Assignment 2

02156 Logical Systems and Logic Programming, Fall 2023

October 8, 2023

Thomas Schiøler Hansen s214968

1 Table of contents

Contents

2 Problem 1

2.1 Problem 1.1

Please refer to the solution written in the Assignment2.txt file. I have issued some tests that test for some normal cases, special cases and variable cases. Please see them below:
Normal case:

Should succeed, as "about" appears with both "adv" and "prep" categories:
?- ambiguous(about).
true.

Should fail, as "abandon" only appears with the "v" category:
?- ambiguous(abandon).
false.

Special case:

Should fail, as "nonexistent" does not appear in the word frequency list:
?- ambiguous(nonexistent).
false.

Variable test:
This query should succeed with multiple solutions, each representing an ambiguous word if any exist:
?- ambiguous(Word).
true.

2.2 Problem 1.2

In this question I have created the prolog code so that display(N) will show all the words and their word classes whose SortOrder is less than or equal to N.
Please refer to the solution written in the Assignment2.txt file. I have issued some tests that

test for some normal cases, special cases and variable cases. Please see them below:

Normal cases:

Should display the top 3 words and categories without frequencies:

```
?- display(5).
a det
false.
```

Should display the top 5 words and categories without frequencies:

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

Special cases:

If we use a higher N than the highest SortOrder there is in the list:

It should display each unique word there is without any errors, even though the display value is larger than largest SortOrder in the list.

```
?- display(9000).
a det
abandon v
abbey n
ability n
able a
abnormal a
abolish v
abolition n
abortion n
```

about adv

about prep

above a

above adv

above prep

abroad adv

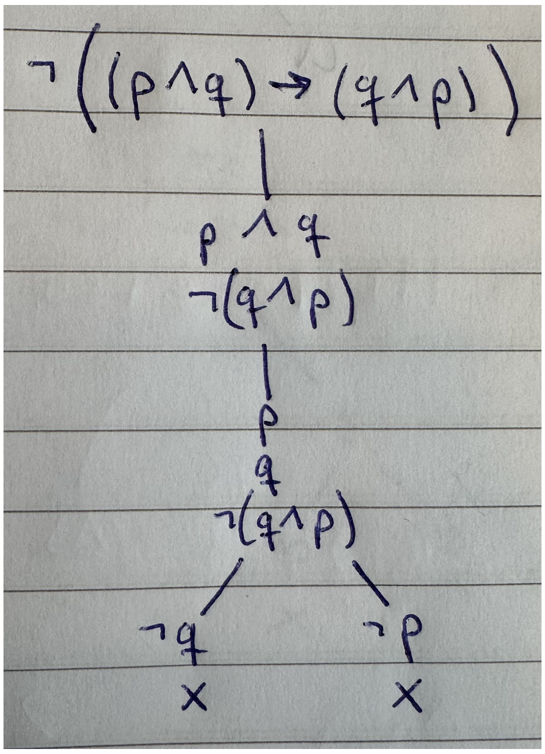
abruptly adv

false.

3 Problem 2

3.1 Problem 2.1

Below please see the semantic tableau I have created for the formula (p ∧ q) → (q ∧ p). You can see that the branches close for the inverted formula meaning that the original formula is valid.



3.2 Problem 2.2

Since there is a closed semantic tableau for ¬ (p ∧ q) → (q ∧ p), we can now create a proof of the formula (p ∧ q) → (q ∧ p) in the Gentzen system. Please see it below:

1. $\vdash q, \neg p, \neg q$	[Axiom]
2. $\vdash p, \neg p, \neg q$	[Axiom]
3. $\vdash (q \wedge p), \neg p, \neg q$	[$\beta\wedge$, 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]

3.3 Problem 2.3

Below please see the proof of the formula (p ∧ q) → (q ∧ p) in the Hilbert system:

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

4 Problem 3

In both problem 3.1 and 3.2 I assume that it is not the same element in the start and end of the list if there is only 1 element in the list. Please refer to the solution written in the Assignment2.txt file. I have issued some tests that test for some normal cases, special cases and variable cases. Please see them below:

4.1 Problem 3.1

Normal cases:
The list starts and ends with 1, so it returns true as is expected.

```
?- firstlast([1,1,1,1]).
true ;
false.
```

The start and end of the list is different so it returns false as expected.

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

The list starts and ends with a, so it returns true as is expected.

```
?- firstlast([a,b,c,a]).
true ;
false.
```

The start and end of the list is different so it returns false as expected.

```
?- firstlast([a,b,c,d]).
false.
```

Now for some special cases:

One element in the list, returns false as it is what I assumed in problem 3:

```
?- firstlast([1]).
false.
```

No elements in the list returns false as there is no start or end elements:

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

Variable cases:

The list starts and ends with X, so it returns true as is expected.

```
?- firstlast([X,X]).
true ;
```

```
false.
```

The list first returns false when X and Y are the same, because then the list starts and ends with the same element.

```
?- firstlast([X,Y]).
X = Y ;
false.
```

4.2 Problem 3.2

I have now constructed some tests for a few different cases. First, the normal cases:
Below is the output from the first test of ”firstlasta”. Here it returns true as expected because the list starts and ends with a ”1”.

```
?- firstlasta([1,1,1,1]).
true.
```

The next test returns false as it should because it starts with ”1” and ends with ”3”

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

The list starts and ends with a so it returns true.

```
?- firstlasta([a,b,c,a]).
true.
```

Again it returns false below as is expected because it does not start and end with the same.

```
?- firstlasta([a,b,c,d]).
false.
```

I will now test it for some special cases as seen below:
The first test is with only one element in the list, which returns false as was expected because I understood that it has to be two different elements:

```
?- firstlasta([1]).
false.
```

Next is when there are no elements at all in the list. It is not surprising that it also returns false:

```
?- firstlasta([]).
false.
```

I will now test for some variable cases below:
Here the first and last elements are both X, so it returns true as it should.

```
?- firstlasta([X,X]).
true ;
false.
```

The last test also returns true when X=Y, meaning that the start X would be the same as the ending Y, so that is what I expected.

```
?- firstlasta([X,Y]).
X = Y ;
false.
```

5 Problem 4

Please refer to the solution written in the Assignment2.txt file. I have issued some tests for the two problems below.

5.1 Problem 4.1

The code for this problem is pretty simple. We simply extract whether it was a test or an exam and the points scored in that.
Let's test if it works how we expect it to. Firstly, we will test for a few normal cases:

Here we want to see if someone scored 88 in the exam or the test. It should return true, as it does, because Bruce scored 88 in the exam:

```
?- test(88).
true.
```

Next we test for 33, which returns true because Carol scored 33 in the exam.

```
?- test(33).
```

```
true.
```

We will now test for some special cases:

No one scored 0 so it makes sense to be false:

```
?- test(0).
false.
```

Again, no one scored 100, so it is also expected to be false:

```
?- test(100).
false.
```

Now let's try to test for a somewhat variable case:

This returns all scores as it is expected to:

```
?- test(Score).
Score = 50 ;
Score = 99 ;
Score = 22 ;
Score = 77 ;
Score = 50 ;
Score = 22 ;
Score = 42 ;
Score = 11 ;
Score = 88 ;
Score = 33 ;
Score = 50 ;
Score = 66 ;
Score = 77.
```

5.2 Problem 4.2

This code takes the test score for each student and exam score for each student and checks whether their test score is twice as big as the exam score.
Since alice and carol were the only two to get double the score on the test compared to the exam, it was expected that it would return their names.

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

6 Problem 5

In the problem we are trying to prove the proposition $((p \to q) \to p) \to p$ using a sequent calculus approach. I have tried to create a proof in Isabelle using the examples that was provided as well as the sample lines for the formula p $\to p$ provided in the problem description. However, it seems as if there is an error that I do not know how to fix. Please refer to the Assignment2.txt file to see my possible solution.