



دانشگاه صنعتی امیرکبیر

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

عوامل‌های هوشمند

«هوش مصنوعی: یک رهیافت نوین»، فصل ۲

ارائه‌دهنده: سیده فاطمه موسوی

نیم‌سال اول ۱۴۰۰-۱۳۹۹

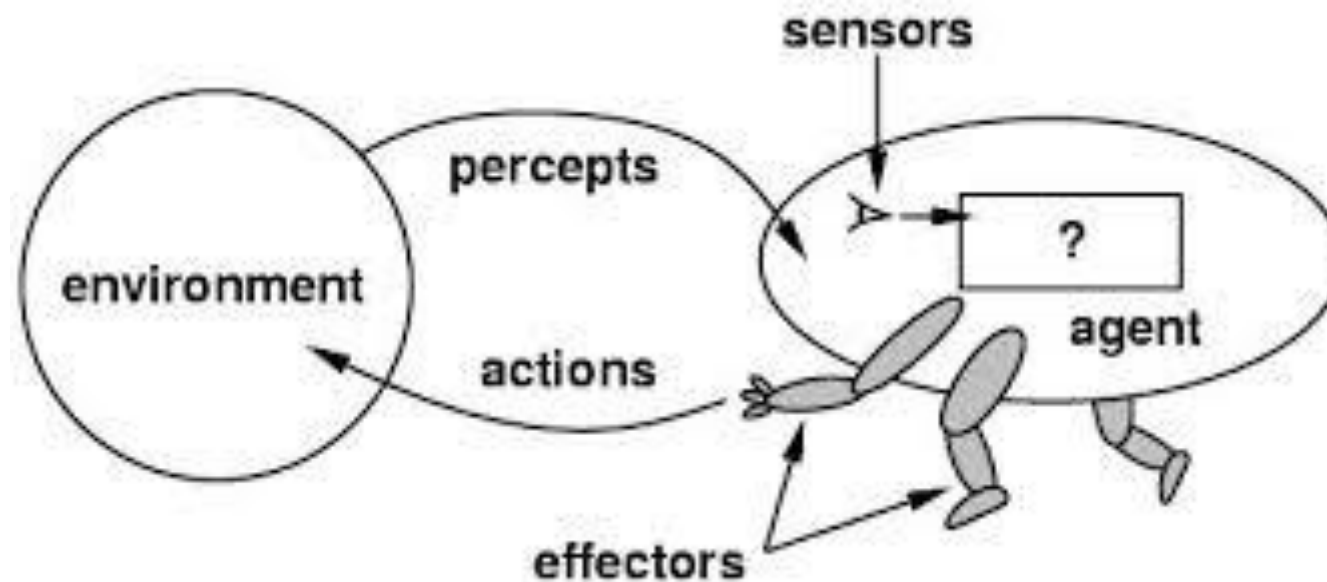
رئوس مطالب

- مفهوم عقلانیت
 - عامل ها و محیط
 - عقلانیت
 - دانای کل، یادگیری، خودمختاری
 - ماهیت محیطها
 - تعیین محیط وظیفه (PEAS)
 - ویژگی های محیط وظیفه
- ساختار عامل ها
 - برنامه های عامل
 - عامل مبتنی بر جدول
 - انواع عامل
 - عامل های واکنشی ساده
 - عامل های واکنشی مبتنی بر مدل
 - عامل های مبتنی بر هدف
 - عامل های مبتنی بر سودمندی
 - عامل های مبتنی بر یادگیری

مفهوم عقلانیت

عامل چیست؟

- عامل (Agent) هر چیزی است که بتواند
- محیط (environment) خود را از طریق حس‌گرهایش (sensor) درک (percept) کند
- و از طریق اقدام‌گرها (actuators) در آن محیط اقدامی (action) انجام می‌دهد.



نمونه‌هایی از عامل‌ها

- عامل انسانی

- حس‌گرها: چشم‌ها، گوش‌ها و ...
- اقدام‌گرها: دست‌ها، پاها، دهان و ...

- عامل روبات

- حس‌گرها: دوربین‌ها و فاصله‌یاب مادون قرمز
- اقدام‌گرها: انواع موتورها

- عامل نرم‌افزاری

- حس‌گرها: دریافت‌ها از صفحه‌کلید، محتویات فایل، بسته‌های شبکه
- اقدام‌گرها: نمایش روی صفحه‌نمایش، نوشتن در فایل، ارسال بسته‌ها در شبکه

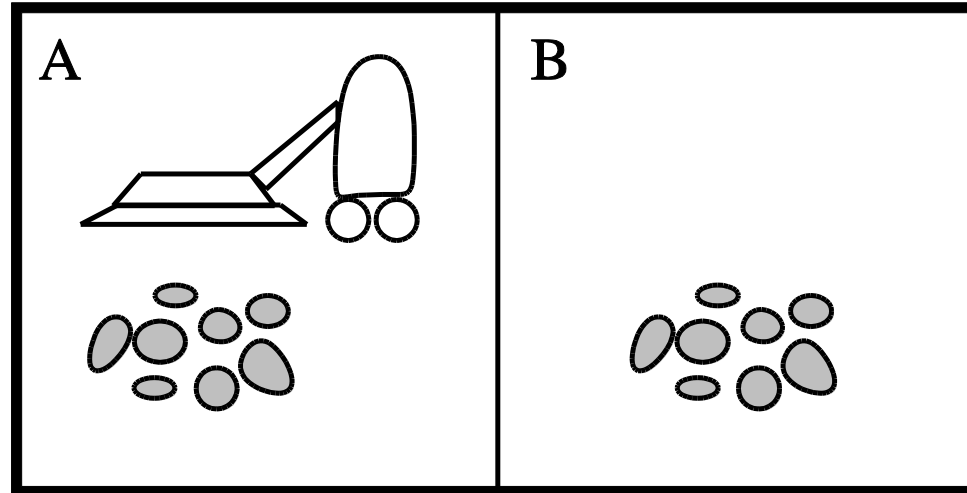
عامل و محیط (Agents and Environments)

- رفتار عامل توسط تابع عامل (Agent function) توصیف می‌شود که دنباله ادراک را به یک عمل نگاشت می‌دهد:

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

- تابع عامل را می‌توان به صورت یک **جدول** نمایش داد. مشکل؟
 - جدول** یک **مشخصه‌ی بیرونی (external)** از عامل است.
 - از **دیدگاه درونی (internally)**، تابع عامل برای یک عامل هوشمند به صورت یک **برنامه‌ی عامل (agent program)** پیاده‌سازی می‌شود.
- تابع عامل یک توصیف ریاضی انتزاعی است در حالی که برنامه عامل یک پیاده‌سازی واقعی است که در سیستم فیزیکی در حال اجرا است.

دنیای جاروبرقی



- محیط: دو مربع A و B، گرد و خاک
- ادراک: [مکان، وضعیت تمیزی/کثیفی]
- مانند: [A, Dirty]
- اقدام‌ها: برو به چپ، برو به راست، تمیز کن، هیچ کاری نکن (NoOp)

عامل دنیای جاروبرقی

if the current square is dirty,
 then suck;
otherwise,
 move to the other square.

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 عقلانی (Rational agent)

- عامل عقلانی عاملی است که کار درست انجام دهد.
- تمامی ردیف‌ها در جدول برای تابع عامل به‌درستی پر شوند.
- کار درست چیست؟
- عامل در محیط با توجه به ادراکی که دریافت می‌کند، دنباله‌ای از اقدام‌ها انجام می‌دهد. این دنباله از اقدام‌ها، باعث می‌شود محیط به دنباله‌ای از وضعیت‌ها برسد. اگر دنباله مطلوب (desirable) بود آنگاه عامل خوب رفتار کرده است.
- این مفهوم از مطلوبیت با یک معیار کارایی (performance measure) سنجیده می‌شود که هر دنباله ورودی از وضعیت‌های محیط (environment states) را ارزیابی می‌کند.
- نمونه‌هایی از معیارهای کارایی عامل جاروبرقی؟
- اندازه آشغال‌هایی که در یک شیف ۸ ساعته تمیز شده است.
- اعطای یک امتیاز مثبت به هر مربع تمیز در هر گام زمانی و جریمه برای میزان برق مصرفی و سروصدای تولیدشده

عقلانیت (Rationality)

- این که در هر لحظه چه چیزی عقلانی است به چهار چیز وابسته است:
- معیار کارایی که میزان موفقیت را تعیین می کند.
- دانش قبلی (prior knowledge) عامل در مورد محیط
- عمل هایی که عامل می تواند انجام دهد.
- دنباله ی ادارک عامل تا این لحظه

• تعریف عامل عقلانی (Rational agent)

- برای هر دنباله ادراکی ممکن، **عامل عقلانی** باید عملی را برای انجام انتخاب کند که **انتظار می رود** معیار کارایی اش را **ماکزیمم کند**. انتخاب عمل، باید با توجه به شواهد و نتایجی که از دنباله ادراکی و دانش درونی عامل به دست می آید، صورت گیرد.

For each possible percept sequence, **a rational agent** *should select an action* that is **expected to maximize** its *performance measure*, given the evidence provided by *the percept sequence* and whatever *built-in knowledge* the agent has.

عقلانیت (Rationality)

- آیا این عامل جاروبرقی عقلانی است؟
- اگر مربع کثیف بود آن را تمیز کن در غیر این صورت به مربع مجاور برو.
- بستگی دارد!!
- معیار کارایی
- جریمه برای مصرف سوخت؟
- محیط
- آشغال جدید در محیط ظاهر می شود؟
- اقدام گرها
- عمل No-op؟
- حس گرها
- تنها حس گر کثیفی محل را دارد؟

عقلانیت در مقابل دانای کل (Rationality vs. Omniscience)

- عقلانیت \neq دانای کل، عقلانیت \neq کامل بودن (perfection)
- یک عامل **دانای کل** (Omniscience) نتیجه واقعی اعمالش را می‌داند و می‌تواند براساس آن عمل کند.
- همه چیزدانی با دانش نامحدود در دنیای واقعی غیر ممکن است.
- عقلانیت، کارایی مورد انتظار را ماکزیمم می‌کند در حالی که کامل بودن کارایی واقعی را ماکزیمم می‌کند.
Rationality maximizes *expected* performance, while perfection maximizes *actual* performance.
- مثال: رد شدن از خیابان و صدمه دیدن به دلیل سقوط درب هواپیما
- چرا تعریف ما از عقلانیت نیاز به همه چیزدانی ندارد؟
- زیرا انتخاب عقلانی تنها وابسته به ادراک تا کنون است.
The rational choice depends only on the percept sequence *to date*.
- همچنین وابسته به اعمالی است که عامل می‌تواند انجام دهد.

نیازمندی‌های یک عامل عقلانی

- جمع‌آوری اطلاعات/اکتشاف (Information gathering/exploration)

- انجام اعمال به منظور به‌روزرسانی ادراک آینده برای به‌دست آوردن اطلاعات مفید

Doing actions *in order to modify future percepts*



- جمع‌آوری اطلاعات کارایی مورد انتظار را افزایش می‌دهد.

- مثال: عمل «نگاه کردن» در هنگام عبور از خیابان

- شکل دیگری از جمع‌آوری اطلاعات، از طریق اکتشاف صورت می‌گیرد.

- مثال: انجام اکتشاف عامل جاروبرقی در هنگام رها کردن آن در یک محیط اولیه ناشناخته

- یادگیری از ادراک

- به‌روزرسانی و اصلاح دانش قبلی عامل از محیط

- اگر محیط عامل کاملاً از قبل شناخته شده باشد در چنین مواردی عامل نیاز به درک کردن و آموختن ندارد زیرا

به درستی عمل خواهد کرد. البته چنین عواملی خیلی آسیب‌پذیرند.

نیازمندی‌های یک عامل عقلانی

- هر عامل براساس دو دانش می‌تواند تصمیم‌گیری کند:
 - دانش اولیه (prior knowledge)
 - تجربیات (experience)
- **خودمختاری (Autonomy)**
 - یک عامل خودمختار است اگر رفتار آن با استفاده از تجربیات خودش (با قابلیت برای یادگیری و تطبیق) تعیین گردد و تنها وابسته به دانش قبلی طراح نباشد.
 - یادگیری برای جبران و اصلاح دانش غلط یا ناقص قبلی خود
 - هنگامی که عامل تجربه ندارد یا تجربه کمی دارد باید تصادفی عمل کند مگر آن که طراح با قرار دادن دانش اولیه به آن کمک کند.
 - مثال: جاروبرقی که یاد می‌گیرد کجا و چه وقت آشغال می‌ریزد بهتر از جاروبرقی عمل می‌کند که چنین توانایی نخواهد داشت.

ماهیت محیط‌ها

محیط و وظیفه (Task Environment)

- مواردی که عقلانیت عامل به آن وابسته است، محیط وظیفه گفته می‌شود. محیط وظیفه عامل با توصیف PEAS تعیین می‌شود.
 - معیار کارایی (Performance measure)
 - محیط (Environment)
 - اقدام‌گرها (Actuators)
 - حس‌گرها (Sensors)
- اولین گام در طراحی عامل‌های هوشمند، تعیین محیط وظیفه به کامل‌ترین شکل ممکن است.

نمونه‌های PEAS

• عامل راننده تاکسی خودکار

- معیار کارایی: امنیت، سرعت، رعایت قوانین، راحتی مسافرت، حداکثر بهره‌وری
- محیط: جاده‌ها، ترافیک‌های دیگر، عابرین پیاده، مشتریان
- اقدام‌گرها: هدایت‌کننده‌ها، شتاب‌دهنده‌ها، ترمز، سیگنال، نمایش‌گر، بوق
- حس‌گرها: دوربین‌ها، مکان‌یاب، سرعت‌سنج، GPS، صفحه‌کلید، حس‌گرهای موتور، کیلومترشمار، شتاب‌سنج



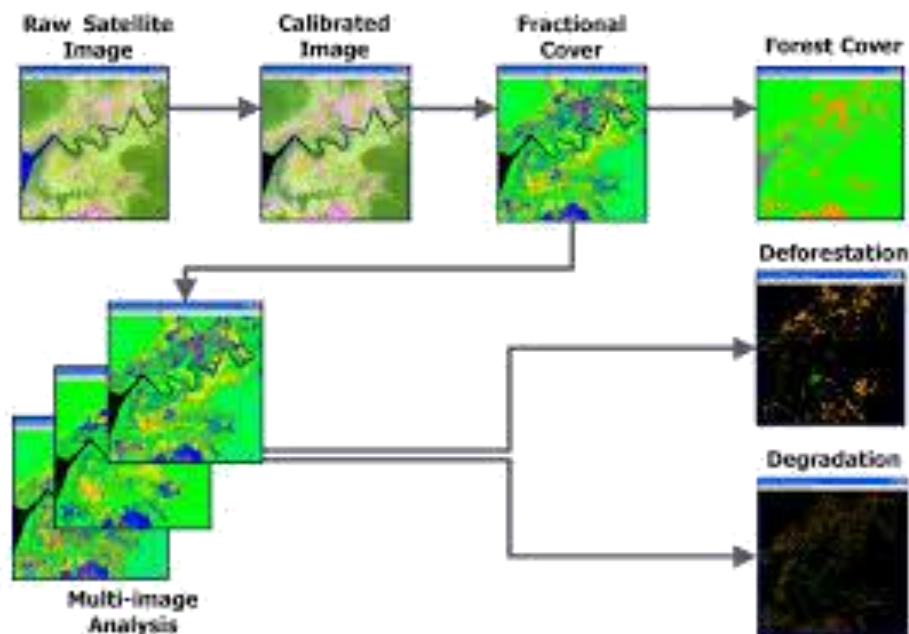
نمونه‌های PEAS

• عامل سیستم تشخیص پزشکی

- معیار کارایی: سلامتی بیمار، کاهش هزینه‌ها
- محیط: بیمار، بیمارستان، کارکنان
- اقدام‌گرها: نمایش پرسش‌ها، آزمایش‌ها، تشخیص‌ها، تجویزها و مراجعه‌ها در صفحه نمایش
- حس‌گرها: وارد کردن علائم بیماری، یافته‌ها و پاسخ‌های بیمار از طریق صفحه کلید



نمونه‌های PEAS



• عامل سیستم آنالیز تصاویر ماهواره‌ای

- معیار کارایی: دسته‌بندی صحیح تصاویر
- محیط: ماهواره و تصاویر آن
- اقدام‌گرها: نمایش دسته‌بندی
- حس‌گرها: آرایه‌هایی از پیکسل‌های رنگی

نمونه‌های PEAS

• عامل ربات جابه‌جا کننده‌ی قطعات

- معیار کارایی: درصد قطعاتی که در ظروف درست قرار گرفته‌اند.
- محیط: کمر بند نقاله همراه با قطعات، ظروف
- اقدام‌گرها: بازو و دست مفصل‌دار
- حس‌گرها: دوربین، حس‌گرهای زاویه مفصل



نمونه‌های PEAS



- عامل **کنترل کننده پالایشگاه**
- معیار کارایی: ماکزیمم کردن خلوص محصول و افزایش ایمنی
- محیط: پالایشگاه، کاربران
- اقدام‌گرها: شیرها، پمپ‌ها، گرماساژها، صفحه‌های نمایش
- حس‌گرها: حس‌گرهای دما و فشار و شیمیایی

• عامل معلم انگلیسی محاوره‌ای

- معیار کارایی: پیشینه کردن نمره امتحان دانش‌آموز
- محیط: مجموعه دانش‌آموزان
- اقدام‌گرها: نمایش تمرین‌ها، پیشنهادات و تصحیحات بر روی صفحه
- حس‌گرها: ورودی صفحه‌کلید



انواع محیط (Fully Observable vs. Partially Observable)

- **کاملاً مشاهده‌پذیر** (در مقابل **نیمه مشاهده‌پذیر**): حس‌گرهای یک عامل، امکان دسترسی به وضعیت کامل محیط در هر لحظه از زمان را به عامل بدهند.

If an agent's sensors give it access to the complete state of the environment at each point in time

- تنها جنبه‌هایی از محیط مهم هستند که در انتخاب عمل عامل دخالت داشته باشند.
- جنبه‌های مهم محیط بر اساس معیار کارایی تعیین می‌شوند.
- کار در این نوع محیط به دلیل عدم نیاز به حالت درونی (internal state) راحت است.
- یک محیط ممکن است به دلیل عدم دقت حس‌گرها و نویز آن‌ها یا به این دلیل که قسمت‌هایی از حالت‌ها در داده‌های حس‌گر حذف می‌شوند نیمه مشاهده‌پذیر شود.
- جاروبرقی با داشتن تنها حس‌گر کثیفی محل
- راننده تاکسی خودکار نمی‌داند که رانندگان دیگر در حال فکر کردن به چه چیزی هستند.
- محیط غیرقابل مشاهده محیطی است که فاقد هرگونه حسگری است.

انواع محیط (Deterministic vs. Stochastic)

- **قطعی** (درمقابل **تصادفی**): وضعیت بعدی محیط به طور کامل براساس وضعیت فعلی و عمل انجام شده مشخص باشد.
If the next state of the environment is completely determined by the current state and the action executed by the agent
- اگر محیط به جز در اثر فعالیت‌های عامل‌های دیگر قطعی باشد گوییم محیط **راهبردی (Strategic)** است.
- در یک محیط تصادفی
 - ممکن است طبیعت اشیای موجود در محیط تصادفی باشد.
 - احتمال‌هایی برای خروجی‌های ممکن در نظر گرفته می‌شود.
 - جزئی قابل مشاهده بودن محیط ممکن است باعث تصادفی شدن محیط شود.
- یک محیط **نامعلوم (uncertain)** است اگر کاملاً مشاهده‌پذیر نباشد یا قطعی نباشد.
- در یک محیط **غیرقطعی (non-deterministic)** برای هر عمل خروجی‌های ممکن مشخص می‌شوند اما احتمالی برای این خروجی‌ها در نظر گرفته نمی‌شود.

انواع محیط (Episodic vs. Sequential)

- **مرحله‌ای یا اپیزودیک** (درمقابل **ترتیبی**): تجربه عامل به اپیزودهای مجزا تقسیم می‌شود. در هر اپیزود، عامل محیط را درک می‌کند، تصمیم می‌گیرد و عمل می‌کند.

In an episodic task environment, the agent's experience is divided into atomic episodes. In each episode the agent receives a percept and then performs a single action

- انتخاب عمل در هر اپیزود فقط از روی دریافت‌ها در همان اپیزود صورت می‌گیرد و ربطی به دریافت‌ها یا تصمیم‌ها در اپیزودهای قبلی ندارد.
- بسیاری از کارهای دسته‌بندی، تعمیر، شناسایی و ... از نوع اپیزودیک هستند.
- برای مثال عاملی که باید قطعات معیوب را روی خط مونتاژ شناسایی کند.
- در محیط‌های ترتیبی، تصمیم فعلی می‌تواند بر روی تصمیمات آینده تاثیر بگذارد.

In sequential environments, the current decision could affect all future decisions

- مانند شطرنج یا راننده تاکسی خودکار
- طراحی عامل برای محیط‌های ساده‌تر است چون نیاز نیست عامل نگران آینده باشد و به آینده فکر کند.

انواع محیط (Single Agent vs. Multi Agent)

- **تک عاملی** (در مقابل **چند عاملی**): تنها خود عامل در محیط عمل می کند.
- عامل جدول کلمات متقاطع تک عاملی و عامل بازی شطرنج چند عاملی است.
- آیا شیء B یک عامل است یا تنها یک شیء در محیط است؟
- زمانی B یک عامل است که برای پیشینه کردن معیار کارایی خود که مقدار آن به رفتار عامل A وابسته است تلاش کند.
- چند عاملی: رقابتی یا مشارکتی (competitive or cooperative)
- رفتار تصادفی و ارتباط می تواند منطقی باشد.

The agent-design problems in multiagent environments are often quite different from those in single-agent environments; for example, **communication** often emerges as a rational behavior in multiagent environments; in some competitive environments, **randomized behavior** is rational because it avoids the pitfalls of predictability

انواع محیط (Static vs. Dynamic)

- **ایستا** (درمقابل **پویا**): در مدتی که عامل در حال تصمیم‌گیری و انجام عمل است محیط تغییر

نمی‌کند. If the environment can change while an agent is deliberating, then we say the environment is dynamic for that agent; otherwise, it is static.

- برای مثال عامل حل‌کننده جدول کلمات متقاطع ایستا و راننده تاکسی خودکار پویا است.

- **نیمه‌پویا (Semi-dynamic)**: اگر محیط پویا نباشد ولی با گذر زمان میزان کارایی تغییر کند گوییم

محیط نیمه‌پویا است. If the environment itself does not change with the passage of time but the agent's performance score does, then we say the environment is **semi-dynamic**.

- شطرنج زمان‌دار

- طراحی عامل برای محیط‌های ایستا ساده‌تر است زیرا لازم نیست عامل درحین تصمیم‌گیری، محیط اطرافش را درک کند و لازم نیست نگران گذر زمان باشد.

انواع محیط (Discrete vs. Continuous)

- **گسسته** (درمقابل پیوسته): حالت‌های محیط، دریافت‌های عامل از محیط، فعالیت‌های عامل و نحوه مدیریت زمان کمیت‌های گسسته و مجزا باشند.

The discrete/continuous distinction applies to the *state* of the environment, to the way *time* is handled, and to the *percepts* and *actions* of the agent.

- برای مثال شطرنج دارای تعداد متناهی از حالات گسسته و مجموعه گسسته از ادراکات و اعمال است.
- برای مثال راننده تاکسی دارای مجموعه‌ای از حالات، اعمال و زمان پیوسته است.
- ورودی دوربین‌های دیجیتال گسسته است اما در اغلب موارد به صورت تغییرات پیوسته شدت روشنایی و رنگ‌ها در نظر گرفته می‌شود.

انواع محیط (Known vs. Unknown)

- **شناخته شده** (درمقابل **شناخته نشده**): نتایج برای همه اعمال مشخص باشد. یا اگر محیط تصادفی است احتمالات حاصل از اعمال مشخص باشد.

In a known environment, the outcomes (or outcome probabilities if the environment is stochastic) for all actions are given.

- یک ویژگی از محیط نیست.
- مرتبط با وضعیت دانش عامل (یا طراح) در مورد «قوانین فیزیک» محیط است.
- اگر محیط شناخته شده نباشد، عامل برای اتخاذ تصمیمات مناسب باید یاد بگیرد محیط چگونه کار می کند.
if the environment is unknown, the agent will have to learn how it works in order to make good decisions.
- قابل مشاهده بودن یا نبودن محیط ارتباطی با شناخته شده بودن یا نبودن آن ندارد.
- بازی solitaire card دارای یک محیط شناخته شده و نیمه مشاهده پذیر است.
- یک بازی جدید دارای یک محیط ناشناخته و مشاهده پذیر است.

نمونه‌هایی از 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

کدام یک از عبارات زیر صحیح است؟ (کامپیوتر ۸۵)

(۱) همه محیط‌های نیمه قابل مشاهده، غیرقطعی هستند.

✓ (۲) عاملی که به زبان طبیعی محاوره کند در یک محیط نیمه قابل مشاهده عمل می‌کند.

(۳) هر عاملی که فقط بخشی از محیط را حس (دریافت) می‌کند نمی‌تواند عقلایی (rational) باشد.

(۴) عاملی که در محیط کاملاً قابل مشاهده عمل می‌کند نیازی به حالات درونی (internal state) ندارد.

(مهندسی کامپیوتر آزاد ۹۳)

محیط کدام یک از موارد زیر پویا است؟

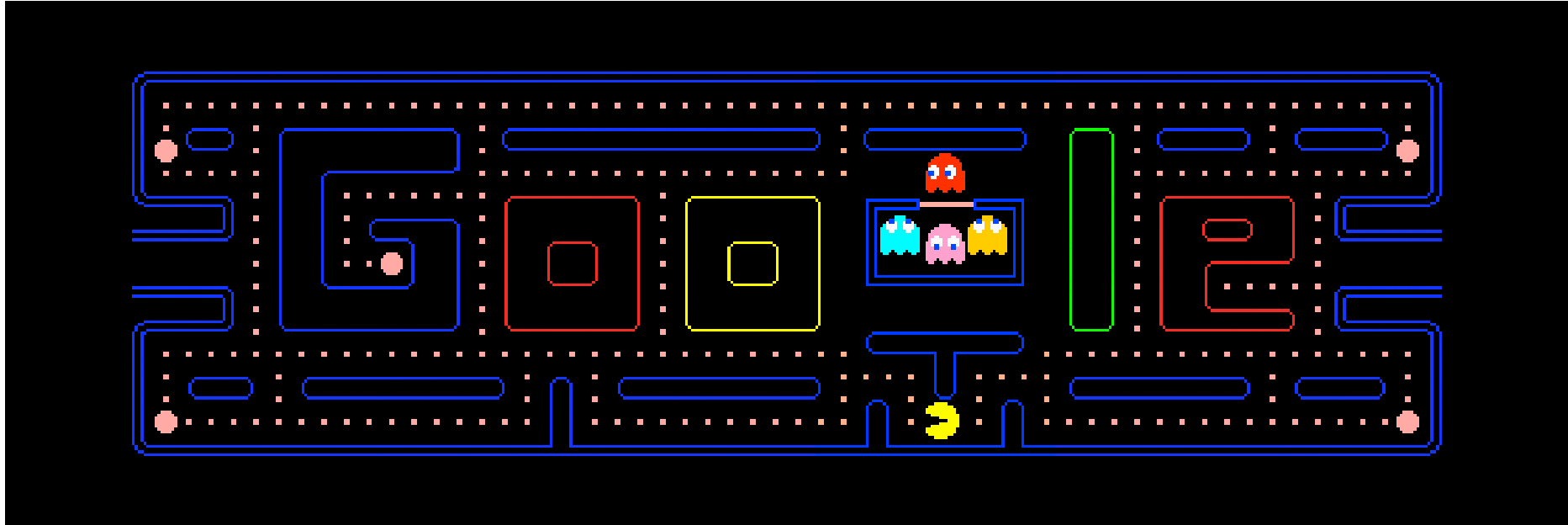
(۱) تخته‌نرد

(۲) شطرنج

(۳) رانندگی تاکسی ✓

(۴) پوکر

یادآوری محیط وظیفه



- تک عاملی؟
- گسسته؟
- مرحله‌ای؟

- مشاهده‌پذیر؟
- قطعی؟
- ایستا؟

ساختار عامل‌ها

نحوه کار درونی عامل چگونه است؟

- وظیفه هوش مصنوعی، طراحی برنامه عامل (Agent program) است که تابع عامل (Agent function) را که ادراک را به اعمال، نگاشت می کند، پیاده سازی کند.

$$\text{عامل} = \text{معماری} + \text{برنامه}$$

- ساختار برنامه عامل
 - ورودی = تنها ادراک فعلی (زیرا هیچ چیز دیگری از محیط قابل دسترسی نیست).
 - خروجی = عمل
 - برنامه = روندی که با استفاده از ورودی، خروجی را تولید می کند.
 - به تفاوت برنامه عامل با تابع عامل توجه کنید.
 - در صورتی که اعمال عامل، به کل رشته ادراک وابسته باشد عامل باید ادراک را به یاد داشته باشد.
(حالت درونی یا Internal State)

- عامل جدول گرا (*Table-driven agent*)
- انواع پایه‌ای 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 مبتنی بر یادگیری (*Learning-based agents*)

عامل جدول گرا (Table-driven agent)

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 \leftarrow 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 گرا (ادامه)

- مزایا
 - ساده برای پیاده سازی
 - معایب
 - هیچ عامل فیزیکی در این دنیا فضای ذخیره سازی این جدول را ندارد.
 - $P = \text{مجموعه ادراک ممکن}$ ، $T = \text{طول عمر}$
$$\sum_{t=1}^T |P|^t$$
 - طراح وقت کافی برای ساختن جدول ندارد.
 - عامل نمی تواند اعمال مناسب را از طریق تجربیاتش یاد بگیرد.
 - طراح برای چگونه پر کردن سطرهای جدول هیچ راهنمایی ندارد.

برنامه عامل (Agent program)

- نداشت لزوماً با استفاده از جدول نیست.
- چالش کلیدی هوش مصنوعی، نوشتن برنامه‌هایی برای تولید رفتار عقلانی است که تا حد امکان براساس یک برنامه‌ی کوچک به جای یک جدول عظیم باشد.

```
function Sqrt( double X )
```

```
    double r = 1.0 ;
```

```
    while ( fabs( r * r -x ) > 10e-8)
```

```
        r = r -( r * r -x ) / 2r ;
```

```
    return r ;
```

Huge tables of square roots before 1970s

1.0	1.00000
1.1	1.04898
1.2	1.09565
1.3	1.14056

...

عامل‌های واکنشی ساده (Simple reflex agents)

- مثال دنیای جاروبرقی

function REFLEX-VACUUM-AGENT (*[location, status]*) **return** an action

if *status* == *Dirty* **then return** *Suck*

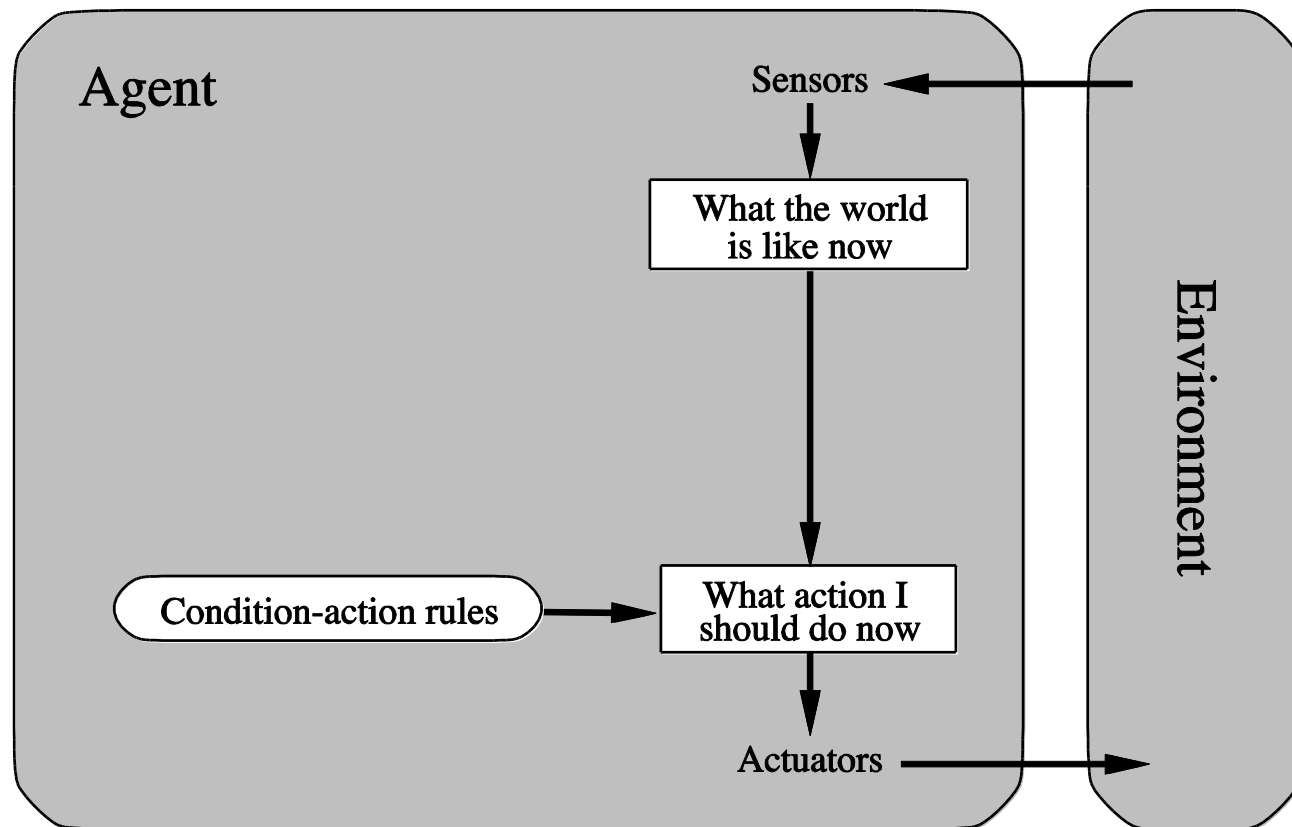
else if *location* == *A* **then return** *Right*

else if *location* == *B* **then return** *Left*

- کاهش تعداد امکانات ممکن به دلیل نادیده گرفتن تاریخچه ادراک
- از 4^T به 4

عامل‌های واکنشی ساده (Simple reflex agents)

- انتخاب عمل تنها براساس ادراک فعلی
- تاریخچه ادراک تاکنون را نادیده می‌گیرد.
- کاهش چشم‌گیر در وضعیت‌های ادراک/عمل ممکن
- پیاده‌سازی شده از طریق قوانین شرط-عمل
- اگر خودروی جلویی درحال ترمز کردن است
آن‌گاه شروع به ترمز گرفتن کن.
- واکنش‌های ذاتی مانند وقتی که چیزی به چشم نزدیک می‌شود.

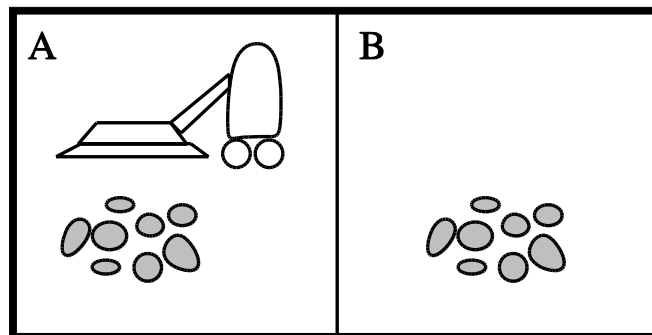


عامل‌های واکنشی ساده (Simple reflex agents)

```
function SIMPLE-REFLEX-AGENT(percept) returns an action
  persistent: rules, a set of condition-action rules
  state ← INTERPRET-INPUT(percept)
  rule ← RULE-MATCH(state, rule)
  action ← rule.ACTION
  return action
```

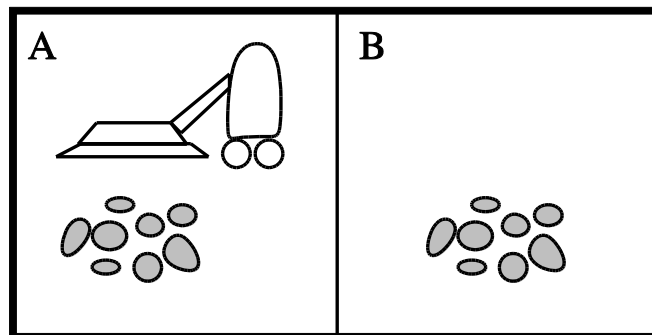
- ساده ولی دارای سطح هوشمندی بسیار پایین. چرا؟
- تنها در صورتی درست کار می‌کند که براساس ادراک فعلی بتوان درست تصمیم گرفت و این بدان معناست که محیط باید کاملاً مشاهده‌پذیر باشد.
- حلقه‌های نامتناهی در محیط‌های نیمه‌مشاهده‌پذیر
- اگر عامل امکان تصادفی عمل کردن داشته باشد گریز از حلقه‌های بی‌نهایت امکان‌پذیر می‌شود.

عامل‌های واکنشی ساده (Simple reflex agents)



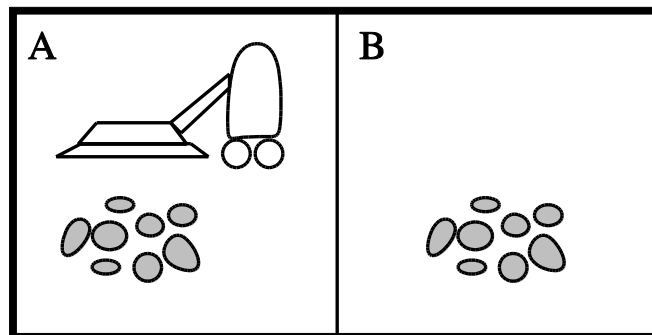
- ادراکات: [مکان، وضعیت تمیزی/کثیفی خانه A، وضعیت تمیزی/کثیفی خانه B]
- مانند: [A, Dirty, Dirty]
- اقدام‌ها: برو به چپ، برو به راست، تمیز کن، هیچ کاری نکن
- معیار کارایی: امتیاز مثبت به هر مربع تمیز در هر گام زمانی و امتیاز منفی برای انجام هر یک از اعمال

عامل‌های واکنشی ساده (Simple reflex agents)



- ادراکات: [مکان، وضعیت تمیزی/کثیفی]
- مانند: [A, Dirty]
- اقدام‌ها: برو به چپ، برو به راست، تمیز کن، هیچ کاری نکن
- معیار کارایی: امتیاز مثبت به هر مربع تمیز در هر گام زمانی و امتیاز منفی برای انجام هر یک از اعمال

عامل‌های واکنشی ساده (Simple reflex agents)



- ادراکات: [مکان، وضعیت تمیزی/کثیفی]
- مانند: [A, Dirty]
- اقدام‌ها: برو به چپ، برو به راست، تمیز کن، هیچ کاری نکن
- معیار کارایی: امتیاز مثبت به هر مربع تمیز در هر گام زمانی

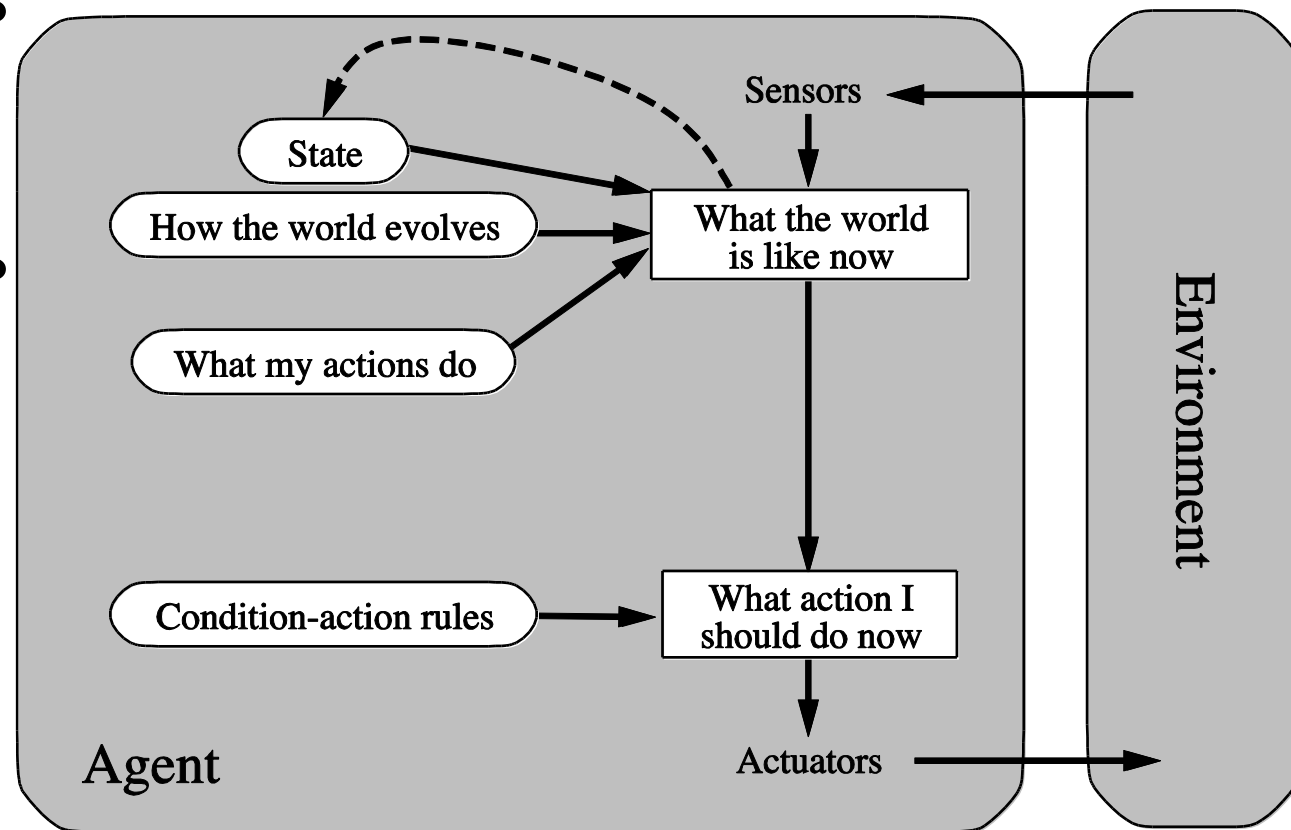
عامل‌های واکنشی مبتنی بر مدل (*Model-based reflex agents*)

- روش برخورد عامل با نیمه‌مشاهده‌پذیری
- نگهداری سوابق آن بخش از دنیا که اکنون نمی‌تواند ببیند. (حالت داخلی *Internal state*)
- به‌روزرسانی حالت درونی نیاز به دو نوع دانش دارد:

- محیط به مرور زمان بدون دخالت عامل چگونه تغییر می‌کند.
- هر عملی که عامل انجام می‌دهد باعث چه تغییر وضعیتی در محیط می‌شود.

← **مدل دنیا (Model of the world)**

- تنها بهترین حدس در مورد وضعیت فعلی یک محیط نیمه‌مشاهده‌پذیر را تعیین می‌کند.



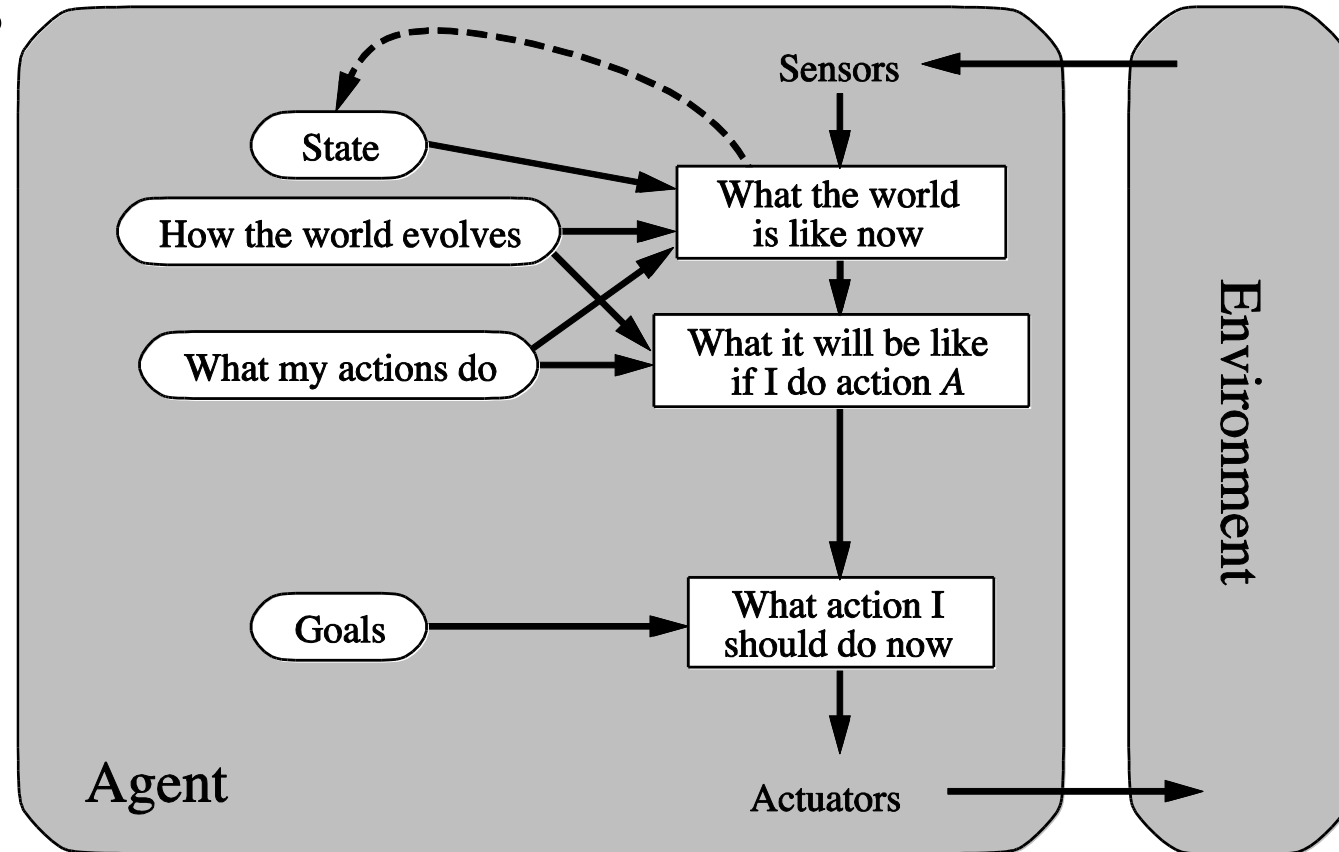
عامل‌های واکنشی مبتنی بر مدل (Model-based reflex agents)

function REFLEX-AGENT-WITH-STATE(*percept*) **returns** an action
persistent: *rules*, a set of condition-action rules
state, the agent's current conception of the world state
model, a description of how the next state depends on current state and action
action, the most recent action.

state \leftarrow UPDATE-STATE(*state*, *action*, *percept*, *model*)
rule \leftarrow RULE-MATCH(*state*, *rules*)
action \leftarrow *rule*.ACTION
return *action*

عامل‌های مبتنی بر هدف (*Goal-based agents*)

- علاوه بر توصیف وضعیت فعلی عامل به نوعی اطلاعات درباره هدف که وضعیت مطلوب را توصیف کند نیاز دارد.
- هنگامی که عامل باید رشته اعمال طولانی و متعددی را در نظر بگیرد تا راهی برای رسیدن به هدف بیابد، انتخاب عمل مبتنی بر هدف دشوارتر خواهد بود.
- زیرشاخه‌های جستجو و برنامه‌ریزی از هوش مصنوعی به بررسی این موضوع پرداخته‌اند.

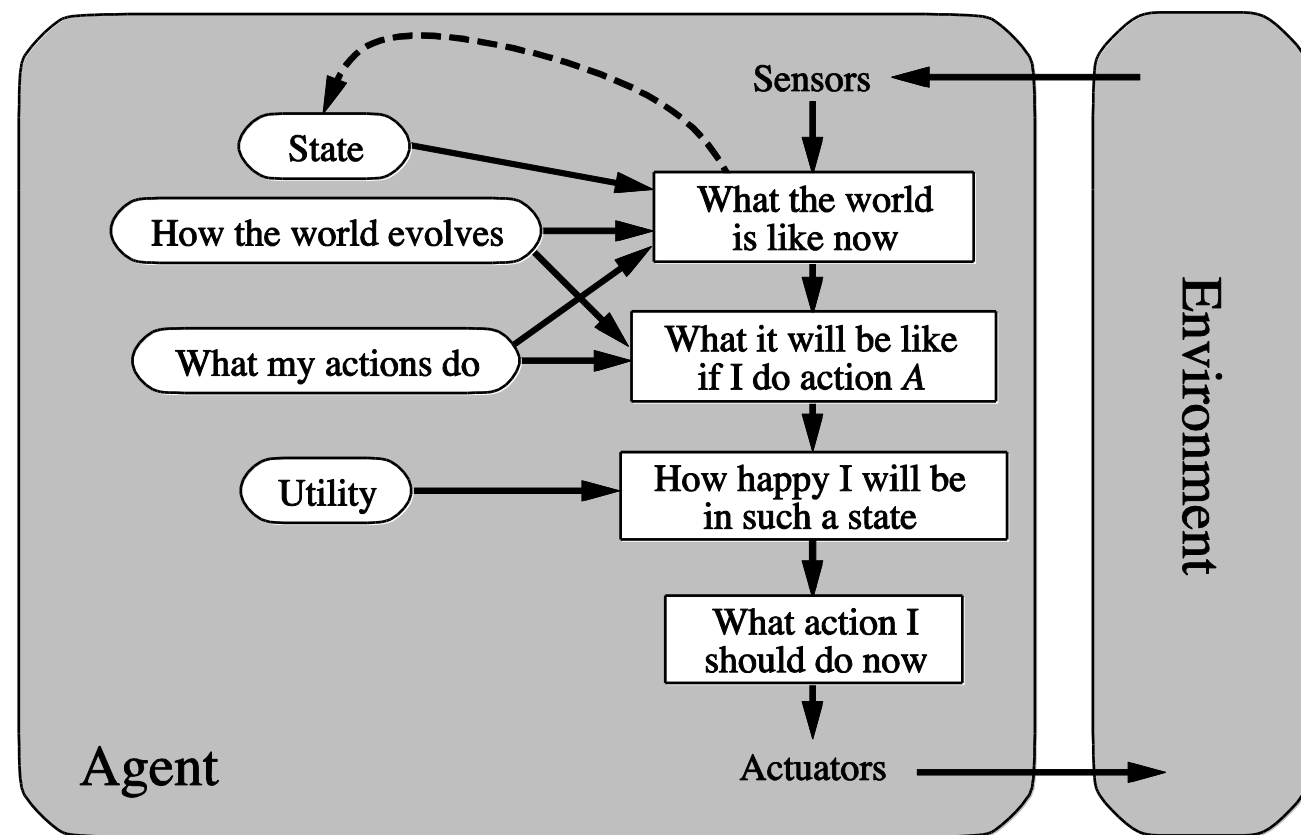


عامل‌های مبتنی بر هدف در مقابل عامل‌های واکنشی

- عامل‌های مبتنی بر هدف آینده را در نظر می‌گیرند.
- در طراحی عامل‌های واکنشی دو سوال «اگر فلان کار را کنم چه اتفاقی خواهد افتاد؟» و «چه چیزی مرا خوشنود می‌کند؟» به‌طور صریح داده نمی‌شود زیرا قواعد داخلی مستقیماً از ادراک به اعمال نگاشت می‌کنند.
- طراح عامل از قبل عمل صحیح را برای موارد مختلف محاسبه کرده است.
- عامل‌های مبتنی بر هدف کارایی کم‌تر ولی انعطاف‌پذیری بیشتری دارند.
- دانشی که از تصمیماتش پشتیبانی می‌کند به‌طور صریح ارائه شده است و می‌تواند تغییر کند.
- مثال: با تغییر سطح زمین، در عامل مبتنی بر هدف تنها کافی است نحوه عمل ترمزها عوض شود اما بخش بزرگی از قوانین عامل واکنشی باید بازنگری شود.
- در عامل‌های واکنشی با تغییر هدف عامل باید تمام قوانین شرط – عمل بازنویسی شوند ولی عامل هدف‌گرا نسبت به هدف‌های جدید انعطاف‌پذیرتر است.

عامل‌های مبتنی بر سودمندی (Utility-based agents)

- در اکثر محیط‌ها، اهداف به‌تنهایی برای تولید رفتاری با کیفیت بالا کافی نیستند.
- از بین مسیرهای مختلف برای رسیدن به هدف، عامل کدام مسیر را باید در پی گیرد؟
- مسیری که سودمندتر باشد.
- هر وضعیت چقدر عامل را خوشنود می‌سازد؟



عوامل های مبتنی بر سودمندی (Utility-based agents)

- تابع سودمندی (utility function)

- تابع سودمندی به هر وضعیت محیط (یا هر دنباله ای از وضعیت ها) یک عدد حقیقی به عنوان سودمندی نسبت می دهد که این عدد میزان رضایت عامل از آن وضعیت است.
- روشی برای مقایسه میزان کارایی روش های مختلف رسیدن به هدف
- نوعی پیاده سازی داخلی از معیار کارایی

- نکته:

- یکسان بودن تابع سودمندی و معیار کارایی باعث می شود که تلاش برای بیشینه سازی تابع سودمندی به رفتار عقلانی عامل بیانجامد.
- عقلانی رفتار کردن عامل نیازمند یکسان بودن تابع سودمندی و معیار کارایی و یا حتی مبتنی بر سودمندی بودن عامل نیست.

عامل‌های مبتنی بر سودمندی (Utility-based agents)

- مزایا
 - مانند عامل‌های مبتنی بر هدف مزایای یادگیری و انعطاف‌پذیری را دارد.
 - در شرایطی که عامل **چند هدف متضاد** داشته باشد (مانند سرعت و امنیت در رانندگی) یا در شرایطی که **چندین هدف وجود دارد و عامل نمی‌تواند با اطمینان مشخص کند که باید به کدام دست یابد**، عامل هدف‌گرا با مشکل روبه‌رو است ولی عامل مبتنی بر سودمندی می‌تواند توازن مناسبی بین اهداف مختلف برقرار کند.
 - موفقیت عامل براساس اهمیت اهداف وزن‌دهی می‌شود.
- عامل عقلانی مبتنی بر سود در محیط‌های نامطمئن به گونه‌ای عمل می‌کند که سودمندی مورد انتظار نتایج عمل را بیشینه کند.
- در صورتی که تابع سودمندی عامل به شکل صریح بیان شده باشد می‌توان از الگوریتم‌های عام منظوره برای یافتن عمل بهینه استفاده کرد.

عوامل های مبتنی بر یادگیری (Learning-based agents)

• چهار مولفه یک عامل یادگیری

• **عنصر اجرایی (performance element):**

مسئول انتخاب فعالیت های بیرونی (بدنه هر یک از عامل های قبلی)

• **منتقد (ناظر) (critic):** تعیین میزان موفقیت

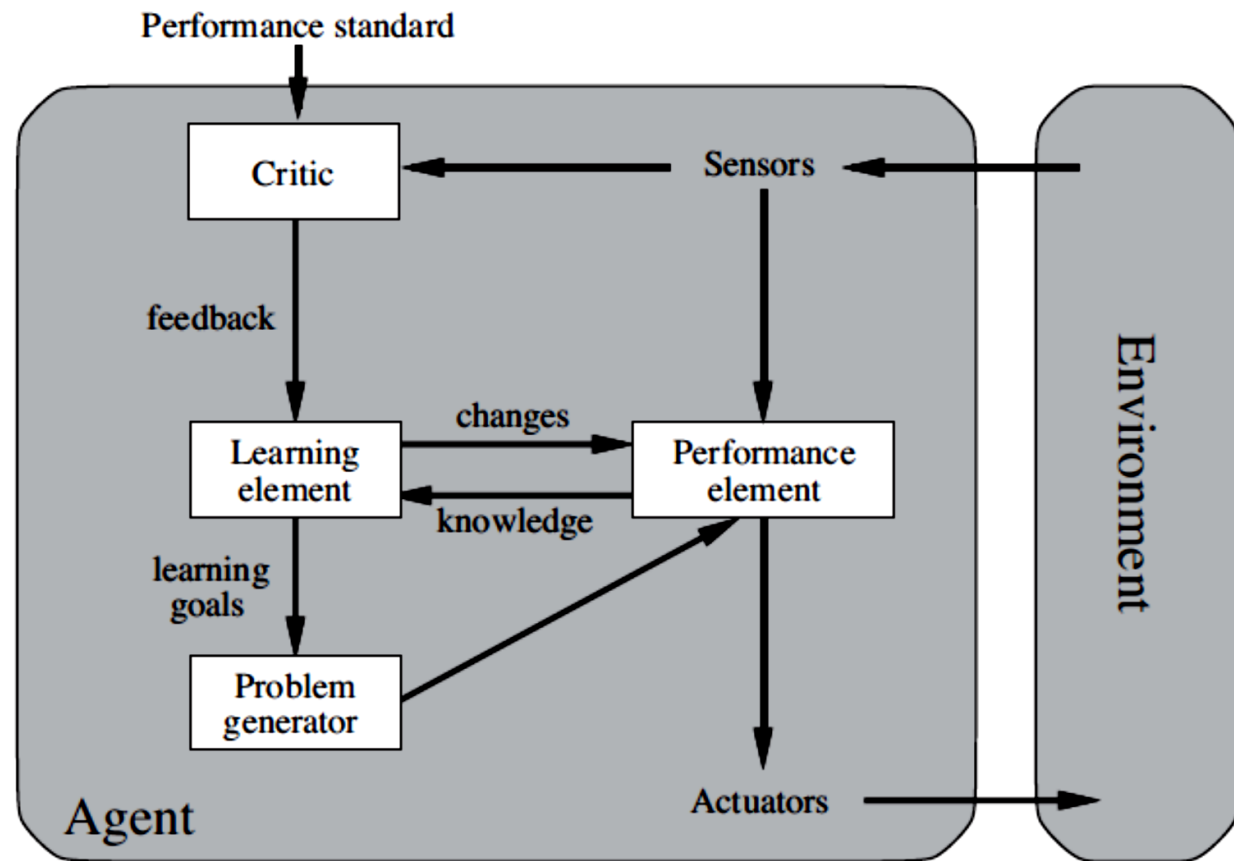
عامل با استفاده از استاندارد کارایی ثابت و دریافت ادراک ها

• **عنصر یادگیری (learning element):** تغییر

عنصر اجرایی با استفاده از بازخورد منتقد به نحوی که در آینده بهتر عمل کند.

• **مولد مسئله:** پیشنهاد اقداماتی جهت یافتن

تجربیات تازه، راه کاری برای اکتشاف محیط



کدام یک از جملات زیر صحیح است؟ (کامپیوتر ۸۶)

(۱) ممکن است agent function وجود داشته باشد که نتوان آن را با هیچ agent program پیاده‌سازی نمود.

(۲) یک عامل مبتنی بر دانش (knowledge base) را نمی‌توان با کمک معماری انعکاسی ساده ساخت.

(۳) عامل مبتنی بر مدل (Model based) برای محیط‌های با حالات و اعمال پیوسته مناسب نیست.

(۴) در محیط‌های کاملاً قابل مشاهده دلیلی برای داشتن حالات داخلی (internal states) نیست.

اگر هدف رساندن یک مسافر از مبدأ به مقصد توسط یک عامل هوشمند باشد و معیار کارایی امنیت، زمان و مسیر خلوت باشد چه نوع عاملی مناسب است؟
(فناوری اطلاعات ۹۳)

(۱) هدف‌گرا

(۲) مدل‌گرا

(۳) سودمندی ✓

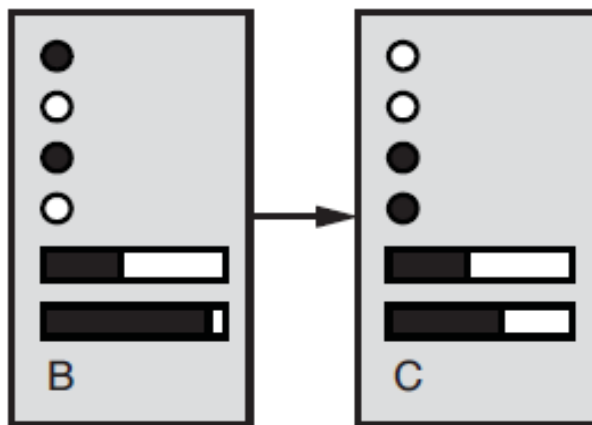
(۴) واکنشی ساده

روش‌های ارائه حالت

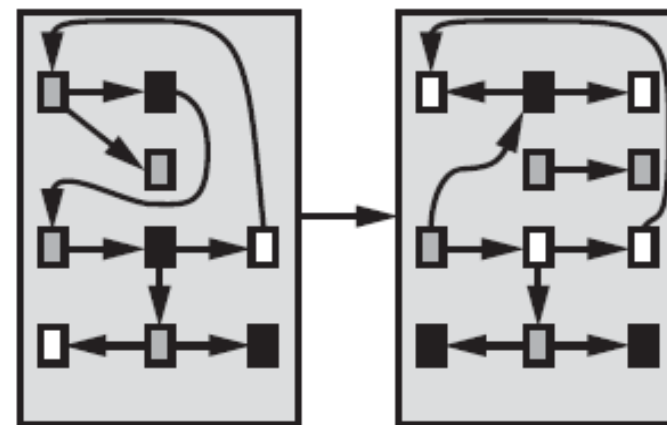
- اتمیک: هر حالت دنیا غیرقابل تقسیم است. (توصیف دنیا با استفاده از یک اسم یا یک ویژگی)
- فاکتور شده: هر حالت به ویژگی‌های مختلف تقسیم می‌شود.
- ساخت‌یافته: روابط بین اشیاء مختلف در هر حالت باید مشخص باشد.



Atomic



Factored



Structured