مترجم: سهراب جلوه گر
ویرایش دوّم ، بهار ۱۳۸۸

ریوس مطالب

- عامل ها و محیط ها
- عقلانیّت [۱]
- PEAS (ارزیابی عملکرد [۲] ، محیط [۳] ، عمل کننده ها [٤] و حسگرها [٥])
- انواع محیط
- انواع عامل ها

---

[۱] rationality

[۲] Performance measure یا معیار کارایی

[۳] Environment

[۴] Actuators

[۵] Sensors

مترجم: سهراب جلوه گر
ویرایش دوّم ، بهار ۱۳۸۸

عامل ها

* یک عامل می تواند هر چیزی باشد که در یک **محیط** عمل می کند ؛ [وضعیّت] محیط خود را با استفاده از **حسگرها** دریافت می کند ؛ روی محیط خود با استفاده از **عمل کننده ها** عمل می نماید و طوری عمل می کند که بیش ترین نزدیکی را به اهدافش داشته باشد .

تعریف های دیگری در مورد عامل :

- **تعریف راسل و نورویگ (R & N) :** هر چیزی که می تواند وضعیّت محیط خود را با استفاده از حسگرها دریافت کند و توسط محرّک ها بر روی محیط عمل کند یک عامل است .

- **وولریج** [۱] **:** یک عامل ، یک سیستم کامپیوتری است که در یک محیط قرار داده می شود و قادر است به صورت خودمختار عمل نماید .

* **ابزار ادراکی برای تحلیل سیستم ها :** روبوت ها [۲] ، سافتبوت ها [۳] ، چراغ های ترافیک ، ترموستات ها [۴] و غیره

* عامل های هوشمند از این جهت برای ما اهمّیّت دارند که به هوش مصنوعی کمک می کنند .

مثال هایی از عامل ها

---

[۱] Woolridge
[۲] robots
[۳] softbots
[۴] thermostats

مترجم: سهراب جلوه گر
ویرایش دوّم ، بهار ۱۳۸۸

عامل انسانی مثل : چشم ها ، گوش ها ، پوست ، حس چشایی [۱] و ... برای حسگرها و دست ها ، انگشتان ، پاها ، دهان [۲] و ... برای عمل کننده ها

برای روبوت : دوربین ، دوربین مادون قرمز [۳] ، ضربه گیر یا سپر [۴] و ... برای حسگرها و گیرنده ها (ابزارهایی که چیزی را می گیرند) [۵] ، تایرها ، نورها ، بلند گوها و ... برای عمل کننده ها

برای عامل نرم افزاری یا سافتبوت ، حسگرها به صورت توابع هستند و ورودی توابع ، اطّلاعات آماده شده به صورت رشته های بیتی یا نشان ها (سمبل ها) ی کد شده هستند . و توابع ، که عمل کننده ها نیز هستند خروجی ها را اجرا می کنند . پس توابع هم به صورت حسگر هستند و اطّلاعات را به صورت رشته های بیتی یا نشان ها (سمبل ها) ی کد شده می گیرند و هم به صورت عمل کننده هستند که بر روی محیط ، عمل می کنند .

خودمختاری [۶] در عامل ها

عامل ها وظایف خود را تا حد زیادی به صورت مستقل انجام می دهند و به وسیله ی کاربران یا برنامه های دیگر برنامه ریزی می شوند و این برنامه ها عامل را برای انجام کارش راهنمایی می کند . مبنای سیستم های خودمختار ، تجربه و دانش آن ها می باشد . عامل ها به دانش اوّلیه نیاز دارند و باید بتوانند یاد بگیرند و برای کارهای به مراتب پیچیده تر دارای انعطاف باشند .

مطلب دیگری در مورد خودمختاری

---

[۱] taste buds
[۲] mouth
[۳] infrared
[۴] bumper
[۵] grippers
[۶] autonomy

مترجم: سهراب جلوه گر
ویرایش دوّم ، بهار ۱۳۸۸

عامل ها اغلب دارای خود مختاری هستند ولی این خودمختاری ، اغلب به صورت کامل نمی باشد . حال مشکلی در این مورد وجود دارد و آن این است که عامل خود مختار چه هنگام باید درخواست کمک کند ؟ . یک عامل خودمختار قادر است با توجّه به دریافت هایش تصمیم بگیرد .

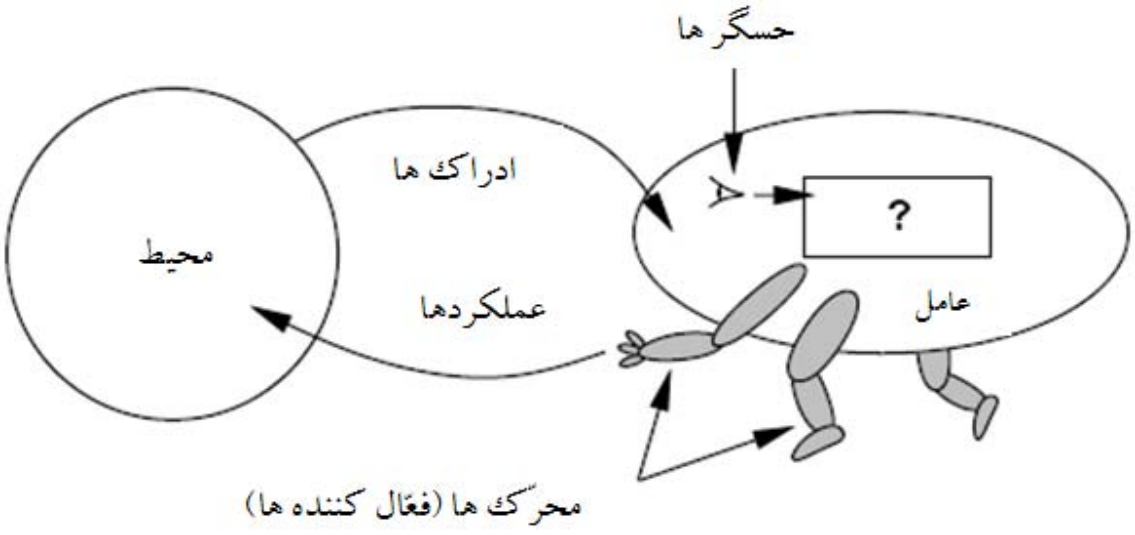
چگونه با این دردسرها رفتار کنیم ؟ : عامل ها دوست دارند که کاری که انجام می دهند دارای پیش نیاز نباشد . مشخص کردن این که در هر موردی باید چه کاری انجام دهند مشکل است . در مورد عامل ها بهتر است فرض را بر این بگذاریم که آن ها هوشمند هستند و سپس در مورد آن ها صحبت کنیم .

عامل ها و محیط ها

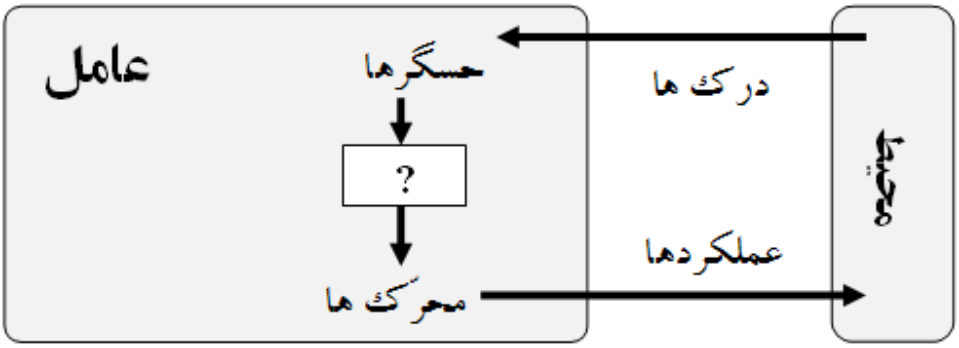
عامل ها شامل انسان ها ، ربوت ها ، ترموستات ها (تنظیم کننده های حرارت) و ... می باشند ؛ مولفان دارای عامل ها هستند ، ورزشکاران حرفه ای دارای عامل ها هستند ، ستاره های سینما دارای عامل ها هستند و شما نیز دارای عامل ها می باشید . زیرا عامل ، کسی یا چیزی دارای تخصص می باشد که عهده دار انجام کاری برای شما می شود . برنامه های کامپیوتری که به شما کمک می کنند که دانسته های محاسباتی خود را افزایش دهید " عامل ها " می باشند . تحقیقات دانشمندان هوش مصنوعی در حال افزایش توانایی ، استقلال و ... عامل ها می باشد . [۱] تابع عامل از تاریخچه های ادراکی ، به عملیّات نگاشت می کند : $f : p^* \to A$. برنامه ی عامل ، در معماری فیزیکی برای تولید f اجرا می شود.

[۱] http://www.aaai.org/AITopics/html/agents.html

مترجم: سهراب جلوه گر
ویرایش دوّم ، بهار ۱۳۸۸



یا



ادراک ها (دریافت ها) : اطّلاعات فرستاده شده به حسگرهای یک عامل می باشند . مثل ، نور ، صدا ، امواج الکترو مغناطیسی و علایم (سیگنال ها) .

حسگرها : روش های های یک عامل برای جمع آوری اطّلاعات در مورد محیطش می باشند . مثل : چشم ها ، گوش ها ، سلول های فتوالکتریک و ...

مترجم: سهراب جلوه گر
ویرایش دوّم ، بهار ۱۳۸۸

محرّک ها : عامل ، به وسیله ی محرّک ها بر روی محیط ، عمل می نماید . تایرها ، بازوها ، رادیوها ، نورها و ... مثال هایی از محرّک ها هستند .

عملکرد : کار یا عملی است که بر روی محیط انجام می شود ؛ مثل : حرکت و غلطیدن .

ساختار عامل های هوشمند

عامل ، از معماری و برنامه تشکیل شده است . *معماری ، ابزاری است که به وسیله ی آن می توان برنامه ی عامل را اجرا کرد .* مثل : کامپیوتر همه منظوره ، ابزار تخصصی ، روبوت و ...

برنامه ی عامل ها

ما می توانیم رفتار عامل مان را با یک تابع F توصیف نماییم . داریم :

عملکرد = (تاریخچه ی ادراکی ، ادراک فعلی) F

به کار گیری این تابع ، کار یک برنامه ی عامل می باشد .

مثال : دنیای جاروبرقی

بیایید با یک مثال خیلی ساده شروع کنیم ؛ فرض کنید که دو فضای A و B وجود دارند که هر فضا یا می تواند تمیز یا کثیف باشد و این ، محیط عامل است . حسگر ها عبارتند از : حسگر کثیفی و تعیین محل . محرّک ها عبارتند از : جارو برقی و چرخ ها . ادراک ها هم تمیز و پاک بودن محیط می باشد . عملکردها ؛ حرک به چپ ، راست ، مکش و هیچ کار می باشند . در این مثال ساده ، ما می توانیم همه ی رشته های ادراکی ممکن و عملکردهای وابسته را لیست نماییم و این ، یک عامل برمبنای جدول [1] نام دارد .

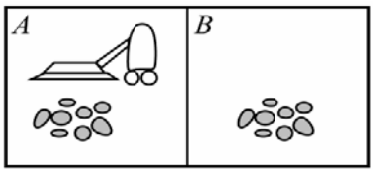
---

[1] table-based

مترجم: سهراب جلوه گر
ویرایش دوّم ، بهار ۱۳۸۸

عقلانیّت [۱]

به طورکلّی ، عقلانیّت به معنی " انجام چیز درست می باشد " در این مورد دقّت بیش تری لازم می باشد ؛ "چیز درست " ، چیست ؟ ما به یک تعریف از موفّقیّت هم نیازمندیم . یک عملکرد عقلانی (عاقلانه یا از روی عقل) ، عملکردی است که معیار کارایی (ارزیابی عملکرد) یک عامل ، ادراک ها و عملکردهای آن را بیشینه می نماید . راسل و نورویگ در این باره می گویند : عامل های عقلانی ، برای هر رشته ی ادراکی ، باید عملکردی که انتظار می رود معیار کارایی و ملاک ارایه شده توسط رشته ی ادراکی و هر چه در دانش عامل وجود دارد را بیشینه می کند ، انتخاب نماید . ما نیازی نداریم که عامل قادر باشد که آینده را پیش بینی نماید یا رویدادهای بد را پیش گویی نماید . جمع آوری اطّلاعات هم ممکن است یک عملکرد عقلانی باشد ؛ به عنوان مثال ، عبور بدون توجّه از خیابان ، نامعقولانه می باشد . عامل های عقلانی باید دارای توانایی یادگیری (آموزش) [۲] باشند (به جز در موارد خیلی ساده و محیط هایی که به خوبی قابل فهم می باشند) . *آموزش* به معنی بهبوددادن عملکرد یک عامل است . آموزش ، ممکن است به معنی کاهش نامعلومی (احتمال) یا ارایه ی ملاحظاتی به حساب یا گزارش باشد .



عامل های یک جارو برقی

---

[۱] rationality

[۲] learning

مترجم: سهراب جلوه گر
ویرایش دوّم ، بهار ۱۳۸۸

چیز آگاهی [۲] ندارد . دریافت ادراکی ممکن است همه ی اطّلاعات را به وجود نیاورد. عامل هوشمند ، نهان بین [۳] هم نمی باشد . عملکرد به وجود آمده ممکن است به صورتی که انتظار داشتیم ، نباشد . در این صورت ، عامل هوشمند ، موفّقیّت آمیز نمی باشد . عامل هوشمند دارای ابتکار [۴] ، یادگیری [۵] و خودمختاری می باشد .

PEAS - برای طرّاحی یک عامل هوشمند ، ما باید محیط عملیّاتی [۶] را مشخص کنیم . مثلا درطرّاحی یک تاکسی خودکار ، موارد زیر را باید مشخص نماییم : معیارکارایی ؟؟ | محیط ؟؟ | عمل کننده ها (محرّک ها) ؟؟ | حسگرها ؟؟

مثال ، طرّاحی یک تاکسی خودکار ، معیارکارایی ؟؟ ایمنی [۷] ، مقصد [۸] ، مزایا [۹] ، رعایت قانون [۱۰] ، آسایش و راحتی [۱۱] و ..... | محیط ؟؟ خیابان های ایالات متحده ی آمریکا / آزاد راه های ایالات متحده ی آمریکا ، ترافیک [۱۲] ، عابران پیاده [۱۳] ، هوا [۱] و ..... | عمل کننده ها ؟؟ فرمان [۲] ، گاز ، ترمز [۳] ، بوق

---

[۱] rational agent
[۲] omniscient
[۳] clairvoyant
[۴] exploration
[۵] learning
[۶] task environment
[۷] safety
[۸] destination
[۹] profit
[۱۰] legality
[۱۱] comfort
[۱۲] traffic
[۱۳] pedestrains

مترجم: سهراب جلوه گر
ویرایش دوّم ، بهار ۱۳۸۸

محیط ها

یک معیار برای وجود یک عامل ، وجود آن در یک محیط می باشد . لازم نیست که یک عامل به صورت فیزیکی در محیط باشد ؛ ممکن است محیط ، به صورت یک محیط نرم افزاری باشد . به نظر راسل و نورویگ ، محیط عملیّاتی تشکیل شده از : معیار کارایی ، محیط و محرّک هایی که برای عامل قابل دسترسی می باشند .

ویژگی محیط ها

عبارتند از : قابل مشاهده بودن [۳] ؛ قطعی / اتّفاقی بودن ؛ دوره ای (اپیزودیک) و ترتیبی بودن ؛ ثابت (استاتیک) و پویا (دینامیک) بودن ؛ گسسته و پیوسته بودن ؛ تک عاملی و چند عاملی بودن .

قابل مشاهده بودن : در صورتی که حسگرهای عامل همیشه اطّلاعات کاملی در مورد اجزای مربوط محیط بدهند ، محیط ، کاملاً قابل مشاهده است . به عنوان مثال ، محیط بازی شطرنج ، کاملاً قابل مشاهده می باشد . محیط جارو برقی ، به صورت جزیی قابل مشاهده می باشد (در صورتی که در فضای مجاور ، کثیفی وجود داشته باشد نمی تواند آن را ببیند) .

قطعی / اتّفاقی بودن : ما می توانیم فکر کنیم که جهان دارای انتقال میان وضعیّت ها می باشد . در این مورد داریم ، وضعیّت جاری * عملکردهای عامل ← (نتیجه می دهد) وضعیّت جدید . در صورتی که انتقال وضعیّت ، منحصر به فرد باشد ، جهان به صورت قطعی می باشد . به عنوان مثال ، بازی شطرنج ، بازی ای قطعی می باشد . دنیای جاروبرقی ، قطعی است . رانندگی یک اتومبیل ، تصادفی می باشد .

---

[۱] shippers

[۲] display to user

[۳] observability

مترجم: سهراب جلوه گر
ویرایش دوّم ، بهار ۱۳۸۸

نکته : در این جا ما اجتناب می کنیم که سؤالات را با مکانیک کوانتومی (ذره ای) موشکافی نماییم – ممکن است که جهان ، قطعی باشد ، امّا با توجّه به پیچیدگی اش به نظر تصادفی (اتّفاقی) برسد .

دوره ای و ترتیبی بودن : در حالت دوره ای ، هر عملکرد به صورت مستقل می باشد و عامل ، ادراک می کند ، تصمیم می گیرد و عمل می نماید و دوباره [از نو] شروع می نماید و تصمیم گیری بعدی ، وابسته به وضعیّت های قبلی نمی باشد . به عنوان مثال ، عامل فیلترکننده ی اسپم [۱] ، دوره ای می باشد . در صورتی که عامل باید یک سری از عملیّات را برای رسیدن به یک عمل یا رسیدن به یک هدف انجام دهد ، محیط ترتیبی می باشد . در محیط ترتیبی ، باید به تصمیم گیری های آینده توجّه شود ؛ به عنوان مثال ، رانندگی یک اتومبیل ، ترتیبی می باشد .

ثابت و پویا (دینامیک) بودن : یک محیط ثابت ، مادامی که عامل در حال تصمیم گیری در مورد یک عملکرد می باشد ، ثابت باقی می ماند و عامل برای رسیدن به یک تصمیم ، تحت محدودیّت یا فشار زمانی نمی باشد ؛ محیط فیلترکردن اسپم ، ثابت می باشد ؛ بازی شطرنج ، ثابت می باشد . در یک محیط پویا ، در زمانی که عامل در حال تصمیم گیری در مورد این می باشد که چه کاری را انجام بدهد ، محیط تغییر پیدا می کند . عامل در این مورد باید " به اندازه ی کافی با سرعت " عمل نماید ؛ به طور مثال ، رانندگی یک اتومبیل ، پویا می باشد . در محیط **نیمه پویا** ، محیط تغییر نمی کند ، امّا معیارکارایی در طول زمان تغییر می کند . انجام یک آزمایش در زمان محدود و بازی شطرنج با یک ساعت مثال هایی از محیط های نیمه پویا می باشند .

گسسته و پیوسته بودن : ما می توانیم در مورد گسسته و پیوسته بودن ادراک ها ، عملکردهای عامل یا وضعیّت های ممکن محیط ، صحبت کنیم . در صورتی که مقادیر هر کدام از موارد گفته شده به صورت مجموعه ای گسسته باشد ، آن گاه محیط گسسته می باشد . محیط گسسته ، شبیه محیط محدود نمی

---

[۱] spam ، پیام های نامربوط یا نامناسبی می باشند که در اینترنت برای تعداد زیادی از کاربران یا گروه های خبری فرستاده می شوند .

مترجم: سهراب جلوه گر
ویرایش دوّم ، بهار ۱۳۸۸

باشد . یک محیط فیلترکننده ی اسپم ، گسسته می باشد ، حتی در میان تعداد (قابل شمارش) نامحدود از پیام های الکترونیکی . یک محیط پیوسته ، دارای متغیّرهای تغییر کننده به صورت پیوسته است . به طور مثال ، زاویه های فرمان در یک محیط رانندگی اتومبیل . در مورد حسگر خوان های با مقادیر واقعی (ما می توانیم در مورد ادراک ها در این مورد موشکافی نماییم ؛ در این جا نکته این است که آیا یک تغییر قابل تشخیص میان دو مقدار وجود دارد یا نه) .

تک عاملی یا چندعاملی بودن : در محیط تک عاملی ، عامل ما در حال عمل برروی خودش می باشد . به طور مثال ، محیط فیلترکردن اسپم ، به صورت تک عاملی می باشد . در محیط چند عاملی ، عملکردها / اهداف / روش های عامل های دیگر باید به حساب آیند . مثل محیط بازی شطرنج . *گرچه یک جهان ممکن است دارای عامل های دیگری باشد ، امّا ما ممکن است به خاطر پیچیدگی استنتاج ، با آن به صورت تک عاملی و اتّفاقی رفتار نماییم* . به عنوان مثال یک عامل کنترل کننده ی علایم ترافیکی .

برخی مثال ها عبارتند از : بازی شطرنج ، ماشینی که در آن پول می ریزند و کالایی را دریافت می کنند [۱] ، روبوتی که در هارنی [۲] به من قهوه می دهد ، مدار مریخ [۳] ، عامل مکالمه ای و عامل تشخیص طبی .

مثال :

	تاکسی	فروشگاه اینترنتی	تخته نرد
قابل مشاهده ؟؟	نه	نه	بله

[۱] slot-machine

[۲] Harney : نام جایی در ایالات متحده ی آمریکا

[۳] Mars orbiter

نتیجه ی فرایند ؟؟	نه	تاحدودی	نه
دوره ای ؟؟	نه	نه	نه
ایستا ؟؟	نه	نیمی	نیمی
مجزا ؟؟	نه	بله	بله
تک مولفه ای ؟؟	نه	بله (به جز حراج ها)	نه

نوع محیط تا حد زیادی طرح عامل را تعیین می کند . البتّه دنیای واقعی اندکی قابل مشاهده ، دوره ای ، ترتیبی ، پویا ، دنباله دار و چند عاملی می باشد.

ارزیابی عملکرد (معیار کارایی)

در مورد عامل جاروبرقی ، تعدادی از کاشی ها در یک زمان معین تمیز می شوند . انرژی ، اغتشاش [1] ، گم شدن قطعات مفید ، اثاثیه ی خراب ، خراش کف را هم در ارزیابی عملکرد در نظر می گیریم .

عامل های نرم افزاری ، معمولاً به صورت سافتبوت ها شناخته می شوند . در محیط های مصنوعی که براساس کامپیوترها و شبکه ها می باشند وجود دارند . می توانند خیلی پیچیده با نیازمندی های

[1] noise

مترجم: سهراب جلوه گر
ویرایش دوّم ، بهار ۱۳۸۸

زیاد به عامل باشند وب جهانی [۱] مثالی از عامل های نرم افزاری می باشند . در این عامل ها ، محیط های طبیعی و مصنوعی ممکن است با هم ادغام شده باشند .

عامل های متحرک [۲] ، در عامل های متحرّک ، برنامه ها می توانند از یک ماشین به ماشین دیگر بروند ، در یک محیط اجرایی مستقل از پایگاه [۳] اجرا می شوند ، به محیط اجرایی عامل نیازمندند ، تغییر پذیری ، لازم یا مناسب وضعیّت برای عامل نمی باشد . این عامل ها ، هزینه ی ارتباط را کاهش می دهند . کاربرد آن ها در بازیابی اطّلاعات توزیع شده و مسیریابی شبکه ی مخابراتی می باشد .

عامل های اطّلاعاتی [٤] ، رشد انفجاری اطّلاعات را مدیریّت می نمایند ، اطّلاعات چند منبع توزیع شده را به کار می گیرند یا مقایسه [۵] می نمایند . عامل های اطّلاعاتی می توانند ثابت یا متحرک باشند مثل ، اطّلاعات درون وب یا یک سند واقعی (مثلا یک سند کاغذی) ، برای صفحات تفسیر کننده ی وب شناخته می شوند و منبع داده ای در داده های بی ساختارند و در آن ها پاسخگویی به سؤال با استفاده از دانش قوی متدهای آماری انجام می شود .

برنامه های محیط ، شبیه سازی کننده های محیط برای آزمایش هایی که توسط عامل ها انجام می شود می باشند ، یک ادراک را به یک عامل ارایه می نمایند ، یک عملکرد را دریافت می نمایند و محیط را به روز می نمایند . اغلب به کلاس های محیط برای عملیّات وابسته یا انواع عامل ها تقسیم می شوند و خیلی از اوقات مکانیزم هایی را برای اندازه گیری عملکرد عامل ها ارایه می نمایند .

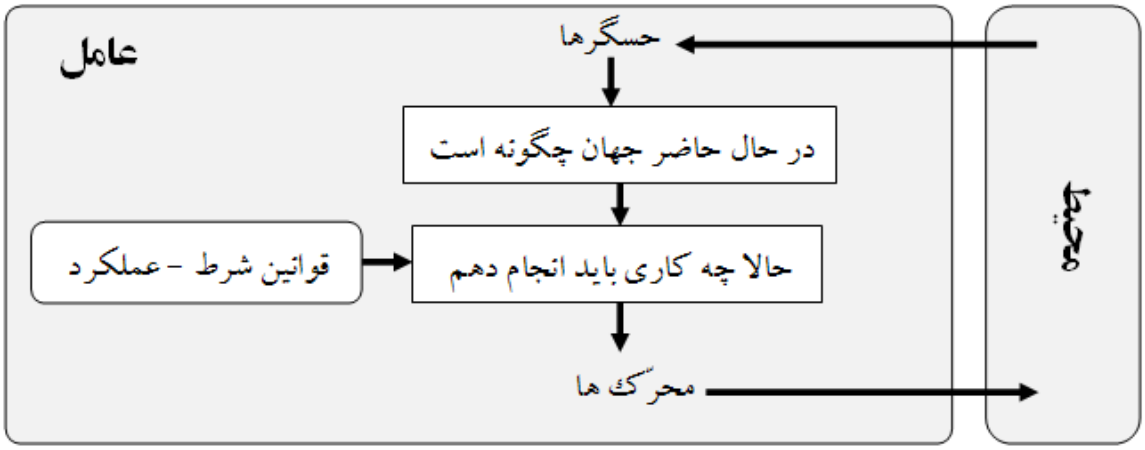
انواع عامل ها

---

[۱] World Wide Web
[۲] mobile agents
[۳] independent-platform
[٤] information agents
[۵] collate

چهار نوع اصلی وجود دارد : عامل های بازتابی ساده [۱] – عامل های بازتابی با حالت یا وضعیّت یا عامل هایی که وضعیّت جهان را حفظ می نمایند [۲] – عامل های براساس هـدف یا هدف گرا [۳] و عامل های سودمند [۴] . همه ی این ها می توانند در عامل های آموزشی بـه کـار گرفتـه شوند.

عامل های بازتابی ساده : دارای جدول جستجوی سـاده مـی باشـند و ادراک هـا را بـه عملیّـات نگاشت می نمایند (خیلی بزرگ و پرهزینه می باشند) و در آن ها تعدادی از وضـعیّت هـا مـی تواننـد توسـط قانون های شرط – عملکرد خلاصه شوند . پیاده سازی این نوع عامل ها آسـان مـی باشـد ولـی دارای کـاربرد کمی می باشند .



مثال : تابع Reflex-Vacuum-Agent([location,status]) یک عملکرد را برمی گرداند ، در صورتی که وضعیّت برابر کثیف (Dirty) بود ، مکش (Suck) را برمـی گردانـد ، در غیـر ایـن صـورت ،

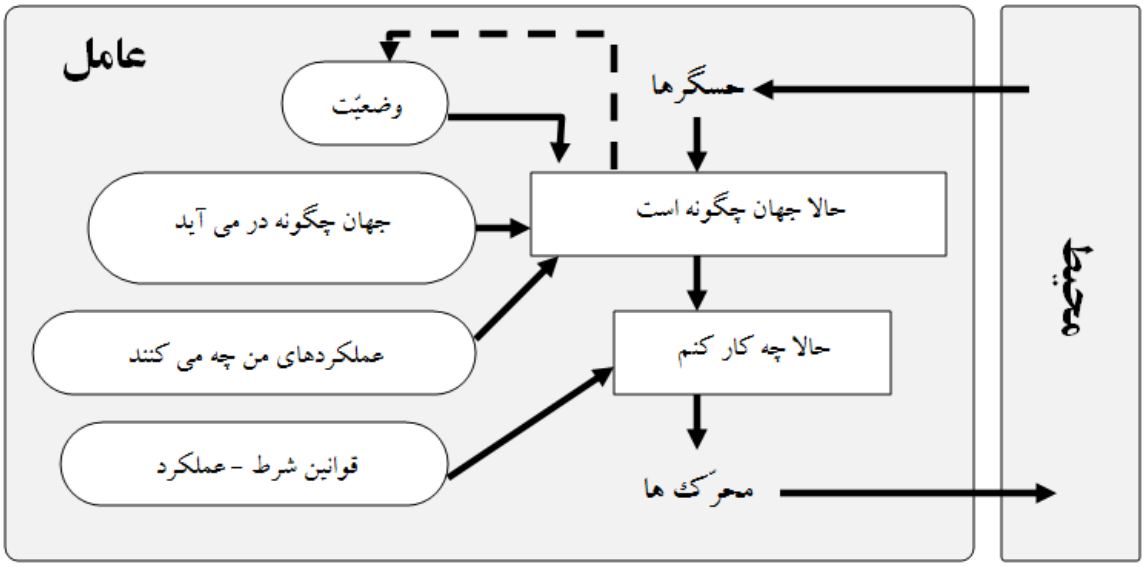
---

[۱] simple reflex agents
[۲] reflex agents with state
[۳] goal-based agents
[۴] utility-based agents

اگر محل [1] برابر A بود ، راست (Right) را برمی گرداند ، در غیر این صورت ، اگر محل برابر B بود ، چپ (Left) را بر می گرداند .

عامل هایی که وضعیّت جهان را حفظ می کنند (عامل های بازتابی براساس مـدل) :

اطّلاعات عامل به تنهایی در مورد مشاهده پذیری جزیی کـافی نمـی باشـند ، لازم اسـت کـه جریـان تغییـرات جهان را نگهداری نماییم و تکامل ها ، به طور مستقل از عامل یا سبب شده توسط عملکرد عامل می باشند .



مثال : تابع Reflex-Vacuum-Agent([location,status]) یک عملکرد را برمی گرداند

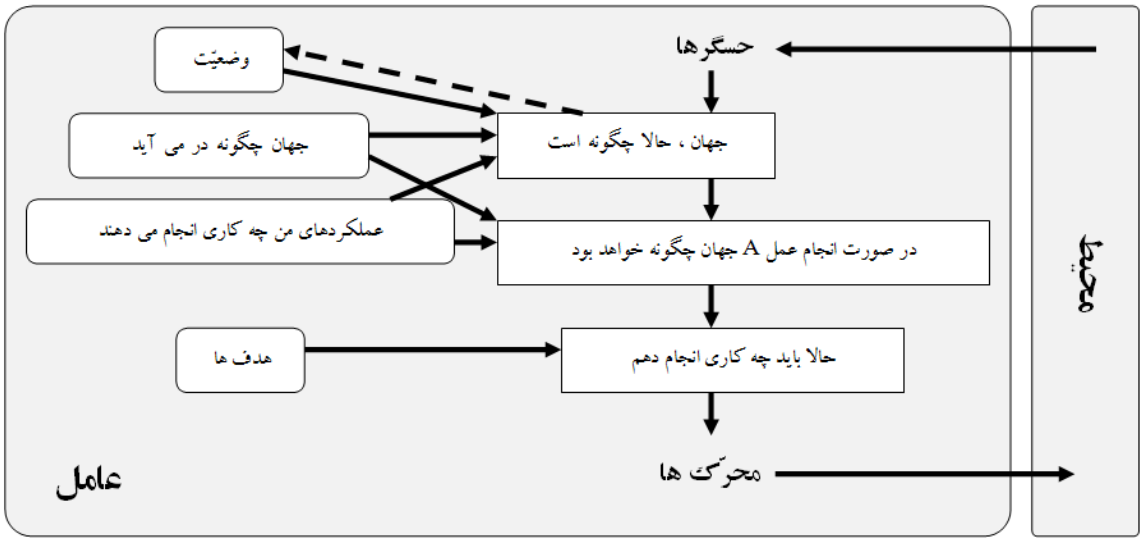
متغیّرهای static : last_A,last_B,numbers دارای مقدار اوّلیه ی ∞

در صورتی که وضعیّت (status) = کثیف (Dirty) باشد ...

[1] location

مترجم: سهراب جلوه گر
ویرایش دوّم ، بهار ۱۳۸۸

عامل های هدف گرا : در این گونه عامل ها وضعیّت و عملکردها نمی گویند که کجا برویم و به **اهداف** برای ساختن رشته ای از عملکردها (برنامه ریزی) نیازمندیم در هدف گرایی ، از قوانین یکسان برای اهداف مختلف استفاده می نماید و در بازتاب ، به مجموعه ای کامل از قوانین برای هر هدف نیاز خواهیم داشت .

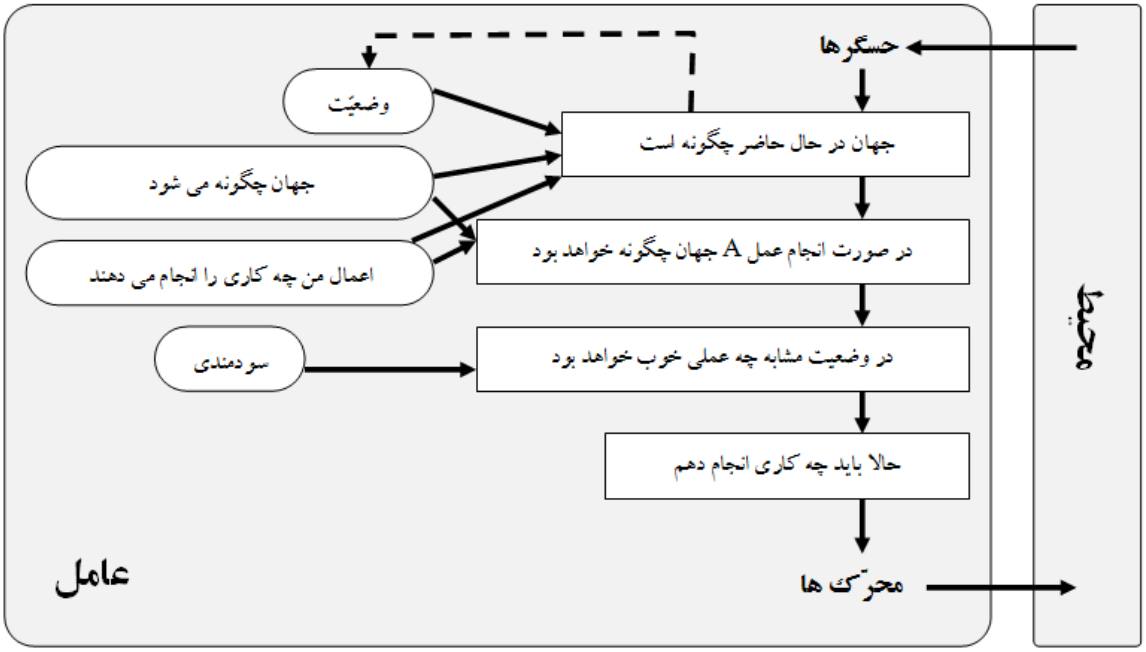


مطالبی در مورد عامل های هدف گرا : دانستن وضعیّت فعلی محیط ، همیشه کافی نمی باشد . عملکرد درست ممکن است همچنین وابسته به این که عامل در تلاش برای رسیدن به چه چیزی است باشد . عملکردهایی را که برای رسیدن به هدف ها کمک می کنند را انتخاب نمایید . جستجو و برنامه ریزی برای حل این مسأله مورد استفاده قرار می گیرند . استدلال هدف گرا برای محیط های ترتیبی خیلی مفید می باشد . مثل ، بازی شطرنج ، رانندگی تاکسی ، خلبانی سفینه ی فضایی . عملکرد درست برای یک رشته ی ادراکی ارایه شده وابسته به دانش عامل و وضعیّت جاری آن و چیزی که در حال حاضر در تلاش برای رسیدن به آن است می باشد .

عامل های سودمند : در عامل های سودمند ، دانستن هدف ها ممکن است در محیط های با پیچیدگی بالا ، کافی نباشند ؛ چند رشته از عملکردها برای دسترسی به برخی اهداف لازمند (پردازش

مترجم: سهراب جلوه گر
ویرایش دوّم ، بهار ۱۳۸۸

دودویی) ؛ ممکن است نیازمند انتخاب میان عملکردها و رشته ها می باشیم و سودمندی ، وضعیّت را به عددی حقیقی نگاشت می نماید و درجه ی ارضا را بیان می کند و سبک و سنگینی میان موارد متناقض را مشخص می نماید ، از سودمندی برای مقایسه ی شرایط مطلوب وابسته به رشته های عملکرد استفاده می شود و می تواند تخمینی از هزینه ، زمان ، یا مقدار وابسته به نتایج مختلف باشد . سودمندی ها در محیط هایی که به صورت جزیی قابل مشاهده اند یا محیط های اتّفاقی ، خیلی مفید می باشند . سودمندی ها برخی اوقات یک چیز بحث انگیز در هوش مصنوعی می باشند . فرض می کنیم نتایج می توانند به صورت خطی در یک مقیاس یکسان ، مرتّب شده باشند . مثل : عامل رانندگی تاکسی . در این مورد ، طراح باید نتایج را به طور صریح ارزیابی نماید (به صورت کمّی و کیفی) . سودمندی در دامنه های با احتمال ، خیلی مفید می باشد ، مثل : تجارت برخط ، اکتشاف در محیط های دارای نامعلومی و قماربازی [۱] .



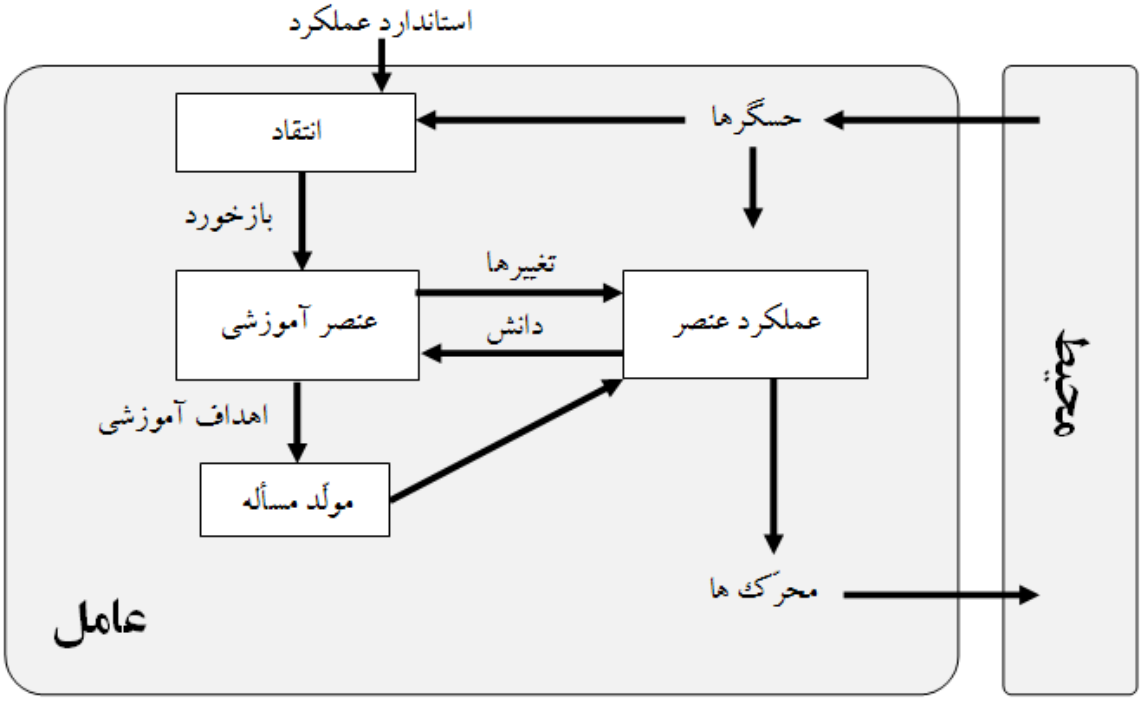
[۱] gambling

مترجم: سهراب جلوه گر
ویرایش دوّم ، بهار ۱۳۸۸

عامل های آموزشی : در این عامل ها عنصر آموزشی [۱] ، بهبودهایی را به وجود می آورد . عنصر کارایی [۲] ، انتخاب عملکردهای خارجی را ممکن می سازد . انتقاد [۳] ، جمع آوری بازخورد [۴] در مورد چگونگی عمل عامل می باشد . تولید کننده ی مسأله [۵] ، عملکرد (آزمایش) های پیشنهادی (اکتشافی) را در نظر می گیرد . اغلب ، یک عامل ممکن است نیاز به بروزرسانی برنامه ی عامل خود را داشته باشد . برنامه نویسان ممکن است فهم کاملی از محیط نداشته باشند . محیط ، ممکن است در طول زمان ، تغییر نماید در این مورد برنامه نویسی با دست ممکن است خسته کننده باشد . یک عامل آموزشی ، عاملی است که عملکرد خود را با توجّه به مجموعه ای از عملکردها در طول زمان ، بهبود می بخشد . آموزش یا انطباق در محیط های پیچیده ، ضروری می باشد . یک عامل آموزشی هم به *عنصر عملکرد یا کارایی* [۶] و هم به *عنصر آموزشی* [۷] نیازمند می باشد ؛ عنصر کارایی ، عملکرد یا عملکردهای جاری را انتخاب می نماید و عنصر آموزشی ، درستی عنصر کارایی را ارزیابی می نماید .

---

[۱] learning element
[۲] performance element
[۳] critic
[۴] feedback
[۵] problem generator
[۶] performance element
[۷] learning element



آموزش ، می تواند به صورت برخط [۱] یا برون از خط [۲] باشد ؛ آموزش ، ممکن است *غیرفعال* یا *فعال* باشد ؛ آموزش ممکن است نظارت شده یا نظارت نشده باشد . *انتساب اعتبار* [۳] در زمانی که آموزش در محیط های ترتیبی می باشد ، مسأله ای بزرگ می باشد .

نکته : اغلب ، آموزش در هوش مصنوعی به صورت یک موضوع مجزا ، عمل می کند ؛ ما برای جمع کردن آن با دیگر موضوعات ، تلاش می کنیم .

خلاصه

---

[۱] online

[۲] offline

[۳] credit assignment

مترجم: سهراب جلوه گر
ویرایش دوّم ، بهار ۱۳۸۸

عامل ها **، روی** ***محیط ها*** **به وسیله ی** ***محرّک ها*** **و** ***حس گرها*** **اثر مـی گذارنـد. تـابع عامل چگونگی عملکـرد عامـل را در همـه ی شـرایط تشـریح مـی کنـد. ارزیـابی عملکـرد (معیارکارایی) چگونگی محیط را ارزیابی می کند . یک عامل هوشمند بـی عیـب ، عملکـرد مورد انتظار را بیشینه می کند . برنامه های عامل برخی از توابـع عامـل را اجـرا مـی کننـد. توضیحات** ***PEAS*** **، محیط های کاری را تعریف می کند.**

محیط ها در چند بعد طبقه بندی می شوند: ***قابل مشاهده ؟ - قطعی ؟ - دوره ای ؟ - پویا ؟ - گسسته ؟ - تک مولفه ای ؟***

چند معماری موجود عامل های پایه عبارتند از: ***عامـل هـای بازتـابی سـاده- عامـل هایی که وضعیّت جهان را حفظ می نمایند- عامل های هدف گرا - عامل های سودمند***

فصل سوّم

حل مسأله و جستجو [1]

[1] problem solving and search

مترجم: سهراب جلوه گر
ویرایش دوّم ، بهار ۱۳۸۸

خلاصه ی ریوس مطالب

- **عامل های حل کننده ی مسأله**
- **انواع مسأله**
- **فرمول بندی [۱] مسأله**
- **مثال هایی از مسایل**
- **الگوریتم های اساسی (پایه ای) جستجو**

یک عامل هدف گرا می تواند برّرسی کند که چه کاری را می تواند انجام دهـد و عملکردهـایی کـه به نتیجه می رسند را انتخاب نماید . برنامه ی عامل از درک ها و هدف ، به صورت ورودی استفاده می کنـد . ما به یک نوع عامل هدف گرا به نام عامل حل مسأله را برّرسی می کنیم .

---

[۱] formulation

مترجم: سهراب جلوه گر
ویرایش دوّم ، بهار ۱۳۸۸

جستجو به عنوان روش حلّ مسأله

بسیاری از مسایل می توانند به صورت رسیدن به یک حالت هدف [۱] ، از یک نقطه ی شروع دیده شوند . فضای حالت [۲] ، مسأله و راه حل های ممکن آن را به یک صورت رسمی تری تعریف می نماید ؛ معمولاً مسایل باید به صورت تجزیه شده [۳] باشند . عملکردهای عامل ، وضعیّت را تغییر می دهد و فضای حالت را برای یک راه حل ، جستجو (ملاقات) می نماید . اطّلاعات ، در مورد مسأله ی معین یا با دامنه ی عمومی می توانند برای بهبود جستجو استفاده شوند . این اطّلاعات عبارتند از : تجربه از موارد قبلی مسأله ؛ روش های بیان شده به صورت اکتشافات ؛ نوع های ساده تر مسأله و محدودیّت های وضعیّت های مشخّص (معیّن) مسأله .

چرا روش های جستجو را برّرسی می نماییم ؟

روش های جستجو ، روش هایی مهم برای حلّ بسیاری از مسایل می باشند . استفاده از جستجو ، به یک فرمول بندی مجزای مسأله و مراحل ممکن برای به وجود آوردن راه حل ها نیازمند می باشد و الگوریتم های جستجو ، پایه ای برای بهینه سازی [۴] و روش های برنامه ریزی می باشند .

عامل های حل کننده ی مسأله

یک عامل حل کننده ی مسأله ، برای پیداکردن یک رشته از عملکردها که به هدف می رسند تلاش می کند . به عنوان مثال ، چه رشته از حرکت ها یک مکعّب رابیک [۱] را حل می کنند ؟ . چگونه من از

---

[۱] goal state
[۲] state space
[۳] abstracted
[۴] optimization

هوش مصنوعی

مترجم: سهراب جلوه گر
ویرایش دوّم ، بهار ۱۳۸۸

دانشگاه فلوریدای جنوبی [۲] با اتومبیل به سمت لیورمور رانندگی نمایم ؟ . چگونه من می توانم اجزای روی یک تراشه را بچینم ؟ . کدام رشته از عملیّات یک روبوت را در یک فضا حرکت می دهد ؟ . در زیر تصویری از بازی مکعّب رابیک را مشاهده می نمایید :

برای فرمول بندی مسأله این موارد را در نظر می گیریم : وضعیّت های ممکن وابسته به دنیا برای حل مسأله کدامند ؟ . چه اطّلاعاتی برای عامل در دسترس می باشند ؟ . رفتن عامل از یک وضعیّت به وضعیّت دیگر چگونه می تواند باشد ؟ .

برای فرمول بندی هدف مسأله این موارد را در نظر می گیریم : وضعیّت هدف چیست ؟ . ویژگی های مهم وضعیّت هدف چه می باشند ؟ . چگونه عامل می فهمد که به هدف رسیده است ؟ در این مورد برّرسی می کند که آیا چند وضعیّت پایانی ممکن وجود دارد ؟ و آیا آن ها با هم برابرند یا برخی بهترند ؟ .

جستجو

---

[۱] Rubik's cube ، یک پازل به شکل یک مکعب پلاستیکی که با مربع هایی با چند رنگ پوشانده شده است ، هر بازیگر تلاش می کند مربع ها را تغییر دهد تا این که همه ی مربع های هر وجه دارای یک رنگ بشوند [انتشارات دانشگاه آکسفورد ، سال ۲۰۰۴ میلادی ، گرفته شده از لغتنامه ی Babylon]

[۲] University of South Florida = USF

مترجم: سهراب جلوه گر
ویرایش دوّم ، بهار ۱۳۸۸

تعریف جستجو : پردازه ی رسیدگی کردن به صورت ترتیبی به عملکردها برای پیداکردن رشته ای از عملیّات که ما را از شروع به هدف می رسانند ، جستجو نام دارد . یک الگوریتم جستجو ، یک رشته از عملیّات که برای عامل اجرا می شوند را برمی گرداند و جستجو معمولاً به صورت "برون خطی " انجام می شود . در این مورد نکته ای که باید در نظر بگیریم این است که ، فرض بر این است که محیط به صورت استاتیک (ثابت) می باشد ، همچنین فرض شده که محیط ، گسسته می باشد . محیط (معمولاً) به صورت قطعی می باشد .

برخی از مسایل جستجوی کلاسیک

مسایل سرگرمی : برای مطالعه به صورت مثال یا برای مقایسه ی الگوریتم ها ، مفید می باشد ؛ مثل : پازل ۸ – تایی ، دنیای جارو ، مکعّب رابیک و N – وزیر .

مسایل دنیای واقعی : معمولاً کار زیادی می برند ، امّا جواب معمولاً جالب می باشد . مثل : مسیر یابی ، مسافرت شخص دوره گرد ، طرح VLSI و جستجو در اینترنت

وضعیّت

ما معمولاً در مورد وضعیّتی که یک عامل در آن می باشد صحبت می کنیم . وضعیّت ، به مقادیر متغیّرهای مربوط توصیف کننده ی محیط و عامل ، اشاره می کند . به عنوان مثال در دنیای جارو ، وضعیّت (۰و۰) ، تمیز می باشد . در مسأله ی رومانی ، در $t=0$ در بخارست هستیم و در مکعّب رابیک ، وضعیّت ، نظم فعلی مکعّب (چگونگی قرار گرفتن مربّع های روی مکعّب در حال حاضر) می باشد .

فرمول بندی مسأله

فرمول بندی مسأله شامل موارد زیر است :

مترجم: سهراب جلوه گر
ویرایش دوّم ، بهار ۱۳۸۸

۱- **شناختن نوع مسأله :** این که عامل چه دانشی در مورد وضعیّت جهان و نتیجه ی عملکرد خودش دارد ؛ آیا اجرای عمل نیازمند به روز کردن اطّلاعات دارد ؟ و تعریف وضعیّت های جهان .

۲- **توضیح رسمی برای عملکرد عامل :** توضیح هدف و عملگرها که همان عملکردهای عامل می باشد .

انتخاب وضعیّت ها و عملکردها

مسایل برای حل باید **مجزا** باشند . وضعیّت های قابل شناخت در مدت فرایند حل مسأله باید توضیح داده شوند . عملکردها (عملگرها) عامل را از وضعیّتی به وضعیّت دیگر می برند و وابسته به وضعیّت ها ، توانایی عامل و خصوصیات محیط می باشند و انتخاب وضعیّت ها و عملگرهای مناسب ، می تواند باعث تفاوت میان این که یک مسأله می تواند حل شود یا نمی تواند حل شود بشود .

انواع مسأله

مسایل تک حالته [۱] **:** در صورتی که مسأله قطعی و کاملاً قابل مشاهده باشد ، در نتیجه ، مسأله ، تک حالته خواهد بود ؛ در این حالت ، عامل دقیقاً می داند در کدام حالت قرار خواهد گرفت و راه حل به صورت ترتیبی است . در این حالت ، جهان در دسترس می باشد و اثر عملکردها را می دانیم و عامل ، وضعیّتی که بعد از یک رشته از عملیّات در آن خواهد بود را می داند .

مسایل چندحالته [۲] **:** در صورتی که مسأله غیرقابل مشاهده باشد ، در نتیجه مسأله ، تطبیقی (ترکیبی [۳]) خواهد بود . عامل ، ممکن است تصمیمی در مورد کجا قرار گرفتن نداشته باشد ؛ راه حل (در

---

[۱] single-state problems

[۲] multiple-state problems

[۳] conformant

صورت وجود) ترکیبی می باشد . در مسایل چند حالته ، جهان فقط تا حدودی در دسترس می باشد و عامل به چند وضعیّت ممکن توجّه دارد .

مسایل احتمالی [۱] **:** در صورتی که مسأله ، احتمالی (غیرقطعی [۲]) و یا اندکی قابل مشاهده [۳] باشد ، در نتیجه ، مسأله ی احتمال می باشد . در مسایل احتمالی ، در طول اجرا به مشاهده و حس کردن نیازمندیم . همچنین به درخت عملیّات هم نیاز داریم .

مسایل اکتشافی [۴] **:** در صورتی که مسأله دارای فضای حالت ناشناخته باشد ، در نتیجه ، مسأله ، اکتشافی می باشد . در مسایل اکتشافی ، عامل نتایج عملکردش را نمی داند و آزمایش های عامل برای کشف وضعیّت های جهان و اثرات عملکردها می باشد .

مسایل خوب تعریف شده

در مسایل خوب تعریف شده ، وضعیّت اوّلیه ، نقطه ی شروع که عامل از آن شروع می کند می باشد ، مثلا در مورد مسأله ی کشور رومانی ، وضعیّت اوّلیه ، شهر آراد می باشد . فضای حالت ، مجموعه ای از همه ی وضعیّت هایی که از وضعیّت اوّلیه توسط رشته ای از عملکردها قابل دسترسی می باشند است و مسیر ، رشته ای از عملکردها که راهنمایی کننده از یک وضعیّت به وضعیّت دیگر می باشد هستند . عملکردها (عملگرها و توابع جانشین [۵]) می گویند که عامل چه عملکردهایی را می تواند داشته باشد ؟ . عملکردها ،

---

۱ contingency problems
۲ nondeterministic
۳ partially observable
۴ exploration problems
۵ successor functions

توضیح مجموعه ای از عملکردهای ممکن می باشد ، مثلا در مورد جارو ، چپ ، راست ، بالا ، پایین ، مکش و هیچ کار می تواند باشد . آزمایش هدف ، تشخیص می دهد که آیا یک وضعیّت ارایه شده ، یک وضعیّت هدف می باشد یا نه ؟ و راه حل ، مسیری از وضعیّت اوّلیه به یک وضعیّت هدف می باشد .

تابع جانشین

تابع جانشین برای یک وضعیّت داده شده ، یک مجموعه از جفت های عملکرد / وضعیّت جدید را برمی گرداند و به ما می گوید که برای یک وضعیّت داده شده ، چه کارهایی را می توانیم انجام دهیم و در چه جاهایی مقدم هستند . در یک جهان قطعی ، هر عملکرد با یک وضعیّت منفرد ، جفت شده است . مثلا در دنیای جارو برقی :

$$(In(0,0)) \rightarrow (\text{'Left'},In(0,0)),(\text{'Right'},In(0,0)),(\text{'Suck'},In(0,0),\text{'Clean'})$$

در مسأله ی رومانی :

$$In(Arad) \rightarrow ((Go(Timisoara),In(Timisoara),(Go(Sibiu),In(Sibiu)),(Go(Zerind),In(Zerind))$$

در جهان های اتّفاقی ، یک عملکرد ، شاید با تعدادی وضعیّت ها ، جفت شده باشد .

آزمون هدف

آزمون هدف ، تشخیص می دهد که آیا یک وضعیّت داده شده ، یک وضعیّت هدف است . در این مورد باید توجّه داشته باشیم که شاید یک وضعیّت هدف منحصر به فرد وجود داشته باشد یا تعدادی . مثلا در دنیای جاروبرقی ، هر فضای تمیز در بازی شطرنج ، مات و در مسأله ی رومانی ، در بخارست بودن وضعیّت هدف می باشند .

هزینه ی مسیر

مترجم: سهراب جلوه گر
ویرایش دوّم ، بهار ۱۳۸۸

هزینه ی مسیر ، هزینه ی یک عامل برای رفتن از وضعیّت اوّلیه به وضعیّتی که به تازگی برّرسی شده است می باشد و اغلب ، مجموع هزینه برای هر عملکرد می باشد و هزینه ی مرحله [۱] نام دارد . ما فرض می کنیم که هزینه های مراحل ، مثبت (غیرمنفی) هستند .

فضای حالت

ترکیب حالت های مسأله و توابع جانشین (راه های رسیدن به وضعیّت ها) منجر به مفهوم فضای حالت می شود و گرافی است که همه ی وضعیّت های ممکن جهان و انتقال های میان آن ها را ارایه می نماید . ما اغلب در مورد اندازه ی این فضاها به صورت یک معیار سختی مسأله صحبت می کنیم . به عنوان مثال در پازل ۸- تایی فضای حالت برابر است با : $\frac{9!}{2} = 181{,}000$ حالت که فضای حالتی ساده است . در مورد پازل ۱۵ – تایی ، فضای خالت برابر است با : تقریبا ۱.۳ تریلیون [۲] حالت که تقریبا ساده است . در مورد پازل ۲۴- تایی : تقریبا 10^{25} وضعیّت که فضای حالتی پیچیده است و در مسأله ی مسافرت شخص دوره گرد : !۲۰ = 2.43×10^{18} وضعیّت که خیلی پیچیده می باشد .

عملیّات حل مسأله

هزینه ی مسیر ، هزینه ی عامل را برای اجرای عملیّات در یک مسیر تعیین می نماید و مجموع هزینه های عملیّات فردی در یک مسیر می باشد . هزینه ی جستجو ، زمان و حافظه ی لازم برای محاسبه ی یک راه حل است و داریم مجموع هزینه = هزینه ی مسیر + هزینه ی جستجو . برای مسایل چند حالته ما از **مجموعه ای از وضعیّت ها** استفاده می نماییم و هزینه ها و دیگر تعریف ها تغییر نمی کنند . شکل محدود شده ی کلّی ، دارای الگوریتم زیر می باشد :

---

[۱] step cost

[۲] در انگلیسی برابر است با عدد ۱ که ۱۸ صفر در جلوی آن قرار داده شود . (گرفته شده از Babylon)

تابع Simple-Problem-Solving-Agent(percept) یک عملکرد را بر می گرداند

متغیّر static :

seq ، که یک ترتیب عملکرد است و در ابتدا دارای مقدار تهی می باشد

state ، توصیف وضعیّت فعلی جهان

goal ، یک هدف که به صورت اوّلیه دارای مقدار NULL می باشد

problem ، یک فرمول بندی مسأله

state←Update-State(state,percept)

در صورتی که seq ، خالی است کارهای زیر را انجام بده

goal←Formulate-Goal(state)

problem ← Formula-Problem(state,goal)

seq ← Search(problem)

(پایان شرط)

action ← Recommendation(seq,state)

seq ← Remainder(seq,state)

عمل (action) را برگردان

مترجم: سهراب جلوه گر
ویرایش دوّم ، بهار ۱۳۸۸

نکته : این ، راه حلی **برون خطی** [۱] می باشد ؛ حل به صورت " چشم بسته " اجرا می شود. راه حل های **برخط** [۲] شامل عمل کننده ی بدون دانش تمام می شوند .

استراتژی های جستجو - یک استراتژی با چیدن گره ها به صورت مرتّب تعریف می شود . استراتژی ها به وسیله ی موارد زیر ارزیابی می شوند : تمامیّت [۳] – آیا همیشه ، اگر راه حل موجود باشد آن را پیدا می کند ؟ | پیچیدگی زمانی – تعداد گره های به وجود آمده / توسعه داده شده | پیچیدگی فضا – بیشینه ی تعداد گره های موجود در حافظه | بهینگی - آیا همیشه ، کم هزینه ترین راه را پیدا می کند ؟ | زمان و پیچیدگی فضا توسط پارامترهای زیر ارزیابی می شوند : b – ماکزیمم عوامل منشعب شده از درخت جستجو . d – عمق کم هزینه ترین راه حل . m – ماکزیمم عمق فضای حالت (ممکن است بی نهایت باشد) .

روش های ناآگاهانه [۴] - این روش ها فقط از داده های قابل دسترس از تعریف مسأله استفاده می کنند .

پیچیدگی

در مورد پیچیدگی الگوریتم ها نگران هستیم ، زیرا یک مسأله ممکن است در حالت کلّی قابل حل باشد امّا در حالت عملی خیلی طولانی باشد . با استفاده از تحلیل مجانبی [۵] ، تخمین زمان (یا تعداد عملیّات) لازم برای حل یک نمونه از اندازه ی n یک مسأله در زمانی که n به سمت بی نهایت می رود می توان پیچیدگی الگوریتم ها را ارزیابی نمود .

---

۱ offline

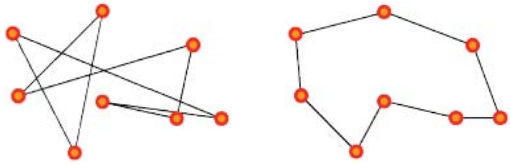
۲ online

۳ completeness

۴ uninformed search

۵ asymptotic

مثال : مسأله مسافرت فروشنده ی دوره گرد [۱] – در این مسأله ، n شهر توسط جاده به هم متصل می باشند و مسأله ، پیدا کردن یک مسیر که از تمام شهرها عبور کند و حتی الامکان کوتاه ترین مسیر باشد است .



سختی ما در این مورد این است که الگوریتم های شناخته شده دارای پیچیدگی نمایی هستند و تعداد عملیّات به صورت نمایی رشد می کند .

چرا پیچیدگی نمایی " مشکل " است ؟

این به آن معنی است که تعداد عملیّات لازم برای محاسبه ی راه حل دقیق مسأله به صورت نمایی با اندازه ی مسأله رشد می کند (در زیر تعداد شهر ها آورده شده است) .

- $\exp(1)=2.72$
- $\exp(10)=2.20\times10^{4}$
- $\exp(100)=2.69\times10^{43}$
- $\exp(500)=1.40\times10^{217}$
- $\exp(250{,}000)=10^{108{,}573}$

در حالی که در کامپیوترهای سریع ما 10^{12} عملیّات در ثانیه را ما داریم . **بنابراین** ، در حالت کلّی ، مسایل با پیچیدگی نمایی در همه ی موارد قابل حل نمی باشند امّا در اندازه ها ی کوچک نمایی قابل حل می باشند !

---

[۱] Traveling Salesperson Problem (TSP)

مترجم: سهراب جلوه گر
ویرایش دوّم ، بهار ۱۳۸۸

پیچیدگی

مسایل با زمان چندجمله ای [۱] (P) : ما می توانیم الگوریتم هایی را پیدا کنیم که این مسایل را در یک زمان که به صورت چندجمله ای با اندازه ی ورودی رشد می کنند حل کنیم . برای مثال ، در مرتّب کردن n عدد به صورت صعودی ، الگوریتم های ضعیف این کار را با پیچیدگی n^2 انجام می دهند ، الگوریتم های بهتر این کار را در $n \log(n)$ انجام می دهند . مسأله در این جا این است که ما نمی توانیم وضعیّت مرتبه ی چند جمله ای را مشخص کنیم ، شاید خیلی بزرگ باشد ! حال سؤال این است که آیا الگوریتم هایی که به بیش از زمان چندجمله ای نیاز داشته باشند هم وجود دارند ؟ . برای برخی از مسایل ، ما هیچ الگوریتم چند جمله ای نداریم که مسایل غیرقطعی چندجمله ای [۲] جزء این دسته هستند . برای مثال ، مسأله ی مسافرت فروشنده ی دوره گرد این گونه است .

نکته ای در مورد مسایل NP-hard

یک مسأله به صورت چندجمله ای غیرقطعی است اگر یک راه حل (گمان) برای آن مسأله پیدا شده باشد . به عبارت دیگر ، الگوریتمی وجود داشته باشد که دارای زمان چند جمله ای باشد ؛ چه گمان درست باشد و چه نباشد . دشواری ما این جاست که شاید هیچ الگوریتم چندجمله ای که بتواند مسأله را حل کند وجود نداشته باشد و در صورت وجود هم ممکن است نمایش آسان نباشد و معمولاً قابل کاهش به صورت مسایل شناخته شده هم نمی باشد . در عمل ، با فرض این که $P \neq NP$ ، الگوریتم ها زمان بیش تر از چند جمله ای برای حل مسأله دارند .

به دست آوردن پیچیدگی با استفاده تحلیل مجانبی

---

[۱] polynomial-time

[۲] nondeterministic-polynomial-time(NP)

مترجم: سهراب جلوه گر
ویرایش دوّم ، بهار ۱۳۸۸

اگر اندازه ی ورودی مسأله برابر n باشد ، تعداد عملیّات برای حل مسأله ، f(n) باشد و g(n) یک تابع ارایه شده باشد . برای برخی از مقادیر k ، تعریف f(n) ، در صورتی که f(n)≤k g(n) باشد ، برابر O(g(n)) می باشد . به عنوان مثال ، $O(n^2)$ بدتر از O(n) می باشد امّا برای n<110 داریم $n^2 + 1 < 100n + 1000$.

یادآوری : مسایل خوب تعریف شده

در مسایل خوب تعریف شده ، وضعیّت اوّلیه ، نقطه ی شروعی که عامل از آن آغاز می نماید است . فضای حالت ، مجموعه ی همه ی وضعیّت های قابل دسترس از وضعیّت اوّلیه توسط هر رشته از عملیّات می باشد و مسیر ، رشته ای از عملیّات مرتبط از یک وضعیّت به وضعیّت دیگر می باشد . عملیّات (عملگرها ، توابع جانشین) ، تشریح مجموعه ای از عملیّات ممکن است . آزمون هدف ، این مطلب را که آیا وضعیّت ارایه شده یک وضعیّت هدف است را تشخیص می دهد . راه حل ، مسیری از یک وضعیّت اوّلیه به یک وضعیّت هدف می باشد .

یادآوری : جستجو برای راه حل ها

برای پیدا کردن راه حل ها ، فضای جستجو را برّرسی می کنیم و از وضعیّت اوّلیه به وضعیّت نهایی (هدف) حرکت می کنیم که رشته ای مجاز از عملیّات به صورت تعریف شده توسط توابع جانشین (عملگرها) می باشد .

مثال : رومانی [۱] – در یک تعطیلی در کشور رومانی ؛ در حال حاضر در شهر آراد [۲] هستیم و هواپیما فردا آراد را به مقصد شهر بخارست [۱] ترک می کند و ما می خواهیم در شهر بخارست باشیم.

---

[۱] Romania

[۲] Arad