



هوش مصنوعی

فصل هشتم

منطق مرتبه اول

ایمان مختاری
دانشگاه پیام نور شهرکرد

مقدمه

منطق مرتبه اول یا FOL در جهت برطرف کردن مشکلات منطق گزاره ای به وجود آمده است و اساس کار بسیاری از زبانهای برنامه نویسی است.

خصوصیات منطق گزاره ای

۱. ماهیت اعلانی و توصیفی: معنای آن به رابطه درستی بین جملات و دنیاهای ممکن وابسته است دانش و استنتاج از هم مجزا هستند و استنتاج وابسته به دامنه است
۲. ماهیت ترکیبی و سازگاری:

ترکیب یعنی، معنای یک جمله مرکب با توجه به جملات سازنده آن تعیین می شود
سازگاری معنای هر جمله تابعی از خصوصیات اجزای آن است

۳. مستقل از متن: دنیا از حقایق تشکیل شده و بر خلاف زبان طبیعی ابهام در آن وجود ندارد؛ لذا عبارتی مثل "سپس او آنرا دید" در آن مبهم است .

در زبان طبیعی یکبار برای همیشه می گوییم مربعهای مجاور گودال دارای نسیم هستند ولی در زبان گزاره ای به صورت جداگانه برای هر مربع این قانون را بیان می کنیم

خصوصیات منطق مرتبه اول

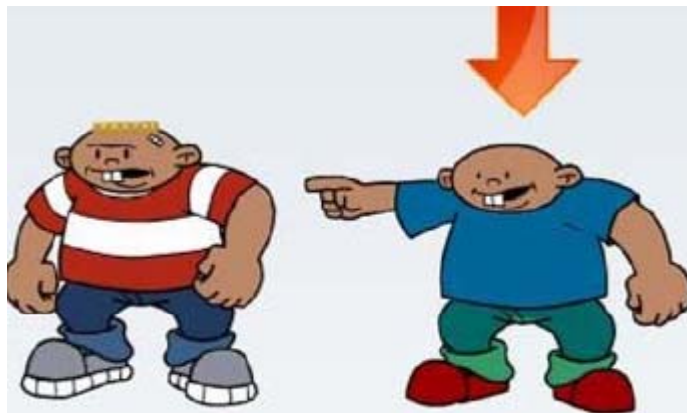
در حالیکه منطق گزاره ای فرض می کند دنیا از حقایق تشکیل شده است، منطق مرتبه اول (مانند زبان طبیعی) فرض می کند دنیا شامل موارد زیر است:

اشیا - روابط - توابع

دنیا در منطق مرتبه اول به جای حقایق شامل :



۱. اشیا: دنیا از اشیا تشکیل شده است که به خاطر خصوصیاتشان از یکدیگر قابل تشخیص اند
مثل : اشخاص ، منازل ، اعداد ، نظریه ها ، جنگ ها و



۲. روابط: بین اشیا روابط وجود دارد که روابطی که یک خاصیت را به یک شی نسبت می دهند (یکانی) یا دو یا چند شی را به هم مرتبط می کنند. مثل : رابطه برادری، قرمز، گرد بودن ، پسر بودن
مثلی علی با پسر بودن - پرسپولیس با قرمز بودن رابطه دارد

۳. توابع (حقایق): برای بیان انتساب مقدار به یک شی یا برگرداندن خصوصیت از یک شی استفاده می شود

Father (x, y)

Brother (x, y)

LeftLegOf (KingJohn)

مثال

- Sum(1, 2, 3), Even(2), Odd(3), ...
- Parent (Bob, Jim), Male(Bob), ...
- Add(1, 2), LeftLegof(John), ...

مهمترین تفاوت منطق گزاره ای و مرتبه اول

تعهد هستی شناسی: یعنی آن چیزی که هریک در باره ماهیت واقعیت فرض می کنند

انواع منطق ها:

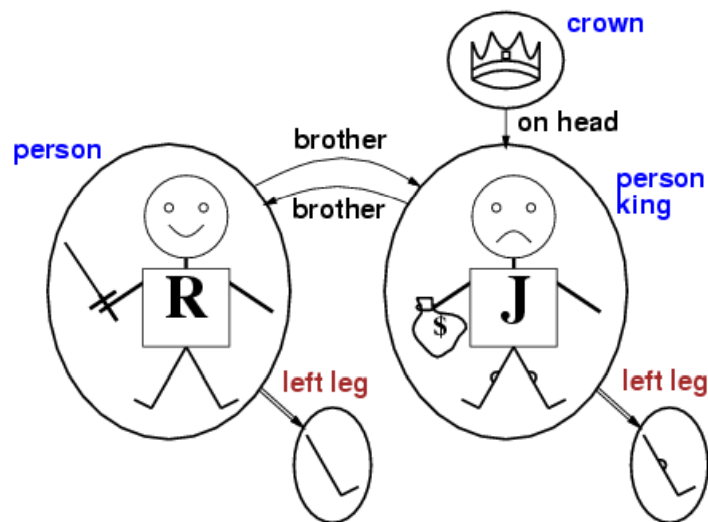
در منطق گزاره ای ، واقعیتها یا برقرارند یا برقرار نیستند
در منطق مرتبه اول ، ارتباط بین اشیا یا برقرار است یا برقرار نیست
در منطق لحظه ای ، واقعیتها در زمان خاصی برقرارند یا برقرار نیستند
در منطق احتمالات با مشاهده واقعیتها ، درجه ای از اعتقاد بین ۰ تا ۱ به وجود آید
در منطق فازی واقعیتها را نیز در بازه ۰ تا ۱ درک می کنیم

زبان	تعهدات هستی شناسی (آنچه که در دنیا وجود دارد)	تعهدات معرفت شناسی (آنچه که یک عامل درباره واقعیتها اعتقاد دارد)
منطق گزاره ای	واقعیتها	درست / غلط / نامعلوم
منطق مرتبه اول	واقعیتها، اشیا، روابط	درست / غلط / نامعلوم
منطق لحظه ای	واقعیتها، اشیا، روابط، زمانها	درست / غلط / نامعلوم
نظریه احتمالات	واقعیتها	درجه ای از اعتقاد بین $[0,1]$
منطق فازی	حقایق با درجه ای از درستی بین $[0,1]$	بازه ای از مقادیر معلوم

نحو و معنا در منطق مرتبه اول

□ مدل‌های منطق مرتبه اول

این مدل‌ها مجموعه از اشیا و روابط را نمایش می‌دهند
مثال:



✓ پنج شیء (پای چپ جان، پای چپ ریچارد، تاج، جان، ریچارد)

✓ دو رابطه دودویی (۱-تاج و جان) (۲-جان و ریچارد)

✓ سه رابطه یگانی (خواص: شخص بودن، پادشاه بودن، تاج بودن)

✓ یک تابع (خروجی یک شی) یگانی به نام پای چپ

ثابت‌ها (KingJohn, ۲, ...)

مسندها (Brother, >, ...)

توابع (Sqrt, LeftLegOf, ...)

متغیرها (x, y, a, b, ...)

رابط‌های منطقی ($\Leftrightarrow, \vee, \wedge, \Rightarrow, \neg$)

تساوی (=)

سورها (\forall, \exists)

ساختار جملات در منطق مرتبه اول

Sentence \rightarrow AtomicSentence

- | Sentence Connective Sentence
- | Quantifier Variable, ... Sentence
- | \sim Sentence
- | (Sentence)

AtomicSentence \rightarrow Predicate(Term, ...) | Term = Term

Term \rightarrow Function(Term, ...)

- | Constant
- | Variable

- Constants KingJohn, 2,...
- Predicates Brother, >,...
- Functions Sqrt, LeftLegOf,...
- Variables x, y, a, b,...
- Connectives $\neg, \Rightarrow, \wedge, \vee, \Leftrightarrow$
- Equality =
- Quantifiers \forall, \exists

انواع نمادها برای بیان اشیا، روابط و توابع منطق مرتبه اول

نمادهای ثابت: به یک شی خاص در مدل رجوع می کنند

– A, B, John,...

نمادهای مسندی (خبر دهنده): به یک رابطه ویژه در مدل اشاره می کند:

– Brother, Mother,...

نمادهای تابعی: یعنی هر شی توسط رابطه دقیقا به یک شیء دیگر رجوع می کند (شروع با حرف بزرگ)

– MotherOf, Cos, LeftLegOF,...

اصطلاحات (ترم ها)

هر اصطلاح یک عبارت منطقی است که به شیئی اشاره می کند.

○ نمادهای ثابت، اصطلاح هستند.

○ تخصیص یک نماد مشخص برای هر شیء همیشه کار درستی نیست، باید از نماد های تابعی نیز استفاده کنیم.

LeftLeg(John)

○ مثال: پای چپ پای پادشاه John به جای نماد جدید

جملات بسیط (جملات اتمیک)

جملات بسیط: یک جمله بسیط از یک نماد مسند که به دنبال آن فهرستی از اصطلاحات درون

پرانتز می آید، تشکیل می شود.

Predicate(Term₁ , ... , Term_n)

Married(FatherOf(Richard), MotherOF(John))

Brother(KingJohn,RichardTheLionheart)

جملات مرکب

می‌توان از رابطهای منطقی (مثل \wedge یا \vee یا \dots) برای ساخت جملات پیچیده تر استفاده کرد، درست همان گونه که در حسابان گزاره ای داشتیم.

$\neg S, S1 \wedge S2, S1 \vee S2, S1 \Rightarrow S2, S1 \Leftrightarrow S2$

$\neg \text{Brother}(\text{LeftLeg}(\text{Richard}), \text{John})$

$\text{Brother}(\text{Richard}, \text{John}) \wedge \text{Brother}(\text{John}, \text{Richard})$

$\text{King}(\text{Richard}) \vee \text{King}(\text{John})$

$\neg \text{King}(\text{Richard}) \Rightarrow \text{King}(\text{John})$

سورها

سورها کمک می‌کنند تا به جای برشماری اشیا با نام، خواص مجموعه کل اشیا را بیان کنیم.

❖ سور عمومی: \forall "برای همه"

❖ سور وجودی: \exists "وجود دارد به طوریکه..."

سور عمومی

\forall (Variables) (Sentence)

$\forall x P$ که در آن P یک عبارت منطقی است، بیان می کند که P به ازاء هر شیء x درست است. به بیان دقیق تر $\forall x P$ در یک مدل مفروض و با یک تفسیر مفروض درست است اگر P در تمامی تفاسیر بسط یافته ممکن که از تفسیر داده شده ساخته می شود درست باشد.

مثال: $\forall x \text{King}(x) \Rightarrow \text{Person}(x)$ همه پادشاهان شخص هستند

رابط منطقی استلزام زیاد استفاده می شوند. سور عمومی بیانیه ای درباره تمامی اشیا است.

سور وجودی

\exists (Variables) (Sentence)

جمله $\exists x P$ می گوید که P حداقل برای یک شیء x درست است. به بیان دقیقتر، $\exists x P$ در یک مدل مشخص با تفسیر مشخص درست است اگر P حداقل در یک تفسیر توسعه یافته که x را به یک عنصر دامنه منتسب می کند، درست باشد.

مثال: $\exists x \text{Crown}(x) \wedge \text{OnHead}(x, \text{John})$ پادشاه جان تا جی روی سرش دارد

سورهای تو در تو

ترکیب چند سور، سور تو در تو را ایجاد می کند که اگر سورها از یک نوع باشند نیازی به نوشتن دوباره

$\forall x, y \text{Parent}(x, y) \Rightarrow \text{Child}(y, x)$

نماد سور نیست به ازای هر x فرزند، y والد وجود دارد

$\forall x, y \text{Sibling}(x, y) \Rightarrow \text{Sibling}(y, x)$

همزادی رابطه ی متقارن است

$\forall x (\exists y \text{Loves}(x, y))$

هر شخصی کسی را دوست دارد

$\exists y \forall x \text{Loves}(x, y)$

شخصی وجود دارد که همه او را دوست دارند

ارتباط بین سورها

سور عمومی و وجودی از طریق تناقض با یکدیگر در ارتباط هستند
”هر کسی بستنی را دوست دارد“ به معنای این است که ”هیچکس وجود ندارد که
بستنی را دوست نداشته باشد“

$\neg \exists x \neg \text{Likes}(x, \text{IceCream})$ هم ارز $\forall x \text{ Likes}(x, \text{IceCream})$

به ازای همه x ها، P اشتباه است $\neg \exists x P$ هم ارز $\forall x \neg P$

$\neg \forall x P \equiv \neg \exists x P$

$\forall x P \equiv \neg \exists x \neg P$ همه پسر زرننگ

$\exists x P \equiv \neg \forall x \neg P$

تساوی

می توان از نماد تساوی برای بیان عباراتی با این مضمون که دو اصطلاح به یک شیء اشاره می کنند، استفاده کنیم.

تعیین درستی یک جمله تساوی، در حقیقت تشخیص یکسان بودن اشیای مورد اشاره دو اصطلاح می باشد

$Father(Joh) = Henry$

ریچارد حداقل دو برادر دارد $\exists x, y \text{ Brother}(x, Richard) \wedge \text{Brother}(y, Richard) \wedge \neg(x=y)$

منطق درجه بالاتر

منطق درجه اول را از آن جهت درجه اول میخوانیم که با استفاده از آن میتوان روی اشیاء مجموعه مرجع سور اعمال کرد اما نمیتوان روی روابط توابع این اشیاء سور اعمال کرد.

منطق درجه بالاتر به ما این امکان را میدهد که روی روابط و توابع نیز مانند اشیاء سور اعمال کنیم. به عنوان مثال، در منطق درجه بالاتر میگوییم دو شیء برابرند اگر و تنها اگر تمام ویژگیهای اطلاق شده به آنها با هم برابر باشند:

$$\forall x,y (x=y) \Leftrightarrow (\forall p \ p(x) \Leftrightarrow p(y))$$

استفاده از منطق مرتبه اول

□ اظهارات و پرس و جو ها در منطق مرتبه اول

◀ جملات از طریق TELL به یک پایگاه دانش افزوده می شوند
◀ چنین جملاتی را اظهارات می نامیم

- TELL (KB , King(John))
- TELL (KB , $\forall x \text{ King}(x) \Rightarrow \text{Person}(x)$)

◀ با استفاده از ASK سوالاتی را از پایگاه دانش می پرسیم.
◀ این پرسشها، پرس و جو یا هدف نام دارد

- ASK (KB , Person(John)) true
- ASK(KB , $\exists x \text{ Person}(x)$) true

◀ فهرست جایگزینی یا الزام آور

◀ لیستی از جایگزینی ها به صورت (متغیر/اصطلاح) در صورت وجود بیش از یک پاسخ

- ◀ ASK(KB , $\forall x \text{ Person}(x)$) {x/John, x/Richard}

Mother(c)	ارجاع به مادر m (تابع)
Male/Female(m)	مرد یا زن بودن (رابطه)
Child(c,p)	فرزند کسی بودن (رابطه)
Parent(p,c)	والد کسی بودن (رابطه)
GrandParent(g,c)	جد کسی بودن (رابطه)
Spouse(h,w)	همسر بودن (رابطه)
Hasband(h,w)	شوهر کسی بودن

مثال دامنه: خویشاوندی

اولین مثال، در مورد دامنه روابط خانوادگی یا خویشاوندی است. این دامنه در بردارنده حقایقی مانند

الیزابت مادر چارلز است
چارلز پدر ویلیام است

اشیاء در این دامنه آدمها هستند. جنسیت جزء ویژگیهای آنهاست و توسط روابطی مانند والد بودن، برادر بودن، ازدواج و غیره به هم مربوط میشوند. بنابراین، ما دو مسند تکی زن و مرد داریم.

بیشتر روابط خویشاوندی مسند دوتایی است: والد، خواهر، برادر، خواهر برادر، اولاد، دختر، پسر، همسر، زن، شوهر، جد، نوه، عموزاده، عمه زاده، دایی زاده، خاله زاده، خاله، عمه، عمو و دایی. ما از توابع برای مادر و پدر استفاده میکنیم؛ زیرا هر فرد دقیقاً یک پدر و مادر دارد. (حداقل بر اساس طراحی طبیعت)

مادر یک شخص والد مؤنث وی است

$$\forall m, c \text{ Mother}(c) = m \Leftrightarrow \text{Female}(m) \wedge \text{Parent}(m, c)$$

شوهر یک شخص، همسر مذکر اوست

$$\forall w, h \text{ Husband}(h, w) \Leftrightarrow \text{Male}(h) \wedge \text{Spouse}(h, w)$$

مذکر و مؤنث دو دسته جدا هستند

$$\forall x, \text{Male}(x) \Leftrightarrow \neg \text{Female}(x)$$

والدین و فرزند، روابط معکوس یکدیگرند:

$$\forall p, c \text{ Parent}(p, c) \Leftrightarrow \text{Child}(c, p)$$

جد، والد والدین شخص است

$$\forall g, c \text{ Grandparent}(g, c) \Leftrightarrow \exists p \text{ Parent}(g, p) \wedge \text{Parent}(p, c)$$

مثال دامنه : اصول موضوعه(فرمولها)، تعاریف و نظریه ها

ریاضی دانها اصول موضوع را برای مجسم کردن حقایق پایه‌ای دامنه می نویسند و مفاهیم دیگر را براساس این حقایق تعریف میکنند و سپس از این اصول موضوعه و تعاریف برای اثبات نظریه ها استفاده میکنند.

در هوش مصنوعی، ما به ندرت از واژه نظریه استفاده میکنیم. اما به عبارتی که در پایگاه دانش اولیه است گاهی اصول موضوعه میگوییم و به طور معمول از تعاریف استفاده میکنیم. این موضوع سوال مهمی ایجاد میکند: چگونه متوجه میشویم که اصول موضوعه کافی برای تصریح کردن دامنه نوشتنهایم؟ یک روش آن است که در مورد مجموعه‌ای از مسندات فکر کنیم که براساس آنها بتوان دیگر مسندها را تعریف کرد. به عنوان مثال در دامنه خویشاوندی، اولاد، همسر، زن و مرد گزینه های منطقی برای مسندهای پایه‌ای هستند. در دامنه های دیگر، همان طور که خواهیم دید، مجموعه پایه‌ای قابل تشخیص وجود ندارد.

مثال: اعداد ، مجموعه ها

دامنه مجموعه های ریاضی نسبتاً انتزاعی تر از مجموعه شاه ها است. اما با این حال این دامنه بدنه منسجمی از دانشی را که ما علاقه مند به نمایش آن هستیم، تشکیل میدهد. ما نیاز به روشی داریم که مجموعه هایی با افزودن یک عنصر به یک مجموعه یا اجتماع گیری یا اشتراک گیری بین دو مجموعه بسازیم. ما میخواهیم بدانیم که آیا یک عنصر عضو یک مجموعه است یا خیر و بتوانیم بین مجموعه ها و اشیاءای که در مجموعه نیستند تمایز قائل شویم. ما از واژگان معمول در نظریه مجموعه ها استفاده میکنیم:

مجموعه تهی یک ثابت است؛

عنصر و زیرمجموعه مسند هستند

اشتراک، اجتماع و پیوست تابع هستند (پیوست یک تابع است که یک عنصر را به مجموعه اضافه میکند).
مجموعه یک مسند است که تنها مجموعه صادق از مجموعه هاست.

Union(s1,s2)	تابع اجتماع $s1 \cup s2$
Intersection(s1,s2)	تابع اشتراک $s1 \cap s2$
Subset(s1,s2)	رابطه زیر مجموعه بودن $s1 \subseteq s2$
Member(x,s1)	رابطه عضو بودن $x \in s1$
Adjoint(x,s2)	تابع اضافه کردن x به $s2$
EmptySet	تابع معادل مجموعه تهی

مجموعه ممکن از اصول موضوع

هشت اصل موضوعه زیر این موضوع را فراهم میکنند:

- ۱- مجموعه ها تنها مجموعه تهی و آنهایی هستند که با افزودن چیزی به یک مجموعه بدست میآیند.
- ۲- به مجموعه تهی هیچ عنصری پیوست نشده است. (به عبارت دیگر، نمیتوان آن را به یک مجموعه کوچکتر و یک عنصر تبدیل کرد.)

$$\neg \exists x, s \text{ Adjoin}(x, s) = \text{EmptySet}$$
- ۳- افزودن یک عنصری که در مجموعه موجود است، تاثیری بر مجموعه ندارد.

$$\forall x, s \text{ Member}(x, s) \Leftrightarrow s = \text{Adjoin}(x, s)$$
- ۴- تنها اعضاء یک مجموعه، عناصری هستند که به آن پیوست شده اند. ما این مورد را به صورت بازگشتی بیان میکنیم، به این شکل که x عضو s است. اگر و تنها اگر s یک مجموعه ای مانند $s2$ باشد که عنصر y به آن پیوند شده است به این شرط که y همان x باشد یا x عضو $s2$ باشد

$$\forall x, s \text{ Member}(x, s) \Leftrightarrow \exists y, s2 (s = \text{Adjoin}(y, s2) \wedge ((x=y) \vee \text{Member}(x, s2)))$$
- ۵- یک مجموعه زیر مجموعه مجموعه دیگری است اگر و تنها اگر تمام اعضاء مجموعه اول عضو مجموعه دوم باشند.

$$\forall s1, s2 \text{ Subset}(s1, s2) \Leftrightarrow (\forall x \text{ Member}(x, s1) \Rightarrow \text{Member}(x, s2))$$
- ۶- دو مجموعه برابر هستند. اگر و تنها اگر هر یک زیر مجموعه دیگر باشند.

$$\forall s1, s2 (s1 = s2) \Leftrightarrow (\text{Subset}(s1, s2) \wedge \text{Subset}(s2, s1))$$
- ۷- یک شیء عضو اشتراک دو مجموعه است اگر و تنها اگر عضو هر دو مجموعه باشد.

$$\forall x, s1, s2 \text{ Member}(x, \text{Intersection}(s1, s2)) \Leftrightarrow \text{Member}(x, s1) \wedge \text{Member}(x, s2)$$
- ۸- یک شیء عضو اجتماع دو مجموعه است اگر و تنها اگر عضو یکی از مجموعه ها باشد.

$$\forall x, s1, s2 \text{ Member}(x, \text{Union}(s1, s2)) \Leftrightarrow \text{Member}(x, s1) \vee \text{Member}(x, s2)$$



مثال : دامنه لیستها

دامنه لیستها بسیار شبیه به دامنه مجموعه ها است. تفاوت در آن است که لیستها ترتیب دارند و یک عنصر میتواند چندین بار در لیست تکرار شود. ما میتوانیم از واژگان لیست در لیستها استفاده کنیم:

Nil ثابتی است که مشخص کننده یک لیست بدون هیچ عنصر است

Reset و First و Append و Cons توابع هستند

Find یک مسند است که همان نقش Member را در مجموعه ها دارد

List مسندی است که تنها هنگامی صادق است که لیست باشد.

مثال: دنیای ومپوز و منطق مرتبه اول

Stench(S)	رابطه بوی بد در S
Breezy(S)	رابطه نسیم در S
Pit(S)	رابطه گودال در S
AtAgent(Agent,s)	رابطه عامل در خانه S است
AtGold(T)	در خانه ای حاوی طلا هستیم
Action(عمل,T)	انجام دادن عمل در لحظه خاص

اولین قدم در ایجاد يك مأمور برای دنیای ومپوز (یا هر دنیای دیگر) تعریف واسطی بین مأمور و محیط اطراف است. جمله دریافتی باید شامل خود دریافت و زمان وقوع آن باشد؛ در غیر اینصورت مأمور نمی داند که چه موقع چه رویدادی را مشاهده کرده است. ما از اعداد صحیح برای بیان مراحل زمانی استفاده می کنیم. يك جمله دریافت معمول به شکل زیر است:

Percept([Stench, Breeze,Glitter,None,None],5)

Percept مسند دودویی است و Stench و بقیه درکیات ، ثابتهای موجود در لیست هستند

function KB-AGENT(*percept*) **returns an action**
 static: *KB*, a knowledge base
t, a counter, initially 0, indicating time

TELL(*KB*, MAKE-PERCEPT-SENTENCE(*percept*, *t*))
action ← ASK(*KB*, MAKE-ACTION-QUERY(*t*))
 TELL(*KB*, MAKE-ACTION-SENTENCE(*action*, *t*))
t ← *t* + 1
return *action*

عمل مأمور بایستی یکی از ۷ موارد زیر باشد:
 بالا رفتن، رها کردن، برداشتن، شلیک کردن، حرکت به جلو و چرخش به چپ و راست.
 Turn(Right), Turn(Left), Forward, Shoot, Grab, Release, Climb

برای تعیین اینکه کدام عمل بهترین است، برنامه عامل (Make – Action – Query) يك پرس و جو مثل زیر ایجاد می کند

$\exists a \text{Action}(a,5)$

با این انگیزه که تابع ASK يك لیست مرتبط بر می گرداند مانند {a/Grab} و Grab که در آن Grab به عنوان مقدار به متغیر Action منسوب شده است. برنامه مأمور پس تابع Tell را یکبار دیگر صدا می زند تا عمل انجام شده را ثبت و ذخیره کند. چون مأمور آنچه را خود انجام می دهد دریافت نمی کند، این تنها راهی است که منبع آگاهی می تواند بفهمد مأمور چه انجام

ادامه مثال: دنیای ومپوز و منطق مرتبه اول

يك مأمور واكنشي ساده

ساده ترین نوع مأمور ممکن قوانینی دارد که دریافتها را مستقیماً به اعمال مرتبط می کنند. این قوانین شبیه به واکنشها یا غرایزند، به عنوان مثال، اگر مأمور يك درخشش ببیند، بایستی عمل برداشتن را برای برداشتن طلا انجام دهد:

$\forall s, b, u, c, t \text{ Percept } ([s, b, \text{Glitter}, u, c], t) \Rightarrow \text{Action } (\text{Grab}, t)$ بادیدن درخش ، عملیات برداشتن

اتصال بین دریافت و عمل می تواند توسط قوانینی برای دریافت میانگیری شود، که دریافت ورودی لحظه ای را به اشکال مفیدتری مجرد سازی می کنند:

$\forall b, g, u, c, t \text{ Percept } ([\text{Stench}, b, g, u, c], t) \Rightarrow \text{Stench } (t)$

$\forall s, g, u, c, t \text{ Percept } ([s, \text{Breeze}, g, u, c], t) \Rightarrow \text{Breeze } (t)$

$\forall s, b, u, c, t \text{ Percept } ([s, b, \text{Glitter}, u, c], t) \Rightarrow \text{AtGold } (t)$

سپس ، می توان یک ارتباط از این مسندات به انتخابهای عمل ایجاد کرد:

$\forall t \text{ AtGold } (t) \Rightarrow \text{Action } (\text{Grab}, t)$

این قانون منعطف تر از روش اتصال مستقیم عمل و دریافت است، زیرا می تواند به همراه مفاهیم دیگری برای نتیجه گیری AtGold بکار رود- مثلاً برخورد انگشت کسی با طلا.

در يك محیط پیچیده تر، دریافت ممکن است آرایه کاملی از مقادیر مقیاس - خاکستری (تصویر سیاه و سفید) یا رنگی (تصویر رنگی يك دوربین) باشد، و قوانین مربوط به دریافت ها ملزم به استنتاج چیزهایی مثل "يك کامیون بزرگ درست پشت سر من است و چراغ هایش را روشن و خاموش می کند" باشند.

ادامه مثال: دنیای ومپوز و منطق مرتبه اول

استنتاج خواص پنهان دنیا

$\forall l, s$

$At(Agent, l, s) \wedge Breeze(s) \Rightarrow Breezy(l)$ زمانی که مأمور درمی یابد که کجاست، می تواند مهارتها را علاوه بر وضعیت ها با مکانها نیز مربوط کند. بنابراین به عنوان مثال، شخصی ممکن است بگوید اگر مأمور در جایی است و یک نسیم خوش دریافت می کند، در این صورت آن مکان خوش آب و هوا است و اگر مأمور یک بوی بد دریافت می کند آن مکان بد بوست

$\forall l, s$

$At(Agent, l, s) \wedge Stench(s) \Rightarrow Smelly(l)$

اینکه بدانیم که یک مکان خوش آب و هوا یا بدبو است، مفید است چرا که ما می دانیم که اژدها و چاله ها نمی توانند به اطراف حرکت کنند. دقت کنید نه مسند بدبو و نه مسند خوش آب و هوا نیازی به آرگومان وضعیت (زمان) ندارند. پس از کشف اینکه کدام مکان ها بدبو یا خوش آب و هوا هستند و بطور بسیار مهمی غیر بدبو یا غیر خوش آب و هوا، مأمور می تواند استنتاج کند که چاله ها کجا هستند و اژدها کجاست. به علاوه او می تواند استنتاج کند که رفتن به کدام خانه ها ایمن برای بیان این موضوع استفاده می کنیم). او می تواند از این اطلاعات برای یافتن طلا استفاده کند. اصولی که ok است (ما از مسند برای دریافت اطلاعات ضروری برای این استنتاج ها نوشته خواهد شد، **قوانین همزمان** نامیده می شوند، چرا که آنها خواص یک حالت دنیا را به خواص دیگر همان حالت دنیا مرتبط می کنند. دو نوع کلی قوانین همزمان داریم:

قوانین علی قوانین علی جهت مفروض علیت در دنیا را منعکس می کنند: برخی خواص پنهان دنیا باعث دریافت های خاصی می شوند. (پنهانها - < واقعی)

$\forall r Pit(r) \Rightarrow [\forall s Adjacent(r, s) \Rightarrow Breezy(s)]$

وجود یک گودال سبب می شود تمام مربعهای مجاور دارای نسیم باشند:

اگر تمامی مربعهای مجاور یک مربع مفروض بدون گودال باشند، آن مربع نسیم دار نخواهد بود: $\forall s [\forall r Adjacent(r, s) \Rightarrow \neg Pit(r)] \Rightarrow \neg Breezy(s)$ سامانه هایی که با استفاده از قوانین علی استدلال می کنند سامانه های **استدلال مدل منیا** نامیده می شوند.

قوانین شناختی قوانین شناختی وجود خواص پنهان را مستقیماً از اطلاعات مشتق شده نتیجه می گیرند. ما قبلاً در قانون تشخیصی دیده ایم: (واقعی - < پنهان)

$\forall s Breezy(s) \Rightarrow \exists r Adjacent(r, s) \wedge Pit(r)$

اگر مربعی دارای نسیم باشد یک مربع مجاور بایست دارای گودال باشد

$\forall s \neg Breezy(s) \Rightarrow \neg \exists r Adjacent(r, s) \wedge Pit(r)$

اگر مربعی دارای نسیم نباشد هیچ مربع مجاور نبایست دارای گودال باشد

$\forall s Breezy(s) \Leftrightarrow \exists r Adjacent(r, s) \wedge Pit(r)$

ترکیب این دوشروط:

مهندسی دانش در FOL

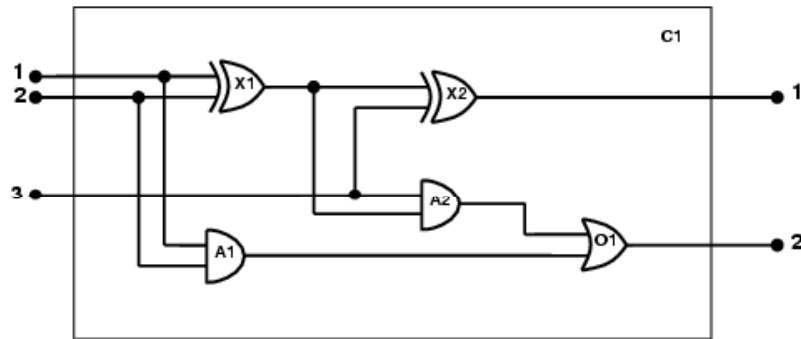
فرآیند کلی ساخت پایگاه دانش را مهندسی دانش گویند
یک مهندس دانش شخصی است که بر روی دامنه به خصوصی تحقیق می کند و می آموزد که چه مفاهیمی در آن دامنه مهم هستند و بازنمایی رسمی از اشیا و روابط آن دامنه ایجاد می کند

فازهای مهندسی دانش (ص ۳۰۷ – ص ۳۰۸) : هفت خان

۱. **مشخص نمودن وظیفه** : شناخت دامنه مسئله و ... (مشابه PEAS)
۲. **جمع آوری دانش مربوطه** : اکتساب دانش، عدم بازنمایی رسمی دانش و درک چگونگی کار دامنه
۳. **تعیین مسندها، توابع و ثابت ها** : ترجمه مفاهیم مهم دامنه در سطح منطق
۴. **کد نمودن دانش عمومی دامنه** : نوشتن اصول موضوع . هستی شناسی لیستی غیر رسمی از مفاهیم موجود در قلمرو میباشد. با نوشتن جملات منطقی یا قضایا درباره عبارتهای موجود در هستی شناسی به دو هدف میرسیم: اولاً، عبارتها را دقیقتر میکنیم به گونهای که انسانها روی تفسیر آنها توافق خواهند کرد. بدون اصول ما نمیفهمیم که برای مثال آیا منظور از خرس یک خرس واقعی است یا یک خرس ساختگی یا هردو. ثانیاً، ما اجرای رویه استنتاج برای بدست آوردن خودکار نتیجه گیری از پایگاه معلومات را ممکن میکنیم. هرگاه اصول مناسب باشند ما میتوانیم بگوییم که پایگاه معلومات ایجاد شده است.
۵. **کد نمودن توصیف نمونه مساله خاص** : اگر هستی شناسی، خوب مورد بررسی قرار گرفته باشد این مرحله آسان خواهد بود. این مرحله عمدتاً شامل نوشتن حملات بنیادی ساده (بسیط) درباره نمونه هایی از مفاهیمی که پیش از این قسمتی از هستی شناسی بوده اند، میباشد.
۶. **اعمال پرس و جو به رویه استنتاج و در یافت پاسخ** : مزیت این مرحله این است که: ما به رویه استنتاج اجازه میدهیم تا برای پی بردن به حقایقی که علاقه به دانستن آنها داریم، روی اصول و حقایق ویژه مسئله عملیات انجام دهد.
۷. **اشکال زدایی پایگاه دانش**

مثال : فازهای مهندسی دانش در حوزه مدارهای الکتریکی

• جمع کننده یک بیتی



۱. مشخص نمودن وظیفه

- آیا مدار واقعا به درستی جمع می کند؟ (واریسی مدار)

۲. جمع آوری دانش مربوطه

- ترکیب سیم ها و گیت ها؛ انواع گیت ها (AND, OR, XOR, NOT)

- دانش نامربوط: اندازه، شکل، رنگ، قیمت گیت ها

۳. تصمیم گیری در مورد لغات


- راه های مختلف:

تصمیم گیری در مورد توابع، مسندها، روابط

$$\text{Type}(X_1) = \text{XOR}$$

$$\text{Type}(X_1, \text{XOR})$$

$$\text{XOR}(X_1)$$



۴. کد نمودن دانش عمومی دامنه

$$-\forall t_1, t_2 \text{ Connected}(t_1, t_2) \Rightarrow \text{Signal}(t_1) = \text{Signal}(t_2)$$

$$-\forall t \text{ Signal}(t) = 1 \vee \text{Signal}(t) = 0$$

$$-1 \neq 0$$

$$-\forall t_1, t_2 \text{ Connected}(t_1, t_2) \Rightarrow \text{Connected}(t_2, t_1)$$

$$\begin{aligned} -\forall g \quad \text{Type}(g) = \text{OR} &\Rightarrow \text{Signal}(\text{Out}(1, g)) = 1 \\ &\Leftrightarrow \exists n \text{ Signal}(\text{In}(n, g)) = 1 \end{aligned}$$

$$\begin{aligned} -\forall g \quad \text{Type}(g) = \text{AND} &\Rightarrow \text{Signal}(\text{Out}(1, g)) = 0 \\ &\Leftrightarrow \exists n \text{ Signal}(\text{In}(n, g)) = 0 \end{aligned}$$

$$\begin{aligned} -\forall g \quad \text{Type}(g) = \text{XOR} &\Rightarrow \text{Signal}(\text{Out}(1, g)) = 1 \\ &\Leftrightarrow \text{Signal}(\text{In}(1, g)) \neq \text{Signal}(\text{In}(2, g)) \end{aligned}$$

۵. کد نمودن دانش مربوط به یک نمونه مساله خاص

Type(X_1) = XOR

Type(A_1) = AND

Type(O_1) = OR

Type(X_2) = XOR

Type(A_2) = AND

Connected(Out(1, X_1),In(1, X_2))

Connected(Out(1, X_1),In(2, A_2))

Connected(Out(1, A_2),In(1, O_1))

Connected(Out(1, A_1),In(2, O_1))

Connected(Out(1, X_2),Out(1, C_1))

Connected(Out(1, O_1),Out(2, C_1))

Connected(In(1, C_1),In(1, X_1))

Connected(In(1, C_1),In(1, A_1))

Connected(In(2, C_1),In(2, X_1))

Connected(In(2, C_1),In(2, A_1))

Connected(In(3, C_1),In(2, X_2))

Connected(In(3, C_1),In(1, A_2))

۶. اعمال پرس و جو بر رویه استنتاج

چه ترکیبی از ورودی ها باعث می شوک که اولین خروجی مدار C_1 (بیت جمع) به صفر و خروجی دوم مدار C_1 (بیت نقلی) به یک تبدیل شود؟

$$\exists i_1, i_2, i_3 \text{ Signal(In}(1, C_1)) = i_1 \wedge \text{Signal(In}(2, C_1)) = i_2 \wedge \text{Signal(In}(3, C_1)) = i_3 \wedge \\ \text{Signal(Out}(1, C_1)) = 0 \wedge \text{Signal(Out}(2, C_1)) = 1$$

پاسخ های ممکن:

$$\{i_1/1, i_2/1, i_3/0\},$$

$$\{i_1/1, i_2/0, i_3/1\},$$

$$\{i_1/0, i_2/1, i_3/1\}$$

$$\exists i_1, i_2, i_3, o_1, o_2 \text{ Signal(In}(1, C_1)) = i_1 \wedge \text{Signal(In}(2, C_1)) = i_2 \wedge \\ \text{Signal(In}(3, C_1)) = i_3 \wedge \text{Signal(Out}(1, C_1)) = o_1 \wedge \text{Signal(Out}(2, C_1)) = o_2$$

پاسخ: جدول ورودی/خروجی کامل که می تواند برای بررسی مدار استفاده شود.

۷. اشکال زدایی پایگاه دانش

اگر حقیقت $1 \neq 0$ را در پایگاه دانش نداشته باشیم:

$$\exists i_1, i_2, o \quad \text{Signal}(\text{In}(1, C_1)) = i_1 \wedge \text{Signal}(\text{In}(2, C_1)) = i_2 \wedge \\ \text{Signal}(\text{Out}(1, X_1)) = o$$

این پرسش مشخص می کند که برای ورودی های ۱۰ و ۰۱ هیچ خروجی ای در X_1 مشخص نمی باشد. با توجه به اصل موضوعی در مورد XOR

$$\forall g \text{ Type}(g) = \text{XOR} \Rightarrow \text{Signal}(\text{Out}(1, g)) = 1 \Leftrightarrow \text{Signal}(\text{In}(1, g)) \neq \text{Signal}(\text{In}(2, g))$$

اگر ورودی هاسفر و یک باشند:

$$\text{Signal}(\text{Out}(1, X_1)) = 1$$

$$\Leftrightarrow 1 \neq 0$$