02156 - Logical Systems and Logic Programming
Fall 2021

DTU - Technical University of Denmark

Date submitted: November 23, 2021

Assignment 5

Daniel F. Hauge (s201186)

Problem 1

See programs in **02156-A5-s201186.pl**

Question 1.1

The program should succed when list given is a non empty list and has the first element occur twice.

```
?- member1([1,3,1,2]).
true.
?- member1([1,1]).
true.
?- member1([1,1,1]).
true.
```

The program should fail if first element does not occur twice, or the list given is empty.

```
?- member1([1,3,4,2]).
false.
?- member1([]).
false.
?- member1([1]).
false.
```

The program can be queried with a variable and will give a list where first element occur twice.

```
?- member1(X).
X = [_3324, _3324|_3332].
```

Question 1.2

The program succeds with same solutions as the sample query presented in the assignment description:

```
?- member(L,[[a,[a,b,[],c],[[d]]],[123,[4,5]],[[[]]],f(a)]), p(L,R).
L = [a, [a, b, [], c], [[d]]],
R = [a, b, c, d] ;
L = [123, [4, 5]],
R = [4, 5, 123] ;
L = [[[]]],
R = [] ;
L = f(a),
R = [f(a)].
```

The program should fail when:

- List is not sorted
- Term is missing in List
- List contains List
- Terms are not given.

```
?- p([123, [4, 5]], [5,4,123]).
false.
?- p([123, [4, 5]], [4,123]).
false.
p([123, [4,5], [2]], [4, 123, [2]]).
false.
?- p(R,[1,2]).
false.
```

The program can be queried with variables:

```
?- p([1,4,5,[1,2]],L).
L = [1, 2, 4, 5].
false.
?- p(R,L).
R = L, L = [].
```

Problem 2

See programs in **02156-A5-s201186.pl**

Question 2.1

The program **selectlist(?List)** will succesfully return a list of all nouns and verbs which have a sort order of 100 or less when queried with variable. The program will succed if given the correct list.

```
?- selectlist(List), length(List,L).
List = [be, come, do, find, get, give, go, have, know|...],
L = 20.
?- selectlist([be, come, do, find, get,
give, go, have, know, look,
make, people, say, see,
take, think, time, use,
way, year] ).
true.
```

The program will fail when:

- The list contains duplicates
- The list is missing a noun or verb that should be included
- The list includes a wrong noun or verb.

N.B. **...** indicates the first part of the correct list.

```
?- selectlist([..., year, year]).
false.
?- selectlist([...]).
false.
?- selectlist([..., year, witness]).
false.
```

Question 2.2

The program **dump** succesfully finds all words that are both adjectives and adverbs and prints all additional categories with the word.

```
?- dump.
above prep
back n
back v
close v
```

Given the data from database.pl, then the complete list yields 29 lines in total. See appendix for full output.

Problem 3

Question 3.1

Refuting the validity of the formular, we negate the formular and try to find counter example.

The clausal form (PCNF) of the formular is found using skolemization:

$$
\begin{aligned}
&\neg(\exists x \forall y p(x,y) \rightarrow \forall y \exists x p(x,y)) \\
&\equiv \neg(\neg(\exists x \forall y p(x,y)) \vee \forall y \exists x p(x,y)) \\
&\equiv \neg\neg(\exists x \forall y p(x,y)) \wedge \neg(\forall y \exists x p(x,y)) \\
&\equiv \exists x \forall y p(x,y) \wedge \neg(\forall y \exists x p(x,y)) \\
&\equiv \exists x \forall y p(x,y) \wedge \exists y \forall x \neg p(x,y)) \\
&\equiv \exists x_1 \forall y_1 p(x_1,y_1) \wedge \exists y_2 \forall x_2 \neg p(x_2,y_2)) \\
&\equiv \exists x_1 \forall y_1 \exists y_2 \forall x_2 (p(x_1,y_1) \wedge \neg p(x_2,y_2))) \\
&\equiv \forall y_1 \exists y_2 \forall x_2 (p(a,y_1) \wedge \neg p(x_2,y_2))) \\
&\equiv \forall y_1 \forall x_2 (p(a,y_1) \wedge \neg p(x_2,f(y_1))))
\end{aligned}
$$

The 2 clauses are now clear:

$$[p(a,y_1)], [\neg p(x_2,f(y_1))]$$

The clauses do not clash, but as the prenex conjunctive normal form is used, a unification with the following substitution can be used to get the most general unifier:

$$\{a \leftarrow x_2, y_1 \leftarrow f(y_1)\}$$

The clauses become clashing literals:

$$[p(a,y_1)], [\neg p(a,y_1)]$$

Immedietly, it should be clear that the empty clause can be derived from the 2 clashing clauses, thus making the formular unsatisfiable. But as the formular was refuted by negation, the contradiction becomes a tautology and the original formular is therefor valid.

Question 3.2

Refuting the validity of the formular, we negate the formular and try to find counter example.

The clausal form (PCNF) of the formular is found using skolemization:

$$
\begin{aligned}
&\neg((\forall x(p(x) \rightarrow \neg\exists y q(y,x))) \rightarrow (p(a) \rightarrow \neg q(a,a))) \\
&\equiv \neg(\neg\forall x(p(x) \rightarrow \neg\exists y q(y,x))) \vee (p(a) \rightarrow \neg q(a,a))) \\
&\equiv \neg(\neg\forall x(p(x) \rightarrow \neg\exists y q(y,x))) \vee (\neg p(a) \vee \neg q(a,a))) \\
&\equiv \neg(\neg\forall x(\neg p(x) \vee \neg\exists y q(y,x))) \vee (\neg p(a) \vee \neg q(a,a))) \\
&\equiv (\neg\neg\forall x(\neg p(x) \vee \neg\exists y q(y,x))) \wedge \neg(\neg p(a) \vee \neg q(a,a))) \\
&\equiv (\forall x(\neg p(x) \vee \forall y \neg q(y,x))) \wedge p(a) \wedge q(a,a)) \\
&\equiv \forall x \forall y((\neg p(x) \vee \neg q(y,x)) \wedge p(a) \wedge q(a,a)))
\end{aligned}
$$

The 3 clauses are now clear:

$$[\neg p(x), \neg q(y,x)], [p(a)], [q(a,a)]$$

As the prenex conjunctive normal form is used, a unification with the following substitution can be used to get the most general unifier:

$$\{a \leftarrow x, a \leftarrow y\}$$

Then the set of clauses becomes:

$$\{[\neg p(a), \neg q(a,a)], [p(a)], [q(a,a)]\}$$

Resolution is then used with the clauses:

1.	$\neg p(a), \neg q(a,a)$	
2.	$p(a)$	
3.	$q(a,a)$	
4.	$\neg q(a,a)$	1,2
5.	$\square$	3,4

The empty clause $\square$ is reached. The formular is unsatisfiable, and with refutation the original formular is then valid.

Question 3.3

Refuting the validity of the formular, we negate the formular and try to find counter example.

The clausal form (PCNF) of the formular is found using skolemization:

$$
\begin{aligned}
&\neg((\forall x p(x) \land \forall x q(x)) \rightarrow \forall x(p(x) \land q(x))) \\
&\equiv \neg((\forall x_1 p(x_1) \land \forall x_2 q(x_2)) \rightarrow \forall x_3(p(x_3) \land q(x_3))) \\
&\equiv \neg(\neg(\forall x_1 p(x_1) \land \forall x_2 q(x_2)) \lor \forall x_3(p(x_3) \land q(x_3))) \\
&\equiv (\neg\neg(\forall x_1 p(x_1) \land \forall x_2 q(x_2)) \land \neg\forall x_3(p(x_3) \land q(x_3))) \\
&\equiv ((\forall x_1 p(x_1) \land \forall x_2 q(x_2)) \land \neg\forall x_3(p(x_3) \land q(x_3))) \\
&\equiv ((\forall x_1 p(x_1) \land \forall x_2 q(x_2)) \land \exists x_3 \neg(p(x_3) \land q(x_3))) \\
&\equiv ((\forall x_1 p(x_1) \land \forall x_2 q(x_2)) \land \exists x_3(\neg p(x_3) \lor \neg q(x_3))) \\
&\equiv \exists x_3 \forall x_1 \forall x_2 (p(x_1) \land q(x_2) \land (\neg p(x_3) \lor \neg q(x_3))) \\
&\equiv \forall x_1 \forall x_2 (p(x_1) \land q(x_2) \land (\neg p(a) \lor \neg q(a)))
\end{aligned}
$$

The 3 clauses are now clear:

$$[p(x_1)], [q(x_2)], [\neg p(a), \neg q(a)]$$

As the prenex conjunctive normal form is used, a unification with the following substitution can be used to get the most general unifier:

$$\{a \leftarrow x_1, a \leftarrow x_2\}$$

Then the set of clauses becomes:

$$\{[p(a)], [q(a)], [\neg p(a), \neg q(a)]\}$$

Resolution is then used with the clauses:

1.	$p(a)$	
2.	$q(a)$	
3.	$\neg p(a), \neg q(a)$	
4.	$\neg q(a)$	1,3
5.	$\square$	2,4

The empty clause $\square$ is reached. The formular is unsatisfiable, and with refutation the original formular is then valid.

Question 3.4

Refuting the validity of the formular, we negate the formular and try to find counter example.

The clausal form (PCNF) of the formular is found using skolemization:

$$\begin{aligned}
&\neg((\forall x \exists y (p(x) \land \neg p(y))) \rightarrow \neg q(a)) \\
&\equiv \neg(\neg(\forall x \exists y (p(x) \land \neg p(y))) \lor \neg q(a)) \\
&\equiv \neg\neg(\forall x \exists y (p(x) \land \neg p(y))) \land \neg\neg q(a)) \\
&\equiv \forall x \exists y (p(x) \land \neg p(y)) \land q(a)) \\
&\equiv \forall x \exists y (p(x) \land \neg p(y) \land q(a)) \\
&\equiv \forall x (p(x) \land \neg p(f(x)) \land q(a))
\end{aligned}$$

The 3 clauses are now clear:

$$[p(x)], [\neg p(f(x))], [q(a)]$$

As the prenex conjunctive normal form is used, a unification with the following substitution can be used to get the most general unifier:

$$\{x \leftarrow f(x)\}$$

Then the set of clauses becomes:

$$\{[p(x)], [\neg p(x)], [q(a)]\}$$

Resolution is then used with the clauses:

$$\begin{array}{lll}
1. & p(x) & \\
2. & \neg p(x) & \\
3. & q(a) & \\
4. & \square & 1,2
\end{array}$$

The empty clause $\square$ is reached. The formular is unsatisfiable, and with refutation the original formular is then valid.

Problem 4

The shortened SeCav is included outcommented in **02156-A4-s201186.pl**

Appendix

Question 2.2 dump

Full output from **dump** program from Question 2.2

```
?- dump.
above prep
back n
back v
close v
direct v
fair n
like conj
like n
like prep
like v
little det
long v
near prep
open v
opposite n
outside n
outside prep
overall n
past n
past prep
right interjection
right n
round n
round prep
round v
short n
well interjection
well n
wrong n
true.
```