

CPT302 W4

Practical Reasoning Agents

Practical Reasoning

Practical reasoning 是针对 actions 的推理 – 确定要做什么的过程。

Practical reasoning 是权衡支持和反对竞争选项之间相互冲突的考虑，其中相关考虑因素由 agent 的愿望/价值观/关心的内容以及 agent 所相信的内容提供。

Practical reasoning 和 theoretical reasoning (symbolic reasoning) 不同：

- Practical reasoning 针对 actions
- Theoretical reasoning 针对逻辑 (或 beliefs, 因为 symbolic reasoning agent 的 internal state 类似 beliefs, agent 依据此进行推理)

注：beliefs 可以理解为代理对环境 (或情况) 的理解和认知。

The Components of Practical Reasoning

Practical Reasoning = Deliberation + Means-Ends Reasoning

- Deliberation – 决定我们想要实现/到达的状态，deliberation 的输出是意图 (intentions)
- Means-ends reasoning – 决定如何实现这些状态，其输出是计划 (plans)

Computations and Resource Bounds

Agent 具有固定数量的内存和固定数量的处理器 (resource bounds) 以执行必要的计算。

- 资源界限对计算的规模施加限制
- 实际环境中的 time constraints (时间限制) 限制了计算的规模

这暗示了：

- 代理必须有效地控制其推理，如果它要表现良好
- 代理不能无限期地进行 deliberate (即使确定的状态不是最优状态)

Deliberate and Intentions

Intentions (意图) 对行动的影响比 desires (愿望) 更大 (意图说明真要干, 愿望只是想想)。

Intentions in Practical Reasoning

- Intentions drive means-ends reasoning
 - 如果我已经形成了一个意图, 那么我将尝试实现它 (包括决定如何实现它)
 - 如果一个行动方案失败了, 那么我通常会尝试其他行动方案
- Intentions persist
 - 我不会在没有充分理由的情况下放弃我的意图
 - 意图会一直存在, 直到我实现它们, 或我相信我无法实现它们, 或者我相信意图的原因不再存在
- Intentions constraint further deliberation
 - 我不会考虑与我目前的意图不相容的选择
 - Filter of admissibility (过滤与意图相悖的选项)
- Intentions influence beliefs upon which practical reasoning is based
 - 如果我采用一个意图, 那么我相信我会实现这个意图
 - 但是, 我也必须认识到, 我可能无法实现意图

Symbolic Representation

Agent 必须保持其 beliefs, desires, 和 intentions 的明确的 symbolic representation (例如, Prolog facts, 序言事实)。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

我们通过两个函数对 deliberation 进行建模:

- an option generation function
$$options : 2^{Bel} \times 2^{Int} \rightarrow 2^{Des}$$
- a filtering function
$$filter : 2^{Bel} \times 2^{Des} \times 2^{Int} \rightarrow 2^{Int}$$

注：deliberation 最终产生的是 intention。因此，option generation function 根据当前的 beliefs 和 intention 生成一系列可能的 desires。之后的 filtering function 则产生最终的 intention。

我们通过一个 belief revision function 对代理的 belief 更新过程进行建模：

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

注：Per 代表对环境的观察所得到的信息。

Means-Ends Reasoning

Means-ends reasoning 是决定如何使用可用手段（means，即代理可以执行的操作）来实现目的（end，代理的意图）的过程。

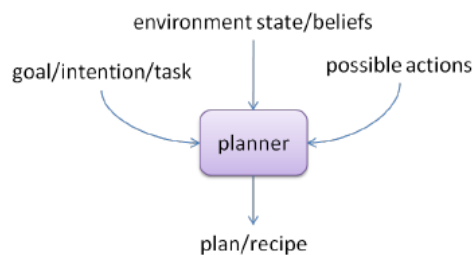
Means-ends reasoning 更广为人知的名字是 planning。Planning 本质上是一种 automatic programming。

Planner

Planner 是一个将以下内容作为输入（表示）的系统：

- a goal, intention or task
- the current state of the environment (the agent's beliefs)
- the actions available to the agent

Planner 的输出是一个 plan (行动方针)。



STRIPS

STRIPS 是第一个真正的 planner。

STRIPS 有两个基本的组件：

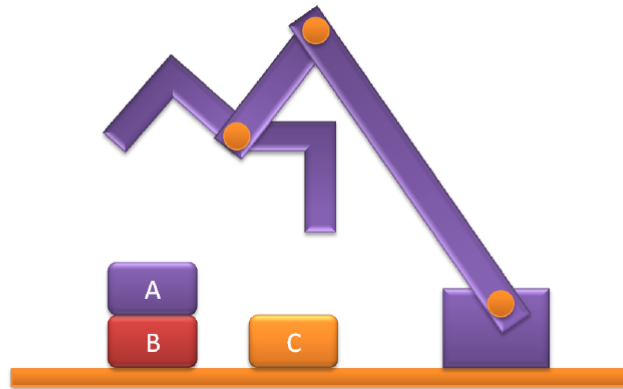
- a model of the world (set of formulas of first-order logic)
- 一组 action schemata (操作纲要) 描述所有操作的前提条件和效果

The planning algorithm was based on:

- 找到当前世界状态和目标状态之间的差异
- 通过采取适当的操作减少这种偏差

Example - 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)

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

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

- A goal to achieve (intention to bring about)

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

注：上面的表达式便是 first-order logic

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

$Stack(x, y)$

$pre \quad \{Clear(y), Holding(x)\}$
 $del \quad \{Clear(y), Holding(x)\}$
 $add \quad \{ArmEmpty, On(x, y)\}$

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

$UnStack(x, y)$

$pre \quad \{On(x, y), Clear(x), ArmEmpty\}$
 $del \quad \{On(x, y), ArmEmpty\}$
 $add \quad \{Holding(x), Clear(y)\}$

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

Pickup(x)

<i>pre</i>	$\{Clear(x), OnTable(x), ArmEmpty\}$
<i>del</i>	$\{OnTable(x), ArmEmpty\}$
<i>add</i>	$\{Holding(x)\}$

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

PutDown(x)

<i>pre</i>	$\{Holding(x)\}$
<i>del</i>	$\{Holding(x)\}$
<i>add</i>	$\{ArmEmpty, OnTable(x)\}$

Basic Control Structure

Decision-making process 的基本结构是一个 loop，其中 agent 持续地：

- 观察世界，更新 beliefs
- deliberates 来决定要实现的 intentions（首先确定可用的 options，然后进行 filtering）
- 通过 means-ends reasoning 来找到一个 plan 以实现 intentions
- 执行 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

注: $body(\pi)$ 是 plan π 的内容; $hd(\pi)$ 只包含 π 的第一个 action; $tail(\pi)$ 包含除了第一个 action 外的所有 actions。

```

1   $B \leftarrow B_0$ 
2   $I \leftarrow I_0$ 
3  while true do
4      get next percept  $\rho$  through see(...) function
5       $B \leftarrow brf(B, \rho)$ 
6       $D \leftarrow options(B, I)$ 
7       $I \leftarrow filter(B, D, I)$ 
8       $\pi \leftarrow plan(B, I, Ac)$ 
9      while not ( $empty(\pi)$  or  $succeeded(I, B)$  or  $impossible(I, B)$ ) do
10          $\pi \leftarrow hd(\pi)$ 
11         execute( $\pi$ )
12          $\pi \leftarrow tail(\pi)$ 
13         get next percept  $\rho$  through see(...) function
14          $B \leftarrow brf(B, \rho)$ 
15         if reconsider( $I, B$ ) then
16              $D \leftarrow options(B, I)$ 
17              $I \leftarrow filter(B, D, I)$ 
18         end
19         if not correct( $\pi, I, B$ ) then
20              $\pi \leftarrow plan(B, I, Ac)$ 
21         end
22     end
23 end

```

注：上面第 9 行的 *succeeded(I, B), impossible(I, B)* 对应下面的 Single-minded commitment (成功，或者不再可能)。这一行代码是 agent 做出承诺。

Commitments

当一个 option (desire) 成功通过 filter function 并被代理选择时，我们说代理已经做出了承诺 (commitment)。

承诺意味着暂时的持久性 (temporal persistence)，一种意图一旦通过，就不应该立即消失。

Commitments Strategies

Commitment strategy 是代理用来确定何时以及如何放弃意图的机制

- Blind commitment (盲目承诺)

一个盲目承诺的代理将继续保持一个意图，直到它相信这个意图已经实现。盲目承诺有时也被称为狂热承诺 (fanatical commitment)

- Single-minded commitment (专心承诺)

一个专心承诺的代理将继续保持一个意图，直到它认为这个意图已经实现，或者不再可能实现这个意图

- Open-minded commitment (开明承诺)

一个开明承诺的代理会保持一个意图，只要它仍然被认为是可能的

注：必须实现 => 已经实现/不可能实现 => 可能实现

Intention Reconsideration

代理何时应该停下来重新考虑其意图 (上面代码的第 15 行)?

- 一个不停下来认真重新考虑的代理往往会继续试图实现其意图，即使很明显它们无法实现，或者不再有任何理由实现它们
- 但不断重新考虑其意图的代理人可能没有足够的时间来实现这些意图，因此有可能永远无法真正实现意图