02156 - Logical Systems and Logic Programming
Fall 2021

DTU - Technical University of Denmark

Date submitted: November 9, 2021

Assignment 4

Daniel F. Hauge (`s201186`)

Problem 1

See programs in **02156-A4-s201186.pl**

Question 1.1

The program should succesfully:

- Query the students whom scored over 40 on the exam.
- Succed if given the correct sorted list of students whom scores over 40 on the exam.

```
?- students(S).
S = [alice, bruce, carol, dorit, erica, james, peter, xenia].
?- students([alice, bruce, carol, dorit, erica, james, peter, xenia]).
true.
```

The program should fail when:

- The list given is not sorted.
- The list given is missing a student.
- The list given contains a student whom did not score higher than 40 on the exam
- The list given contains duplications

```
?- students([xenia, alice, bruce, carol, dorit, erica, james, peter]).
false.
?- students([alice, bruce, carol, dorit, erica, james, peter]).
false.
?- students([alice, bruce, carol, daniel, dorit, erica, james, peter]).
false.
?- students([alice, alice, bruce, carol, dorit, erica, james, peter,
xenia]).
false.
```

Question 1.2

The program should succesfully:

- Query the amount of reward money
- Succed only when given the correct amount of reward money.

```
?- money(M).
M = 5000.
?- money(5000).
true.
?- money(4000).
false.
```

Problem 2

Question 2.1

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

The clausal form (CNF) of the formular is found:

$$\begin{aligned}
\neg((p \wedge q) \rightarrow (q \wedge p)) &\equiv \neg((p \wedge q) \vee (q \wedge p)) \\
&\equiv \neg((\neg p \vee \neg q) \vee (q \wedge p)) \\
&\equiv (\neg(\neg p \vee \neg q) \wedge \neg(q \wedge p)) \\
&\equiv (\neg(\neg p \vee \neg q) \wedge (\neg q \vee \neg p)) \\
&\equiv ((\neg\neg p \wedge \neg\neg q) \wedge (\neg q \vee \neg p)) \\
&\equiv p \wedge q \wedge (\neg q \vee \neg p)
\end{aligned}$$

The clauses are now clear and can be used in resolution:

1.	p	
2.	q	
3.	$\bar{q}, \bar{p}$	
4.	$\bar{q}$	1,3
5.	$\square$	2,4

The empty clause $\square$ is derived, thus no interpretation of the original set of clauses can hold, it is unsatisfiable. As a result of refutation hence negation, the formular then holds in all interpretation. The formular is valid and therefor a tautology.

Question 2.2

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

$$\begin{array}{c}
\neg((\forall x \exists y (p(x) \wedge \neg p(y))) \rightarrow \neg q(a)) \\
| \\
\forall x \exists y (p(x) \wedge \neg p(y)), \neg\neg q(a) \\
| \\
\forall x \exists y (p(x) \wedge \neg p(y)), q(a) \\
| \\
\forall x \exists y (p(x) \wedge \neg p(y)), \exists y (p(a) \wedge \neg p(y)), q(a) \\
| \\
\forall x \exists y (p(x) \wedge \neg p(y)), p(a) \wedge \neg p(b), q(a) \\
| \\
\forall x \exists y (p(x) \wedge \neg p(y)), p(a), \neg p(b), q(a) \\
| \\
\forall x \exists y (p(x) \wedge \neg p(y)), \exists y (p(b) \wedge \neg p(y)), p(a), \neg p(b), q(a) \\
| \\
\forall x \exists y (p(x) \wedge \neg p(y)), p(b) \wedge \neg p(c), p(a), \neg p(b), q(a) \\
| \\
\forall x \exists y (p(x) \wedge \neg p(y)), p(b), \neg p(c), p(a), \neg p(b), q(a) \\
| \\
\times
\end{array}$$

The tableau closes as we see the contradiction with: $p(b)$ and $\neg p(b)$. This concludes a complete closed tableau, thus we can say that the formula is valid. Because we negated the formular, the contradiction becomes a tautology.

Problem 3

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

Notes on proving $(\forall x p(x) \land \forall x q(x)) \to \forall x(p(x) \land q(x))$

- N.B Second Ext rule removes unnecesary formulars.