02156 - Logical Systems and Logic Programming
Fall 2021

DTU - Technical University of Denmark

Date submitted: October 4, 2021

Assignment 1

Daniel F. Hauge (s201186)

Problem 1

Question 1.1

Truth table is constructed using semantics written in the assignment description.

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftrightarrow B$
T	T	F	T	T	T	T
T	F	F	F	T	F	F
T	X	F	X	T	X	X
F	T	T	F	T	T	F
F	F	T	F	F	T	T
F	X	T	F	X	T	¬ X
X	T	¬ X	X	T	T	X
X	F	¬ X	F	X	¬ X	¬ X
X	X	¬ X	X	X	T	T

The lack of semantics for X will make some cases just terminate with $\neg X$. An interesting observation from comparison is that some logical operations disregard or will work just fine without classical truth values. Like implication $X \rightarrow X$ will give True, as with implication it does not matter what value it is operating with.

Question 1.2

When p is T:

$$\neg T \wedge T = F \wedge T = \underline{\underline{F}}$$

When p is F:

$$\neg F \wedge F = T \wedge F = \underline{\underline{F}}$$

When p is X:

$$\neg X \wedge X = \underline{\underline{F}}$$

As there is no semantics for the negation of X, there is no better evaluation of ¬X. Hence we conclude the X case to be false, as the values is not equal and neither of them is T.

Problem 2

Question 2.1

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

$$\neg(((s \to p) \land ((\neg s \to t) \land (\neg p \to \neg t))) \to p)$$

$$((s \to p) \land ((\neg s \to t) \land (\neg p \to \neg t))), \neg p$$

$$s \to p, (\neg s \to t) \land (\neg p \to \neg t), \neg p$$

$$p, (\neg s \to t) \land (\neg p \to \neg t), \neg p \qquad\qquad \neg s, (\neg s \to t) \land (\neg p \to \neg t), \neg p$$

$$\times \qquad\qquad \neg s, (\neg s \to t), (\neg p \to \neg t), \neg p$$

$$\neg s, s, (\neg p \to \neg t), \neg p \qquad \neg s, t, (\neg p \to \neg t), \neg p$$

$$\times \qquad \neg s, t, p, \neg p \qquad \neg s, t, \neg t, \neg p$$

$$\times \qquad \times$$

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

Considering the logical equivalence, we observe that it can be used to argument for the following:

$$s \to p \equiv \neg p \to \neg s$$

and

$$\neg p \to \neg t \equiv t \to p$$

Therefor we can "swap" the parts marked with underline below to show equivalence with the formular:

$$((\underline{s \to p}) \land ((\neg s \to t) \land (\underline{\neg p \to \neg t}))) \to p$$

1 Problem 3

1.1 Question 3.1

See the prolog file for the predicate.

The predicate should fail when Elem is not occuring twice.

```
?- member2(1,[1,2,3,4]).
false.
```

The predicate should succed when 2 occurences of Elem are present in List.

```
?- member2(1,[1,2,3,4,1]).
true ;
false.
```

The predicate can be used as quiery to find elements occuring twice in List.

```
?- member2(X,[1,2,3,4,2,4]).
X = 2 ;
X = 4 ;
false.
```

The predicate will give multiple answers on backtracking.

```
?- member2(3, [3,3,3]).
true ;
true ;
true ;
false.
```

The predicate can give potentially infinite answers for lists which has Elem twice occuring.

```
?- member2(1,X).
X = [1, 1|_358] ;
X = [1, _1022, 1|_1036] .
```

The predicate can used in a query with variables only. Although is seems pointless to do so.

```
?- member2(X,A).
A = [X, X|_2830] ;
A = [X, _3564, X|_3578] .
```

1.2 Question 3.2

See the prolog file for the predicate. All tests from Question 3.1 has been done with *member2a*. All tests give identical results (apart from generic values).