

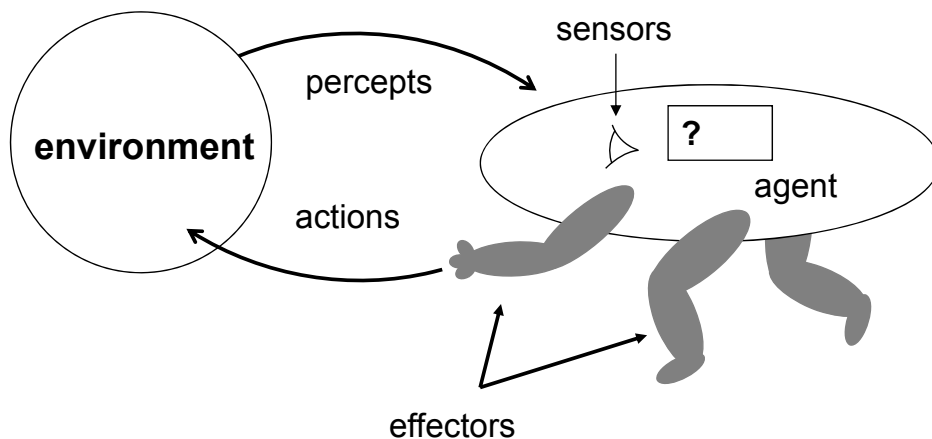
هوش مصنوعی

درس دوم: عامل‌های هوشمند

سید کاوه احمدی

عامل (Agent)

- به هر چیزی اطلاق می‌شود که بتواند محیط (Environment) پیرامون خود را از طریق حسگرهایی (Sensor) درک (Perceive) کند و از طریق عملگرها/اثرکننده‌ها (Actuators, Effectors) روی محیط اثرگذاری داشته باشد.



عامل	حسگرها	مشاهدات	عملگرها	عملها
انسان	چشم، گوش و سایر ارگانها	<ul style="list-style-type: none"> در پایین‌ترین سطح سیگنال‌های الکتریکی پس از پردازش: اشیا (موقعیت، رنگ و...)، صدا (جهت، قدرت و...) 	دست، پا	حرکت دست، راه رفتن، جیغ زدن
ربات	دوربین، محدوده یاب مادون قرمز		موتور و...	
نرم افزار	رشته‌های بیتی کد شده		رشته‌های بیتی کد شده	

عامل عقلانی (Rational Agent)

- عاملی که کار درست را انجام می‌دهد.

؟

— کار درست چیزی است که باعث می‌شود عامل حداکثر موفقیت را به دست آورد.

چگونه و کی ارزیابی می‌شود؟
نیاز به اندازه‌گیری موفقیت داریم.

- توجه کنید که عامل عقلانی کار درست را براساس آنچه درک می‌کند و اعمالی که می‌تواند انجام دهد انتخاب می‌کند.

عامل عقلانی (Rational Agent)

- معیار کارایی (Performance Measure):
 - یک معیار عینی برای اندازه‌گیری میزان موفقیت رفتار عامل است.
 - معیار کارایی بسته به نوع عامل و هدف آن می‌تواند متفاوت باشد.
- عامل عقلانی باید در هر دنباله‌ی ممکن از مشاهدات عامل (دنباله ادراکی) عملی را انتخاب کند که به نظر می‌رسد معیار کارایی را حداکثر می‌کند.

عامل عقلانی (Rational Agent)

- معیار کارایی: آیا عامل به خوبی عمل کرده؟
 - نمی‌توان معیار ثابتی برای تمام عامل‌ها در نظر گرفت.
 - می‌توان از عامل پرسید که چقدر از عملکرد خود راضی است؟
 - نمی‌داند.
 - خود را فریب می‌دهد.
 - بر اندازه کارایی عینی تاکید می‌کنیم:
 - چیزی که مد نظر طراح بوده است.
 - توسط یک مشاهده‌گر خارجی باید مورد ارزیابی و قضاوت قرار گیرد.

عامل عقلانی (Rational Agent)

■ معیار کارایی:

– مثال معیار کارایی برای عامل جاروبرقی؟

■ تمیزی محیط؟

■ میزان زباله جمع‌آوری شده؟

■ میزان جریان برق مصرفی؟

■ میزان سر و صدای تولید شده؟

عامل عقلانی (Rational Agent)

■ عامل عقلانی باید به ازای هر دنباله ادراکی، عملی را انجام دهد که معیار کارایی مشخص شده برایش را حداکثر کند.

– با توجه به شواهد ارائه شده توسط دنباله ادراکی و دانش درونی‌ای که عامل دارد.

■ منطقی \neq عقل کل (Omniscience)

– مشاهدات ممکن است تمام اطلاعات مربوطه را عرضه نکند.

■ منطقی \neq روشن بین

– نتیجه اعمال ممکن است آنگونه که عامل انتظار داشته نباشد

■ بنابراین منطقی \neq موفق

■ منطقی \leftarrow پویش، یادگیری، خود مختار

عامل عقلانی (Rational Agent)

■ عقل کل (Omniscience)

- عقل کل (عامل کامل) خروجی واقعی فعالیت خود را می‌داند و می‌تواند بر اساس آن عمل کند.
- عامل کامل در واقعیت غیرممکن است.
- برای عبور از خیابان به چپ و راست نگاه کردم و شروع به عبور کردم... ناگهان شهاب سنگی بر سرم سقوط کرد و مرا کف خیابان پهن کرد!
- آیا من کار غیر عقلانی‌ای کرده بودم؟
- آیا من سبک مغزم؟

عامل عقلانی (Rational Agent)

■ عقل کل (Omniscience)

- عقلانی بودن بر امید موفقیتی که درک می‌شود متمرکز است.
- در عبور از خیابان اگر در قفس بودم موفق‌تر بودم اما عقلانی‌تر نبودم!
- تعریف ما از عقلانیت به دانش کل نیازی ندارد و انتخاب عقلانی تنها وابسته به دنباله ادراکی تاکنون است.
- آیا بر اساس آنچه درک کرده‌ام درست رفتار کرده‌ام؟
- اگر درکیات کافی نبود چطور؟ مثلاً به چپ و راست نگاه نکردن پیش از عبور از خیابان
- عامل عقلانی برای انتخاب عمل فقط به حداکثر کردن معیار کارایی‌اش فکر می‌کند.

عامل عقلانی (Rational Agent)

■ بنابراین می‌توان گفت عقلانیت عامل به چهار پارامتر بستگی دارد:

1. معیار کارایی
2. دنباله ادراکات
3. دانش درباره محیط
4. عملیاتی که عامل می‌تواند انجام دهد

■ عامل عقلانی ایده‌آل:

- باید برای هر دنباله ادراکی ممکن، بر مبنای دانش درونی خود (Built-in Knowledge) عملی را انجام دهد که معیار کارایی‌اش را ماکزیمم کند.

عامل عقلانی (Rational Agent)

■ نگاهت ایده‌آل از دنباله ادراکی به عمل

- هر عامل را می‌توان از طریق ساخت یک جدول از کارهایی که عامل باید در پاسخ به هر دنباله ادراکی ممکن بروز دهد توصیف کرد.
- این جدول معمولاً حجیم است بنابراین باید محدودیتی برای اندازه دنباله ادراکات مشخص کرد.
- اغلب می‌توان چنین نگاهتی ایجاد کرد، حتی در مورد عامل‌هایی با رفتار تصادفی.
- در چنین مواردی باید رفتار میانگین عامل را استخراج کرد.

عامل عقلانی (Rational Agent)

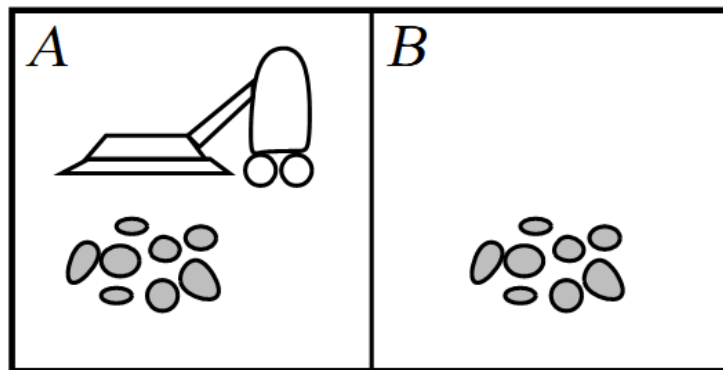
■ نگاهت ایده‌آل از دنباله ادراکی به عمل

– گاهی نیاز به ایجاد جدول نیست و می‌توان جدول را با یک تابع جایگزین کرد.

Percept x	Action z	Function SQRT(x)
1.0	1.0000000000000000	$z \leftarrow 1.0$ // initial guess
1.1	1.048808848170152	Repeat until $ z^2 - x < 10^{-15}$
1.2	1.095445115010332	
1.3	1.140175425099138	$z \leftarrow z - \frac{(z^2 - x)}{(2z)}$
1.4	1.183215956619923	
1.5	1.224744871391589	end
1.6	1.264911064067352	
1.7	1.303840481040530	return z
.	.	
.	.	

قسمتی از نگاهت ایده‌آی برای مسئله جذرگیری و یک تابع که این نگاهت را انجام می‌دهد.

عامل دنیای مکش



■ محیط: خانه‌های A و B

■ ادراکات (Percepts): موقعیت (location) و شرایط (contents)

– [A, Dirty]

■ عمل‌ها (Actions): چپ، راست، مکش، بدون عمل

عامل دنیای مکش

■ معیار کارایی:

- برحسب میزان خاک تمیز شده
- عامل با تمیز کردن زمین، دوباره ریختن خاک و دوباره تمیز کردن آن معیار کارایی خود را حداکثر می کند
- امتیاز برای تمیز کردن خاک و جریمه برای مصرف برق
- معیار کارایی: «کف تمیز»
- انجام کار معمولی ولی طولانی
- انجام کار با انرژی زیاد اما کوتاه
- معیار کارایی باید برحسب چیزی که واقعا در محیط نیاز داریم طراحی شود نه برحسب چیزی که فکرمی کنیم عامل باید آنگونه باشد

عامل دنیای مکش

■ برنامه عامل:

- یک Lookup-Table شامل تمام دنباله‌های ادراکی ممکن و عمل مناسب با توجه با آن

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
...	...

- این جدول بینهایت سطر داشته دارد در حالیکه برای انتخاب عمل به همهی سطرها نیاز نداریم (چرا؟).

■ برنامه عامل:

– انتخاب فعالیت می‌تواند بر اساس یکسری قوانین شرط – عمل (Condition-Action Rules) انجام شود.

```
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
```

خودمختاری (Autonomous)

- یک عامل خودمختار است اگر رفتارش براساس تجربیات خودش تعیین شود.
- اگر رفتار عامل کاملاً بر مبنای دانش درونی پیش ساخته باشد، آنگاه هیچ نیازی به توجه به ادراکات بیرونی نخواهد داشت و این معادل است با از دست دادن خودمختاری
- یک عامل مصنوعی هوشمند، با دارا بودن مقداری دانش اولیه به همراه قابلیت یادگیری و داشتن وقت کافی برای ایجاد انطباق، قادر است در حوزه‌ی وسیعی از محیط‌ها موفق شود.

محیط کار (Task Environment)

▪ یک محیط کار از ۴ جز تشکیل می‌شود:

1. معیار کارایی میزان موفقیت عامل را ارزیابی می‌کند (Performance)
2. آنچه عامل در مورد محیط می‌داند – دانش اولیه از محیط (Environment)
3. اعمالی که عامل می‌تواند انجام دهد (Actuator)
4. مشاهداتی که عامل دریافت کرده است – دنباله ادراکی تا کنون (Sensor)

- در طراحی یک عامل، اولین گام مشخص کردن محیط کار آن در حد ممکن است (PEAS)
- در برخی از مراجع تحت عنوان PAGE به این اجزا اشاره شده است:

Perceptions, Actions, Goals, Environment

PEAS

▪ Agent: سامانه تشخیص پزشکی

- معیار کارایی: سلامت بیماران، هزینه کم
- محیط: بیمار، بیمارستان، کارمندان
- اعمال: سوالات، آزمایش‌ها، تشخیص‌ها
- حسگرها: ورودی اطلاعات در مورد بیمار

▪ Agent: آموزنده تعاملی زبان انگلیسی

- معیار کارایی: گرفتن حداکثر نمره دانش آموزان در آزمون
 - محیط: مجموعه دانش آموزان
 - اعمال: درس‌ها، تمرین‌ها، پیشنهادات
 - حسگرها: صفحه کلید
-

نوبت شماست:

▪ مشخص کنید PEAS را برای:

- eProf
- eStudent

- کاملاً قابل مشاهده، قابل مشاهده جزئی
- قطعی، غیر قطعی
 - راهبردی
- رویدادی، ترتیبی
- ایستا، پویا
- گسسته، پیوسته
- تک عامله، چند عامله
 - چند عامله رقابتی، چند عامله همیاری

محیط کاملاً قابل مشاهده در مقابل قابل مشاهده جزئی

- Fully vs. Partially Observable
 - در یک محیط قابل مشاهده، تمامی جنبه‌های محیط توسط حسگرهای عامل قابل تشخیص است.
 - محیط قابل دسترسی راحت است، زیرا عامل نیازمند نگهداری هیچ وضعیت داخلی برای حفظ اثرات دنیا نیست.

■ Deterministic vs. Stochastic

■ در یک محیط قطعی، وضعیت بعدی با دانستن وضعیت فعلی و عمل انتخاب شده به طور کامل قابل پیش‌بینی است.

– اگر محیط غیرقابل دسترس باشد، معمولاً غیرقطعی است.

■ در مثال جاروبرقی اگر حسگرها خوب عمل نکنند، با اینکه محیط در دسترس نیست، ولی می‌توان وضعیت بعدی را تعیین نمود.

– اگر محیط برای یک عامل قطعی و برای سایر عامل‌ها غیرقطعی باشد، محیط استراتژیک (Strategic) است.

محیط رویدادی (مرحله‌ای یا تقسیم پذیر) در مقابل پی در پی (ترتیبی)

■ Episodic vs. Sequential

■ در یک محیط رویدادی، تجربیات عامل می‌تواند به بخش‌های اتمیک تقسیم شود به نحوی که کیفیت هر عمل فقط به بخش خودش بستگی دارد.

■ در یک محیط رویدادی تجربه‌ی عامل به بخش‌های مجزا تقسیم می‌شود که در یک شامل توالی ادراکات و اعمال عامل هستند. هر بخش از آنچه در بخش قبلی انجام شده مستقل است.

– عامل نیازی به برنامه‌ریزی برای آینده ندارد.

■ در محیط‌های ترتیبی تصمیم جاری بر کلیه‌ی تصمیمات آینده، تأثیر می‌گذارد.

■ Static vs. Dynamic

- در یک محیط پویا، ممکن است محیط هنگام تصمیم‌گیری (یا عمل) عامل تغییر کند.
 - محیط پویا قابل دسترس نیست.
- در محیط ایستا عامل نگران گذشت زمان نیست و نیازی ندارد که در هنگام تصمیم‌گیری، دنیای اطرافش را تحت نظر داشته باشد.
- محیط نیمه پویا (Semi-Dynamic) محیطی که با گذر زمان تغییر نمی‌کند اما معیار کارایی عامل تغییر می‌کند.
 - شطرنج با ساعت (۴۰ حرکت در یک ساعت پس از آن بدون ساعت)

■ Discrete vs. Continuous

- در یک محیط گسسته، تعداد محدود، مشخص و صریحا تعریف شده‌ای از ادراکات و اعمال وجود دارد.
- محیط گسسته است اگر مشاهدات و اعمال محیط با اعداد گسسته قابل بیان باشند.
 - محیط بازی شطرنج گسسته است. (تعداد متناهی حرکت در هر نوبت وجود دارد).
 - محیط رانندگی تاکسی پیوسته است. (سرعت، موقعیت تاکسی و ... دارای مقادیر پیوسته هستند).

محیط تک عاملی در مقابل چند عاملی

- Single-agent vs. Multi-agent
 - محیطی که یک عامل در آن تصمیم گیری می کند، تک عامله است.
 - محیط حل جدول کلمات متقاطع تک عامله است.
 - محیطی که بیش از یک عامل در آن وجود دارند، محیط چند عامله است.
 - چند عامله رقابتی (Competitive Multi-agent) عوامل با هم رقیب هستند. مانند شطرنج.
 - چند عامله همکاری (Cooperative Multi-agent) عوامل با هم همکاری هستند. مانند زمانی که در یک بازی سه نفره دو نفر علیه دیگری متحد می شوند.
 - به نظر ساده می رسد اما کدامیک از موجودیت های محیط باید به عنوان عامل در نظر گرفته شوند؟

انواع محیط

گسسته	ایستا	اپیزودیک	قطعی	قابل مشاهده	عامل
YES	Semi	NO	YES	FULL	شطرنج با ساعت
YES	YES	NO	NO	FULL	شطرنج بدون ساعت
NO	NO	NO	NO	PARTIAL	راننده تاکسی
NO	NO	NO	NO	PARTIAL	تشخیص پزشکی
NO	Semi	YES	YES	FULL	تحلیل تصاویر
NO	NO	YES	NO	PARTIAL	روبات جا به جا کننده قطعات
NO	NO	NO	NO	PARTIAL	کنترلر پالایشگاه
YES	NO	NO	NO	PARTIAL	آموزگار زبان تعاملی

- ساده‌ترین محیط:
 - کاملاً قابل مشاهده، قطعی، رویدادی، ایستا، گسسته و تک عامله
- محیط بیشتر پیش رو:
 - قابل مشاهده جزئی، غیر قطعی، ترتیبی، پویا، پیوسته و چند عامله

ساختار یک عامل

عامل = معماری + برنامه

- برنامه عامل: برای اجرای تابع عامل (f) روی معماری عامل اجرا می‌شود.
 - تابع عامل: نگاشتی از دنباله ادراکات عامل (دنباله ادراکی) به عمل‌ها

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

- عمل درست ممکن است وابسته به وضعیت فعلی یا کل یا بخشی از دنباله ادراکی باشد.
- برای ایجاد f ممکن است نیاز باشد برنامه عامل در محیط اجرا شود.
- معماری عامل: سخت افزار/نرم افزار که برنامه عامل روی آن اجرا می‌شود.

■ عامل مراجعه کننده به جدول (Table-lookup Agent):

- اگر بتوانیم عمل عامل را برای هر دنباله ادراکی ممکن مشخص کنیم، هرچه درباره عامل لازم است را خواهیم دانست.
- استفاده از یک جدول جستجو (Lookup-Table) برای نگه داشتن کلیه وضعیت‌ها و انتخاب یک عمل از جدول با توجه به شرایط.

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

```
static: percepts, a sequence initially empty  
         table, a table of actions, indexed by percept sequence  
  
append percept to the end of percepts  
action ← LOOKUP(percepts, table)  
return action
```

■ معایب استفاده از جدول وضعیت:

- جدول بسیار بزرگ و جستجو در آن زمانبر است (برای بازی شطرنج 35^{100} سطر خواهد داشت).
- ساخت جدول برای طراح زمانبر است.
- عامل هیچگونه خودمختاری ندارد و اگر در محیط ناشناخته‌ای قرار بگیرد شکست می‌خورد.
- حتی اگر قابلیت یادگیری داشته باشد، یادگیری در محیط برای همه‌ی مدخل‌های جدول، بسیار طول می‌کشد.

انواع عامل‌ها

- جنبه‌های مختلف یک عمل، انواع مختلف برنامه‌های عامل را پیشنهاد خواهد کرد.
- انواع برنامه‌های عامل
 - عامل‌های واکنشی ساده
 - عامل‌های واکنشی مدل‌گرا
 - عامل‌های هدف‌گرا
 - عامل‌های سودمند
- تمام اینها می‌توانند به عامل‌های یادگیر تبدیل شوند.

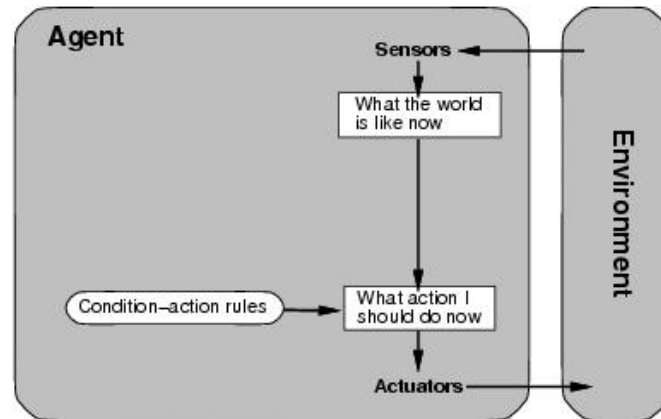
عامل واکنشی ساده (Simple Reflex Agent)

- این عامل‌ها با استفاده از قواعد از پیش تعریف شده‌ای به فرم «اگر ... آنگاه ...» عمل می‌کنند.

If dirty then suck

- به این قواعد Condition-Action Rules گفته می‌شود.

عامل واکنشی ساده (Simple Reflex Agent)



■ انتخاب یک عمل بر اساس درک فعلی.

– بدون در نظر گرفتن سابقه ادراک

■ کاهش بسیار حالت‌ها (نسبت به جدول) با شناختن شرط/عمل‌ها

عامل واکنشی ساده (Simple Reflex Agent)

```
function SIMPLE-REFLEX-AGENT(percept) returns an action
```

```
static: rules, a set of condition-action rules
```

```
state ← INTERPRET-INPUT(percept)
```

```
rule ← RULE-MATCH(state, rule)
```

```
action ← RULE-ACTION[rule]
```

```
return action
```

■ تنها زمانی کار می‌کند که محیط کاملاً قابل مشاهده باشد. در غیر این صورت حلقه

بینهایت ممکن است رخ دهد

عامل دنیای مکش (عامل واکنشی ساده)

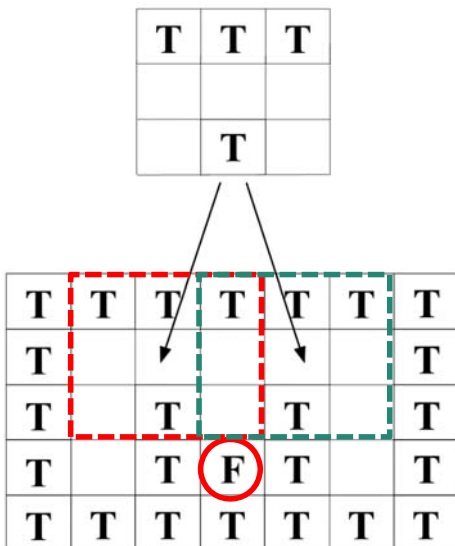
- تصمیم گیری در آن بر اساس مکان فعلی و کثیف بودن آن مکان است.
- انتخاب فعالیت بر اساس موقعیت شرطی است:

```
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
```

کاهش از 4^T ورودی به 4 ورودی اما هنوز ممکن است تعداد حالتها بسیار زیاد باشد

عامل واکنشی مدل گرا (Model-based Reflex Agent)

- عاملها گاهی نیازمند داشتن تعدادی وضعیت داخلی (Internal State) است تا بین حالت‌هایی از دنیای بیرون که در عین تولید دنباله‌ی ادراکات یکسان، ماهیتا با هم متفاوتند تمایز قائل شود.



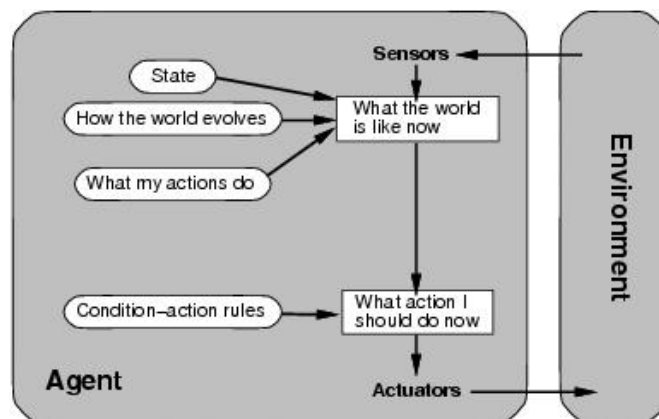
عامل واکنشی مدل گرا (Model-based Reflex Agent)

- عامل واکنشی ای که از یک وضعیت داخلی برای نگهداری مشاهدات قبلی استفاده می کند تا بتواند با وضعیت های به ظاهر مشابه اما ماهیتاً متفاوت روبرو شود.
- تغییر وضعیت بر اساس دانشی (که در طول زمان) از محیط کسب شده.
- برای این منظور به دو نوع دانش نیاز داریم:

یک مدل از دنیا {
■ اطلاعاتی در مورد اینکه جهان خارج مستقل از عامل چگونه رفتار می کند.
■ اطلاعاتی در مورد اینکه واکنش های عامل چه تاثیری روی جهان بیرون دارد.

- توانایی مقابله با محیط های نیمه قابل مشاهده.

عامل واکنشی مدل گرا (Model-based Reflex Agent)



- عامل دنیایی را که می بیند ردیابی می کند.
- تا از طریق آن تمایز بین وضعیت های دنیا که در ظاهر ورودی ادراکی یکسانی دارند ولی در واقع معنی کاملاً متفاوتی دارند را میسر سازد.
- از آنجایی ناشی می شود که حسگرها نمی توانند دسترسی کامل به وضعیت دنیا را به وجود آورند.
- عامل باید یک حالت داخلی را ذخیره کند.

عامل واکنشی مدل گرا (Model-based Reflex Agent)

- دانش در مورد اینکه دنیا چگونه کار یا تغییر می کند یک مدل از دنیا نامیده می شود.
 - عاملی که از این مدل استفاده می کند، عامل مبتنی بر مدل نامیده می شود. (Model-Based Agent)
 - عامل واکنشی مبتنی بر مدل، عاملی که مسیر دنیا را دنبال می کند نیز نامیده می شود. (Agent that keeps track of the world)

عامل واکنشی مدل گرا (Model-based Reflex Agent)

```
function REFLEX-AGENT-WITH-STATE(percept) returns an action

static: rules, a set of condition-action rules
         state, a description of the current world state
         action, the most recent action.

state ← UPDATE-STATE(state, action, percept)
rule ← RULE-MATCH(state, rule)
action ← RULE-ACTION[rule]
return action
```

- آیا عامل واکنشی برای بازی شطرنج مناسب است؟
- عامل مراجعه کننده به جدول برای بازی شطرنج نیاز به 35^{100} مدخل دارد و در مورد عامل واکنشی نیز اوضاع چندان بهتر نیست!

عامل هدف گرا (Goal-based Agent)

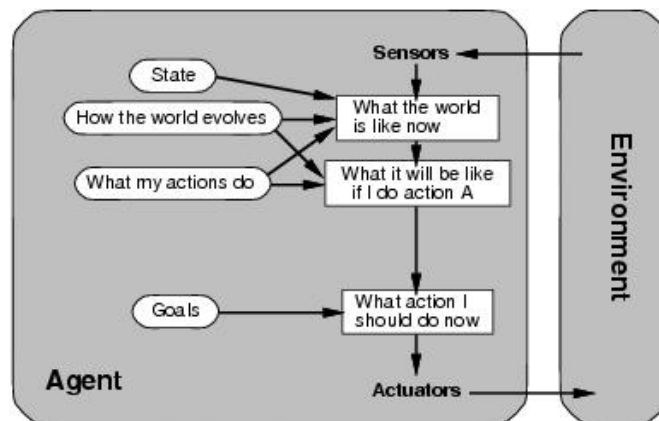
- دانستن وضعیت فعلی همیشه کافی نیست.
- عامل به اطلاعات نیاز دارد که وضعیت هدف را برای او توصیف کند.
- در این صورت عامل می تواند با ترکیب این اطلاعات با اطلاعاتی در مورد نتایج اقدامات ممکنش، بهترین عملی که می تواند او را به هدف نزدیک تر کند را انتخاب نماید.

T	T	T	T	T	T	T
T						T
T		T		T		T
T		T	F	T		T
T	T	T	T	T	T	T

عامل هدف گرا (Goal-based Agent)

- گاه رسیدن به هدف با یک تصمیم حاصل می‌شود:
 - Action → Goal
- و گاه با چند تصمیم:
 - Action → Action → ... → Goal
- جستجو (Search) و برنامه‌ریزی (Planning) زیرشاخه‌هایی از هوش مصنوعی هستند که روش‌های یافتن این توالی را بررسی می‌کنند.
- لازم است همواره آینده را در نظر بگیرد.
 - چه اتفاقی می‌افتد اگر...
 - با قوانین شرط عمل تفاوت دارد

عامل هدف گرا (Goal-based Agent)



- انتخاب اعمال به نحوی که به یک هدف مشخص شده یا محاسبه شده برسیم
 - منظور از هدف توصیفی از یک وضعیت مطلوب است.
 - دانش موجود درباره حالت جاری همیشه برای تصمیم‌گیری کافی نیست و باید تصمیم‌گیری در مورد وضعیت‌های خوب هدف را در نظر گرفت.
 - متدبرانه (Deliberative) به جای واکنشی
 - این نوع عامل کارایی کمتری نسبت به روش‌های دیگر دارد، اما قابلیت انعطاف بیشتری دارد.

تفاوت عامل‌های واکنشی و هدف‌گرا

■ در طراحی عامل‌های واکنشی طراح برای حالات متفاوت، عملی درست را محاسبه می‌کند. در عامل‌های هدف‌گرا، عامل می‌تواند دانش خود را در مورد چگونگی واکنش به‌هنگام سازد.

– برای عامل واکنشی ما مجبور به دوباره نویسی تعداد زیادی قوانین شرط – عمل خواهیم بود.

– عامل هدف‌گرا نسبت به رسیدن به مقاصد متفاوت انعطاف پذیر است.

– به سادگی با تعیین یک هدف تازه، می‌توانیم عامل هدف‌گرا را به رفتار تازه برسانیم.

عامل مبتنی بر سودمندی (Utility-based agents)

■ اغلب برای رسیدن به هدف، توالی اعمال متعددی وجود دارد اما برخی از توالی‌ها بهترند.

– زمانی که چند هدف داریم نیز به همین ترتیب است.

■ اگر یک وضعیت نسبت به وضعیت دیگر ارجحیت داشته باشد، می‌گوییم از سودمندی

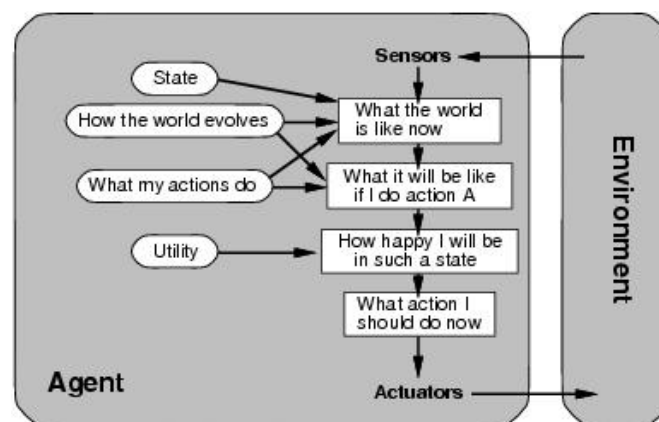
بیشتری برای عامل برخوردار است.

T	T	T	T	T	T	T
T						T
T		T		T		T
T		T	F	T		T
T	T	T	T	T	T	T

عامل مبتنی بر سودمندی (Utility-based agents)

- سودمندی: تابعی که یک وضعیت (دنباله‌ای از وضعیت‌ها) را به یک عدد حقیقی (درجه‌ی مطلوبیت) نگاشت می‌کند.
- یک توصیف کامل از تابع سودمندی اجازه تصمیم‌گیری عقلانی در دو موقعیت زیر که در آنها اهداف مشکل‌ساز هستند را فراهم می‌کند:
 1. اهداف متضاد که فقط یکی از اهداف قابل ارضا است مانند سرعت در مقابل ایمنی.
 2. اهداف چندگانه که هیچکدام با قطعیت قابل ارضا نیستند.

عامل مبتنی بر سودمندی (Utility-based agents)



- آیا عامل هدف گرا برای بازی شطرنج مناسب است؟
 - نشدنی است! برای هر حرکت باید بررسی کند آیا این حرکت در نهایت می تواند منجر به مات کردن حریف شود یا خیر!
- عامل مبتنی بر سودمندی انتخاب بهتری است.
 - هدف نهایی (مات کردن) را در نظر نمی گیرد و برای دستیابی به معیارهای محلی تلاش می کند.

عامل یادگیرنده (Learning Agent)

- تمام برنامه های عامل گذشته روش های انتخاب یک عمل را توضیح می دهند. اما هنوز مبدا این برنامه ها توضیح داده نشده است.
- برنامه های عامل به دو طریق به وجود می آیند:
 - به صورت دستی
 - از طریق یادگیری
- آموزش به آنها به جای دستور دادن به آنها
- مزیت: استحکام (robustness) برنامه در محیط های در ابتدا ناشناخته

عامل یادگیرنده (Learning Agent)

■ اجزای مفهومی جهت یادگیری:

- عنصر کارایی: ادراکات را دریافت و در مورد اقدامات تصمیم‌گیری می‌کند.
- عنصر یادگیری: براساس بازخورد دریافتی از منتقد، عنصر کارایی را توسعه می‌دهد.
- منتقد: براساس ادراکات و یک استاندارد کارایی ثابت، میزان موفقیت عامل را به عنصر یادگیری اعلام می‌کند.
- مولد مسئله (Problem Generator): مسئول پیشنهاد اقداماتی است که به تجربیاتی تازه و آموزنده منجر خواهد شد.

■ Exploration (اکتشاف) در برابر Exploitation

عامل یادگیرنده (Learning Agent)

