Planning: introduction

Madalina Croitoru

University of Montpellier

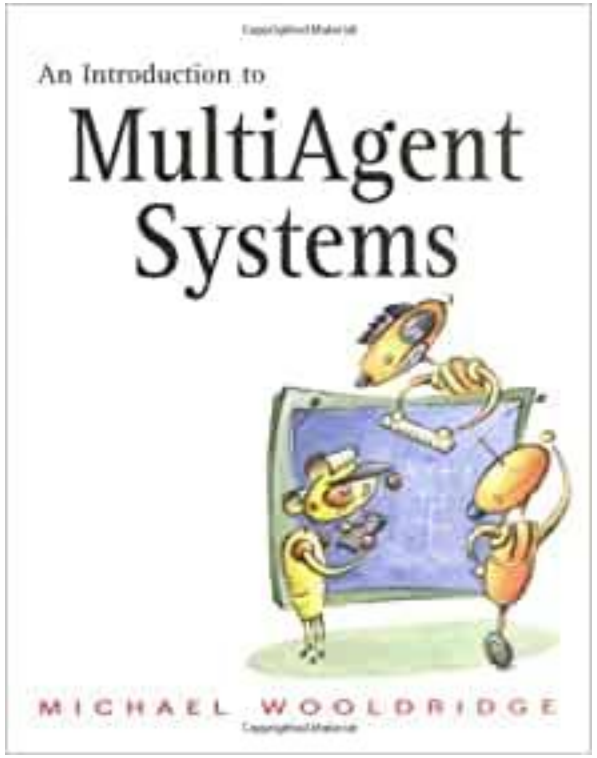
Quick first order logic notations

- In all cities there exists a dog which is not kind
- In all cities and for all dogs there does not exist somebody owning them
- In all cities and for all dogs if there does not exist a veterinary all dogs are unvaccinated

$[\forall c \; \exists d \; DOG(d); CITY(c); LiveIn(d,c)] \rightarrow \neg KIND(d)$

$\forall d \; \forall c \; [DOG(d); CITY(c); LiveIn(d,c) \rightarrow \neg \; \exists p \; [OWNER(p,d,c); PERSON(p)]]$

$\forall d \; \forall c \; [DOG(d); CITY(c); LiveIn(d,c) \neg \exists v \; [VET(v, d,c)] \rightarrow UNVACC(d,c)]$



Based on “An Introduction to MultiAgent Systems” by Michael Wooldridge, John Wiley & Sons, 2002.

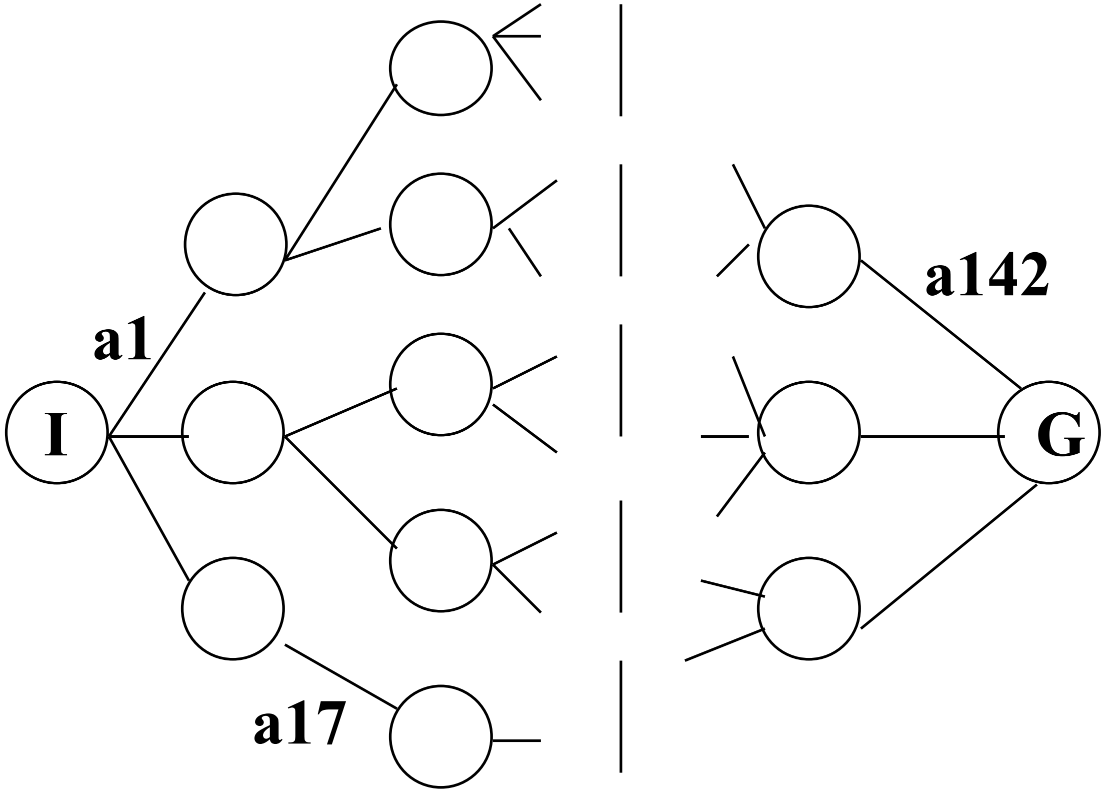
http://www.csc.liv.ac.uk/˜mjw/pubs/imas/

More Problems with Deductive Systems

- The “logical approach” that was presented implies <u>adding and removing things from a database</u>
- That’s not pure logic
  - Early attempts at creating a “planning agent” tried to use true logical deduction to the solve the problem

Planning Systems (in general)

- Planning systems find a sequence of actions that transforms an initial state into a goal state

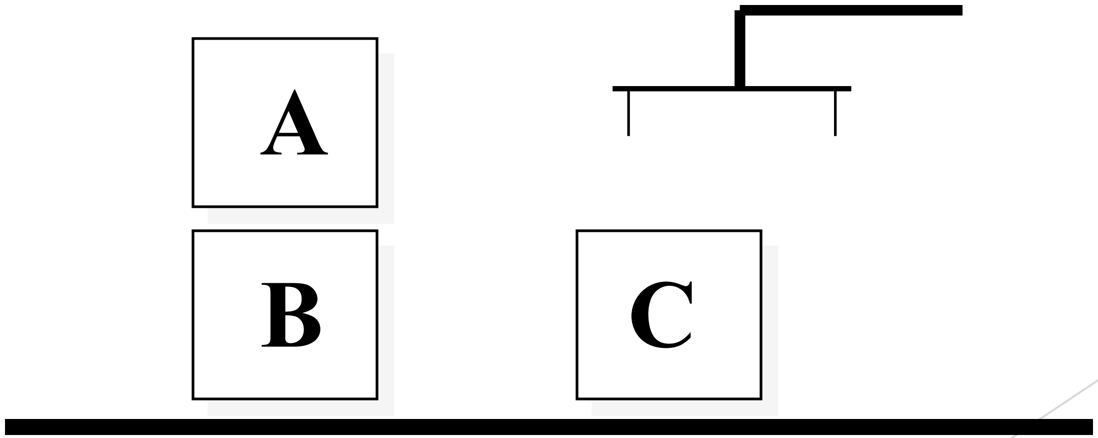


The Blocks World

- The Blocks World (today) consists of equal sized blocks on a table
- A robot arm can manipulate the blocks using these intuitive actions:
  - UNSTACK(a, b)
  - STACK(a, b)
  - PICKUP(a)
  - PUTDOWN(a)

The Blocks World

- Predicates describing the world are indexed by the current state:
  - ON(A,B,s)
  - ONTABLE(B,s)
  - ONTABLE(C,s)
  - CLEAR(A,s)
  - CLEAR(C,s)
  - ARMEMPTY(s)



Logical Formulas to Describe Facts Always True of the World

- And of course we can write general logical truths relating the predicates:

$[\exists x \forall s \text{ HOLDING}(x,s)] \rightarrow \neg \text{ARMEMPTY}(s)$

$\forall x \forall s [\text{ ONTABLE}(x,s) \rightarrow \neg \exists y [\text{ON}(x,y,s)]]$

$\forall X \forall s [\neg \exists y [\text{ON}(y, x,s)] \rightarrow \text{CLEAR}(x,s)]$

So…how do we use theorem-proving techniques to construct plans?

Green’s Method

- Add state variables to the predicates, and use a function DO that maps actions and states into new states
  DO: A x S $\rightarrow$ S

- Example:
  DO(UNSTACK(x, y), S) is a new state

UNSTACK

- So to characterize the action UNSTACK we could write:

$[\, CLEAR(x, s) \wedge ON(x, y, s) \,] \rightarrow [HOLDING(x, DO(UNSTACK(x,y),s)) \wedge CLEAR(y, DO(UNSTACK(x,y),s))]$

- We can prove that if S0 is such that:

$ON(A,B,S0) \wedge ONTABLE(B,S0) \wedge CLEAR(A, S0)$ then

$HOLDING(A,DO(UNSTACK(A,B),S0)) \wedge CLEAR(B,DO(UNSTACK(A,B),S0))$

More Proving

- The proof could proceed further; if we characterize PUTDOWN:

$$HOLDING(x,s) \rightarrow ONTABLE(x,DO(PUTDOWN(x),s))$$

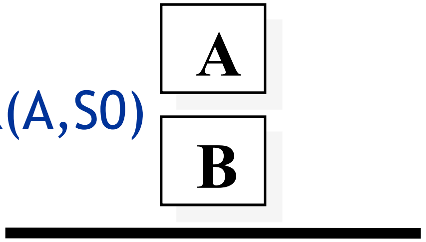
- Then we could prove:
  ONTABLE(A,
  DO(PUTDOWN(A),
  DO(UNSTACK(A,B), S0)))

- The nested actions in this constructive proof give you the plan:

1. UNSTACK(A,B); 2. PUTDOWN(A)

More Proving

- So if we have in our database:
  ON(A,B,S0) ∧ ONTABLE(B,S0) ∧ CLEAR(A,S0)
  and our goal is
  ∃ s(ONTABLE(A, s))



we could use theorem proving to find the plan

- But could I prove:
  ONTABLE(B,
  DO(PUTDOWN(A),
  DO(UNSTACK(A,B), S0)))

The Frame Problem

- How do you determine *what changes* and *what doesn't change* when an action is performed?
- One solution: "Frame axioms" that specify how predicates can remain unchanged after an action
- Example:

1. ONTABLE(z, s) →
   ONTABLE(z,DO(UNSTACK(x,y),s))

Frame Axioms

- Problem: Unless we go to a higher-order logic, Green's method forces us to write many frame axioms
- Example:
  COLOR(x, c, s) →
  COLOR(x,c,DO(UNSTACK(y,z),s))
- We want to avoid this…other approaches are needed…

AOP and planning

- Much of the interest in agents from the AI community has arisen from Shoham's notion of *agent oriented programming* (AOP)
- AOP a 'new programming paradigm, based on a societal view of computation'. The key idea that informs AOP is that of directly programming agents in terms of intentional notions like belief, desire, and intention
- Planning is essentially automatic programming: the design of a course of action that will achieve some desired goal.
  - Building largely on the early work of Fikes & Nilsson, many planning algorithms have been proposed, and the theory of planning has been well-developed
  - But in the mid 1980s, Chapman established some theoretical results which indicate that AI planners will ultimately turn out to be unusable in any time-constrained system

Exercise

- Continue last week’s work and imagine a goal for your agent. Design the plan and the rules needed.

1. Describe it theoretically using logic.
2. How will you implement it?