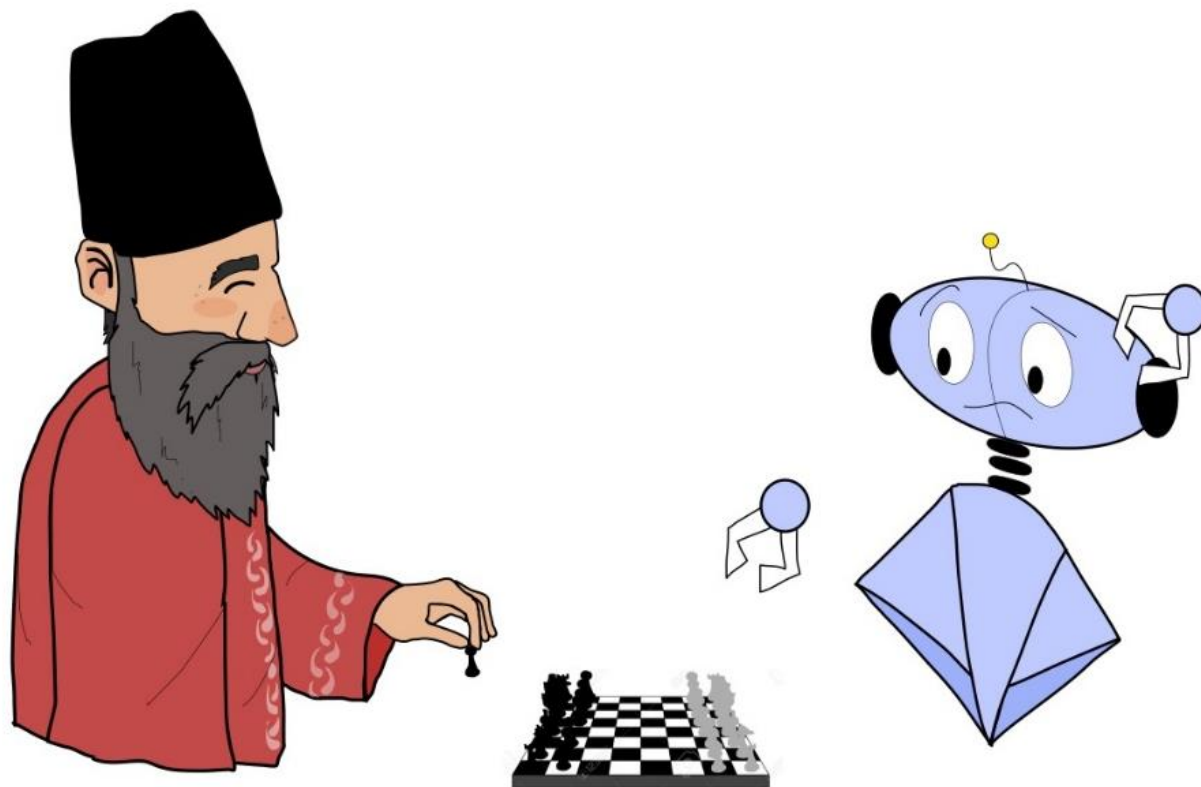


عامل‌های هوشمند

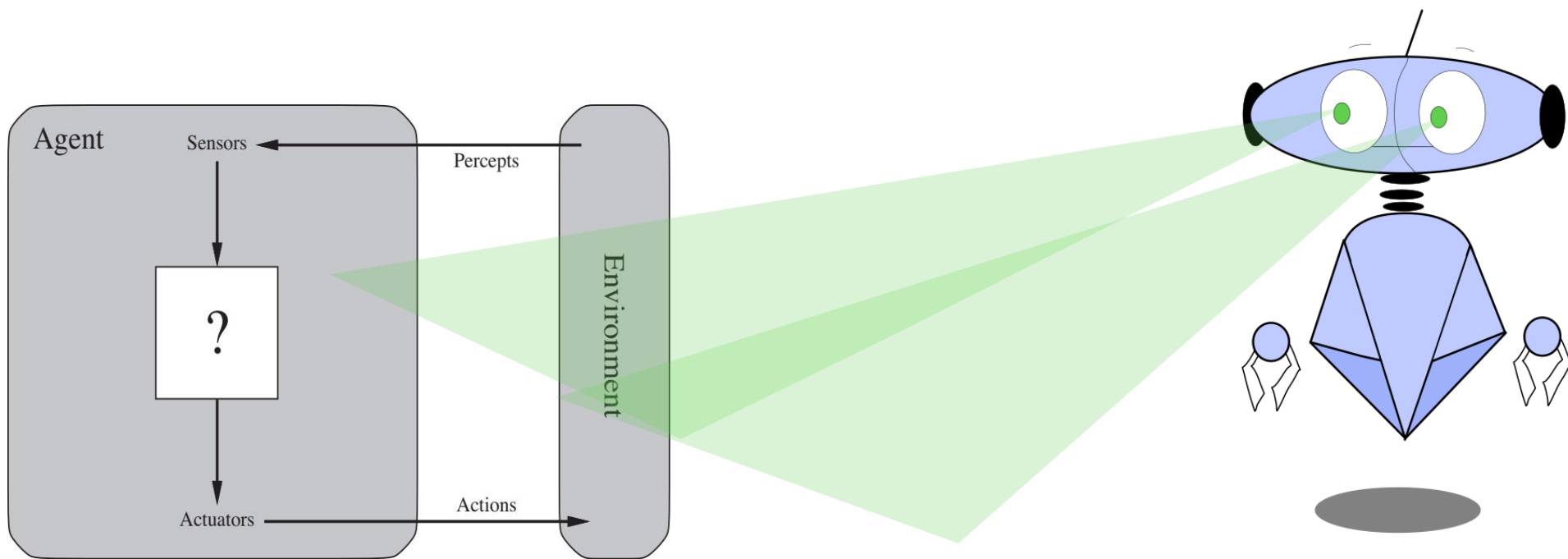


رئوس مطالب

- عامل‌ها و محیط‌ها
- رفتار خوب: مفهوم عقلانیت (rationality)
- ماهیت محیط‌ها
- ساختار عامل‌ها

عامل‌ها

- یک عامل (agent) هر چیزیست که بتواند اطلاعاتی را از طریق حسگرها (sensors) از محیط ادراک (perceive) کند و از طریق عملگرها (actuators) بر روی محیط عمل کند



نمونه‌هایی از عامل‌ها

● عامل انسانی

● **حسگرها:** چشم‌ها، گوش‌ها، و سایر اندام‌های حسی

● **عملگرها:** دست‌ها، پاها، تارهای صوتی و ...

● عامل رباتی

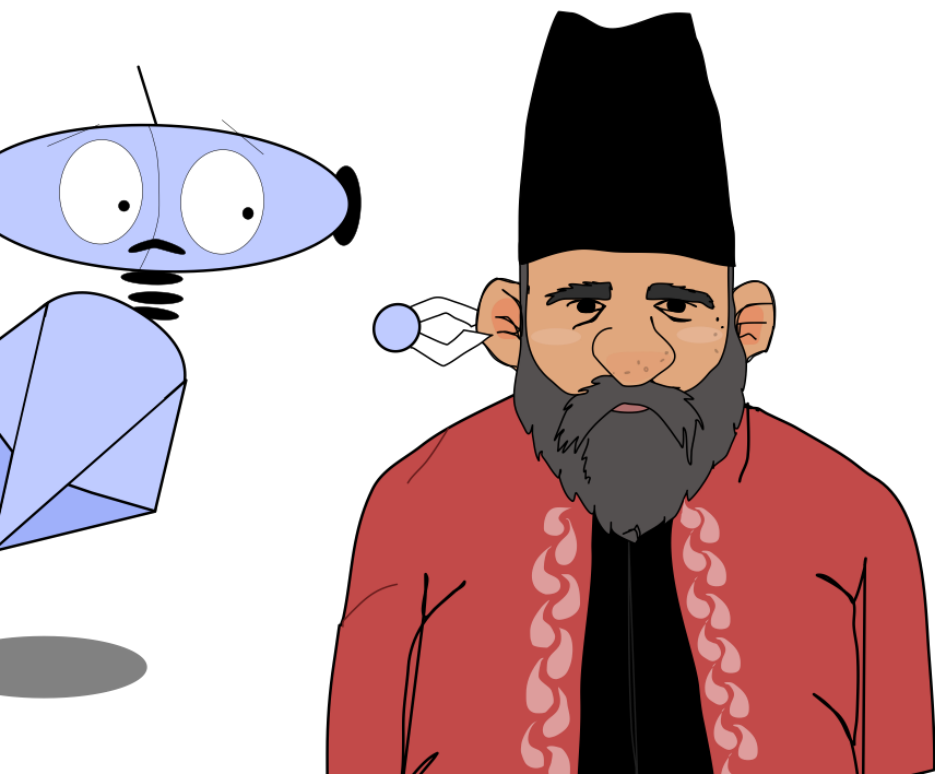
● **حسگرها:** دوربین‌ها، بردیاب مادون قرمز

● **عملگرها:** موتورهای مختلف

● عامل‌های نرم‌افزاری

● **حسگرها:** فشردن صفحه کلید، محتویات فایل، بسته‌های دریافتی شبکه

● **عملگرها:** صفحه‌ی نمایشگر، فایل‌ها، بسته‌های ارسال شده شبکه



عامل‌ها و محیط‌ها

- رفتار یک عامل توسط تابع عامل توصیف می‌شود که هر دنباله‌ی ادراکی معین را به یک عمل نگاشت می‌کند

$$f: P^* \rightarrow A$$

- می‌توان این تابع را با یک جدول توصیف کرد
- برای بیشتر عامل‌ها، این جدول بسیار بزرگ خواهد بود
- باید دقت داشت که این جدول یک توصیف خارجی از عامل است
- در توصیف داخلی، تابع عامل برای یک عامل مصنوعی توسط یک برنامه عامل پیاده‌سازی می‌شود

تابع عامل یک توصیف انتزاعی ریاضی است. برنامه عامل یک پیاده‌سازی مشخص است که در یک سیستم فیزیکی اجرا می‌شود.

دنیای جارو برقی

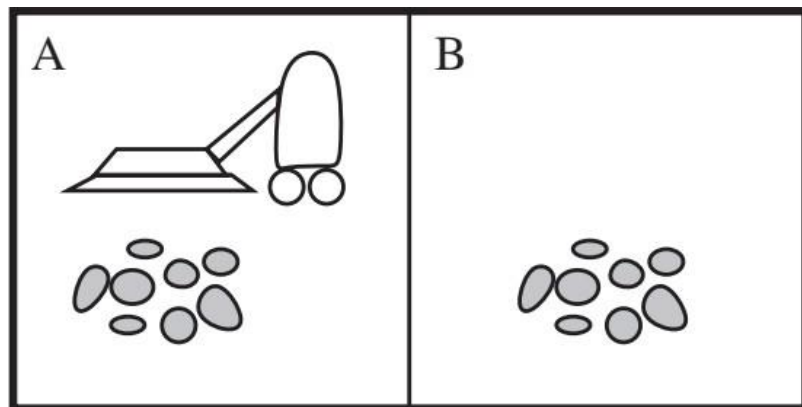
• ادراکات:

- موقعیت و وضعیت کثیف/تمیز بودن محل آن

برای مثال: [A, dirty]

• اقدامات:

- چپ، راست، مکش، بدون عملیات



یک عامل جارو برقی

Percept sequence	Action
$[A, \textit{Clean}]$	\textit{Right}
$[A, \textit{Dirty}]$	\textit{Suck}
$[B, \textit{Clean}]$	\textit{Left}
$[B, \textit{Dirty}]$	\textit{Suck}
$[A, \textit{Clean}], [A, \textit{Clean}]$	\textit{Right}
$[A, \textit{Clean}], [A, \textit{Dirty}]$	\textit{Suck}
\vdots	\vdots
$[A, \textit{Clean}], [A, \textit{Clean}], [A, \textit{Clean}]$	\textit{Right}
$[A, \textit{Clean}], [A, \textit{Clean}], [A, \textit{Dirty}]$	\textit{Suck}
\vdots	\vdots

اگر مربع فعلی کثیف است،
در نتیجه مکش انجام شود؛
در غیر این صورت، به مربع دیگر برود

عوامل‌های عقلانی

Samsung's new fridge will ping your phone if you leave the door open



Internet of
@internetofsh

why doesn't it just close the door
itself if it's so smart

- بر اساس تاریخچه‌ی ادراک و اقداماتی که می‌تواند انجام دهد، "کار درست را انجام دهد"

- چه چیزی کار درست برای انجام دادن است؟

- هنگامی که یک عامل در محیطی قرار می‌گیرد، بر اساس ادراکی که دریافت می‌کند،

دنباله‌ای از اعمال را ایجاد می‌کند. این توالی اعمال باعث می‌شود که محیط از یک دنباله

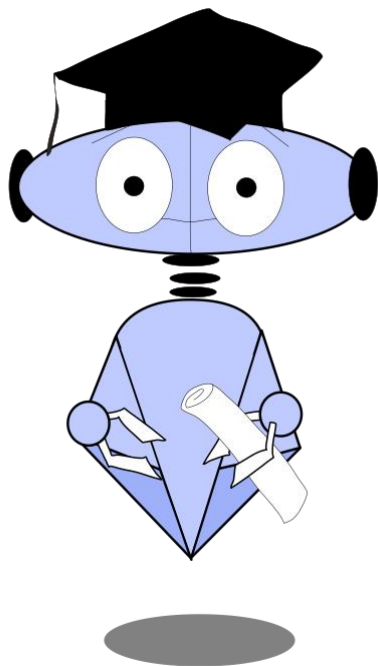
از حالت‌ها عبور کند. **اگر این دنباله مطلوب است، پس عامل به خوبی عمل کرده است**

- این مفهوم مطلوب بودن توسط یک معیار عملکرد (performance measure) توصیف می‌شود

معیار عملکرد

- دنباله حالت‌های محیط را بررسی می‌کند
- به حالت‌های محیط وابسته است نه به حالت‌های عامل
- بدیهی ست که یک معیار عملکرد ثابت برای همه‌ی وظایف و عامل‌ها وجود ندارد
- نمونه‌های معیار عملکرد در عامل جاروبرقی:
 - مقدار کثیفی تمیز شده X
 - یک امتیاز برای هر مربع تمیز در هر مرحله زمانی و جریمه برای مصرف برق و صدای تولید شده ✓

عقلانیت



- عقلانی بودن در هر زمان به چهار موضوع بستگی دارد:

- معیار عملکرد

- دانش قبلی عامل از ساختار محیط

- اقداماتی که عامل می‌تواند انجام دهد

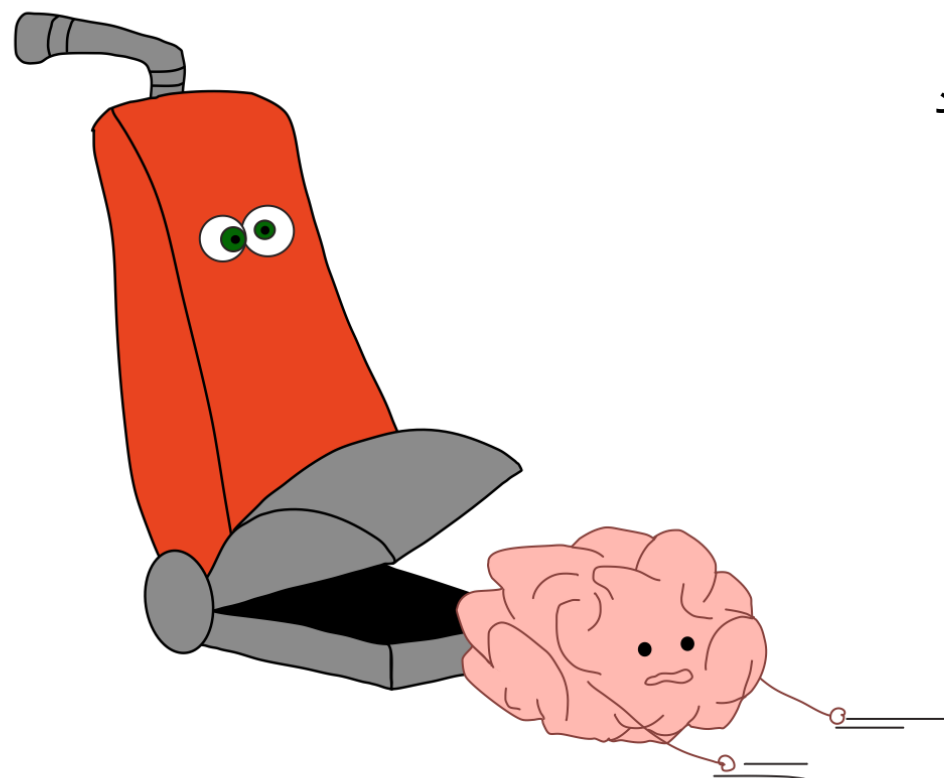
- دنباله ادراک عامل تا به این لحظه

- عامل عقلانی

- برای هر دنباله ادراکی ممکن، یک عامل عقلانی باید عملی را انتخاب کند که **انتظار می‌رود** با توجه به شواهد ارائه شده

توسط **دنباله ادراکی** و **هر دانشی که عامل در داخل خود دارد**، **معیار عملکرد** آن را به حداکثر برساند

عقلانیت (مثال جاروبرقی)



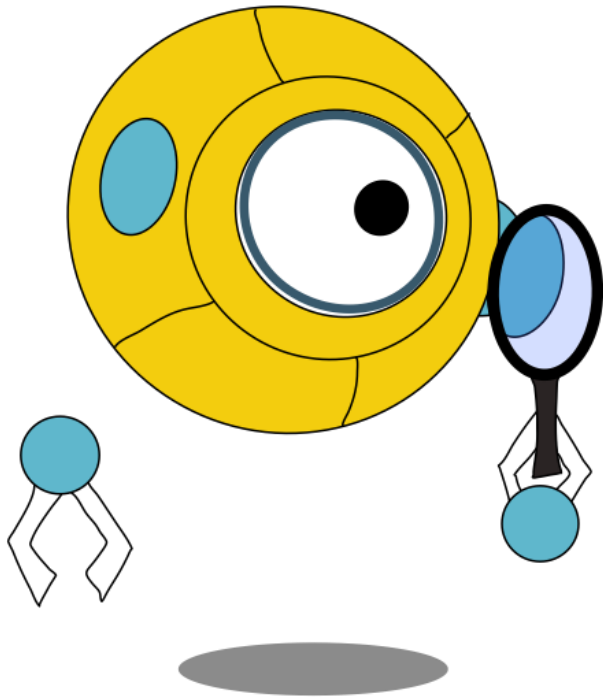
آیا این عقلانی است؟

- اگر کثیف است، مکش کند، در غیر این صورت به مربع دیگر برود
- بستگی دارد به:
 - معیار عملکرد
 - به عنوان مثال، جریمه مصرف انرژی؟
 - محیط (دانش قبلی عامل از ساختار محیط)
 - به عنوان مثال، کثیفی جدید می تواند ظاهر شود؟
 - عملگرها (اقداماتی که عامل می تواند انجام دهد)
 - به عنوان مثال، اقدام بدون عملیات؟
 - حسگرها (دنباله ادراک عامل تا به این لحظه)
 - به عنوان مثال، فقط کثیفی را در محل آن احساس کنید؟

عقلانیت در مقابل دانای کل

- **عقلانیت از دانای کل (Omniscience) بودن متمایز است**
- عاملی که دانای کل باشد:
- یک عامل دانای کل **نتیجه واقعی** اعمال خود را می داند و می تواند بر اساس آن عمل کند. اما دانایی مطلق و دانای کل بودن در **واقعیت غیرممکن** است
- **عقلانیت معادل کمال نیست:**
- عقلانیت **عملکرد مورد** انتظار را به حداکثر می رساند، در حالی که کمال **عملکرد واقعی** را به حداکثر می رساند
- عقلانیت به دانایی مطلق نیاز ندارد:
- زیرا انتخاب منطقی تنها به دنباله ادراک تا به امروز بستگی دارد

نیازمندی‌های یک عامل منطقی



جمع‌آوری یا اکتشاف اطلاعات:

- انجام اقداماتی به منظور اصلاح ادراکات آینده برای به دست آوردن اطلاعات مفید
- مثال: اکتشاف باید توسط عامل جاروبرقی در یک محیط ناشناخته اولیه انجام شود

• یادگیری:

- پیکربندی اولیه عامل می‌تواند منعکس کننده برخی دانش قبلی از محیط باشد
- با کسب تجربه عامل، این پیکربندی ممکن است اصلاح و تقویت شود

• خودمختاری (در ادامه)

خودمختاری

یک عامل **خودمختار (autonomous)** است اگر رفتارش با تجربه خودش تعیین شود (با توانایی **یادگیری** و **انطباق**)

- تنها به دانش قبلی طراح متکی نیست
- می‌آموزد که دانش قبلی جزئی یا نادرست را جبران کند
- مزیت: محیط‌های تغییرکننده
- با عمل تصادفی یا بر اساس دانش طراح شروع می‌کند و سپس از طریق تجربه یاد می‌گیرد
- عامل منطقی باید خودمختار باشد
- مثال: عامل جارو برقی
- اگر کثیف است، مکش کند، در غیر این صورت به مربع دیگر برود
- آیا این عامل یک عامل خودمختار است؟
- یادگیری پیش بینی وقوع کثیفی در مربع



محیط وظیفه-Task environment (PEAS)

- معیار عملکرد (Performance measure)

- محیط (Environment)

- عملگرها (Actuators)

- حسگرها (Sensors)

در طراحی یک عامل، اولین قدم همیشه باید این باشد که محیط وظیفه (task environment) را تا حد امکان به طور کامل مشخص کنید

نمونه‌های محیط وظیفه



- عامل: راننده تاکسی خودران

- معیار عملکرد

- سفر ایمن، سریع، قانونی، راحت، به حداکثر رساندن سود، ...

- محیط

- جاده‌ها، سایر فضاهای ترافیکی، عابران پیاده، مشتریان و ...

- عملگرها

- فرمان، گاز، ترمز، سیگنال، بوق، نمایشگر

- حسگرها

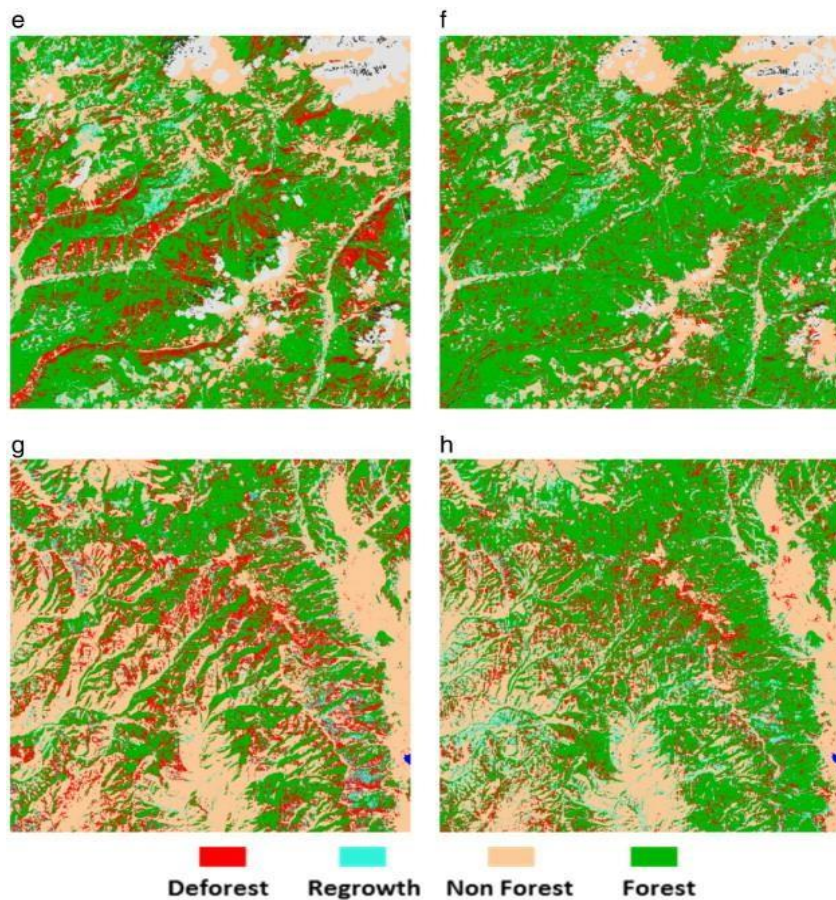
- دوربین، سونار، سرعت سنج، GPS، کیلومتر شمار، شتاب سنج، سنسور موتور، صفحه کلید

نمونه‌های محیط وظیفه



- عامل: سیستم تشخیص پزشکی هوشمند
 - معیار عملکرد
 - بیمار سالم، هزینه‌ها را به حداقل برساند، ...
 - محیط
 - بیمار، بیمارستان، پرسنل، ...
 - عملگرها
 - صفحه نمایش (سوالات، آزمایشات، تشخیص، درمان، ارجاعات)
 - حسگرها
 - صفحه کلید، شبکه و ... (ورود علائم، یافته‌ها، پاسخ‌های بیمار)

نمونه‌های محیط وظیفه



- عامل: سیستم تحلیل تصاویر ماهواره‌ای

- معیار عملکرد

- دسته بندی صحیح تصاویر

- محیط

- لینک دانلودی از ماهواره در مدار

- عملگرها

- نمایش تصویر دسته بندی شده

- حسگرها

- آرایه پیکسل رنگی

نمونه‌های محیط وظیفه



- عامل: ربات قطعه چین
 - معیار عملکرد
 - درصد قطعات در سبدهای صحیح محیط
 - تسمه نقاله با قطعات، سبد
 - عملگرها
 - بازو مفصلی و دست
 - حسگرها
 - دوربین، سنسورهای زاویه مشترک

نمونه‌های محیط وظیفه



- عامل: مدرس تعاملی زبان انگلیسی
 - معیار عملکرد
 - نمره دانش آموز در آزمون را به حداکثر برساند
 - محیط
 - مجموعه ای از دانش آموزان، آزمون هدف و ...
 - عملگرها
 - صفحه نمایش، بلندگو و ... (تمرینات، پیشنهادات، اصلاحات)
 - حسگرها
 - صفحه کلید، میکروفون، دوربین و ...

نمونه‌های محیط وظیفه

- عامل: Pacman

- معیار عملکرد

- امتیاز، تعداد جان‌های باقیمانده

- محیط

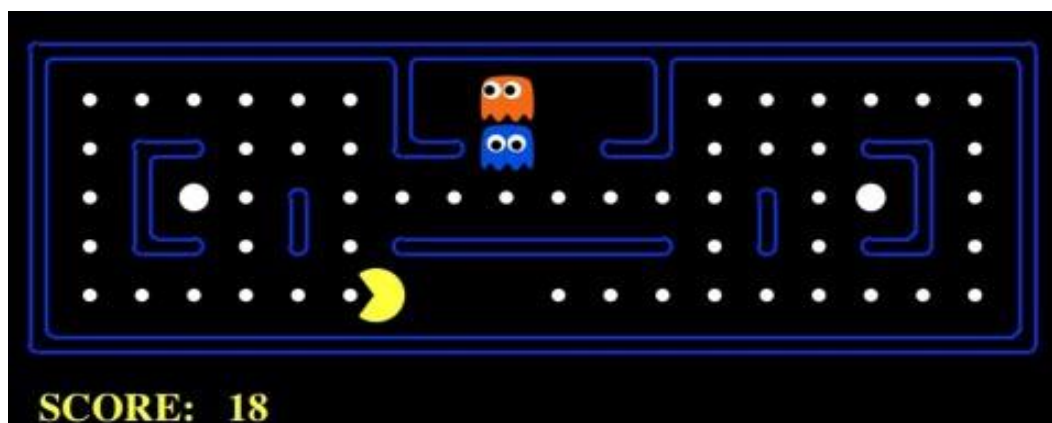
- هزارتو نقاط سفید، چهار روح، قرص‌های قدرت، میوه‌هایی که گاهی ظاهر می‌شوند و ...

- عملگرها

- کلیدهای جهت دار

- حسگرها

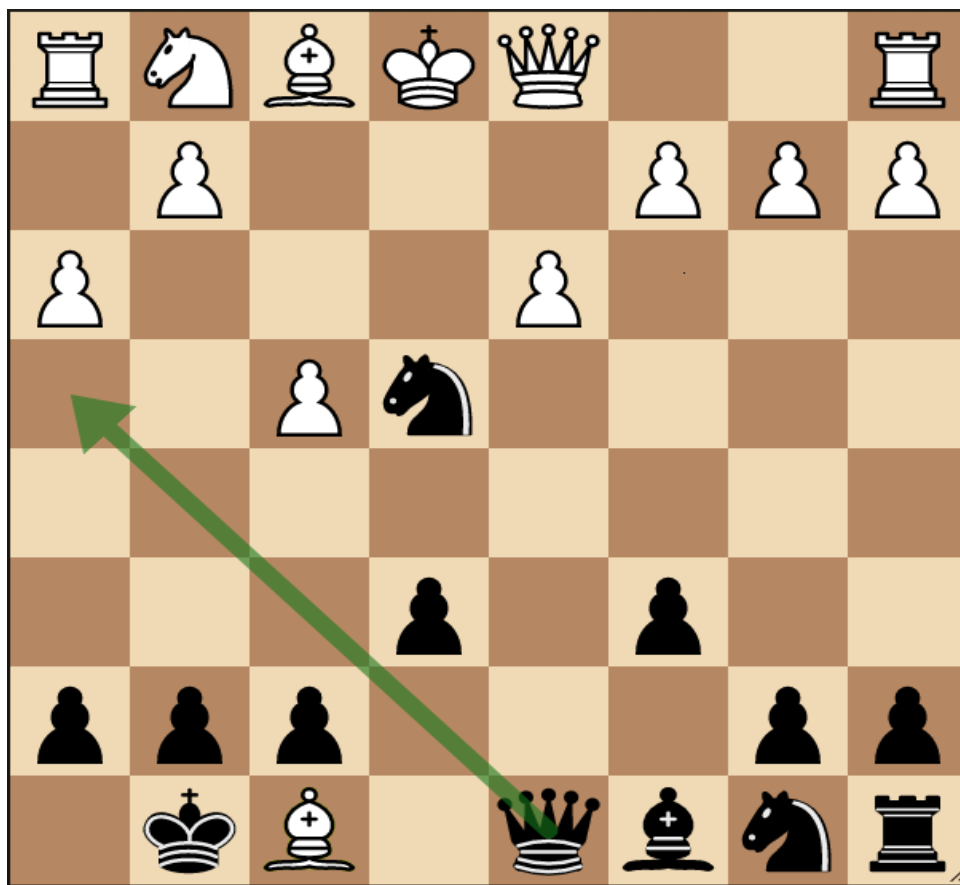
- صفحه نمایش بازی



انواع محیط

- کاملاً قابل مشاهده fully observable (در مقابل تا حدی قابل مشاهده partially observable):
 - حسگرها در هر لحظه به ما دسترسی از وضعیت کامل محیط را می دهند
 - حسگرها تمام جنبه‌های مربوط به انتخاب عمل را تشخیص می دهند
 - راحت (نیازی به نگهداری حالت داخلی ندارد)
 - سنسورهای نویزی و نادقیق یا بخش‌های برداشت نشده حالت که ناشی از حسگرها باشد محیط را تا حدی قابل مشاهده partially observable می‌کند
- تک عامل (در مقابل چند عامل):
 - جدول کلمات متقاطع (crossword) یک بازی تک عاملی است (شطرنج یک بازی چند عاملی است)
 - آیا B یک عامل است یا فقط یک شی در محیط؟
 - B زمانی عامل است که رفتار آن را بتوان در قالب **بیشینه کردن معیار عملکرد** توصیف کرد که مقدار آن به رفتار A بستگی داشته باشد
 - چند عاملی: رقابتی، مبتنی بر همکاری
 - رفتار و ارتباط تصادفی نیز در مواردی می تواند منطقی باشد

انواع محیط - مثال



- بهترین حرکت را برای مشکی پیدا کنید

انواع محیط

- **قطعی Deterministic** (در مقابل تصادفی stochastic):

- وضعیت یا حالت بعدی را بتوان به طور کامل با **وضعیت فعلی** و **عمل انجام شده** تعیین کرد

- اگر محیط به استثنای اقدامات سایر عوامل قطعی باشد، آنگاه محیط **استراتژیک** است (این عدم قطعیت را نادیده میگیریم)

- محیط تا حدی قابل مشاهده می‌تواند تصادفی در نظر گرفته شود

- اگر محیط به طور کامل قابل مشاهده نباشد یا **قطعی نباشد**، **غیرقطعی uncertain** است

- **گسسته Discrete** (در مقابل پیوسته continuous):

- تعدادی محدود و مشخص از **حالت‌های** واضح تعریف شده، **ادراکات** و **اعمال**، **مراحل زمانی (time step)**

- شطرنج دارای تعداد متناهی حالت‌های گسسته و مجموعه‌ای گسسته از ادراکات و اعمال است در حالی که رانندگی تاکسی دارای

حالات و عمل‌های پیوسته است

انواع محیط

- **قسمت‌بندی شده یا اپیزودیک Episodic** (در مقابل ترتیبی sequential):
 - تجربه عامل به «قسمت‌های» اتمی تقسیم می‌شود که انتخاب عمل در هر قسمت فقط به خود آن قسمت بستگی دارد
 - به عنوان مثال، تشخیص قطعات معیوب در خط مونتاژ (مستقل)
 - در محیط های متوالی، اقدامات کوتاه مدت می تواند پیامدهای بلندمدتی داشته باشد
 - محیط اپیزودیک می تواند بسیار ساده تر باشد
- **ایستا Static** (در مقابل پویا dynamic):
 - در صورتی که عامل در حال فکر کردن باشد محیط بدون تغییر بماند محیط ایستا است
 - **نیمه پویا:** اگر خود محیط با گذشت زمان تغییر نکند اما امتیاز عملکرد عامل تغییر کند
 - استاتیک (پازل های متقاطع)، پویا (راننده تاکسی)، نیمه پویا (شطرنج ساعت‌دار)

انواع محیط

- شناخته شده Known (در مقابل ناشناخته unknown):
- نتایج یا احتمال نتایج برای همه‌ی اعمال ارائه شده است
- این کاملاً یک ویژگی محیطی نیست
- مربوط به وضعیت دانش عامل یا طراح در مورد "قوانین فیزیک" محیط است
- اگر محیط ناشناخته باشد، عامل باید یاد بگیرد که چگونه در آن کار کند تا بتواند تصمیمات خوبی بگیرد
- یک محیط ناشناخته می‌تواند کاملاً قابل مشاهده باشد
- جهان واقعی
- تا حدی قابل مشاهده، چند عاملی، تصادفی، متوالی، پویا، پیوسته، (و ناشناخته)
- سخت‌ترین نوع محیط
- نوع محیط تا حد زیادی طراحی عامل را تعیین می‌کند

انواع محیط

Task Environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with a clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete
Backgammon	Fully	Multi	Stochastic	Sequential	Static	Discrete
Taxi driving	Partially	Multi	Stochastic	Sequential	Dynamic	Continuous
Medical diagnosis	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Image analysis	Fully	Single	Deterministic	Episodic	Semi	Continuous
Part-picking robot	Partially	Single	Stochastic	Episodic	Dynamic	Continuous
Refinery controller	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Interactive English tutor	Partially	Multi	Stochastic	Sequential	Dynamic	Discrete

ساختار عامل‌ها

عامل = معماری + برنامه

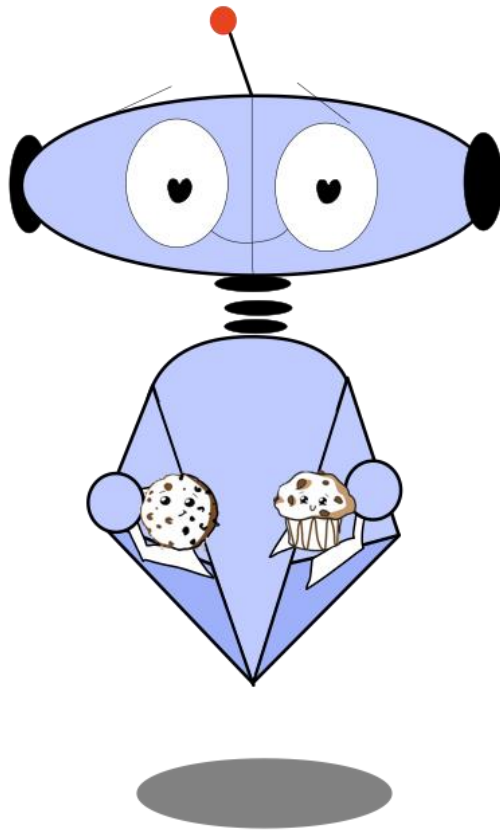
- برنامه عامل

- وظیفه هوش مصنوعی طراحی یک **برنامه عامل (agent program)** است که **عملکرد عامل** را پیاده سازی کند
- برنامه عامل فقط برداشت فعلی را به عنوان ورودی می گیرد
- عامل باید در صورت نیاز کل دنباله ادراک را به خاطر بسپارد
- **حالت/وضعیت داخلی**

- معماری

- ما فرض می کنیم که برنامه‌ی عامل بر روی نوعی دستگاه محاسباتی با حسگرها و عملگرهای فیزیکی اجرا می‌شود

انواع برنامه عامل



- عامل جدول محور (Table driven agent)
- انواع اصلی برنامه عامل به ترتیب افزایش عمومیت
- عامل‌های واکنشی ساده (Simple reflex agents)
- عامل‌های واکنشی مبتنی بر مدل (Model-based reflex agents)
- عامل‌های مبتنی بر هدف (Goal-based agents)
- عامل‌های مبتنی بر سودمندی (Utility-based agents)
- عامل‌های یادگیری محور (Learning-based agents)

عامل جدول محور

Function TABLE-DRIVEN_AGENT(*percept*) **returns** an action

persistent: *percepts*, a sequence initially empty
table, a table of actions, indexed by percept sequence, initially fully specified

append *percept* to the end of *percepts*
action ← LOOKUP(*percepts*, *table*)
return *action*

Percept sequence	Action
[A, Clean]	Right
[A, Dirty]	Suck
[B, Clean]	Left
[B, Dirty]	Suck
[A, Clean],[A, Clean]	Right
[A, Clean],[A, Dirty]	Suck
...	...
[A, Clean],[A, Clean], [A, Clean]	Right
...	...

عامل جدول محور

- فواید

- پیاده‌سازی آسان

- اشکالات

- فضا

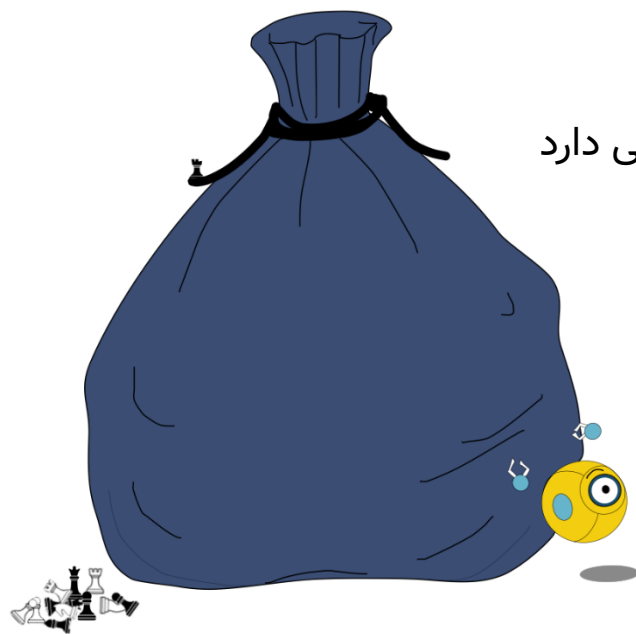
$$\sum_{t=1}^T |P|^t \quad (\text{طول عمر } T: \text{مجموعه ای از ادراکات ممکن } P)$$

- برای شطرنج حداقل 10^{150} ورودی وجود دارد در حالی که کمتر از 10^{80} اتم در جهان وجود خارجی دارد

- طراح وقت کافی برای ایجاد جدول نخواهد داشت

- هیچ عاملی هرگز نمی‌تواند ورودی‌های جدول مناسب را از تجربه خود بیاموزد

- چگونه ورودی‌های جدول را پر کنیم؟



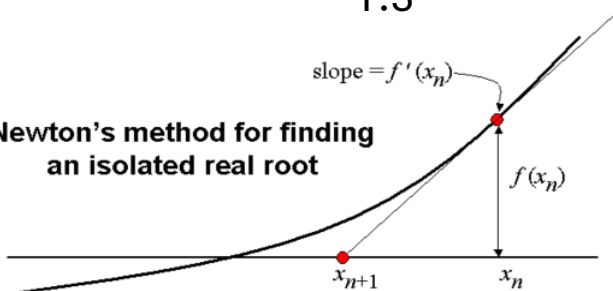
برنامه عامل

- نگاشت (از ادراک به عمل) لزوما استفاده از جدول نیست
- هوش مصنوعی قصد دارد که برنامه‌هایی را پیدا کند که رفتار عقلانی را به جای جدول بی‌اندازه بزرگ، با برنامه‌هایی کوچک تولید کند

جدول‌های بزرگ کاربردی برای به دست آوردن ریشه‌های مربع قبل از دهه 1970

1.0	1.00000
1.1	1.04898
1.2	1.09565
1.3	1.14056

Newton's method for finding
an isolated real root



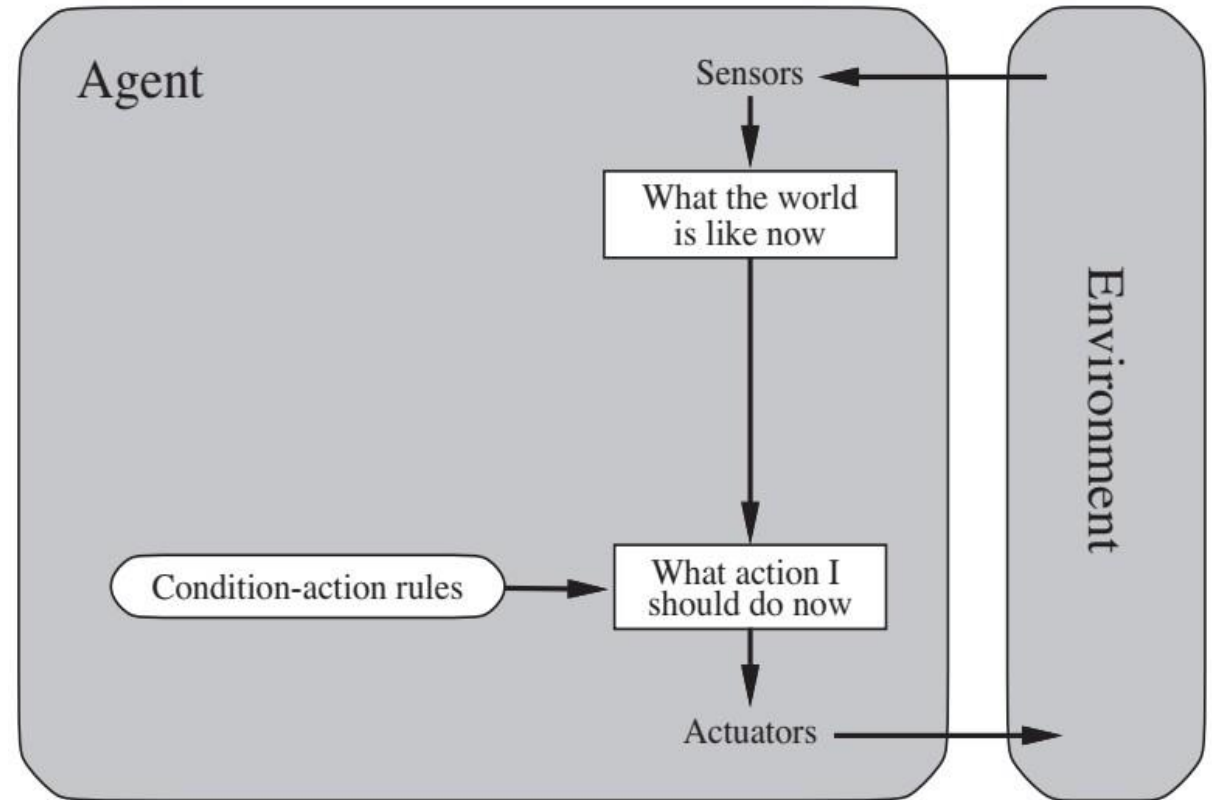
$$x_{n+1} = x_n - \frac{f'(x_n)}{f(x_n)}$$

```
function SQRT( double X )
{
    double r = 1.0;
    while ( fabs( r * r - x ) > 10-8 )
        r = r - ( r * r - x ) / 2r;
    return r ;
}
```


عامل واکنشی ساده

```
function SIMPLE-REFLEX-AGENT(percept)
  returns an action
  persistent: rules, a set of condition-action
  rules

  state ← INTERPRET-
  INPUT(percept)
  rule ← RULE-MATCH(state, rules)
  action ← rule.ACTION
  return action
```



عامل واکنشی ساده

- اقدامات را بر اساس ادراک فعلی انتخاب کنید و بقیه تاریخچه ادراکی را نادیده بگیرید

• If **car-in-front-is-braking** then **initiate-braking**

• رفلکس پلک زدن

• مثال عامل جاروبرقی

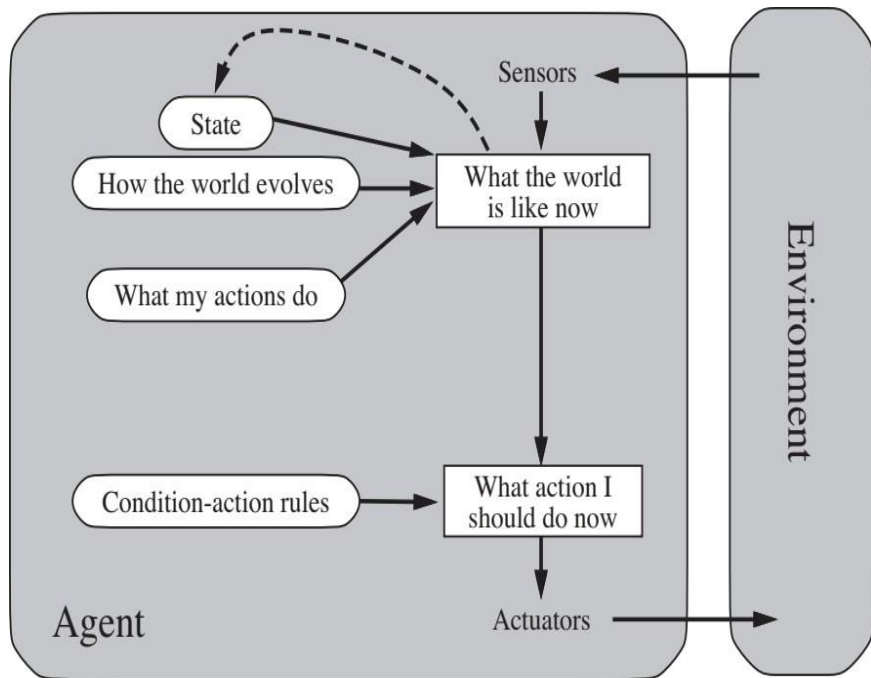
```
function REFLEX-VACUUM-AGENT ([location, status]) return an action
  if status == Dirty then return Suck
  else if location == A then return Right
  else if location == B then return Left
```

- تعداد احتمالات را از 4^T به تنها 4 کاهش می دهد

عامل واکنشی ساده

- هوش ساده، اما بسیار محدود
- فقط در صورتی کار می‌کند که بتوان تصمیم درست را بر اساس ادراک فعلی گرفت (مشاهده‌پذیری کامل)
- حلقه (loop) های نامحدود در محیط‌های قابل مشاهده جزئی
- فرار از حلقه‌های نامحدود در صورتی امکان پذیر است که عامل بتواند اقدامات خود را تصادفی انتخاب کند

عامل واکنشی مبتنی بر مدل



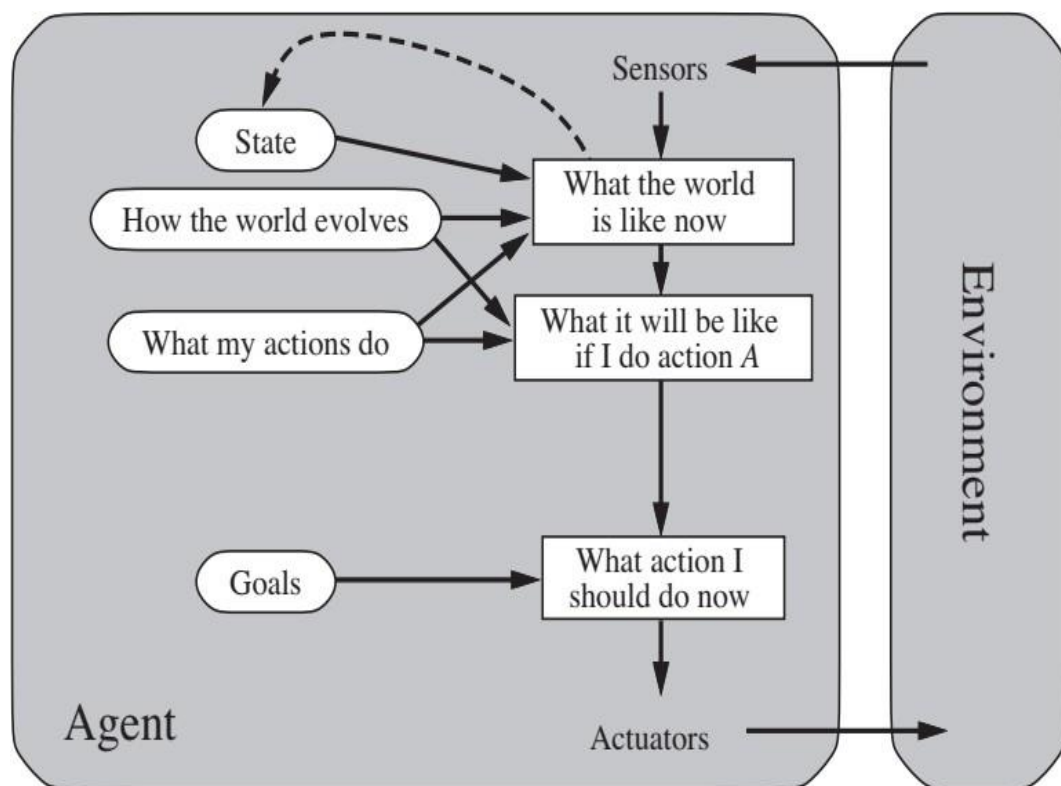
- مشاهده پذیری جزئی
- وضعیت داخلی (بر اساس تاریخچه ادراکی)
- برخی از جنبه‌های مشاهده نشده از وضعیت فعلی را منعکس می‌کند
- به روز رسانی اطلاعات وضعیت داخلی به دو نوع دانش نیاز دارد
- اطلاعات در مورد چگونگی تکامل جهان (مستقل از عامل)
- اطلاعاتی در مورد اینکه چگونه اقدامات خود عامل بر جهان تأثیر می‌گذارد
- فقط بهترین حدس را برای وضعیت فعلی یک محیط مشاهده‌پذیر جزئی تعیین می‌کند

عامل واکنشی مبتنی بر محیط

```
function MODEL-BASED-REFLEX-AGENT(percept) returns an action
  persistent: state, the agent's current conception of the world state
               model, a description of how the next state depends on current state and action
               rules, a set of condition-action rules
               action, the most recent action, initially none

  state ← UPDATE-STATE(state, action, percept, model)
  rule ← RULE-MATCH(state, rules)
  action ← rule.ACTION
  return action
```

عامل مبتنی بر هدف



- آگاهی از وضعیت فعلی همیشه برای تصمیم گیری برای انجام دادن یک عمل کافی نیست
- موقعیت هایی که مطلوب هستند (هدف) باید مشخص شوند
- معمولاً نیاز به جستجو و برنامه ریزی دارد
- برای یافتن توالی عمل برای دستیابی به هدف

State = goal then “happy” if not “unhappy” so continue search

عامل واکنشی در مقابل عامل مبتنی بر هدف

- در نظر گرفتن آینده
- عامل‌های مبتنی بر هدف ممکن است کارایی کمتری داشته باشند اما انعطاف پذیرتر هستند
- دانش به صراحت نشان داده می شود و به راحتی قابل تغییر است (جدول قوائد در مقابل هدف)
- مثال: رفتن به مقصد جدید
- عامل مبتنی بر هدف: تعیین آن مقصد به عنوان هدف
- عامل واکنشی: قوانین عامل برای زمان چرخش و زمان مستقیم رفتن باید بازنویسی شود

عامل مبتنی بر سودمندی - 1

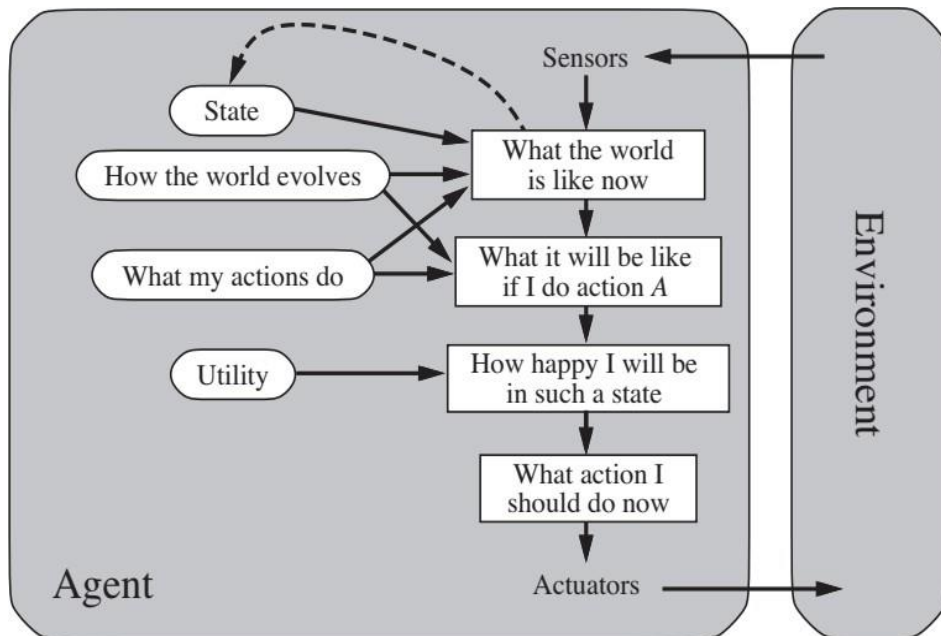
- اهداف (goals) به تنهایی برای ایجاد رفتار با کیفیت در بسیاری از محیط‌ها **کافی نیستند**

State = goal then "happy" if not "unhappy"

- اهداف فقط یک **تمایز دوتایی خام** بین وضعیت‌های "خوشحال" و "ناراحت" ارائه می‌دهند!

- چگونه وضعیت‌های مختلف جهان را با توجه به اینکه **عامل را چه میزان خوشحال می‌کنند** مقایسه کنیم؟

- برای اینکار از عبارت "**سودمند**" به جای "خوشحالی" استفاده می‌کنیم



عامل مبتنی بر سودمندی - 2

- تابع سودمندی

- تابع سودمندی عامل اساساً **درونی سازی** شده‌ی **معيار عملکرد** است

- معيار عملکرد به هر توالی معینی از حالات محیطی امتیازی اختصاص می‌دهد

- اگر **تابع مطلوبیت داخلی** و **معيار عملکرد خارجی** با هم **توافق داشته باشند**، آنگاه عاملی که اقداماتی را برای به

حداکثر رساندن مطلوبیت خود انتخاب می‌کند، بر اساس معيار عملکرد خارجی، **منطقی(عقلانی)** خواهد بود

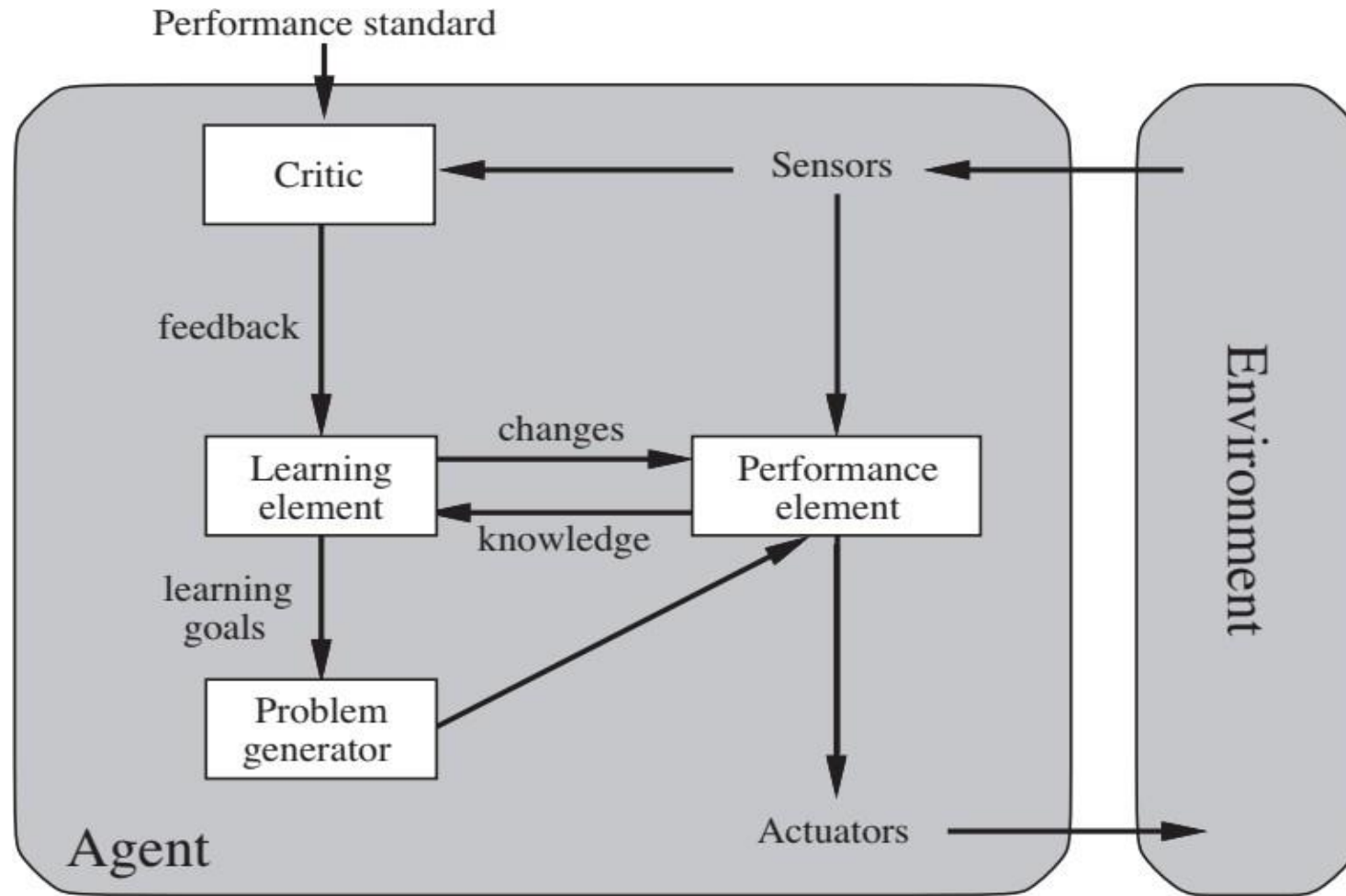
- این تنها راه منطقی بودن نیست

- مانند عامل‌های مبتنی بر هدف، یک عامل مبتنی بر سودمندی از نظر **انعطاف پذیری** و **یادگیری** مزایای بسیاری دارد

عامل مبتنی بر سودمندی - 3

- در دو مورد، اهداف ناکافی هستند، اما یک عامل مبتنی بر سودمندی همچنان می‌تواند تصمیمات منطقی بگیرد
- هنگامی که اهداف متناقضی وجود داشته باشد که فقط برخی از آن‌ها قابل دستیابی هستند، تابع سودمندی مبادله مناسب بین اهداف را مشخص می‌کند
- مثلاً سرعت و ایمنی
- هنگامی که چندین هدف وجود دارد که عامل می‌تواند آن‌ها را هدف قرار دهد، که هیچ یک از آن‌ها با قطعیت قابل دستیابی نیست، سودمندی راهی را فراهم می‌کند که از طریق آن بتوان احتمال موفقیت را با اهمیت اهداف سنجید
- یک عامل منطقی مبتنی بر سودمندی، اقدامی را انتخاب می‌کند که **سودمندی مورد انتظار** از نتایج عمل را به **حداکثر** برساند
- عاملی که تابع سودمندی صریح دارد می‌تواند با یک **الگوریتم همه منظوره (general-purpose algorithm)** تصمیمات منطقی بگیرد

عامل مبتنی بر یادگیری



عامل مبتنی بر یادگیری

- یک عامل یادگیرنده را می توان به چهار جزء مفهومی تقسیم کرد:

- **عنصر عملکرد** (قبلاً کل عامل در نظر گرفته می شد)

- مسئول انتخاب اقدامات خارجی است

- **منتقد**

- به عنصر یادگیری می گوید که عامل با توجه به یک استاندارد عملکردی (performance measure) ثابت چقدر خوب عمل می کند

- **عنصر یادگیری**

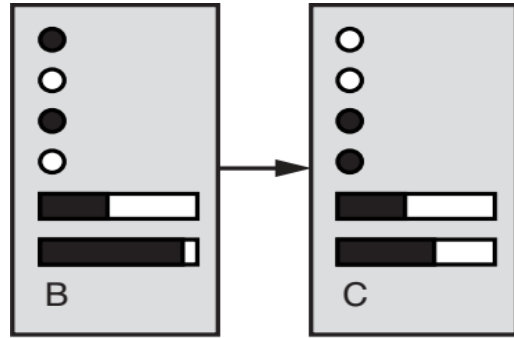
- مسئول ایجاد بهبود

- عنصر عملکرد را برای انجام بهتر در آینده بر اساس بازخوردهای منتقد اصلاح می کند

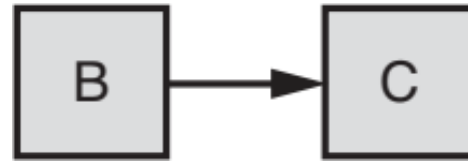
- **مولد مشکل**

- مسئول پیشنهاد اقداماتی است که منجر به تجربیات جدید و آموزنده می شود

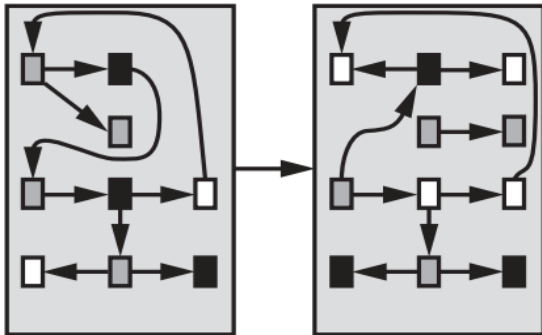
نشان دادن حالت‌ها



(b) Factored



(a) Atomic



(b) Structured

- نمایش اتمی
- هر حالتی از جهان تجزیه ناپذیر است
- ساختار داخلی ندارد
- نمایش فاکتورها
- هر حالت را به مجموعه ثابتی از متغیرها یا ویژگی‌ها تقسیم می‌کند،
- که هر کدام می‌توانند مقداری داشته باشند
- نمایش ساختار یافته
- اشیاء و روابط مختلف و متفاوت آن‌ها را می‌توان به صراحت توصیف کرد