Introduction to AI: Relational Logic (First Order Logic)

François Jacquenet

Laboratoire Hubert Curien
Université de Saint-Etienne
Francois.Jacquenet@univ-st-etienne.fr

October, 2020

Outline

Limitations of PL

We may only express relationship between individuals: likes(abby, cody), likes(dana, bess), etc

Information must be given extensively

We cannot express more general relationships such as:

- everybody likes somebody
- everyone doesn't like herself
- if a girl X likes a girl Y then Y likes X
- ...

First Order Logic extends Propositional Logic to allow that by introducing:

- variables
- quantifiers

Vocabulary

In First Order Logic we have

- Object constants
- Relation constants (predicate symbols)
- Variables

Way of writing constants: [a-z][a-zA-Z0-9_]*

Way of writing variables: [A-Z][a-zA-Z0-9_]*
This is to be consistent with the Prolog notation we will study later

Definitions

A term is either a variable or an object constant

Each relation/predicate constant has an associated arity (number of arguments)

- unary predicate
- binary predicate
- ternary predicate
- ...
- n-ary predicate

Three types of sentences

In First Order Logic, there are three types of sentences

- *relational/predicate sentences* analogous to proposition constants in PL
- *logical sentences* analogous to logical sentences in PL
- *quantified sentences* that have no analog in PL

Relational sentence

A relation sentence is formed from an n-ary predicate symbol and n terms

Example:

- *likes* is a predicate symbol with arity 2
- *cody* is an object constant
- X is a variable
- *likes(cody,X)* is a relation sentence

Logical sentences

Logical sentences are defined as in Propositional Logic. They are built using $\neg$, $\wedge$, $\vee$, $\Rightarrow$ and $\Leftrightarrow$

Examples:

- Negation: $(\neg likes(abby, dana))$
- Conjunction: $(girl(abby) \wedge likes(abby, bess))$
- Disjunction: $(likes(bess, cody) \vee likes(bess, dana))$
- Implication: $(father(X, Z) \wedge father(Z, Y) \Rightarrow grandfather(X, Y))$
- Biconditional: $(father(X, Y) \vee mother(X, Y) \Leftrightarrow parent(X, Y))$

Quantified sentences

Quantified sentences are formed from a quantifier, a variable, and an embedded sentence

The embedded sentence is called the scope of the quantifier

There are two types of quantified sentences in FOL

- universally quantified sentences
- existentially quantified sentences

Quantified sentences

A universally quantified sentence is used to assert that all objects have a certain property

Example: $\forall X.(human(X) \Rightarrow likes(X, X))$ is a universally quantified sentence that states if X is a human, then he/she likes himself/herself

An existentially quantified sentence is used to assert that some objects has a certain property.

Example: $\exists X.(human(X) \wedge \neg likes(X, X))$ is an existentially quantified sentence that states there are some humans that don't like himself/herself

Quantified sentences

Quantified sentences can be nested within other sentences

Examples:

$((\forall X.p(X)) \vee (\exists X.q(X, X)))$

$(\forall X.(\exists Y.q(X, Y)))$

TAKE CARE to the scope a each quantifier!

Quantified sentences

Precedences

$prec(\forall) = prec(\exists) > prec(\neg) > prec(\wedge) > prec(\vee) > prec(\Rightarrow) = prec(\Leftrightarrow)$

Examples:
$\forall X.p(X) \Rightarrow q(X)$ is equivalent $((\forall X.p(X)) \Rightarrow q(X))$
$\exists X.p(X) \wedge q(X)$ is equivalent $((\exists X.p(X)) \wedge q(X))$

Quantified sentences

TAKE CARE to the scope of the quantifier
$\forall X.p(X) \Rightarrow q(X)$
$\exists X.p(X) \wedge q(X)$

If we want to extend the scope, we need parentheses

Examples:
$\forall X.(p(X) \Rightarrow q(X))$
$\exists X.(p(X) \wedge q(X))$

but take care, the meaning is different

Definitions

An expression in FOL is ground if and only if it contains no variable
Example: the sentence $p(a)$ is ground, whereas the sentence $\forall X.p(X)$ is not

An occurrence of a variable is free if and only if it is not in the scope of a quantifier of that variable, otherwise, it is bound
Example: Y is free and X is bound in $\exists X.q(X, Y)$

A sentence is open if and only if it has free variables. Otherwise, it is closed
Example: the first sentence below is open and the second is closed
$p(Y) \Rightarrow \exists X.q(X, Y)$
$\forall Y.(p(Y) \Rightarrow \exists X.q(X, Y))$

Herbrand base

In this course, we study the Herbrand semantics (Jacques Herbrand (1908-1931) was a French mathematician who died at 23)

The Herbrand base for a vocabulary is the set of all ground relational sentences that can be formed from the constants of the language

Example: for a vocabulary with object constants a and b and relation constants p and q where p has arity 1 and q has arity 2, the Herbrand base is: $\{p(a), p(b), q(a, a), q(a, b), q(b, a), q(b, b)\}$

Remark on the Herbrand base

For a given n-ary relation constant and a finite set of size s of object constants, there are s^n ground relational sentences

Since the number of relation constants in a vocabulary is finite, this means that the Herbrand base is also finite

Truth assignment

Similarly to truth assignment for PL, a truth assignment for a FOL language is a function that maps each ground relational sentence in the Herbrand base to a truth value

Example:

$$
\begin{gathered}
p(a) \rightarrow 1 \\
p(b) \rightarrow 0 \\
q(a, a) \rightarrow 1 \\
q(a, b) \rightarrow 0 \\
q(b, a) \rightarrow 1 \\
q(b, b) \rightarrow 0
\end{gathered}
$$

Truth assignment

The rules for logical sentences in FOL are the same as those for logical sentences in PL

- A truth assignment satisfies a negation $\neg\Phi$ if and only if it does not satisfy Φ
- A truth assignment satisfies a conjunction $(\Phi_1 \wedge \cdots \wedge \Phi_n)$ if and only if it satisfies every Φ_i
- A truth assignment satisfies a disjunction $(\Phi_1 \vee \cdots \vee \Phi_n)$ if and only if it satisfies at least one Φ_i
- A truth assignment satisfies an implication $(\Phi \Rightarrow \Psi)$ if and only if it does not satisfy Φ or does satisfy Ψ
- A truth assignment satisfies an equivalence $(\Phi \Leftrightarrow \Psi)$ if and only if it satisfies both Φ and Ψ or it satisfies neither Φ nor Ψ

Truth assignment

And what about the quantified sentences?

An instance of an expression is an expression in which all free variables have been consistently replaced by ground terms (if one occurrence of a variable is replaced by a ground term, then all occurrences of that variable are replaced by the same ground term)

A universally quantified sentence is true for a truth assignment if and only if every instance of the scope of the quantified sentence is true for that assignment

An existentially quantified sentence is true for a truth assignment if and only if some instance of the scope of the quantified sentence is true for that assignment

A truth assignment satisfies a sentence with free variables if and only if it satisfies every instance of that sentence

How to evaluate FOL sentences?

Evaluation for FOL is similar to evaluation for PL, the only difference is that we need to deal with quantifiers

In order to evaluate a universally quantified sentence, we check that all instances of the scope are true

In order to evaluate an existentially quantified sentence, we check that at least one instance of the scope is true

Example of evaluation

Consider the Herbrand base:
$\{p(a), p(b), q(a,a), q(a,b), q(b,a), q(b,b)\}$

and the following assignment

$$
\begin{aligned}
p(a) &\rightarrow 1\\
p(b) &\rightarrow 0\\
q(a,a) &\rightarrow 1\\
q(a,b) &\rightarrow 0\\
q(b,a) &\rightarrow 1\\
q(b,b) &\rightarrow 0
\end{aligned}
$$

Example of evaluation

Evaluate the truth value of the sentence $\forall X.(p(X) \Rightarrow q(X,X))$ under this assignment

There are two instances of the scope of this sentence:

$$p(a) \Rightarrow q(a,a)$$

$$p(b) \Rightarrow q(b,b)$$

Given the truth assignment (see previous slide), we have

$$(p(a) \Rightarrow q(a,a)) \rightarrow 1$$

$$(p(b) \Rightarrow q(b,b)) \rightarrow 1$$

Since both instances are true, the original quantified sentence is true as well

$$\forall X.(p(X) \Rightarrow q(X,X)) \rightarrow 1$$

Example of evaluation

Evaluate the truth value of the sentence $\forall X.\exists Y.q(X, Y)$
The two possible instances of the scope are:

$$\exists Y.q(a, Y)$$

$$\exists Y.q(b, Y)$$

To determine the truth of $\exists Y.q(a, Y)$, we must find at least one instance of $q(a, Y)$ that is true. The possibilities are:

$$q(a, a)$$

$$q(a, b)$$

The truth assignment is

$$q(a, a) \rightarrow 1$$

$$q(a, b) \rightarrow 0$$

Since one of these instances is true, the existential sentence as a whole is true.

$$\exists Y.q(a, Y) \rightarrow 1$$

Example of evaluation

We do the same for the second existentially quantified $\exists Y.q(b, Y)$
The possible instances are:

$$q(b, a)$$

$$q(b, b)$$

The truth assignment is

$$q(b, a) \rightarrow 1$$

$$q(b, b) \rightarrow 0$$

Since one of these instances is true, the existential sentence as a whole is true.

$$\exists Y.q(b, Y) \rightarrow 1$$

Since both instances of the scope of our original universal sentence are true, the sentence as a whole must be true as well

$$\forall X.\exists Y.q(X, Y) \rightarrow 1$$

Truth table method

As in PL, we can build a truth table for any set of sentences in FOL

We can then use it to determine which truth assignments satisfy a given set of sentences

Example

Truth table for the sentences: $p(a) \vee p(b)$, $\forall X.(p(X) \Rightarrow q(X))$ and $\exists X.q(X)$

$p(a)$	$p(b)$	$q(a)$	$q(b)$	$p(a) \vee p(b)$	$\forall x.(p(x) \Rightarrow q(x))$	$\exists x.q(x)$
1	1	1	1	1	1	1
1	1	1	0	1	0	1
1	1	0	1	1	0	1
1	1	0	0	1	0	0
1	0	1	1	1	1	1
1	0	1	0	1	1	1
1	0	0	1	1	0	1
1	0	0	0	1	0	0
0	1	1	1	1	1	1
0	1	1	0	1	0	1
0	1	0	1	1	1	1
0	1	0	0	1	0	0
0	0	1	1	0	1	1
0	0	1	0	0	1	1
0	0	0	1	0	1	1
0	0	0	0	0	1	0

Assignments that satisfy these sentences: 1, 5, 6, 9, 11

Chess team

Consider again the example of the description of the chess team girls

Using FOL we may formalize this word with:

- 4 object constants: *abby*, *bess*, *cody* and *dana*
- one binary relation/predicate: *likes*

Chess team

If we want to model the word represented by the table:

	Abby	Bess	Cody	Dana
Abby	F	F	T	F
Bess	F	F	T	F
Cody	T	T	F	T
Dana	F	F	T	F

We can model that world writing those ground sentences:

$$\begin{array}{llll}
\neg likes(abby, abby) & \neg likes(abby, bess) & likes(abby, cody) & \neg likes(abby, dana) \\
\neg likes(bess, abby) & \neg likes(bess, bess) & likes(bess, cody) & \neg likes(bess, dana) \\
likes(cody, abby) & likes(cody, bess) & \neg likes(cody, cody) & likes(cody, dana) \\
\neg likes(dana, abby) & \neg likes(dana, bess) & likes(dana, cody) & \neg likes(dana, dana)
\end{array}$$

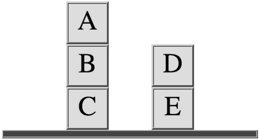
Chess team

But may be we don't have an extensional definition of the world

Consider some english sentences and there translation to FOL

Bess likes Cody or Dana	$likes(bess, cody) \vee likes(bess, dana)$
Abby likes everyone Bess likes	$\forall Y.(likes(bess, Y) \Rightarrow likes(abby, Y))$
Cody likes everyone who likes her	$\forall X.(likes(X, cody) \Rightarrow likes(cody, X))$
Bess likes somebody who likes her	$\exists Y.(likes(bess, Y) \wedge likes(Y, bess))$
Nobody likes herself	$\neg\exists X.likes(X, X)$
Everybody likes somebody	$\forall X.\exists Y.likes(X, Y)$
There is someone everyone likes	$\exists Y.\forall X.likes(X, Y)$

Blocks world



To model this world, we can define 5 object constants: $\{a, b, c, d, e\}$

Various predicates may be defined

The *onblock* predicate may be define by:

$\neg onblock(a, a)$	$onblock(a, b)$	$\neg onblock(a, c)$	$\neg onblock(a, d)$	$\neg onblock(a, e)$
$\neg onblock(b, a)$	$\neg onblock(b, b)$	$onblock(b, c)$	$\neg onblock(b, d)$	$\neg onblock(b, e)$
$\neg onblock(c, a)$	$\neg onblock(c, b)$	$\neg onblock(c, c)$	$\neg onblock(c, d)$	$\neg onblock(c, e)$
$\neg onblock(d, a)$	$\neg onblock(d, b)$	$\neg onblock(d, c)$	$\neg onblock(d, d)$	$onblock(d, e)$
$\neg onblock(e, a)$	$\neg onblock(e, b)$	$\neg onblock(e, c)$	$\neg onblock(e, d)$	$\neg onblock(e, e)$

Blocks world

A block satisfies the *clear* predicate if and only if there is nothing on it. This can be modeled by:

$$\forall Y.(clear(Y) \Leftrightarrow \neg \exists X.onblock(X, Y))$$

A block satisfies the *table* predicate if and only if it is not on some block.

$$\forall X.(table(X) \Leftrightarrow \neg \exists Y.onblock(X, Y))$$

Three blocks satisfy the *stack* predicate if and only if the first is on the second and the second is on the third.

$$\forall X.\forall Y.\forall Z.(stack(X, Y, Z) \Leftrightarrow onblock(X, Y) \wedge onblock(Y, Z))$$

Blocks world

One block is above another block if and only if the first block is on the second block or it is on another block that is above the second block
Also, no block can be above itself

$$\forall X.\forall Z.(above(X,Z) \Leftrightarrow onblock(X,Z) \vee \forall Y.(onblock(X,Y) \wedge above(Y,Z)))$$

$$\neg\exists X.above(X,X)$$

Valid/Unsatisfiable/Contingent sentences

Similar to Propositional Logic

- A sentence is valid if and only if it is satisfied by every truth assignment
- A sentence is unsatisfiable if and only if it is not satisfied by any truth assignment
- A sentence is contingent if and only if there is some truth assignment that satisfies it and some truth assignment that falsifies it

- A sentence is satisfiable if and only if it is satisfied by at least one truth assignment (it is either valid or contingent)
- A sentence is falsifiable if and only if there is at least one truth assignment that makes it false (it is either contingent or unsatisfiable)

Valid sentences

Of course, some ground sentences in FOL are valid, in a similar way as in PL

Examples:

- $p(a) \vee \neg p(a)$ (law of the excluded middle)
- $p(a) \Leftrightarrow \neg\neg p(a)$ (double negation)
- $\neg(p(a) \wedge q(a, b)) \Leftrightarrow (\neg p(a) \vee \neg q(a, b))$ (De Morgan's law)
- $\neg(p(a) \vee q(a, b)) \Leftrightarrow (\neg p(a) \wedge \neg q(a, b))$ (De Morgan's law)

Valid sentences

Common Quantifier Reversal: reversing quantifiers of the same type has no effect on truth assignment

$$\forall X.\forall Y.q(X, Y) \Leftrightarrow \forall Y.\forall X.q(X, Y)$$

$$\exists X.\exists Y.q(X, Y) \Leftrightarrow \exists Y.\exists X.q(X, Y))$$

Valid sentences

Existential Distribution: it is okay to move an existential quantifier inside of a universal quantifier

$$\exists Y.\forall X.q(X,Y) \Rightarrow \forall X.\exists Y.q(X,Y)$$

Example:

There is someone everyone likes: $\exists Y.\forall X.likes(X,Y)$
Everybody likes somebody: $\forall X.\exists Y.likes(X,Y)$

Valid sentences

Negation Distribution: it is okay to distribute negation over quantifiers of either type by flipping the quantifier and negating the scope of the quantified sentence.

$$\neg\forall X.p(X) \Leftrightarrow \exists X.\neg p(X)$$

$$\neg\exists X.p(X) \Leftrightarrow \forall X.\neg p(X)$$

Logical entailment (Definition)

Similar to Propositional Logic

A set of Relational Logic sentences Δ logically entails a sentence Φ (written $\Delta \models \Phi$) if and only if every truth assignment that satisfies Δ also satisfies Φ

Examples

For ground sentences we obtain similar results to the case of propositional logic

Examples:

The sentence $p(a)$ logically entails $(p(a) \vee p(b))$

The sentence $p(a)$ does not logically entail $(p(a) \wedge p(b))$

Examples

The presence of variables allows for additional logical entailments

Example:
The premise $\exists Y.\forall X.q(X, Y)$ entails the conclusion
$\forall X.\exists Y.q(X, Y)$

indeed, if there is some object Y that is paired with every X, then every X has some object Y that it pairs with

remember the example: $\exists Y.\forall X.likes(X, Y) \models \forall X.\exists Y.likes(X, Y)$

Examples

The premise $\forall X.\forall Y.q(X, Y)$ logically entails the conclusion $\forall X.\forall Y.q(Y, X)$

The first sentence says that q is true for all pairs of objects, and the second sentence says the exact same thing $\rightarrow$ we can interchange variables

$\forall X.\forall Y.q(X, Y) \models \forall X.\forall Y.q(Y, X)$

Exercise 1

Suppose we want to model a world with

- two object constants: *jim* and *molly*
- a unary predicate constant: *person*
- a binary predicate constant: *parent*

Say whether each of the following expressions is a syntactically legal sentence of Relational Logic.

1. $parent(jim, molly)$
2. $parent(molly, molly)$
3. $\neg person(jim)$
4. $person(jim, molly)$
5. $parent(molly, z)$
6. $\exists X.parent(molly, X)$
7. $\exists Y.parent(molly, jim)$
8. $\forall Z.(Z(jim, molly) \Rightarrow Z(molly, jim))$

Exercise 2

Consider a language with object constants a and b and relation constants p and q where p has arity 1 and q has arity 2.
Imagine a truth assignment that makes p(a), q(a,b), q(b,a) true and all other ground atoms false
Say whether each of the following sentences is true or false under this truth assignment

1. $\forall X.(p(X) \Rightarrow q(X,X))$
2. $\forall X.\exists Y.q(X,Y)$
3. $\exists Y.\forall X.q(X,Y)$
4. $\forall X.(p(X) \Rightarrow \exists Y.q(X,Y))$
5. $\forall X.p(X) \Rightarrow \exists Y.q(Y,Y)$

Exercise 3

Consider the chess team world that satisfies the following sentences

$\neg likes(abby, abby)$	$likes(abby, bess)$	$\neg likes(abby, cody)$	$likes(abby, dana)$
$likes(bess, abby)$	$\neg likes(bess, bess)$	$likes(bess, cody)$	$\neg likes(bess, dana)$
$\neg likes(cody, abby)$	$likes(cody, bess)$	$\neg likes(cody, cody)$	$likes(cody, dana)$
$likes(dana, abby)$	$\neg likes(dana, bess)$	$likes(dana, cody)$	$\neg likes(dana, dana)$

Say which of the following sentences is satisfied by this state of the world

1. $likes(dana, cody)$
2. $\neg likes(abby, dana)$
3. $likes(bess, cody) \vee likes(bess, dana)$
4. $\forall Y.(likes(bess, Y) \Rightarrow likes(abby, Y))$
5. $\forall Y.(likes(Y, cody) \Rightarrow likes(cody, Y))$
6. $\forall X.\neg likes(X, X)$

Exercise 4

Consider a universe with 2 constants a and b, 3 unary predicates p, q and r, and one binary predicate s. Say whether each of the following sentences is valid, contingent, or unsatisfiable

1. $\forall X.p(X) \Rightarrow \exists X.p(X)$
2. $\exists X.p(X) \Rightarrow \forall X.p(X)$
3. $\forall X.p(X) \Rightarrow p(X)$
4. $\exists X.p(X) \Rightarrow p(X)$
5. $p(X) \Rightarrow \forall X.p(X)$
6. $p(X) \Rightarrow \exists X.p(X)$
7. $\forall X.\exists Y.s(X, Y) \Rightarrow \exists Y.\forall X.s(X, Y)$
8. $\forall X.(p(X) \Rightarrow q(X)) \Rightarrow \exists X.(p(X) \land q(X))$
9. $\forall X.(p(X) \Rightarrow q(X)) \land \exists X.(p(X) \land \neg q(X))$
10. $(\exists X.p(X) \Rightarrow \forall X.q(X)) \lor (\forall X.q(X) \Rightarrow \exists X.r(X))$