Practical Reasoning Agents

Based on "An Introduction to MultiAgent Systems" and slides by Michael Wooldridge

What is Practical Reasoning?

- Practical reasoning is reasoning directed towards actions — the process of figuring out what to do

Practical reasoning is a matter of weighing conflicting considerations for and against competing options, where the relevant considerations are provided by what the agent desires/values/cares about and what the agent believes.

- Practical reasoning is different than *theoretical reasoning* (directed towards beliefs) – e.g., all men are mortal $\wedge$ Socrates is a man $\rightarrow$ Socrates is mortal

The Components of Practical Reasoning

Practical Reasoning = Deliberation + Means-Ends Reasoning

- Deliberation – deciding ***what*** state of affairs we want to achieve — the outputs of deliberation are ***intentions***
- Means-ends reasoning – deciding ***how*** to achieve these states of affairs — the outputs of means-ends reasoning are ***plans***

Computations and Resource Bounds

An agent has a fixed amount of memory and a fixed processor (*resource bounds*) to carry out necessary computations

- Resource bounds impose limits on the size of computations
- *Time constraints* in real environments impose limit on the size of computations

Implications

1. Agents must control its reasoning effectively if it is to perform well
2. Agents cannot deliberate indefinitely (even if the decided state of affairs is not optimal)

Intentions and Desires

Intentions are stronger in influencing actions, than desires

> My desire to play basketball this afternoon is merely a potential influencer of my conduct this afternoon. It must vie with my other relevant desires [. . .] before it is settled what I will do. In contrast, once I intend to play basketball this afternoon, the matter is settled: I normally need not continue to weigh the pros and cons. When the afternoon arrives, I will normally just proceed to execute my intentions.

Intentions in Practical Reasoning

- Intentions drive means-ends reasoning
  - If I have formed an intention, then I will attempt to achieve it (including deciding how to achieve it)
  - If one course of actions fails, then I will typically attempt others
- Intentions persist
  - I will not give up my intentions without good reason
  - Intentions will persist until I have achieved them, I believe I cannot achieve them, or I believe the reason for intention is no longer present

Intentions in Practical Reasoning

- Intentions constraint further deliberation
  - I will not consider options that are incompatible with my current intentions
  - Filter of admissibility
- Intentions influence beliefs upon which practical reasoning is based
  - If I adopt an intention, then I believe I will achieve the intention
  - However, I must also recognize the possibility that I can fail to bring the intention about

Symbolic Representation of Beliefs, Desires and Intentions

Agent has to maintain an explicit symbolic representation (e.g., Prolog facts) of its beliefs, desires, and intentions. Let

- B be a variable for current beliefs
- Bel be the set of all such beliefs
- D be a variable for current desires
- Des be the set of all desires
- I be a variable for current intentions
- Int be the set of all intentions

Deliberation

Deliberation is modeled via two functions

- an option generation function

$$options : 2^{Bel} \times 2^{Int} \rightarrow 2^{Des}$$

- a filtering function

$$filter : 2^{Bel} \times 2^{Des} \times 2^{Int} \rightarrow 2^{Int}$$

An agent's belief update process is modeled trough a belief revision function

$$brf : 2^{Bel} \times Per \rightarrow 2^{Bel}$$

Means-Ends Reasoning

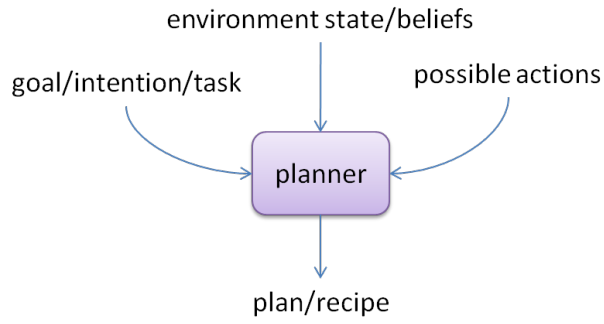
- Means-ends reasoning is the process of deciding how to achieve an **end** (an intention an agent has) using the available **means** (i.e., the actions that an agent can perform)
- Means-end reasoning is better known as planning (AI community)
- Planning is essentially an automatic programming

Planner

Planner is a system that takes as input (representations of) the following:

1. a *goal*, *intention* or *task*
2. the current state of the environment – the agent's *beliefs*
3. the *actions* available to the agent

As output a planner generates a *plan* (course of actions / a "recipe")

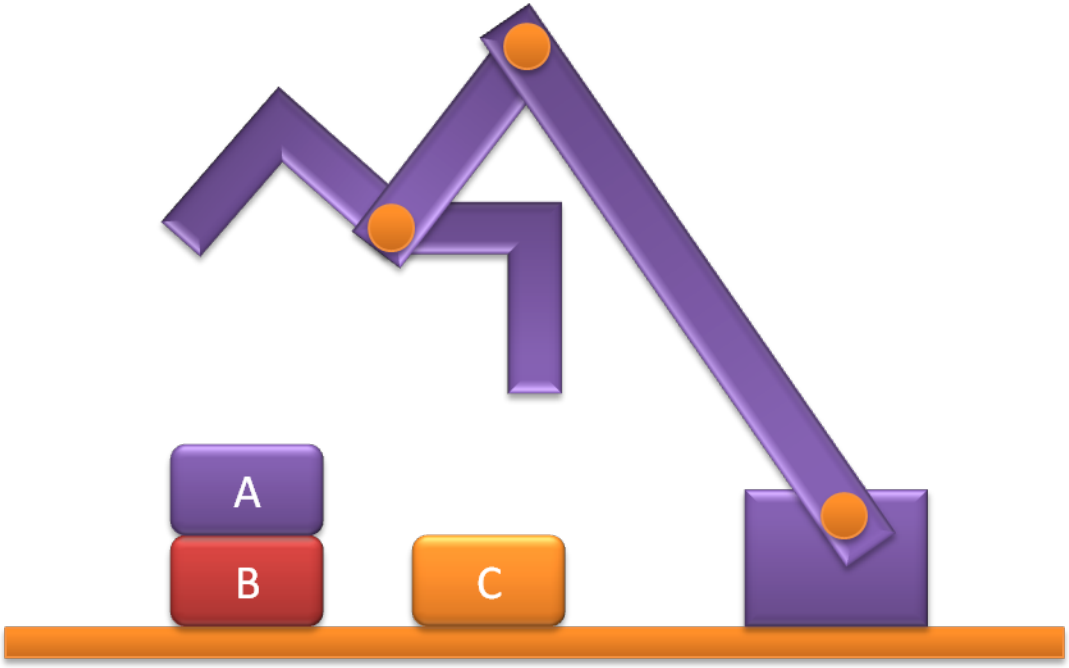


STRIPS

- STRIPS – the first real planner developed in 1960/70s
- Two basic components of STRIPS
  - a model of the world (set of formulas of first-order logic)
  - a set of action schemata describing preconditions and effects of all actions
- The planning algorithm was based on
  - finding a difference between the current state of the world and the goal state
  - reducing this difference by applying an appropriate action

The Blocks World

Contains a robot arm, 3 blocks (A, B, C) of equal size, and a table-top.



Representations in the Blocks World

Predicates for describing the Blocks World

Predicate	Meaning
$On(x, y)$	object x on top of object y
$OnTable(x)$	object x on the table
$Clear(x)$	nothing is on the top of object x
$Holding(x)$	robot arm is holding x
$ArmEmpty$	robot arm is empty (not holding anything)

Representations in the Blocks World

- The current (initial) state of the Blocks World (closed world assumption)

$$\{Clear(A), On(A,B), OnTable(B), OnTable(C), Clear(C)\} \cup \cup \{ArmEmpty\}$$

- A goal to achieve (intention to bring about)

$$\{OnTable(A), OnTable(B), OnTable(C)\}$$

Actions in the Blocks World

Each action is characterized by

- a *name* – which may have arguments
- a *precondition list* – a list of facts that must be true for the action to be executed
- a *delete list* – a list of facts that are no longer true after the action is performed
- an *add list* – a list of facts made true by executing action

Actions in the Blocks World

- The *Stack* action occurs when the robot arm places the object x it is holding is placed on top of object y

$$Stack(x, y)$$

$$\begin{array}{ll} pre & \{Clear(y), Holding(x)\} \\ del & \{Clear(y), Holding(x)\} \\ add & \{ArmEmpty, On(x, y)\} \end{array}$$

- The *UnStack* action occurs when the robot arm picks an object x up from on top of another object y

$$UnStack(x, y)$$

$$\begin{array}{ll} pre & \{On(x, y), Clear(x), ArmEmpty\} \\ del & \{On(x, y), ArmEmpty\} \\ add & \{Holding(x), Clear(y)\} \end{array}$$

Actions in the Blocks World

- The *Pickup* action occurs when the arm picks up an object x from the table

$$Pickup(x)$$

$$\begin{array}{ll} pre & \{Clear(x), OnTable(x), ArmEmpty\} \\ del & \{OnTable(x), ArmEmpty\} \\ add & \{Holding(x)\} \end{array}$$

- The *PutDown* action occurs when the arm places the object x onto the table

$$PutDown(x)$$

$$\begin{array}{ll} pre & \{Holding(x)\} \\ del & \{Holding(x)\} \\ add & \{ArmEmpty, OnTable(x)\} \end{array}$$

Basic Control Structure for a Practical Reasoning Agent

The basic structure of the decision-making process is a loop, in which the agent continually

- observes the world, and updates beliefs
- deliberates to decide what intention to achieve (by first determining the available options and then by filtering)
- uses means-ends reasoning to find a plan to achieve these intentions
- executes the plan

Implementing a Practical Reasoning Agent

Let π be a plan, and *Plans* be the set of all plans (over some set of actions Ac). Then

- $pre(\pi)$ is the precondition of π, and $body(\pi)$ is the body of π
- $empty(\pi)$ is a Boolean function that indicates whether π is empty or not
- $execute(\pi)$ is a procedure that executes π without stopping (i.e., executes each action in the plan body in turn)
- $hd(\pi)$ is the plan made up by the first action in the body of π
- $tail(\pi)$ is the plan made up by all but the first action in the body of π
- $correct(\pi, I, B)$ is a Boolean function that indicates whether π is a correct plan for intentions I given beliefs B

Implementing a Practical Reasoning Agent

```
1  B ← B0
2  I ← I0
3  while true do
4  |    get next percept ρ through see(...) function
5  |    B ← brf(B, ρ)
6  |    D ← options(B, I)
7  |    I ← filter(B, D, I)
8  |    π ← plan(B, I, Ac)
9  |    while not (empty(π) or succeeded(I, B) or impossible(I, B)) do
10 |    |    π ← hd(π)
11 |    |    execute(π)
12 |    |    π ← tail(π)
13 |    |    get next percept ρ through see(...) function
14 |    |    B ← brf(B, ρ)
15 |    |    if reconsider(I, B) then
16 |    |    |    D ← options(B, I)
17 |    |    |    I ← filter(B, D, I)
18 |    |    end
19 |    |    if not correct(π, I, B) then
20 |    |    |    π ← plan(B, I, Ac)
21 |    |    end
22 |    end
23 end
```

Commitments to Ends and Means

- When an option (desire) successfully passes through *filter* function and is chosen by the agent, we say the agent has made a *commitment*
- Commitment implies *temporal persistence* – an intention, once adopted, should not immediately evaporate

How committed an agent should be to its intentions

- How long should an intention persist?
- Under what circumstances should an intention vanish?

Commitments Strategies

Commitment strategy is the mechanism that agent uses to determine when and how to drop intentions

- Blind commitment
  A blindly committed agent will continue to maintain an intention until it believes the intention has actually been achieved. Blind commitment is also sometimes referred to as *fanatical* commitment
- Single-minded commitment
  A single-minded agent will continue to maintain an intention until it believes that either the intention has been achieved, or else that it is no longer possible to achieve the intention
- Open-minded commitment
  An open-minded agent will maintain an intention as long as it is still believed possible

Intention Reconsideration

When should an agent stop to reconsider its intentions (it costs!)?

- An agent that does not stop to reconsider sufficiently often will continue attempting to achieve its intentions even after it is clear that they cannot be achieved or that there is no longer any reason for achieving them
- An agent that constantly reconsiders its intentions may spend insufficient time working to achieve them, and hence runs a risk of never actually achieving them