

# Foundations of Artificial Intelligence

## 10. Knowledge Representation: Modeling with Logic

Concepts, Actions, Time, and all the Rest

Joschka Boedecker and Wolfram Burgard and Marco Ragni



Albert-Ludwigs-Universität Freiburg

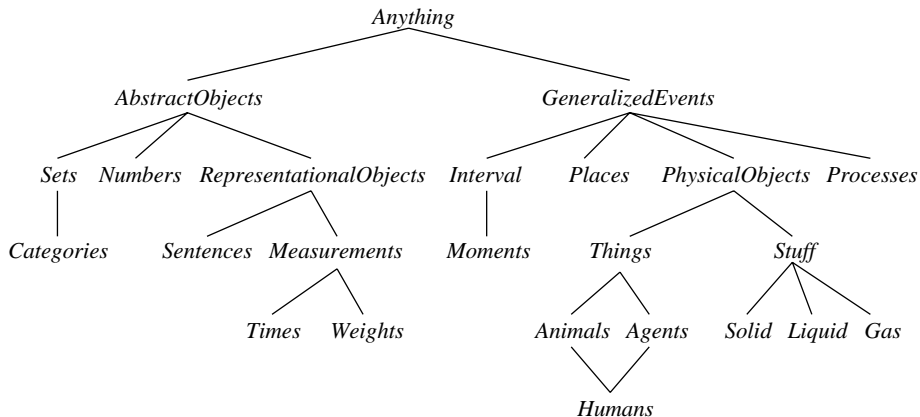
June 10th & 15th, 2016

- 1 Knowledge Representation and Reasoning
- 2 Concepts & Description Logics
- 3 Logical Agents for the WUMPUS World
- 4 Situation Calculus
- 5 The Frame Problem
- 6 Time Intervals and Their Description

- Often, our agents need **knowledge** before they can start to act intelligently
- They then also need some **reasoning component** to exploit the knowledge they have
- Examples:
  - Knowledge about the important **concepts** in a domain
  - Knowledge about **actions** one can perform in a domain
  - Knowledge about **temporal relationships** between events
  - Knowledge about the world and how properties are related to actions

- We need to describe the objects in our world using [categories](#)
- Necessary to establish a common category system for different applications (in particular on the web)
- There are a number of quite general categories everybody and every application uses

# The Upper Ontology: A General Category Hierarchy



- How to describe more specialized things?
- Use definitions and/or necessary conditions referring to other already defined *concepts*:
  - A *parent* is a *human* with at least one *child*.
- More complex description:
  - A *proud-grandmother* is a *human*, which is *female* with at least two *children* that are in turn *parents* whose *children* are all *doctors*.

Typical questions of interest:

- **Subsumption**: Determine whether one description is more general than (subsumes) the other
- **Classification**: Create a subsumption hierarchy
- **Satisfiability**: Is a description satisfiable?
- **Instance relationship**: Is a given object instance of a concept description?
- **Instance retrieval**: Retrieve all objects for a given concept description

# Special Properties of Description Logics

- Semantics of description logics (DLs) can be given using ordinary PL1
- Alternatively, DLs can be considered as modal logics
- Reasoning for most DLs is much more efficient than for PL1
- Nowadays, W3C standards such as OWL (formerly DAML+OIL) are based on description logics



**function**  $\text{KB-AGENT}(\textit{percept})$  **returns** an *action*

**persistent:**  $KB$ , a knowledge base

$t$ , a counter, initially 0, indicating time

$\text{TELL}(KB, \text{MAKE-PERCEPT-SENTENCE}(\textit{percept}, t))$

$\textit{action} \leftarrow \text{ASK}(KB, \text{MAKE-ACTION-QUERY}(t))$

$\text{TELL}(KB, \text{MAKE-ACTION-SENTENCE}(\textit{action}, t))$

$t \leftarrow t + 1$

**return** *action*

Query ( $\text{MAKE-ACTION-QUERY}$ ):  $\exists x \textit{Action}(x, t)$

A variable assignment for  $x$  in the WUMPUS world example should give the following answers: *turn(right)*, *turn(left)*, *forward*, *shoot*, *grab*, *release*, *climb*.

... only react to percepts.

Example of a percept statement (at time 5):

$$\text{Percept}(\text{stench}, \text{breeze}, \text{glitter}, \text{none}, \text{none}, 5)$$

1.  $\forall b, g, u, c, t[\text{Percept}(\text{stench}, b, g, u, c, t) \Rightarrow \text{Stench}(t)]$

$$\forall s, g, u, c, t[\text{Percept}(s, \text{breeze}, g, u, c, t) \Rightarrow \text{Breeze}(t)]$$
$$\forall s, b, g, u, c, t[\text{Percept}(s, b, \text{glitter}, u, c, t) \Rightarrow \text{AtGold}(t)]$$

...

2. Step: Choice of action

$$\forall t[\text{AtGold}(t) \Rightarrow \text{Action}(\text{grab}, t)]$$

...

**Note:** Our reflex agent does not know when it should climb out of the cave and cannot avoid an infinite loop.

... have an internal model

- of all basic aspects of their environment,
- of the executability and effects of their actions,
- of further basic laws of the world, and
- of their own goals.

Important aspect: How does the world change?

→ **Situation calculus**: (McCarthy, 63).

- A way to describe **dynamic worlds** with PL1.
- **States** are represented by terms.
- The world is in state  $s$  and can only be altered through the execution of an **action**:  $do(a, s)$  is the **resulting situation**, if  $a$  is executed.
- Actions have **preconditions** and are described by their **effects**.
- Relations whose truth value changes over time are called **fluents**.  
Represented through a predicate with two arguments: the fluent and a state term. For example,  $At(x, s)$  means, that in situation  $s$ , the agent is at position  $x$ .  $Holding(y, s)$  means that in situation  $s$ , the agent holds object  $y$ .
- **Atemporal** or **eternal** predicates, e.g.,  $Portable(gold)$ .

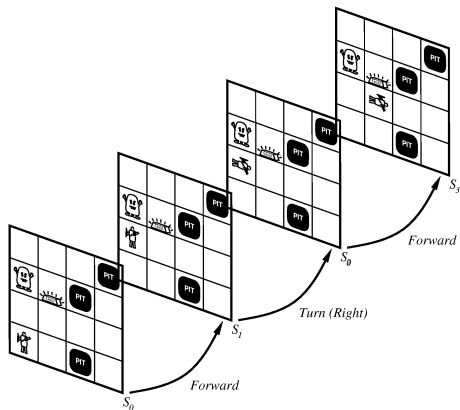
# Example: WUMPUS-World

Let  $s_0$  be the initial situation  
and

$$s_1 = do(\text{forward}, s_0)$$

$$s_2 = do(\text{turn}(\text{right}), s_1)$$

$$s_3 = do(\text{forward}, s_2)$$



**Preconditions:** In order to pick something up, it must be both present and portable:

$$\forall x, s [Poss(\text{grab}(x), s) \Leftrightarrow Present(x, s) \wedge Portable(x)]$$

In the WUMPUS-World:

$$Portable(\text{gold}), \forall s [AtGold(s) \Rightarrow Present(\text{gold}, s)]$$

**Positive effect axiom:**

$$\forall x, s [Poss(\text{grab}(x), s) \Rightarrow Holding(x, do(\text{grab}(x), s))]$$

**Negative effect axiom:**

$$\forall x, s \neg Holding(x, do(\text{release}(x), s))$$

# The Frame Problem

We had:  $Holding(gold, s_0)$ .

Following situation:  $\neg Holding(gold, do(release(gold), s_0))$ ?

We had:  $\neg Holding(gold, s_0)$ .

Following situation:  $\neg Holding(gold, do(turn(right), s_0))$ ?

- We must also specify which *fluents* remain unchanged!
- The frame problem: Specification of the properties that *do not* change as a result of an action.

→ Frame axioms must also be specified.

# Number of Frame Axioms

$$\forall a, x, s [ \text{Holding}(x, s) \wedge (a \neq \text{release}(x)) \Rightarrow \text{Holding}(x, \text{do}(a, s)) ]$$

$$\forall a, x, s [ \neg \text{Holding}(x, s) \wedge \{ (a \neq \text{grab}(x)) \vee \neg \text{Poss}(\text{grab}(x), s) \} \\ \Rightarrow \neg \text{Holding}(x, \text{do}(a, s)) ]$$

Can be very expensive in some situations, since  $O(|F| \times |A|)$  axioms must be specified,  $F$  being the set of fluents and  $A$  being the set of actions.



# Successor-State Axioms

A more *elegant way* to solve the frame problem is to *fully describe the successor situation*:

*true* after action

$\Leftrightarrow$  [ action made it true or, already true and the action did not *falsify* it ]

Example for *grab*:

$\forall a, x, s [ \text{Holding}(x, \text{do}(a, s))$

$\Leftrightarrow \{ (a = \text{grab}(x) \wedge \text{Poss}(a, s)) \vee (\text{Holding}(x, s) \wedge a \neq \text{release}(x)) \}$ ]

Can also be automatically compiled by only giving the effect axioms (and then applying *explanation closure*). Here we suppose that only certain effects can appear.

# Limits of this Version of Situation Calculus

- No explicit **time**. We cannot discuss how long an action will require, if it is executed.
  - **Only one agent**. In principle, however, several agents can be modeled.
  - **No parallel** execution of actions.
  - **Discrete situations**. No continuous actions, such as moving an object from A to B.
  - **Closed world**. Only the agent changes the situation.
  - **Determinism**. Actions are always executed with absolute certainty.
- Nonetheless, sufficient for many situations.

We can describe the temporal occurrence of event/actions:

- **absolute** by using a date/time system
- **relative** with respect to other event occurrences
- **quantitatively**, using time measurements (5 secs)
- **qualitatively**, using comparisons (before/overlaps)

- Allen proposed a calculus about **relative order** of *time intervals*
- Allows us to describe, e.g.,
  - Interval  $I$  **occurs before** interval  $J$
  - Interval  $J$  **occurs before** interval  $K$
- and to conclude
  - Interval  $I$  **occurs before** interval  $K$

→ 13 jointly exhaustive and pair-wise disjoint relations between intervals

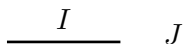
# Allen's 13 Interval Relation



$I < J, J > I$   
*before/after*



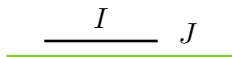
$I m J, J m^{-1} I$   
*meets*



$I o J, J o^{-1} I$   
*overlaps*



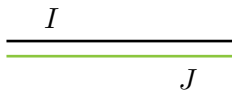
$I s J, J s^{-1} I$   
*starts*



$I d J, J d^{-1} I$   
*during*



$I f J, J f^{-1} I$   
*finishes*



$I = J$

- Using Allen's relation system one can describe temporal configurations as follows:

$$X < Y, Y o Z, Z > X$$

- One can also use disjunctions (unions) of temporal relations:

$$X(<, m)Y, Y(o, s)Z, Z > X$$

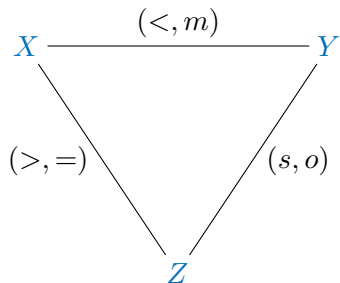
# Reasoning in Allen's Relations System

How do we reason in Allen's system

- Checking whether a set of formulae is **satisfiable**
- Checking whether a temporal formula **follows logically**

→ Use a **constraint propagation technique** for CSPs with infinite domains (3-consistency), based on *composing relations*

# Constraint Propagation



$$X < Y \text{ s } Z = X \text{ } Z$$

$$X < Y \text{ o } Z = X \text{ } Z$$

$$X \text{ m } Y \text{ s } Z = X \text{ } Z$$

$$X \text{ m } Y \text{ o } Z = X \text{ } Z$$

Do that for every triple until nothing changes anymore, then CSP is 3-consistent



# Concluding Remarks: Use of Logical Formalisms

- In many (but not all) cases, full inference in PL1 is simply too slow (and therefore too unreliable).
  - Often, special (logic-based) representational formalisms are designed for specific applications, for which specific inference procedures can be used. Examples:
    - Description logics for representing conceptual knowledge.
    - James Allen's time interval calculus for representing qualitative temporal knowledge.
    - Planning: Instead of situation calculus, there is a specialized calculus (STRIPS) that allows us to address the frame problem efficiently.
- Generality vs. efficiency
- In every case, logical semantics is important!