بسمه تعالی

هوش مصنوعی

جستجو در محیطهای پیچیده- ۳

نیمسال اوّل ۱۴۰۲-۱۴۰۱

دکتر مازیار پالهنگ

آزمایشگاه هوش مصنوعی

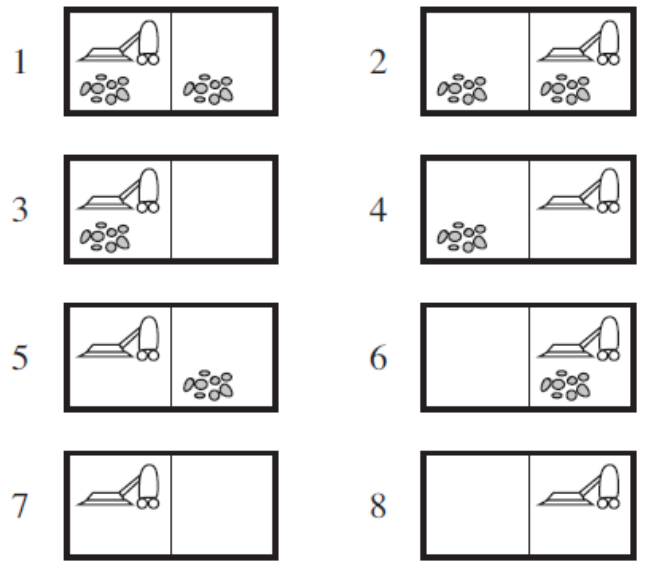
دانشکدهٔ مهندسی برق و کامپیوتر

دانشگاه صنعتی اصفهان

یادآوری

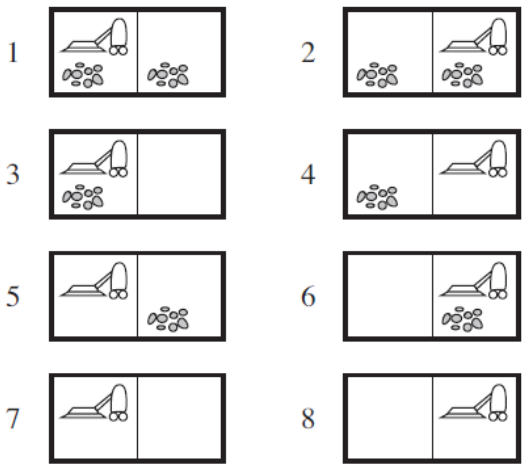
- الگوریتمهای جستجوی محلی
  - حالت فعلی را نگهدار – سعی کن آن را بهبود دهی
- جستجوی تپه نوردی
- تنوعها:
  - تپه نوردی تصادفی، تپه نوردی اولین انتخاب، تپه نوردی با بازشروع تصادفی
- سردشدن شبیه سازی شده
- جستجوی پرتو محلی، و تصادفی
- الگوریتم ژنتیک
- جستجوی محلی در فضای پیوسته
  - گرادیان، نیوتن- رافسن

جستجو با اعمال قطعی



- محیط قطعی – مشاهده پذیر:
  مسئلهٔ تک حالته
- عامل دقیقاً می داند که در چه حالتی خواهد بود.
- حل یک دنباله
- با شروع از ۵:
- [راست، مکش]

جستجو با اعمال غیرقطعی



- مثال ربات جاروی سرگردان:
- محیط مشاهده پذیر–غیرقطعی
- عمل مکش:
  - در خانه کثیف آن خانه را تمیز ولی گاهی خانهٔ همسایه را نیز تمیز می کند،
  - در خانهٔ تمیز گاهی آشغال می ریزد.
- تعمیم مدل انتقال در فصل ۳
- تابع Result() یک مجموعه از حالات قابل دستیابی ممکن باز می گرداند.
- بطور مثال: مکش در حالت ۱ به مجموعهٔ {۵و۷} می رود.

4

جستجو با اعمال غیرقطعی

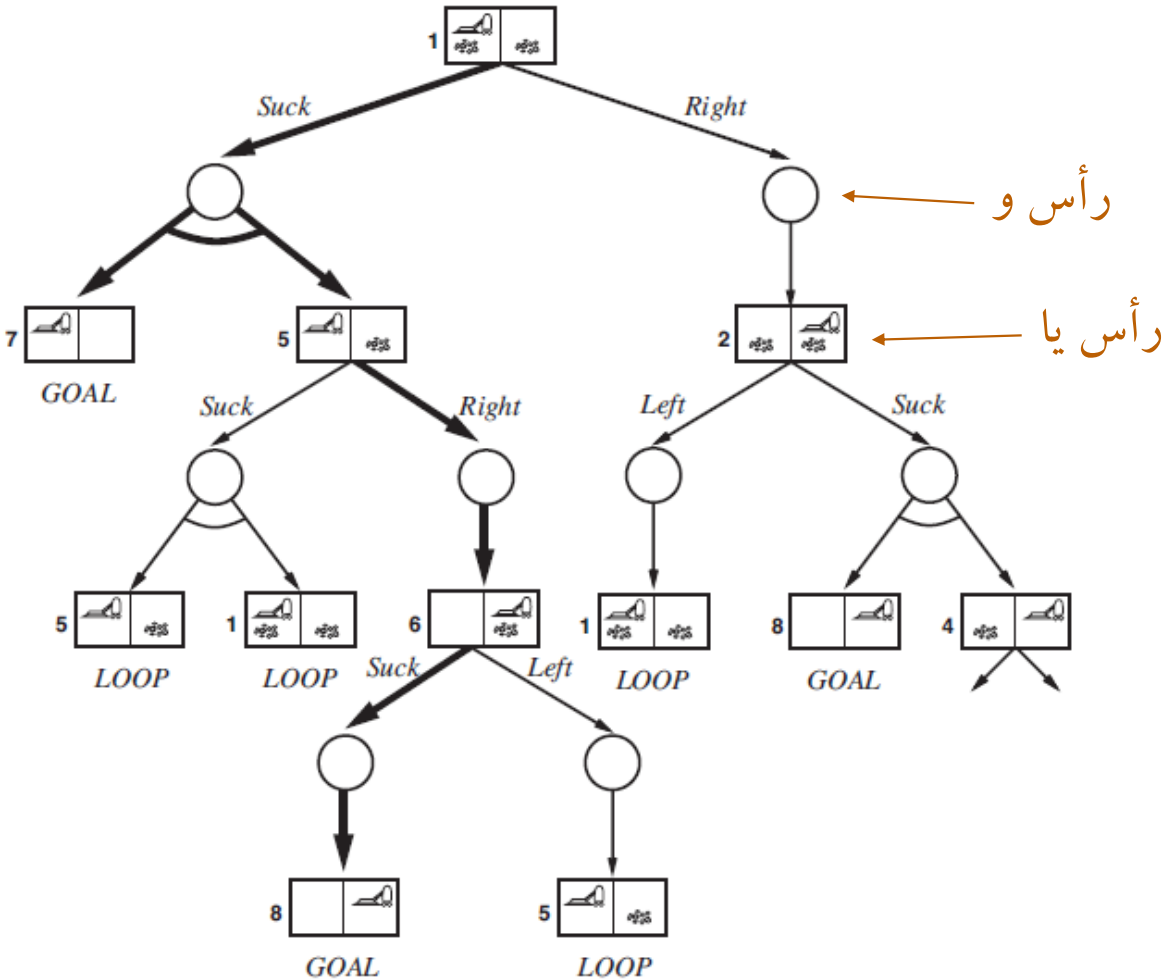
- اصلاح حل: بجای یک دنباله یک طرح شرطی (یا طرح اقتضائی یا راهبرد (استراتژی))
- بطور مثال با شروع از حالت ۱:

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

- حل مسائل همراه با عدم قطعیت شامل جملات اگر-آنگاه
- حل بصورت یک درخت بجای یک دنباله
- انتخاب عمل بر اساس اقتضاء بعد از عمل

درخت جستجوی AND–OR

- جستجو مجدداً با ایجاد یک درخت جستجو
- در حالت قطعی شاخه ها با اعمالی که عامل می توانست انجام دهد ایجاد می شد.
- به چنین رئوس ایجاد شده، رئوس-یا (OR nodes) می گوئیم.
- در حالت غیرقطعی باید همهٔ رئوسی که ممکن است نتایج انجام عمل باشند نیز ایجاد شود.
- این رئوس، رئوس- و (AND nodes) گفته می شوند.


1
Suck
Right
رأس و
7
GOAL
5
2
رأس یا
Suck
Right
Left
Suck
5
LOOP
1
LOOP
6
Suck
Left
1
LOOP
8
GOAL
4
8
GOAL
5
LOOP

7

جستجو با اعمال غیر قطعی

- یک حل برای یک جستجو و-یا (AND-OR) یک زیر درخت از درخت جستجوی کامل است بطوری که:
- در هر برگ یک رأس هدف وجود دارد،
- در هر یک از رئوس یا در آن فقط یک عمل مشخص شده، و
- در هر یک از رئوس و در آن تمامی پیشامدها در نظر گرفته شده اند.

Figure 4.11

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

Figure 4.11

```
function AND-OR-SEARCH(problem) returns a conditional plan, or failure
  return OR-SEARCH(problem, problem.INITIAL, [])

function OR-SEARCH(problem, state, path) returns a conditional plan, or failure
  if problem.IS-GOAL(state) then return the empty plan
  if IS-CYCLE(path) then return failure
  for each action in problem.ACTIONS(state) do
      plan ← AND-SEARCH(problem, RESULTS(state, action), [state] + path])
      if plan ≠ failure then return [action] + plan]
  return failure
```

چون عمق نخست است هر شاخه یا به جواب می رسد یا شکست می خورد

Figure 4.11

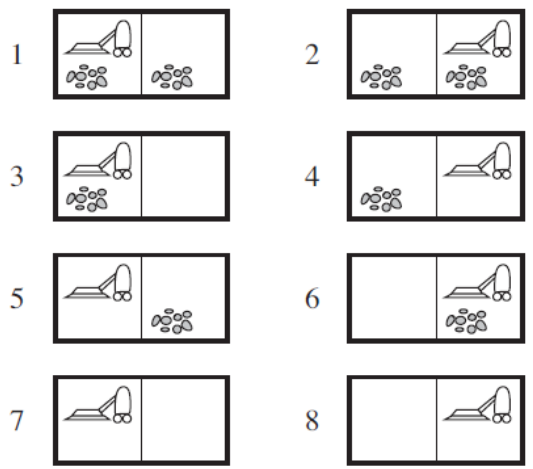
```
function AND-OR-SEARCH(problem) returns a conditional plan, or failure
  return OR-SEARCH(problem, problem.INITIAL, [])

function OR-SEARCH(problem, state, path) returns a conditional plan, or failure
  if problem.IS-GOAL(state) then return the empty plan
  if IS-CYCLE(path) then return failure
  for each action in problem.ACTIONS(state) do
      plan ← AND-SEARCH(problem, RESULTS(state, action), [state] + path])
      if plan ≠ failure then return [action] + plan]
  return failure

function AND-SEARCH(problem, states, path) returns a conditional plan, or failure
  for each s_i in states do
      plan_i ← OR-SEARCH(problem, s_i, 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]
```

An algorithm for searching AND–OR graphs generated by nondeterministic environments. A solution is a conditional plan that considers every nondeterministic outcome and makes a plan for each one.

تلاش و تلاش



- دنیای جاروی لغزنده را در نظر بگیرید، همانند دنیای جاروی معمولی ولی
- عمل راست یا چپ گاهی انجام می شود و گاهی در همان خانه ای که بودیم می مانیم.
- مثلاً اگر در حالت ۱، به راست برویم به حالت ۱ یا ۲ ممکن است برویم.
- در این وضعیت، حل بدون حالت تکراری نخواهیم داشت.

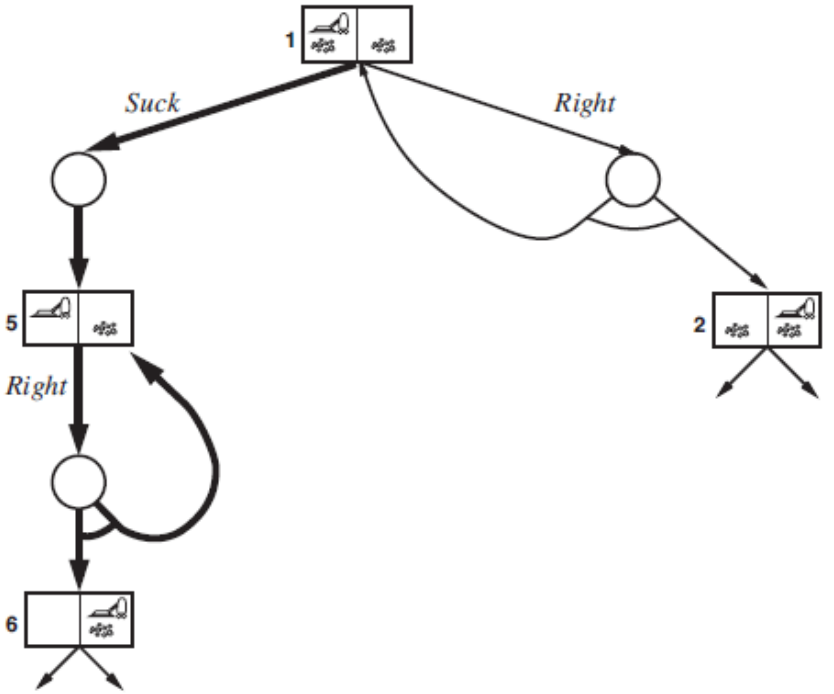
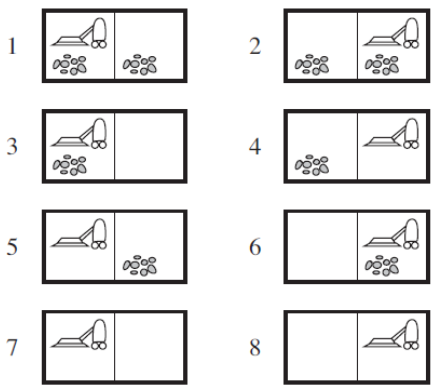


Figure 4.12 Part of the search graph for the slippery vacuum world, where we have shown (some) cycles explicitly. All solutions for this problem are cyclic plans because there is no way to move reliably.

- حل چرخشی وجود دارد.
- حداقل هر برگ در طرح یک حالت هدف بوده و برگ قابل رسیدن باشد.
- نمایش با اضافه کردن برچسب:

$$[Suck, L_1 : \ Right, \textbf{if } State = 5 \textbf{ then } L_1 \textbf{ else } Suck]$$



- یا نمایش با حلقهٔ while:

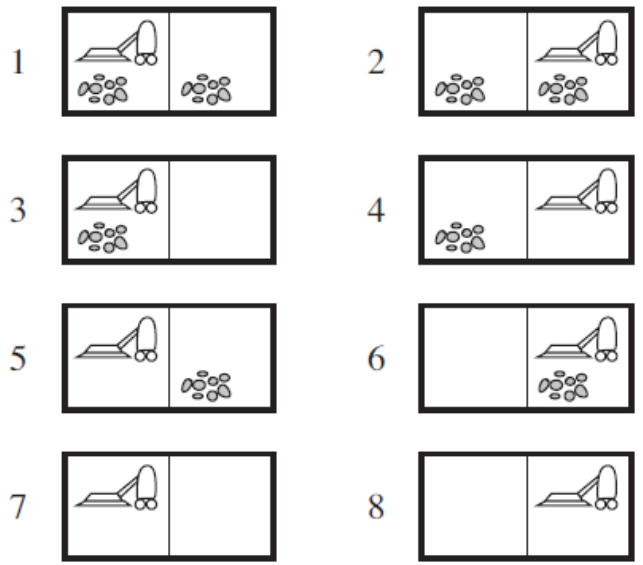
$$[Suck, \textbf{while } State = 5 \textbf{ do } Right, Suck]$$

جستجو در محیط نیمه مشاهده پذیر

- عامل دقیق نمی داند در چه حالتی است.
- حتی اگر اعمال قطعی باشند، انجام عمل منجر به یک مجموعه حالات خواهد شد.
- در این حالت، بهتر است مسئله را در فضای باور بجای فضای فیزیکی حل کنیم.
- حالت باور شامل مجموعه ای از حالات فیزیکی است که عامل تصور می کند در آن قرار دارد.

جستجو بدون مشاهده

- بدترین حالت، عامل حسگر نداشته باشد.
- عامل دانش محیطی که در آن هست را در اختیار دارد ولی اینکه در چه مکانی قرار دارد یا توزیع آشغال را نمی داند.
- اعمال قطعی
- حالت اولیه {۱و۲و۳و۴و۵و۶و۷و۸}
- [راست] به {۲و۴و۶و۸}
- [راست، مکش] به {۴و۸}
- نهایتاً [راست، مکش، چپ، مکش]



- جستجو در فضای باور
- حل در صورت وجود یک دنباله است.
- چون حسگری هم وجود ندارد مسئله اقتضائی نیست و می دانیم پس از حرکت درکی هم نخواهیم داشت.
- حتی اگر محیط غیرقطعی باشد.
- در این حالت در فضای باور محیط به نوعی مشاهده پذیر است چون عامل می داند به چه باوری می رود.
- می توان اجزائی که برای تعریف یک مسئله وجود دارد را برای فضای باور تعریف نمود.

- چون ۸ حالت داریم.
- تعداد حالات باور $2^8=256$ است.
- مجموعه قوهٔ یک مجموعهٔ ۸ عضوی
- همه آنها در این مثال دسترسی نمی شوند.
- می توان از همان جستجوهای کلاسیک استفاده کرد،
- با تعریف اجزاء مسئله برای فضای باور

- اجزاء مسئلهٔ اوّلیه P را با اندیس P نمایش می دهیم، مثل $Actions_P$، $Result_P$
- حالات: اگر P شامل N حالت باشد، مسئلهٔ فضای باور دارای 2^N باور است.
- حالت اوّلیه: یکی از باورها، معمولاً آنکه شامل همهٔ حالات P است.

- اعمال: ممکن است: $b = \{s_1, s_2\}$ $\text{ACTIONS}_P(s_1) \neq \text{ACTIONS}_P(s_2)$
- اگر انجام عملی مجاز روی حالتی دیگر مضر نباشد:

$$\text{ACTIONS}(b) = \bigcup_{s \in b} \text{ACTIONS}_P(s) .$$

- اگر مضر باشد شاید بهتر باشد اشتراک اعمال مجاز روی هر حالت را در نظر بگیریم.

- مدل انتقال:
  - اعمال قطعی: $b' = \text{RESULT}(b,a) = \{s' : s' = \text{RESULT}_P(s,a) \text{ and } s \in b\}$.
  - اعمال غیرقطعی:

$$\begin{aligned} b' = \text{RESULT}(b,a) &= \{s' : s' \in \text{RESULTS}_P(s,a) \text{ and } s \in b\} \\ &= \bigcup_{s \in b} \text{RESULTS}_P(s,a) \,, \end{aligned}$$

- تست هدف: هر حالت در باور یک حالت هدف است.
- هزینهٔ عمل:
  - فرض می کنیم هزینهٔ اعمال یک عمل روی حالات مختلف در یک باور یکسان باشد، در این صورت همان منظور خواهد شد.
  - حالتی که تفاوت هزینه وجود دارد فعلاً مورد بحث ما نیست.

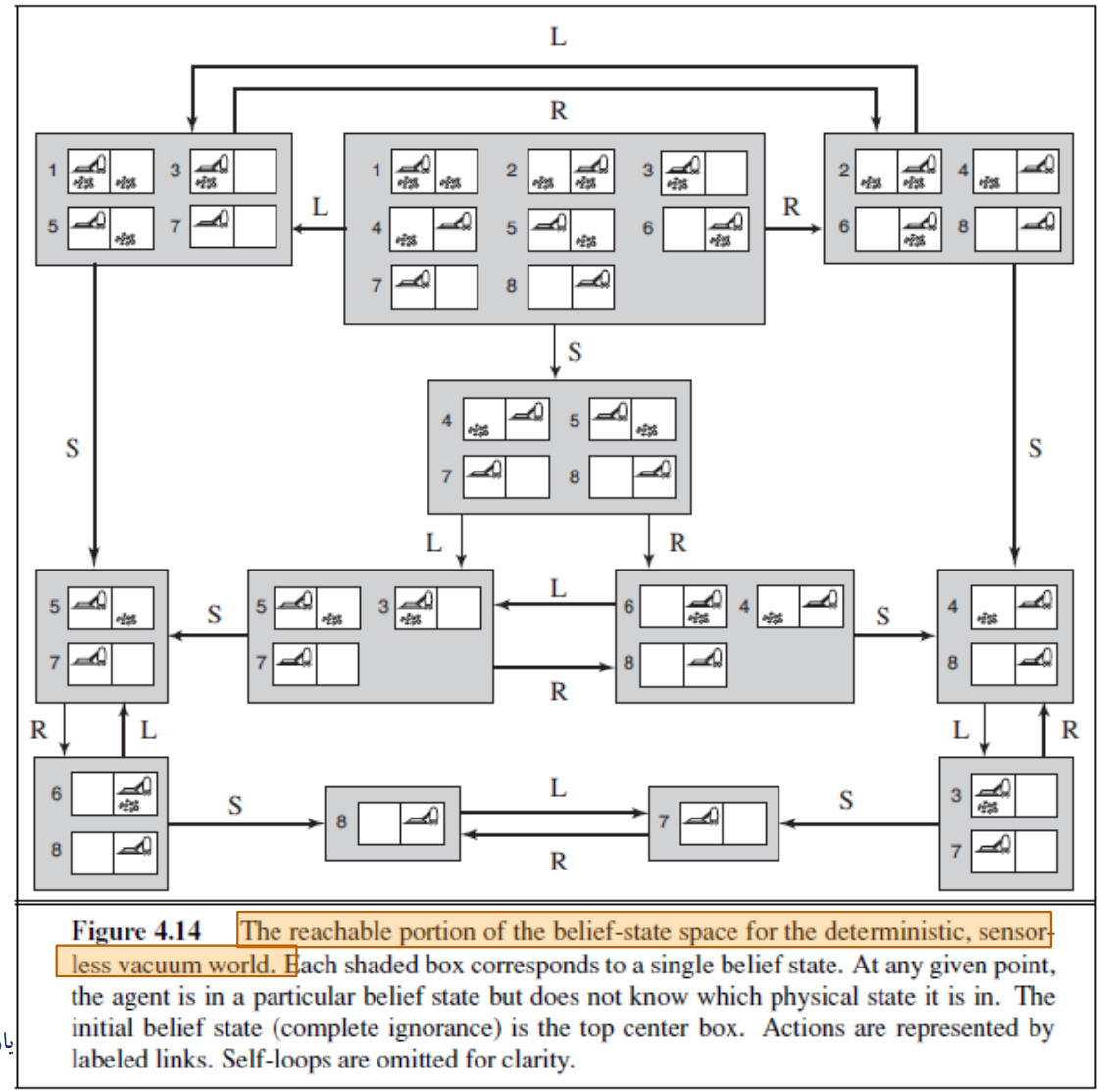
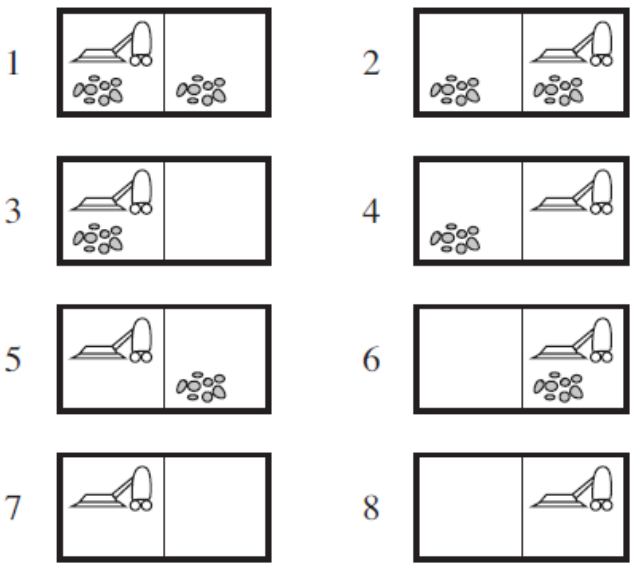


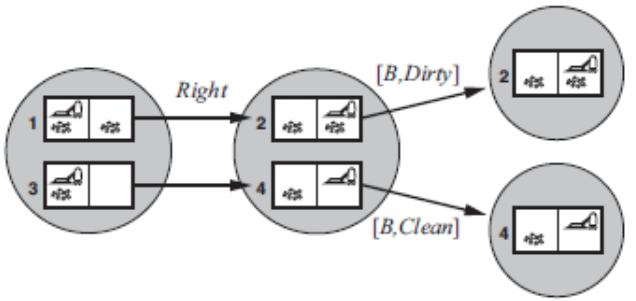
Figure 4.14 The reachable portion of the belief-state space for the deterministic, sensorless vacuum world. Each shaded box corresponds to a single belief state. At any given point, the agent is in a particular belief state but does not know which physical state it is in. The initial belief state (complete ignorance) is the top center box. Actions are represented by labeled links. Self-loops are omitted for clarity.

جستجو با مقداری درک



- فرض درک فقط شامل مکان و وجود آشغال در خانهٔ فعلی
- درک [چپ، تمیز] در حالت {۵و۷}
- حل [راست اگر آشغال مکش]
- مسئلهٔ شرطی (اقتضائی)

قطعی



غیر قطعی

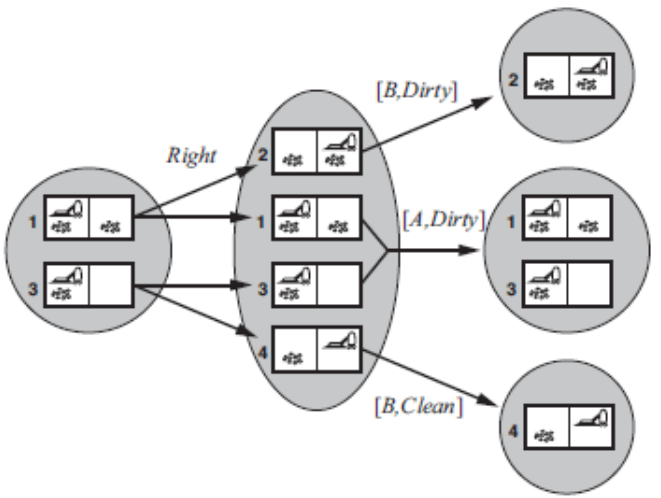


Figure 4.15 Two example of transitions in local-sensing vacuum worlds. (a) In the deterministic world, *Right* is applied in the initial belief state, resulting in a new belief state with two possible physical states; for those states, the possible percepts are $[B, Dirty]$ and $[B, Clean]$, leading to two belief states, each of which is a singleton. (b) In the slippery world, *Right* is applied in the initial belief state, giving a new belief state with four physical states; for those states, the possible percepts are $[A, Dirty]$, $[B, Dirty]$, and $[B, Clean]$, leading to three belief states as shown.

الهنگ

24

حل مسائل نیمه مشاهده پذیر

- استفاده از الگوریتم جستجوی AND–OR

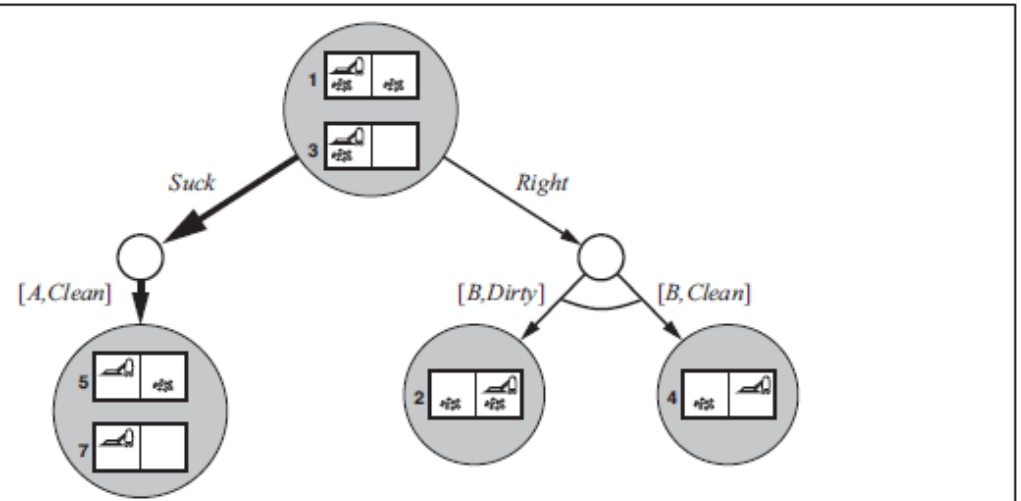


Figure 4.16 The first level of the AND–OR search tree for a problem in the local-sensing vacuum world; *Suck* is the first step of the solution.

- حل یک طرح شرطی

$$[Suck, Right, \textbf{if } Bstate = \{6\} \textbf{ then } Suck \textbf{ else } [\,]\,].$$

خلاصه

- جستجو با اعمال غیرقطعی
  - حل یک طرح شرطی، استفاده از درخت AND–OR با استفاده از فضای حالت
- جستجو برای عامل بدون حسگر
  - جستجو در فضای باور همانند حالت مشاهده پذیر با تعمیم تعریف اجزاء مسئله
- جستجو در محیط نیمه مشاهده پذیر
  - حل یک طرح شرطی، استفاده از درخت AND–OR با استفاده از فضای باور


امام علی (ع):
وَالْفُرْصَةُ تَمُرُّ مَرَّ السَّحابِ، فَانْتَهِزُوا فُرَصَ الْخَيْرِ
فرصت به مانند ابر گذرا می گذرد، پس فرصت های نیک را غنیمت دانید
پایان

27

- دقت نمائید که پاورپوینت ابزاری جهت کمک به یک ارائهٔ شفاهی می باشد و **به هیچ وجه یک جزوهٔ درسی نیست** و شما را از خواندن مراجع درس بی نیاز نمی کند.
- لذا حتماً مراجع اصلی درس را مطالعه نمائید.