

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



هوش مصنوعی

فصل ۸

منطق مرتبه اول

First-Order Logic

کاظم فولادی
دانشکده مهندسی برق و کامپیوتر
دانشگاه تهران

<http://courses.fouladi.ir/ai>

هوش مصنوعی

منطق مرتبه اول

۱

بازبینی
بازنمایی

مزایا و معایب «منطق گزاره‌ای»

منطق گزاره‌ای = منطق مرتبه صفر

منطق گزاره‌ای	
معایب	مزایا
قدرت بیان بسیار محدود <i>Very Limited Expressive Power</i>	اعلانی بودن <i>Declarative</i>
(بر خلاف زبان‌های طبیعی)	اجزای نحوی متناظر با واقعیت‌ها هستند
مثلاً: در مسئله‌ی دنیای اژدها نمی‌توانیم بگوییم «چاله‌ها موجب نسیم در خانه‌های مجاور می‌شوند» مگر اینکه برای هر خانه دقیقاً یک جمله بنویسیم!	امکان بیان اطلاعات جزئی/فصلی/نفی‌شده <i>Partial/Disjunctive/Negated Information</i>
	برخلاف بیشتر ساختمان‌داده‌ها و پایگاه‌های داده
	ترکیبی بودن <i>Compositional</i>
	معنای جمله از معنای اجزای آن استخراج می‌شود
	معنای مستقل از بستر <i>Context-Independent Meaning</i>
	بر خلاف زبان‌های طبیعی که معنا وابسته به بستر است

انواع منطق‌ها

منطق مرتبه اول و تعهدات آن

تعهدات منطق		
تعهدات معرفت‌شناختی <i>Epistemological Commitment</i>	تعهدات هستی‌شناختی <i>Ontological Commitment</i>	
درست / نادرست / ناشناخته <i>True / False / Unknown</i>	واقعیت‌ها <i>Facts</i>	منطق گزاره‌ای <i>Propositional Logic</i>
درست / نادرست / ناشناخته <i>True / False / Unknown</i>	واقعیت‌ها، اشیاء، رابطه‌ها <i>Facts, Objects, Relations</i>	منطق مرتبه اول <i>First-Order Logic</i>
درست / نادرست / ناشناخته <i>True / False / Unknown</i>	واقعیت‌ها، اشیاء، رابطه‌ها، زمان‌ها <i>Facts, Objects, Relations, Times</i>	منطق زمانی <i>Temporal Logic</i>
درجه‌ی باور بین صفر تا یک <i>Degree of Belief 0...1</i>	واقعیت‌ها <i>Facts</i>	نظریه‌ی احتمال <i>Probability Theory</i>
بازه‌ی معلومی از مقادیر <i>Known interval value</i>	واقعیت‌هایی با درجه‌ی درستی بین ۰ و ۱ <i>Facts with Degree of Truth in [0,1]</i>	منطق فازی <i>Fuzzy Logic</i>

منطق مرتبه اول

اسامی دیگر

FIRST-ORDER LOGIC (FOL)

منطق مرتبه اول

First-Order Logic

منطق محمولات مرتبه اول

First-Order Predicate Logic

منطق محمولات (مسندات)

Predicate Logic

حساب محمولات (مسندات)

Predicate Calculus

۲

نحو و معناشناسی منطق مرتبه اول

منطق مرتبه اول

هستی‌شناسی منطق مرتبه اول در مقابل منطق گزاره‌ای

FIRST-ORDER LOGIC (FOL)

منطق مرتبه اول <i>First-Order Logic</i>	منطق گزاره‌ای <i>Propositional Logic</i>
منطق مرتبه اول فرض می‌کند که دنیا حاوی واقعیت‌ها، اشیا، روابط و توابع است.	منطق گزاره‌ای فرض می‌کند که دنیا حاوی واقعیت‌هاست.
واقعیت‌ها <i>Facts</i>	واقعیت‌ها <i>Facts</i>
اشیا <i>Objects</i>	
روابط <i>Relations</i>	
توابع <i>Functions</i>	

منطق مرتبه اول

هستی‌شناسی منطق مرتبه اول: مثال

FIRST-ORDER LOGIC (FOL)

منطق مرتبه اول

First-Order Logic

منطق مرتبه اول فرض می‌کند که دنیا حاوی واقعیت‌ها، اشیا، روابط و توابع است.

واقعیت‌ها

Facts

افراد، خانه‌ها، اعداد، نظریه‌ها، اسامی خاص، رنگ‌ها، بازی‌ها، جنگ‌ها، قرن‌ها، ...

اشیا

Objects

قرمز بودن، گرد بودن، قلبی بودن، چندطبقه بودن، ...
برادر، بزرگ‌تر از، داخل، جزئی از، دارای رنگ، رخ داده پس از، مالک، آمده میان، ...

روابط

Relations

پدر، بهترین دوست، یکی بیشتر از، آخر، ...

توابع

Functions

منطق مرتبه اول

FIRST-ORDER LOGIC (FOL)

منطق مرتبه اول: منطقی نزدیک تر به زبان طبیعی

قابل استفاده در کاربردهای عملی

اجزای منطق مرتبه اول

تئوری اثبات

Proof Theory

استفاده از قواعد استنتاج
بررسی مدل

معناشناسی

Semantics

قواعد
درست / نادرست بودن
گزاره‌ها

نحو

Syntax

نمادهای گزاره‌ای
رابطه‌ها
گزاره‌های ترکیبی

منطق مرتبه اول

نحو منطق مرتبه اول: عناصر پایه

SYNTAX OF FOL: BASIC ELEMENTS

(شروع با حروف بزرگ)	ثابت‌ها	Constants	<i>KingJohn, 2, UCB, ...</i>
(شروع با حروف بزرگ)	محمول‌ها (مسندها)	Predicates	<i>Brother, >, ...</i>
(شروع با حروف بزرگ)	تابع‌ها	Functions	<i>Sqrt, LeftLegOf, ...</i>
(شروع با حروف کوچک)	متغیرها	Variables	<i>x, y, a, b, ...</i>
	رابطها	Connectives	$\wedge \vee \neg \Rightarrow \Leftrightarrow$
	تساوی	Equality	$=$
	سورها	Quantifiers	$\forall \exists$

نکته: در منطق مرتبه اول، **محمول‌ها** و **تابع‌ها** نمی‌توانند متغیر باشند، لذا نحوی مشابه **ثابت‌ها** دارند.



منطق مرتبه اول

نحو منطق مرتبه اول: جمله‌های اتمیک

SYNTAX OF FOL: ATOMIC SENTENCES

Atomic sentence = $predicate(term_1, \dots, term_n)$
 جمله‌ی اتمیک or $term_1 = term_2$

Term = $function(term_1, \dots, term_n)$
 ترم or *constant* or *variable*

جمله‌ی اتمیک: یک **محمول** (با آرگومان‌هایش) یا **تساوی** دو ترم
 ترم: یک **تابع** (با آرگومان‌هایش) یا یک **ثابت** یا یک **متغیر** 

ترم، یک عبارت منطقی است که به یک **شیء** اشاره می‌کند.

A *term* is a logical expression that refers to an *object*.

ترم

Term

منطق مرتبه اول

نحو منطق مرتبه اول: جمله‌های اتمیک: مثال

SYNTAX OF FOL: ATOMIC SENTENCES

Brother(KingJohn, RichardTheLionheart)



> (Length(LeftLegOf(Richard)), Length(LeftLegOf(KingJohn)))



جمله‌ی اتمیک: یک **محمول** (با آرگومان‌هایش) یا **تساوی** دو ترم
 ترم: یک **تابع** (با آرگومان‌هایش) یا یک **ثابت** یا یک **متغیر**



منطق مرتبه اول

نحو منطق مرتبه اول: جمله‌های ترکیبی

SYNTAX OF FOL: COMPLEX SENTENCES

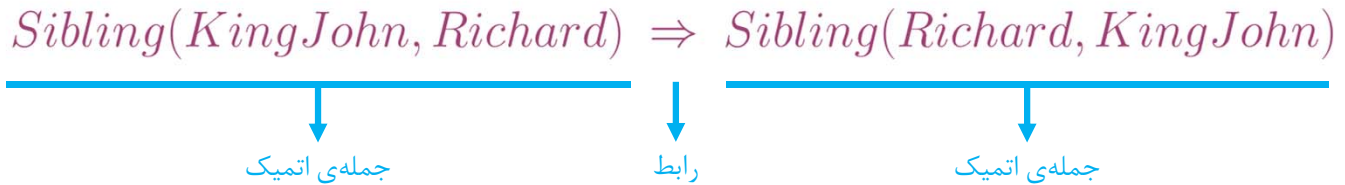
جمله‌های ترکیبی از روی جمله‌های اتمیک با استفاده از رابط‌ها ساخته می‌شوند.

$$\neg S, \quad S_1 \wedge S_2, \quad S_1 \vee S_2, \quad S_1 \Rightarrow S_2, \quad S_1 \Leftrightarrow S_2$$

منطق مرتبه اول

نحو منطق مرتبه اول: جمله‌های ترکیبی: مثال

SYNTAX OF FOL: COMPLEX SENTENCES



$$>(1, 2) \wedge \neg >(1, 2)$$

$$>(1, 2) \vee \leq(1, 2)$$

منطق مرتبه اول

نحو منطق مرتبه اول: گرامر

SYNTAX OF FOL: GRAMMAR

$$\text{Sentence} \rightarrow \text{AtomicSentence} \mid \text{ComplexSentence}$$

$$\text{AtomicSentence} \rightarrow \text{Predicate} \mid \text{Predicate}(\text{Term}, \dots) \mid \text{Term} = \text{Term}$$

$$\text{ComplexSentence} \rightarrow (\text{Sentence}) \mid [\text{Sentence}]$$

$$\mid \neg \text{Sentence}$$

$$\mid \text{Sentence} \wedge \text{Sentence}$$

$$\mid \text{Sentence} \vee \text{Sentence}$$

$$\mid \text{Sentence} \Rightarrow \text{Sentence}$$

$$\mid \text{Sentence} \Leftrightarrow \text{Sentence}$$

$$\mid \text{Quantifier Variable}, \dots \text{Sentence}$$

$$\text{Term} \rightarrow \text{Function}(\text{Term}, \dots)$$

$$\mid \text{Constant}$$

$$\mid \text{Variable}$$

$$\text{Quantifier} \rightarrow \forall \mid \exists$$

$$\text{Constant} \rightarrow A \mid X_1 \mid \text{John} \mid \dots$$

$$\text{Variable} \rightarrow a \mid x \mid s \mid \dots$$

$$\text{Predicate} \rightarrow \text{True} \mid \text{False} \mid \text{After} \mid \text{Loves} \mid \text{Raining} \mid \dots$$

$$\text{Function} \rightarrow \text{Mother} \mid \text{LeftLeg} \mid \dots$$

OPERATOR PRECEDENCE : $\neg, =, \wedge, \vee, \Rightarrow, \Leftrightarrow$

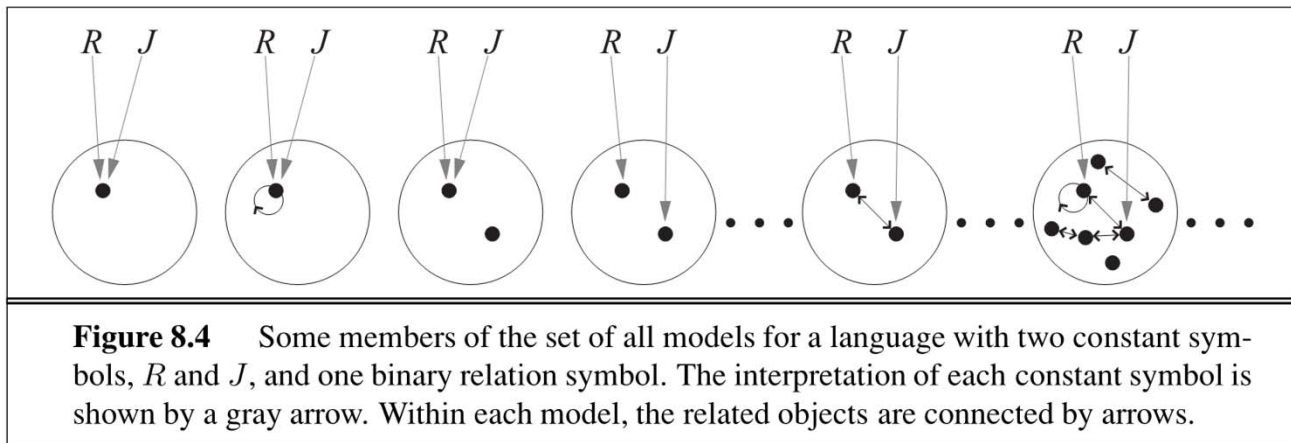


Figure 8.4 Some members of the set of all models for a language with two constant symbols, R and J , and one binary relation symbol. The interpretation of each constant symbol is shown by a gray arrow. Within each model, the related objects are connected by arrows.

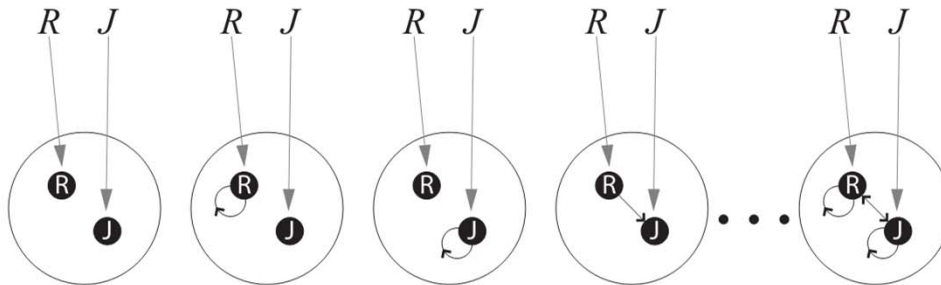
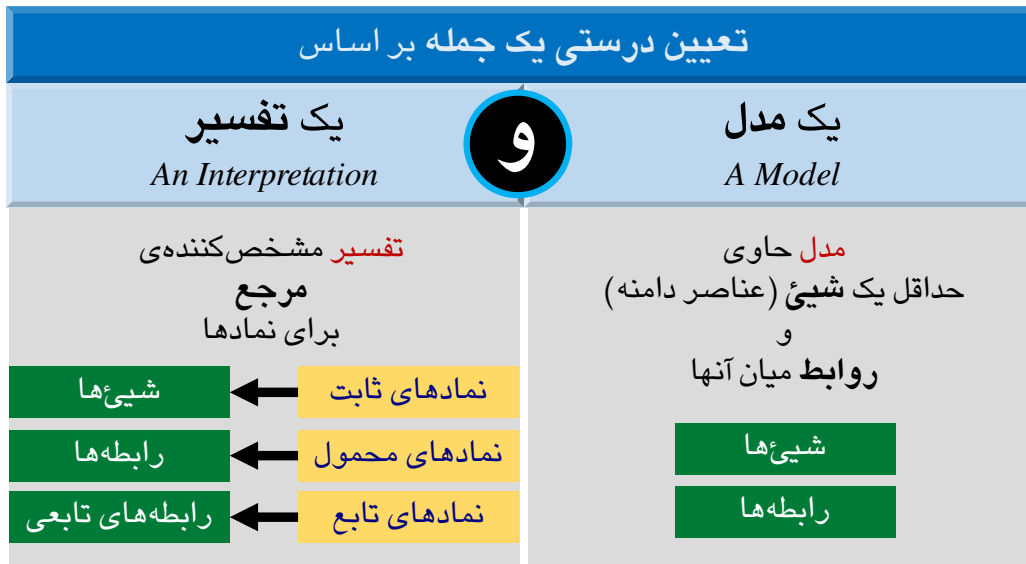


Figure 8.5 Some members of the set of all models for a language with two constant symbols, R and J , and one binary relation symbol, under database semantics. The interpretation of the constant symbols is fixed, and there is a distinct object for each constant symbol.

منطق مرتبه اول

معناشناسی منطق مرتبه اول: درستی در منطق مرتبه اول

SEMANTICS OF FOL: TRUTH IN FIRST-ORDER LOGIC

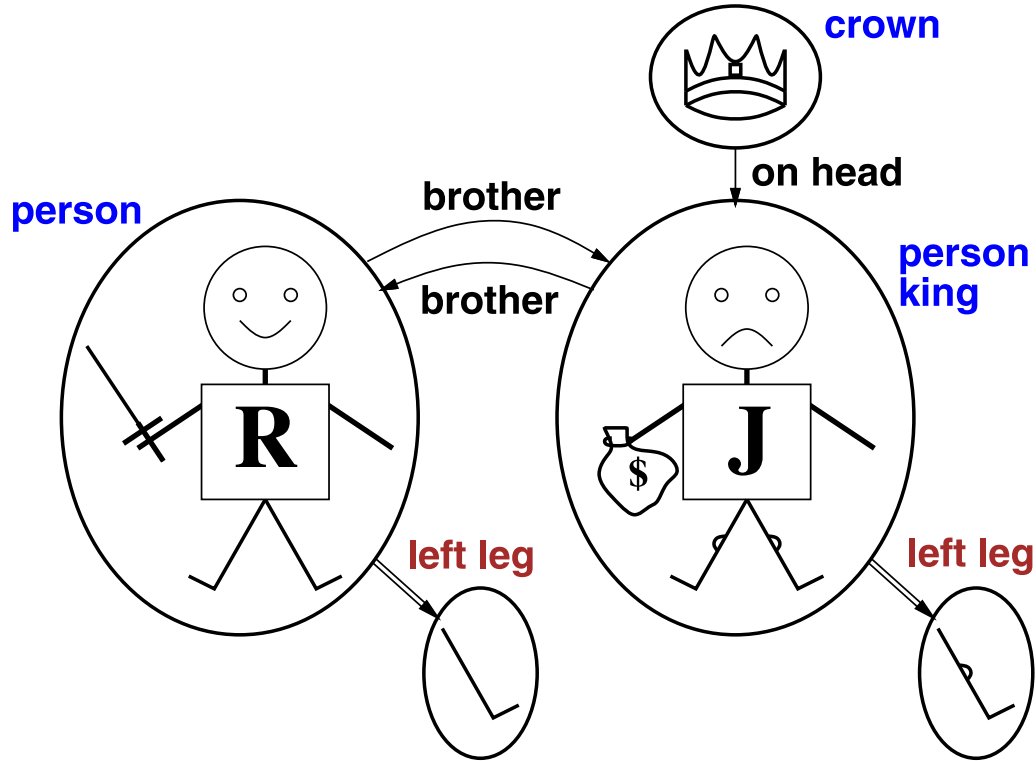


An atomic sentence $predicate(term_1, \dots, term_n)$ is true
iff the **objects** referred to by $term_1, \dots, term_n$
are in the **relation** referred to by $predicate$

منطق مرتبه اول

مدلهایی برای منطق مرتبه اول: مثال (دامنه‌ی ریچارد شیردل!)

MODELS FOR FOL: EXAMPLE

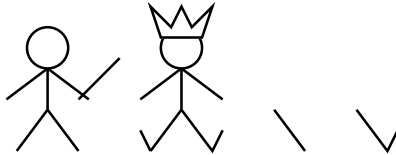


منطق مرتبه اول

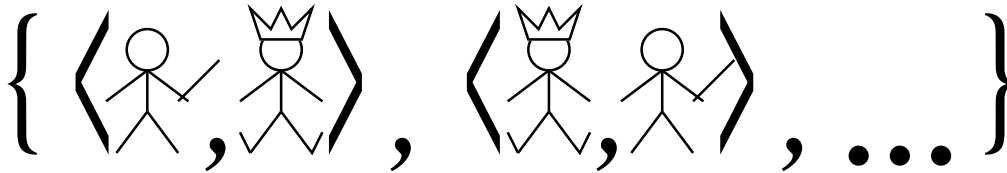
مدلهایی برای منطق مرتبه اول: مثال (دامنه‌ی ریچارد شیردل!)

MODELS FOR FOL: EXAMPLE

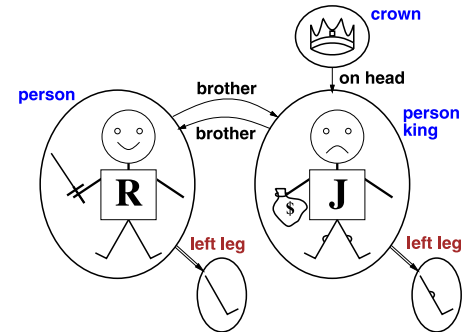
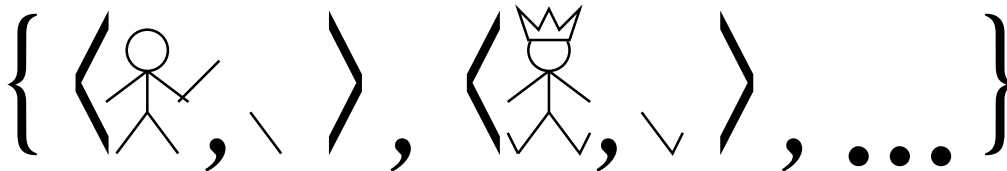
objects



relations: sets of tuples of objects



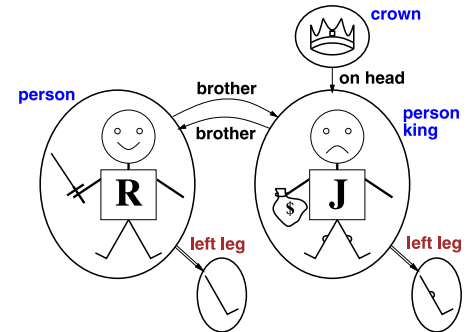
functional relations: all tuples of objects + "value" object



منطق مرتبه اول

مدلهایی برای منطق مرتبه اول: مثال (دامنه‌ی ریچارد شیردل!): مثال از درستی

TRUTH EXAMPLE



Consider the interpretation in which

Richard → Richard the Lionheart

John → the evil King John

Brother → the brotherhood relation

Under this interpretation, *Brother(Richard, John)* is true just in case Richard the Lionheart and the evil King John are in the brotherhood relation in the model

منطق مرتبه اول

مدلهایی برای منطق مرتبه اول: بسیار زیاد!

MODELS FOR FOL: LOTS!

استلزام در منطق گزاره‌ای می‌توانست با شمارش مدل‌ها محاسبه شود.

می‌توانیم مدل‌های FOL را برای یک مجموعه واژگان داده شده از یک KB بشماریم:

For each number of domain elements n from 1 to ∞
 For each k -ary predicate P_k in the vocabulary
 For each possible k -ary relation on n objects
 For each constant symbol C in the vocabulary
 For each choice of referent for C from n objects ...

محاسبه‌ی استلزام به‌وسیله‌ی شمارش مدل‌های FOL ساده نیست!



منطق مرتبه اول

سور عمومی

UNIVERSAL QUANTIFICATION $\forall \langle variables \rangle \langle sentence \rangle$ $\forall x P$ در یک مدل m درست است

اگر و فقط اگر

 P درست باشد به ازای هر شیئی ممکن در آن مدل به جای x $\forall x P$ معادل است با عطف همه‌ی نمونه‌سازی‌های P

$$\forall x P(x) \equiv P(X_1) \wedge P(X_2) \wedge P(X_3) \wedge \dots$$

منطق مرتبه اول

سور عمومی: مثال

UNIVERSAL QUANTIFICATION

هر کس در برکلی است، باهوش است. Everyone at Berkeley is smart:

$$\forall x \text{ At}(x, \text{Berkeley}) \Rightarrow \text{Smart}(x)$$

معادل است با

$$\begin{aligned} & (\text{At}(\text{KingJohn}, \text{Berkeley}) \Rightarrow \text{Smart}(\text{KingJohn})) \\ \wedge & (\text{At}(\text{Richard}, \text{Berkeley}) \Rightarrow \text{Smart}(\text{Richard})) \\ \wedge & (\text{At}(\text{Berkeley}, \text{Berkeley}) \Rightarrow \text{Smart}(\text{Berkeley})) \\ \wedge & \dots \end{aligned}$$

منطق مرتبه اول

سور عمومی: یک اشتباه متداول که باید از آن اجتناب کرد

UNIVERSAL QUANTIFICATION

معمولاً $\Rightarrow$ رابط اصلی با $\forall$ است. 

اشتباه متداول: استفاده از $\wedge$ به عنوان رابط اصلی با $\forall$

$$\forall x \text{ } At(x, Berkeley) \wedge Smart(x)$$

«همه کس در برکلی است و همه کس باهوش است.»

منطق مرتبه اول

سور وجودی

EXISTENTIAL QUANTIFICATION

$$\exists \langle variables \rangle \langle sentence \rangle$$

$$\exists x P$$
در یک مدل m درست است

اگر و فقط اگر

 P درست باشد به ازای برخی اشیای ممکن در آن مدل به جای x

$$\exists x P$$
معادل است با فصل همه‌ی نمونه‌سازی‌های P

$$\exists x P(x) \equiv P(X_1) \vee P(X_2) \vee P(X_3) \vee \dots$$

منطق مرتبه اول

سور وجودی: مثال

EXISTENTIAL QUANTIFICATION

Someone at Stanford is smart: برخی افراد در استنفورد است، باهوش اند.

$$\exists x \text{ } At(x, Stanford) \wedge Smart(x)$$

معادل است با

$$\begin{aligned} & (At(KingJohn, Stanford) \wedge Smart(KingJohn)) \\ \vee & (At(Richard, Stanford) \wedge Smart(Richard)) \\ \vee & (At(Stanford, Stanford) \wedge Smart(Stanford)) \\ \vee & \dots \end{aligned}$$

منطق مرتبه اول

سور وجودی: یک اشتباه متداول که باید از آن اجتناب کرد

EXISTENTIAL QUANTIFICATION

معمولاً $\wedge$ رابط اصلی با $\exists$ است. 

اشتباه متداول: استفاده از $\Rightarrow$ به عنوان رابط اصلی با $\exists$

$$\exists x \text{ } At(x, Stanford) \Rightarrow Smart(x)$$

«درست است اگر کسی وجود داشته باشد که در استنفورد نباشد.»

منطق مرتبه اول

خواص سورها

PROPERTIES OF QUANTIFIERS

$$\forall x \forall y \equiv \forall y \forall x$$

$$\exists x \exists y \equiv \exists y \exists x$$

$$\exists x \forall y \not\equiv \forall y \exists x$$

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

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

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

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

دوگانی سورها (همزادی)

Quantifier Duality

قاعده‌ی دمورگان

منطق مرتبه اول

خواص سورها: مثال

PROPERTIES OF QUANTIFIERS

$\exists x \forall y$ is **not** the same as $\forall y \exists x$

$\exists x \forall y \text{ Loves}(x, y)$ شخصی وجود دارد که همه‌ی افراد جهان را دوست دارد.

“There is a person who loves everyone in the world”

$\forall y \exists x \text{ Loves}(x, y)$ هر کسی در دنیا توسط حداقل یک شخص دوست داشته می‌شود.

“Everyone in the world is loved by at least one person”

متغیر سور وجودی تابعی از متغیر سورهای عمومی دربردارنده‌ی آن است.



کسی وجود ندارد که بستنی را دوست نداشت باشد $\equiv$ همه بستنی را دوست دارند

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

این گونه نیست که همه کلم بروکلی را دوست نداشته باشند $\equiv$ بعضی کلم بروکلی را دوست دارند.

$\exists x \text{ Likes}(x, \text{Broccoli}) \quad \neg \forall x \neg \text{Likes}(x, \text{Broccoli})$

منطق مرتبه اول

بازی با جمله‌ها: مثال

Brothers are siblings

برادرها همزاد هستند.

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

“Sibling” is symmetric

«همزادی» متقارن است.

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

One's mother is one's female parent

مادر یک نفر، والد مؤنث او است.

$$\forall x, y \text{ Mother}(x, y) \Leftrightarrow (\text{Female}(x) \wedge \text{Parent}(x, y)).$$

A first cousin is a child of a parent's sibling

کوزین یک فرزند همزاد یک والد است.

$$\forall x, y \text{ FirstCousin}(x, y) \Leftrightarrow \exists p, ps \text{ Parent}(p, x) \wedge \text{Sibling}(ps, p) \wedge \text{Parent}(ps, y)$$

منطق مرتبه اول

تساوی

EQUALITY

$$term_1 = term_2$$

تحت یک تفسیر درست است

اگر و فقط اگر

 $term_1$ و $term_2$ به شیئ‌های یکسانی ارجاع بدهند.

منطق مرتبه اول

تساوی: مثال

EQUALITY

$$1 = 2 \quad \text{ارضاپذیر}$$

$$\forall x \times (Sqrt(x), Sqrt(x)) = x \quad \text{ارضاپذیر}$$

$$2 = 2 \quad \text{معتبر (همیشه درست)}$$

تعریف همزاد بر حسب والد:

E.g., definition of (full) *Sibling* in terms of *Parent*:

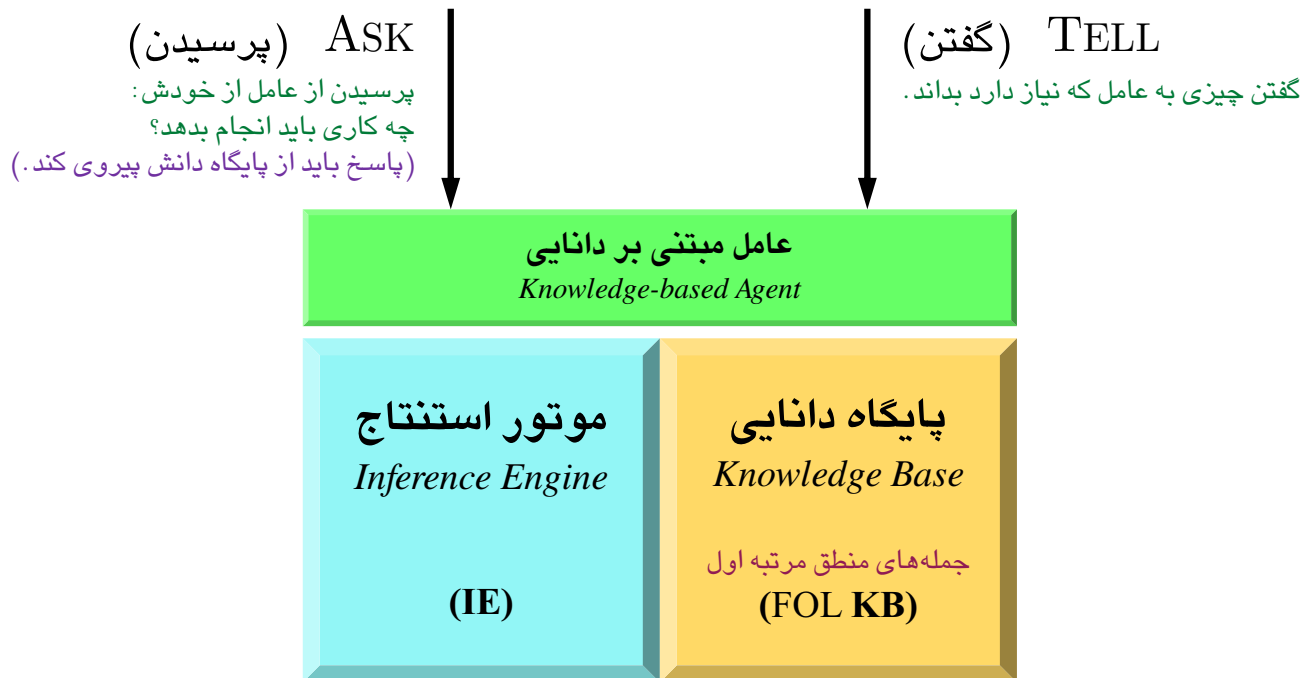
$$\forall x, y \quad Sibling(x, y) \Leftrightarrow [\neg(x = y) \wedge \exists m, f \quad \neg(m = f) \wedge \\ Parent(m, x) \wedge Parent(f, x) \wedge Parent(m, y) \wedge Parent(f, y)]$$

۳

استفاده از
منطق
مرتبه اول

عامل مبتنی بر دانایی

کار با پایگاه دانایی مبتنی بر منطق مرتبه اول



عامل مبتنی بر دانایی

کار با پایگاه دانایی مبتنی بر منطق مرتبه اول: مثال

Suppose a wumpus-world agent is using an FOL KB and perceives a smell and a breeze (but no glitter) at $t = 5$:

$Tell(KB, Percept([Smell, Breeze, None], 5))$

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

I.e., does KB entail any particular actions at $t = 5$?

Answer: $Yes, \{a/Shoot\}$ ← substitution (binding list)

منطق مرتبه اول

جانشانی

SUBSTITUTION

یک جمله‌ی S و یک لیست جانشانی σ داده شده است:
 $S\sigma$ نتیجه‌ی قرار دادن σ درون S را نشان می‌دهد.

Given a sentence S and a substitution σ ,
 $S\sigma$ denotes the result of plugging σ into S ; e.g.,
 $S = \text{Smarter}(x, y)$
 $\sigma = \{x/\text{Hillary}, y/\text{Bill}\}$
 $S\sigma = \text{Smarter}(\text{Hillary}, \text{Bill})$

منطق مرتبه اول

عمل پرسیدن: استلزام با جانشانی

$$Ask(KB, S) \text{ returns some/all } \sigma \text{ such that } KB \models S\sigma$$

استفاده از منطق مرتبه اول

مثال: پایگاه دانایی برای دنیای اژدها

KNOWLEDGE BASE FOR THE WUMPUS WORLD

ادراک "Perception"

$$\forall b, g, t \text{ Percept}([Smell, b, g], t) \Rightarrow Smelt(t)$$

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

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

بازتاب با Reflex with internal state: do we have the gold already?
 حالت داخلی $\forall t \text{ AtGold}(t) \wedge \neg \text{Holding}(\text{Gold}, t) \Rightarrow \text{Action}(\text{Grab}, t)$

 $\text{Holding}(\text{Gold}, t)$ cannot be observed $\Rightarrow$ keeping track of change is essential

برای گزاره‌هایی که نمی‌توانند مشاهده شوند:
 $\Leftarrow$ نگهداری رد تغییرات ضروری است.

استفاده از منطق مرتبه اول

استنباط خصوصیات پنهان

DEDUCING HIDDEN PROPERTIES

برای استنتاج علت (cause) از معلول (effect)

قاعده‌ی تشخیصی
Diagnostic Rule

برای استنتاج معلول (effect) از علت (cause)

قاعده‌ی علیّ
Causal Rule

تعیین خصوصیات لازم و کافی ($\Leftrightarrow$)

تعریف
Definition

استفاده از منطق مرتبه اول

استنباط خصوصیات پنهان: مثال (دنیای اژدها)

DEDUCING HIDDEN PROPERTIES

خصوصیات مکان‌ها Properties of locations:

$$\forall x, t \text{ } At(Agent, x, t) \wedge Smelt(t) \Rightarrow Smelly(x)$$

$$\forall x, t \text{ } At(Agent, x, t) \wedge Breeze(t) \Rightarrow Breezy(x)$$

خانه‌های نزدیک یک چاله، دارای نسیم هستند:

قاعده تشخیصی Diagnostic rule—infer cause from effect

$$\forall y \text{ } Breezy(y) \Rightarrow \exists x \text{ } Pit(x) \wedge Adjacent(x, y)$$

قاعده علی Causal rule—infer effect from cause

$$\forall x, y \text{ } Pit(x) \wedge Adjacent(x, y) \Rightarrow Breezy(y)$$

هیچ کدام از این دو کامل نیستند؛

برای مثال: قاعده‌ی علی بیان نمی‌کند که آیا خانه‌های دور از چاله‌ها می‌توانند دارای نسیم باشند یا خیر؟

تعریف Definition for the *Breezy* predicate:

$$\forall y \text{ } Breezy(y) \Leftrightarrow [\exists x \text{ } Pit(x) \wedge Adjacent(x, y)]$$

استفاده از منطق مرتبه اول

نگهداری رد تغییرات

KEEPING TRACK OF CHANGE

واقعیت‌ها در وضعیت‌ها برقرار هستند (نه ابدی)

حساب وضعیت

Situation Calculus

راهی برای بازنمایی تغییرات در FOL

یک آرگومان وضعیت به هر محمول غیرابدی اضافه می‌کند.

وضعیت‌ها از طریق تابع *Result* به هم متصل می‌شوند.

$Result(a, s)$ وضعیتی است که از انجام a در s حاصل می‌شود.

استفاده از منطق مرتبه اول

نگهداری رد تغییرات: مثال (دنیای اژدها)

KEEPING TRACK OF CHANGE

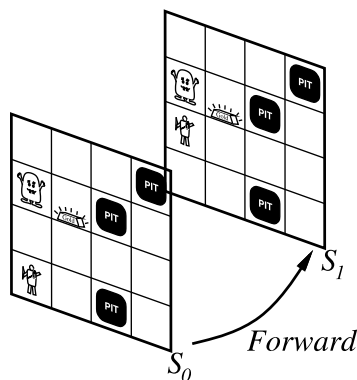
واقعیت‌ها در وضعیت‌ها برقرار هستند (نه ابدی)

E.g., *Holding(Gold, Now)* rather than just *Holding(Gold)*

در حساب وضعیت: یک آرگومان وضعیت به هر محمول غیرابدی اضافه می‌شود:

E.g., *Now* in *Holding(Gold, Now)* denotes a situation

وضعیت‌ها از طریق تابع *Result* به هم متصل می‌شوند.



$$Result(Forward, S_0) = S_1$$

۴

مهندسی
دانایی
در
منطق
مرتبه اول

مهندسی دانایی

KNOWLEDGE ENGINEERING

مهندسی دانایی

Knowledge Engineering

فرآیند ساخت پایگاه دانایی $\Leftarrow$ سیستم مبتنی بر دانایی

مهندس دانایی

Knowledge Engineer

کسی که یک دامنه‌ی خاص را بررسی می‌کند،
و یک **بازنمایی صوری** از اشیا و روابط موجود در آن دامنه ایجاد می‌کند.

مهندسی دانایی

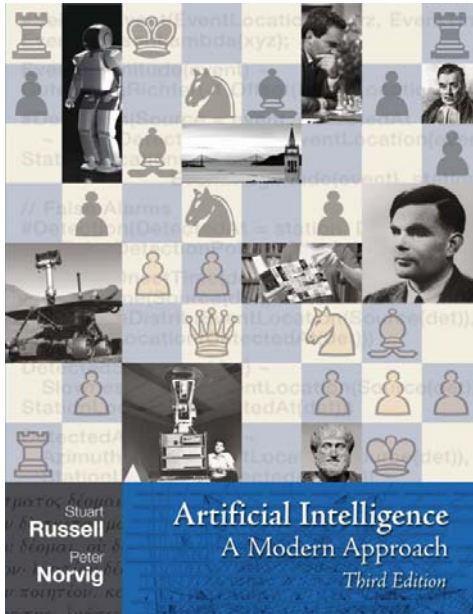
فرآیند

KNOWLEDGE ENGINEERING

<i>Identify the task.</i>	شناسایی وظیفه	۱
<i>Assemble the relevant knowledge.</i>	اسمبل کردن دانایی مرتبط	۲
<i>Decide on a vocabulary of predicates, functions, and constants.</i>	تصمیم‌گیری در مورد واژگان	۳
<i>Encode general knowledge about the domain.</i>	کدگذاری دانایی عمومی در مورد دامنه	۴
<i>Encode a description of the specific problem instance.</i>	کدگذاری توصیف یک نمونه مسئله خاص	۵
<i>Pose queries to the inference procedure and get answers.</i>	قرار دادن پرسش در روال استنتاج	۶
<i>Debug the knowledge base.</i>	اشکال‌زدایی پایگاه دانایی	۷

۵

منابع،
مطالعه،
تکلیف



Stuart Russell and Peter Norvig,
Artificial Intelligence: A Modern Approach,
 3rd Edition, Prentice Hall, 2010.

Chapter 8

8 FIRST-ORDER LOGIC

In which we notice that the world is blessed with many objects, some of which are related to other objects, and in which we endeavor to reason about them.

In Chapter 7, we showed how a knowledge-based agent could represent the world in which it operates and deduce what actions to take. We used propositional logic as our representation language because it sufficed to illustrate the basic concepts of logic and knowledge-based agents. Unfortunately, propositional logic is too puny a language to represent knowledge of complex environments in a concise way. In this chapter, we examine **first-order logic**,¹ which is sufficiently expressive to represent a good deal of our commonsense knowledge. It also either subsumes or forms the foundation of many other representation languages and has been studied intensively for many decades. We begin in Section 8.1 with a discussion of representation languages in general; Section 8.2 covers the syntax and semantics of first-order logic; Sections 8.3 and 8.4 illustrate the use of first-order logic for simple representations.

FIRST-ORDER LOGIC

8.1 REPRESENTATION REVISITED

In this section, we discuss the nature of representation languages. Our discussion motivates the development of first-order logic, a much more expressive language than the propositional logic introduced in Chapter 7. We look at propositional logic and at other kinds of languages to understand what works and what fails. Our discussion will be cursory, compressing centuries of thought, trial, and error into a few paragraphs.

Programming languages (such as C++ or Java or Lisp) are by far the largest class of formal languages in common use. Programs themselves represent, in a direct sense, only computational processes. Data structures within programs can represent facts; for example, a program could use a 4×4 array to represent the contents of the wumpus world. Thus, the programming language statement *World[2,2] ← Pit* is a fairly natural way to assert that there is a pit in square [2,2]. (Such representations might be considered *ad hoc*; database systems were developed precisely to provide a more general, domain-independent way to store and

¹ Also called **first-order predicate calculus**, sometimes abbreviated as **FOL** or **FOPC**.