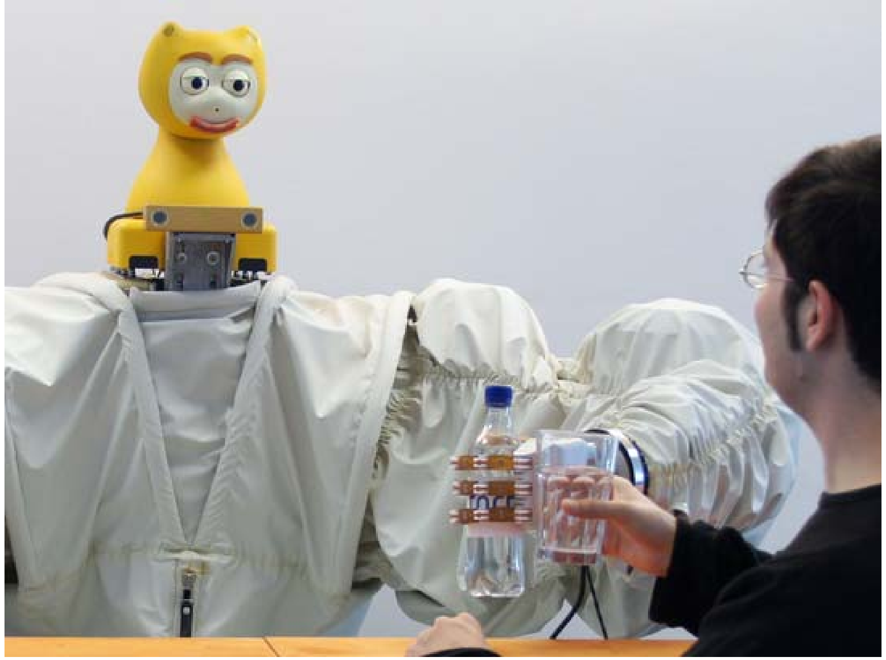
Vorlesung

Grundlagen der Künstlichen Intelligenz

Reinhard Lafrenz / Prof. A. Knoll

Robotics and Embedded Systems
Department of Informatics – I6
Technische Universität München

www6.in.tum.de
lafrenz@in.tum.de
089-289-18136
Room 03.07.055



Wintersemester 2012/13 7.1.2013

Chapter 12

Knowledge representation

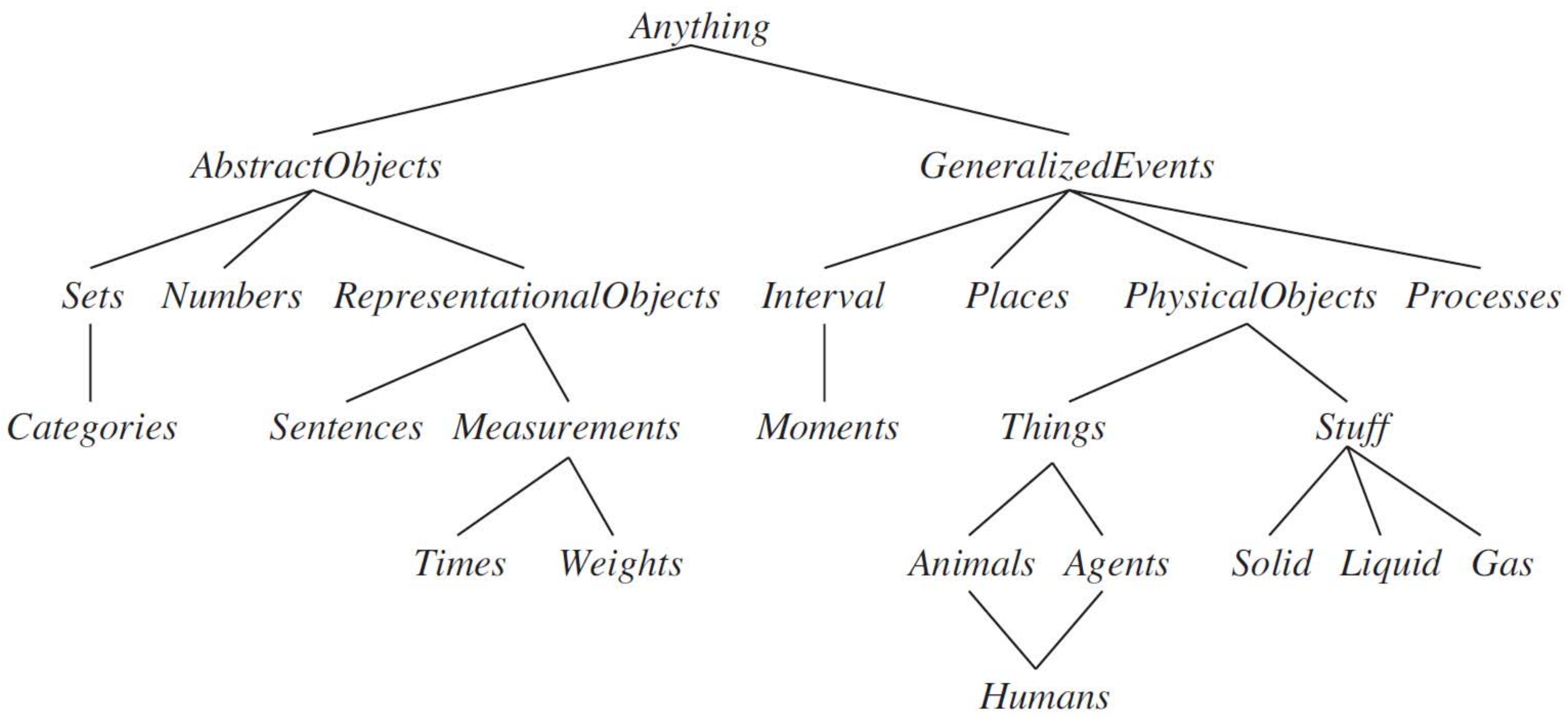
Outline

- Ontological engineering
- Categories and objects
- Actions, situations and events
- Mental events and mental objects
- The internet shopping world
- Reasoning systems for categories
- Reasoning with default information
- Truth maintenance systems

Ontological engineering

- How to create more general and flexible representations.
  - Concepts like actions, time, physical object and beliefs
  - Operates on a bigger scale than K.E.
- Define general framework of concepts
  - Upper ontology
- Limitations of logic representation
  - Red, green and yellow tomatoes: exceptions and uncertainty

The upper ontology of the world



Difference with special-purpose ontologies

- A general-purpose ontology should be applicable in more or less any special-purpose domain.
  - Add domain-specific axioms
- In any sufficiently demanding domain different areas of knowledge need to be unified.
  - Reasoning and problem solving could involve several areas simultaneously
- What do we need to express?

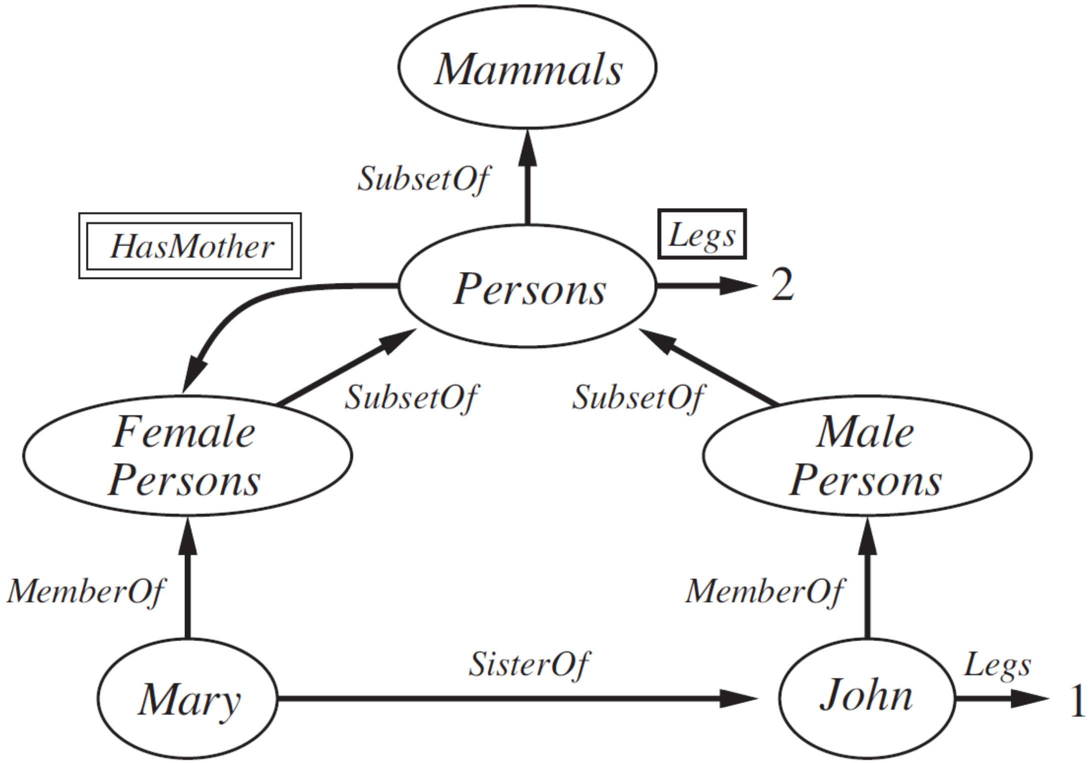
  Categories, Measures, Composite objects, Time, Space, Change, Events, Processes, Physical Objects, Substances, Mental Objects, Beliefs

Categories and objects

- KR requires the organisation of objects into categories
  - Interaction at the level of the object
  - Reasoning at the level of categories
- Categories play a role in predictions about objects
  - Based on perceived properties
- Categories can be represented in two ways by FOL
  - Predicates: apple(x)
  - *Reification* of categories into objects: apples
- Category = set of its members

Category organization

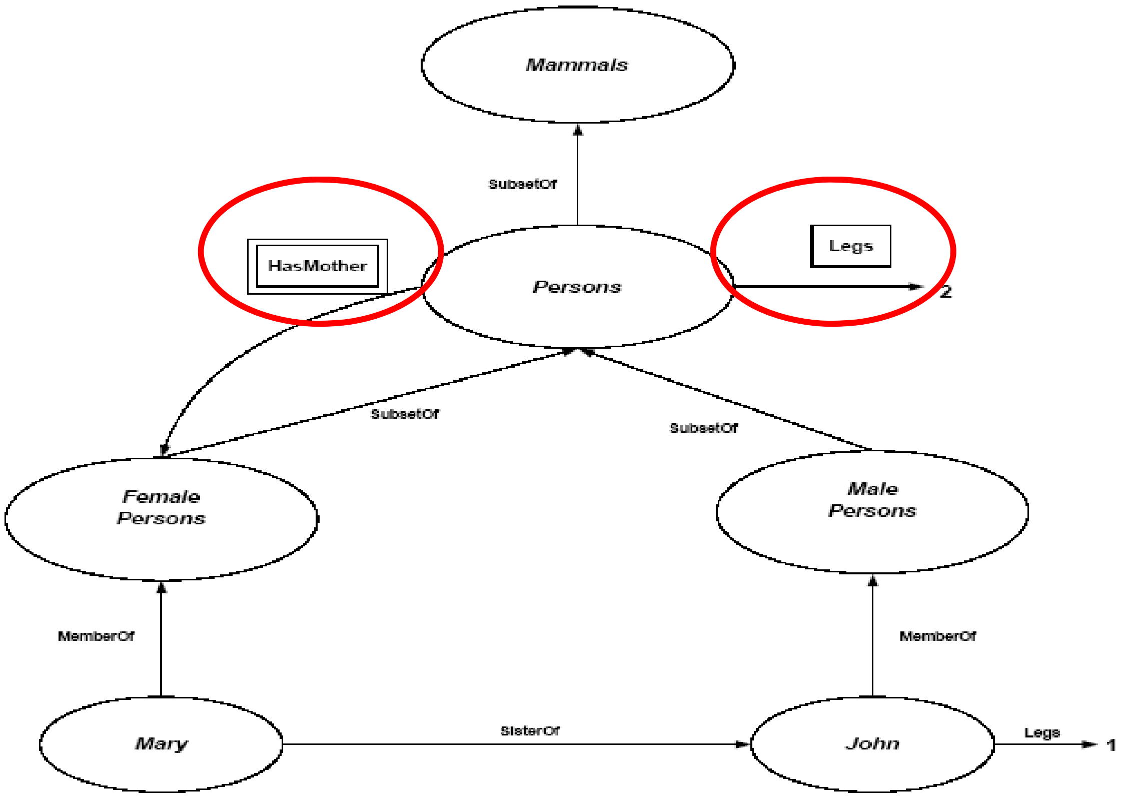
- Relation = *inheritance*:
  - All instance of food are edible, fruit is a subclass of food and apples is a subclass of fruit then an apple is edible.
- Defines a taxonomy



Semantic Networks

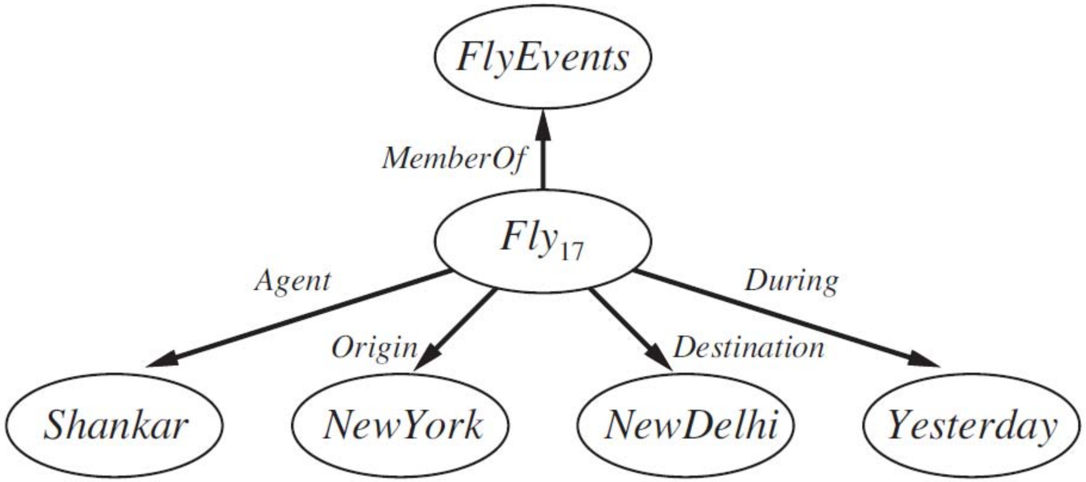
- Logic vs. semantic networks
- Many variations
  - All represent individual objects, categories of objects and relationships among objects.
- Allows for inheritance reasoning
  - Female persons inherit all properties from person.
  - Cfr. OO programming.
- Inference of inverse links
  - SisterOf vs. HasSister

Semantic network example



Semantic networks

- Drawbacks
  - Links can only assert binary relations
  - Can be resolved by reification of the proposition as an event



- Representation of default values
  - Enforced by the inheritance mechanism.

FOL and categories

- An object is a member of a category
  - MemberOf(BB_{12},Basketballs)
- A category is a subclass of another category
  - SubsetOf(Basketballs,Balls)
- All members of a category have some properties
  - $\forall$ x (MemberOf(x,Basketballs) $\Rightarrow$ Round(x))
- All members of a category can be recognized by some properties
  - $\forall$ x (Orange(x) $\wedge$ Round(x) $\wedge$ Diameter(x)=9.5in $\wedge$ MemberOf(x,Balls) $\Rightarrow$ MemberOf(x,BasketBalls))
- A category as a whole has some properties
  - MemberOf(Dogs,DomesticatedSpecies)

Relations between categories

- Two or more categories are ***disjoint*** if they have no members in common:
  - Disjoint(s)$\Leftrightarrow$($\forall$ c_1,c_2 $c_1 \in s \wedge c_2 \in s \wedge c_1 \neq c_2 \Rightarrow$ Intersection(c_1,c_2) ={})
  - Example; Disjoint({animals, vegetables})
- A set of categories *s* constitutes an ***exhaustive decomposition*** of a category *c* if all members of the set *c* are covered by categories in *s*:
  - E.D.(s,c) $\Leftrightarrow$ ($\forall$ i: i $\in$ c $\Rightarrow \exists$ c_2: $c_2 \in s \wedge i \in c_2$)
  - Example: ExhaustiveDecomposition({Americans, Canadians, Mexicans},NorthAmericans).

Relations between categories

- A ***partition*** is a disjoint exhaustive decomposition:
  - Partition(s,c) $\Leftrightarrow$ Disjoint(s) $\wedge$ E.D.(s,c)
  - Example: Partition({Males,Females},Persons).
- Is ({Americans,Canadians, Mexicans},NorthAmericans) a partition?
- Categories can be defined by providing necessary and sufficient conditions for membership
  - $\forall$ x Bachelor(x) $\Leftrightarrow$ Male(x) $\wedge$ Adult(x) $\wedge$ Unmarried(x)

Natural kinds

- Many categories have no clear-cut definitions (chair, bush, book).
- Tomatoes: sometimes green, red, yellow, black. Mostly round.
- One solution: category *Typical(Tomatoes).*
  - $\forall$ x: x $\in$ Typical(Tomatoes) $\Rightarrow$ Red(x) $\wedge$ Spherical(x).
  - We can write down useful facts about categories without providing exact definitions.
- What about "bachelor"? Quine challenged the utility of the notion of *strict definition*. We might question a statement such as "the Pope is a bachelor".

Physical composition

- One object may be part of another:
  - PartOf(Bucharest,Romania)
  - PartOf(Romania,EasternEurope)
  - PartOf(EasternEurope,Europe)
- The PartOf predicate is transitive (and irreflexive), so we can infer that PartOf(Bucharest,Europe)
- More generally:
  - $\forall$ x PartOf(x,x)
  - $\forall$ x,y,z PartOf(x,y) $\wedge$ PartOf(y,z) $\Rightarrow$ PartOf(x,z)
- Often characterized by structural relations among parts.
  - E.g. Biped(a) $\Rightarrow$

$$(\exists l_1, l_2, b)(Leg(l_1) \wedge Leg(l_2) \wedge Body(b) \wedge$$

$$PartOf(l_1, a) \wedge PartOf(l_2, a) \wedge PartOf(b, a) \wedge$$

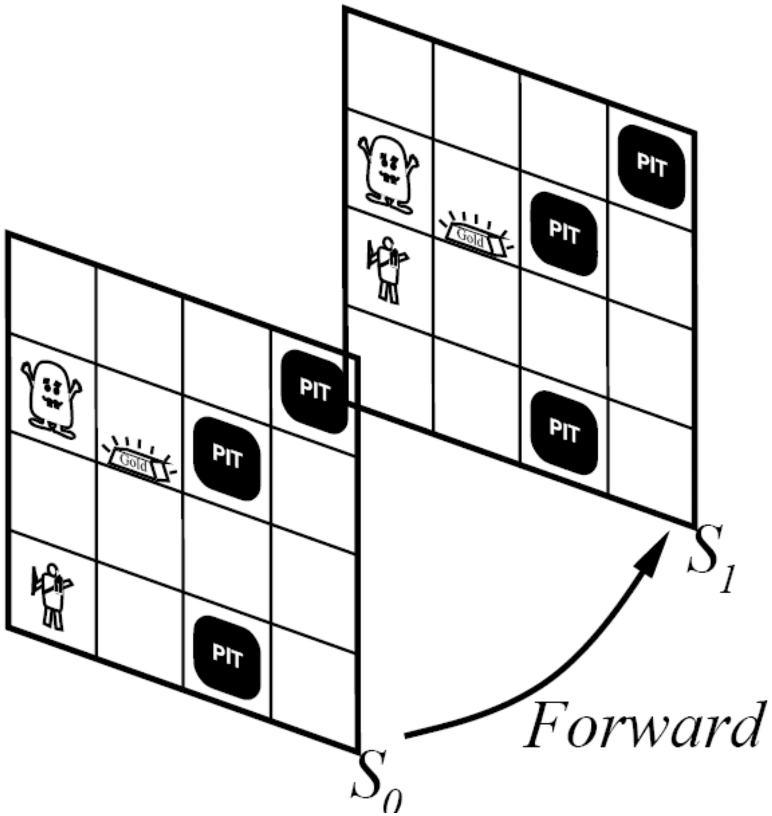
$$Attached(l_1, b) \wedge Attached(l_2, b) \wedge$$

$$l_1 \neq l_2 \wedge (\forall l_3)(Leg(l_3) \Rightarrow (l_3 = l_1 \vee l_3 = l_2)))$$

Measurements

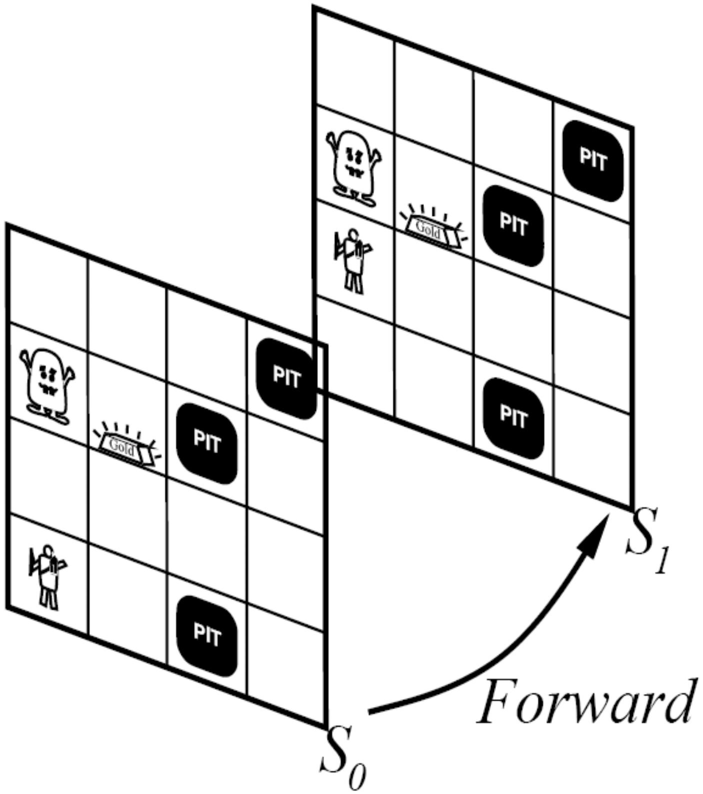
- Objects have height, mass, cost, ...
  Values that we assign to these are **measures**
- Combine Unit functions with a number: Length(L_1) = Inches(1.5) = Centimeters(3.81).
- Conversion between units:
  $\forall$ i Centimeters(2.54 x i)=Inches(i).
- Some measures have no scale: Beauty, Difficulty, etc.
  - Most important aspect of measures: is that they are orderable.
  - Don't care about the actual numbers. (An apple can have deliciousness .9 or .1.)

Actions, events and situations



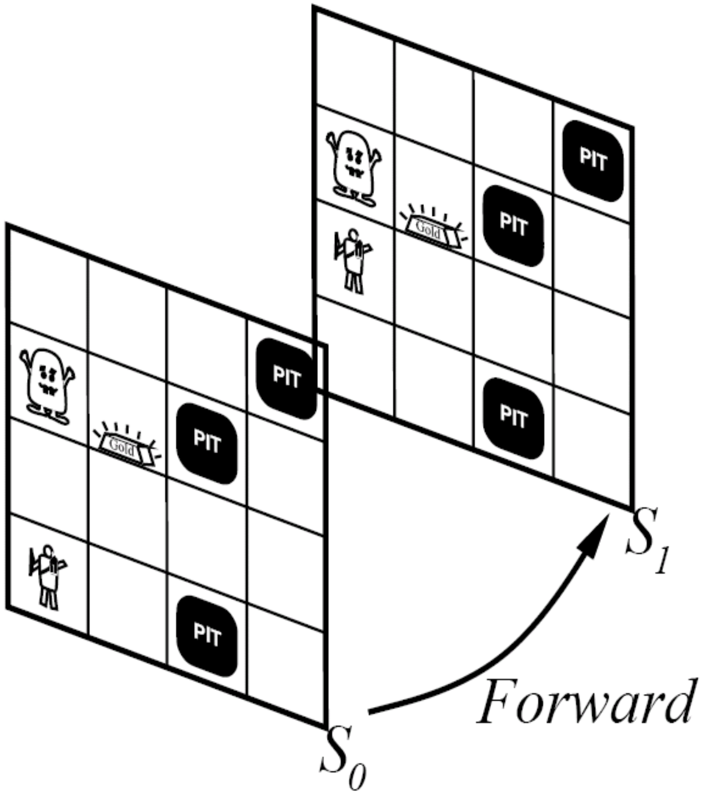
- Reasoning about outcome of actions is central to KB-agent.
- How can we keep track of location in FOL?
  - Remember the multiple copies in PL.
- Representing time by situations (states resulting from the execution of actions).
  - Situation calculus

Actions, events and situations



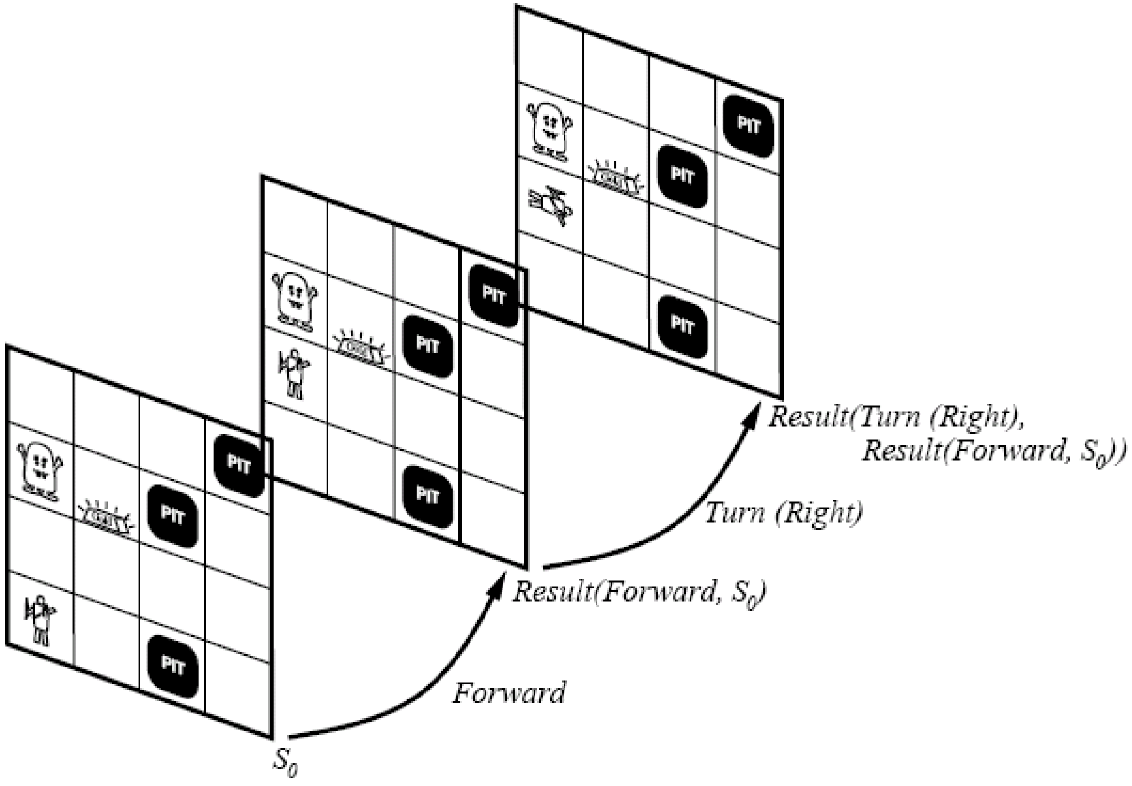
- Situation calculus:
  - Actions are logical terms
  - Situations are logical terms consiting of
    - The initial situation I
    - All situations resulting from the action on I (=*Result(a,s)*)
  - Fluent are functions and predicates that vary from one situation to the next.
    - E.g. $\neg Holding(G_1, S_0)$
  - Eternal predicates are also allowed
    - E.g. $Gold(G_1)$

Actions, events and situations



- Results of action sequences are determined by the individual actions.
- *Projection task:* an SC agent should be able to deduce the outcome of a sequence of actions.
- *Planning task:* find a sequence that achieves a desirable effect

Actions, events and situations



Describing change

- Simples Situation calculus requires two axioms to describe change:
  - Possibility axiom: when is it possible to do the action

    $At(Agent,x,s) \wedge Adjacent(x,y) \Rightarrow Poss(Go(x,y),s)$
  - Effect axiom: describe changes due to action

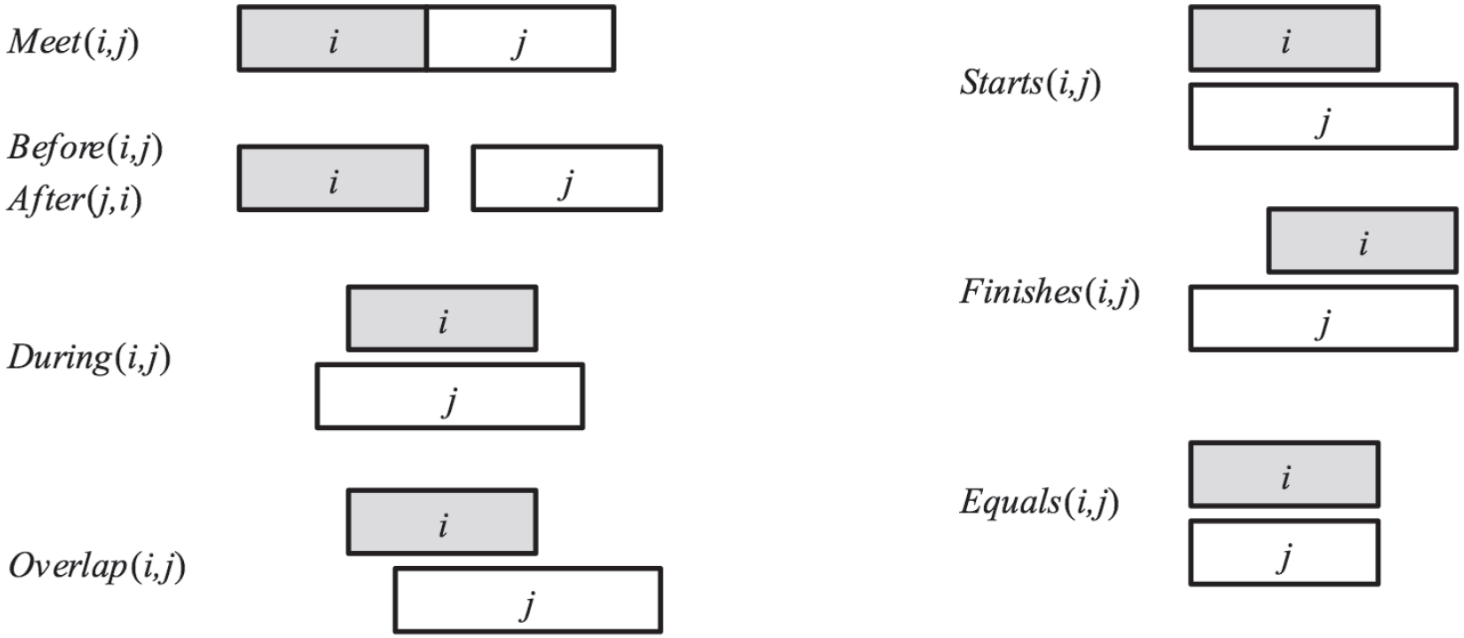
    $Poss(Go(x,y),s) \Rightarrow At(Agent,y,Result(Go(x,y),s))$
- What stays the same?
  - Frame problem: how to represent all things that stay the same?
  - Frame axiom: describe non-changes due to actions

    $At(o,x,s) \wedge (o \neq Agent) \wedge \neg Holding(o,s) \Rightarrow At(o,x,Result(Go(y,z),s))$

Time relations and time logic

- To overcome limitations in the situation calculus, event calculus can be used.
  - E.g. predicate *HoldsAt (fluent, time)* describes that a specific sentence *f* is true at time *t*.

  Time relations can be as follows:



Representational frame problem

- If there are F fluents and A actions then we need AF frame axioms to describe other objects are stationary unless they are held.
  - We write down the effect of each actions
- Solution; describe how each fluent changes over time
  - Successor-state axiom:

$$Pos(a,s) \Rightarrow (At(Agent,y,Result(a,s)) \Leftrightarrow (a = Go(x,y)) \vee$$
$$(At(Agent,y,s) \wedge a \neq Go(y,z))$$

  - Note that next state is completely specified by current state.
  - Each action effect is mentioned only once.

Other problems

- How to deal with secondary (implicit) effects?
  - If the agent is carrying the gold and the agent moves then the gold moves too.
  - Ramification problem
- How to decide EFFICIENTLY whether fluents hold in the future?
  - Inferential frame problem.
- Extensions:
  - Event calculus (when actions have a duration)
  - Process categories

Mental events and objects

- So far, KB agents can have beliefs and deduce new beliefs
- What about knowledge about beliefs? What about knowledge about the inference process?
  - Requires a model of the mental objects in someone's head and the processes that manipulate these objects.
- Relationships between agents and mental objects: believes, knows, wants, …
  - Believes(Lois,Flies(Superman)) with Flies(Superman) being a function … a candidate for a mental object (reification).
  - Agent can now reason about the beliefs of agents.

The internet shopping world

- A Knowledge Engineering example
- An agent that helps a buyer to find product offers on the internet.
  - IN = product description (precise or $\neg$precise)
  - OUT = list of webpages that offer the product for sale.
- Environment = WWW
- Percepts = web pages (character strings)
  - Extracting useful information required.

The internet shopping world

- Find relevant product offers

  RelevantOffer(page,url,query) $\Leftrightarrow$ *Relevant(page, url, query)* $\wedge$ *Offer(page)*

  – Write axioms to define Offer(x)
  – Find relevant pages: Relevant(x,y,z) ?
    - Start from an initial set of stores.
    - What is a relevant category?
    - What are relevant connected pages?
  – Require rich category vocabulary.
    - Synonymy and ambiguity
  – How to retrieve pages: *GetPage(url)*?
    - Procedural attachment
- Compare offers (information extraction).

Reasoning systems for categories

- How to organise and reason with categories?
  - Semantic networks
    - Visualize knowledge-base
    - Efficient algorithms for category membership inference
  - Description logics
    - Formal language for constructing and combining category definitions
    - Efficient algorithms to decide subset and superset relationships between categories.

Description logics

- Are designed to describe defintions and properties about categories
  - A formalization of semantic networks
- Principal inference task is
  - *Subsumption*: checking if one category is the subset of another by comparing their definitions
  - *Classification*: checking whether an object belongs to a category.
  - Consistency: whether the category membership criteria are logically satisfiable.

Reasoning with Default Information

- "The following courses are offered: CS101, CS102, CS106, EE101"
  - Four (data base semantics)
    - Assume that this information is complete (not asserted ground atomic sentences are false)

    = CLOSED WORLD ASSUMPTION
    - Assume that distinct names refer to distinct objects

    = UNIQUE NAMES ASSUMPTION
  - Between one and infinity (logic)
    - Does not make these assumptions
    - Requires completion.

Truth maintenance systems

- Many of the inferences have default status rather than being absolutely certain
  - Inferred facts can be wrong and need to be retracted = BELIEF REVISION.
  - Assume KB contains sentence P and we want to execute TELL(KB, $\neg$P)
    - To avoid contradiction: RETRACT(KB,P)
    - But what about sentences inferred from P?
- Truth maintenance systems are designed to handle these complications.

Summary

- Knowledge representation is crucial for efficient reasoning
- Ontologies are a widely used way for representing knowledge
- Upper ontology to describe main concepts and object classes of the world
- Individual ontologies for specific domains needed
- Different ways of reasoning
  - Navigation in semantic networks
  - Formal reasoning using logical representations
- Problem: Handling of default values