

ACH2016 - Inteligência Artificial

Aula 12 - Busca Não Determinista e Observação Parcial

Valdinei Freire da Silva

valdinei.freire@usp.br - Bloco A1 100-O

Russell e Norvig, Capítulo 4

AlphaGo



- Solução: busca aleatória heurística + aprendizado por reforço + redes neurais
- Documentário: AlphaGo

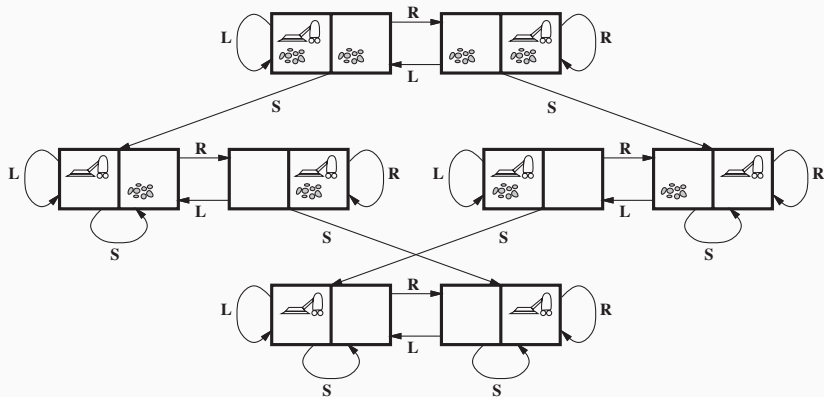
Resolução de Problemas

Ambientes: **completamente observável**, único agente, conhecido, **determinístico**, discreto, sequencial, estático.

Formulação de Problemas

- estado inicial: é o estado no tempo $t = 0$.
- ações possíveis: a função $ACTIONS(s)$ retorna o conjunto de ações que podem ser executadas no estado s .
- modelo de transição: a função $RESULT(s, a)$ retorna o estado resultante de aplicar a ação a no estado s .
- teste de meta: a função $GOAL(s)$ determina se o estado s é um estado meta.
- solução: plano ρ que consiste em uma sequência de ações
 $\rho = a_0, a_1, \dots, a_{T-2}, a_{T-1}$.

Mundo do Aspirador de Pó



Estado Inicial: aspirador do lado esquerdo, dois cômodos sujos

Estado Meta: cômodos limpos

SRS, LRSSLS são planos?

Observação Parcial

Observação Completa: o estado do ambiente sempre é conhecido

Observação Parcial:

- nenhuma observação
- sensores

Como planejar sem conhecer o estado?

Estado de Crença (*belief state*): representa a crença atual do agente de quais estados ele possa estar.

O estado de crença cria um novo espaço de planejamento:

- nenhuma observação: solução, quando houver, é um plano, uma sequência de ações
- sensores: solução é um plano contingencial, isto é, uma política que mapeia cada sequência de observações e ações em uma ação

Espaço de Crença sem Observação

Considere o ambiente adjacente E : $ACTIONS_E(s)$, $RESULT_E(s, a)$, e $GOAL_E(s)$.

- Estados de Crença \mathcal{B} : seja N a quantidade de estados em E , então os estados são todos os possíveis subconjuntos dos estados de E , totalizando 2^N estados de crença;
- Estado Inicial: tipicamente o conjunto de todos os estados de E ;
- Ações Possíveis: assumindo que ações ilegais tem nenhum efeito no mundo, para todo $b \in \mathcal{B}$:

$$ACTIONS(b) = \bigcup_{s \in b} ACTIONS_E(s)$$

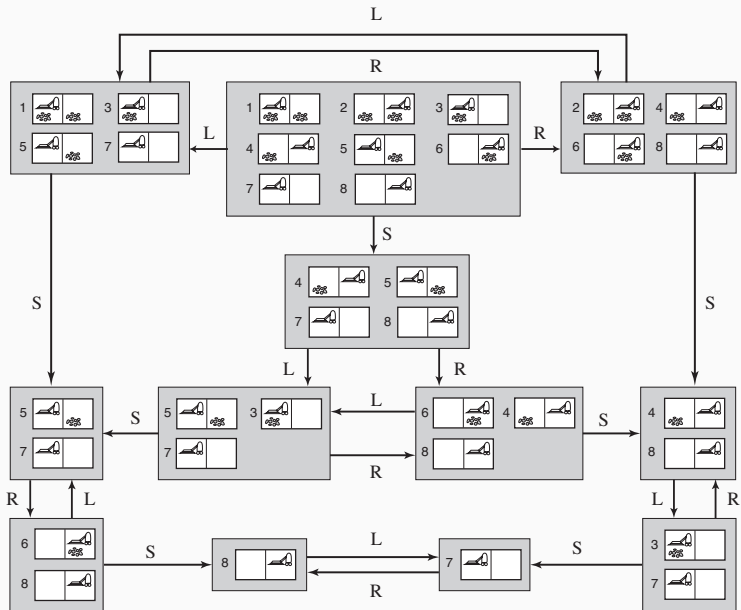
- Modelo de Transição: como o agente não sabe onde está, os estados futuros são todos os estados resultante dos possíveis estados:

$$b' = RESULT(b, a) = \{s' \mid s' = RESULT_E(s, a) \wedge s \in b\}$$

- Teste de Meta: todos estados que formam a crença devem ser estado meta

$$GOAL(b) = \begin{cases} True, & \text{se para todo } s' \in b \text{ tem-se } GOAL_E(s) = True \\ False, & \text{caso contrário} \end{cases}$$

Espaço de Crença: Aspirador de Pó



Espaço de Crença sem Observação

- problemas sem observação também são chamados de conformantes
- nem todo problema é solucionável sem observação
- a solução consiste em uma sequência de ações (um plano)
- considere que ações ilegais tem nenhum efeito no mundo, então se um plano ρ é uma solução para o estado inicial b , então, o mesmo plano é uma solução para qualquer b' subconjunto de b ($b' \subseteq b$)
- se um estado b já foi gerado na árvore de busca, qualquer estado b' que contém o estado b pode ser descartado ($b \subseteq b'$)

Estado de Crença com Observação

Exemplo de Observação Parcial:

- estado completo: (posição do aspirador, condição do cômodo A , condição do cômodo B)
- observação parcial: (posição do aspirador, condição do cômodo ocupado pelo aspirador)
- exemplo: $s = (A, \text{Dirty}, \text{Clean})$ e $o = (A, \text{Dirty})$

Formalização:

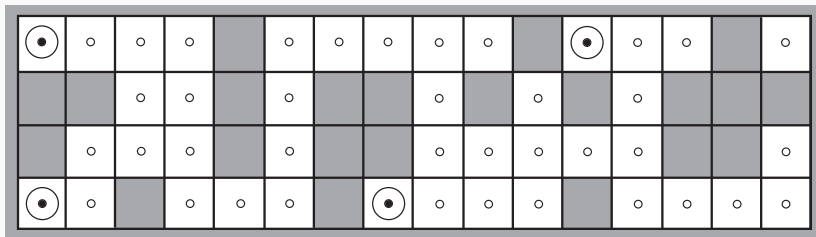
- Predição: é o modelo de transição quando não há observação

$$\hat{b} = \text{PREDICT}(b, a) = \{s' \mid s' = \text{RESULT}_E(s, a) \wedge s \in b\}$$

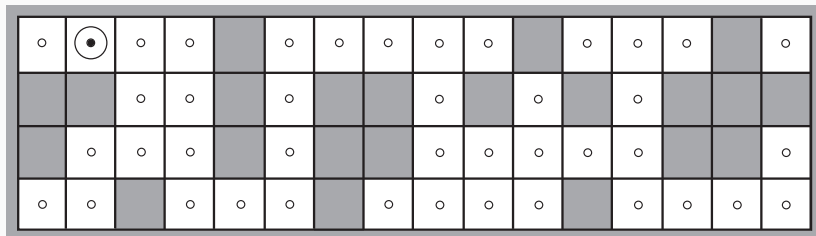
- Atualização: para cada percepção $o \in \mathcal{O}$ determina o estado de crença resultante

$$b' = \text{UPDATE}(\hat{b}, o) = \{s \mid o = \text{PERCEPT}(s) \wedge s \in \hat{b}\}$$

Espaço de Crença: Navegação Robótica



(a) Possible locations of robot after $E_1 = \text{NSW}$



(b) Possible locations of robot After $E_1 = \text{NSW}, E_2 = \text{NS}$

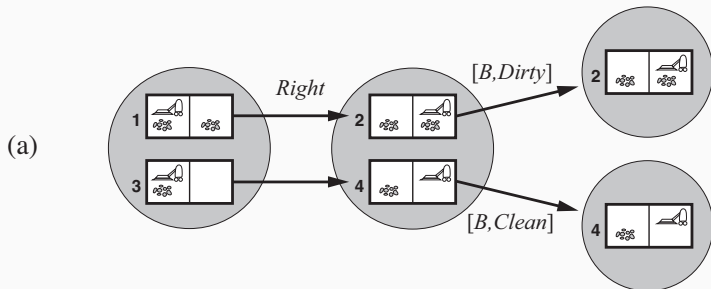
Observação Parcial e Não Determinismo

- Observações Possíveis:

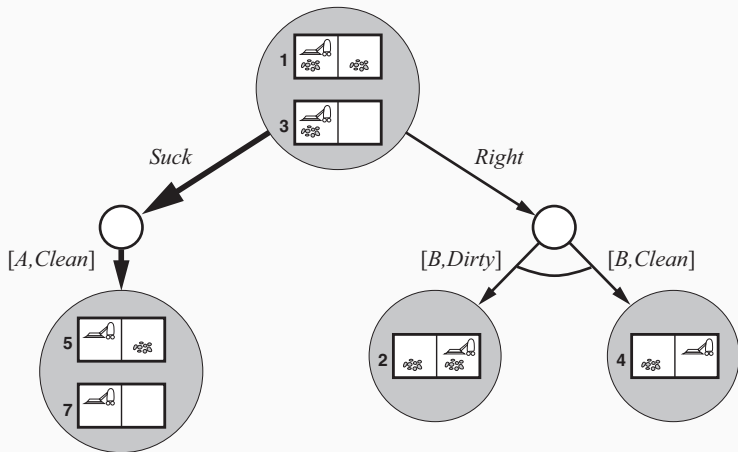
$$\text{POSSIBLE-PERCEPTS}(\hat{b}) = \{o \mid o = \text{PERCEPT}(s) \wedge s \in \hat{b}\}$$

- Resultados Possíveis:

$$\text{RESULTS}(b, a) = \{b' \mid b' = \text{UPDATE}(\text{PREDICT}(b, a), o) \wedge o \in \text{POSSIBLE-PERCEPTS}(\text{PREDICT}(b, a))\}$$



Grafo AND-OR



Ambiente Não-Determinista

- Observação Completa:

$$o = \text{PERCEPT}(s) = s$$

- Modelo de Transição: dados um estado atual e uma ação executada, existem vários possíveis próximos estados

$$s' \in \text{RESULTS}(s, a) \subseteq \mathcal{S}$$

- Solução: plano contingencial

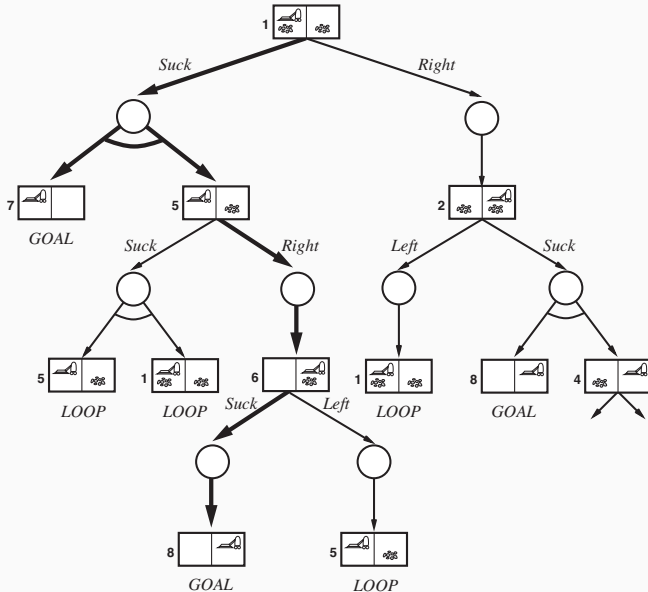
$$[\text{Suck}, \text{if } \text{State} = 5 \text{ then } [\text{Right}, \text{Suck}] \text{ else } []]$$

- Política: mapeia estados em ações

$$\pi : \mathcal{S} \rightarrow \mathcal{A}$$

<i>State</i> = 1	<i>Suck</i>	<i>State</i> = 2	<i>Suck</i>	<i>State</i> = 3	<i>Suck</i>	<i>State</i> = 4	<i>Left</i>
<i>State</i> = 5	<i>Right</i>	<i>State</i> = 6	<i>Suck</i>	<i>State</i> = 7		<i>State</i> = 8	

Aspirador de Pó Errático



function AND-OR-GRAPH-SEARCH(*problem*) **returns** a conditional plan, or failure
OR-SEARCH(*problem*.INITIAL-STATE, *problem*, [])

function OR-SEARCH(*state*, *problem*, *path*) **returns** a conditional plan, or failure
if *problem*.GOAL-TEST(*state*) **then return** the empty plan
if *state* is on *path* **then return** failure
for each *action* **in** *problem*.ACTIONS(*state*) **do**
 plan \leftarrow AND-SEARCH(RESULTS(*state*, *action*), *problem*, [*state* | *path*])
 if *plan* \neq failure **then return** [*action* | *plan*]
return failure

function AND-SEARCH(*states*, *problem*, *path*) **returns** a conditional plan, or failure
for each s_i **in** *states* **do**
 *plan*_{*i*} \leftarrow OR-SEARCH(s_i , *problem*, *path*)
 if *plan*_{*i*} = failure **then return** failure
return [**if** s_1 **then** *plan*₁ **else if** s_2 **then** *plan*₂ **else** ... **if** s_{n-1} **then** *plan*_{*n-1*} **else** *plan*_{*n*}]

Figure 4.11 An algorithm for searching AND-OR graphs generated by nondeterministic environments. It returns a conditional plan that reaches a goal state in all circumstances. (The notation $[x \mid l]$ refers to the list formed by adding object x to the front of list l .)