بسمه تعالی

هوش مصنوعی

جستجو در محیطهای پیچیده- ۴

نیمسال اوّل ۱۴۰۳-۱۴۰۲

دکتر مازیار پالهنگ

آزمایشگاه هوش مصنوعی

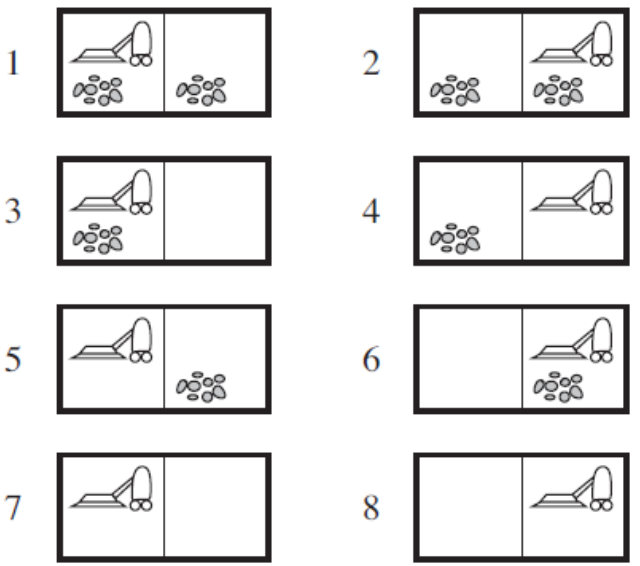
دانشکدهٔ مهندسی برق و کامپیوتر

دانشگاه صنعتی اصفهان

یادآوری

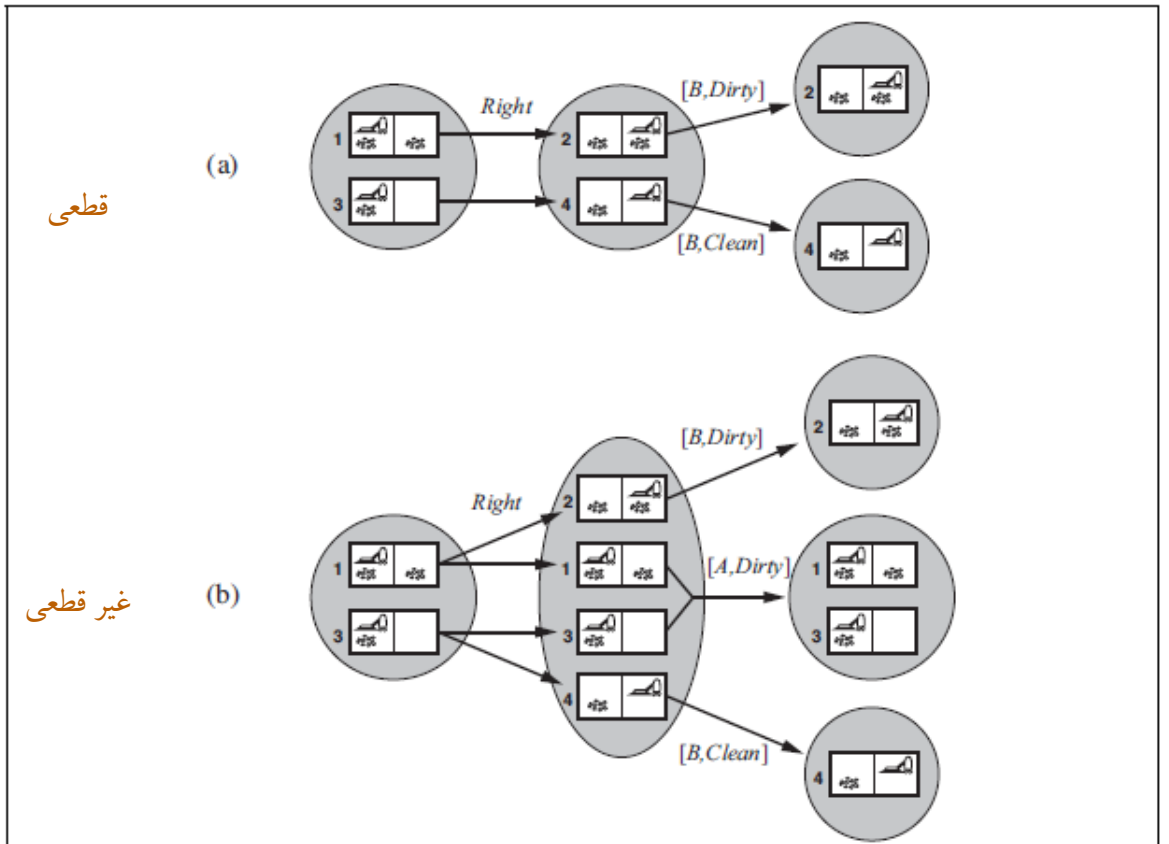
- الگوریتمهای جستجوی محلی
  - حالت فعلی را نگهدار – سعی کن آن را بهبود دهی
- جستجوی تپه نوردی
- تنوعها:
  - تپه نوردی تصادفی، تپه نوردی اولین انتخاب، تپه نوردی با بازشروع تصادفی
- سردشدن شبیه سازی شده
- جستجوی پرتو محلی، و تصادفی، الگوریتم ژنتیک
- جستجوی محلی در فضای پیوسته
  - گرادیان، نیوتن- رافسن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

4

حل مسائل نیمه مشاهده پذیر

- استفاده از الگوریتم جستجوی AND–OR

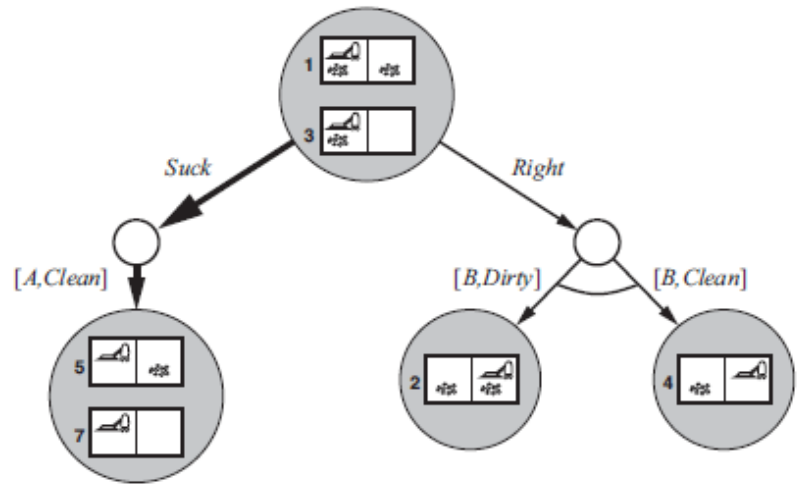


Figure 4.16 The first level of the AND–OR search tree for a problem in the local-sensing vacuum world; *Suck* is the first step of the solution.

- حل یک طرح شرطی

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

عاملی برای محیط نیمه مشاهده پذیر

- دو تفاوت با عاملی که در محیط مشاهده پذیر عمل می کند.
- اول: حل یک طرح شرطی است بجای دنباله
- دوم: عامل باید یک باور در خود ذخیره نماید.
- فرآیند پیش بینی، مشاهده، بروز رسانی

$$b' = \text{UPDATE}(\text{PREDICT}(b, a), o)$$

- شناخته شده تحت نام نظارت کردن، صافی کردن (filtering)، یا تخمین حالت (state estimation)

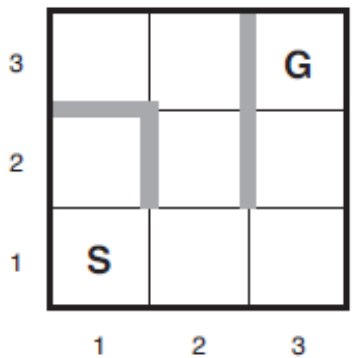
جستجوی برخط و محیطهای ناشناخته

- جستجوی برون خط
  - محاسبهٔ حل کامل قبل از اجرای حل
- جستجوی برخط
  - انجام عمل، مشاهده، محاسبهٔ عمل بعد
- مناسب برای
  - محیطهای پویا و نیمه پویا (جلوگیری از محاسبات با دید طولانی)
  - در محیطهای غیرقطعی (تمرکز تلاش محاسباتی با توجه به موقعیت بوجود آمده نه تمرکز روی وضعیتهائی که بندرت ممکن است پیش آیند)
  - محیطهای ناشناخته (هنگامی که حالات و نتایج اعمال ناشناخته هستند)
- محیطهای ناشناخته – مسائل اکتشافی

مسائل جستجوی برخط

- فعلا با فرض محیط مشاهده پذیر و قطعی
- دانش عامل:
- Actions(s): لیستی از اعمالی که عامل در حالت s می تواند انجام دهد باز می گرداند.
- تابع هزینهٔ c(s،a،s') – تا عامل نداند s' نتیجهٔ عمل a در s است قابل استفاده نیست.
- Goal_Test(s)

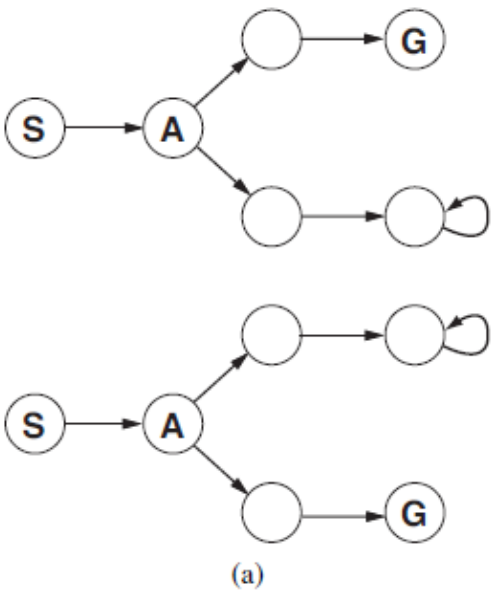
- عامل نمی تواند مقدار Result(s،a) را مشخص کند مگردر s عمل a را انجام دهد.



- عامل ممکن است به یک مکاشفهٔ قابل پذیرش h(s) دسترسی داشته باشد.

- نوعاً عامل علاقمند به طی کردن کوتاهترین مسیر تا هدف است.
- یا ممکن است بخواهد تمامی محیط را اکتشاف کند.
- معمول است که هزینهٔ مسیر انجام شده با هزینهٔ مسیر در حالتی که عامل تمام محیط را می شناخت مقایسه شود.
- یعنی کوتاهترین مسیر (یا کوتاهترین اکتشاف کامل)
- به آن نسبت رقابتی (competitive ratio) گفته می شود.
- علاقمند که کمترین مقدار باشد.

- گاهی این مقدار ممکن است بی نهایت شود.
- مشکل ساز هنگامی که اعمال برگشت پذیر نباشند.
- امکان رسیدن به بن بست.

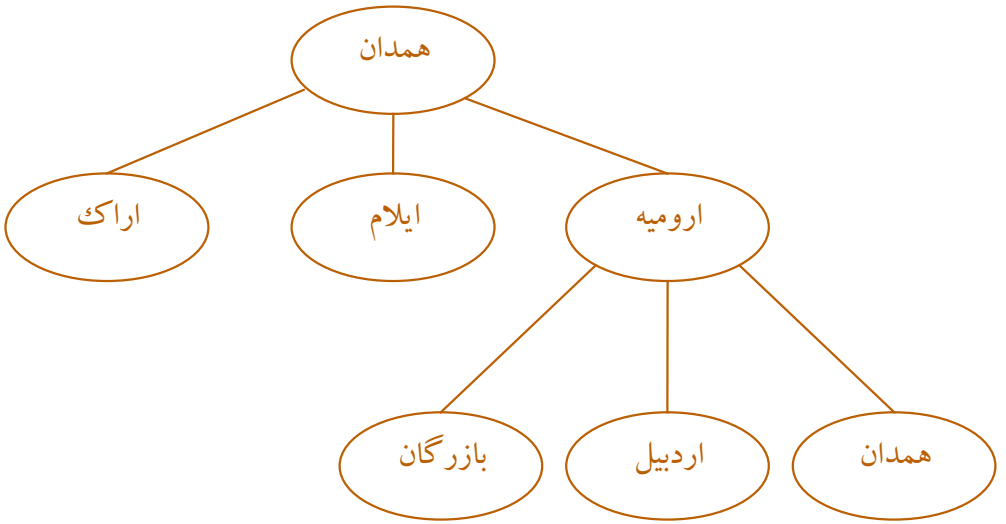


(a)

- فرض می کنیم محیط بطور امن قابل اکتشاف (safely explorable) است.
- یعنی برخی از حالات هدف از هر حالت قابل دسترسی قابل دستیابی می باشند.

عاملهای جستجوی برخط

- پس از عمل، عامل درک می کند در چه حالتی قرار دارد.
- بسط نقشه محیط
- استفاده از نقشه برای عمل بعدی
- متفاوت با روشهای جستجوی برون خط همانند A*
- در A* می توان به قسمتی دیگر از فضای حالت رفت،
- در جستجوی برخط نمی توان چنین کرد.
- جستجوی برخط در دنیای واقعی است ولی برون خط بر روی مدل دنیا


همدان
اراک
ایلام
ارومیه
بازرگان
اردبیل
همدان

- فقط جستجو در حوالی حالتی که در آن قرار دارد.
- همانند عمق نخست (بجز هنگام عقبگرد)

Figure 4.21

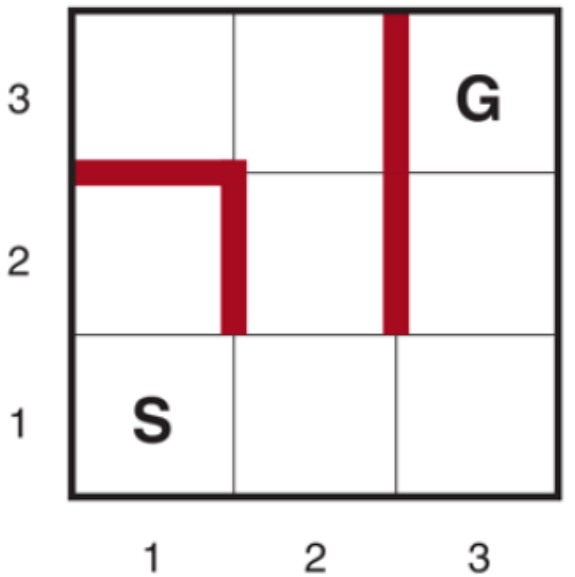
```
function ONLINE-DFS-AGENT(problem, s') returns an action
        s, a, the previous state and action, initially null
  persistent: result, a table mapping (s, a) to s', initially empty
              untried, a table mapping s to a list of untried actions
              unbacktracked, a table mapping s to a list of states never backtracked to

  if problem.IS-GOAL(s') then return stop
  if s' is a new state (not in untried) then untried[s'] ← problem.ACTIONS(s')
  if s is not null then
      result[s, a] ← s'
      add s to the front of unbacktracked[s']
  if untried[s'] is empty then
      if unbacktracked[s'] is empty then return stop
      else a ← an action b such that result[s', b] = POP(unbacktracked[s'])
  else a ← POP(untried[s'])
  s ← s'
  return a
```

چون همهٔ اعمالی که می توان در s' انجام داد تلاش شده اند

An online search agent that uses depth-first exploration. The agent can safely explore only in state spaces in which every action can be "undone" by some other action.

- تلاش کنید رد اجرای الگوریتم را روی محیط زیر بیابید:



خلاصه

- جستجو در محیط نیمه مشاهده پذیر
  - حل یک طرح شرطی، استفاده از درخت AND-OR با استفاده از فضای باور
- تفاوت جستجوی برون خط و برخط
- مواقع مفید برای استفاده از جستجوی برخط
- مسائل جستجوی برخط با فرض محیط مشاهده پذیر قطعی
  - عدم توانائی مشخص کردن Result(s،a) از قبل
  - نسبت رقابتی
- عامل جستجوی برخط عمق نخست Online_DFS_Agent


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

- 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.
- حضور فعال در کلاس دارای امتیاز است.