



Lecture 18 of 42

Knowledge Representation and Ontologies Discussion: Situational Calculus

Friday, 06 October 2006

William H. Hsu

Department of Computing and Information Sciences, KSU

KSOL course page: <http://snipurl.com/v9v3>

Course web site: <http://www.kddresearch.org/Courses/Fall-2006/CIS730>

Instructor home page: <http://www.cis.ksu.edu/~bhsu>

Reading for Next Class:

Sections 10.3, Russell & Norvig 2nd edition



Lecture Outline

- **Today's Reading: Sections 10.3, R&N 2e**
- **Next Week's Reading: Sections 10.4 – 10.6, R&N 2e**
- **Wednesday: Knowledge Rep, Ontologies, Situational Calculus**
- **Today**
 - * **Temporal logic**
 - * **Semantic networks**
 - * **Description Logics**
- **Next Week**
 - * **Description Logics**
 - * **Defeasible reasoning: nonmonotonic logic**
 - * **Intro to Planning**
- **Midterm Exam: 16 Oct 2006**
 - * **Remote students: have exam agreement faxed to DCE**
 - * **Exam will be faxed to proctors Wednesday or Friday**





Describing Actions [1]: Frame, Qualification, and Ramification Problems

“Effect” axiom—describe changes due to action

$$\forall s \text{ AtGold}(s) \Rightarrow \text{Holding}(\text{Gold}, \text{Result}(\text{Grab}, s))$$

“Frame” axiom—describe non-changes due to action

$$\forall s \text{ HaveArrow}(s) \Rightarrow \text{HaveArrow}(\text{Result}(\text{Grab}, s))$$

Frame problem: find an elegant way to handle non-change

(a) representation—avoid frame axioms

(b) inference—avoid repeated “copy-overs” to keep track of state

Qualification problem: true descriptions of real actions require endless caveats—what if gold is slippery or nailed down or ...

Ramification problem: real actions have many secondary consequences—what about the dust on the gold, wear and tear on gloves, ...

Adapted from slides by S. Russell, UC Berkeley



Describing Actions [2]: Successor State Axioms

Successor-state axioms solve the representational frame problem

Each axiom is “about” a predicate (not an action per se):

$$\begin{aligned} P \text{ true afterwards} &\Leftrightarrow [\text{an action made } P \text{ true} \\ &\vee P \text{ true already and no action made } P \text{ false}] \end{aligned}$$

For holding the gold:

$$\begin{aligned} \forall a, s \text{ Holding}(\text{Gold}, \text{Result}(a, s)) &\Leftrightarrow \\ &[(a = \text{Grab} \wedge \text{AtGold}(s)) \\ &\vee (\text{Holding}(\text{Gold}, s) \wedge a \neq \text{Release})] \end{aligned}$$

Adapted from slides by S. Russell, UC Berkeley



Making Plans

Initial condition in KB:

$At(Agent, [1, 1], S_0)$

$At(Gold, [1, 2], S_0)$

Query: $ASK(KB, \exists s \text{ Holding}(Gold, s))$

i.e., in what situation will I be holding the gold?

Answer: $\{s / \text{Result}(Grab, \text{Result}(Forward, S_0))\}$

i.e., go forward and then grab the gold

This assumes that the agent is interested in plans starting at S_0 and that S_0 is the only situation described in the KB

Adapted from slides by S. Russell, UC Berkeley



Making Plans: A Better Way

Represent plans as action sequences $[a_1, a_2, \dots, a_n]$

$PlanResult(p, s)$ is the result of executing p in s

Then the query $ASK(KB, \exists p \text{ Holding}(Gold, PlanResult(p, S_0)))$
has the solution $\{p / [Forward, Grab]\}$

Definition of $PlanResult$ in terms of $Result$:

$\forall s \text{ PlanResult}([], s) = s$

$\forall a, p, s \text{ PlanResult}([a|p], s) = \text{PlanResult}(p, \text{Result}(a, s))$

Planning systems are special-purpose reasoners designed to do this type of inference more efficiently than a general-purpose reasoner

Adapted from slides by S. Russell, UC Berkeley



First-Order Logic: Summary

First-order logic:

- objects and relations are semantic primitives
- syntax: constants, functions, predicates, equality, quantifiers

Increased expressive power: sufficient to define wumpus world

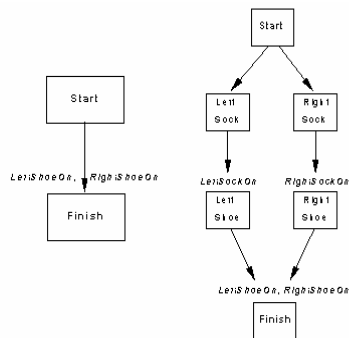
Situation calculus:

- conventions for describing actions and change in FOL
- can formulate planning as inference on a situation calculus KB

Adapted from slides by S. Russell, UC Berkeley



Partially-Ordered Plans



A plan is complete iff every precondition is achieved

A precondition is achieved iff it is the effect of an earlier step
and no possibly intervening step undoes it

Adapted from slides by S. Russell, UC Berkeley



Summary Points

- Previously: Logical Representations and Theorem Proving
 - * Propositional, predicate, and first-order logical languages
 - * Proof procedures: forward and backward chaining, resolution refutation
- Today: Introduction to Classical Planning
 - * Search vs. planning
 - * STRIPS axioms
 - ⇒ Operator representation
 - ⇒ Components: preconditions, postconditions (ADD, DELETE lists)
- Next Monday: More Classical Planning
 - * Partial-order planning (NOAH, etc.)
 - * Limitations



Terminology

- Classical Planning
 - * Planning versus search
 - * Problematic approaches to planning
 - ⇒ Forward chaining
 - ⇒ Situation calculus
 - * Representation
 - ⇒ Initial state
 - ⇒ Goal state / test
 - ⇒ Operators
- Efficient Representations
 - * STRIPS axioms
 - ⇒ Components: preconditions, postconditions (ADD, DELETE lists)
 - ⇒ Clobbering / threatening
 - * Reactive plans and policies
 - * Markov decision processes

Adapted from slides by S. Russell, UC Berkeley

