Artificial Intelligence

First-order predicate logic
Chapter 8, AIMA

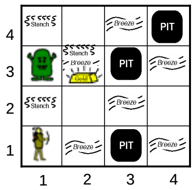
Why first order logic (FOL)?

- **Logic is a language** we use to express knowledge.
- **Propositional (boolean) logic** is too limited; complex environments cannot be described in a concise way.
- **First order logic (predicate calculus)** can express common-sense knowledge.

Limitations of propositional logic

W_{31} = Wumpus in (3,1) $\Rightarrow$ S_{32} = Stench in (3,2)

Propositional logic needs to express this for every square in the Wumpus world.

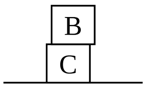


A = John has a bike $\wedge$ B = John has a car

Propositional logic does not express that these two statements are about the same person.

P = Block B is on top of C $\Rightarrow$
$\neg$Q = $\neg$(C is free to be moved)

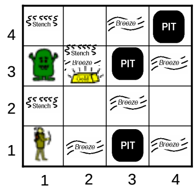
With more blocks, we need lots of statements like this.



We would like something like

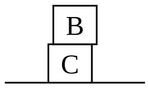
"If there is a Wumpus in square *x*, then there will be a stench in all neighboring squares."

Once and for all.



"John has a bike and a car."

"We cannot move an object that has something on top of it."



First-order logic (FOL)

Builds on:

- **Objects:**
  Man, woman, house, car, conflict,...
- **Relations (between objects):**
  Unary properties (red, green, nice,...)
  N-ary relations (larger, below,...)
- **Functions:**
  Father of, brother of, beginning of

First-order logic (FOL)
Syntax

Components

Constants
A, 125, Q, John, KingJohn, TheCrown, EiffelTower, Wumpus, HiH, Agent,...

Function constants (of all "arities")
FatherOf^{1}, DistanceBetween^{2}, Times^{2}, LeftLegOf^{1}, NeighborOf^{1}, King^{1},...

Relations (predicates, of all "arities")
Parent^{2}, Brother^{2}, Married^{2}, Before^{2}, Ontop^{2}, Orange^{1},...

Connectives and separators
$\vee \wedge \neg \Rightarrow$ ( ) ,

The superscript denotes the "arity" = the number of arguments

First-order logic (FOL)
Syntax
Components

Constants
A, 125, Q, John, KingJohn, TheCrown, EiffelTower, Wumpus, HiH, Agent,...

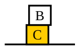
Function constants (of all ”arities”)
FatherOf^1, DistanceBetween^2, Times^2, LeftLegOf^1, NeighborOf^1, KingOf^1,...

FatherOf(John)
DistanceBetween(Timisoara,Lugoj)
Times(6,8)
LeftLegOf(KingJohn)
NeighborOf(Square$_{31}$)
KingOf(England)

First-order logic (FOL)
Syntax
Components

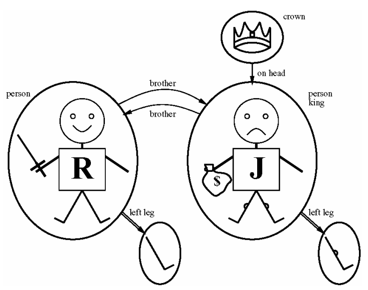
Constants
A, 125, Q, John, KingJohn, TheCrown, EiffelTower, Wumpus, HiH, Agent,...

Relations (predicates, of all ”arities”)
Parent^2, Brother^2, Married^2, Before^2, Ontop^2, Orange^1,...



Parent(Bill,John)
Brother(RichardTheLionheart,KingJohn)
Married(CarlXVIGustaf,Silvia)
Before(A,B)
OnTop(Block(B),Block(C))
Orange(Block(C))

R = RichardTheLionheart
J = KingJohn
C = Crown
} Object constants



Function constants
LeftLegOf(R)
LeftLegOf(J)

Relations (predicates)
Person(R)
Person(J)
King(J)
Crown(C)
} Unary

Brother(J,R)
Brother(R,J)
OnHead(C,J)
} Binary

First-order logic (FOL)
Syntax
Terms

1. An object constant is a term
2. A complete function constant is a term
(complete = all arguments given)
3. A *variable* is a term
(more on this later...)

First-order logic (FOL)
Syntax
Sentences

Atomic sentence
A complete predicate symbol (relation)

Brother(RichardTheLionheart,KingJohn), Dead(Mozart), Married(CarlXVIGustaf,Silvia), Orange(Block(C)),...

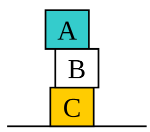
Complex sentence
Formed by sentences and connectives

Dead(Mozart) ∧ Composer(Mozart),
¬King(RichardTheLionheart) ⇒ King(KingJohn),
King(CarlXVIGustaf) ∧ Married(CarlXVIGustaf,Silvia) ⇒ Queen(Silvia)

Semantics: The truth value of complex sentences is determined with truth tables

Example: Block world

On(A,B)	True
On(B,C)	True
On(B,A)	False
On(C,B)	False
On(C,A)	False
On(A,C)	False
Clear(A)	True
Clear(B)	False
Clear(C)	False
Under(B,A)	True
Under(C,B)	True
Under(A,B)	False
Under(C,A)	False
Under(B,A) ∧ Over(B,C)	True
Under(C,B) ∨ Under(C,A)	True



First-order logic (FOL)
Syntax

Variables and quantifiers

Variables refer to unspecified objects in the domain. They are denoted by lower case letters (at the end of the alphabet)
x, y, z, …

Quantifiers constrain the meaning of a variable in a sentence. There are two quantifiers:
"For all" (∀) and "There exists" (∃)
Universal quantifier — Existential quantifier

First-order logic (FOL)
Syntax

Variables and quantifiers
(∀ "For all…")

∀x King(x) ⇒ Person(x)	*"All kings are persons"*
∀x,y Brother(x,y) ⇒ Sibling(y,x)	*"All brothers are siblings"*
∀x,y Son(x,King(y)) ⇒ Prince(x)	*"All sons of kings are princes"*
∀x AIstudent(x) ⇒ Overworked(x)	*"All AI students are overworked"*

$\forall \langle variables \rangle \; \langle sentence \rangle$

Everyone at Berkeley is smart:
$\forall x \;\; At(x, Berkeley) \;\Rightarrow\; Smart(x)$

$\forall x \;\; P$ is equivalent to the conjunction of instantiations of P

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

Typically, $\Rightarrow$ is the main connective with ∀.
Common mistake: using ∧ as the main connective with ∀:

$\forall x \;\; At(x, Berkeley) \wedge Smart(x)$

means "Everyone is at Berkeley and everyone is smart"

Slide from S. Russel @ Berkeley

First-order logic (FOL)
Syntax

Variables and quantifiers
(∃ "There exists…")

∃x King(x) ∧ Person(x)	*"There is a king who is a person"*
∃x Loves(x,KingJohn)	*"There is someone who loves King John"*
∃x ¬Loves(x,KingJohn)	*"There is someone who does not love King John"*
∃x AIstudent(x) ∧ Overworked(x)	*"There is an AI student that is overworked"*

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

Someone at Stanford is smart:
$\exists x \;\; At(x, Stanford) \wedge Smart(x)$

$\exists x \;\; P$ is equivalent to the disjunction of instantiations of P

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

Typically, ∧ is the main connective with ∃.
Common mistake: using $\Rightarrow$ as the main connective with ∃:

$\exists x \;\; At(x, Stanford) \;\Rightarrow\; Smart(x)$

is true if there is anyone who is not at Stanford!

Slide from S. Russel @ Berkeley

First-order logic (FOL)
Syntax

Nested quantifiers

∀x ∃y Loves(x,y)	*"Everybody loves somebody"*
∃y ∀x Loves(x,y)	*"Someone is loved by everyone"*
∀x ∃y Loves(y,x)	*"Everyone is loved by someone"*
∃y ∀x Loves(y,x)	*"Someone loves everyone"*
∀x ∃y Loves(x,y) ∧ (y ≠ x)	*"Everybody loves somebody else"*

Quantifier duality

DeMorgan's rules

∀x ¬P(x) ≡ ¬∃x P(x)

¬∀x P(x) ≡ ∃x ¬P(x)

∀x P(x) ≡ ¬∃x ¬P(x)

∃x P(x) ≡ ¬∀x ¬P(x)

Ponder these for a while...

Family fun

© 1998, Jon A. Davis

Family axioms:
"A mother is a female parent"
"A husband is a male spouse"
"You're either male or female"
"A child's parent is the parent of the child" (sic!)
"My grandparents are the parents of my parents"
"Siblings are two children who share the same parents"

"A first cousin is a child of the siblings of my parents"

...etc.

Family theorems:
Sibling is reflexive

Write these in FOL

Family fun

© 1998, Jon A. Davis

Family axioms:
∀m,c (m = Mother(c)) ⇔ (Female(m) ∧ Parent(m,c))
∀w,h Husband(h,w) ⇔ Male(h) ∧ Spouse(h,w)
∀x Male(x) ⇔ ¬Female(x)
∀p,c Parent(p,c) ⇔ Child(c,p)
∀g,c Grandparent(g,c) ⇔ ∃p (Parent(g,p) ∧ Parent(p,c))
∀x,y Sibling(x,y) ⇔ (∃p (Parent(p,x) ∧ Parent(p,y))) ∧ (x ≠ y)
∀x,y FirstCousin(x,y) ⇔ ∃p,s (Parent(p,x) ∧ Sibling(p,s) ∧ Parent(s,y))

...etc.

Family theorems:
∀x,y Sibling(x,y) ⇔ Sibling(y,x)
...etc.

Spouse(Gomez,Morticia)
Parent(Morticia,Wednesday)
Sibling(Pugsley,Wednesday)
Sister(Ophelia,Morticia)
FirstCousin(Gomez,Itt)
∃p (Parent(p,Morticia) ∧ Sibling(p,Fester))

Matematical fun

- "The square of every negative integer is positive"
  a) ∀x [Integer(x) ∧ (x > 0) ⇒ (x^2 > 0)]
  b) ∀x [Integer(x) ∧ (x < 0) ⇒ (x^2 > 0)]
  c) ∀x [Integer(x) ∧ (x ≤ 0) ⇒ (x^2 > 0)]
  d) ∀x [Integer(x) ∧ (x < 0) ∧ (x^2 > 0)]
- "Not every integer is positive"
  a) ∀x [¬Integer(x) ⇒ (x > 0)]
  b) ∀x [Integer(x) ⇒ (x ≤ 0)]
  c) ∀x [Integer(x) ⇒ ¬(x > 0)]
  d) ¬∀x [Integer(x) ⇒ (x > 0)]

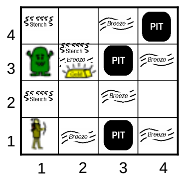
Borrowed from http://people.hofstra.edu/faculty/Stefan_Waner/RealWorld/logic/logic7.html

Matematical fun

- "The square of every negative integer is positive"
  a) ∀x [Integer(x) ∧ (x > 0) ⇒ (x^2 > 0)]
  b) ∀x [Integer(x) ∧ (x < 0) ⇒ (x^2 > 0)]
  c) ∀x [Integer(x) ∧ (x ≤ 0) ⇒ (x^2 > 0)]
  d) ∀x [Integer(x) ∧ (x < 0) ∧ (x^2 > 0)]
- "Not every integer is positive"
  a) ∀x [¬Integer(x) ⇒ (x > 0)]
  b) ∀x [Integer(x) ⇒ (x ≤ 0)]
  c) ∀x [Integer(x) ⇒ ¬(x > 0)]
  d) ¬∀x [Integer(x) ⇒ (x > 0)]

Borrowed from http://people.hofstra.edu/faculty/Stefan_Waner/RealWorld/logic/logic7.html

The Wumpus world revisited



Object constants:
Square **s** = [x,y], Agent, Time (t), Percept **p** = [p_1,p_2,p_3,p_4,p_5], Gold

Predicates:
Pit(**s**), Breezy(**s**), EvilSmelling(**s**), Wumpus(**s**), Safe(**s**), Breeze(**p**,t), Stench(**p**,t), Glitter(**p**,t), Wall(**p**,t), Scream(**p**,t), Adjacent(**s**,**r**), At(Agent,**s**,t), Hold(Gold,t)

∀ x,y,a,b Adjacent([x,y],[a,b]) ⇔ ([a,b] ∈ {[x+1,y],[x-1,y],[x,y+1],[x,y-1]})
∀**s** Breezy(**s**) ⇔ ∃**r** (Adjacent(**r**,**s**) ∧ Pit(**r**))
∀**s** EvilSmelling(**s**) ⇔ ∃**r** (Adjacent(**r**,**s**) ∧ Wumpus(**r**))
∀**s** (¬EvilSmelling(**s**) ∧ ¬Breezy(**s**)) ⇔ ∀**r** (Adjacent(**r**,**s**) ∧ Safe(**r**))

∀**s**,t (At(Agent,**s**,t) ∧ Breeze(**p**,t)) ⇒ Breezy(**s**)
∀**s**,t (At(Agent,**s**,t) ∧ Stench(**p**,t)) ⇒ EvilSmelling(**s**)

cf. the 275 rules in boolean KB (There are other possible representations)