Introduction to AI: Resolution in First Order Logic

François Jacquenet

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

October, 2020

UNIVERSITÉ
JEAN MONNET
SAINT-ÉTIENNE

Outline

1. Clausal Form
2. Unification
3. Res. Principle
4. Res. Reasoning
5. Unsatisfiability and Logical Ent.
6. Exercises

Introduction

Resolution in FOL is similar to Resolution in PL

We need to transform quantified logical sentences to clausal form as in PL

We need a new concept: unification

Definitions

As with Propositional Resolution, Resolution works only on expressions in clausal form

A literal is either a relational sentence or a negation of a relational sentence

A clause is a set of literals and, as in Propositional Logic, represents a disjunction of the literals in the set

A clause set is a set of clauses and represents a conjunction of the clauses in the set

Converting relational sentences to clausal form is similar to that for Propositional Logic

Some of the rules are the same

There are a few additional rules to deal with the presence of variables and quantifiers

Transformation to Clausal Form

Step 1: transforming implications

$$\begin{aligned} \Phi \Rightarrow \Psi &\rightarrow \neg\Phi \vee \Psi \\ \Phi \Leftarrow \Psi &\rightarrow \Phi \vee \neg\Psi \\ \Phi \Leftrightarrow \Psi &\rightarrow (\neg\Phi \vee \Psi) \wedge (\Phi \vee \neg\Psi) \end{aligned}$$

Transformation to Clausal Form

Step 2: transforming negations

$$
\begin{array}{lcl}
\neg\neg\Phi & \rightarrow & \Phi \\
\neg(\Phi \wedge \Psi) & \rightarrow & \neg\Phi \vee \neg\Psi \\
\neg(\Phi \vee \Psi) & \rightarrow & \neg\Phi \wedge \neg\Psi \\
\neg\forall X.\Phi & \rightarrow & \exists X.\neg\Phi \\
\neg\exists X.\Phi & \rightarrow & \forall X.\neg\Phi
\end{array}
$$

Transformation to Clausal Form

Step 3: standardize variables

We rename variables so that each quantifier has a unique variable (the same variable is not quantified more than once within the same sentence)

Example: $\forall X.(p(X) \Rightarrow \exists X.q(X)) \rightarrow \forall X.(p(X) \Rightarrow \exists Y.q(Y))$

Transformation to Clausal Form

Step 4: transform existential quantifiers (Skolemisation)

Step 4.1: if an existential quantifier does not occur within the scope of a universal quantifier, we simply drop the quantifier and replace all occurrences of the quantified variable by a new constant (of course, one that does not occur anywhere else in our database)

The constant used to replace the existential variable in this case is called a Skolem constant

Example: $\exists X.p(X) \rightarrow p(a)$

Transformation to Clausal Form

Step 4: Transform existential quantifiers (Skolemisation)

Step 4.2: if an existential quantifier is within the scope of any universal quantifiers, there is the possibility that the value of the existential variable depends on the values of the associated universal variables
Consequently, we cannot replace the existential variable with a constant. Instead, the general rule is to drop the existential quantifier and to replace the associated variable by a term formed from a new function symbol applied to the variables associated with the enclosing universal quantifiers
Any function defined in this way is called a Skolem function
Example:
$\forall X.(p(X) \wedge \exists Z.q(X, Y, Z)) \rightarrow \forall X.(p(X) \wedge q(X, Y, f(X, Y)))$

Transformation to Clausal Form

Step 5: Drop all universal quantifiers

Example:
$\forall X.(p(X) \wedge q(X, Y, f(X, Y))) \rightarrow p(X) \wedge q(X, Y, f(X, Y))$

Transformation to Clausal Form

Step 6: Put into CNF using classical rules

$$
\begin{array}{lcl}
\Phi \vee (\Psi \wedge \Gamma) & \rightarrow & (\Phi \vee \Psi) \wedge (\Phi \vee \Gamma) \\
(\Phi \wedge \Psi) \vee \Gamma & \rightarrow & (\Phi \vee \Gamma) \wedge (\Psi \vee \Gamma) \\
\Phi \vee (\Phi_1 \vee \cdots \vee \Phi_n) & \rightarrow & \Phi \vee \Phi_1 \vee \cdots \vee \Phi_n \\
(\Phi_1 \vee \cdots \vee \Phi_n) \vee \Phi & \rightarrow & \Phi_1 \vee \cdots \vee \Phi_n \vee \Phi \\
\Phi \wedge (\Phi_1 \wedge \cdots \wedge \Phi_n) & \rightarrow & \Phi \wedge \Phi_1 \wedge \cdots \wedge \Phi_n \\
(\Phi_1 \wedge \cdots \wedge \Phi_n) \wedge \Phi & \rightarrow & \Phi_1 \wedge \cdots \wedge \Phi_n \wedge \Phi
\end{array}
$$

Transformation to Clausal Form

Step 7: Generate clauses

$$\begin{aligned} \Phi_1 \vee \cdots \vee \Phi_n &\rightarrow \{\Phi_1, \ldots, \Phi_n\} \\ \Phi_1 \wedge \cdots \wedge \Phi_n &\rightarrow \{\Phi_1\}, \ldots, \{\Phi_n\} \end{aligned}$$

Example 1

Convert $\exists Y.(g(Y) \wedge \forall Z.(r(Z) \Rightarrow f(Y,Z)))$ to CNF

Transform Implication	$\exists Y.(g(Y) \wedge \forall Z.(\neg r(Z) \vee f(Y,Z)))$
Transform Negations	$\exists Y.(g(Y) \wedge \forall Z.(\neg r(Z) \vee f(Y,Z)))$
Rename Variables	$\exists Y.(g(Y) \wedge \forall Z.(\neg r(Z) \vee f(Y,Z)))$
Skolemisation	$g(a) \wedge \forall Z.(\neg r(Z) \vee f(a,Z))$
Drop $\forall$	$g(a) \wedge (\neg r(Z) \vee f(a,Z))$
Put into CNF	$g(a) \wedge (\neg r(Z) \vee f(a,Z))$
Generate clauses	$\{g(a)\}, \{\neg r(Z), f(a,Z)\}$

Example 2

Convert $\neg\exists Y.(g(Y) \wedge \forall Z.(r(Z) \Rightarrow f(Y,Z)))$ to CNF

Transform Implication	$\neg\exists Y.(g(Y) \wedge \forall Z.(\neg r(Z) \vee f(Y,Z)))$
Transform Negations	$\forall Y.(\neg(g(Y) \wedge \forall Z.(\neg r(Z) \vee f(Y,Z)))$
	$\forall Y.(\neg g(Y) \vee \neg\forall Z.(\neg r(Z) \vee f(Y,Z)))$
	$\forall Y.(\neg g(Y) \vee \exists Z.\neg(\neg r(Z) \vee f(Y,Z)))$
	$\forall Y.(\neg g(Y) \vee \exists Z.(\neg\neg r(Z) \wedge \neg f(Y,Z)))$
	$\forall Y.(\neg g(Y) \vee \exists Z.(r(Z) \wedge \neg f(Y,Z)))$
Rename Variables	$\forall Y.(\neg g(Y) \vee \exists Z.(r(Z) \wedge \neg f(Y,Z)))$
Skolemisation	$\forall Y.(\neg g(Y) \vee (r(k(Y)) \wedge \neg f(Y,k(Y))))$
Drop $\forall$	$\neg g(Y) \vee (r(k(Y)) \wedge \neg f(Y,k(Y)))$
Put into CNF	$(\neg g(Y) \vee r(k(Y))) \wedge (\neg g(Y) \vee \neg f(Y,k(Y)))$
Generate clauses	$\{\neg g(Y), r(k(Y))\}, \{\neg g(Y), \neg f(Y,k(Y))\}$

Remark

TAKE CARE: In Propositional Logic, the clause set corresponding to any sentence is logically equivalent to that sentence BUT in Relational Logic, this is not necessarily the case

Example: $\exists X.p(X)$ is not logically equivalent to {p(a)} (it is not even in the same language)

Since the clause exists in a language with an additional object constant, there are truth assignments that satisfy the sentence but not the clause. On the other hand, the converted clause set has a special relationship to the original set of sentences: over the expanded language, the clause set is satisfiable if and only if the original sentence is satisfiable (also over the expanded language)

Exercise

Consider a language with two object constants a and b and one function constant f

Give the clausal form for each of the following sentences in this language

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

Definition

What differentiates FOL Resolution from PL Resolution is unification

In PL resolution, two clauses resolve if they contain complementary literals

The same idea underlies FOL resolution but the positive literal does not need to be identical to the target of the negative literal; it is sufficient that the two can be made identical by substitutions for their variables

Unification is the process of determining whether two expressions can be unified, i.e. made identical by appropriate substitutions for their variables.

Substitution

A substitution is a finite mapping of variables to terms

Example:

$$\{X \leftarrow a, Y \leftarrow f(b), Z \leftarrow V\}$$

is a substitution which means X is to be replaced by a, Y is to be replaced by $f(b)$, and Z is to be replaced by V

The variables being replaced together constitute the domain of the substitution, and the terms replacing them constitute the range
Example: in the substitution

$$\{X \leftarrow a, Y \leftarrow f(b), Z \leftarrow V\}$$

the domain is $\{X, Y, Z\}$, and the range is $\{a, f(b), V\}$

Pure and Impure substitution

A substitution is pure if and only if all replacement terms in the range are free of the variables in the domain of the substitution

Otherwise, the substitution is impure

Example: the substitution $\{X \leftarrow a, Y \leftarrow f(b), Z \leftarrow V\}$ is pure while the substitution $\{X \leftarrow a, Y \leftarrow f(b), Z \leftarrow X\}$ is impure

Applying a substitution

The result of applying a substitution σ to an expression Φ is the expression $\Phi\sigma$ obtained from the original expression by replacing every occurrence of every variable in the domain of the substitution by the term with which it is associated

$$\begin{aligned} q(X,Y)\{X \leftarrow a, Y \leftarrow f(b), Z \leftarrow V\} &= q(a, f(b)) \\ q(X,X)\{X \leftarrow a, Y \leftarrow f(b), Z \leftarrow V\} &= q(a,a) \\ q(X,W)\{X \leftarrow a, Y \leftarrow f(b), Z \leftarrow V\} &= q(a,W) \\ q(Z,V)\{X \leftarrow a, Y \leftarrow f(b), Z \leftarrow V\} &= q(V,V) \end{aligned}$$

Unifier

A substitution σ is a unifier for an expression Φ and an expression Ψ if and only if $\Phi\sigma = \Psi\sigma$ (the result of applying σ to Φ is the same as the result of applying σ to Ψ

If two expressions have a unifier, they are said to be unifiable, otherwise, they are nonunifiable

Example: The expressions $p(X, Y)$ and $p(a, V)$ have a unifier: $\{X \leftarrow a, Y \leftarrow b, V \leftarrow b\}$ and are, therefore, unifiable
The results of applying this substitution to the two expressions are:
$p(X, Y)\{X \leftarrow a, Y \leftarrow b, V \leftarrow b\} = p(a, b)$
$p(a, V)\{X \leftarrow a, Y \leftarrow b, V \leftarrow b\} = p(a, b)$

Most General Unifier

We say that a substitution σ is as general as or more general than a substitution τ if and only if there is another substitution δ such that $\sigma\delta = \tau$

Example: the substitution $\sigma = \{X \leftarrow a, Y \leftarrow V\}$ is more general than $\tau = \{X \leftarrow a, Y \leftarrow f(c), V \leftarrow f(c)\}$ since there is a substitution $\delta = \{V \leftarrow f(c)\}$ such that $\sigma\delta = \tau$:
$\{X \leftarrow a, Y \leftarrow V\}\{V \leftarrow f(c)\} = \{X \leftarrow a, Y \leftarrow f(c), V \leftarrow f(c)\}$

In resolution, we are interested only in unifiers with maximum generality
A most general unifier, or mgu, σ of two expressions has the property that it is as general as or more general than any other unifier

How to calculate an mgu?

Expressions are represented as sequences of subexpressions

Example: p(a,X,f(Y,b),c) is represented as a sequence of 5 elements

- p
- a
- X
- f(Y,b)
- c

How to calculate an mgu?

The process takes two expressions e_1 and e_2 and build the *mgu* of e_1 and e_2 if they are unifiable or else returns false
While unifying two subexpressions, we first apply the substitution beeing built to each of the two expressions; and we then execute the following procedure on the two modified expressions:

1. If two modified expressions being compared are identical, then nothing more needs to be done
2. If two modified expressions are not identical and both expressions are constants, then we fail, since there is no way to make them look alike
3. If one of the modified expressions is a variable, we check whether the second expression contains the variable (this is called occur check)
   - If the variable occurs within the expression, we fail
   - otherwise, we update our substitution to the composition of the old substitution and a new substitution in which we bind the variable to the second modified expression
4. The only remaining possibility is that the two modified expressions are both sequences. In this case, we simply iterate across the expressions, comparing as described above

Example 1

unify: $p(X, b), p(a, Y), \{\}$
- unify: $p, p, \{\}$
- result: $\{\}$
- unify: $X, a, \{\}$
- result: $\{X \leftarrow a\}$
- unify: $Y, b, \{X \leftarrow a\}$
- result: $\{X \leftarrow a, Y \leftarrow b\}$

result: $\{X \leftarrow a, Y \leftarrow b\}$

Example 2

unify: $p(X, X), p(a, Y), \{\}$
- unify: $p, p, \{\}$
- result: $\{\}$
- unify: $X, a, \{\}$
- result: $\{X \leftarrow a\}$
- unify: $a, Y, \{X \leftarrow a\}$
- result: $\{X \leftarrow a, Y \leftarrow a\}$

result: $\{X \leftarrow a, Y \leftarrow a\}$

Example 3

unify: $p(a, f(X, Z), Z), p(X, f(X, Y), b), \{\}$

- unify: $p, p, \{\}$
- result: $\{\}$
- unify: $a, X, \{\}$
- result: $\{X \leftarrow a\}$
  - unify: $f, f, \{X \leftarrow a\}$
  - result: $\{X \leftarrow a\}$
  - unify: $a, a, \{X \leftarrow a\}$
  - result: $\{X \leftarrow a\}$
  - unify: $Z, Y, \{X \leftarrow a\}$
  - result: $\{X \leftarrow a, Z \leftarrow Y\}$
- result: $\{X \leftarrow a, Z \leftarrow Y\}$
- unify: $Y, b, \{X \leftarrow a, Z \leftarrow Y\}$
- result: $\{X \leftarrow a, Z \leftarrow b, Y \leftarrow b\}$

result: $\{X \leftarrow a, Z \leftarrow b, Y \leftarrow b\}$

Exercise

For each of the following pairs of sentences, say whether the sentences are unifiable and give a most general unifier for those that are unifiable

- p(X,X) and p(a,Y)
- p(X,X) and p(f(Y),Z)
- p(X,X) and p(f(Y),Y)
- p(f(X,Y),g(Z,Z)) and p(f(f(W,Z),V),W)

Basic idea

Resolution Principle in FOL is similar to Resolution Principle in PL

$$\frac{\{\Phi_1, \ldots, \Phi, \ldots, \Phi_n\} \quad \{\Psi_1, \ldots, \neg\Psi, \ldots, \Psi_p\}}{\{\Phi_1, \ldots, \Phi_n, \Psi_1, \ldots, \Psi_p\}\sigma}$$

where $\sigma = mgu(\Phi, \Psi)$

Example

$$\frac{\{p(a, Y), r(Y)\} \quad \{\neg p(X, f(X)), q(g(X))\}}{\{r(f(a)), q(g(a))\}}$$

Variable renaming

BUT: consider $\{p(a, X)\}$ and $\{\neg p(X, b)\}$ given our definition we cannot use the Resolution Principle because there is no mgu that unifies $\{p(a, X)\}$ and $\{\neg p(X, b)\}$

In fact, variables are local to each clauses $\rightarrow$ we can rename the variables that are in common

$$\frac{\begin{array}{c}\{\Phi_1, \ldots, \Phi, \ldots, \Phi_n\} \\ \{\Psi_1, \ldots, \neg\Psi, \ldots, \Psi_p\}\end{array}}{\{\Phi_1\tau, \ldots, \Phi_n\tau, \Psi_1, \ldots, \Psi_p\}\sigma}$$

where τ is a variable renaming on $\{\Phi_1, \ldots, \Phi, \ldots, \Phi_n\}$

where $\sigma = mgu(\Phi\tau, \Psi)$

Example

$\{p(a, X)\}$ is renamed to $\{p(a, Y)\}$

$$\frac{\begin{array}{c}\{p(a, Y)\} \\ \{\neg p(X, b)\}\end{array}}{\{\}}$$

Factors

BUT: there is still a problem with $\{p(X), p(Y)\}$ and $\{\neg p(U), \neg p(V)\}$ we should infer the empty clause but we can't

Let's define the concept of factor

If a subset of the literals in a clause Φ has a most general unifier σ, then the clause Φ' obtained by applying σ to Φ is called a factor of Φ

Example: $p(X)$ and $p(f(Y))$ have a most general unifier $\{X \leftarrow f(Y)\}$, so the clause $\{p(f(Y)), r(f(Y), Y)\}$ is a factor of $\{p(X), p(f(Y)), r(X, Y)\}$

Final definition of the Resolution Principle

$$\frac{\Phi \quad \Psi}{((\Phi' - \{\varphi\}) \cup (\Psi' - \{\neg\psi\}))\sigma}$$

where τ is a variable renaming on Φ

where Φ' is a factor of $\Phi\tau$ and $\varphi \in \Phi'$

where Ψ' is a factor of Ψ and $\neg\psi \in \Psi'$

where $\sigma = mgu(\varphi, \psi)$

Example

Consider once again the premises $\{p(X), p(Y)\}$ and $\{\neg p(U), \neg p(V)\}$

The first premise has the factor $\{p(X)\}$, and the second has the factor $\{\neg p(U)\}$, and these two factors resolve to the empty clause in a single step

Basic idea

Reasoning with the Resolution Principle is analogous to reasoning with the Propositional Resolution Principle

- We start with premises
- We apply the Resolution Principle to those premises
- We apply the rule to the results of those applications
- And so forth until we get to our desired conclusion or until we run out of things to do

Example

Suppose we know that:

- Art is the parent of Bob and Bud
- Bob is the parent of Cal
- Bud is the parent of Coe
- We know that grandparents are parents of parents

This may be converted to 4 premises:

- $\{parent(art, bob)\}$
- $\{parent(art, bud)\}$
- $\{parent(bob, cal)\}$
- $\{parent(bud, coe)\}$
- $\{\neg parent(X, Y), \neg parent(Y, Z), grandparent(X, Z)\}$

and we can run a resolution derivation

Example

1.	$\{parent(art, bob)\}$	Premise
2.	$\{parent(art, bud)\}$	Premise
3.	$\{parent(bob, cal)\}$	Premise
4.	$\{parent(bud, coe)\}$	Premise
5.	$\{\neg parent(X, Y), \neg parent(Y, Z), grandparent(X, Z)\}$	Premise
6.	$\{\neg parent(bob, Z), grandparent(art, Z)\}$	1, 5
7.	$\{grandparent(art, cal)\}$	3, 6
8.	$\{\neg parent(bud, Z), grandparent(art, Z)\}$	2, 5
9.	$\{grandparent(art, coe)\}$	4, 8

Similarity with PL

Demonstrating unsatisfiability in FOL is similar to PL

1.	$\{p(a,b), q(a,c)\}$	Premise
2.	$\{\neg p(X,Y), r(X)\}$	Premise
3.	$\{\neg q(X,Y), r(X)\}$	Premise
4.	$\{\neg r(Z)\}$	Premise
5.	$\{q(a,c), r(a)\}$	1, 2
6.	$\{r(a)\}$	5, 3
7.	$\{\}$	6, 4

Logical entailment

Similarly to Logical Entailment in PL, to demonstrated that a set of sentences Δ logically entails a sentence Φ we have to show that the set of formulas $\Delta \cup \{\neg\Phi\}$ is unsatisfiable

Example 1

Suppose we want to model a world where everybody loves somebody and everybody loves a lover

This can by modelled by:

- $\forall X.\exists Y.loves(X, Y)$
- $\forall U.\forall V.\forall W.(loves(V, W) \Rightarrow loves(U, V))$

Our goal is to show that everybody loves everybody $(\forall X.\forall Y.loves(X, Y))$

Strategy: add the negation of the desired conclusion to the premises and convert to clausal form. We get this set of clauses:

- $\{loves(X, f(X))\}$
- $\{\neg loves(V, W), loves(U, V)\}$
- $\{\neg loves(a, b)\}$

Example 1

We can now run a resolution proof

1.	$\{loves(X, f(X))\}$	Premise
2.	$\{\neg loves(V, W), loves(U, V)\}$	Premise
3.	$\{\neg loves(a, b)\}$	Premise
4.	$\{loves(U, X)\}$	1, 2
5.	$\{\}$	3, 4

Example 2

We want to model a world where:

- We know that every horse can outrun every dog
- Some greyhounds can outrun every rabbit
- Greyhounds are dogs
- The relationship of being faster is transitive
- Harry is a horse
- Ralph is a rabbit

We can write FOL sentences:

- $\forall H.\forall D.(horse(H) \wedge dog(D) \Rightarrow faster(H, D))$
- $\exists G.(greyhound(G) \wedge \forall R.(rabbit(R) \Rightarrow faster(G, R)))$
- $\forall G.(greyhound(G) \Rightarrow dog(G))$
- $\forall X.\forall Y.\forall Z.(faster(X, Y) \wedge faster(Y, Z) \Rightarrow faster(X, Z))$
- $horse(harry)$
- $rabbit(ralph)$

Example 2

We want to prove that Harry is faster than Ralph. We need to add $\neg f(harry, ralph)$ to the premises

1.	$\{\neg horse(H), \neg dog(D), faster(H, D)\}$	Premise
2.	$\{greyhound(a)\}$	Premise
3.	$\{\neg rabbit(R), faster(a, R)\}$	Premise
4.	$\{\neg greyhound(G), dog(G)\}$	Premise
5.	$\{\neg faster(X, Y), \neg faster(Y, Z), faster(X, Z)\}$	Premise
6.	$\{horse(harry)\}$	Premise
7.	$\{rabbit(ralph)\}$	Premise
8.	$\{\neg faster(harry, ralph)\}$	Negated Goal
9.	$\{dog(a)\}$	2, 4
10.	$\{\neg dog(D), faster(harry, D)\}$	6, 1
11.	$\{faster(harry, a)\}$	9, 10
12.	$\{faster(a, ralph)\}$	7, 3
13.	$\{\neg faster(a, Z), faster(harry, Z)\}$	11, 5
14.	$\{faster(harry, ralph)\}$	12, 13
15.	$\{\}$	14, 8

Exercise 1

Give all resolvents, if any, for each of the following pairs of clauses

- $\{p(X, f(X)), q(X)\}$ and $\{\neg p(a, Y), r(Y)\}$
- $\{p(X, b), q(X)\}$ and $\{\neg p(a, X), r(X)\}$
- $\{p(X), p(a), q(X)\}$ and $\{\neg p(Y), r(Y)\}$
- $\{p(X), p(a), q(X)\}$ and $\{\neg p(Y), r(Y)\}$
- $\{p(a), q(Y)\}$ and $\{\neg p(X), \neg q(b)\}$
- $\{p(X), q(X, X)\}$ and $\{\neg q(a, f(a))\}$

Exercise 2

Given the clauses $\{p(a), q(a)\}$, $\{\neg p(X), r(X)\}$, $\{\neg q(a)\}$, use Resolution to derive the clause $\{r(a)\}$

Exercise 3

Given the premises $\forall X.(p(X) \Rightarrow q(X))$ and $\forall X.(q(X) \Rightarrow r(X))$, use Resolution to prove the conclusion $\forall X.(p(X) \Rightarrow r(X))$

Exercise 4

Given $\forall X.(p(X) \Rightarrow q(X))$, use Resolution to prove $\forall X.p(X) \Rightarrow \forall X.q(X)$

Exercise 5

Use Resolution to prove $\forall X.(((p(X) \Rightarrow q(X)) \Rightarrow p(X)) \Rightarrow p(X))$

Exercise 6

Given

- $\forall X.\forall Y.\forall Z.(p(X,Y) \wedge p(Y,Z) \Rightarrow p(X,Z))$
- $\forall X.p(X,a)$
- $\forall Y.p(a,Y)$

use Resolution to prove $\forall X.\forall Y.p(X,Y)$

Exercise 7

Given the following first-order sentences:

a) $\forall X.(p(X) \Rightarrow \exists Y.q(Y))$

b) $\neg\exists X.(q(X) \wedge \exists Y.\neg w(Y))$

c) $\forall X.((p(X) \wedge w(X)) \Rightarrow s(X))$

d) $p(mary)$

Using resolution reasoning, show that: s(mary)

Exercise 8

Here are some informations about a simple world:

- *The antelope Emma is a herbivorous animal.*
- *The lion Harry is a ferocious animal.*
- *If an animal is ferocious then it is a carnivore.*
- *If an animal is carnivorous then it can eat meat.*
- *If an animal is herbivorous then it can eat grass.*
- *All carnivorous drink water.*
- *All herbivorous drink water.*
- *If an animal A1 is a carnivorous and an animal A2 is a herbivorous, then A1 can eat A2.*
- *If and animal eats some food then it consumes this food.*
- *If an animal drinks some liquid then it consumes this liquid.*

Using resolution reasoning in first order logic, answer these questions : *"Is there a ferocious animal in this world, and what does it consume?"*

Exercise 9

Given the three sentences below:

- Every human is a primate.
- Dolphins are not primates.
- There are dolphins who are intelligent.

prove, by a FOL resolution proof, that *it is possible not to be a human and to be intelligent*.