# Lecture 18 of 42

## KR – Situational Calculus, Frame Problems
## Discussion: Midterm Exam Review

**Friday, 07 October 2007**


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-2007/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: 25 Oct 2007
  - Remote students: have exam agreement faxed to DCE
  - Exam will be faxed to proctors Wednesday or Friday

## 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



## Description Logics: Horrocks & Sattler, ECAI

Description Logics—Basics, Applications, and More

Ian Horrocks
Information Management Group
University of Manchester, UK

Ulrike Sattler
Institut für Theoretische Informatik
TU Dresden, Germany

# Decidability Revisited

- See: Section 9.7 Sidebar, p. 288 R&N
- Duals (Why?)

$$L_{VALID} \qquad \overline{L_{VALID}}$$
$$\overline{L_{SAT}} \qquad L_{SAT}$$

- Complexity Classes
- Understand: Reduction to $L_d$, $L_H$



# Successor State Axioms: Review

Successor-state axioms solve the representational frame problem

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

P true afterwards ⇔ [an action made P true
∨ P true already and no action made P false]

For holding the gold:

$$\forall a, s \;\; Holding(Gold, Result(a,s)) \Leftrightarrow$$
$$[(a = Grab \wedge AtGold(s))$$
$$\vee\, (Holding(Gold, s) \wedge a \neq Release)]$$

Adapted from slides by S. Russell, UC Berkeley

## The Planning Problem: Review

Represent <u>plans</u> as action sequences $[a_1, a_2, \ldots, a_n]$

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

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

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

$\forall s \;\; PlanResult([\,], s) = s$

$\forall a, p, s \;\; PlanResult([a|p], s) = PlanResult(p, Result(a, s))$

<u>Planning systems</u> 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



## State Space versus Plan Space

Standard search: node = concrete world state
Planning search: node = <u>partial plan</u>

Defn: <u>open condition</u> is a precondition of a step not yet fulfilled

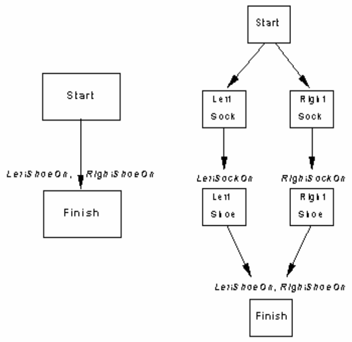
Operators on partial plans:

- <u>add a link</u> from an existing action to an open condition
- <u>add a step</u> to fulfill an open condition
- <u>order</u> one step wrt another

Gradually move from incomplete/vague plans to complete, correct plans

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



## Lecture Outline

- Today's Reading
  - Sections 11.5 – 11.9, Russell and Norvig
  - References: to be posted on class web board
- Next Week's Reading: Chapter 12, Russell and Norvig
- Previously: Logical Representations and Theorem Proving
- Today: More Classical Planning
  - STRIPS axioms (review)
  - Partial-order planning (NOAH, etc.)
  - Limitations of POP
    - Need for abstraction
    - Hierarchical abstraction (ABSTRIPS)
- Midterm Exam: Friday 19 Oct 2007, in class
  - Two pages of notes allowed
  - Remote students: have exam agreement faxed to DCE
  - Exam will be faxed to proctors Friday morning
- Next Week: More Planning – Conditional and Reactive

# POP Algorithm [1]: Sketch

**function** POP(*initial, goal, operators*) **returns** *plan*

  *plan* ← MAKE-MINIMAL-PLAN(*initial, goal*)
  **loop do**
    **if** SOLUTION?( *plan*) **then return** *plan*
    $S_{need}$, $c$ ← SELECT-SUBGOAL( *plan*)
    CHOOSE-OPERATOR( *plan, operators*, $S_{need}$, $c$)
    RESOLVE-THREATS( *plan*)
  **end**

---

**function** SELECT-SUBGOAL( *plan*) **returns** $S_{need}$, $c$

  pick a plan step $S_{need}$ from STEPS( *plan*)
    with a precondition $c$ that has not been achieved
  **return** $S_{need}$, $c$

Adapted from slides by S. Russell, UC Berkeley



# POP Algorithm [2]: Subroutines and Properties

**procedure** CHOOSE-OPERATOR(*plan, operators*, $S_{need}$, $c$)

  **choose** a step $S_{add}$ from *operators* or STEPS( *plan*) that has $c$ as an effect
  **if** there is no such step **then fail**
  add the causal link $S_{add} \xrightarrow{c} S_{need}$ to LINKS( *plan*)
  add the ordering constraint $S_{add} \prec S_{need}$ to ORDERINGS( *plan*)
  **if** $S_{add}$ is a newly added step from *operators* **then**
    add $S_{add}$ to STEPS( *plan*)
    add $Start \prec S_{add} \prec Finish$ to ORDERINGS( *plan*)

---

**procedure** RESOLVE-THREATS(*plan*)

  **for each** $S_{threat}$ that threatens a link $S_i \xrightarrow{c} S_j$ in LINKS( *plan*) **do**
    **choose** either
      *Demotion:* Add $S_{threat} \prec S_i$ to ORDERINGS( *plan*)
      *Promotion:* Add $S_j \prec S_{threat}$ to ORDERINGS( *plan*)
    **if not** CONSISTENT( *plan*) **then fail**
  **end**

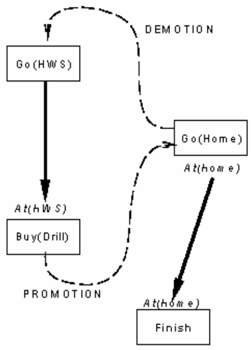
POP is sound, complete, and systematic (no repetition)

Extensions for disjunction, universals, negation, conditionals

Adapted from slides by S. Russell, UC Berkeley

# Clobbering and Promotion / Demotion

A clobberer is a potentially intervening step that destroys the condition achieved by a causal link. E.g., $Go(Home)$ clobbers $At(HWS)$:




Demotion: put before $Go(HWS)$

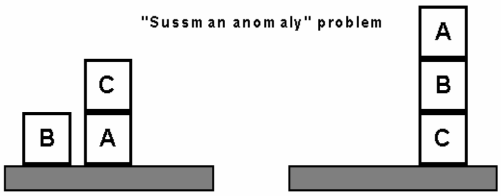
Promotion: put after $Buy(Drill)$

Adapted from slides by S. Russell, UC Berkeley



# Example: Blocks World [1] Specification

"Sussman anomaly" problem



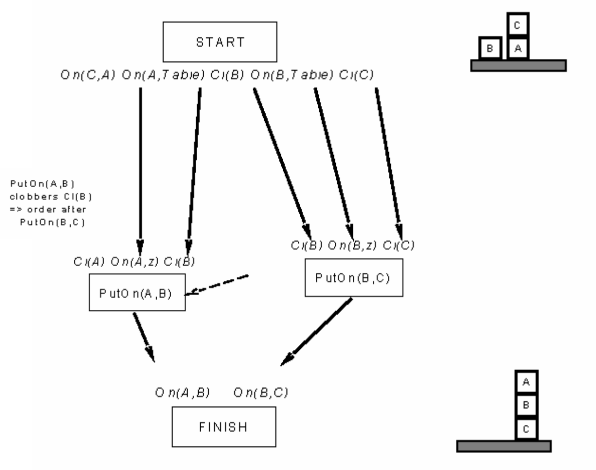

Start State Goal State

| PutOn(x,y) | PutOnTable(x) |
|---|---|
| Preconditions: $Clear(x)\ On(x,z)\ Clear(y)$ | Preconditions: $Clear(x)\ On(x,z)$ |
| Effects: $\sim On(x,z)\ \sim Clear(y)\ Clear(z)\ On(x,y)$ | Effects: $\sim On(x,z)\ Clear(z)\ On(x,Table)$ |

+ several inequality constraints

Adapted from slides by S. Russell, UC Berkeley

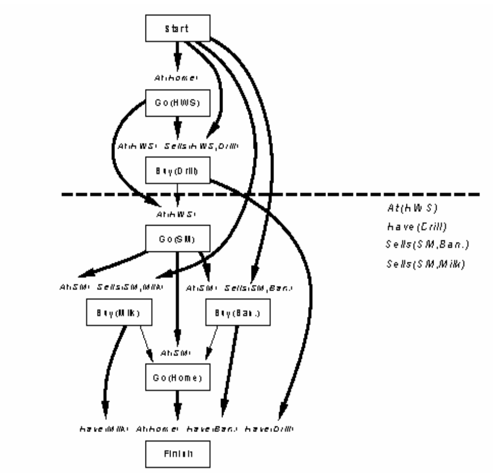
# Example: Blocks World [2]
# POP Trace




Adapted from slides by S. Russell, UC Berkeley



# Example:
# Preconditions for Remaining Plan




Adapted from slides by S. Russell, UC Berkeley

# Hierarchical Abstraction Planning

- Need for Abstraction
  - Question: *What is wrong with uniform granularity?*
  - Answers (among many)
    - Representational problems
    - Inferential problems: inefficient plan synthesis
- Family of Solutions: Abstract Planning
  - But what to abstract in "problem environment", "representation"?
    - Objects, obstacles (quantification: later)
    - Assumptions (closed world)
    - Other entities
    - *Operators*
    - *Situations*
  - Hierarchical abstraction
    - See: Sections 12.2 – 12.3 R&N, pp. 371 – 380
    - Figure 12.1, 12.6 (examples), 12.2 (algorithm), 12.3-5 (properties)

**Adapted from Russell and Norvig**



# Universal Quantifiers in Planning

- Quantification *within* Operators
  - Chapter 11, R&N 2e
  - Examples
    - Shakey's World
    - Blocks World (R&N; also in Winston, Rich and Knight)
    - Grocery shopping
  - Others (from projects?)
- Exercise for Next Tuesday: *Blocks World*

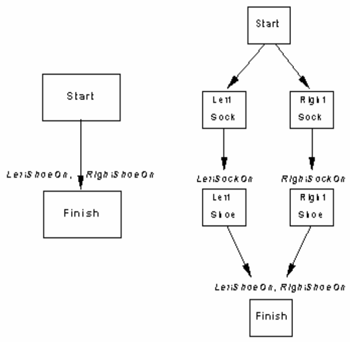
# Practical Planning

- The Real World
  - *What can go wrong with classical planning?*
  - *What are possible solution approaches?*
- Conditional Planning
- Monitoring and Replanning (Next Time)

**Adapted from Russell and Norvig**



# 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**