02156 - Logical Systems and Logic Programming
Fall 2021

DTU - Technical University of Denmark

Date submitted: October 4, 2021

Assignment 2

Daniel F. Hauge (s201186)

Problem 1

See programs in 02156-A2-s201186.pl

Question 1.1

The predicate succeeds if the word is in the list with more than one category:

```
?- ambiguous(above).
true.
?- ambiguous(about).
true.
```

The predicate succeeds only if (+Word) is in the list with more than one category, thus should fail if the word only has one category or is not in the list:

```
?- ambiguous(a).
false.
?- ambiguous(absent).
false.
```

The predicate should fail if (+Word) is a variable.

```
?- ambiguous(X).
false.
```

Question 1.2

Display will print the n'th most frequent words based of the list.

```
?- display(500).
a det
able a
about adv
about prep
true.
?- display(1000).
a det
ability n
able a
about adv
about prep
above adv
above prep
true.
```

Display will fail if no words are in the list for the given n, also if the n given is not an instantiated integer.

```
?- display(1).
false.
?- display(X).
false.
```

Problem 2

Considering the formular: $(p \wedge q) \rightarrow (q \wedge p)$

Question 2.1

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

$$
\begin{array}{c}
\neg((p \wedge q) \rightarrow (q \wedge p)) \\
| \\
(p \wedge q), \neg(q \wedge p) \\
| \\
p, q, \neg(q \wedge p) \\
\begin{array}{cc}
p, q, \neg q & p, q, \neg p \\
| & | \\
\times & \times
\end{array}
\end{array}
$$

Concluding with a complete closed tableau, thus we can say that the formula is valid hence we can say the propositional formular is a tautology.

Question 2.2

The gentzen system, can be seen as previusly constructed tableaux tree upside down and signs reversed, such that we get the following:

1.	$\vdash \neg p, \neg q, p$	Axiom
2.	$\vdash \neg p, \neg q, q$	Axiom
3.	$\vdash \neg p, \neg q, (q \wedge p)$	$\beta\wedge, 1, 2$
4.	$\vdash \neg(p \wedge q), (q \wedge p)$	$\alpha\wedge, 3$
5.	$\vdash (p \wedge q) \rightarrow (q \wedge p)$	$\alpha \rightarrow, 4$

Therefor: $\vdash (A \wedge B) \rightarrow (B \wedge A)$

Question 2.3

Using the hilbert systems axioms and rules:

- One rule MP: $\vdash A, \vdash A \to B / \vdash B$
- 1: $\vdash A \to B \to A$
- 2: $\vdash (A \to B \to C) \to (A \to B) \to A \to C$
- 3: $\vdash (\neg B \to \neg A) \to A \to B$
- Deduction rule: $U \subset A \vdash B / U \vdash A \to B$
- Assumption rule: $U \vdash A_i (A_i \in U)$

The proof can completed:

1.	$\{q \to \neg p, \neg\neg p\} \vdash q \to \neg p$	Assuption
2.	$\{q \to \neg p, \neg\neg p\} \vdash p \to \neg q$	Contrapositive 1
3.	$\{q \to \neg p, \neg\neg p\} \vdash (p \to \neg q) \to (q \to \neg p)$	Assumption
4.	$\{q \to \neg p, \neg\neg p\} \vdash q \to \neg p$	MP 2,3
5.	$\{q \to \neg p, \neg\neg p\} \vdash \neg\neg(q \to \neg p)$	Double negation 4
6.	$\{q \to \neg p\} \vdash \neg\neg p \to \neg\neg(q \to \neg p)$	Deduction 5
7.	$\{q \to \neg p\} \vdash \neg(q \to \neg p) \to \neg p$	Contrapositive 6
8.	$\{q \to \neg p\} \vdash p \to (q \to \neg p)$	Contrapositive 7
9.	$\vdash (q \to \neg p) \to (p \to (q \to \neg p))$	Deduction 8
10.	$\vdash \neg(p \to (q \to \neg p)) \to \neg(q \to \neg p)$	Contrapositive 9
11.	$\vdash (p \wedge q) \to (q \wedge p)$	Def. of $\wedge$

Note, by definition of logical conjunction and negation of implication, it can be seen how the formular is derived.

$$\neg(q \to \neg p) \equiv q \wedge p$$

$$\neg(p \to (q \to \neg p)) \equiv p \wedge \neg(q \to \neg p)$$

$$p \wedge (q \wedge p) \equiv p \wedge q$$

Thus, concluding the hilbert system complete.

Problem 3

See programs in 02156-A2-s201186.pl

Question 3.1

The predicate succeds if the first and last element is the same.

```
?- firstlast([1,2,3,4,5,6,5,4,3,2,1]).
true ;
false.
?- firstlast([4,4]).
true ;
false.
```

The predicate fails if first and last element is not the same.

```
?- firstlast([6,6,2,7,4,1,3,7,6,6,9]).
false.
```

The predicate fails when there is not atleast 2 elements:

```
?- firstlast([6]).
false.
?- firstlast([]).
false.
```

The predicate can be queried with varibles:

```
?- firstlast(X).
X = [_634, _634] ;
X = [_634, _1300, _634] ;
X = [_634, _1300, _1972, _634] .
?- firstlast([X|[5]]).
X = 5 .
?- firstlast([5|X]).
X = [5] ;
X = [_4766, 5] .
```

Question 3.2

All tests from Question 3.1 has been done with $firstlasta$. All tests give identical results (apart from generic values).

Problem 4

See programs in 02156-A2-s201186.pl

Question 4.1

The predicate succeeds if a student scored the given integer for a test or exam.

```
test(11).
true.
test(77).
true.
test(42).
true.
```

The predicate fails if no student scored the given integer.

```
test(12).
false.
test(100).
false.
```

The predicate fails if (+Integer) given is not an instantiated integer.

```
?- test(X).
false.
```

Question 4.2

The program *problem* prints the students names where their test scores are more than twice their exam scores.

```
?- problem.
alice
carol
true.
```

Problem 5

The shortened SeCav is included outcommented in 02156-A2-s201186.pl

Notes on proving $((p \to q) \to p) \to p$.

- $\Gamma \vdash \triangle$ can be proven if and only if $\vdash \neg\Gamma, \triangle$,
  therefor we start applying rules for: $\neg(((p \to q) \to p) \to p)$
- Alpha implication is applied: $\neg(A_1 \to A_2)$
- Beta implication is applied: $B_1 \to B_2$
- Alpha implication is applied to the remaining implication.
- Ext rule is applied to remove q as it is not neccesary for the proof, but also to put the non-negated formular/p first for the basic rule.
- Basic rule is used to end the proof.