

# First-Order Logic

## Chapter 8

# Pros and cons of propositional logic

- ☺ Propositional logic is **declarative**
- ☺ Propositional logic allows partial/disjunctive/negated information
  - (unlike most data structures and databases)
- ☺ Propositional logic is **compositional**:
  - meaning of  $B_{1,1} \wedge P_{1,2}$  is derived from meaning of  $B_{1,1}$  and of  $P_{1,2}$
  -
- ☺ Meaning in propositional logic is **context-independent**
  - (unlike natural language, where meaning depends on context)
  -
- ☹ Propositional logic has very limited expressive power
  - (unlike natural language)
  - E.g., cannot say "pits cause breezes in adjacent squares"
    - except by writing one sentence for each square

# First-order logic

Whereas propositional logic assumes the world contains **facts**,

first-order logic (like natural language) assumes the world contains

- **Objects:** people, houses, numbers, colors, baseball games, wars, ...  
به اشیای دنیای ما اشاره می کنند چه واقعی باشند و چه انتزاعی
- 
- **Relations:** red, round, prime, brother of, bigger than, part of, comes between, ...  
روابطی را روی object ها یا بین آن ها تعریف می کنند.
- **Functions:** father of, best friend, one more than, plus, ...  
یک نگاشت است که یک شی را به شی دیگر map می کند.  
و این function ها قرار است به ازای یک مقدار ورودی یک مقدار خروجی بدهد و نه چند مقدار خروجی

# Syntax of FOL: Basic elements

روش بازنمایی object ها

Constants

Predicates

Functions

Variables

روش بازنمایی object ها

Connectives

$\neg$ ,  $\Rightarrow$ ,  $\wedge$ ,  $\vee$ ,  $\Leftrightarrow$

Equality

=

Reserved Relation for First Order Logic

Quantifiers

$\forall$ ,  $\exists$

سورها : عمومی و وجودی

ممکن است بتوان function ها را به صورت relation بازنمایی کرد.  
ولی هیچ گاه relation نمی تواند جای function را بگیرد

باید معنای کاملا مشخص داشته باشند  
نمایند

KingJohn, 2, SUT,....

Brother, >,....

Sqrt, LeftLegOf,....

x, y, a, b,....

بازنمایی function ها

variable = lowercase  
Constant = first letter Uppercase

عملگر های منطقی که در منطق گزاره ای هم بود  
و بین relation ها قرار می گیرند.  
چون object ها از جنس true و false هستند.

# Atomic sentences

از جنس true و false می باشد

Atomic sentence =  $\text{predicate} (\text{term}_1, \dots, \text{term}_n)$   
or  $\text{term}_1 = \text{term}_2$

از جنس object

Term =  $\text{function} (\text{term}_1, \dots, \text{term}_n)$   
or constant or variable

E.g.,

*Brother(KingJohn, RichardTheLionheart)*

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

# Complex sentences

Complex sentences are made from atomic sentences using connectives

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

معنای sibling را KB مشخص می کند  
تمام چیز هایی که میدانیم را باید در KB بنویسیم

E.g.,

*Sibling(KingJohn, Richard)  $\Rightarrow$   
*Sibling(Richard, KingJohn)**

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

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

# Truth in first-order logic

Sentences are true with respect to a **model** and an **interpretation**

Model contains objects (**domain elements**) and relations among them

مراحل نگاشت

Interpretation specifies referents for

constant symbols  
predicate symbols  
function symbols

→ Real World objects  
→ Real World relations  
→ Real World functional relations

سطح ۱: یک زبانی داشت داخل آن وجود دارد  
سطح ۲: زبانی که قرار است داشت را بازنمایی کند

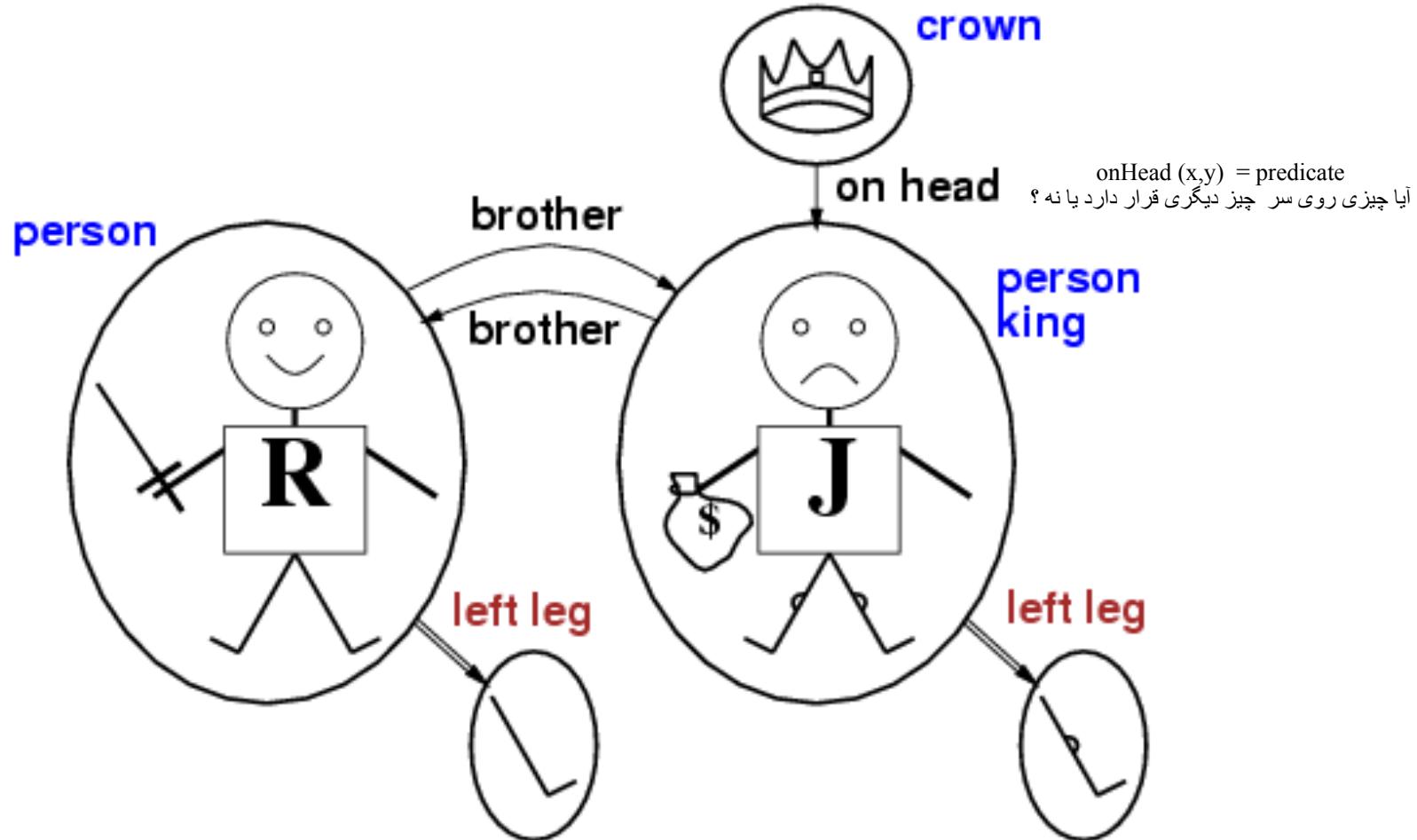
2 - عبارت های داخل منطق مرتبه اول  
را نگاشت می کنیم که مدلی در دنیای  
واقعی داریم .

1- در سطح اول یک مدل تشخیص می  
دهیم  
مثالاً اگر درباره دنیای آدم ها صحبت می  
کنیم و  
رابطه برادری بین آن ها  
باید همه آدم ها مشخص شوند و این که  
ایا بین دو نفر خاص رابطه برادری  
وجود دارد را هم باید مشخص بکنیم =>  
به یک مدل می رسیم )

یک عبارت atomic درست است اگر تمامی نگاشت ها درست انجام شده باشد

An atomic sentence  $\text{predicate}(\text{term}_1, \dots, \text{term}_n)$  is true  
iff the **objects** referred to by  $\text{term}_1, \dots, \text{term}_n$   
are in the **relation** referred to by **predicate**.

# Models for FOL: Example



# Universal quantification

$\forall <variables> <sentence>$

اگر تعداد object ها محدود باشد می توان این سور را با منطق گزاره ای نیز پیاده سازی کرد :  
به جای سور تمام مقادیر قرار دهیم و بعد با یکدیگر and کنیم

Everyone at Sharif is smart:

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

$\forall x P$  is true in a model  $m$  iff  $P$  is true with  $x$  being each possible object in the model

Equivalent to the conjunction of instantiations of  $P$

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

# A common mistake to avoid

Typically,  $\Rightarrow$  is the main connective with  $\forall$

- A universally quantifier is also equivalent to a set of implications over all objects

Common mistake: using  $\wedge$  as the main connective with  $\forall$ :

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



اشتباه رایج

means “Everyone is at SUT and everyone is smart”

معمولا در بازنمایی به فرم منطق مرتبه اول داخل سورها  
and  
نداریم

# Existential quantification

$\exists <\text{variables}> <\text{sentence}>$

اگر تعداد object ها متناهی باشد می توان آن را با منطق گزاره ای نیز پیاده سازی کرد  
OR با

Someone at SUT is smart:

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

$\exists x P$  is true in a model  $m$  iff  $P$  is true with  $x$  being some possible object in the model

Equivalent to the disjunction of instantiations of  $P$

- At(KingJohn, SUT)  $\wedge$  Smart(KingJohn)
- ✓ At(Richard, SUT)  $\wedge$  Smart(Richard)
- ✓ At(SUT, SUT)  $\wedge$  Smart(VUB)
- ✓ ...

# Another common mistake to avoid

Typically,  $\wedge$  is the main connective with  $\exists$

Common mistake: using  $\Rightarrow$  as the main connective with  $\exists$ :

کافی است یک نفر داخل شریف نباشد  
آن وقت این جمله درست است.



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

is true even if there is anyone who is not at SUT!



داخل سور وجودی اصلاً نداریم

# Properties of quantifiers

$\forall x \forall y$  is the same as  $\forall y \forall x$

$\exists x \exists y$  is the same as  $\exists y \exists x$

ترتیب مهم نیست

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

$\exists x \forall y$  Loves(x,y)

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

$\forall y \exists x$  Loves(x,y)

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

Quantifier duality: each can be expressed using the other

$\forall x$  Likes(x,IceCream)

$\neg \exists x \neg$  Likes(x,IceCream)

$\exists x$  Likes(x,Broccoli)

$\neg \forall x \neg$  Likes(x, Broccoli)

علت درستی این عبارت : قانون دمرگان

voice bookmark

# Equality

$term_1 = term_2$  is true under a given interpretation if and only if  $term_1$  and  $term_2$  refer to the same object

باعث میشود معنی آن ها باهم یکی شود اگر اشیایی که به آن ها شاره می کنند در دنیای واقعی یکی باشند

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

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

به شرط بازنمایی درست از parent عبارت بالا باید درست باشد

معمولًا در زمان ارائه یک تعریف از (اگر و فقط اگر) استفاده می کنیم.

# Using FOL

The kinship domain:

Brothers are siblings

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

One's mother is one's female parent

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

“Sibling” is symmetric

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

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

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

# Using FOL

قانون ۴ :  $x$  عضو  $s$  است اگر و فقط اگر یک object  $y$  که اولاً  $s$  از الحق  $y$  به  $s_2$  وجود آمده باشد و یا  $x$  همان  $y$  باشد که اضافه شده یا قبلًا عضو  $s_2$  باشد

یک relation می باشد که مشخص می کند چه جیزهایی مجموعه هاست

می خواهیم شی  $\{ \}$  در نهایت به مجموعه تهی map شود

قانون ۳ : به ازای هر  $x$  و مجموعه  $s$  ای ،  $x$  عضو  $s$  است اگر و فقط اگر  $x$  را که به  $s$  الحق کنیم ، خود  $s$  را دوباره به ما بدهد

۱- شی  $s$  یک مجموعه است اگر و تنها اگر  $s$  تهی باشد یا (یک  $x$  و یک  $(s_2)$  وجود داشته باشد

قانون ۲ : نباید بتوان عضوی به یک مجموعه الحق کرد و نتیجه تهی گرفت (نباید حاصل الحق یک عضو به یک مجموعه ، مجموعه تهی شود )

از الحق  $x$  به  $s_2$  تولید شده باشد

$$1. \forall s \text{ Set}(s) \Leftrightarrow (s = \{\}) \vee (\exists x, s_2 \text{ Set}(s_2) \wedge s = \{x|s_2\})$$

\* یک رابطه است بین دو عضو

\*\* یک رابطه است بین دو عضو

$$3. \forall x, s \ x \in s \Leftrightarrow s = \{x|s\}$$

می خواهد عضویت را تعریف کند طوری که با دانش ما هماهنگی داشته باشد

$$4. \forall x, s \ x \in s \Leftrightarrow [\exists y, s_2 (s = \{y|s_2\} \wedge (x = y \vee x \in s_2))]$$

$$5. \forall s_1, s_2 \ s_1 \subseteq s_2 \Leftrightarrow (\forall x \ x \in s_1 \Rightarrow x \in s_2)$$

\* یک function است  
اجتماع نیز یک function است  
الحق نیز یک function است

$$6. \forall s_1, s_2 (s_1 = s_2) \Leftrightarrow (s_1 \subseteq s_2 \wedge s_2 \subseteq s_1)$$

$$7. \forall x, s_1, s_2 x \in (s_1 \cap s_2) \Leftrightarrow (x \in s_1 \wedge x \in s_2)$$

$$8. \forall x, s_1, s_2 x \in (s_1 \cup s_2) \Leftrightarrow (x \in s_1 \vee x \in s_2)$$

زمانی که یک قانون را حذف می کنیم به این معنا نیست که نمی توان از کاری را که در قانون گفته شده استفاده کرد  
می توان آن را انجام داد و آن جاهایی که ممنوع بود آزاد شدیم

در قانون ۶ علامت (=) بیانگر Equality را رزرو شده در منطق مرتبه اول نیست .

در قانون ۱ علامت (=) همان Equality را رزرو شده در منطق مرتبه اول است

# Using FOL

The set domain:      **WRONG!!!!**      **WHY???**

1.  $\forall s \text{ Set}(s) \Leftrightarrow (s = \{\}) \vee (\exists x, s_2 \text{ Set}(s_2) \wedge s = \{x|s_2\})$
2.  $\neg \exists x, s \{x|s\} = \{\}$
3.  $\forall x, s x \in s \Leftrightarrow s = \{x|s\}$
4.  $\forall x, s x \in s \Leftrightarrow [\exists y, s_2 (s = \{y|s_2\} \wedge (x = y \vee x \in s_2))]$
5.  $\forall s_1, s_2 s_1 \subseteq s_2 \Leftrightarrow (\forall x x \in s_1 \Rightarrow x \in s_2)$
6.  $\forall s_1, s_2 (s_1 = s_2) \Leftrightarrow (s_1 \subseteq s_2 \wedge s_2 \subseteq s_1)$
7.  $\forall x, s_1, s_2 x \in (s_1 \cap s_2) \Leftrightarrow (x \in s_1 \wedge x \in s_2)$
8.  $\forall x, s_1, s_2 x \in (s_1 \cup s_2) \Leftrightarrow (x \in s_1 \vee x \in s_2)$

# Using FOL

فرض کنید می خواهیم عضویت  $a$  را در این مجموعه بررسی کنیم  
طبق مدل باید برقرار باشد یعنی  $a$  عضو این مجموعه باشد

و  $y$  را  $b$  می گیریم

چون نتیجه اضافه شدن  $b$  به تهی  $AB$  است

یا باید  $a$  همان  $b$  باشد ، که نیست

یا باید  $a$  قبلا در  $AB$  عضو باشد که نیست

بنابراین طرف راست عبارت درست نیست

این عبارت نشان می دهد چرا قانون ۴ جدید در مدل ما صدق نمی کند

1.  $\forall s \text{ Set}(s) \Leftrightarrow (s = \{\}) \vee (\exists x, s_2 \text{ Set}(s_2) \wedge s = \{x|s_2\})$
2.  $\neg \exists x, s \{x|s\} = \{\}$
3.  $\forall x, s \ x \in s \Leftrightarrow s = \{x|s\}$
4.  $\forall x, s \ x \in s \Leftrightarrow [\forall y, s_2 (s = \{y|s_2\} \Rightarrow (x = y \vee x \in s_2))]$
5.  $\forall s_1, s_2 \ s_1 \subseteq s_2 \Leftrightarrow (\forall x \ x \in s_1 \Rightarrow x \in s_2)$
6.  $\forall s_1, s_2 (s_1 = s_2) \Leftrightarrow (s_1 \subseteq s_2 \wedge s_2 \subseteq s_1)$
7.  $\forall x, s_1, s_2 \ x \in (s_1 \cap s_2) \Leftrightarrow (x \in s_1 \wedge x \in s_2)$
8.  $\forall x, s_1, s_2 \ x \in (s_1 \cup s_2) \Leftrightarrow (x \in s_1 \vee x \in s_2)$

# Knowledge base for the wumpus world

## Perception

- $\forall t, s, g, m, c \text{ Percept}([s, \text{Breeze}, g, m, c], t) \Rightarrow \text{Breeze}(t)$
- $\forall t, s, b, m, c \text{ Percept}([s, b, \text{Glitter}, m, c], t) \Rightarrow \text{Glitter}(t)$

## Reflex

- $\forall t \text{ Glitter}(t) \Rightarrow \text{BestAction(Grab}, t)$

## Reflex with internal state

- $\forall t \text{ Glitter}(t) \wedge \neg \text{Holding(Gold}, t) \Rightarrow \text{BestAction(Grab}, t)$

$\text{Holding(Gold}, t)$  cannot be observed: keep track of change.

قوانين که در یک زمان اجرا می شوند

All these are **synchronic** (“same time”) sentences.

Sentences allowing reasoning “across time” are called **diachronic**.

قوانين که بین زمان های مختلف است.

می توان هر کدام از این اطلاعات سنسوری را تنها به  $t$  مربوط بکنیم.

BestAction  
یک رابطه است

اگر action های BestAction در هر لحظه یکتا باشد می توان آن را با function هم نمایش داد

اگر فقط reflex را داشته باشیم به مشکل می خوریم زیرا  
بعد از این که طلا را بر می داریم هنوز glitter را احساس می کنیم

بنابراین دوباره BestAction ما همان grab می شود  
به خاطر این است که محیط ما partially observable است و ما اطلاعات  
سنسوری نداریم که به ما بگوید الان ما طلا را برداشتیم  
و این مشکل را با اضافه کردن hold به عنوان knowledge KB به هندل می کنیم  
بنابراین قانون زیر به وجود می آید.

# FOL Version of Wumpus World

اطلاعات سنسوری را به جای اینکه به زمان مربوط کنیم  
به state مربوط کنیم

Typical percept sentence:

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

Actions:

Turn(Right), Turn(Left), Forward, Shoot, Grab,  
Release

To determine best action, construct query:

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

The system should return  $\text{BestAction}(\text{Grab}, 5)$

# Deducing hidden properties

برای این که مشخص کنیم که + و - چه معنایی می دهد باید اصول موضوع حساب را هم به صورت منطق مرتبه اول بازنمایی کنیم

## Environment definition:

$$\forall x,y,a,b \text{ Adjacent}([x,y],[a,b]) \Leftrightarrow [a,b] \in \{[x+1,y], [x-1,y], [x,y+1], [x,y-1]\}$$

Properties of locations:

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

Squares are breezy near a pit:

**Diagnostics** rule--infer hidden cause from observed effect

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

**Causal** rules--some hidden property of the world causes certain percepts to be generated

از سمت cause ها به سمت effect ها رویم از رویه Causal استفاده کرده ایم

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

$$\forall s [\forall r \text{ Adjacent}(r,s) \Rightarrow \neg \text{Pit}(r)] \Rightarrow \neg \text{Breezy}(s)$$

جلوگیری از استنتاج غلط breezy بودن

در این رویکرد به  
هانگاه می کنیم و فکر می  
کنیم که این effect ها  
تحت چه شرایطی به وجود  
آمدند  
بعد قانون را می نویسیم  
این روش از effect ها به  
cause های پاشد