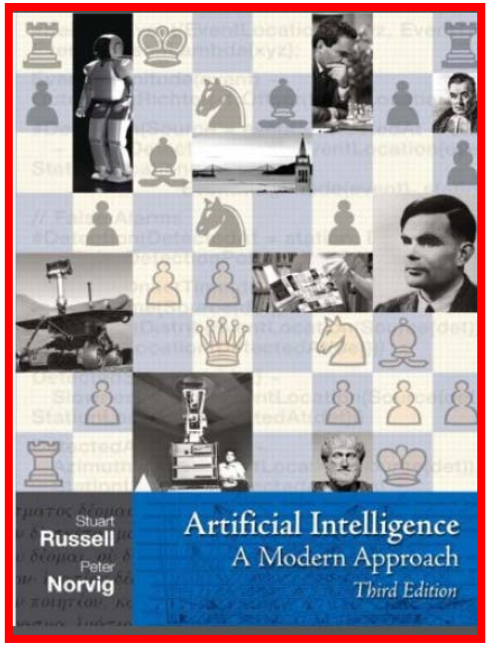


(CHAPTER-12)
KNOWLEDGE REPRESENTATION

Yanmei Zheng

Outline

- ✓ **A general ontology**
- ✓ **The basic categories**
- ✓ **Representing actions**

Ontological Engineering

Dr.Yanmei Zheng

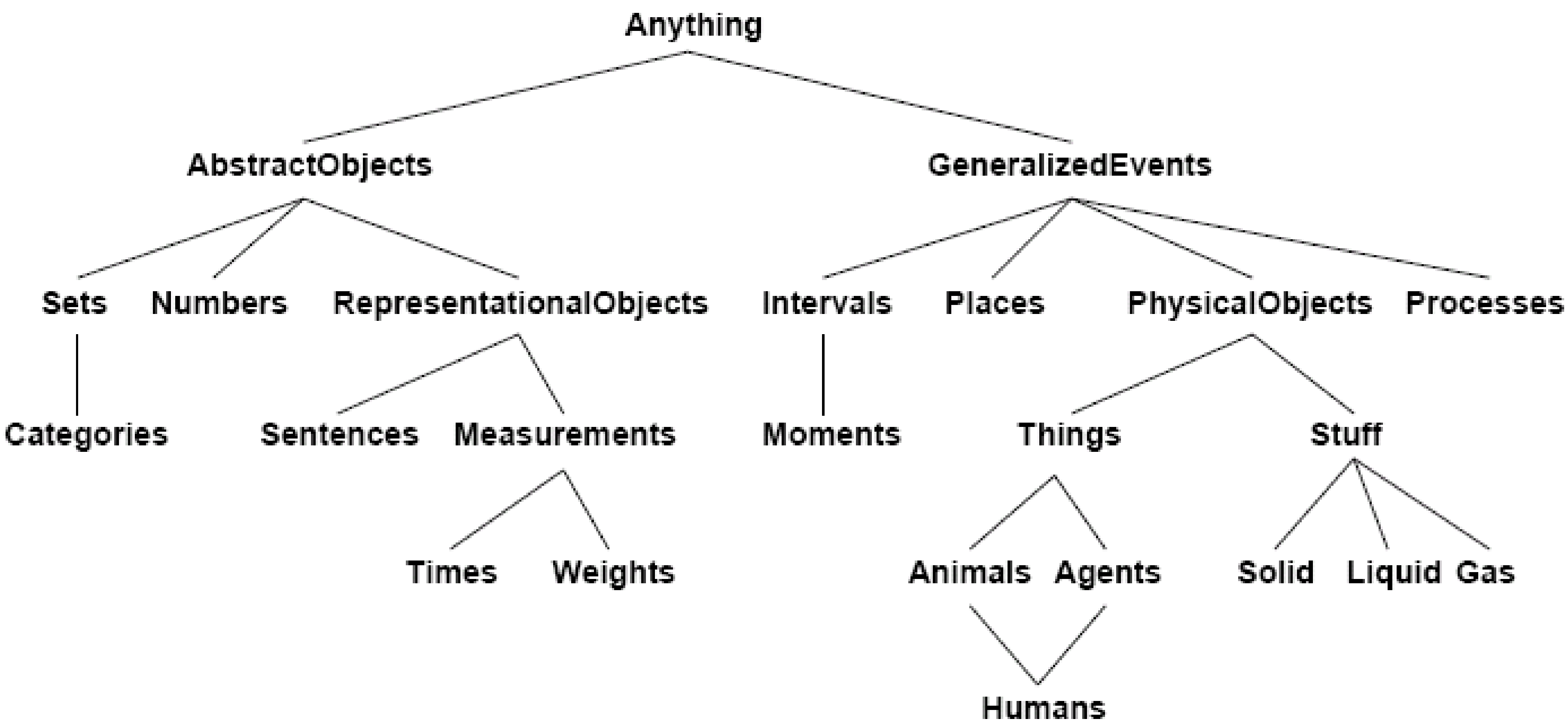
Ontological Engineering

Complex domains

- ✓ Require very general & flexible representations
  - Include actions, time, physical objects, beliefs, ….
- ✓ Ontological engineering
  - The process of finding/deciding on representations for these abstract concepts
- ✓ Somewhat like Knowledge Engineering
  - But at a larger scale
  - Generalized to a more complex real world

Ontological Engineering

Upper ontology



Ontological Engineering

- Our initial discussion may have omitted them
  - ✓ There are limitations to a FOL representation
  - ✓ e.g. there are exceptions to generalizations
    - They hold only to some degree
    - “Tomatoes are red"
    - But there are green, yellow, even purple tomatoes
  - ✓ Exceptions & uncertainty are important topics
    - However, they are orthogonal to a general ontology
      - Their discussions are deferred
      - e.g. uncertainty later in Ch 13

Ontological Engineering

Goal

- ✓ Develop a general purpose ontology
- ✓ One that's usable in any special purpose domain
  - With the addition of domain-specific axioms
- ✓ Deal with any sufficiently demanding domains
  - Different areas of knowledge must be unified
  - Involves several areas simultaneously

We will use it later for the internet shopping agent example

Ontological Engineering

- We begin with objects & categories
  - ✓ Organizing objects into categories
    - Though physical interaction involves individual objects
    - Reasoning processes need to operate at the level of categories

Categories and objects

Categories and objects {12.2}

- KR requires the organization of objects into categories
  - ✓ Interaction at the level of the object
  - ✓ Reasoning at the level of categories

- Category = set of its members
- Categories can be represented by FOL
  - ✓ Predicates: Basketball(b)
  - ✓ Or reify the category as an object: Basketballs. Memberof(b, Basketballs)

Categories and objects {12.2}

- Categories serve to organize and simplify the knowledge base through inheritance
  - ✓ All instances of food are edible
  - ✓ fruit is a subclass of food
  - ✓ apples is a subclass of fruit
  - ✓ then an apple is edible.
- Subclass relations organize categories into a taxonomy, or taxonomic hierarchy

FOL & Categories

- **Expressiveness of FOL**
  - ✓ **Express relations between categories**
    - ***Disjoint***
      - **No members in common between categories**
    - ***Exhaustive decomposition***
      - **Any individual must be in one of the categories**
    - ***Partition***
      - **An exhaustive disjoint decomposition of a category**

FOL and categories {12.2}

1. State facts & quantify over members

- An object is a **member** of a category

$$BB_9 \in Basketballs$$

FOL and categories {12.2}

1. State facts & quantify over members

- An object is a **member** of a category
  - ✓ $BB_9 \in Basketballs$
- A category is a **subclass** of another category
  - ✓ **Basketball is a kind of ball :**

FOL and categories {12.2}

1. State facts & quantify over members

- An object is a **member** of a category
  - ✓ $BB_9 \in Basketballs$
- A category is a **subclass** of another category
  - ✓ $Subsetof(Basketballs, Balls)$, $Basketballs \subset Balls$

FOL and categories {12.2}

1. State facts & quantify over members

- An object is a **member** of a category
  - ✓ $BB_9 \in Basketballs$
- A category is a **subclass** of another category
  - ✓ $Subsetof(Basketballs, Balls)$, $Basketballs \subset Balls$
- All **members** of a category have some **properties**
  - ✓ Basketballs are round.

FOL and categories {12.2}

1. State facts & quantify over members

- An object is a **member** of a category
  - ✓ $BB_9 \in Basketballs$
- A category is a **subclass** of another category
  - ✓ $Subsetof(Basketballs, Balls)$, $Basketballs \subset Balls$
- All **members** of a category have some **properties** ∈
  - ✓ $(x \quad Basketballs) \Rightarrow Spherical(x)$

FOL and categories {12.2}

1. State facts & quantify over members

- An object is a **member** of a category
  - ✓ $BB_9 \in Basketballs$
- A category is a **subclass** of another category
  - ✓ $Subsetof(Basketballs, Balls)$, $Basketballs \subset Balls$
- All **members** of a category have some **properties**
  - ✓ $(x \in Basketballs) \Rightarrow Spherical(x)$
- All members of a category can be recognizedby some properties
  - ✓ A round orange ball with diameter 9.5”is a basketball

FOL and categories {12.2}

1. State facts & quantify over members

- An object is a **member** of a category
  - ✓ $BB_9 \in Basketballs$
- A category is a **subclass** of another category
  - ✓ $Subsetof(Basketballs, Balls)$, $Basketballs \subset Balls$
- All **members** of a category have some **properties**
  - ✓ $(x \in Basketballs) \Rightarrow Spherical(x)$
- All members of a category can be recognized by *some properties*

$$Orange(x) \land Round(x) \land Diameter(x)=9.5'' \land x \in Balls \Longrightarrow x \in BasketBalls$$

FOL and categories {12.2}

- An object is a **member** of a category
  - ✓ $BB_9 \in Basketballs$
- A category is a **subclass** of another category
  - ✓ $Subsetof(Basketballs, Balls)$, $Basketballs \subset Balls$
- All **members** of a category have some **properties**
  - ✓ $(x \in Basketballs) \Rightarrow Spherical(x)$
- All members of a category can be recognizedby some properties

$$Orange(x) \land Round(x) \land Diameter(x)=9.5'' \land x \in Balls \Longrightarrow x \in BasketBalls$$

- A category as a whole has some properties.
  - ✓ Dogs are among Domesticated Species

FOL and categories {12.2}

- An object is a **member** of a category
  - ✓ $BB_9 \in Basketballs$
- A category is a **subclass** of another category
  - ✓ $Subsetof(Basketballs, Balls)$, $Basketballs \subset Balls$
- All **members** of a category have some **properties**
  - ✓ $(x \in Basketballs) \Rightarrow Spherical(x)$
- All members of a category can be recognizedby some properties

$$Orange(x) \wedge Round(x) \wedge Diameter(x)=9.5'' \wedge x \in Balls \Longrightarrow x \in BasketBalls$$

A category as a whole has some properties.

- ✓ $Dogs \in DomesticatedSpecies$

Relations between categories {12.2}

- **Two or more categories are disjoint if they have no members in common:**
  - ✓ *Disjoint({animals, vegetables})*
  - ✓ *Disjoint(s)* $\Leftrightarrow$ ____________________________

Relations between categories {12.2}

- **Two or more categories are disjoint if they have no members in common:**
  - ✓ Disjoint({animals, vegetables})

Disjoint(s) $\Leftrightarrow$

$$(\forall c_1, c_2 \;\; c_1 \in s \wedge c_2 \in s \wedge c_1 \neq c_2 \;\; Intersection(c_1, c_2) = \{\})$$

Relations between categories {12.2}

- **Two or more categories are disjoint if they have no members in common:**
  - ✓ *Disjoint({animals, vegetables})*

$$Disjoint(s) \Leftrightarrow \forall C_1, C_2 (C_1 \in s \land C_2 \in s \land C_1 \neq C_2 \Longrightarrow Intersection(C_1, C_2) = \{ \})$$

- **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:**
  - ✓ *ExhaustiveDecomposition({Americans, Canadians, Mexicans}, NorthAmericans).*
  - ✓ *E.D.(s,c)* $\Leftrightarrow$ ______________________________

Relations between categories {12.2}

- Two or more categories are **disjoint** if they have no members in common:
  - ✓ *Disjoint({animals, vegetables})*

$$Disjoint(s) \Leftrightarrow \forall C_1, C_2 (C_1 \in s \wedge C_2 \in s \wedge C_1 \neq C_2 \Longrightarrow Intersection(C_1, C_2) = \{\ \})$$

- 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**:
  - ✓ *ExhaustiveDecomposition({Americans, Canadians, Mexicans}, NorthAmericans).*

$$E.D.(s,c) \Leftrightarrow (\forall\, i\ i \in c \Rightarrow \exists\, C_2\quad C_2 \in s \wedge i \in C_2)$$

Relations between categories {12.2}

- **A** **partition** **is a disjoint exhaustive decomposition:**
- *Partition({Males, Females}, Animals)*
- *Partition(s,c)* $\Leftrightarrow$ ________________

Relations between categories {12.2}

- **A** **partition** **is a disjoint exhaustive decomposition:**
- *Partition({Males, Females}, Animals)*

$$Partition(s,c) \Leftrightarrow Disjoint(s) \land E.D.(s,c)$$

Define a category {12.2}

- **Categories can be defined by providing necessary and sufficient conditions for membership**

  - ✓ *example: a bachelor is an unmarried adult male*

$$x \in Bachelors \Leftrightarrow Unmarried\,(x) \wedge x \in Adults \wedge x \in Males$$

Physical composition {12.2.1}

- One object may be **part of** another:
  - ✓ Use a **partof** relation
  - ✓ Allows grouping of objects into partof hierarchies
  - ✓ Similar to the subset, subclass hierarchy of categories
  - ✓ *Partof(bucharest,romania)*
  - ✓ *Partof(romania,easterneurope)*
  - ✓ *Partof(easterneurope,europe)*

  - ✓ Properties: the partof relation is reflexive and transitive
    - *Partof(x, x)*
    - *Partof(x, y)* $\wedge$ *partof(y, z)* $\Rightarrow$ *partof(x, z)*
    - Allows the inference: *partof(bucharest, europe)*

Physical Composition

- **Categories of composite objects**
  - ✓ **structural relations among parts**
  - ✓ **Example: a biped has 2 legs attached to a body**

$$\begin{aligned}&\text{Biped}(a) \Rightarrow \exists\, l_1\, l_2\, b\ \text{Leg}(l_1) \land \text{Leg}(l_2) \land \text{Body}(b) \land \\ &\quad \text{PartOf}(l_1, a) \land \text{PartOf}(l_2, a) \land \text{PartOf}(b, a) \land \\ &\quad \text{Attached}(l_1, b) \land \text{Attached}(l_2, b) \land l_1 \neq l_2 \\ &\quad \land [\forall l_3\ \text{Leg}(l_3) \land \text{PartOf}(l_3, a) \Rightarrow (l_3 = l_1 \lor l_3 = l_2)]\end{aligned}$$

  - ✓ **The awkward specification of "exactly two" relaxed later**

Physical Composition {12.2.1}

- **There may also be composite objects**
  - **that have parts but *no specific structure,*use the idea of a *bunch***
  - *BunchOf ({Apple1, Apple2, Apple3})*
  - **a *composite, unstructured* object**
    - **define BunchOf in terms of PartOf relation**
    - **each element of s is a part of the BunchOf(s)**

$$\forall x, x \in s \Rightarrow PartOf(x, BunchOf(s))$$

Measurements

Measurements

- **Measured properties of objects**
  - ✓ **Real objects have length, width, mass, cost, ...**
  - ✓ **We refer to values assigned to these properties as *measures***
  - ✓ **Express them by**
    - **Combining a units function with a number**
    - *Length*(l_1) = *Cm*(3.8)
    - *Cost*$(BasketBall_7)$ = \$(29)

Measurements

Measured properties of objects

- ✓ **We can do conversions**
  - **Between different units for the same property**
  - **By equating multiples of 1 unit to another**
  - *Cm(2.54 x d) = Inches(d)*

$Length(L_1)=Inches(1.5)=Centimeters(3.81)$

$Centimeters(2.54\times d)=Inches(d)$

$Diameter(Basketball_{12})=Inches(9.5)$

$ListPrice(Basketball_{12})=\$(19)$

$d\in Days \Longrightarrow Duration(d)=Hours(24)$

Measurements

- Measured properties of objects
  - ✓ One issue with the approach is that many "measures" have no standard scale
    - Beauty, difficulty, tastiness, ...
  - ✓ The key aspect of measures is not their numeric values, but the ability to order them、 compare them with ordering symbols >, <

Measurements

Measured properties of objects

- ✓ $e1 \in Exercise \wedge e2 \in Exercise \wedge Write(Norvig, e1) \wedge Write(Russell, e2)$

  $\Rightarrow Difficult(e1) > Difficult(e2)$

- ✓ $e1 \in Exercise \wedge e2 \in Exercise \wedge Difficult(e1) > Difficult(e2)$

  $\Rightarrow ExpertedScore(e1) < ExpertedScore((e2)$

Substances & Objects

Substances & Objects

- Some things we wish to reason about
  - ✓ Can be subdivided, yet remain the same
  - ✓ We'll use a generic term:
    - Stuff (opposed to thing)
  - ✓ Stuff
    - Corresponds to mass nouns of Natural Language
  - ✓ Things
    - Correspond to count nouns
  - ✓ Water vs Book, Butter vs Dog, ...

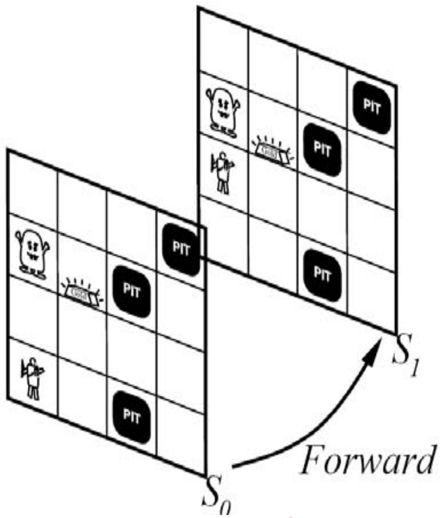
Substances & Objects

- Mass nouns (stuff) vs count nouns (things)
  - ✓ In general, for stuff, mass nouns:
    - Intrinsic properties define the substance
    - These are unchanged under subdivision: colour, taste, ...
      - At least under macroscopic subdivision
  - ✓ While for things, countable nouns:
    - We include extrinsic properties
    - That change under subdivision: weight, length, shape, ...

Situation calculus

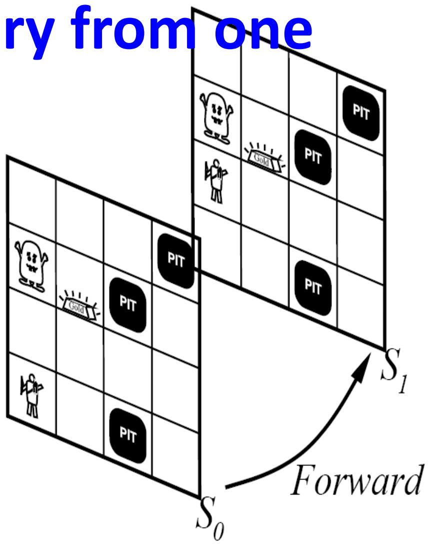
Actions, Situations & Events

- Reasoning about outcomes of actions
  - ✓ Is central to the idea of a KB agent
  - ✓ Recall that when we mentioned action sequences for the Wumpus World agent Forward(), Turn(Right), ...
  - ✓ We required a different copy of an action description for each time the action was executed
- Use the ontology of **situation calculus**
  - ✓ Situations are the results of executing actic



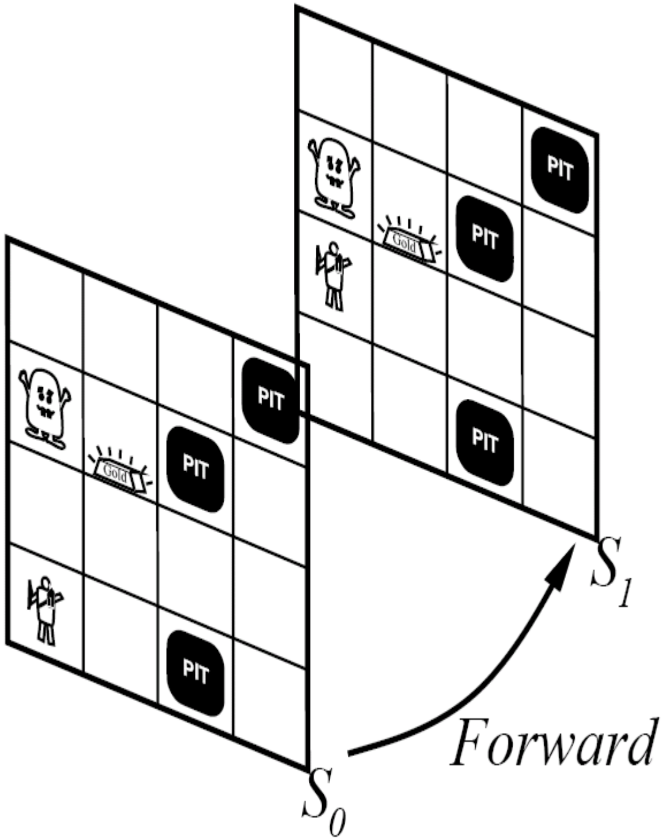
Situation Calculus

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



Dr.Yanmei Zheng

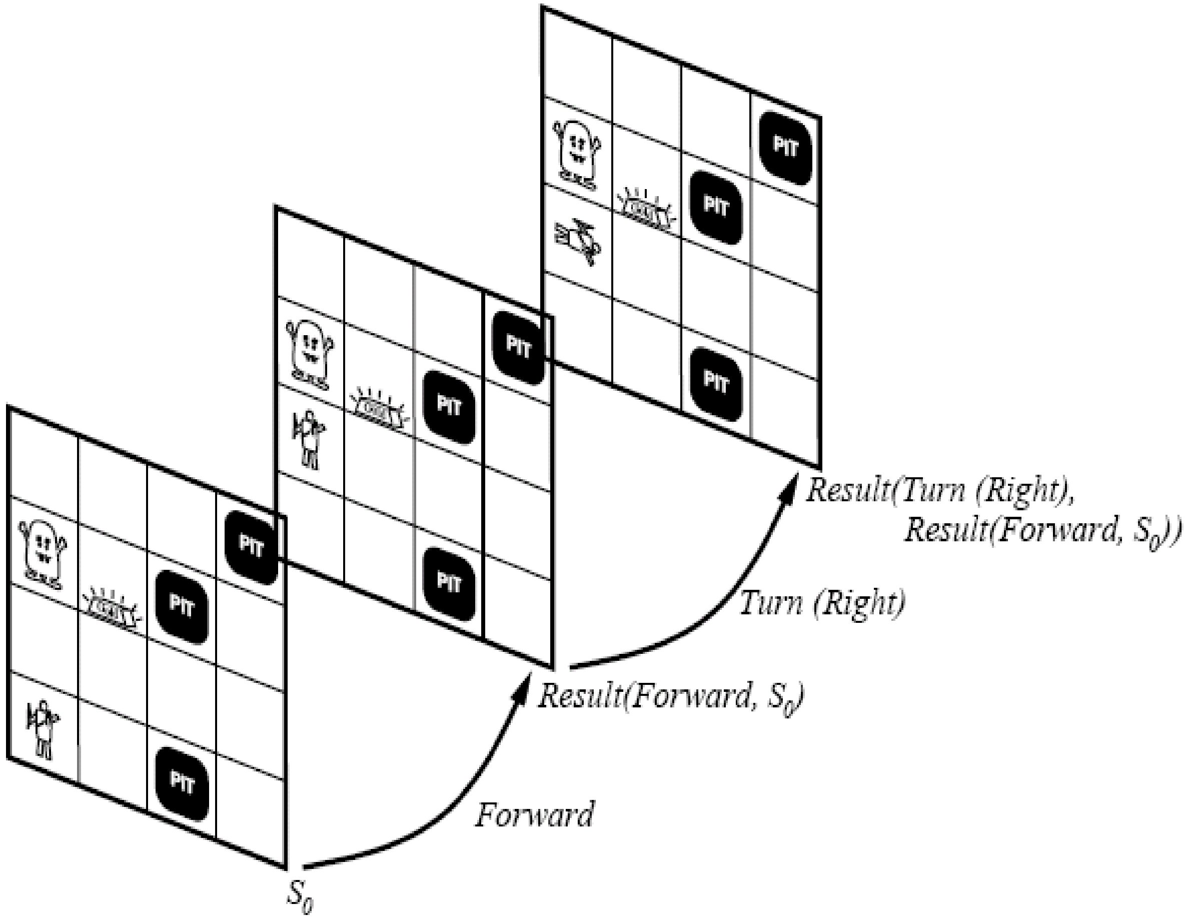
Situation Calculus



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

Situation Calculus

- Situation calculus & the Wumpus World



Events {12.3}

- Now we add an ability to reason about action sequences
  - ✓ A. executing the empty sequence leaves the situation unchanged
    - Result([ ], s) = s
  - ✓ B. executing a non-empty sequence is the same as executing the first action then executing the rest in the resulting situation
    - Result([a]seq, s) = Result(seq, Result(a, s))

Events {12.3}

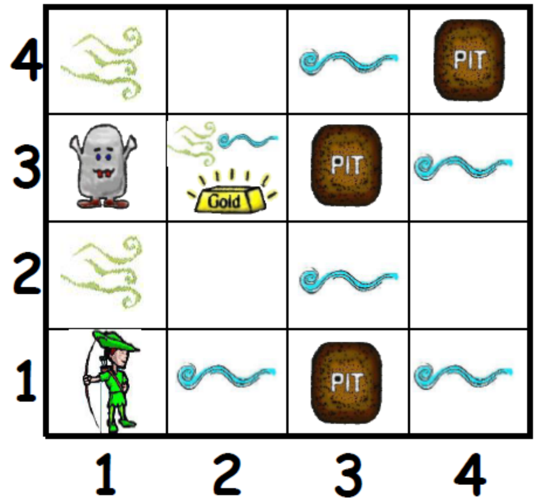
- Describing change in situation calculus
  - ✓ The simplest version
    - Uses *possibility and effect* axioms for each action
  - ✓ A *possibility axiom* & an *effect axiom* specify
    - A. when it is possible to execute an action
    - B. what happens when a possible action is executed

The general forms of these axioms

- ✓ A possibility axiom
  - Preconditions $\Rightarrow$ Poss(a, s)
- ✓ An effect axiom
  - Poss(a, s) $\Rightarrow$ changes resulting from action a

Situation Calculus

- A situation calculus example
  - ✓ Change over time in Wumpus World
  - ✓ Conventions
    - 1. Omit universal quantifiers if scope is a whole sentence
    - 2. Simplify the agent's moves as just *Go*
    - 3. Variables & their ranges
      - s ranges over situations
      - a ranges over actions
      - g ranges over gold
      - x & y range over locations
      - o ranges over objects (including the Agent)



Dr.Yanmei Zheng

Situation Calculus

- **A situation calculus example: Wumpus World**
  - ✓ **Sample possibility axioms**

    $At(Agent, x, s) \wedge Adjacent(x, y) \Rightarrow Poss(Go(x,y), s)$

    $Gold(g) \wedge At(Agent, x, s) \wedge At(g, x, s) \Rightarrow Poss(Grab(g), s)$
  - ✓ **Sample effects axioms**

    $Poss(Go(x,y), s) \Rightarrow At(Agent, y, Result(Go(x, y), s))$

    $Poss(Grab(g),s) \Rightarrow Holding(g, Result(Grab(g), s))$
  - ✓ **these apparently allow an agent**
    - **to make a plan to get the gold**

Situation Calculus

- To make a plan to get the gold requires
  - ✓ Representing **that gold's location stays the same**
  - ✓ The need to represent things that stay the same & to do it *efficiently*

  - ✓ One possible approach is to **use *frame axioms***
    - Explicit axioms to say what stays the same
      - Example: agent's moving does not affect objects not held

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

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.

Thank you

End of Chapter 12-2