GKMCET

Lecture Plan

Subject code & Subject Name: CS2351 & AI **Unit Number: III**

UNIT III-PLANNING

INTRODUCTION

SIMPLE PLANNING AGENT

The agent first generates a goal to achieve and then constructs aplan to achieve it from the Current state

PROBLEMSOLVING TO PLANNING

Representation Using Problem Solving Approach

- Forward search
- Backward search
- Heuristic search

Representation Using Planning Approach

- STRIPS-standard research institute problem solver.
- Representation for states and goals
- Representation for plans
- Situation space and plan space
- Solutions

Why Planning ?

Intelligent agents must operate in the world. They are not simply passive reasoners (Knowledge Representation, reasoning under uncertainty) or problem solvers (Search), they must also acton the world.

We want intelligent agents to act in “intelligent ways”. Taking purposeful actions, predicting the expected effect of such actions, composing actions together to achieve complex goals.

E.g. if we have a robot we want robot to decide what to do; how to act to achieve our goals

Planning Problem

How to change the world to suit our needs

Critical issue: we need to reason about what the world will be like after doing a few actions, not just what it is like now

GOAL: Craig has coffee

CURRENTLY: robot in mailroom, has no coffee, coffee not made, Craig in office etc.

TO DO: goto lounge, make coffee **,...**

PARTIAL ORDER PLANNING

Partial-Order Planning Algorithms

Partially Ordered Plan

• Plan

• Steps

• Ordering constraints

• Variable binding constraints

• Causal links

• POP Algorithm

• Make initial plan

• Loop until plan is a complete

– Select a subgoal

– Choose an operator

– Resolve threats

Choose Operator

• Choose operator(c, Sneeds)

• Choose a step S from the plan or a new step S by instantiating an operator that has c as an effect

• If there's no such step, Fail

• Add causal link S _c Sneeds

• Add ordering constraint S < Sneeds

• Add variable binding constraints if necessary

• Add S to steps if necessary

Nondeterministic choice

• Choose – pick one of the options arbitrarily

• Fail – go back to most recent non-deterministic choice and try a different one that has not been tried before

Resolve Threats

• A step S threatens a causal link Si _c Sj iff ¬ c ∈ effects(S) and it's possible that Si < S < Sj

• For each threat

• Choose

–Promote S : S < Si < Sj

–Demote S : Si < Sj < S

• If resulting plan is inconsistent, then Fail

Threats with Variables

If c has variables in it, things are kind of tricky.

• S is a threat if there is any instantiation of the variables that makes ¬ c ∈ effects(S)

•We could possibly resolve the threat by adding a negative variable binding constraint, saying that two variables or a variable and a constant cannot be bound to one another

• Another strategy is to ignore such threats until the very end, hoping that the variables will become bound and make things easier to deal with

Shopping Domain

• Actions

• Buy(x, store)

– Pre: At(store), Sells(store, x)

– Eff: Have(x)

• Go(x, y)

– Pre: At(x)

– Eff: At(y), ¬At(x)

• Goal

• Have(Milk) ∧ Have(Banana) ∧ Have(Drill)

• Start

• At(Home) ∧ Sells(SM, Milk) ∧ Sells(SM, Banana) ∧ Sells(HW, Drill)

Shopping problem

start
At (Home)
Buy (Drill) Buy (Bananas)
At(HDW) Sells (HDW,D)
Buy (Milk)
At (SM) Sells(SM,M)
finish
Have(D) Have(M) Have(B)
At(SM) Sells(SM,B)
NB: Causal links
imply ordering
of steps
¬At(x2)
GO (SM)

At (x2)
GO (HDW)
At(x1)
¬At(x1)
start
At (Home)
Buy (Drill) Buy (Bananas)
At(HDW) Sells (HDW,D)
Buy (Milk)
At (SM) Sells(SM,M)
finish
Have(D) Have(M) Have(B)
At(SM) Sells(SM,B)
NB: Causal links
imply ordering
of steps
¬At(x2)
GO (SM)
At (x2)
GO (HDW)
At(x1)
¬At(x1)
start
At (Home)
Buy (Drill) Buy (Bananas)
At(HDW) Sells (HDW,D)
Buy (Milk)
At (SM) Sells(SM,M)
finish
Have(D) Have(M) Have(B)
At(SM) Sells(SM,B)
x1=Home x2=Home
NB: Causal links
imply ordering
of steps
!
¬At(x2)
GO (SM)
At (x2)
GO (HDW)
At(x1)

¬At(x1)
start
At (Home)
Buy (Drill) Buy (Bananas)
At(HDW) Sells (HDW,D)
Buy (Milk)
At (SM) Sells(SM,M)
finish
Have(D) Have(M) Have(B)
At(SM) Sells(SM,B)
x1=Home x2=Home
NB: Causal links
imply ordering
of steps
¬At(x2)
GO (SM)
At (x2)
GO (HDW)
At(x1)
¬At(x1)
start
At (Home)
Buy (Drill) Buy (Bananas)
At(HDW) Sells (HDW,D)
Buy (Milk)
At (SM) Sells(SM,M)
finish
Have(D) Have(M) Have(B)
At(SM) Sells(SM,B)
x1=Home x2=Home
NB: Causal links
imply ordering
of steps
start
At (Home)
Buy (Drill) Buy (Bananas)
At(HDW) Sells (HDW,D)
¬At(x2)
GO (SM)
At (x2)
Buy (Milk)

At (SM) Sells(SM,M)
finish
Have(D) Have(M) Have(B)
At(SM) Sells(SM,B)
GO (HDW)
At(x1)
¬At(x1)
x1=Home x2=Home
3
start
At (Home)
Buy (Drill) Buy (Bananas)
At(HDW) Sells (HDW,D)
¬At(x2)
GO (SM)
At (x2)
Buy (Milk)
At (SM) Sells(SM,M)
finish
Have(D) Have(M) Have(B)
At(SM) Sells(SM,B)
GO (HDW)
At(x1)
¬At(x1)
x1=Home x2=Home
x2=HDW
start
At (Home)
Buy (Drill) Buy (Bananas)
At(HDW) Sells (HDW,D)
¬At(x2)
GO (SM)
At (x2)
Buy (Milk)
At (SM) Sells(SM,M)
finish
Have(D) Have(M) Have(B)
At(SM) Sells(SM,B)
GO (HDW)
At(x1)
¬At(x1)

x1=Home x2=Home
x2=HDW
start
At (Home)
Buy (Drill) Buy (Bananas)
At(HDW) Sells (HDW,D)
¬At(x2)
GO (SM)
At (x2)
Buy (Milk)
At (SM) Sells(SM,M)
finish
Have(D) Have(M) Have(B)
At(SM) Sells(SM,B)
GO (HDW)
At(x1)
¬At(x1)
x1=Home x2=Home
x2=HDW

PLANNING GRAPHS

- Levels
- Mutex between actions
- Mutex holds between luents
- Graph plan algorithm

PLANNING AND ACTING IN THE REAL WORLD

- Conditional planning Or Contingency Planning
- Execution monitoring and replanning
- Continuous planning
- Multiagent planning
- Times, schedules, and resources
- Critical path method
- Hierarchical task network planning