فصل اول

«هوش مصنوعی و عامل‌های هوشمند»

مقدمه‌ای بر هوش مصنوعی

حوزه هوش مصنوعی (Artificial Intelligence) یا به اختصار AI علم جدیدی است که پس از جنگ جهانی دوم مطرح شد. اصطلاح هوش مصنوعی در سال ۱۹۵۶ به وجود آمد. هوش مصنوعی دو هدف عمده دارد: ۱ـ درک چگونگی تفکر انسان ۲ـ ساخت موجودیت‌های هوشمند

در حال حاضر AI زیر شاخه‌های وسیعی از موضوعات عمومی مانند ادراک و استدلال منطقی و کارهای خاص مانند بازی شطرنج، اثبات قضایای ریاضی، سرودن شعر و تشخیص امراض را شامل می‌شود.

AI چیست؟

تعاریف هوش مصنوعی در دو بعد اصلی متمایز می‌شوند که در جدول زیر بیان شده‌اند: تعاریف بالای جدول به پردازش‌های فکری و استدلالی پرداخته و تعاریف پایینی بر پردازش‌های رفتاری متمرکز شده‌اند. همچنین تعاریف سمت چپ با نگرش انسان‌گونه و تعاریفی که در سمت راست هستند با نگرش منطقی (عقلانی) از هوشمندی مرتبط هستند.

تعاریف هوش مصنوعی

سیستم‌هایی که منطقی فکر می‌کنند	سیستم‌هایی که مثل انسان فکر می‌کنند
«مطالعهٔ قابلیت‌های ذهنی از طریق مدل‌های محاسباتی» (چارنیاک و مک درموت، ۱۹۸۵) «مطالعه محاسباتی که امکان مشاهده، استدلال و عمل را فراهم می‌نماید» (نیلسون، ۱۹۹۲)	«تلاش برای ساخت کامپیوترهایی که فکر می‌کنند، ماشین‌هایی با قدرت تفکر و حس کامل» (هاولگند، ۱۹۸۵) «[خودکارسازی] فعالیت‌هایی که با تفکر انسان مرتبط هستند از قبیل تصمیم‌گیری، حل مسأله و یادگیری» (بلمن، ۱۹۷۸)
سیستم‌هایی که منطقی عمل می‌کنند	**سیستم‌هایی که مثل انسان عمل می‌کنند**
«هوش مصنوعی شامل مطالعه عامل‌های هوشمند است» (بل اتال، ۱۹۹۸) «هوش مصنوعی با رفتارهای هوشمند در ساخته‌های انسانی مرتبط است» (نیلسون، ۱۹۹۸)	«هنر ایجاد ماشین‌هایی که اعمالی را انجام دهند که انسان برای انجام آن‌ها نیاز به هوشمندی دارد» (کروزیل، ۱۹۹۰) «مطالعه چگونگی ساخت کامپیوترهایی که کارهایی را انجام دهند که در حال حاضر انسان، بهتر انجام می‌دهد» (ریچ ونایت، ۱۹۹۱)

درواقع نگرش انسان‌گونه شامل فرضیات و تجربیات می‌باشد، در حالی‌که نگرش منطقی، ترکیبی از ریاضیات و مهندسی است.

تفکر انسان گونه (Thinking Humanly)

انسانی فکر کردن، یعنی ماشین قادر به شناختن پیرامون خود همانند انسان باشد.

رسیدن به این هدف از دو طریق امکان‌پذیر است:

۱ـ از طریق درون‌گرایی: سعی در به‌دست آوردن افکار خودمان

۲ـ از طریق تجارب روانشناسی: روانشناسان سعی دارند شرایط و عواملی را که باعث می‌شود افراد مرتکب اعمال خاصی شوند کشف کنند؛ زیرا رابطهٔ بین عمل و عامل می‌تواند نشان‌دهندهٔ نحوه‌ی تفکر انسان باشد.

اگر قادر به ایجاد تئوری دقیقی درباره ذهن باشیم، آنگاه قادر خواهیم بود این تئوری را به برنامه کامپیوتری تبدیل کنیم.

ناول و سایمون (Newell and Simon) در سال ۱۹۶۱ برنامه General Problem Solver (GPS) را معرفی کردند؛ اما آن‌ها تنها به این که برنامه‌شان به درستی مسائل را حل کند راضی نبودند، بلکه آن‌ها بیشتر سعی داشتند تا در حل یک مسأله خاص، مراحل استدلال برنامه خود را با سیر تحلیل و استدلال انسان مقایسه کنند.

عملکرد انسان‌گونه (Acting Humanly)

سیستم و ماشین‌هایی که مانند انسان عمل می‌کنند در این دسته قرار می‌گیرند. تست تورینگ که توسط آلن تورینگ در سال ۱۹۵۰ پیشنهاد شد، شامل آزمونی است که در آن یک فرد به‌عنوان آزمون‌گیرنده در یک اتاق نشسته و با یک سیستم که در اتاق دیگر قرار دارد در ارتباط است. فرد آزمون‌گیرنده از سیستم سؤالاتی را می‌پرسد و سیستم به آن‌ها پاسخ می‌دهد. حال اگر فرد آزمون‌گیرنده نتواند تشخیص دهد که با یک انسان مکالمه کرده یا یک سیستم کامپیوتری، می‌توان گفت که سیستم کامپیوتری مثل انسان عمل کرده است.

کامپیوتری که در این آزمون شرکت می‌کند باید قادر به پردازش‌های زیر باشد:

۱ـ پردازش زبان طبیعی (natural language processing): تا قادر به محاوره گردد.

۲ـ بازنمایی دانش (knowledge representation): تا اطلاعات تولید شده قبل یا در حین آزمون را ذخیره کند.

۳ـ استدلال خودکار (automated reasoning): تا از اطلاعات ذخیره شده برای پاسخ به پرسش‌ها، استفاده کرده و نتایج جدیدی را استخراج کند.

۴ـ یادگیری ماشینی (machine learning): تا با محیط و شرایط جدید سازگاری داشته باشد.

تست تورینگ از تعامل فیزیکی مستقیم بین کامپیوتر و شخص مصاحبه‌کننده اجتناب می‌کند؛ زیرا شبیه‌سازی فیزیکی فرد برای هوشمندی ضروری نیست. تست کامل تورینگ شامل یک سیگنال ویدیویی است که مصاحبه‌کننده می‌تواند از طریق آن توانایی‌های ادراکی اشیاء را مورد آزمون قرار دهد. به‌عبارت دیگر اعمال فیزیکی روبات و انسان برای مصاحبه‌کننده به‌صورت یک سیگنال ویدیویی نمایش داده می‌شود. درواقع مشخصات فیزیکی آن‌ها برای مصاحبه‌کننده مشخص نیست، بلکه فقط عملکرد آن‌ها در رابطه با اشیاء مورد بررسی قرار می‌گیرد. برای قبول شدن در تست کامل تورینگ، کامپیوتر به موارد زیر احتیاج دارد:

۱ـ بینایی ماشین (Computer Vision) برای درک محیط و اشیاء اطراف خود؛

۲ـ روباتیک (Robotics) به‌منظور جابه‌جایی اشیاء و دستکاری آن‌ها.

تفکر منطقی (Thinking Rationally)

ارسطو یکی از اولین کسانی بود که تلاش کرد رمز «تفکر درست» را بگشاید؛ یعنی می‌توان فرایندهای استدلال همیشه درست را به‌صورت مجموعه‌ای از دستورالعمل‌ها و قوانین تبدیل کرد. قیاس معروف وی «از مقدمه‌های درست، نتایج درست حاصل می‌شود»، الگویی برای ساختار توافقی ایجاد کرد. برای مثال «سقراط انسان است، تمام انسان‌ها می‌میرند، پس سقراط خواهد مرد»، این تفکر عملیات ذهن را سازمان می‌دهد و حیطه منطق را پایه‌گذاری می‌کند. دو اشکال عمده در این روش وجود دارد. اول این‌که، دریافت دانش غیررسمی و تبدیل‌کردن آن به‌صورت علائم منطقی آسان نیست و دوم این‌که، تفاوت زیادی بین قابل حل بودن مسأله‌ای در تئوری و انجام آن در عمل وجود دارد. هوشمند بودن همان‌قدر که نیازمند استدلال است، به کنش نیاز دارد. ارسطو معتقد بود که کنش‌ها با یک رابطه‌ی منطقی بین اهداف و اطلاع از نتیجه کنش توجیه می‌شود.

➢ **توجه:** در بعضی منابع Rationally را **عقلانی** ترجمه کرده‌اند. پس منظور از عقلانی بودن تفکر یا عملکرد در سایر منابع همان منطقی بودن تفکر یا عملکرد است.

عملکرد منطقی (Acting Rationally)

یک عامل درواقع چیزی است که ابتدا مشاهده و سپس عمل می‌کند. عامل‌های کامپیوتری ویژگی‌هایی دارند که آن‌ها را از یک برنامه متمایز می‌کند. این ویژگی‌ها عبارتند از: عمل‌کردن به‌صورت خودکار، ایستادگی در مدت زمان طولانی، وفق پیدا کردن با تغییرات و قابلیت برآورده کردن اهداف جدید.

در تفکر منطقی تأکید عمده بر روی استنتاج‌های صحیح بوده است. تولید استنتاج‌های صحیح گاهی قسمتی از وجود یک عامل منطقی است؛ به‌عبارت دیگر عملکرد منطقی فقط شامل استنتاج درست نیست، زیرا در بعضی از موقعیت‌ها تصمیم صحیح اثبات شده‌ای برای انجام وجود ندارد، اما عامل باز هم باید عملی را انجام دهد. حتی در بعضی موارد نمی‌توان براساس استنتاج تصمیم‌گیری کرد، مثلاً عقب کشیدن دست (بدون تفکر) از یک اجاق داغ؛ عکس‌العملی است که موفق‌تر از یک عکس‌العمل آهسته است که با استدلال و بررسی انجام می‌شود.

مثال۱: تعریف زیر از هوش مصنوعی جزو کدام دسته از تعاریف هوش مصنوعی است؟

«شاخه‌ای از علوم کامپیوتر که با اتوماسیون رفتار هوشمند مربوط می‌شود»

۱) سیستم‌هایی که به‌طور منطقی فکر می‌کنند. ۲) سیستم‌هایی که مانند انسان فکر می‌کنند.

۳) سیستم‌هایی که به‌طور منطقی عمل می‌کنند. ۴) سیستم‌هایی که مانند انسان عمل می‌کنند.

☑ **پاسخ:** گزینه «۳» خودکارسازی رفتار هوشمند منجر به تولید عملکردی منطقی خواهد شد.

مثال۲: کدام‌یک از جملات زیر صحیح است؟

۱) با استفاده از تست تورینگ می‌توان انسان نبودن هر عامل غیرانسانی را تعیین نمود.

۲) آلن تورینگ قادر نیست مراحل تست تورینگ را با موفقیت طی کند.

۳) عاملی که توانسته تمامی انسان‌ها را در یک بازی شکست دهد قطعاً توسط تست تورینگ قبول خواهد شد.

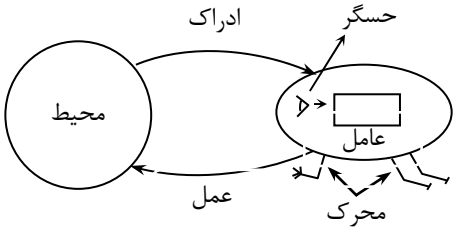
۴) عاملی که توانسته تمامی انسان‌ها را در یک بازی شکست دهد می‌تواند بدون توجه به نحوه‌ی تفکر انسان طراحی شده باشد.

☑ **پاسخ:** گزینه «۴» موفقیت در تست تورینگ صورت می‌گیرد که عامل انسانی، قادر به تشخیص این که فرد پاسخ‌دهنده انسان است یا ماشین نباشد. مسلماً عامل پس از مشاهده برد در بازی در مقابل هر انسانی، می‌تواند مغایر با نحوه تفکر و عملکرد انسان عمل کرده باشد. بنابراین می‌توان گفت که طراحی چنین عاملی ارتباطی به نحوه‌ی تفکر انسان نمی‌تواند داشته باشد.

عامل‌های هوشمند

هر چیزی که بتواند محیط پیرامونش را از طریق **حسگرها** (Sensor) درک نموده و از طریق محرک‌ها یا اثرکننده‌ها (Actuators) روی محیط، **عملی** (Action) انجام دهد، **عامل** (Agent) نامیده می‌شود.

در یک عامل انسانی چشم‌ها، گوش‌ها و سایر ارگان‌ها به‌عنوان حسگرها و دست‌ها، پاها، دهان و دیگر قسمت‌های بدن به‌عنوان اثرکننده‌ها محسوب می‌شوند. اما در یک عامل رباتیکی، دوربین‌ها و یابنده‌های مادون قرمز را به‌عنوان حسگرها و انواع موتورها را به‌عنوان اثرکننده‌ها در نظر می‌گیرند.



شمایی از یک عامل

شکل مقابل شمایی از یک عامل عمومی را نشان می‌دهد.

هدف ما در این کتاب طراحی عامل‌هایی است که کار مناسبی را بر روی محیط انجام می‌دهند. ابتدا در مورد معنی کار مناسب کمی دقیق‌تر خواهیم شد، سپس درباره‌ی انواع طراحی‌ها برای عامل‌های موفق گفتگو خواهیم کرد.

الف ــ عامل‌ها چگونه باید عمل کنند

یک عامل **منطقی** (Rational Agent) به عاملی گفته می‌شود که بتواند براساس هر دنباله ادراکی که از محیط پیرامون خود دریافت کرده است، رشته‌ای از اعمال را انجام دهد، تا بر اساس آن **کارایی** خود را افزایش دهد. معیار کارایی (performance measure)، میزان موفقیت عامل را ارزیابی می‌کند و بدیهی است که میزان یکسانی برای ارزیابی همه عامل‌ها وجود ندارد. بنابراین ما می‌توانیم به‌عنوان ناظران خارجی، یک استاندارد از آنچه به معنی موفقیت در محیط مطرح است را تعریف کنیم و از آن برای محاسبه کارایی عامل‌ها استفاده کنیم.

به‌عنوان مثال یک معیار کارایی برای عاملی مانند جاروبرقی می‌تواند مقدار خاک تمیز شده در بازه زمانی ۸ ساعته باشد. مقدار زمان و برق مصرفی نیز برای عامل می‌تواند به‌عنوان بخشی از معیار کارایی باشد، اما برای عامل راننده تاکسی، امنیت در حین رانندگی را می‌توان به‌عنوان معیار کارایی در نظر گرفت.

دوره زمانی برای ارزیابی کارایی نیز حائز اهمیت است. درواقع ارزیابی کارایی در یک دوره زمانی کوتاه مدت نمی‌تواند جواب قابل قبولی را ارائه نماید. به‌عنوان مثال در عامل جاروبرقی اگر مقدار خاکی که عامل در اولین ساعات روز تمیز کرده را اندازه بگیریم، به عامل‌هایی که در ابتدای روز با سرعت زیاد کار کرده و سپس بقیه روز را کم کار کرده و یا اصلاً کار نکرده باشند، پاداش داده می‌شود و عامل‌هایی که به‌طور یکنواخت کار کرده‌اند، جریمه می‌شوند، بنابراین با توجه به آن‌چه گفته شد بهتر است کارایی را در دراز مدت (long run) اندازه بگیریم.

منطقی بودن (Rationality)

باید به تفاوت میان منطقی بودن و داشتن **دانش کامل** (Omniscience) توجه خاصی نمود، درواقع داشتن دانش کامل غیرممکن است. به مثال زیر توجه کنید:

من در خیابان شانزلیزه در حال قدم زدن هستم و یک دوست قدیمی را در تقاطع خیابان می‌بینم. در اطراف من ماشینی مشاهده نمی‌شود، بنابراین منطقی است که از خیابان بگذرم، در این زمان شیئی از هواپیما جدا شده و قبل از این که به سمت دیگر خیابان بروم، بر روی من می‌افتد. آیا عبور من از خیابان منطقی بوده است؟ عبور از خیابان منطقی است، زیرا بیشتر وقت‌ها این عبور موفقیت‌آمیز است و راهی برای این که بتوان شیئی را دید وجود ندارد.

ما نمی‌توانیم عاملی را به دلیل شکست خوردن در عملی که قادر به محاسبه آن نیستیم سرزنش کنیم. بنابراین منطقی بودن عامل، باید در حیطه‌ی دانشی باشد که عامل از محیط پیرامون خود به‌دست آورده و اعمالی که می‌تواند انجام دهد، سنجیده شود.

نکته۱: منطقی بودن به معنی **کامل بودن** (Perfection) نیست، بلکه منطقی بودن عامل به معنی ماکزیمم کردن **کارایی مورد انتظار** (Expected performance) و کامل بودن به معنی ماکزیمم‌کردن **کارایی واقعی** (Factual performance) است.

به‌طور خلاصه منطقی بودن عامل در هر زمانی به چهار چیز وابسته است:

۱ـ معیار کارایی که میزان موفقیت را بیان می‌کند.

۲ـ هر آنچه را که تاکنون عامل مشاهده کرده است. ما این تاریخچه کامل ادراکی را **دنباله ادراکی** می‌نامیم.

۳ـ آنچه عامل درباره محیط خود می‌داند.

۴ـ اعمالی که عامل می‌تواند انجام دهد.

یک عامل منطقی ایده‌آل براساس رشته ادراکی و دانش پیش زمینه‌ای که دارد، عملی را انجام می‌دهد که انتظار می‌رود معیار کارایی‌اش را ماکزیمم کند. به‌عنوان مثال اگر عامل قبل از گذشتن از جاده شلوغ به دو طرف نگاه نکند، آنگاه رشته مشاهدات به عامل نخواهد گفت که کامیون بزرگی با سرعت زیاد در حال نزدیک شدن است اما یک عامل منطقی ایده‌آل عمل نگاه کردن را قبل از رد شدن از خیابان انجام می‌دهد، زیرا این عمل کارایی مورد انتظارش را افزایش می‌دهد.

یادگیری (Learning)

اعمالی که عامل برای کسب اطلاعات مفید و یا به عبارتی برای تغییر مشاهدات آینده انجام می‌دهد، **جمع‌آوری اطلاعات** (information gathering) نامیده می‌شود. عامل منطقی علاوه بر جمع‌آوری اطلاعات باید از مشاهدات خود یاد بگیرد. برای مثال، عامل راننده تاکسی که یاد گرفته پیش‌بینی کند کدام مسیر ترافیک سنگین‌تری دارد از عامل مشابهی که این توان را ندارد بهتر عمل می‌کند.

خودمختاری (Autonomy)

مورد دیگری که در تعریف عامل منطقی ایده‌آل لحاظ می‌شود، **دانش درونی** (built in knowledge) و یا **دانش پیش زمینه** (background knowledge) می‌باشد. چنانچه عامل هیچ توجهی به ادراک خود نکند می‌گوییم عامل فاقد **خودمختاری** است. برای مثال یک ساعت را به‌عنوان یک عامل در نظر بگیرید، اگر ساعت و صاحبش به یک مسافرت از کالیفرنیا به استرالیا بروند، کار صحیح برای ساعت این است که خودش را ۶ ساعت به عقب برگرداند. حال اگر سازنده ساعت اطلاع داشت که صاحب آن ساعت به استرالیا سفر می‌کند، آنگاه یک مکانیزم در درون آن ایجاد کند تا در زمان خاصی عقربه‌هایش به‌طور خودکار ۶ ساعت به عقب برگردند، این رفتار موفق خواهد بود اما هوشمندی حاصل از این عمل به طراح ساعت برمی‌گردد نه خود ساعت! درواقع چنین عاملی فاقد خودمختاری است. رفتار عامل می‌تواند متکی بر دو پایه تجربه خود و دانش درونی استفاده شده در ساخت عامل باشد. درواقع عامل منطقی باید خودمختار باشد. یک عامل خودمختار باید یاد بگیرد که برای دانش ناصحیح خود جریمه پرداخت کند. زمانی که عامل فاقد تجربه یا کم‌تجربه است، مجبور است اعمال را به‌صورت تصادفی از میان مشاهدات خود انتخاب کند، مگر آن که طراح کمک‌هایی به آن داده باشد. در حالت کلی یک عامل خودمختار عاملی است که خودش در مورد رفتارش تصمیم بگیرد.

بهترین حالت برای طراحی عامل هوشمند، این است که یک عامل هوشمند را با مقداری دانش اولیه همراه با قدرت یادگیری طراحی کنیم. بعد از این‌که عامل تجربه کافی را از محیط اطراف خود به‌دست آورد، رفتارش می‌تواند مستقل از دانش اولیه او باشد.

نکته۲: خودمختاری نیاز به یادگیری دارد و قدرت یادگیری را نیز می‌توان به هر عاملی داد.

ب ــ ساختار عامل‌های هوشمند

تاکنون درباره عامل‌ها از طریق توصیف رفتارشان بحث شد. حال زمان آن رسیده که به اصل مطلب بپردازیم و درباره چگونگی کارکرد داخلی آن گفتگو کنیم. وظیفه طراح هوش مصنوعی، طراحی برنامه عامل (agent program) است. یعنی طراحی تابعی (agent function) که نگاشتی از رشته مشاهدات (percept sequence) به یک عمل (action) را پیاده‌سازی کند.

بین تابع عامل و برنامه عامل تفاوت وجود دارد. درواقع تابع عامل شامل یک توصیف ریاضی است که فاقد جزئیات پیاده‌سازی می‌باشد و برنامه عامل نیز پیاده‌سازی این تابع است که به‌وسیله نوعی دستگاه محاسباتی به‌نام **معماری** (architecture) اجرا خواهد شد. ارتباط مابین عامل، معماری برنامه و برنامه را می‌توان به‌صورت روبه‌رو تعریف کرد:

معماری + برنامه = عامل

مثال۳: «نگاشتی از ادراک به عمل در حین به هنگام‌سازی وضعیت داخلی» تعریف کدام‌یک از موارد زیر است؟

۱) برنامه عامل ۲) معماری عامل ۳) اهداف عامل ۴) عملیات عامل

☑ **پاسخ:** گزینه «۱» تابعی که نگاشتی از رشته مشاهدات را به یک عمل پیاده‌سازی کند، برنامه عامل نامیده می‌شود.

محیط وظیفه (محیط کاری)

برای طراحی برنامه یک عامل هوشمند، باید اطلاعاتی در مورد محیطی که عامل در آن قرار می‌گیرد، داشته باشیم. محیطی که عامل در آن عمل می‌کند **محیط وظیفه** (Task Environment) نام دارد. این محیط از چهار مؤلفه تشکیل شده است که به دو شکل PAGE و PEAS نمایش داده می‌شود.

چهار مؤلفه معرفی شده به شکل PAGE عبارتند از:

۱) ادراک (مشاهده، Percept) ۲) عمل (Action) ۳) هدف (Goal) ۴) محیط (Environment)

چهار مؤلفه معرفی شده به شکل PEAS عبارتند از:

۱ـ کارایی (Performance) ۲ـ محیط (Environment) ۳ـ محرک‌ها یا اثرکننده‌ها (Actuator) ۴ـ حسگرها (Sensor)

در جدول زیر توصیف محیط وظیفه را برای تعدادی از انواع مختلف عامل‌ها ارائه کرده‌ایم:

مثال‌هایی از انواع عامل

نوع عامل	ادراک (حسگرها)	عمل (محرک‌ها)	هدف (معیار کارایی)	محیط
سیستم تشخیص پزشکی	نشانه‌ها، نتایج آزمایشات، پاسخ‌های بیمار	آزمایشات، تجویز دارو، پرسیدن سؤال	سلامتی بیمار، کاهش هزینه‌ها	بیمار، بیمارستان
سیستم آنالیز تصاویر ماهواره	پیکسل‌ها با شدت روشنایی مختلف، رنگ	طبقه‌بندی تصاویر	طبقه‌بندی صحیح	تصاویر ماهواره‌ای
ربات جابه‌جاکننده قطعات	پیکسل‌ها با شدت روشنایی مختلف	برداشتن قطعات و قرار دادن آن‌ها در جعبه‌ها	قرار دادن قطعات در جعبه‌ها به‌طور صحیح	تسمه حمل‌کننده و قطعات
کنترل‌کننده پالایشگاه	دما و فشار	باز و بسته کردن شیرها، تنظیم دما	ماکزیمم کردن خلوص محصول و افزایش ایمنی	پالایشگاه
معلم درس انگلیسی	کلمات تایپی	چاپ تمرین، پیشنهادات مطرح شده، تصحیح اشکالات	ماکزیمم کردن نمره شاگرد در آزمون	مجموعه شاگردان

مثال۴: یک عامل برای آموزش زبان انگلیسی به روش ارتباط متقابل طراحی شده است. «کلمات تایپ شده» جزو کدام‌یک از ویژگی‌های محیط است؟

۱) محیط ۲) حسگرها ۳) محرک‌ها ۴) معیار کارایی

☑ **پاسخ:** گزینه «۲» کلمات تایپ شده در این سیستم، ورودی و مشاهده عامل هستند که عامل باید به‌ازای آن عملی را انجام دهد. این کلمات، ادراک عامل از محیط خواهند بود.

مثال۵: کدام مورد جزء حسگرهای عامل راننده تاکسی محسوب نمی‌شود؟

۱) دوربین ۲) سرعت‌سنج ۳) شتاب‌دهنده ۴) کیلومترشمار

☑ **پاسخ:** گزینه «۳» شتاب‌دهنده جزء مجموعه محرک‌ها محسوب می‌شود.

ج ـ برنامه‌های عامل (Agent program)

ساده‌ترین راه ممکن برای نوشتن برنامه عامل، استفاده از یک جدول جست‌وجو (Lookup table) می‌باشد که در آن تمام رشته مشاهدات و عملی که باید به‌ازای هر مشاهده انجام گیرد، در یک جدول ذخیره می‌شود. شکل زیر چارچوب برنامه یک عامل با استفاده از جدول را نشان می‌دهد. در این شکل، عامل مشاهده‌ای از محیط خواهد داشت. طبق این مشاهده، حافظه‌اش را به‌روزرسانی می‌کند و سپس بهترین عمل را انتخاب می‌نماید. پس از آن با توجه به تأثیر این عمل بر محیط، حافظه‌اش را به‌روزرسانی می‌کند. در آخر نیز عمل انتخاب شده را باز می‌گرداند تا محرک‌های مربوطه، آن را انجام دهند.

```
Function SKELETON – AGENT (percept) return action
    Static: memory (the agents memory of the world)
    memory ← UPDATE – MEMORY (memory , percept)
    action ← CHOOSE – BEST – ACTION (Memory)
    memory ← UPDATE – MEMORY (memory , action)
    return action
```

چارچوب یک عامل

به دو نکته زیر درباره شالوده برنامه دقت کنید:

۱ـ اگرچه برنامه عامل به‌صورت تابعی از رشته مشاهدات به اعمال است، اما فقط یک مشاهده (percept) را به‌عنوان ورودی دریافت می‌کند. البته در بعضی از محیط‌ها بدون ذخیره‌سازی رشته مشاهدات نیز موفقیت کامل تضمین می‌شود.

۲ـ هدف یا معیار کارایی، قسمتی از برنامه عامل نمی‌باشد، زیرا معیار کارایی برای قضاوت رفتار عامل به‌صورت خارجی اعمال می‌شود و ممکن است بدون اطلاع صریح از معیار کارایی، به بالاترین مقدار کارایی برسیم. همان‌طور که گفتیم، ساده‌ترین شکل ممکن برای ساخت برنامه عامل، استفاده از جدول است. شکل زیر برپایه حفظ تمام مشاهدات در جدول عمل نموده و از مشاهدات برای اشاره به مدخل‌های جدول (که شامل عمل مناسب برای همه رشته مشاهدات ممکن است) استفاده می‌کند.

این روش با مشکلات عمده زیر مواجه است:

۱ـ حافظه مورد نیاز برای ذخیره‌سازی جدول زیاد است. به‌عنوان مثال بازی شطرنج به حدود $۳۵^{۱۰۰}$ حالت نیاز دارد.

۲ـ ساخت جدول برای طراح زمان‌بر است.

۳ـ عامل خود مختاری نخواهد داشت، زیرا محاسبه بهترین عمل فقط بر اساس دانش درونی است.

۴ـ حتی اگر عامل از یک مکانیزم یادگیری به منظور ایجاد درجه‌ای از خودمختاری استفاده کند، پیدا کردن مقدار صحیح برای همه مدخل‌ها و به‌روزرسانی آن‌ها زمان‌بر است.

```
Function TABLE-DRIVEN-AGENT (percept) return action:
    Static: percepts (a sequence, initially empty)
            table a table, indexed by percept sequences, initially fully specified
    append percept to the end of percepts
    action ← LOOK-UP (percept, table)
    return action
```

پیاده‌سازی عامل با استفاده از جدول

با وجود مشکلات ذکر شده، عامل فوق آنچه ما می‌خواهیم را انجام می‌دهد. نکته این است که چگونه عامل می‌تواند از طریق پرهیز از چهار مشکل ذکر شده بهتر عمل کند.

در اینجا بد نیست محیط خاصی را در نظر بگیریم تا بحث‌ها واقعی‌تر گردد. این محیط خاص «سیستم هوشمند راننده تاکسی» است. این محیط ترکیبی از شرایط و واقعیات مختلف است، به همین دلیل جهت بررسی انواع برنامه‌های عامل مورد توجه قرار می‌گیرد. در جدول زیر توصیف PEAS برای محیط وظیفه راننده تاکسی خلاصه شده است.

توصیف PEAS برای محیط وظیفه راننده تاکسی

نوع عامل	ادراک (حسگرها)	محرک	هدف (معیار کارایی)	محیط
راننده تاکسی	دوربین‌های سرعت‌سنج، GPS، صداها، میکروفون	هدایت‌کننده‌ها، شتاب گرفتن، ترمز، گاز، بوق، صحبت کردن با مسافر	امنیت، سرعت، رعایت قانون، سهولت مسافرت، افزایش سودمندی	جاده‌ها، ترافیک، مشتریان، عابران پیاده

راننده تاکسی باید بداند کجاست، در جاده چه خبر است و با چه سرعتی حرکت می‌کند. این مشاهدات را با استفاده از یک یا چند دوربین، سرعت‌سنج و ... به‌دست می‌آورد و برای کنترل‌کردن وسیله نقلیه نیاز به شتاب‌سنج دارد و همچنین برای یافتن موقعیت خود لازم است مجهز به سیستم مکان‌یاب عمومی و ماهواره‌ای (GPS) باشد. اعمال یک برنامه هوشمند راننده تاکسی، کم و بیش شبیه اعمالی است که یک راننده انسانی باید انجام دهد.

اعمالی مانند شتاب گرفتن، ترمز کردن، صحبت کردن با مسافر از طریق یک صفحه نمایش و یا به‌کارگیری بلندگو برای ارتباط با سایر وسایل نقلیه باید در نظر گرفته شود.

در این مثال معیار کارایی چیست؟ معیارهای مطلوب شامل رسیدن به مقصد، مینیمم کردن سوخت، مینیمم کردن زمان و هزینه سفر، مینیمم کردن عدم رعایت قوانین ترافیکی و کاهش تصادفات می‌تواند در معیار کارایی در نظر گرفته شود، درواقع در معیارکارایی باید بین این اهداف مختلف تعادل برقرار شود.

در پایان، نوع محیطی که تاکسی با آن مواجه می‌شود باید تعیین شود. مثلاً تاکسی در جاده محلی است یا در بزرگراه قرار دارد. آیا تاکسی در جنوب کالیفرنیاست که به‌ندرت برف ایجاد مشکل می‌کند یا در آلاسکا قرار دارد، آیا تاکسی باید سمت راست حرکت کند یا مانند انگلیس و ژاپن سمت چپ. بدیهی است که محدودسازی محیط، طراحی را ساده‌تر می‌کند.

حال باید تصمیم بگیریم برای یک مسأله خاص از چه روشی استفاده کنیم و چگونه یک برنامه عامل برای آن بنویسیم.

عموماً چهار نوع برنامه عامل مورد بررسی قرار می‌گیرد:

۱ـ عامل واکنشی (بازتابی، انعکاسی) ساده (Simple reflex agents)

۲ـ عامل واکنشی مبتنی بر مدل (Model-based reflex agents)

۳ـ عامل مبتنی بر هدف (Goal-based agents)

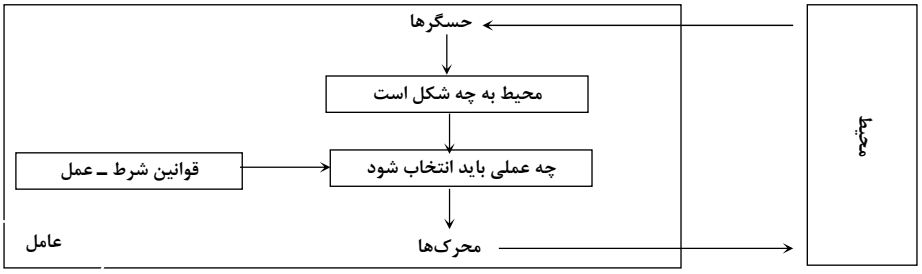
۴ـ عامل مبتنی بر سودمندی (Utility-based agents)

عامل واکنشی ساده

عامل واکنشی ساده، ساده‌ترین نوع عامل است که در هر لحظه، عملی را براساس درک فعلی انتخاب می‌کند. این عامل‌ها براساس مجموعه‌ای از **قوانین شرط ـ عمل** (condition-action rules) ایجاد شده و در حالت کلی از قانون اگر (شرط) آنگاه (عمل) تبعیت می‌کند. به‌طور مثال اگر چراغ ترمز خودروی جلویی روشن شد، آنگاه ترمز کن. شکل‌های زیر به ترتیب ساختار و برنامه یک عامل واکنشی ساده را نشان می‌دهند.

اگرچه عامل‌های واکنشی ساده را می‌توان به راحتی پیاده‌سازی کرد، اما از هوش کمی برخوردار هستند و در صورتی، به‌درستی عمل می‌کنند که تصمیم فعلی براساس ادراک فعلی قابل نتیجه‌گیری باشد. در عامل واکنشی ساده اگر اطلاعات کافی از محیط نداشته باشیم (محیط مشاهده‌پذیر جزئی باشد) با مشکل مواجه می‌شویم. به‌طور مثال یک عامل جاروبرقی که از نوع واکنشی ساده است و حسگر آلودگی دارد اما حسگر مکان ندارد، در صورتی‌که حسگرهای آن کثیف بودن را درک کنند، شروع به تمیزکردن می‌کند اما اگر حسگرها تمیز بودن را درک کنند آنگاه عمل بعدی چه خواهد بود؟

در صورتی‌که این عامل، اگر حسگر مکان داشت می‌توانست در صورت تمیز بودن مکان فعلی به مکان بعدی تغییر مکان دهد.



ساختار یک عامل واکنشی ساده

```
Function SIMPLE- REFLEX- AGENT (percept) return action:
  Static: rules (a set of condition – action rules)
  State ← INTERPRET.INPUT (percept)
  rule ← RULE – MATCH (State, rules)
  action ← RULE – ACTION [rule]
  return action
```

برنامه یک عامل واکنشی ساده

عامل واکنشی مبتنی بر مدل

عامل واکنشی ساده در صورتی درست کار می‌کند که بتواند تصمیم صحیحی از مشاهدات کنونی بگیرد. در غیر این‌صورت نیازمند حافظه‌ای خواهد بود که وضعیت عامل و محیط را در خود ذخیره کند تا عامل بتواند با توجه به اطلاعات ذخیره شده، درک بهتری از محیط داشته باشد.

به‌طور مثال در ماشین‌های قدیمی، چراغ‌های ترمز و چراغ‌های راهنما تفاوت‌هایی دارند که از روی یک تصویر امکان تشخیص اینکه ماشین در حال ترمز کردن است یا راهنما زدن وجود ندارد. بنابراین حتی برای یک قانون ساده ترمز، مجبوریم مجموعه‌ای از حالات درونی (internal states) را برای انتخاب صحیح عمل استفاده کنیم.

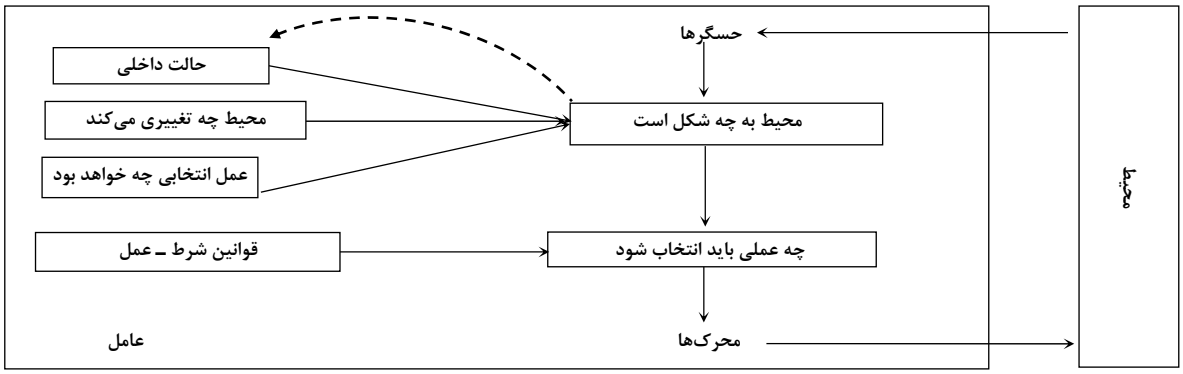
به‌دلیل این‌که حسگرها، دسترسی به حالت کامل محیط را برای عامل فراهم نمی‌کنند، عامل، نیاز به نگهداری اطلاعات قبلی در حالت درونی خود دارد تا قادر باشد حالت‌های محیط را که ورودی (percept) یکسانی توصیه می‌کنند اما درواقع با هم متفاوتند را تشخیص دهد (یعنی اعمال متفاوتی برای آن‌ها انتخاب کند).

به‌هنگام سازی دانش درونی عامل، مستلزم دو نوع دانش زیر خواهد بود:

۱ـ دانش مربوط به تغییرات دنیا: به‌طور مثال ماشینی که در حال سبقت‌گرفتن از تاکسی است، اکنون نسبت به یک دقیقه پیش نزدیک‌تر است.

۲ـ دانش مربوط به تأثیر اعمال عامل بر دنیا: به‌طور مثال وقتی تاکسی تغییر خط بدهد به‌طور موقت یک جای خالی در خط قبلی ایجاد می‌شود و یا این‌که بعد از رانندگی به مدت ۵ دقیقه در یک آزاد راه به سمت شمال ۵ مایل نسبت به جایی که ۵ دقیقه پیش بودیم بالاتر هستیم.

شکل‌های زیر به‌ترتیب ساختار و برنامه یک عامل مبتنی بر مدل (حافظه‌دار) را نشان می‌دهند.



ساختار یک عامل واکنشی مبتنی بر مدل

قسمت قابل توجه برنامه عامل، تابع UPDATE-STATE است که مسئول ایجاد وضعیت داخلی جدید است.

```
Function REFLEX-AGENT-WITH STATE (percept) return an action
    Static: (state a description of the current world state)
    rules (a set of condition – action rules)
    action (the most recent action, initially non)
    state ← UPDATE – STATE (state, action, percept)
    rules ← RULE – MATCH (state, rules)
    action ← RULE – ACTION (rule)
    return action
```

عامل واکنشی با حالت داخلی که توسط یافتن قانونی که موقعیتش با شرایط جاری صدق می‌کند کار می‌کند (همان‌گونه که توسط ادراک و حالت داخلی ذخیره شده تعریف می‌شود) و سپس عمل مربوط به آن قانون انجام می‌شود.

مثال ۶: یک نسخه تغییر یافته از محیط جاروبرقی (Vacuum-Cleaner environment) را در نظر بگیرید که در آن، عامل به‌ازای هر حرکت، یک امتیاز منفی می‌گیرد. کدام‌یک از جملات زیر صحیح است؟

۱) یک Simple reflex agent می‌تواند در این محیط کاملاً Rational (منطقی) باشد.

۲) یک Model-based reflex agent می‌تواند در این محیط Rational باشد.

۳) اگر ادراک‌های عامل، وضعیت تمیزی همه خانه‌های محیط را به او منتقل کنند، آنگاه یک Simple reflex agent می‌تواند در این محیط Rational باشد.

۴) گزینه ۲ و ۳ صحیح است.

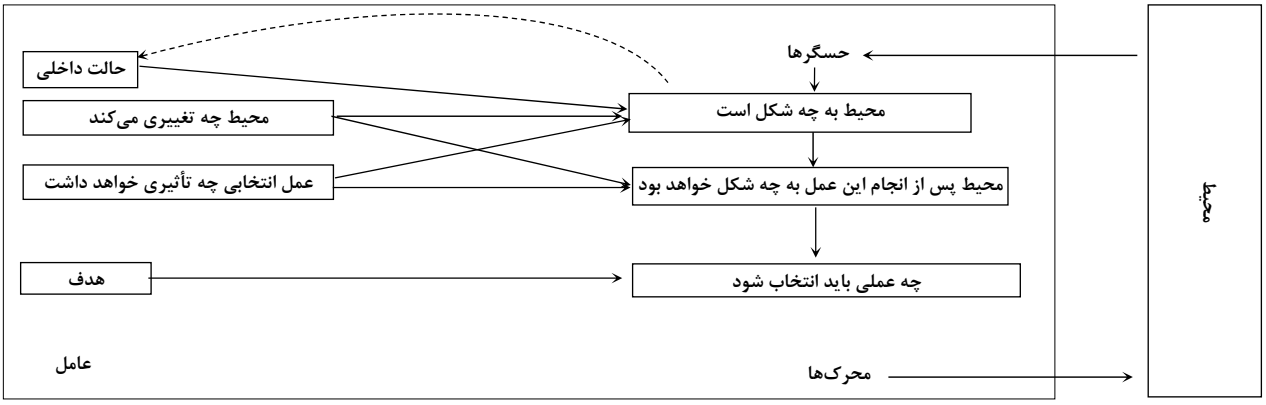
☑ **پاسخ:** گزینه «۴» گزینه ۱ غلط است؛ زیرا عامل در شرایطی که تمام خانه‌ها تمیز باشد، همچنان به حرکت خود ادامه می‌دهد و امتیاز از دست می‌دهد.

گزینه ۲ صحیح است. در این حالت، عامل با توجه به وضعیت درونی که نگه می‌دارد، می‌تواند کمترین بازگشت به خانه‌های ملاقات شده را داشته باشد و پس از تمیز شدن همه محیط، متوقف شود. گزینه ۳ صحیح است. زیرا در این حالت Simple reflex می‌تواند کاملاً rational باشد. کافی است در جدول به‌ازای هر وضعیت دنیای عامل، بهترین عملی که عامل باید انجام دهد را مشخص کنیم. عامل با دریافت درک از دنیا، بهترین عمل ممکن برای آن وضعیت را از جدول پیدا کرده و انجام می‌دهد. با انجام این عمل، دنیا تغییر می‌کند اما عامل با ادراک بعدی از تمام این تغییرات آگاه می‌شود.

نکته ۳: عامل خودمختار می‌تواند به‌صورت واکنشی پیاده‌سازی شود و قوانین شرط و عمل را به‌مرور زمان یاد بگیرد.

عامل مبتنی بر هدف

دانش موجود درباره‌ی حالت جاری همیشه برای تصمیم‌گیری کافی نیست. به‌طور مثال تاکسی در یک چهارراه می‌تواند به سمت چپ یا راست بپیچد و یا مستقیم برود، درحالی‌که تصمیم صحیح به مقصد مسافر بستگی دارد. به‌عبارت دیگر، عامل، نیاز به یک هدف‌گذاری و جهت‌دهی برای تصمیمات آینده خود دارد. رسیدن به هدف گاهی اوقات آسان است ولی در شرایط پیچیده‌تر (یعنی زمانی که برای رسیدن به هدف نیاز به چندین انتخاب و عمل باشد) می‌بایست جست‌وجو و برنامه‌ریزی کرد و درواقع جست‌وجو (Search) و برنامه‌ریزی (planning) زمینه‌هایی از هوش مصنوعی هستند که دنباله اعمال را برای رسیدن عامل به هدف پیدا می‌کنند. در طراحی عامل واکنشی، تصمیم‌گیری درباره این‌که چنین یا چنان کنم وجود ندارد بلکه طراح از قبل، عمل صحیح را برای موارد مختلف محاسبه کرده است. اما در عامل واکنشی مبتنی بر هدف عامل توانایی وفق‌پذیری با شرایط جدید را دارا بوده و بسیار انعطاف‌پذیر عمل می‌کند. شکل زیر ساختار عامل‌های هدف‌گرا را نشان می‌دهد.



ساختار یک عامل مبتنی بر هدف

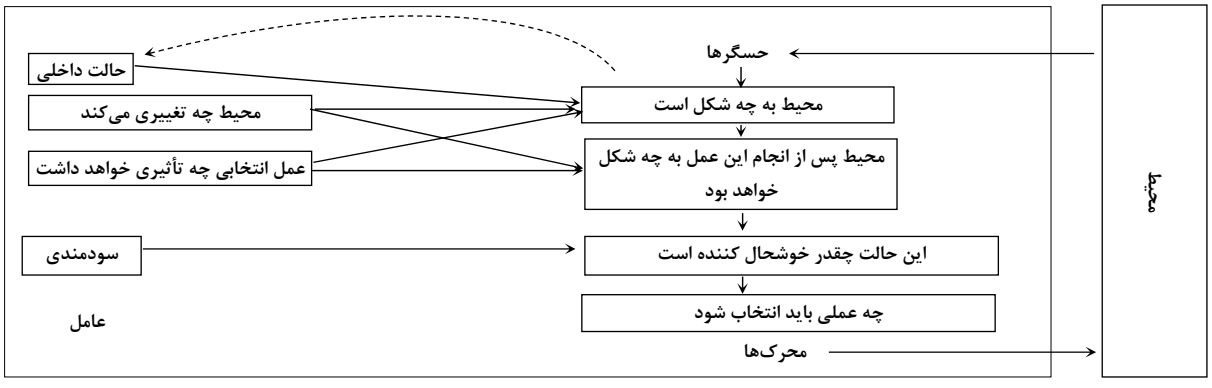
عامل مبتنی بر سودمندی

در بسیاری از محیط‌ها اهداف برای تولید رفتاری با کیفیت بالا کافی نیستند. برای مثال در محیط عامل راننده تاکسی ممکن است چندین مسیر برای رسیدن به مقصد موجود باشد اما بعضی از آن‌ها سریع‌تر، امن‌تر و یا ارزان‌تر از بقیه هستند. اهداف فقط یک مقصد ساده بین حالت‌های مطلوب یا نامطلوب را فراهم می‌کنند. برای تعیین درجه مطلوبیت از تابع سودمندی استفاده می‌شود. بنابراین سودمندی، تابعی است که هر حالت را به یک عدد (که درجه رضایت عامل از آن حالت را مشخص می‌کند) نگاشت می‌نماید.

تابع سودمندی منجر به تصمیمات منطقی در دو حالت می‌شود:

۱ـ حالت **اهداف متناقض** (مثل سرعت، امنیت در مثال راننده تاکسی) که در این حالت اهداف در تناقض یکدیگر هستند که فقط برخی از آن‌ها تأمین می‌شود. درواقع تابع سودمندی تعادلی مناسب بین آن‌ها برقرار می‌کند.

۲ـ حالت **اهداف چندگانه** که در آن عامل در انتخاب یک هدف بین چندین هدف ناتوان است. وظیفه تابع سودمندی برای این حالت پیشنهاد کردن راهی است که در آن موفقیت براساس اهمیت اهداف وزن‌دهی می‌شود.



ساختار یک عامل مبتنی بر سودمندی

مثال۷: کدام‌یک از برنامه‌های عامل زیر از قوانین شرط ـ عمل استفاده نمی‌کنند؟

۱) عامل مبتنی بر مدل ۲) عامل مبتنی بر هدف ۳) عامل مبتنی بر سودمندی ۴) گزینه‌های ۲ و ۳

☑ **پاسخ:** گزینه «۴» عامل مدل‌گرا با درنظر گرفتن چند وضعیت داخلی به‌عنوان حالت سیستم، با توجه به این وضعیت و حس از این سیستم، قوانینی را که بر اساس شرط ـ عمل هستند، اعمال می‌کند، ولی عامل‌های مبتنی بر هدف و مبتنی بر سودمندی، عامل‌های پیچیده‌تری هستند که برای انتخاب عمل نیاز به یک تابع هدف یا تابع سودمندی دارند.

مثال۸: کدام‌یک از گزینه‌های زیر نادرست است؟

۱) محیط‌های کاری E_1 و E_2 و یک عامل A وجود دارند. حتی اگر دو محیط یکسان نباشند، عامل A می‌تواند در هر دو محیط عقلانی عمل می‌کند.

۲) محیط‌های کاری وجود دارند که در آن برخی از عامل‌های بازتابی، عملکرد عقلانی دارند.

۳) عاملی که تنها بخشی از محیط را درک می‌کند نمی‌تواند کاملاً عقلانی باشد.

۴) محیط کاری وجود دارد که در آن هر عاملی عقلانی است.

☑ **پاسخ:** گزینه «۳» زیرا عقلانیت و دانش کامل داشتن دو مقوله کاملاً جدا هستند. به‌طور مثال انسان یک موجود عقلانی است و بخشی از محیط را نیز درک می‌کند.

گزینه ۱ صحیح است؛ زیرا یک عامل می‌تواند در محیط‌های کاری مختلف تصمیم‌گیری کاملاً عقلانی داشته باشد.

گزینه ۲ صحیح است؛ به‌طور کلی هر محیط کاملاً مشاهده‌پذیر می‌تواند توسط عامل بازتابی اداره شود و عملکرد عقلانی داشته باشد.

عامل‌های یادگیرنده

عامل یادگیرنده عاملی است که از چهار مؤلفه‌ی اصلی تشکیل شده است:

۱ـ عنصر اجرایی (performance element): درواقع همان عامل است؛ یعنی مشاهدات را می‌گیرد و اعمال را انتخاب می‌کند.

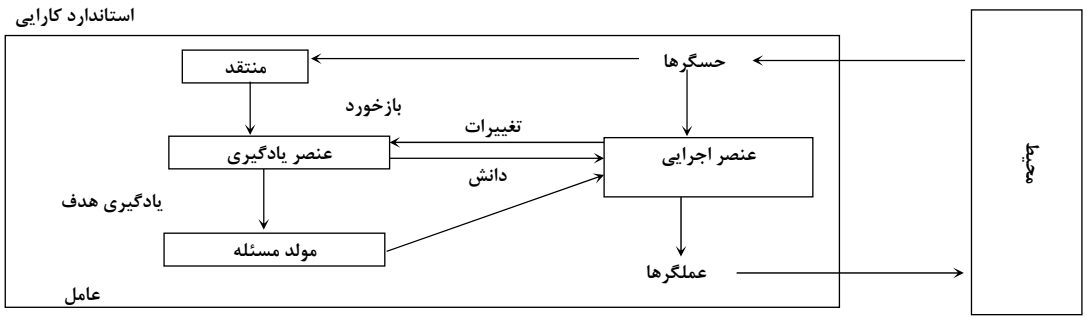
۲ـ عنصر یادگیری (learning element): به منظور بهبود عملکرد، از بازخورد منتقد استفاده می‌کند.

۳ـ منتقد (critic): این مؤلفه مشخص می‌کند عامل چگونه عمل کرده است و عنصر کارایی باید چگونه تغییر کند تا در آینده عامل بهتر عمل کند.

وجود منتقد ضروری است؛ زیرا مشاهدات به تنهایی نمی‌توانند نشانه‌هایی از موفقیت عامل را ارائه دهند. بنابراین داشتن یک **استاندارد کارایی** ثابت الزامی است؛ زیرا از نظر مفهومی استاندارد کارایی خارج از عامل است و عامل نباید برای مناسب کردن عمل خود استاندارد کارایی را تغییر دهد. درواقع منتقد به عنصر یادگیری اطلاع می‌دهد که عامل براساس معیار کارایی چقدر خوب عمل کرده است.

۴ـ مولد مسأله (problem generator): این جزء مسئول پیشنهاد اعمالی است که منجر به تجربه‌های ارزشمند و مفید می‌گردد.

عنصر یادگیرنده می‌تواند از طریق «عمل یادگیری مستقیم از رشته ادراکی»، «یادگیری نحوه‌ی تغییرات دنیا» و «یادگیری از طریق مشاهده نتایج اعمال خود» عمل یادگیری را انجام دهد و در کنار این موارد می‌تواند از مکانیزم پاداش و جریمه نیز استفاده کند. برای مثال میزان پاداش دریافتی از مسافران توسط عامل راننده تاکسی می‌تواند معیاری برای چگونگی عملکرد و به تبع آن یادگیری‌های بعدی عامل باشد.



ساختار عامل یادگیرنده

شکل صفحه بعد تمام حالت‌های قرار گرفتن یک عامل جاروبرقی در دو اتاق را با توجه به وضعیت تمیز و کثیف بودن اتاق‌ها نمایش می‌دهد. با استفاده از یک متغیر که می‌تواند هر یک از مقادیر مجموعه S را داشته باشد، می‌توان وضعیت داخلی محیط را ثبت کرد.