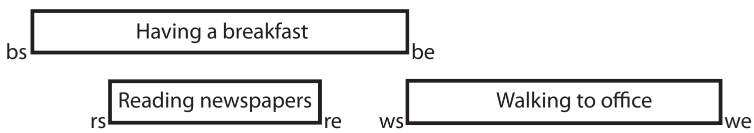
The planning system will use a **database of temporal references** with a procedure for **verifying consistency** and an **inference mechanism** (to deduce new information).

We can model time in two ways:

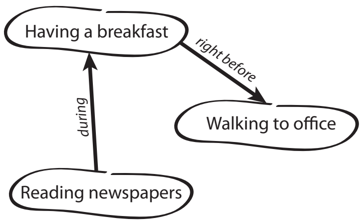
- **qualitative**
  relative relations (A finished before B)
- **quantitative**
  metric (numerical) relations (A started 23 minutes after B)

Qualitative approach

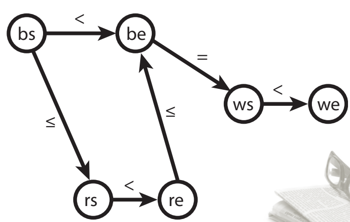
- Based on **relative temporal relations** between temporal references.
- *"I read newspapers during breakfast and after breakfast I walked to my office"*



Temporal intervals (activities)



Time points (important events)



Qualitative frameworks

When **modeling time** we are interested in:

- **temporal references**
  (when something happened or hold)
  - **time points** (instants) when a state is changed
    instant is a variable over the real numbers
  - **time periods** (intervals) when some proposition is true
    interval is a pair of variables (x,y) over the real numbers, such that $x<y$
- **temporal relations** between temporal references
  - **ordering** of temporal references

Typical problems solved:

- verifying **consistency** of the temporal database
- asking **queries** (*"Did I read newspapers when entering the office?"*)
- finding **minimal networks** to deduce inevitable relations

Vilain & Kautz (1986)

Point algebra - foundations

Symbolic calculus modelling qualitative relations between instants.

- There are three possible **primitive relations** between instants t_1 and t_2:
  - $[t_1 < t_2]$
  - $[t_1 > t_2]$
  - $[t_1 = t_2]$

  Relations P = {<,=,>} are called **primitive relations**.
- Partially known relation between two instants can be modelled using a set (disjunction) of primitive relations:
  - {}, {<}, {=}, {>}, {<,=}, {>,=}, {<,>}, {<,=,>}
- **Relation** r between temporal instants t and t' is denoted
  [t r t']
- Point algebra allows us to **work with relative relations** without placing the instants to particular (numeric) times.

Point algebra - operations

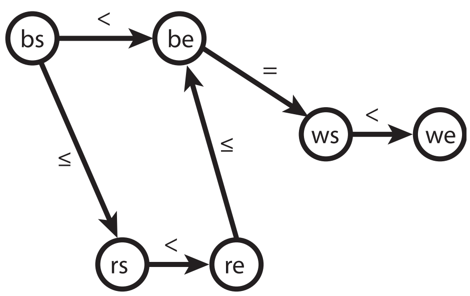
- Let R be a set of all possible relations between two instants
  - {{}, {<}, {=}, {>}, {<,=}, {>,=}, {<,>}, {<,=,>}}
- Symbolic operations over R:
  - **set operations ∩,** ∪
    - they express conjunction and disjunction of relations
  - **composition operation •**
    - transitive relation for a pair of connected relations
    - $[t_1\ r\ t_2]$ and $[t_2\ q\ t_3]$ gives $[t_1\ r \bullet q\ t_3]$ using the table

•	<	=	>
<	<	<	P
=	<	=	>
>	P	>	>

- The most widely used operations are ∩ and •, that allow combining existing and inferred relations:
  - $[t_1\ r\ t_2]$ and $[t_1\ q\ t_3]$ and $[t_3\ s\ t_2]$ gives $[t_1\ r \cap (q \bullet s)\ t_2]$

Point algebra – inference

"I read newspapers during breakfast and after breakfast I walked to my office"



- Query: *"Did I read newspapers when entering the office?"*
- [rs < we] ∧ [we < re]

$$(r_{re,be} \bullet r_{be,ws} \bullet r_{ws,we}) \cap (r_{re,we})$$
$$= (\{=,<\} \bullet \{=\} \bullet \{<\}) \cap \{>\}$$
$$= \{<\} \cap \{>\} = \{\}$$

•	<	=	>
<	<	<	P
=	<	=	>
>	P	>	>

Point algebra - consistency

- A set of instants X together with the set of (binary) temporal relations $r_{i,j} \in R$ over these instants C forms a **PA network** (X,C).
  - If some relation is not explicitly assumed in C then we assume universal relation P.
- The **PA network** consisting of instants and relations between them is **consistent** if it is possible to assign a real number to each instant in such a way that all the relations between instants are satisfied).

Claim:

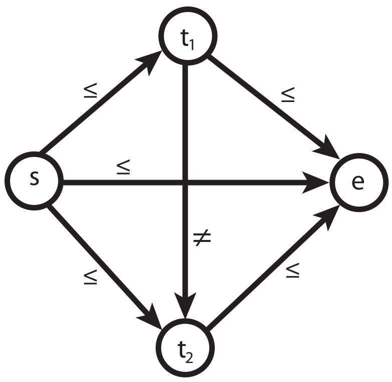
The PA network (X,C) is consistent if and only if there exists a set of primitive relations $p_{i,j} \in r_{i,j}$ such that for any triple of such relations $p_{i,j} \in p_{i,k} \bullet p_{k,j}$ holds.

Efficient consistency checking:

To make the PA network consistent it is enough to make its transitive closure, for example using techniques of **path consistency**.

- for each k: for each i,j: do $r_{i,j} \leftarrow r_{i,j} \cap (r_{i,k} \bullet r_{k,j})$
- obtaining {} means that the network is inconsistent

Point algebra – minimal networks



- PC verifies consistency but does not remove redundant constraints.
- Primitive constraint $p_{i,j}$ is **redundant** if there does not exist any solution where $[t_i\ p_{i,j}\ t_j]$ holds.

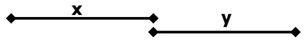
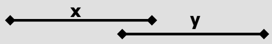
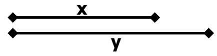
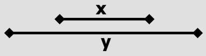
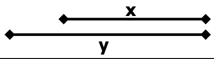
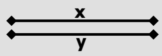
- **PA network is minimal** if it has no primitive constraints that are redundant.
- To make the network minimal we need 4-consistency.

Allen (1983)

Interval algebra

Symbolic calculus modelling relations between intervals
(interval is defined by a pair of instants i^- and i^+, $[i^- < i^+]$)

- There are thirteen primitive relations:

x **b**efore y	$x^+ < y^-$	
x **m**eets y	$x^+ = y^-$	
x **o**verlaps y	$x^- < y^- < x^+ \wedge x^+ < y^+$	
x **s**tarts y	$x^- = y^- \wedge x^+ < y^+$	
x **d**uring y	$y^- < x^- \wedge x^+ < y^+$	
x **f**inishes y	$y^- < x^- \wedge x^+ = y^+$	
x **e**quals y	$x^- = y^- \wedge x^+ = y^+$	
bi,mi,oi,si,di,fi	symmetrical relations	

Interval algebra – consistency

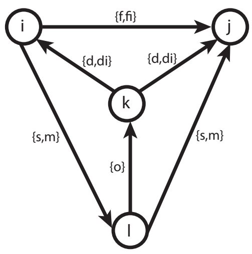
- Primitive relations can be again combined in sets (2^{13} relations).
  - Sometimes we select only a subset of possible relations that are useful for a particular application.
    - for example {b,m,b',m'} means no-overlaps and it is useful to model unary resources
- set operations ∩, ∪ and the composition operation •
- The **IA network** is **consistent** when it is possible to assign real numbers to x_i^-, x_i^+ of each interval x_i in such a way that all the relations between intervals are satisfied.

Claim:

The IA network (X,C) is consistent if and only if there exists a set of primitive relations $p_{i,j} \in r_{i,j}$ such that for any triple of such relations $p_{i,j} \in p_{i,k} \bullet p_{k,j}$ holds.

Notes:

- Path consistency is not a complete consistency technique for interval algebra.
- Consistency-checking problem for IA networks is an NP-complete problem.
- Intervals can be converted to instants but some interval relations will not be binary relations among the instants.



Qualitative algebra

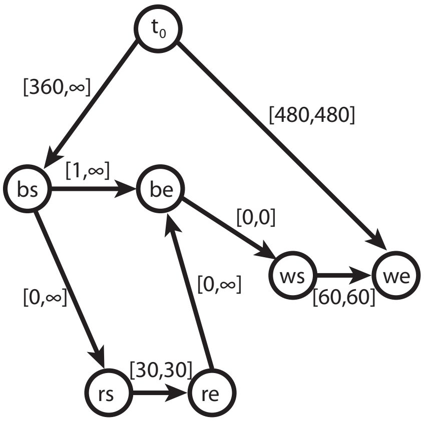
- Points in the end of interval are not fully translatable to instants.
- *"A light bulb is off and after switching the toggle, the light becomes on"*
  - Can be modelled using two intervals *on* and *off* and one interval relation *on* {m} *off*.
  - Is light on or off at the instant between the intervals?
- **Qualitative algebra** uses interval and instants as first-order objects:

p **b**efore i (i after p)	$p < i^-$	p i
p **s**tarts i (i started-by p)	$p = i^-$	p i
p **d**uring i (i includes p)	$i^- < p$, $p < i^+$	p i
p **f**inishes i (i finished-by p)	$p = i^+$	p i
p **a**fter i (i before p)	$i^+ < p$	i p

Quantitative approach

"I got up at 6 o'clock. I read newspapers for 30 minutes during the breakfast. After the breakfast I walked to my office which took me one hour. I entered the office at 8:00AM".

When did I start my breakfast?



- 360 =< bs, *"I got up at 6 o'clock"*
- bs =< rs, re =< be, *"I read newspapers during breakfast"*
- re-rs = 30, "I read newspapers for 30 minutes"
- be = ws, *"after breakfast I walked to my office"*
- we-ws = 60, *"[walking] took me one hour"*
- we = 480, *"I entered the office at 8:00AM"*

bs =< rs = re-30 =< be-30 = ws-30 = (we-60)-30 = 390

I started my breakfast between 6:00AM and 6:30AM.

- The basic temporal primitives are again **time points**, but now the relations are numerical.
- Simple **temporal constraints** for instants t_i and t_j:
  - unary: $a_i \leq t_i \leq b_i$
  - binary: $a_{ij} \leq t_i - t_j \leq b_{ij}$,

  where a_i, b_i, a_{ij}, b_{ij} are (real) constants

Notes:

- Unary relation can be converted to a binary one, if we use some fix origin reference point t_0.
- $[a_{ij},b_{ij}]$ denotes a constraint between instants t_i a t_j.
- It is possible to use disjunction of simple temporal constraints.

Simple Temporal Network (STN)

- only simple temporal constraints $r_{ij} = [a_{ij},b_{ij}]$ are used
- **operations**:
  - composition: $r_{ij} \bullet r_{jk} = [a_{ij}+a_{jk}, b_{ij}+b_{jk}]$
  - intersection: $r_{ij} \cap r'_{ij} = [\max\{a_{ij},a'_{ij}\}, \min\{b_{ij},b'_{ij}\}]$
- **STN** is **consistent** if there is an assignment of values to instants satisfying all the temporal constraints.
- **Path consistency** is a complete technique making STN consistent (all inconsistent values are filtered out, one iteration is enough). Another option is using all-pairs minimal distance **Floyd-Warshall algorithm**.

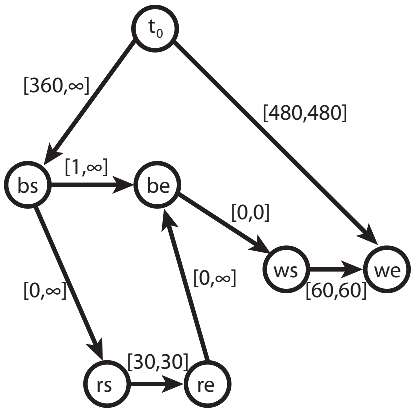
Distance graph

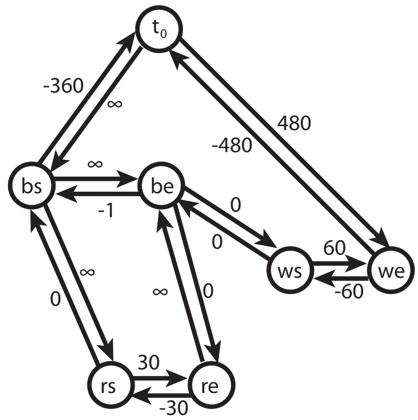
Relations $a_{ij} \leq t_i - t_j \leq b_{ij}$ can be expressed as maximal distances between the time points:

- $t_i - t_j \leq b_{ij}$
- $t_j - t_i \leq -a_{ij}$

This gives a **distance graph**.

- Negative cycle in the distance graph means inconsistency.





Algorithms

- **Path consistency**
  - finds a transitive closure of binary relations r
  - one iteration is enough for STN (in general, it is iterated until any domain changes)
  - works incrementally

one iteration for STN

```
PC(X, C)
    for each k : 1 ≤ k ≤ n do
        for each pair i, j : 1 ≤ i < j ≤ n, , i ≠ k, j ≠ k do
            r_ij ← r_ij ∩ [r_ik • r_kj]
            if r_ij = Ø then exit(inconsistent)
end
```

general

```
PC(C)
    until stabilization of all constraints in C do
        for each k : 1 ≤ k ≤ n do
            for each pair i, j : 1 ≤ i < j ≤ n, i ≠ k, j ≠ k do
                c_ij ← c_ij ∩ [c_ik • c_kj]
                if c_ij = Ø then exit(inconsistent)
end
```

- **Floyd-Warshall algorithm**
  - finds minimal distances between all pairs of nodes
  - First, the temporal network is converted into a **distance graph**
    - there is an arc from i to j with distance b_{ij}
    - there is an arc from j to i with distance $-a_{ij}$.
  - STN is consistent iff there are no negative cycles in the graph, that is, d(i,i)≥0

```
Floyd-Warshall(X, E)
    for each i and j in X do
        if (i, j) ∈ E then d(i, j) ← l_ij else d(i, j) ← ∞
        d(i, i) ← 0
    for each i, j, k in X do
        d(i, j) ← min{d(i, j), d(i, k) + d(k, j)}
end
```

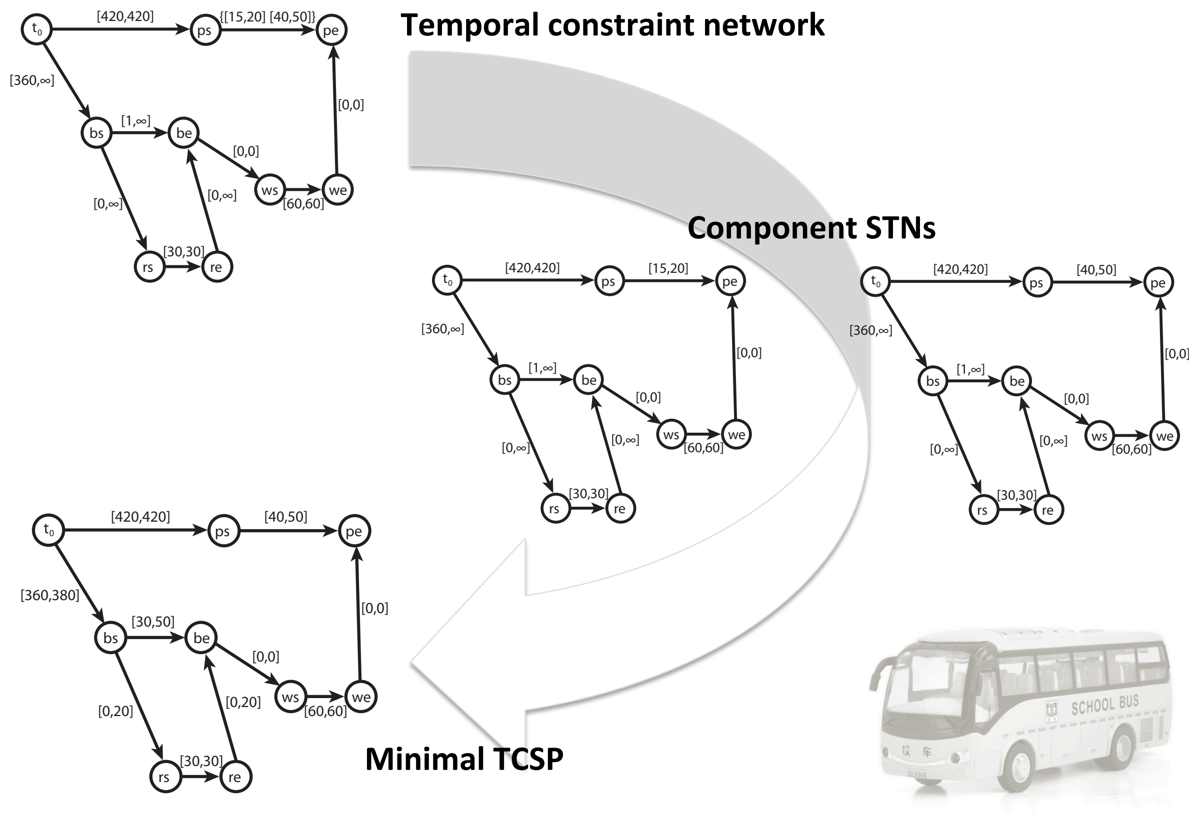
"I got up at 6 o'clock. I read newspapers for 30 minutes during the breakfast. After the breakfast I walked to my office which took me one hour. I entered the office exactly at the same time as Peter who left his home at 7:00AM. ***Peter is going to office either by a car, which takes him 15-20 minutes, or by a bus, which takes 40-50 minutes".***

We need to express a disjunction of simple temporal constraints between the same pair of temporal points:

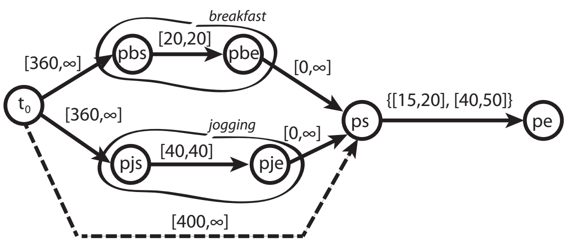
40 =< pe-ps =< 50 ∨ 15 =< pe-ps =< 20

Temporal Constraint Network (TCSP)

- It is possible to use **disjunctions of simple temporal constraints** over the same variable.
- Operations • and ∩ are being done over the sets of intervals.
- **TCSP** is **consistent** if there is an assignment of values to instants satisfying all the temporal constraints.
- Path consistency does not guarantee in general the consistency of the TCSP network!
- A straightforward **approach** (constructive disjunction):
  - decompose the temporal network into several STNs (component STNs) by choosing one disjunct for each constraint
  - solve obtained STN separately (find the minimal network)
  - combine the result with the union of the minimal intervals



"Peter got up at 6 o'clock and before leaving home we went jogging for 40 minutes and had breakfast which took him 20 minutes. Peter is going to office either by car, which takes him 15-20 minutes, or by a bus, which takes 40-50 minutes."

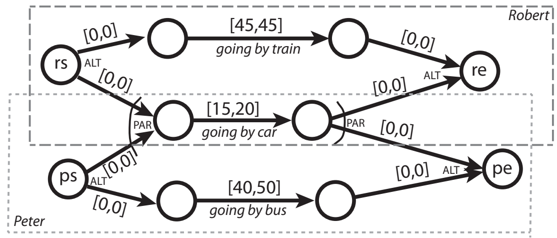


- We need to express that jogging and breakfast do not overlap in time!

  pbe =< pjs ∨ pje =< pbs
- This is a so called a **Disjunctive Temporal Problem** (opposite to TCSP, n-ary disjunctions can be used).
- DTN can be solved similarly to TCSP – by decomposition to component STNs.

Temporal network with alternatives

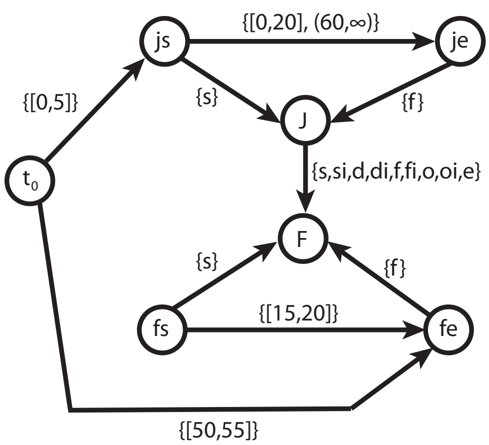
"When Peter goes by car then Robert joins him, otherwise Robert goes by train which takes him 45 minutes."



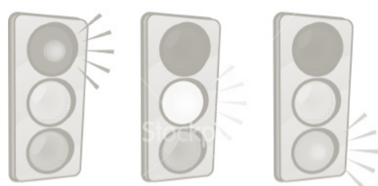
- We need to express that some points do not appear in the network by adding branching (logical) constraints.
- **Temporal Network with Alternatives** assumes parallel/alternative branching constraints in addition to temporal constraints.
  - Solution consists of selection of nodes satisfying the branching and temporal constraints.

General temporal constraint network

"John and Fred work for a company that has local and main offices in Los Angeles. They usually work at the local office, in which case it takes John less than 20 minutes and Fred 15–20 minutes to get to work. Twice a week John works at the main office, in which case his commute to work takes at least 60 minutes. Today John left home between 7:05–7:10 a.m., and Fred arrived at work between 7:50–7:55 a.m. We also know that Fred and John met at a traffic light on their way to work."



General Temporal Constraint Network combines points and intervals and supports constraints from the qualitative algebra and a from a TCSP.



Summary

	name	approach	temporal reference	temporal propositions	complexity
PA	point algebra	qualitative	time points	{<,=,>}	tractable
IA	interval algebra	qualitative	intervals	{b,m,o,s,d,f,e,bi,mi,oi,si,di,fi}	NP-c
QA	qualitative algebra	qualitative	time points, intervals	IA, PA, interval-to-point	NP-c
STP	simple temporal problem	quantitative	time points	binary difference	tractable
TCSP	temporal CSP	quantitative	time points	binary disjunctive difference	NP-c
DTP	disjunctive temporal problem	quantitative	time points	n-ary disjunctive difference	NP-c
TNA	temporal network with alternatives	quantitative	time points	precedence, logical	NP-c
	general temporal CSP	qualitative, quantitative	time points, intervals	TCSP, QA	NP-c

Department of Theoretical Computer Science and Mathematical Logic
bartak@ktiml.mff.cuni.cz