CSc 355 - AI: Knowledge and Reasoning

Gerd Kortuem
kortuem@comp.lancs.ac.uk

Admin Details

- Gerd Kortuem
- Room D17, Infolab21
- www.comp.lancs.ac.uk/~kortuem
- 2 Lectures this week

Readings

- Russel & Norvig, Section III Knowledge & Reasoning
- Finlay, Dix, An Introduction to Artificial Intelligence, 1996
- J.F. Allen. Time and Time Again: The Many Ways to Represent Time. Intl. Journal on Intelligent SYstems 6(4) 1991
- J. F. Allen. Maintaining Knowledge about Temporal Intervals. CACM, 26(11):832 – 843, 1983.
- Online resources on Prolog

Content

- Motivation
  - Smart Homes
- Reasoning
  - Deduction, Induction, Abduction
  - Prolog (horn-logic, backtracking algorithm, proof trees)
- Spatial / Temporal Reasoning
  - Allen’s Interval Algebra
  - Temporal reasoning with constraint propagation algorithms

The Smart Home Dream

"They walked down the hall of their soundproofed Happylife Home, which had cost them thirty thousand dollars installed, this house which clothed and fed and rocked them to sleep and played and sang and was good to them."

The Illustrated Man, Ray Bradbury

Georgia Tech Aware Home

the AWARE HOME

A residential laboratory developing technology
to solve the needs of home life now and in the future.

http://www-static.cc.gatech.edu/fce/ahri/

the Aware Home

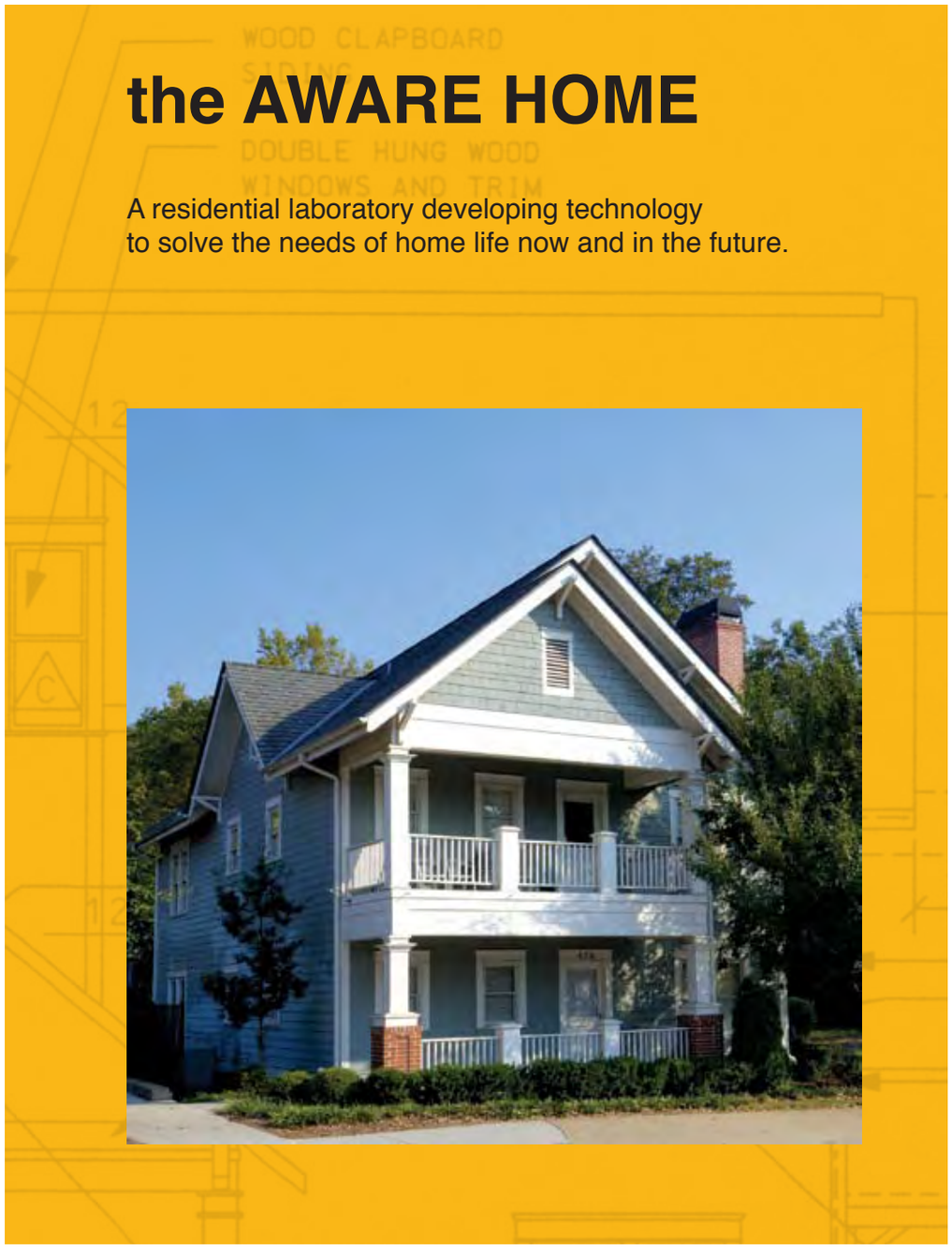
Memory Mirror

Quan T. Tran, Elizabeth D. Mynatt
quantt@cc.gatech.edu, mynatt@cc.gatech.edu

About This Project
There are particular household items that people use for specific tasks (e.g. taking a pill, feeding the cat) and these tasks are usually simple and brief. However, these tasks become difficult to recall performing when they are repeated often and are not part of a strict routine. Memory confusion arises between the repeated episodes of frequent tasks (e.g. “Did I take my vitamin today or was that yesterday?” , “Has anyone fed the fish?” , “Did I take pain medication an hour ago, or did I decide to wait a bit longer?”)

Memory mirror reflects the use of specified objects during a period of time (e.g. 24 hours of a day). As a person uses an item, it is visually posted to the mirror and is recorded in a history log. If an item was previously used, the mirror reflects details of the previous number of usages. The memory mirror also warns of possibly lost items that have yet to be returned.

Georgia Tech Aware Home



the Aware Home

Automatic Blind and Light System

Andy Rowan, Gregory D. Abowd
andy_rowan@yahoo.com, abowd@cc.gatech.edu

About This Project

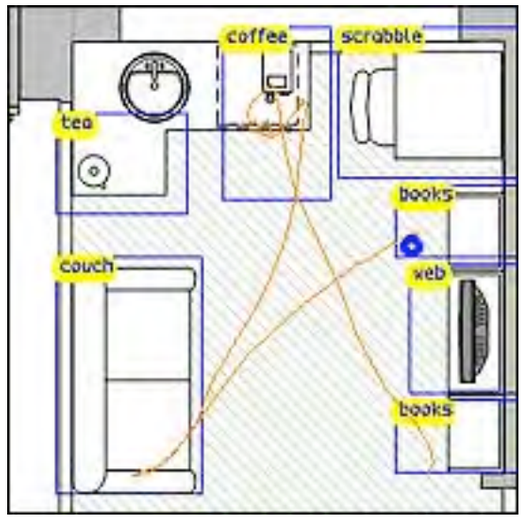
This project uses simple sensors and actuators to maintain the optimal lighting for a room that is also energy efficient. Light sensors detect what the current light settings are for the room and motion sensors detect if people are in the room. This information is combined with the time of day to determine the optimal light setting. Actuators automatically adjust the lights and the blinds to obtain the optimal light setting. This can save residents time by adjusting the lights for them, and save energy by turning off the lights if no one is there.

Next Steps:
We are looking into how this system will scale to cover the entire first floor of the Aware Home and look at providing a more usable and scalable Graphical User Interface.

http://www-static.cc.gatech.edu/fce/ahri/

Smart Homes - Cyber Crumbs

People need to have good spatial orientation (a sense of where they are in the home) as well as good wayfinding abilities (the ability to follow a planned route to a particular destination). The ability to do this normally depends on good eyesight and good memory, as well as proprioceptive (the ability to sense your body and its parts) and vestibular (the ability to maintain balance) senses. All of these abilities decline with age.



Cyber Crumbs are like breadcrumbs left to find your way; they make use of special tags worn by the user, and readers that are able to sense the location of the user. The reader is then able to feed back this information verbally through headphones or speaker, helping the user orient themselves within the home. It can also provide guidance with typical routes through the home.

http://news.bbc.co.uk/1/hi/technology/3144405.stm

Robot bears watch over elderly

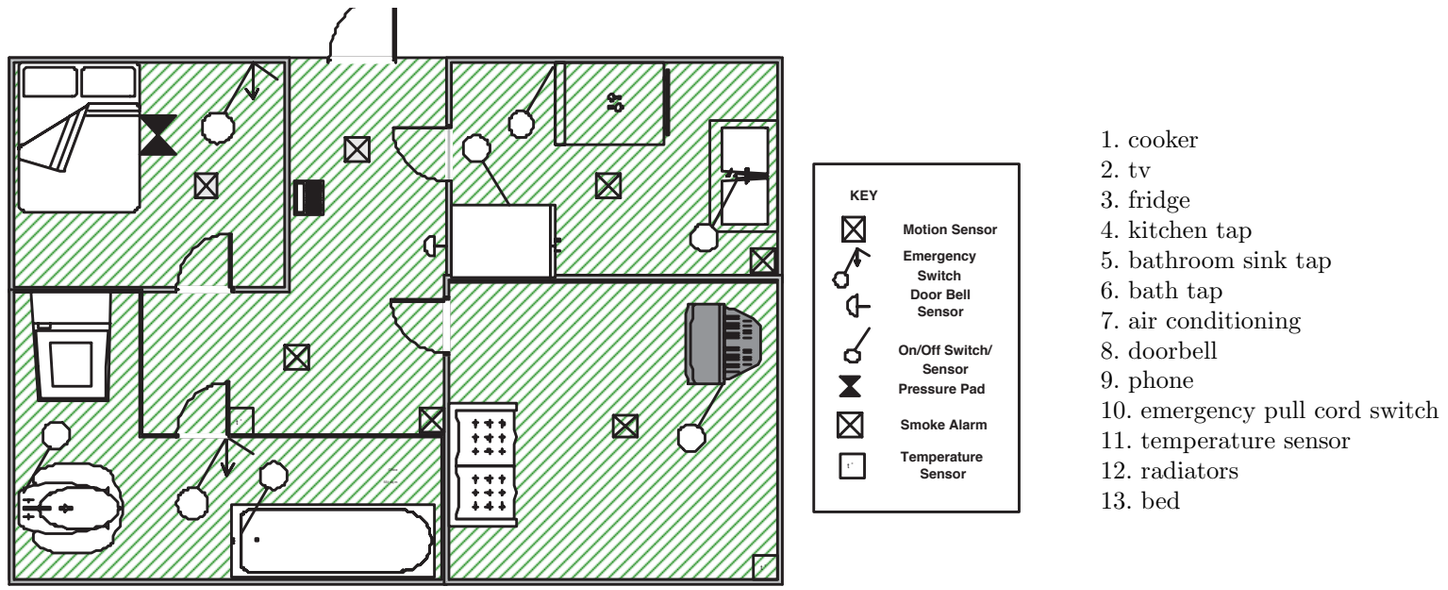
The bears monitor patients' response times to spoken questions. They record how long they spend performing various tasks, before relaying conclusions to staff or alerting them to unexpected changes.
The voice recognition interface helps remove the barriers presented by using traditional computers for similar tasks.
The fur-covered robotic assistant, simply known as Teddy, hides a microcomputer and a local network connection.

The Sincere Kourien
Hi-tech retirement home

http://news.bbc.co.uk/1/hi/sci/tech/1829021.stm

What does this all have to do with AI and reasoning?

- Let’s consider an intelligent monitoring system that can detect when an undesirable situation may be developing on the home (e.g., hazards, security threat - cooking, fireplace, water, doors, windows)
- Let’s further assume the home is equipped with networked embedded sensors (smoke, movement, temperature, moisture, human presence, ...)



1. cooker
2. tv
3. fridge
4. kitchen tap
5. bathroom sink tap
6. bath tap
7. air conditioning
8. doorbell
9. phone
10. emergency pull cord switch
11. temperature sensor
12. radiators
13. bed

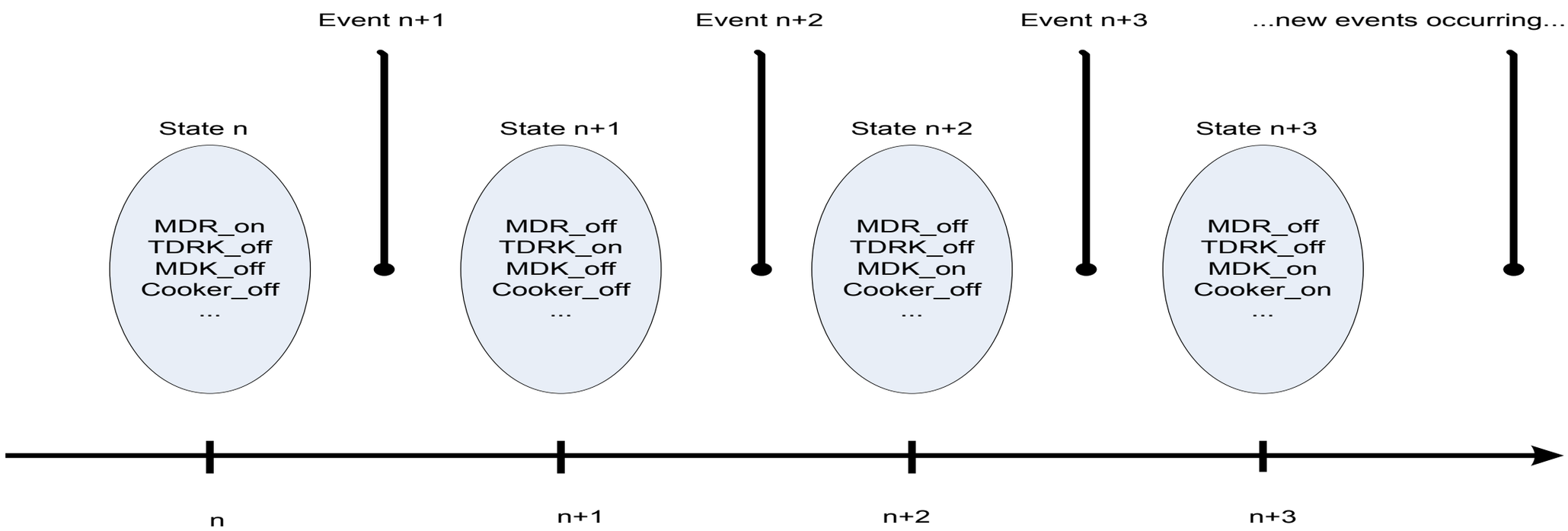
http://www.infj.ulst.ac.uk/~jcaug/icost2004.pdf

Observed Event Sequences

```
0 at_kitchen_on          1 cooker_on
2 at_reception_on        3 no_event
4 at_toilet_on           5 tapSinkBathroom_on
6 no_event               7 tapSinkBathroom_off
8 no_event               9 at_reception_on
11 at_bedroom_on         11 inbed_on
12 no_event              13 no_event
...                      ...
```

Question: is this a ‘normal’ sequence of user activities or should a health worker visit this person?

Representation of the Home Dynamics



State based representation

Reasoning in Smart Homes

We want to be able to reason about

- if a state represents hazardous or dangerous situation
- if a sequence of events is ‘normal’ or indicates unusual human behavior
- possible future states and events
- ...

What is Reasoning?

- Def: The ability to use knowledge to draw new conclusions about the world
- Without reasoning we simply recall stored information (database)
- Without reasoning there is no intelligence
- Reasoning is a process that can be implemented by an algorithms
- Reasoning = inference

3 Modes of Reasoning

- Deduction
- Induction
- Abduction

(see http://web.cs.mun.ca/~ulf/gloss/logic.html)

1. Deduction

Facts:

a = the battery is flat

b = the lights won’t work

Axioms:

∀x: a(x) -> b(x)

a(my car)

Deduction:

b(my car)

(Example from Alan’s book)

Deduction

Deduction uses a rule, called `modus ponens', of the following general form in proofs:

(S1) If A is true then B is true

And:

(S2) A is true

Therefore:

(S3) B is true

Deduction is independent of the meaning of statements S1, S2, S3. Deduction works for all statements of the corresponding form.

2. Induction

(S1) all ravens we see are black

Therefore:

(S2) all ravens are black

Induction

- **Generalization** from cases seen to infer information about cases unseen
- Inductive inferences are those that project beyond the known data, as in the paradigm of generalizing that **all ravens are black** (how do we 'know' this? we can assume it is true because nobody has seen a non-black raven)
- Forms the basis of machine learning

Induction

- «It embodies the scientific method in ideal form: From the observation of single facts one obtains by induction a general law, and by deduction one obtains other specific facts from the general law» [Bertrand Russel about Isaac Newton's "Mathematical Foundations of Natural Sciences"]
- Natural laws are not verified (proven, deduced) but only shown to be true for more and more specific cases. Which makes them more and more certain. But a single counter-example (a white raven) can falsify the ``law''. What is required from a natural law (instead of verification) is that it explains (allows to deduce) correctly all the already known single cases and that makes (allows to deduce) a prediction on yet unknown cases.
- (Historical sciences do not strive for generalizations, but for explaining historic events - they use non-inductive reduction.)

Induction is a Special form of Reduction

The general Reduction is the derivation from B to A "against" the direction of the implication.

(S1) If A is true then B is true

And:

(S2) B is true

Therefore:

(S3) A is true

Induction is a special case of reduction where A is a generalization of B

3. Abduction

(S1) A friend is annoyed with you

(S2) You are late for your appointment

Inference:

(S3) Your lateness caused his anger

Abduction is unreliable

It provides a best guess given the evidence available

Abduction

- Abduction accepts a conclusion on the grounds that it explains the available evidence.
- The term was introduced by Charles Peirce to describe an inference pattern sometimes called 'hypothesis' or 'inference to the best explanation'.
- He used the example of arriving at a Turkish seaport and observing a man on horseback surrounded by horsemen holding a canopy over his head. He inferred that this was the governor of the province since he could think of no other figure who would be so greatly honoured.

Abduction vs Deduction

- Structurally similar
- The semantics and the implementation of abduction cannot be reduced to those for deduction, as explanation cannot be reduced to implication.
- «justifying the postulation of unobservable phenomena on the strength of explanations they afford of observable phenomena» «Accepting a statement because it is the best available explanation of one's evidence; deriving the conclusion that best explains one's premisses.
- Applications include fault diagnosis, plan formation and default reasoning.

Reasoning for Smart Homes

- Spatio-temporal reasoning
  - temporal order and place of events
  - moving? trajectory?
- Causal reasoning
  - relation between actions performed by human and the resulting state
  - predict possible outcomes of behavior
- Planning
  - identify actions to achieve desired outcome
  - home automation

Knowledge Representation Schemes

- First order predicate logic (FOL)

  is_person(Jane) $\wedge$ meeting(Jane,10am,tax_office)

- may have probabilities, weights ...

  meeting(Jane,time,tax_office), time=10am 75%, time=11am 25%

- Frames (a bit like objects)

  Meeting { who:Jane, when:10am, where: tax_office}

- Semantic Web - triples/RDF

  id#15 class Person, id#15 name 'Jane',

  id#37 class Meeting, id#37 time '10am', id#37 who id#15

Metrics for knowledge representation schemes

- Expressiveness: type and level of detail of knowledge?
- Effectiveness: means of inferring new knowledge from old? should exist
- Efficiency: inference mechanisms efficient (space and time complexity)?
- Explicitness: explanation of inferences and justifications for inferred knowledge?

Finlay & Dix, AI, p 13.

Logic as Knowledge Representation Scheme

- Logic is extremely powerful
- Say what's true, not how to use it
  - $\forall$ x,y ($\exists$z Parent(x,z) $\wedge$ Parent(z,y)) $\Leftrightarrow$ Grandparent(x,y)
    - Given parents, find grandparents
    - Given grandparents, find parents
- But theorem provers for logic tend to be too inefficient
- FOL is undecidable

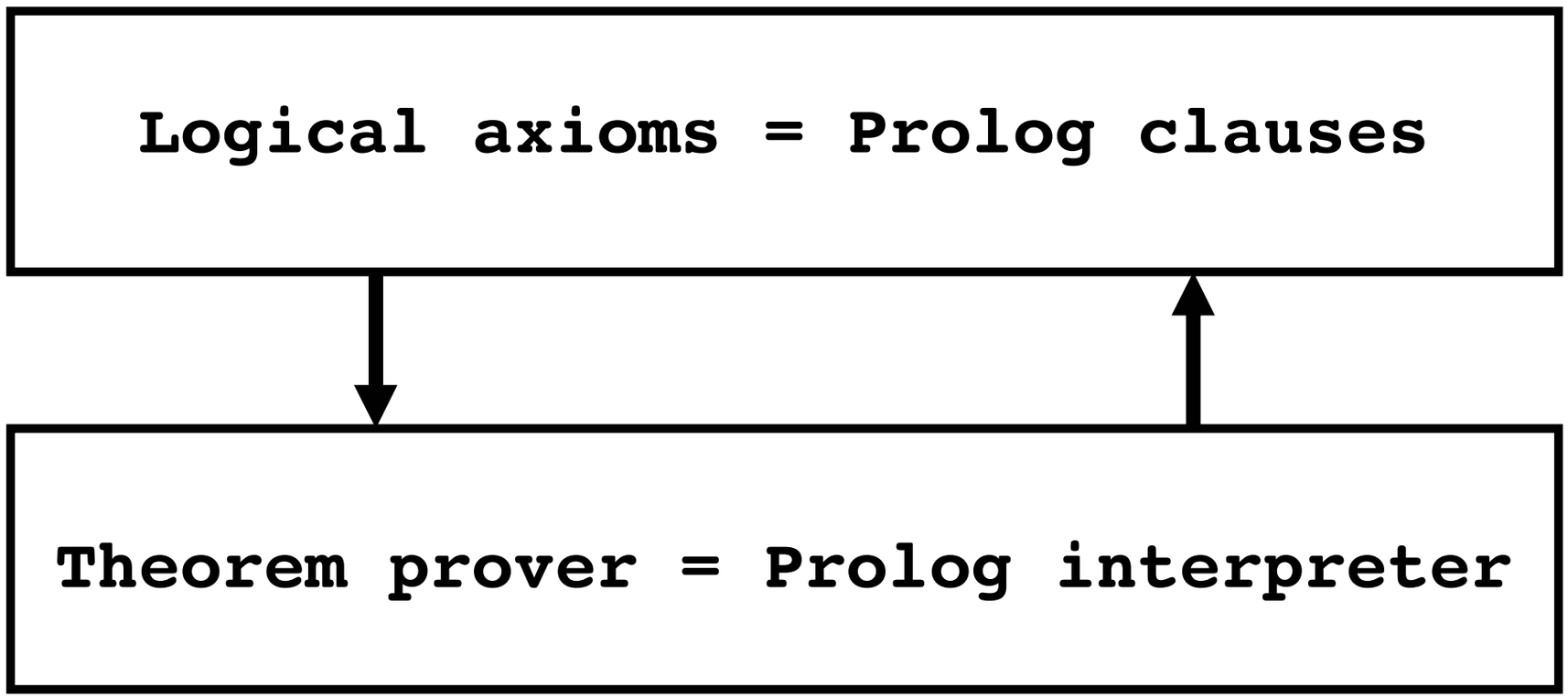
Solution: Horn Logic

- To regain practicality:
  - limit the language
  - simplify the proof algorithm

- Horn clauses
  - $P_1 \wedge \ldots \wedge P_n \rightarrow Q$ (Horn logic)
  - If $P_1 \ldots P_1$ Then Q (Rule-based system)
  - Q :- P_1, ..., P_n (Prolog)

Prolog

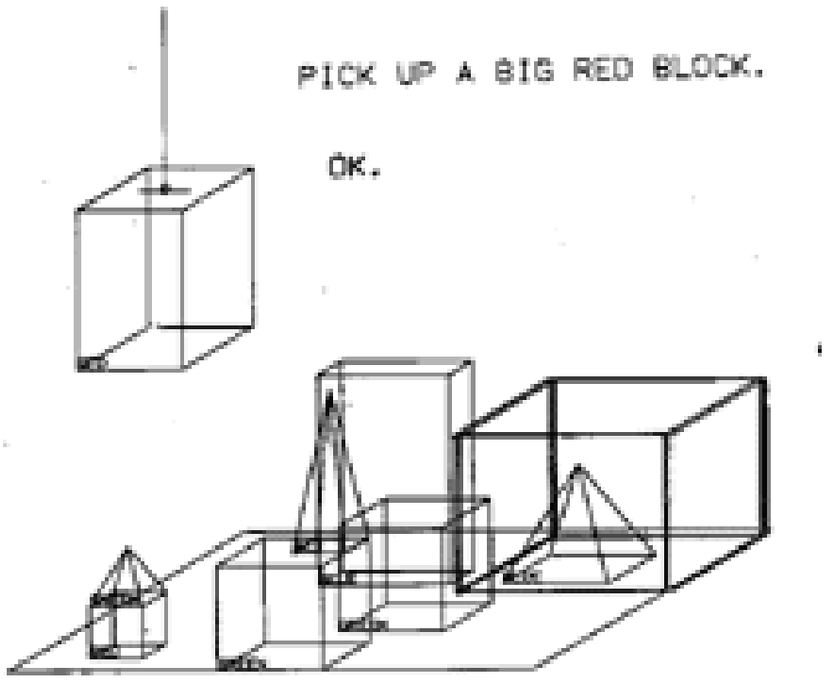
- **Prolog = Programming in Logic**



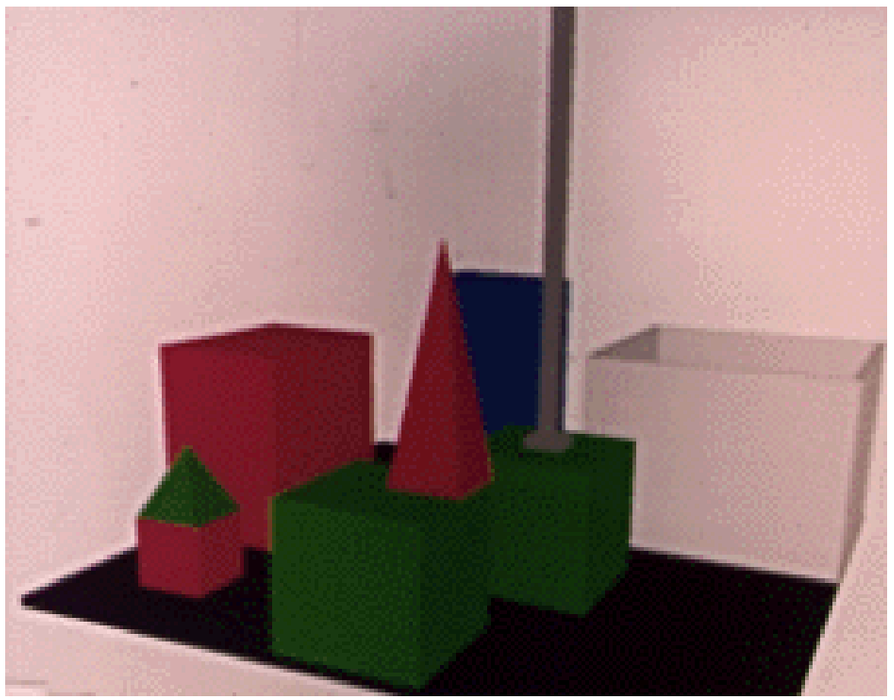
Horn Clauses in Prolog

Q :- P_1, ..., P_n	(Rule)
Q	(Fact)
:- P	(Goal, often written as ?- P)

SHRDLU (Winograd 68-70)



Original screen display



Later color rendering

http://hci.stanford.edu/~winograd/shrdlu/

The dialog that was used as a SHRDLU demo

Person: PICK UP A BIG RED BLOCK.

Computer: OK. (does it)

Person: GRASP THE PYRAMID.

Computer: I DON'T UNDERSTAND WHICH PYRAMID YOU MEAN.

Person: FIND A BLOCK WHICH IS TALLER THAN THE ONE YOU ARE HOLDING AND PUT IT INTO THE BOX.

Computer: BY "IT", I ASSUME YOU MEAN THE BLOCK WHICH IS TALLER THAN THE ONE I AM HOLDING.

Computer: OK. (does it)

Person: WHAT DOES THE BOX CONTAIN?

Computer: THE BLUE PYRAMID AND THE BLUE BLOCK.

Person: WHAT IS THE PYRAMID SUPPORTED BY?

Computer: THE BOX.

Person: HOW MANY BLOCKS ARE NOT IN THE BOX?

Computer: FOUR OF THEM.

Person: IS AT LEAST ONE OF THEM NARROWER THAN THE ONE WHICH I TOLD YOU TO PICK UP?

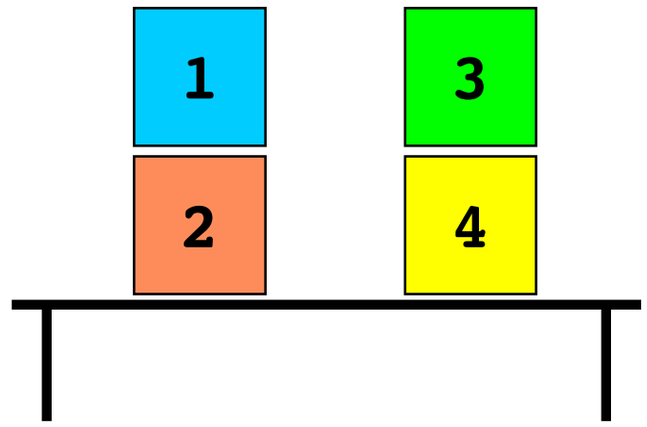
Computer: YES, THE RED CUBE.

Person: IS IT SUPPORTED?

Computer: YES, BY THE TABLE.

The Representation

```
block(block1).
block(block2).
block(block3).
block(block4).

table(table1).

on(block1,block2).
on(block2,table1).
on(block3,block4).
on(block4,table1).
```



The Above-Rules

- X is above Y , iff
  - X is a block and
  - Y is a block and
  - X is on Y.
- X is above Y, iff
  - X is a block and
  - Y is a table and
  - X is on Y.
- X is above Y,iff
  - X is a block and
  - there is a block Z and X is on Z and
  - Z is above Y.

The Above-Program

- `above(X,Y) :- block(X), block(Y), on(X,Y).`
- `above(X,Y) :- block(X), table(Y), on(X,Y).`
- `above(X,Y) :- block(X), block(Z), on(X,Z), above(Z,Y).`

Some datatypes

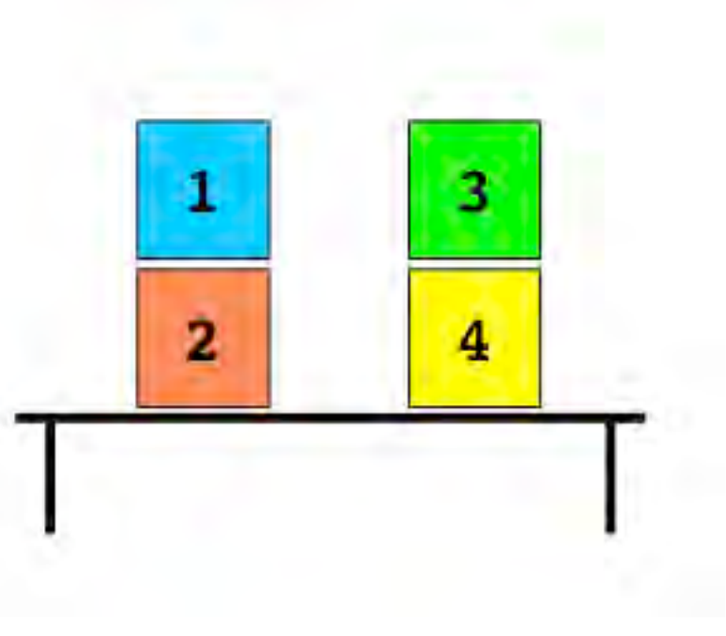
- **Atom:**
  - **starting with small character**
  - **Number**
    - **kai, kaiLuck, 4711, block1**
- **Variable:**
  - **starting with capitalized character or '_'**
    - **Block, _block**

On and above

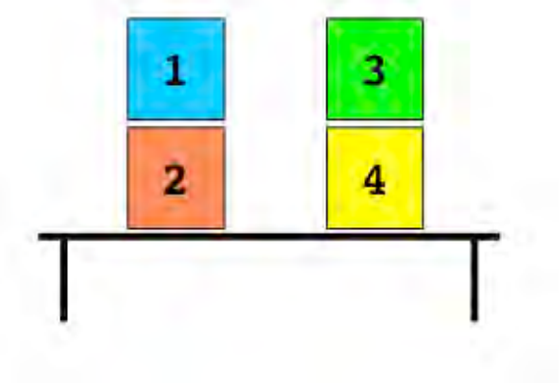
```
on(block1,block2).
on(block2,block3).
on(block3,block4).
on(block4,table1).


above(X,Y) :-
  on(X,Y).
above(X,Y) :-
  on(X,Z),
  above(Z,Y).
```

Database

```
?- on(block1,table1).
No
?- above(block1,table1).
Yes
```



More Queries



?- above(block1, block2)

yes

?- above(block2, block3)

no

?- above(block2, X)

X = block1

?- above(block3, X)

no

?- above(table1, X)

X = block1

X = block2

X = block3

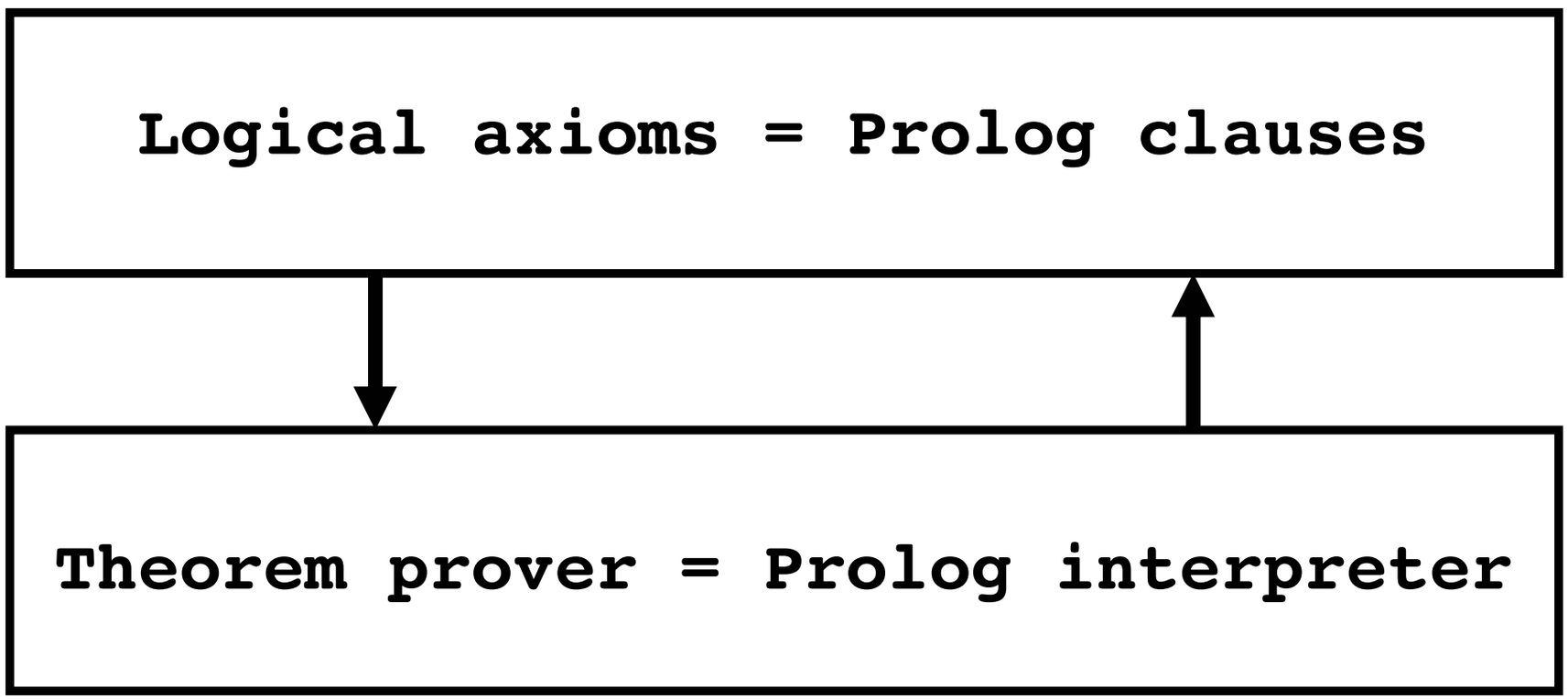
X = block4

?- above(X,Y)

[left as exercise]

Prolog

- **Prolog = Programming in Logic**



Inference: Backchaining

To prove a goal C:

Push C on a stack

Repeat until stack empty

Pop L off stack

Choose [with backup] a rule (or fact) whose consequent unifies with L

Push antecedents onto stack

If no match, fail [backup to last choice]

$$C :- A_1, ..., A_n$$

consequent antecedents

Example

Program

- father(a,b).
- mother(b,c).
- grandp(X,Z) :- parent(X,Y), parent(Y,Z).
- parent(X,Y) :- father(X,Y).
- parent(X,Y) : - mother(X,Y).

Example

Program

- father(a,b).
- mother(b,c).
- grandp(X,Z) :- parent(X,Y), parent(Y,Z).
- parent(X,Y) :- father(X,Y).
- parent(X,Y) : - mother(X,Y).

Prove

- grandp(G,c)

Example

Program

- father(a,b).
- mother(b,c).
- grandp(X,Z) :- parent(X,Y), parent(Y,Z).
- parent(X,Y) :- father(X,Y).
- parent(X,Y) : - mother(X,Y).

Prove

- grandp(G,c)
- parent(G,Y), parent(Y,c)

Example

Program

- father(a,b).
- mother(b,c).
- grandp(X,Z) :- parent(X,Y), parent(Y,Z).
- parent(X,Y) :- father(X,Y).
- parent(X,Y) : - mother(X,Y).

Prove

- grandp(G,c)
- parent(G,Y), parent(Y,c)
- father(G,Y), parent(Y,c)

Example

Program

- father(a,b).
- mother(b,c).
- grandp(X,Z) :- parent(X,Y), parent(Y,Z).
- parent(X,Y) :- father(X,Y).
- parent(X,Y) : - mother(X,Y).

Prove

- grandp(G,c)
- parent(G,Y), parent(Y,c)
- father(G,Y), parent(Y,c)
- father(a,b) G==a, Y==b

Example

Program

- father(a,b).
- mother(b,c).
- grandp(X,Z) :- parent(X,Y), parent(Y,Z).
- parent(X,Y) :- father(X,Y).
- parent(X,Y) : - mother(X,Y).

Prove

- grandp(G,c)
- parent(G,Y), parent(Y,c)
- father(G,Y), parent(Y,c)
- father(a,b) G==a, Y==b
- parent(Y,c)

Example

Program

- father(a,b).
- mother(b,c).
- grandp(X,Z) :- parent(X,Y), parent(Y,Z).
- parent(X,Y) :- father(X,Y).
- parent(X,Y) : - mother(X,Y).

Prove

- grandp(G,c)
- parent(G,Y), parent(Y,c)
- father(G,Y), parent(Y,c)
- father(a,b) G==a, Y==b
- parent(Y,c)
- father(Y,c) fail (backtrack)

Example

Program

- father(a,b).
- mother(b,c).
- grandp(X,Z) :- parent(X,Y), parent(Y,Z).
- parent(X,Y) :- father(X,Y).
- parent(X,Y) : - mother(X,Y).

Prove

- grandp(G,c)
- parent(G,Y), parent(Y,c)
- father(G,Y), parent(Y,c)
- father(a,b) G==a, Y==b
- parent(Y,c)
- father(Y,c) fail (backtrack)
- mother(Y,c) success

Example

Program

- father(a,b).
- mother(b,c).
- grandp(X,Z) :- parent(X,Y), parent(Y,Z).
- parent(X,Y) :- father(X,Y).
- parent(X,Y) : - mother(X,Y).

Prove

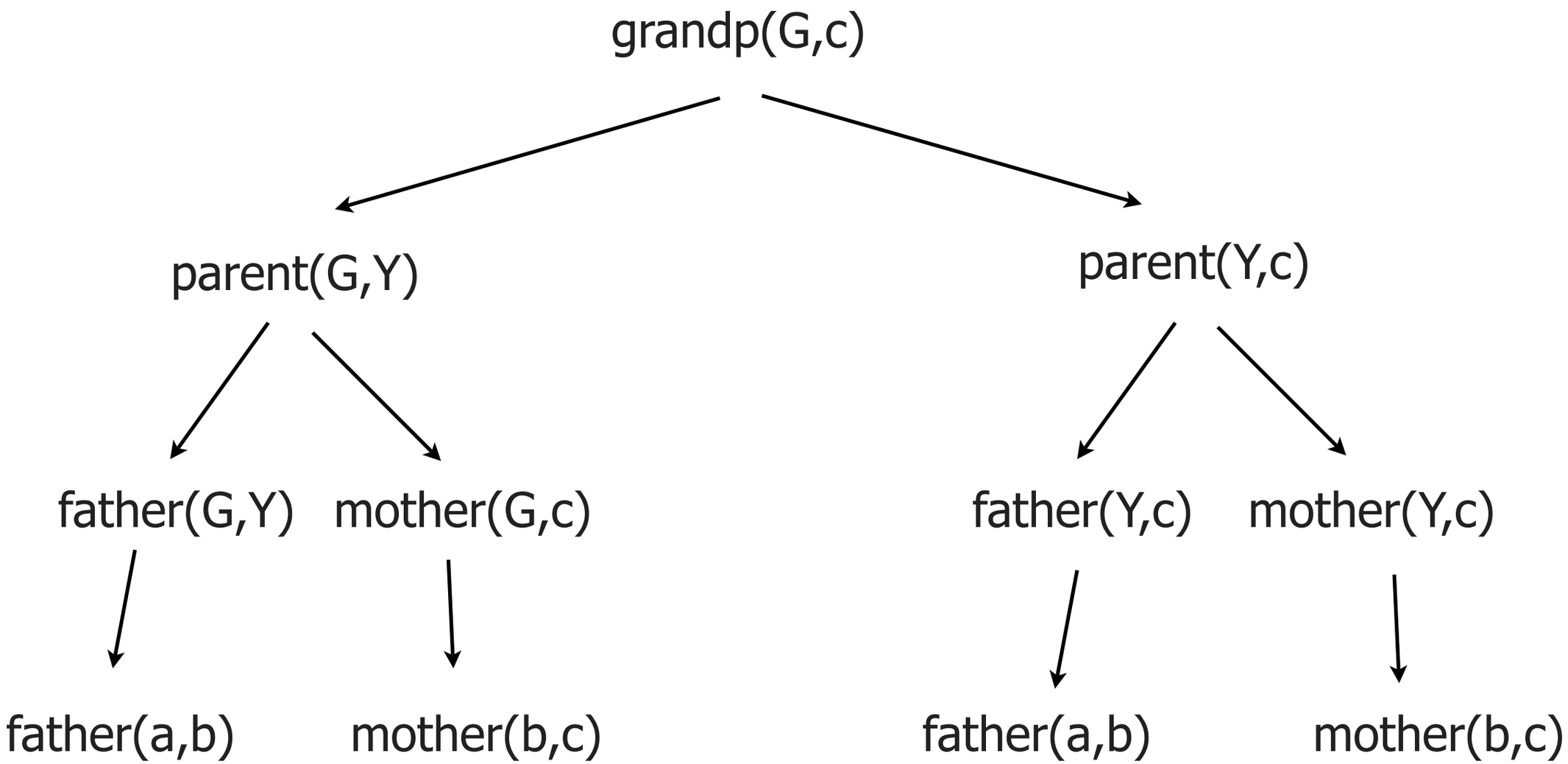
- grandp(G,c)
- parent(G,Y), parent(Y,c)
- father(G,Y), parent(Y,c)
- father(a,b) G==a, Y==b
- parent(Y,c)
- father(Y,c) fail (backtrack)
- mother(Y,c) success

Solution

- grandp(a,c)

Proof Tree

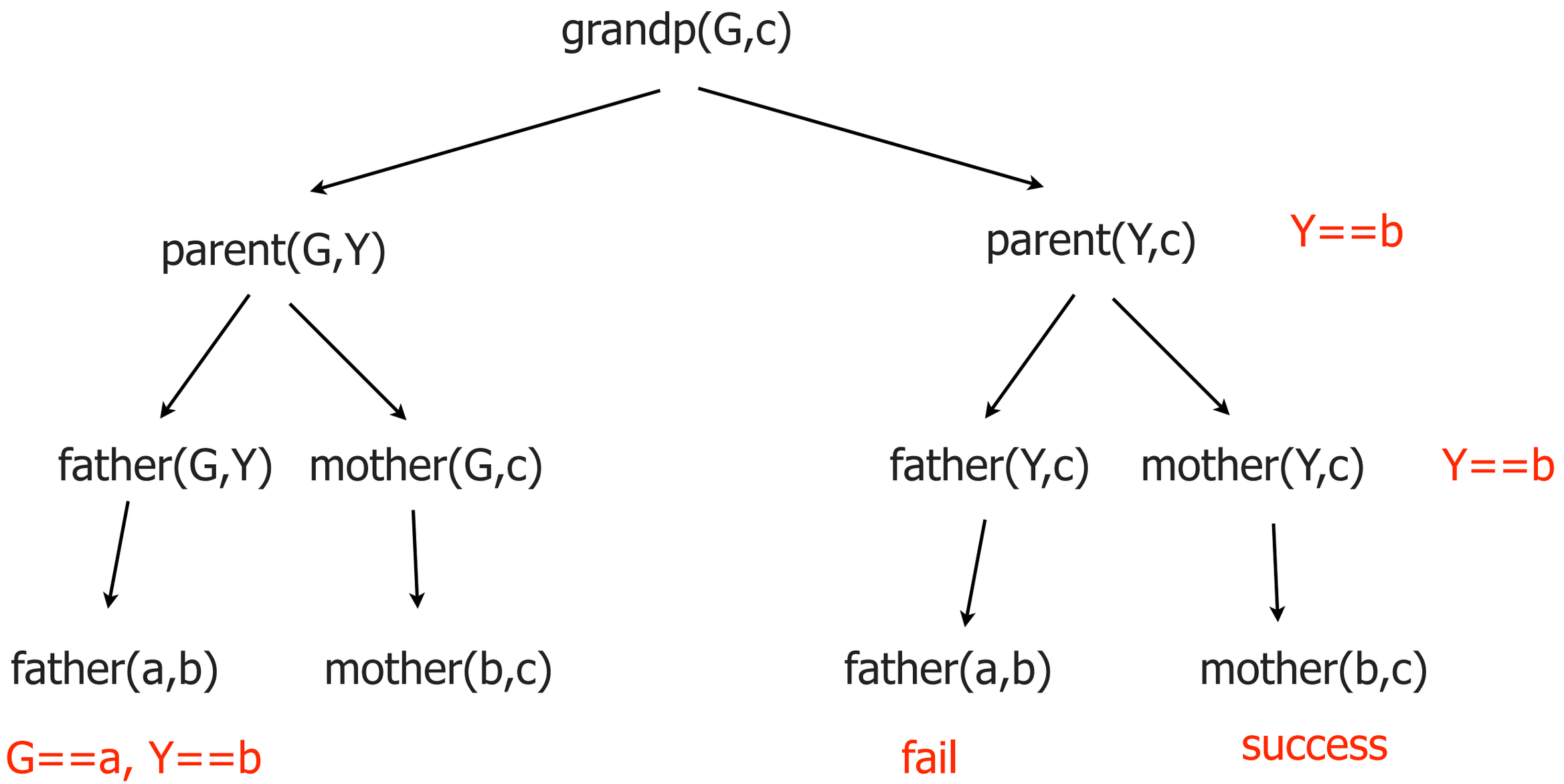
- father(a,b).
- mother(b,c).
- grandp(X,Z) :- parent(X,Y), parent(Y,Z).
- parent(X,Y) :- father(X,Y).
- parent(X,Y) : - mother(X,Y).



depth-first search / left to right

Proof Tree

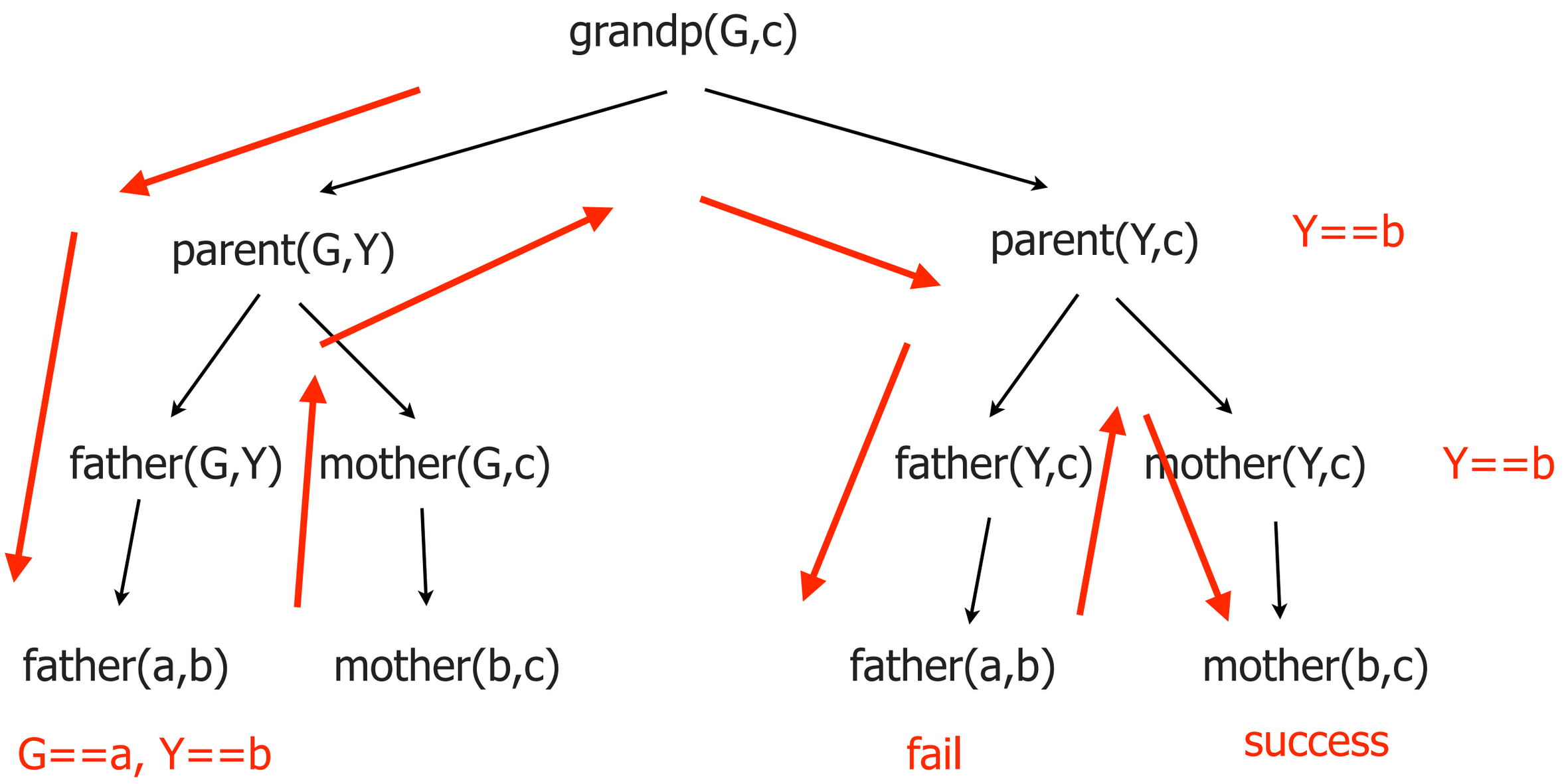
- father(a,b).
- mother(b,c).
- grandp(X,Z) :- parent(X,Y), parent(Y,Z).
- parent(X,Y) :- father(X,Y).
- parent(X,Y) : - mother(X,Y).



depth-first search / left to right

Proof Tree

- father(a,b).
- mother(b,c).
- grandp(X,Z) :- parent(X,Y), parent(Y,Z).
- parent(X,Y) :- father(X,Y).
- parent(X,Y) : - mother(X,Y).



depth-first search / left to right

Problems with Prolog

- Negation cannot be expressed
- Rule order is significant - not pure logic
- Prolog reasoning algorithm is incomplete:
  - Proofs may not terminate - infinite recursive loops

```
relative(a,b).
relative(X,Z) :- relative(X,Y), relative(Y,Z).
relative(c,d).
```

We did you learn?

- Knowledge representation and reasoning important for smart home scenarios
- 3 reasoning modes
- Prolog is an efficient logic-based reasoning system
- Prolog is based on horn-logic
- Prolog reasoning algorithms (backtracking)
- Prolog proof trees

Spatial & Temporal Reasoning

CSc355 - Artificial Intelligence

Gerd Kortuem

What did we learn yesterday?

- Rule-based representation of knowledge about the world (e.g., smart homes)
- Describing what is true in the world
- Making inferences about events, relations between objects etc.
- Usage: generalizations, predictions, planning etc.
- Prolog as efficient (yet incomplete) representation and reasoning system

Temporal Reasoning is Important

- "While Nigel and Joe were driving from Lancaster to Manchester, Joe was listening to a CD by Lily Allen. Just after Preston a white van cut in front of Nigel's car forcing him to break hard and causing Joe to spill his coffee."
- What is the temporal relationship between Joe listening to music and Joe spilling coffee? Relationship must be inferred, it is not stated in the text.

Temporal Reasoning is Important

Safety Aware Barrel

Don't shake me so much!

I shouldn't be standing here!

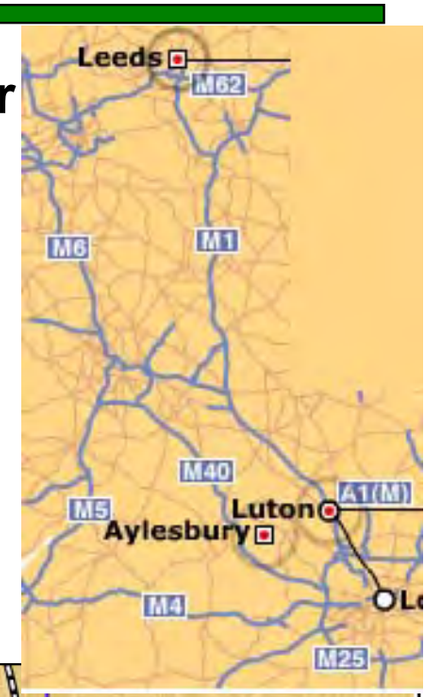
I am too hot!

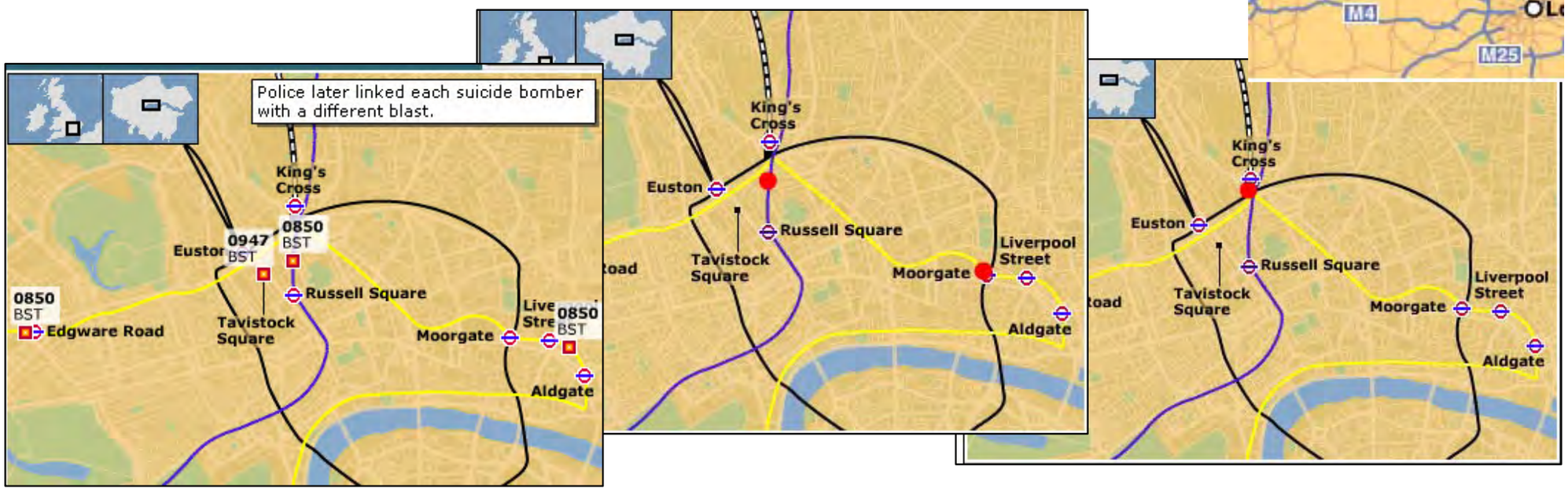
I have been sitting here too long!

I have been tampered with!

Temporal Issues in Forensics

- Information regarding events unfolds in no specific order
- Temporal information may be both qualitative and quantitative
- Information may be inconsistent/incorrect
- Information may contain hidden patterns or temporal relations that can help identify missing links
- An automated tool for temporal knowledge representation, verification and reasoning is required





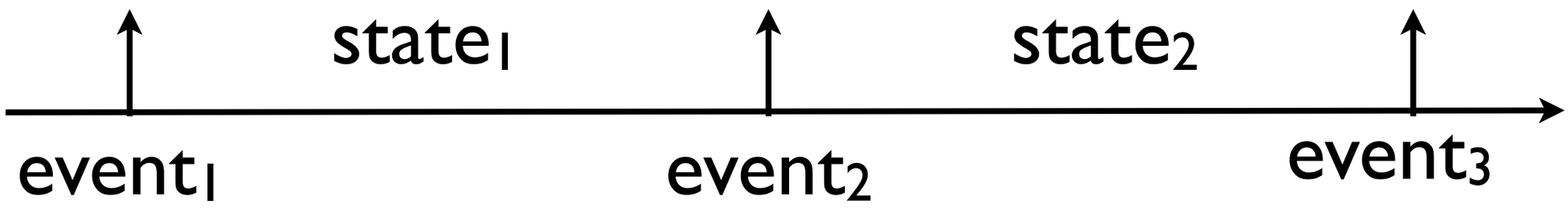
Spatial Reason is Important

Barrels use ultrasound transceivers for direct measurement of small distances.
Distances between far apart barrels must be inferred.

Spatial Reasoning Methods

Change-based Representations

- Represent time implicitly
- Represent things that bring about change: events, actions
- E.g., Situation calculus, event calculus
- Limitations:
  - hard to express delayed effects (*when the electrical cooker is turned on, the surface will be hot after 2 minutes*)
  - hard to express concurrent actions, actions that happen at a particular time (14:00), actions with a duration

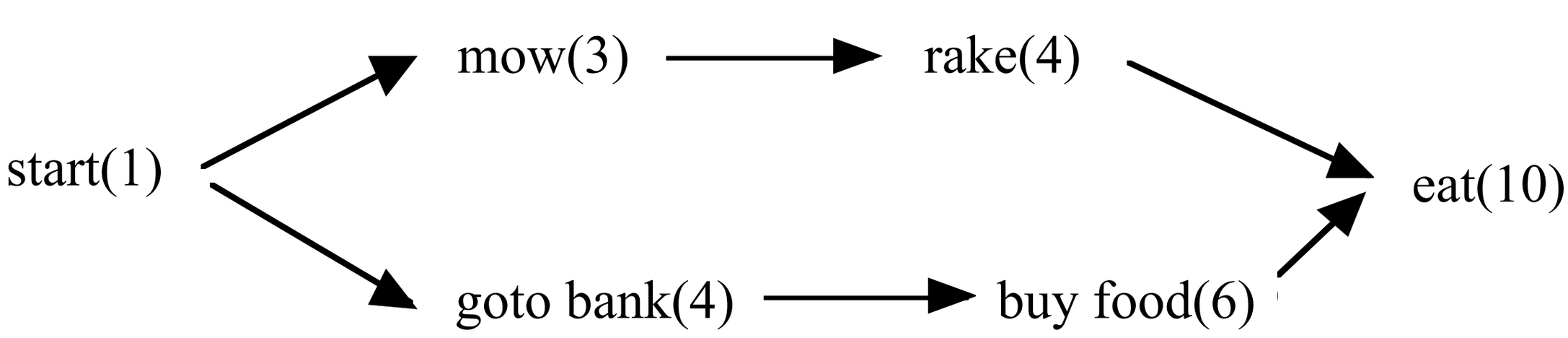


Time-based Approaches

- Represent time explicitly
- Many possibilities:
  - Which time elements: **points** or **intervals**?
  - Is time **linear** or **branching**?
  - Is time **discrete** or **continuous**?
  - Is time **bound** or **unbounded**?

First Attempt: Pseudo Times

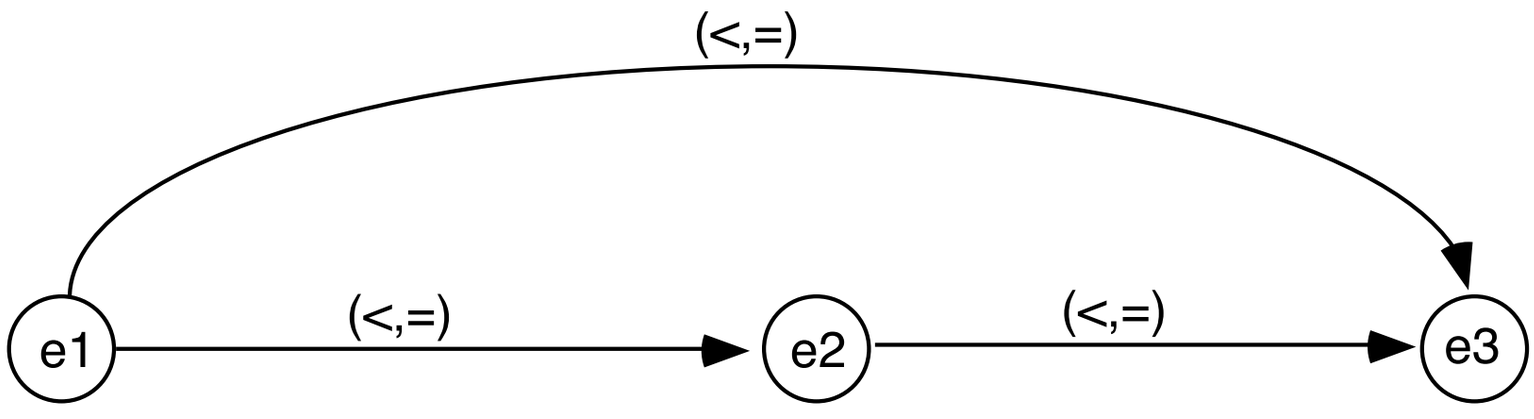
- Assign pseudo times to events to indicate temporal order
  - start, then mow, then, rake, then eat
  - start, then goto bank, then buy food, then eat



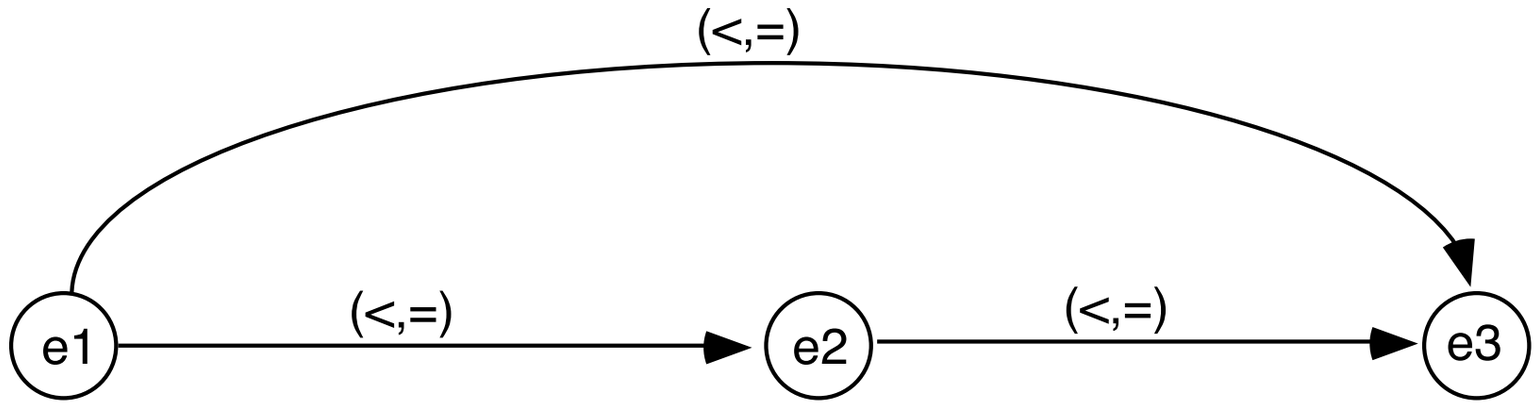
  - Problem: implicit temporal relation between mow and goto bank

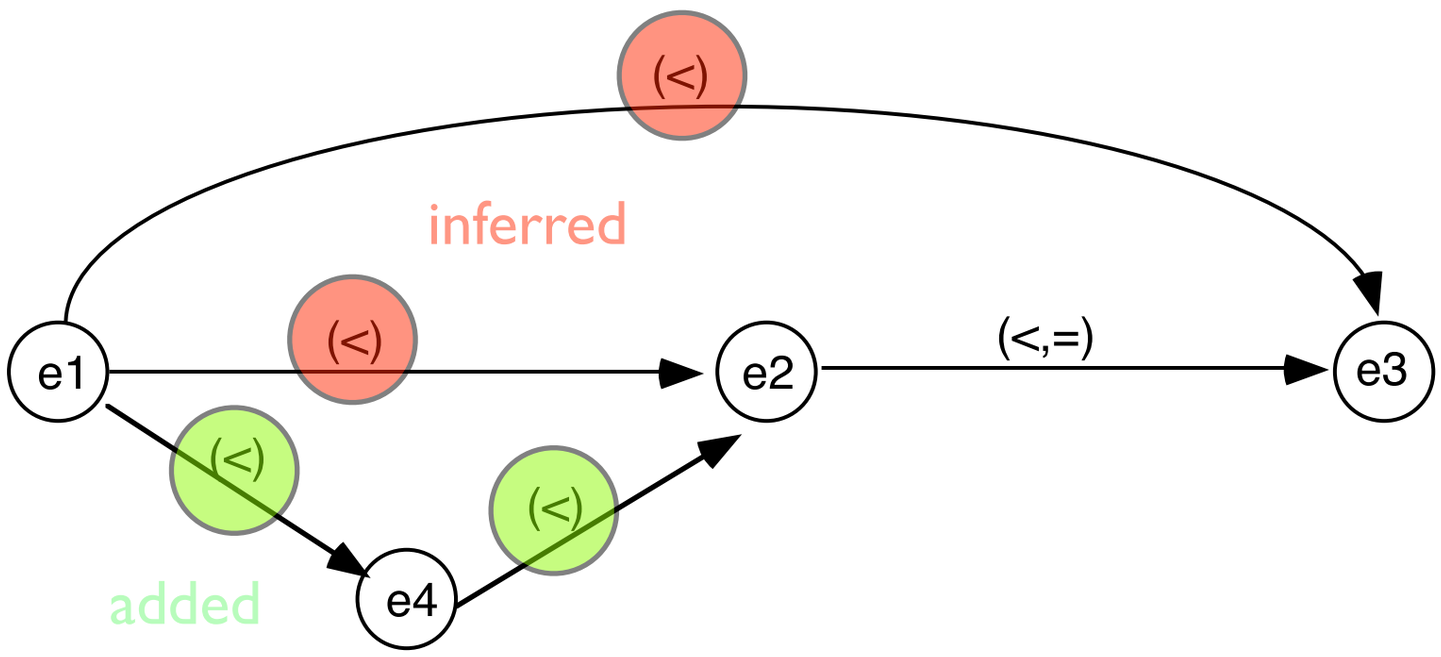
A Simple Point-Based Representation

- Points and temporal relation (<,=,>)
- Graph representation
  - nodes represent time point
  - edges indicate temporal relationships between points
  - each each is labeled with a set indicating disjunction



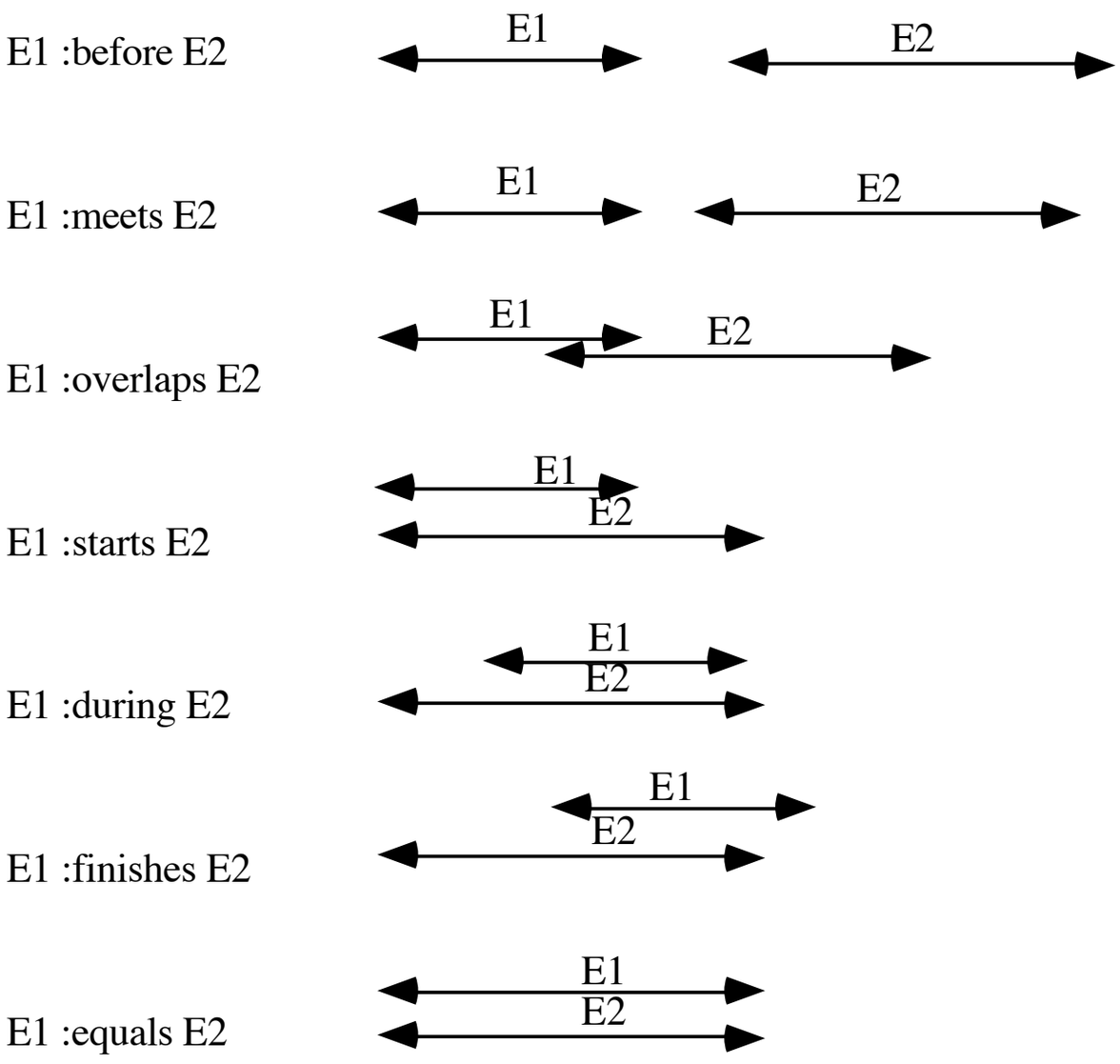
Temporal Reasoning using Constraint-Propagation





Make use of transitive relations ($e_1 < e_2$ and $e_2 < e_3$ then $e_1 < e_3$)

James F. Allen's Interval Algebra



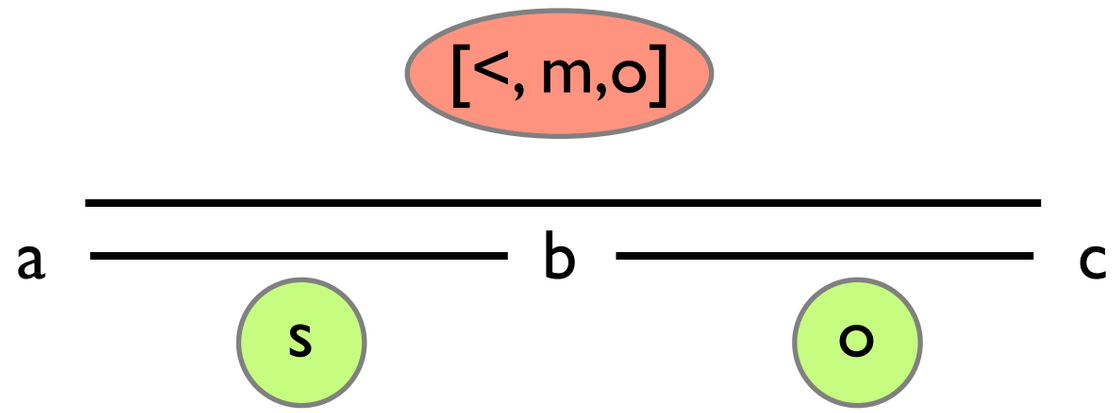
13 interval relationships

2^{13} possible (disjunctive) constraints between any two intervals

Interval Relations are Transitive

If we know a and b are in relation R1 and b and c are in R2, then we can restrict the set of possible relations for a and c.

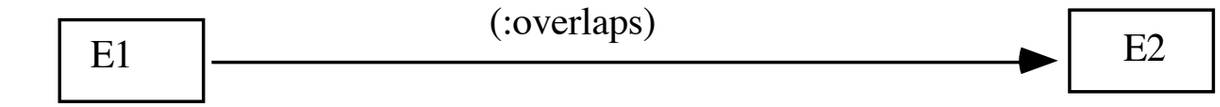
- Given: s(a, b) and o(b, c)
- Infer: <(a, c) or m(a, c) or o(a, c)

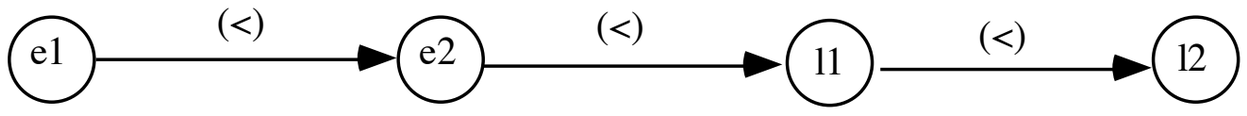


Transitivity Table

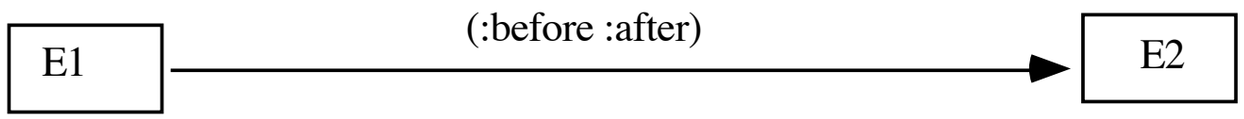
B r2 C A r1 B	<	>	d	di	o	oi	m	mi	s	si	f	fi
"before" <	<	no info	< o m d s	<	<	< o m d s	<	< o m d s	<	<	< o m d s	<
"after" >	no info	>	> oi mi d f	>	> oi mi d f	>	> oi mi d f	>	> oi mi d f	>	>	>
"during" d	<	>	d	no info	< o m d s	> oi mi d f	<	>	d	> oi mi d f	d	< o m d s
"contains" di	< o m di fi	> oi di mi si	o oi dur con =	di	o di fi	oi di si	o di fi	oi di si	di fi o	di	di si oi	di
"overlaps" o	<	> oi di mi si	o d s	< o m di fi	< o m	o oi dur con =	<	oi di si	o	di fi o	d s o	< o m
"over-lapped-by" oi	< o m di fi	>	oi d f	> oi mi di si	o oi dur con =	> oi mi	o di fi	>	oi d f	oi > mi	oi	oi di si
"meets" m	<	> oi mi di si	o d s	<	<	o d s	<	f fi =	m	m	d s o	<
"met-by" mi	< o m di fi	>	oi d f	>	oi d f	>	s si =	>	d f oi	>	mi	mi
"starts" s	<	>	d	< o m di fi	< o m	oi d f	<	mi	s	s si =	d	< m o
"started by" si	< o m di fi	>	oi d f	di	o di fi	oi	o di fi	mi	s si =	si	oi	di
"finishes" f	<	>	d	> oi mi di si	o d s	> oi mi	m	>	d	> oi mi	f	f fi =
"finished-by" fi	<	> oi mi di si	o d s	di	o	oi di si	m	si oi di	o	di	f fi =	fi

Interval Algebra is more powerful than Point-based Representations





‘overlaps’ can be represented with point relations



(‘before’ or ‘after’) cannot

Reasoning with Intervals

- A NP-complete algorithm exists that can reason over Allen's interval representations.
- Allen himself developed a heuristic O(n3) algorithm. This algorithm is complete with respect to any situation that can be expressed as point-based constraints.
- The algorithm also handles more complex situations but without the guarantee of completeness.

Allen's Algorithm

Initialize a queue Q with all constrained edges

Do until Q is empty

e = pop (Q)

for all triangles (e, e_1, e_2) formed by e do

update e_1 using (e and e_2)

update e_2 using (e and e_1)

if e_i becomes null return INCONSISTENCY

else if e_i gets further constrained push(e_i, Q)

Allen's algorithm does not return correct answer for full Interval Algebra: not all inconsistencies are detected [Approximate algorithm]

A Temporal Reasoning Problem

Input:

Meeting_A should be {b | a} Person_A office hour

Meeting_A should be {a} Person_B office hour

Meeting_A should be {b} Person_C office hour

Meeting_A should have office hour {overlap} that of Person_B

Person_B should have office hour {overlap} that of Person_C

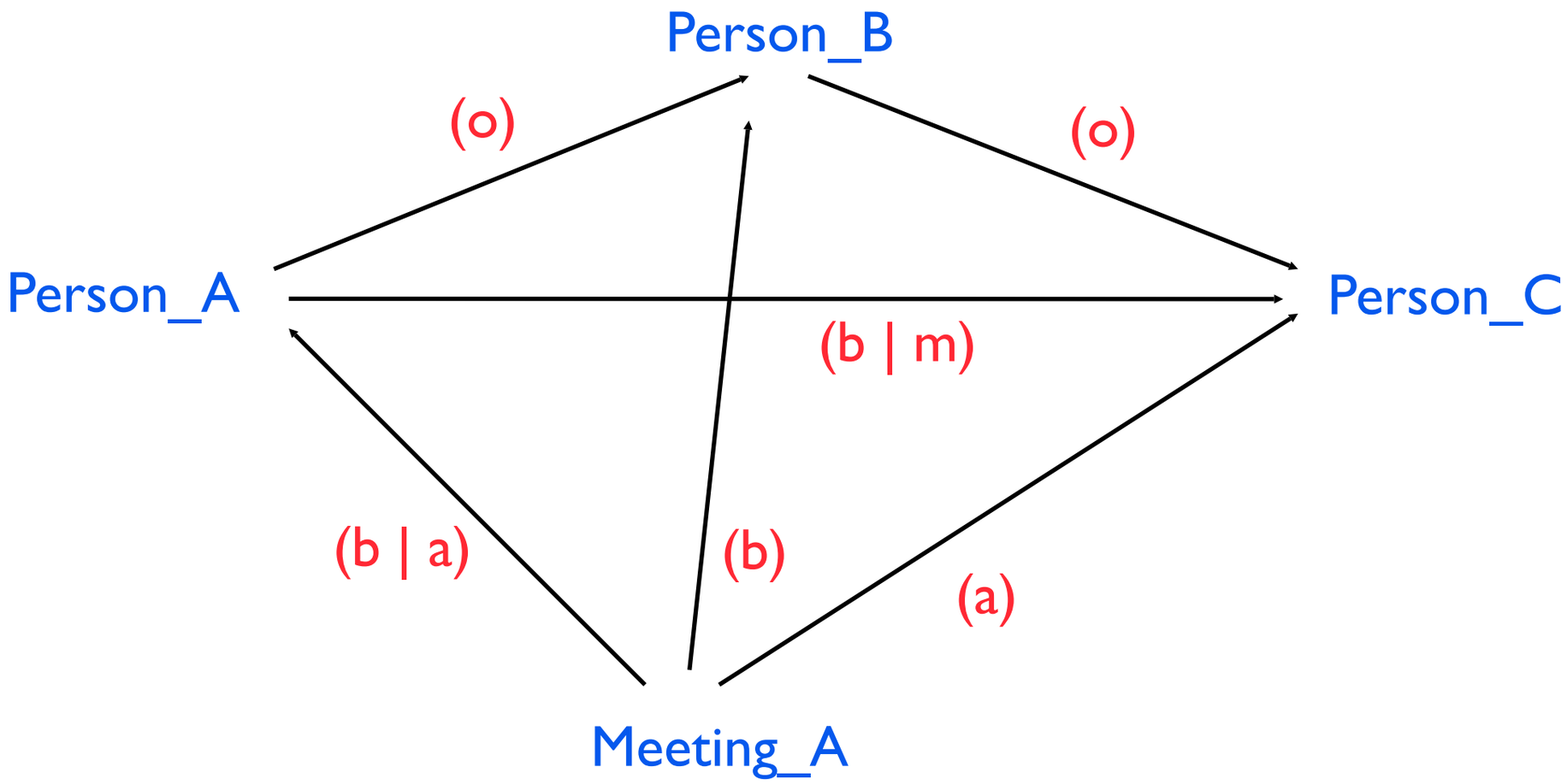
Person_A should have office hour {b | m} that of Person_C

Question 1: Is the information consistent? (decision problem)

Question 2: Develop a scenario, if it is consistent

Does a solution exist: No! [2, 3, and 5 contradicts]

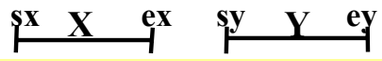
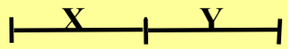
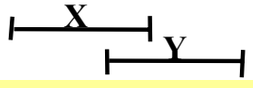
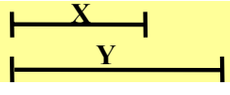
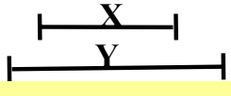
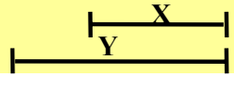
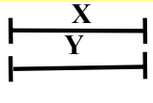
Step 1: Add known Relations



Step 2: Infer new Relations

Pointisable Logic

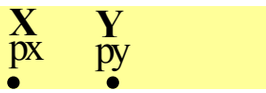
- Def: a tractable (not NP complete) subclass of Allen's interval logic
- Example: Point-Interval Logic (PIL)
- Uses limited set of interval-interval relationships:

Before	X	<	Y	ex < sy	
Meets	X	m	Y	ex = sy	
Overlaps	X	o	Y	sx < sy; sy < ex; ex < ey	
Starts	X	s	Y	sx = sy; ex < ey	
During	X	d	Y	sx > sy; ex < ey	
Finishes	X	f	Y	sy < sx; ey = ex	
Equals	X	=	Y	sx = sy; ex = ey	

Point-Interval Logic (PIL)

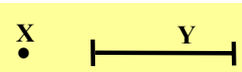
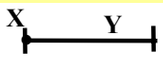
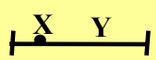
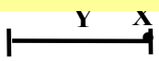
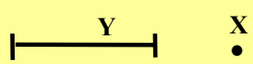
- Point-point relationships

X = [px] and Y = [py] with sx = ex and sy = ey

Before	X < Y	px < py	
Equals	X = Y	px = py	

- Point-interval relationships

X = [px] and Y = [sy,ey] with px = sx and sy < ey

Before	X < Y	px < sy	
Starts	X s Y	px = sy	
During	X d Y	sy < px < ey	
Finishes	X f Y	px = ey	
Before	Y < X	ey < px	

Point-Interval Logic (PIL)

- Quantitative temporal information

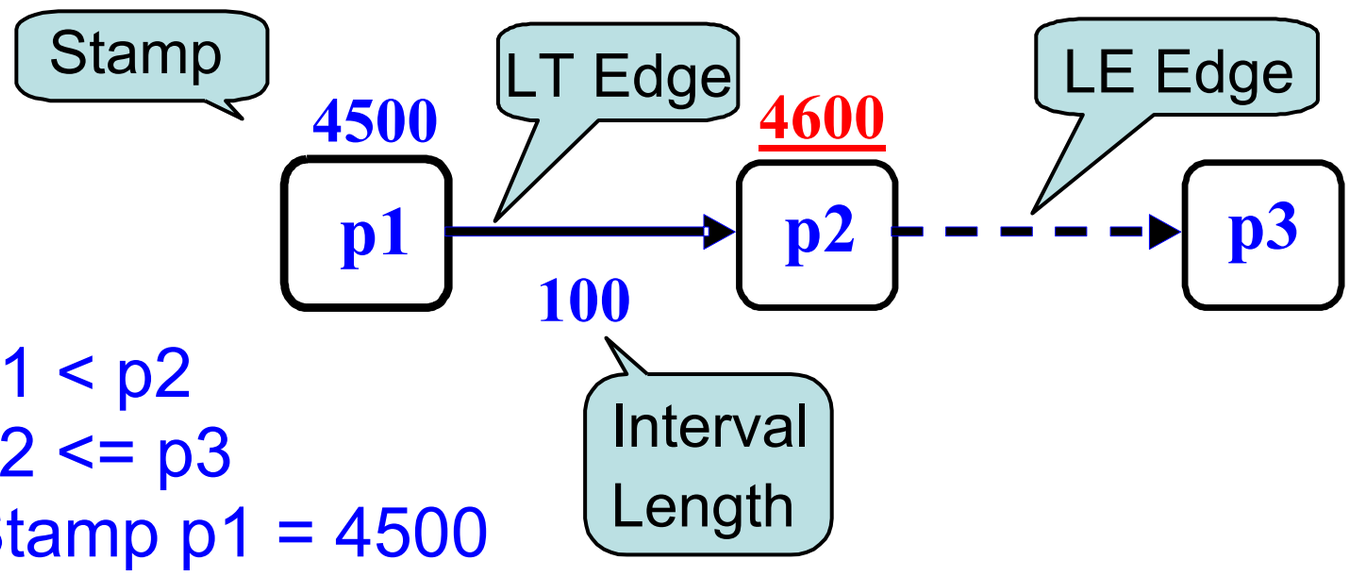
  $d1 \leq length[X,Y] \leq d2$

  $t1 \leq stamp[X] \leq t2$

  where d1, d2, t1 and t2 are rational numbers, and X, Y are points

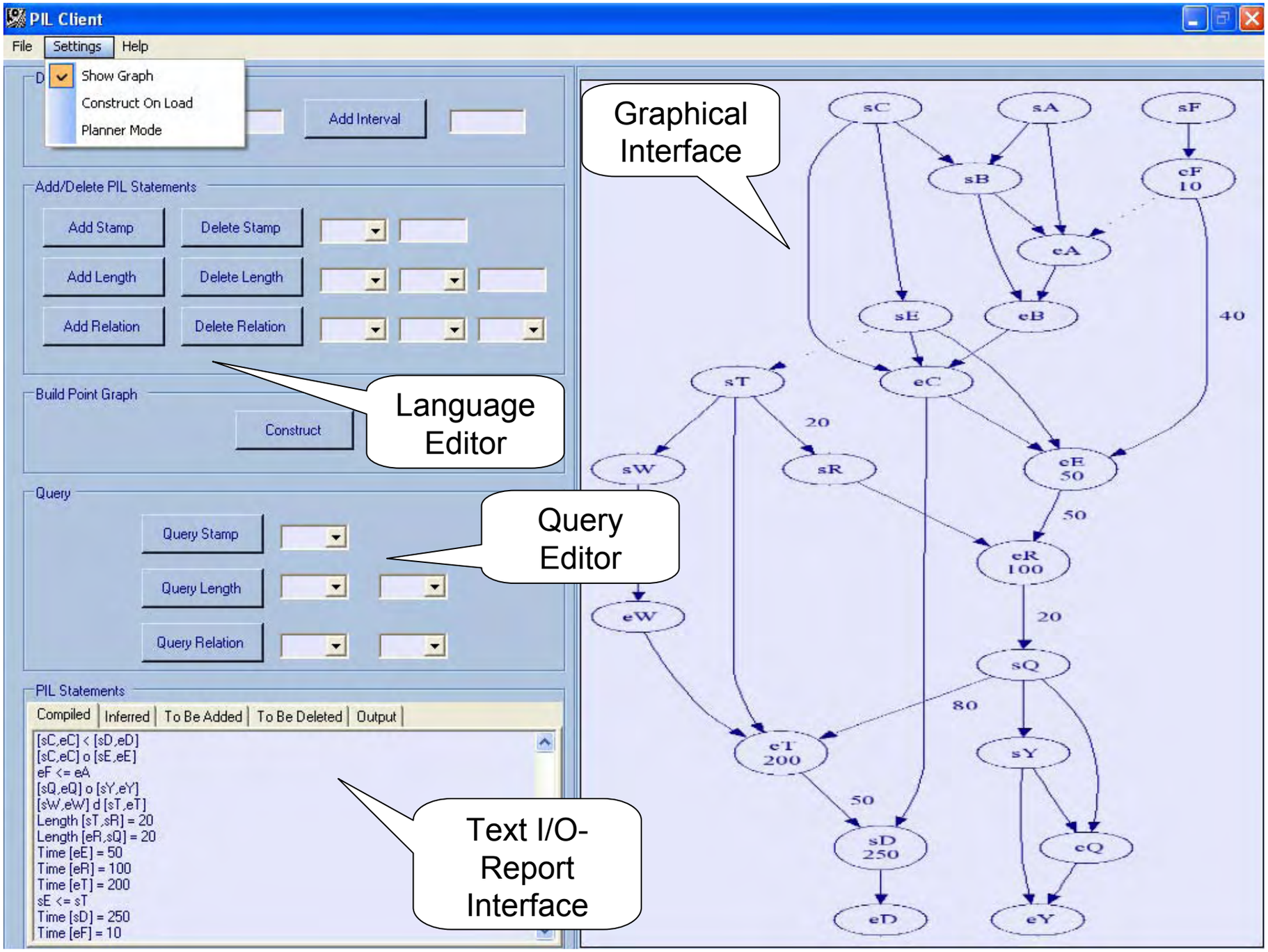
- This allows for "at least" and "at most" temporal relationships between time events

PIL Point-Graph Representation



p1 < p2
p2 <= p3
Stamp p1 = 4500
Length[p1, p2] = 100

PIL Statements and the corresponding Point Graph



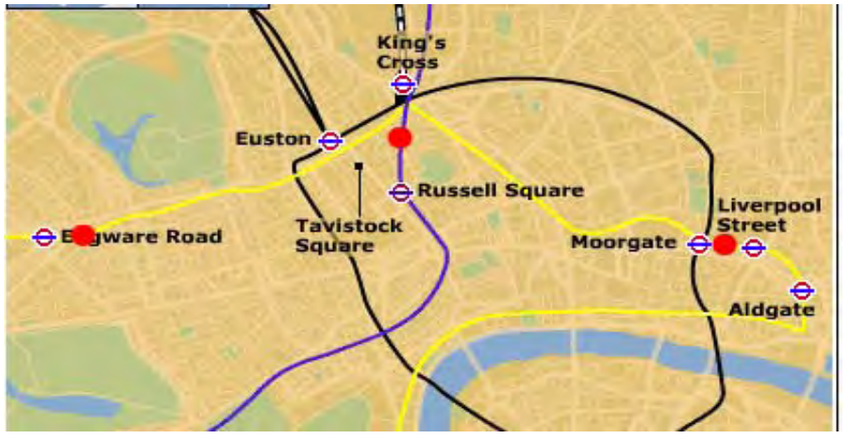
http://viking.gmu.edu/software.php

System Architectures Laboratory
George Mason University

Temper
viking.gmu.edu

Example: London Bombing

- There were four explosions in London.
- The sites of these explosions were: Travistock Square, Edgware Road, Aldgate and Russell Square.
- Three of these explosions (Edgware, Aldgate and Russell Square) were in trains.
- These trains left from King's Cross station. The journey of these trains ended in explosions.
- The time it takes a train from King's Cross to reach Edgware is at least 5 time units.
- The time it takes a train from King's Cross to reach Aldgate is at least 4 time units.
- The time it takes a train from King's Cross to reach Russell Square is at least 5 time units.



Interval **Train_King_Cross_to_Edgware,**
Train_King_Cross_to_Aldgate,
Train_King_Cross_to_Russell_Sq

Point **Explosion_at_Travistock_Square,**
Explosion_near_Edgware,
Explosion_near_Aldgate,
Explosion_near_Russell_Sq

Explosion_near_Edgware *finishes*
Train_King_Cross_to_Edgware

Explosion_near_Aldgate *finishes*
Train_King_Cross_to_Aldgate

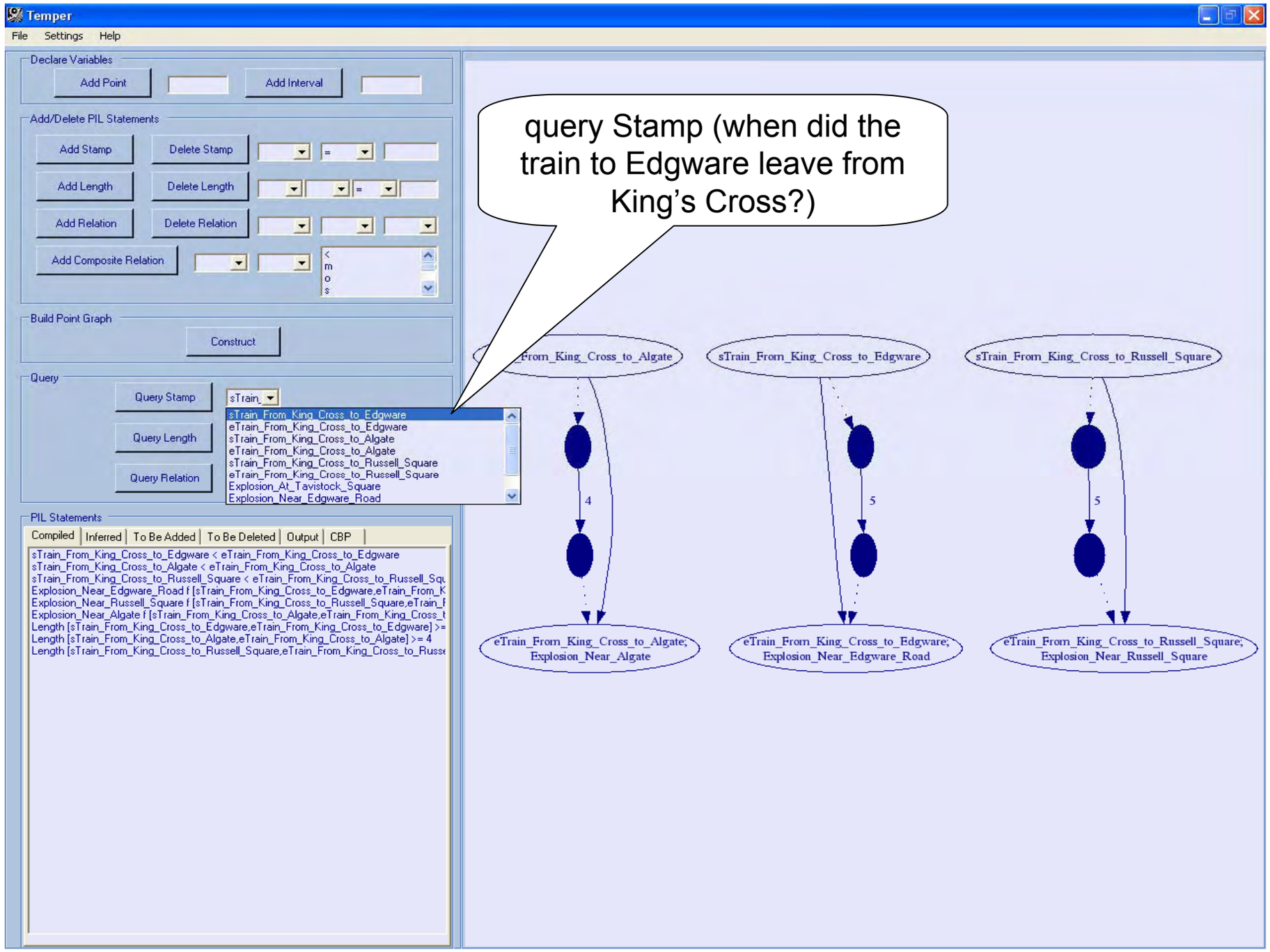
Explosion_near_Russell_Sq *finishes*
Train_King_Cross_to_Russell_Sq

Length **[Train_King_Cross_to_Edgware] >= 5**

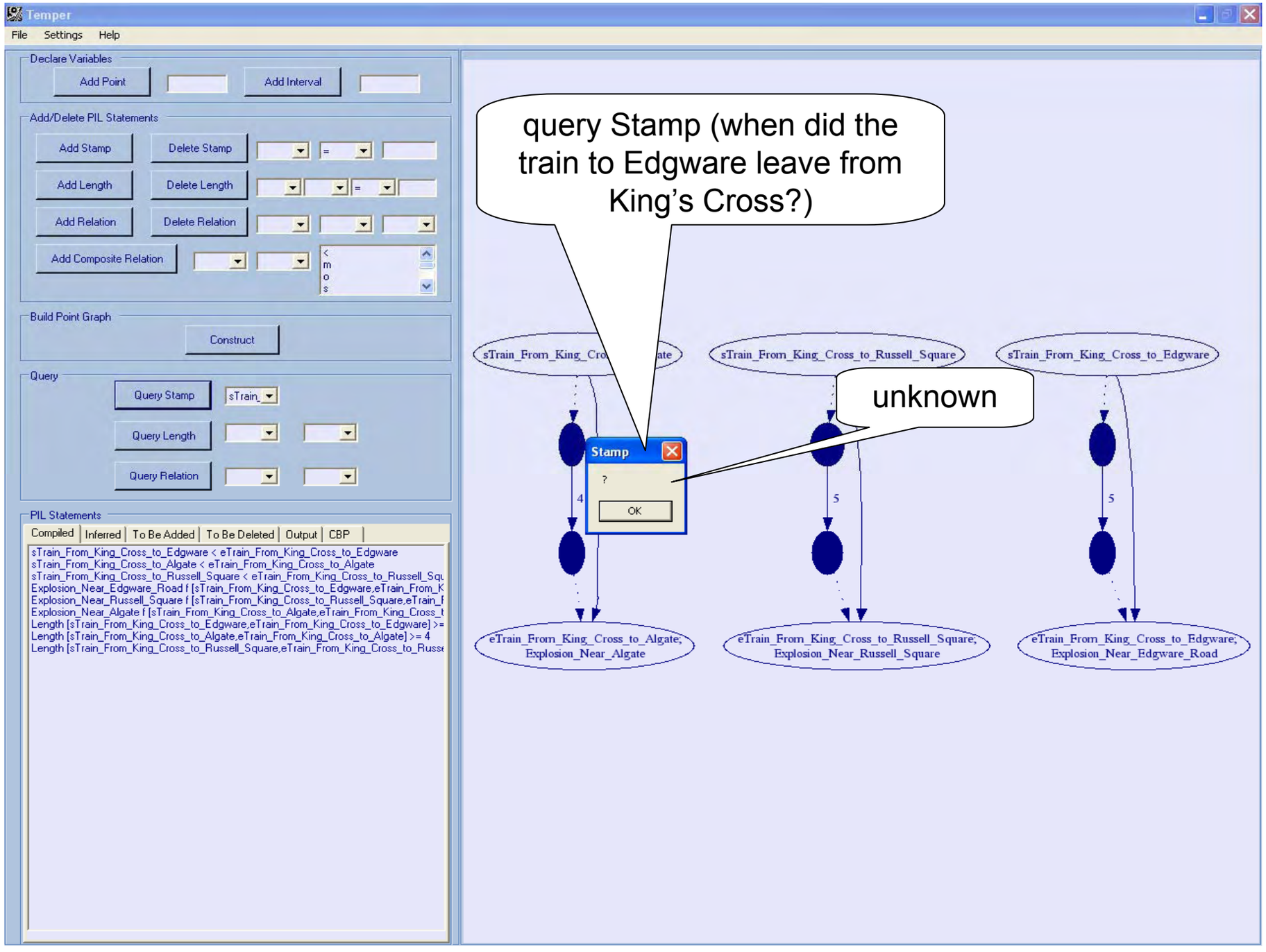
Length **[Train_King_Cross_to_Aldgate] >= 4**

Length **[Train_King_Cross_to_Russell_Sq] >= 5**

http://viking.gmu.edu/software.php



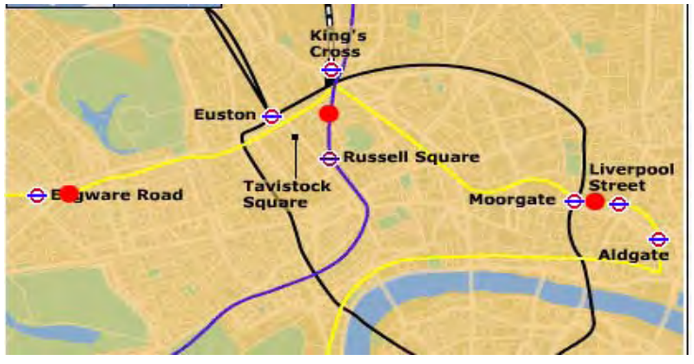
http://viking.gmu.edu/software.php



http://viking.gmu.edu/software.php

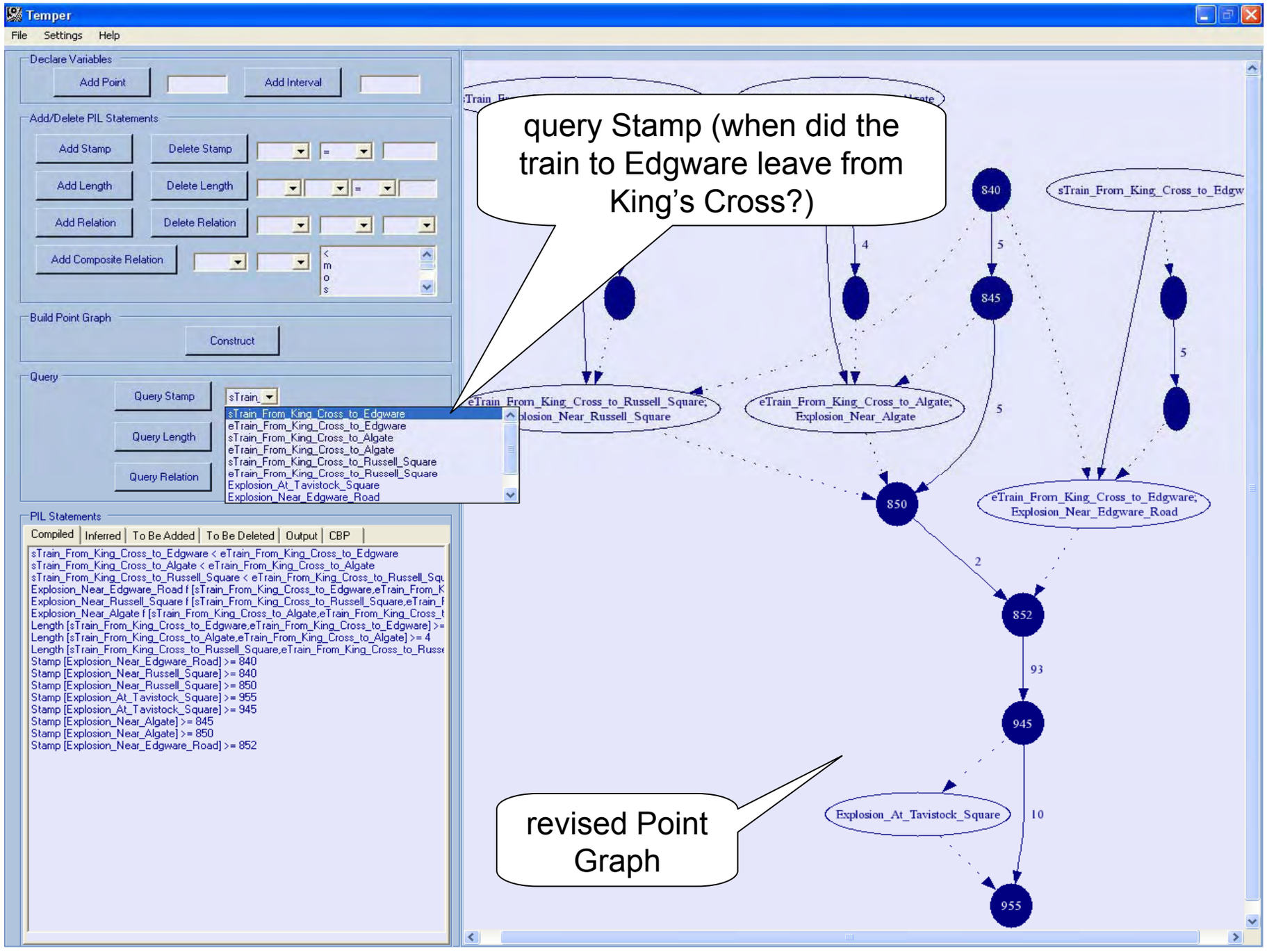
Example: London Bombing (cont'd)

- The explosion near Edgware Road took place between time units 840 and 852.
- The explosion near Aldgate took place between time units 845 and 850.
- The explosion near Russell Square took place between time units 840 and 850.
- The explosion at Travistock Square took place between time units 945 and 955.

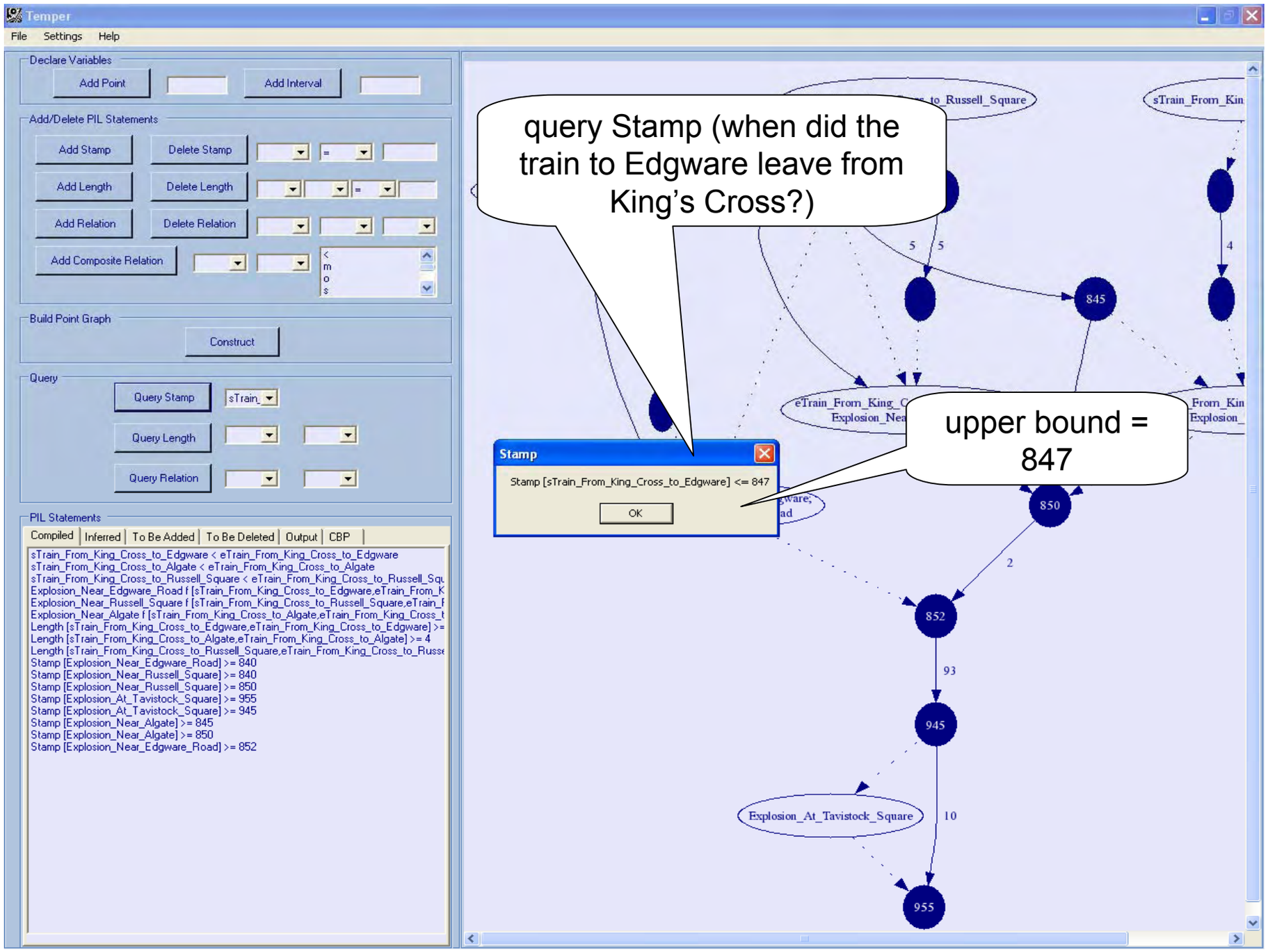


840 <= *Stamp* [Explosion_near_Edgware] <= 852
845 <= *Stamp* [Explosion_near_Aldgate] <= 850
840 <= *Stamp* [Explosion_near_Russell_Sq] <= 850
945 <= *Stamp* [Explosion_at_Travistock_Square] <= 955

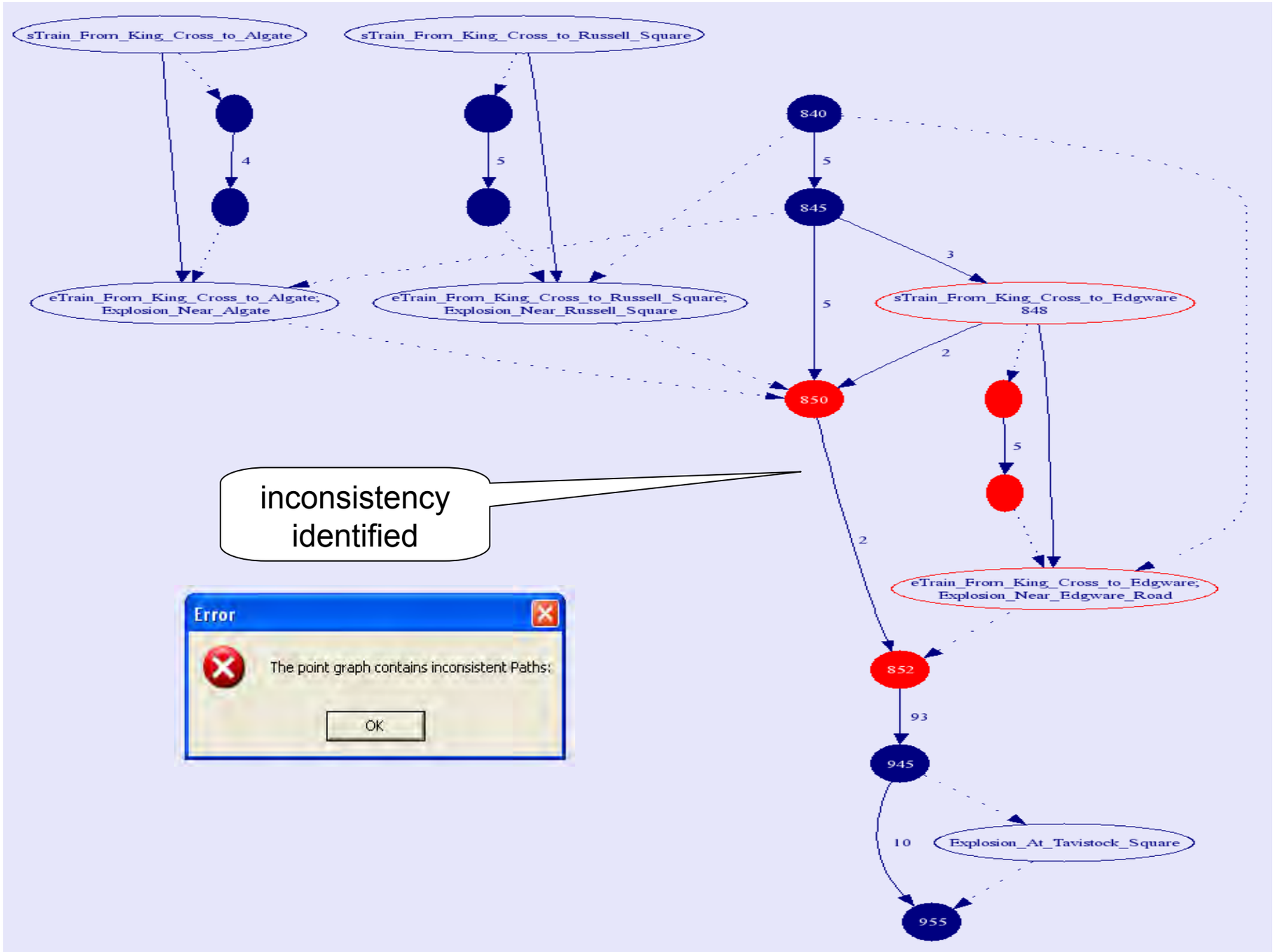
http://viking.gmu.edu/software.php



http://viking.gmu.edu/software.php



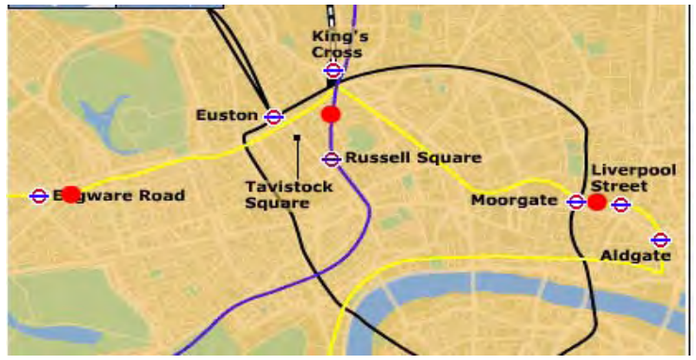
http://viking.gmu.edu/software.php



http://viking.gmu.edu/software.php

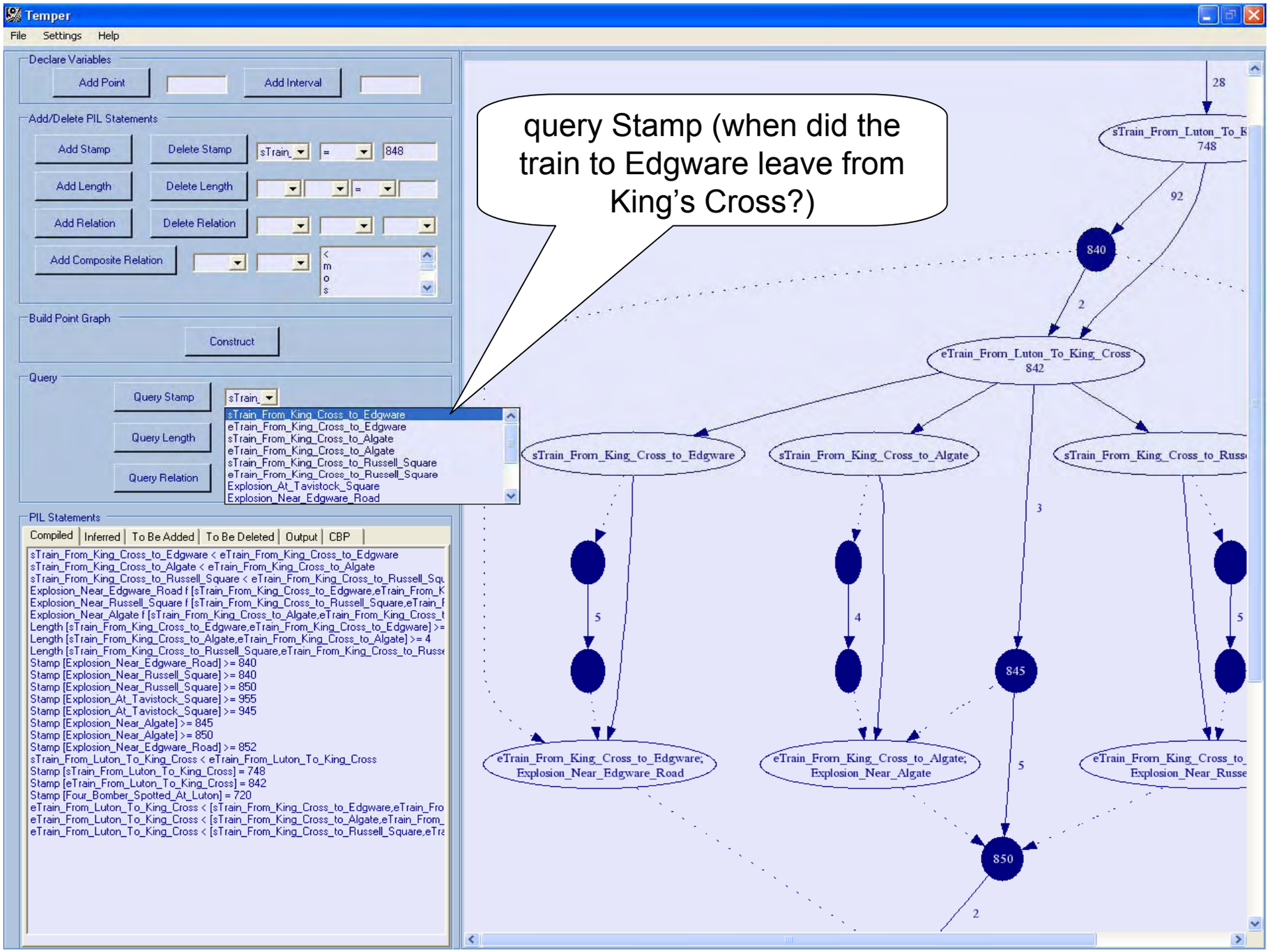
Example:
London Bombing (cont'd)

- **The alleged four bombers spotted entering the Luton station at time unit 720.**
- **The next train from Luton to King's Cross left at 748 reaching King's Cross at 842.**
- **The three trains from King's Cross station in which the explosions took place, must have left King's Cross after the train from Luton reached King's Cross.**

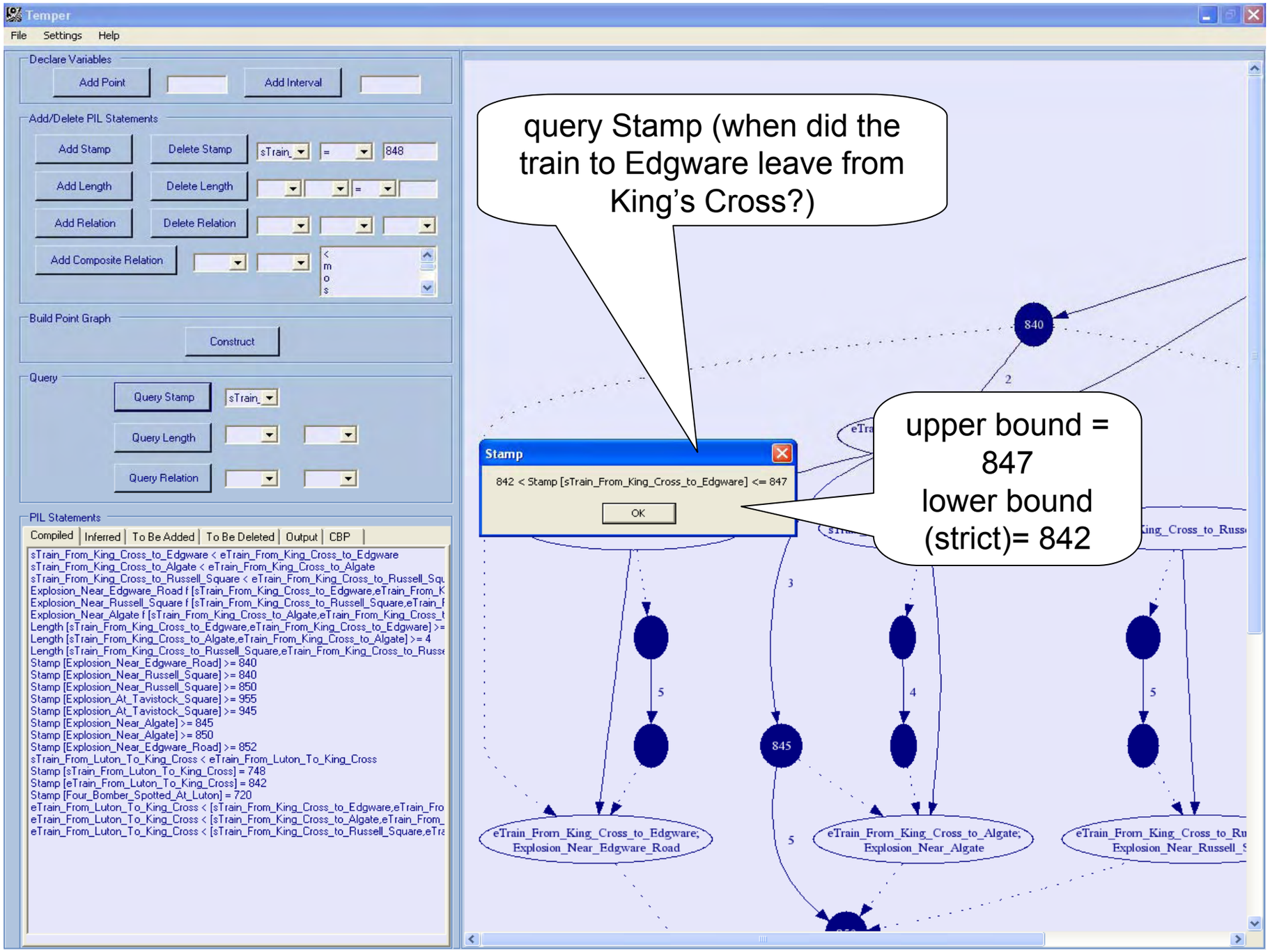


Interval Train_Luton_to_King_Cross
Point Bombers_spotted_at_Luton
Stamp [Bombers_spotted_at_Luton] = 720
Stamp [sTrain_Luton_to_King_Cross] = 748
Stamp [eTrain_Luton_to_King_Cross] = 842
eTrain_Luton_to_King_Cross *before* Train_King_Cross_to_Edgware
eTrain_Luton_to_King_Cross *before* Train_King_Cross_to_Aldgate
eTrain_Luton_to_King_Cross *before* Train_King_Cross_to_Russell_Sq

http://viking.gmu.edu/software.php



http://viking.gmu.edu/software.php



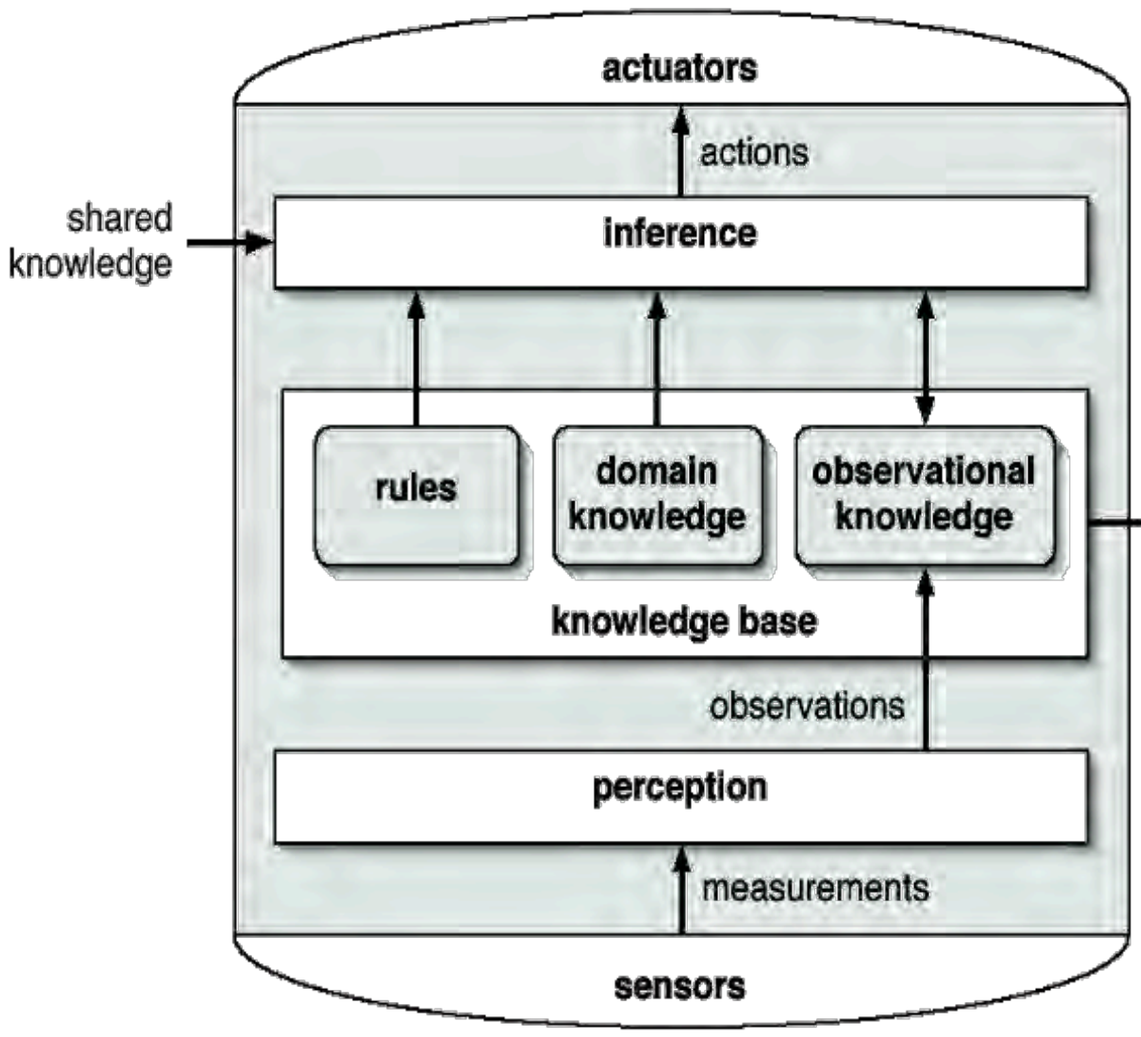
http://viking.gmu.edu/software.php

Conclusion: Temporal Reasoning

- Persistence problem
  - If P is true over some time interval T1, what can be said about P being true after T1.
  - Techniques designed to tackle this problem are crude and depend on assumptions.
- Current techniques on time representation work well in select domains.
- Not effective in domains where agent has limited knowledge of the world.

Spatial Reasoning for Smart Objects

Embedded Reasoning Architecture



Domain Knowledge

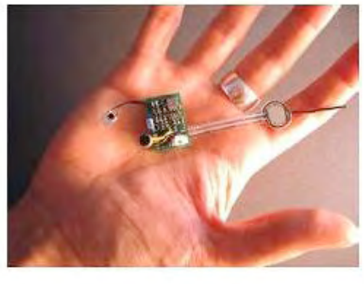
reactive(<chemical>,<chemical>)
critical_mass(<chemical>,<number>)
content(me,<chemical>)
mass(me,<number>

Safety Rules

```
hazard:-
        content(me, CH1),
        content(C, CH2),
        reactive(CH1, CH2),
        min_dist(CH1, CH2, D1),
        distance(me, C, D2),
        D2 > D1.
```

Observational Knowledge

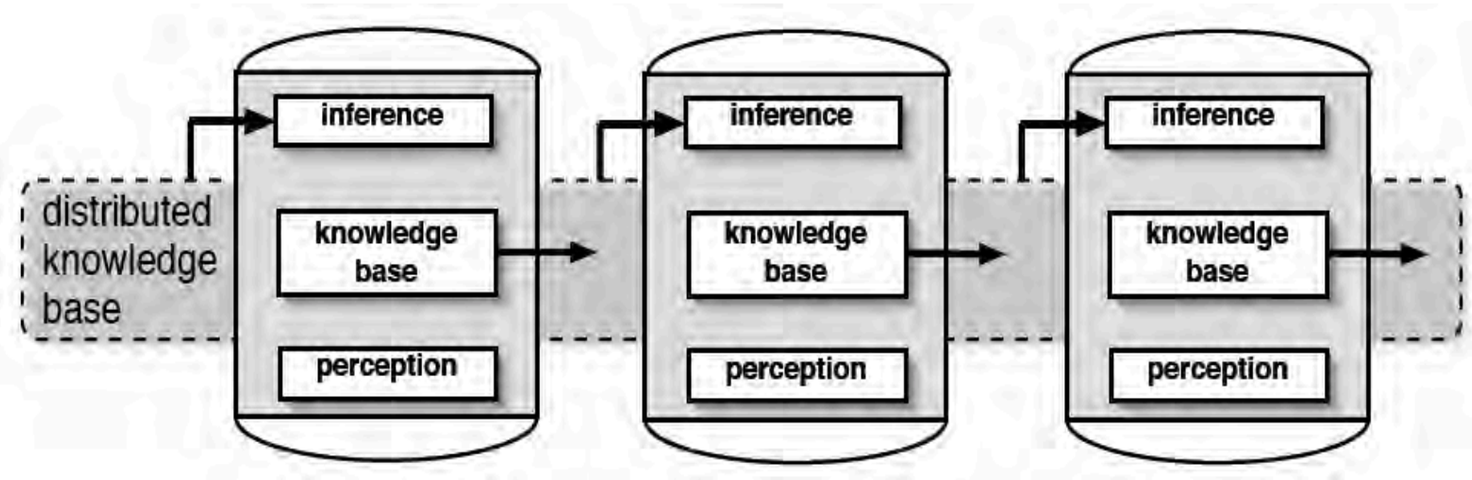
distance(<container>,<container>,<dist>)



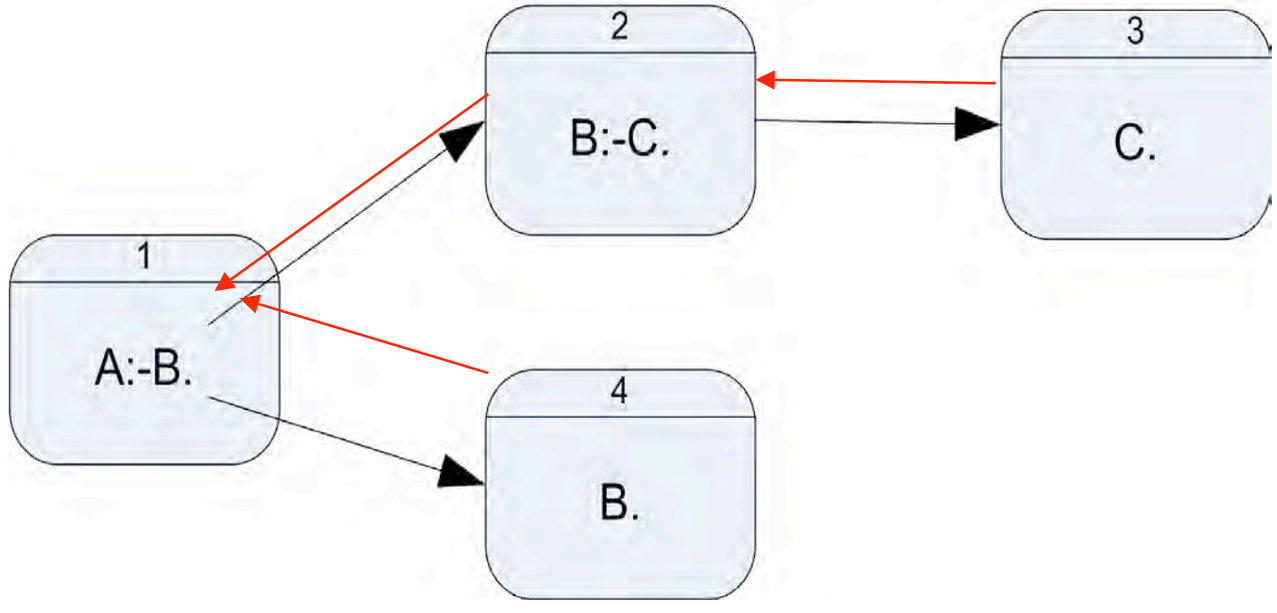
Particle

Cooperative Reasoning

View containers as distributed knowledge base



Peer-to-peer embedded reasoning



Autonomous observations + collective assessment

Cooperative Distance Measurements

Ultrasound-based relative positioning

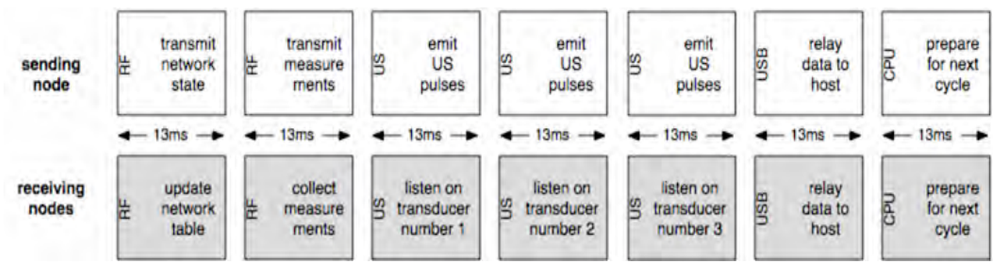
Each container has 4 ultrasound transducers that emit and detect ultrasonic pulses

Measurements

Time-of-flight → distance
max. distance < 3m

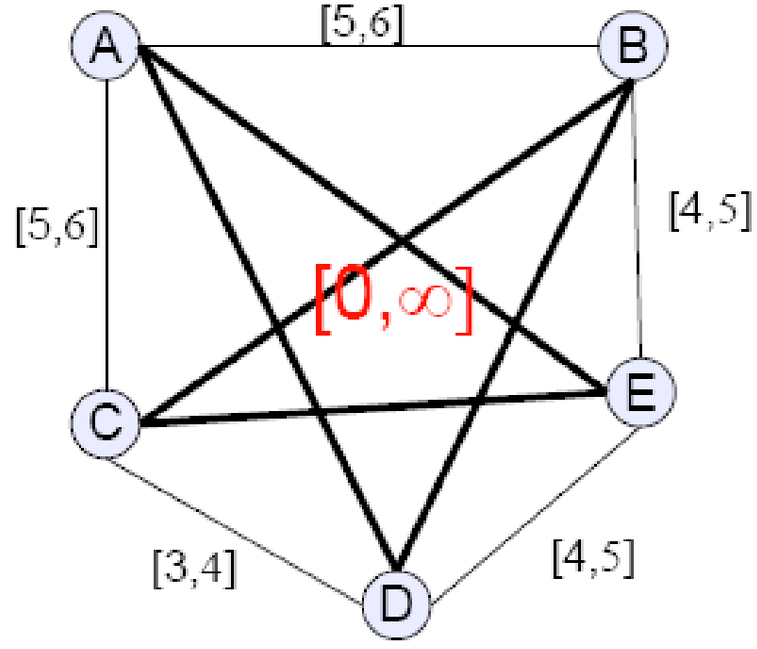
Time-slotted measurement protocol

For measuring and exchanging measurement data



Hazas et al. A Relative Positioning System for Co-Located Devices. MobiSys 2005

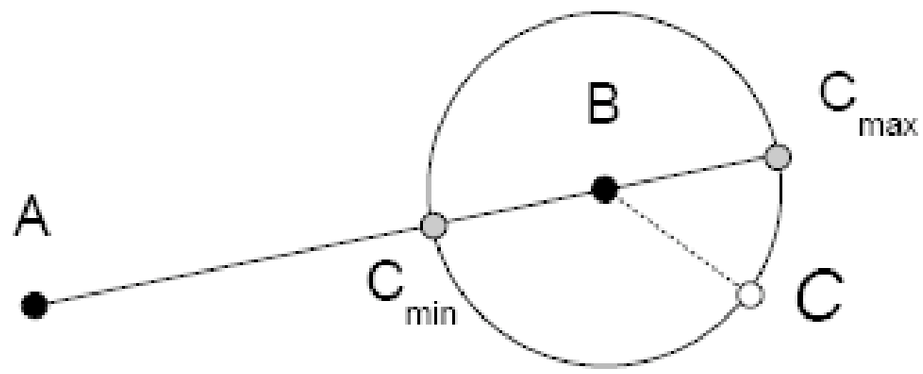
Spatial Reasoning Algorithm



- Query types
  - Distance queries: d(A, B) = ?
  - Range queries: d(A,?) ≤ c
    - Need not support absolute positions
- Technique:
  - Constraint Propagation
  - Transitive inference steps

    d(A,B) & d(B,C) -> d(A,C)

Constraint network with distance intervals



Bischoff et al. Constraint-based Distance Estimation in Ad-hoc Wireless Sensor Networks. EWSN 2005

Using Spatial Knowledge in Rules

R3: A hazard occurs if incompatible materials are stored in proximity.

INFER $hazard(incompatible)$

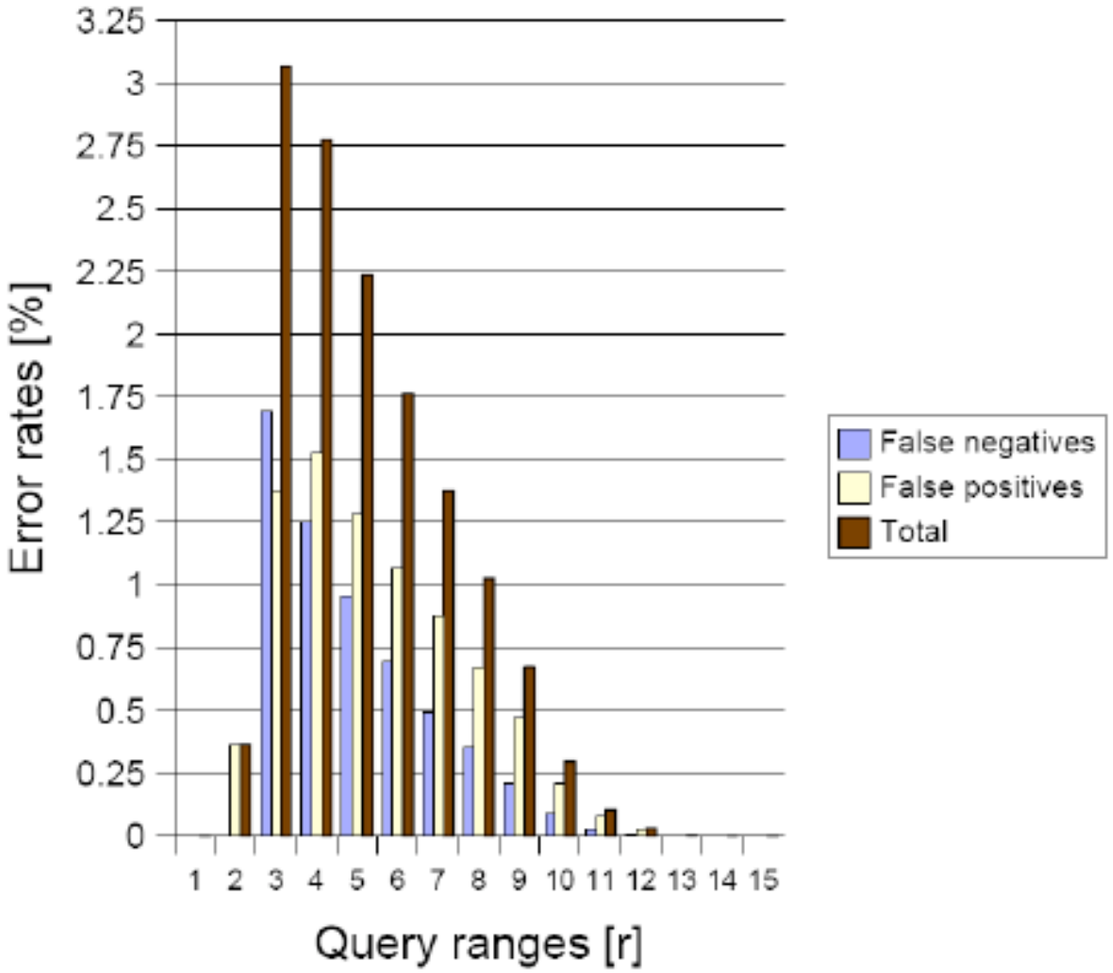
IF $content(me, Ch1)$ AND

IF $incompatibleRange(Ch1, Ch2, R)$ AND

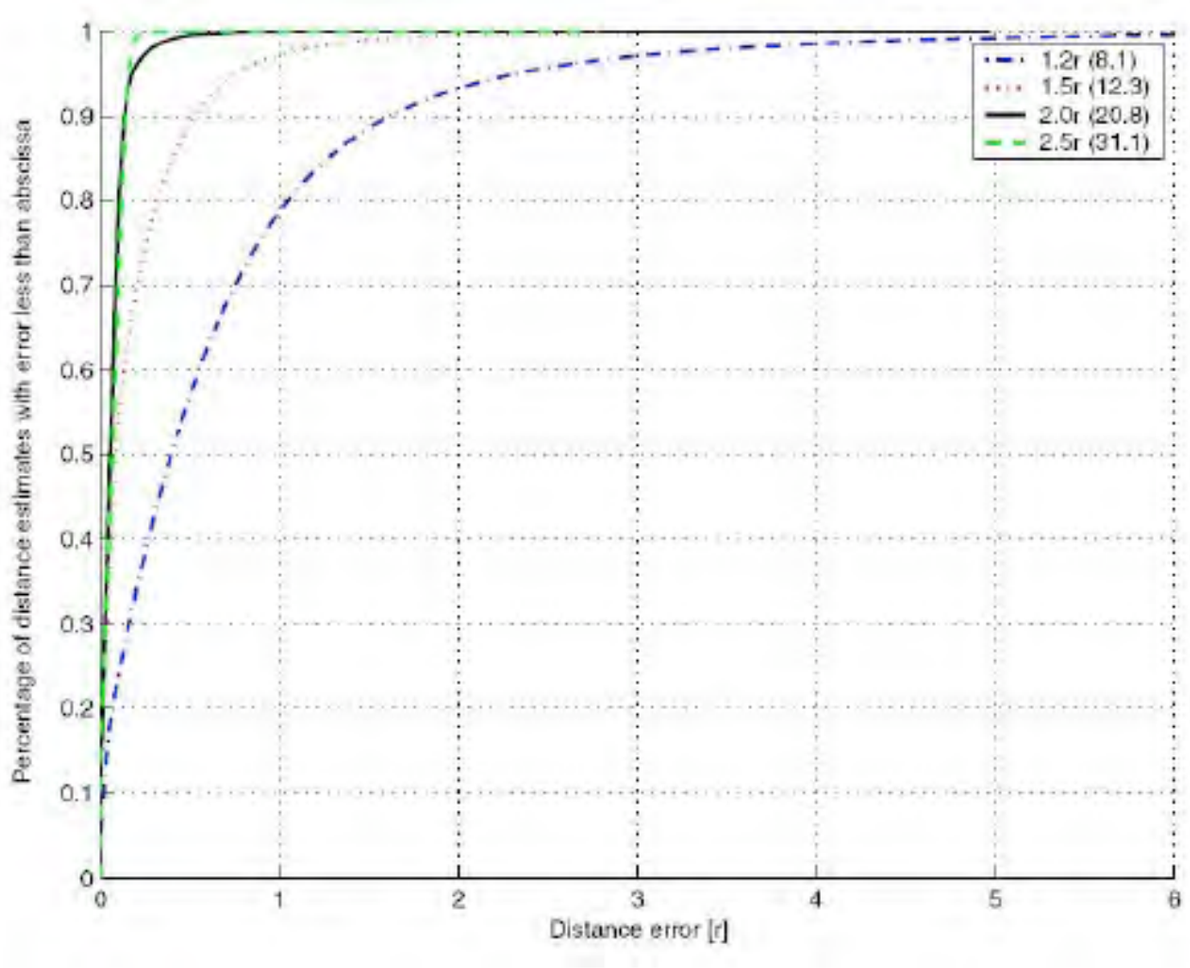
IF $inRange(me, C, R)$ AND

IF $content(C, Ch2)$

Evaluation - Accuracy



Classification error



Absolute distance error

Bischoff et al. Constraint-based Distance Estimation in Ad-hoc Wireless Sensor Networks. EWSN 2005

Conclusion

- Reasoning is not just about logic
- It's about building intelligent systems that are part of the physical world
- Reasoning about time and space is essential for intelligent systems
- Constraint-based approaches are explicit, effective, and strike a balance between expressiveness and efficiency

END