CMPE 58S: Sp. Tp. Computer Aided Verification

Project: Interactive Propositional Logic Engine using Natural Deduction

Mehmet Utkan Gezer

2018700060

December 2, 2018

1 Introduction

Propositional logic, also known as propositional calculus, is the branch of logic dealing with propositions. Natural deduction is a way of handling propositions, whereby we apply a collection of rules to infer new conclusions from zero or more premises, all of which themselves are propositions.

For a more detailed introduction, we will be giving some definitions for propositions and natural deduction, along with its semantics.

1.1 Propositions

Propositions are declarative sentences with a truth value either **true** or **false**, and can be recursively defined as the composition of some other, smaller, propositions. Smallest of propositions are called *atomic* propositions, which are *indecomposable* and are given a unique symbol for the declarations they make.

Compositionals are propositions composed of other propositions, i.e. propositions that are not atomic. In propositional logic, there are 4 different ways of composing a *compositional*:

1. Negation ($\neg$): Negation of a proposition. E.g. "it does not rain" is the negation of "it rains". If the original proposition has the truth value **true**/**false**, then its negation has the truth value **false**/**true**, respectively.

2. Conjunction ($\wedge$): Conjunction of two propositions. E.g. "it does not rain and I am 25" is the conjunction of the propositions "it does not rain" and "I am 25". The compositional has the truth value **true** only if both of its constituents have the truth value **true**, and otherwise **false**.

3. Disjunction ($\vee$): Disjunction of two propositions. E.g. the non-exclusive sense of the statement "it is sunny or I am 5" is the conjunction of the propositions "it is sunny" and "I am 5". The compositional has the truth value **false** only if both of its constituents have the truth value **false**, otherwise (if either one or both of its constituents have the truth value **true**) the compositional is **true**.

4. Implication ($\rightarrow$): Implication of the second proposition, also known as the *consequent*, by the first proposition, also known as the *antecedent*. E.g. "if it is sunny then I am happy" is the implication of the proposition "I am happy" by the proposition "it is sunny". The compositional has the truth value **false** only if the antecedent and consequent are **true** and **false**, respectively.

To formally define the *language* of propositional logic's *well-formed formulas*, we give its Backus Naur form (BNF) as follows:

$$
\begin{aligned}
\langle\phi\rangle \quad &::= \quad \langle\text{atom}\rangle \;\mid\; \Big(\neg\langle\phi\rangle\Big) \;\mid\; \Big(\langle\phi\rangle\wedge\langle\phi\rangle\Big) \;\mid\; \Big(\langle\phi\rangle\vee\langle\phi\rangle\Big) \;\mid\; \Big(\langle\phi\rangle\rightarrow\langle\phi\rangle\Big) \\
\langle\text{atom}\rangle \quad &::= \quad p \;\mid\; q \;\mid\; \ldots \;\mid\; p_1 \;\mid\; p_2 \;\mid\; \ldots
\end{aligned}
$$

While this definition requires each use of a compositional operator to introduce a new pair of parenthesis for the newly generated proposition to be a well-formed formula, in practice we encounter propositons with many of those parenthesis omitted. In absence of parenthesis to enforce an explicit precedence of operator application, following precedence conventions are consulted:

- $\neg$ binds most tightly, followed by $\wedge$, $\vee$, and finally $\rightarrow$, in the given order.

- Implication ($\rightarrow$) is right-associative, i.e. the rightmost implication is to be evaluated the first.

Some books, including *Logic in Computer Science* by Huth and Ryan, regard $\wedge$ and $\vee$ operations as equal in precedence, in which case a proposition like $p \vee q \wedge r$ should either be regarded as unintelligible, or the same as $((p \vee q) \wedge r)$. We will be adhering to the above listed convention.

1.2 Natural deduction

Natural deduction is a way of reasoning about a given set of propositions and inferring new ones from them. Using a collection of *proof rules*, natural deduction allows us to come up with *conclusions* starting off by a set of *premises*. This relation between premises and conclusions are formalized by expressions called *sequents*, such as:

$$\phi_1, \phi_2, \ldots, \phi_n \vdash \psi.$$

Sequents with no premises are also valid sequents, and are called *theorems*.

Rules of natural deduction, also known as the previously mentioned proof rules, are at the heart of natural deduction, and this project. They allow us to establish a valid proof step with a proposition, using the previously validated list of propositions.

At any step, we may introduce an *assumption* to the proof, which, however, will introduce an assumption box along with it. The top line of the assumption box is drawn right above the step at which the assumption is introduced. We may close the assumption box after any step. A proof rule may only be applied to propositions, such that;

- Their validity must have been previously established, and
- Either they must be outside any assumption box, or their assumption box must not have been closed, yet.

We refer to such proof steps as the *accessible steps*.

A proof of a sequent is complete when the conclusion of the sequent is established using only the rules of natural deduction and starting off with only the premises of the sequent. It is important to note that an established proposition may not be the conclusion if it is found within an *assumption box*.

Here is a list of all natural deduction rules we have embedded into our program, given in sequents:

Name		**Rule**
Conjunction	Introduction	$\phi, \psi \vdash \phi \land \psi$
	Elimination #1	$\phi \land \psi \vdash \phi$
	Elimination #2	$\phi \land \psi \vdash \psi$
Disjunction	Introduction #1	$\phi \vdash \phi \lor \psi$
	Introduction #2	$\psi \vdash \phi \lor \psi$
	Elimination	$\phi \lor \psi, \boxed{\phi \cdots \chi}, \boxed{\psi \cdots \chi} \vdash \chi$
Implication	Introduction	$\boxed{\phi \cdots \psi} \vdash \phi \implies \psi$
	Elimination	$\phi, \phi \implies \psi \vdash \psi$
Negation	Introduction	$\boxed{\phi \cdots \bot} \vdash \neg\phi$
	Elimination	$\phi, \neg\phi \vdash \bot$
false	Elimination	$\bot \vdash \phi$
Double negation	Introduction	$\phi \vdash \neg\neg\phi$
	Elimination	$\neg\neg\phi \vdash \phi$
MT (Modus Tollens)		$\phi \to \psi, \neg\psi \vdash \neg\phi$
PBC (Proof by Contraditcion)		$\boxed{\neg\phi \cdots \bot} \vdash \phi$
LEM (Law of Excluded Middle)		$\vdash \phi \lor \neg\phi$
Copy		$\phi \vdash \phi$

The boxed premises such as $\boxed{\phi \cdots \psi}$ on implication introduction, is an assumption box in the proof that starts right before the proposition ϕ and ends right after the proposition ψ. To make step on the proof, one of the proof rules given above must be used. To use a rule, we need accessible propositions and/or assumption boxes that fits to the rule's sequent's propositions. Only if we are able to fulfill the *requirements* of the rule, we then may establish the validity of a new proposition according to the definition of the rule, and specifically the rule's sequent's conclusion. In natural deduction, this is the only way to make a proof step and approach to the ultimate conclusion.

Accessibility of an assumption box is defined similar to the accessibility of individual proof steps. This time, not a single step but the assumption box as a whole should be accessible as a single entity.

2 Exercises from book

Following exercises are from the book "Logic in Computer Science: Modelling and Reasoning about Systems" by Michael Huth and Mark Ryan.

2.1 Exercise 1.2.1.x

Validity of the following is to be proven.

$$p \to (q \lor r), q \to s, r \to s \vdash p \to s$$

Proof:

1.	$p \to (q \vee r)$	premise
2.	$q \to s$	premise
3.	$r \to s$	premise
4.	p	assumption
5.	$q \vee r$	$\to_e$ 1, 4
6.	q	assumption
7.	s	$\to_e$ 2, 6
8.	r	assumption
9.	s	$\to_e$ 3, 8
10.	s	$\vee_e$ 5, 6–7, 8–9
11.	$p \to s$	$\to_i$ 4–10

2.2 Exercise 1.2.2.d

Validity of the following is questioned. We claim it to be true, and prove it.

$$p \lor q, \neg q \lor r \vdash p \lor r$$

Proof:

1.	$p \lor q$	premise
2.	$\neg q \lor r$	premise
3.	$q \lor \neg q$	*LEM*
4.	q	assumption
5.	$\neg q$	assumption
6.	$\bot$	$\neg_e$ 4, 5
7.	r	$\bot_e$ 6
8.	r	assumption
9.	r	$\lor_e$ 2, 5–7, 8–8
10.	$p \lor r$	$\lor_{i_2}$ 9
11.	$\neg q$	assumption
12.	p	assumption
13.	q	assumption
14.	$\bot$	$\neg_e$ 13, 11
15.	p	$\bot_e$ 14
16.	p	$\lor_e$ 1, 12–12, 13–15
17.	$p \lor r$	$\lor_{i_1}$ 16
18.	$p \lor r$	$\lor_e$ 3, 4–10, 11–17

2.3 Exercise 1.2.3.q

Validity of the following is to be proven using *LEM*.

$$\vdash (p \to q) \lor (q \to r)$$

Proof:

1.	$q \lor \neg q$	*LEM*
2.	q	assumption
3.	p	assumption
4.	q	copy 2
5.	$p \to q$	$\to_i$ 3–4
6.	$(p \to q) \lor (q \to r)$	$\lor_{i_1}$ 5
7.	$\neg q$	assumption
8.	q	assumption
9.	$\bot$	$\neg_e$ 7, 8
10.	r	$\bot_e$ 9
11.	$q \to r$	$\to_i$ 8–10
12.	$(p \to q) \lor (q \to r)$	$\lor_{i_2}$ 11
13.	$(p \to q) \lor (q \to r)$	$\lor_e$ 1, 2–6, 7–12

2.4 Exercise 1.3.4.b

Parse tree of the following formula is to be drawn:

$$((p \to \neg q \lor p \land r) \to s) \lor \neg r$$

The parse tree can be seen in Figure 1.

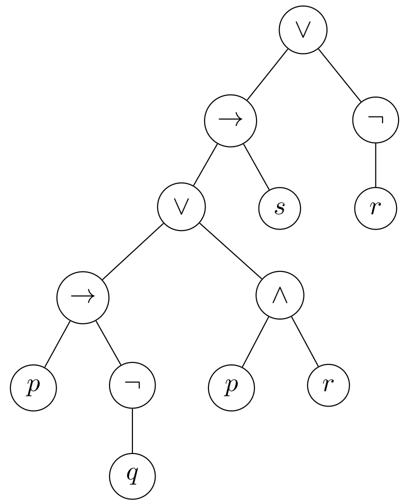

Figure 1: Parse tree for the formula.

2.5 Exercise 1.4.5

Validity and satisfiability of the formula of the parse tree in Figure 1.10 on page 44 is asked. You can see the redrawn parse tree on Figure 2.

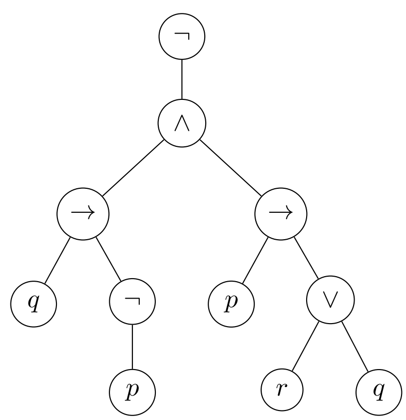

Figure 2: Redrawn parse tree of Figure 1.10 on page 44.

The formula of the parse tree is as follows:

$$(\neg\,(q \to \neg p \wedge p \to (r \vee q)))$$

Using the semantic equivalence rules (elimination of implication, De Morgan's laws and elimination of double negation), we can transform this formula as follows:

$$
\begin{aligned}
&\equiv (\neg\,(\neg q \vee \neg p \wedge \neg p \vee (r \vee q))) && \text{(elimination of implication)}\\
&\equiv (\neg\,(\neg q \vee \neg p \wedge \neg p \vee r \vee q)) && \text{(elimination of parenthesis)}\\
&\equiv (\neg\neg q \vee \neg p) \vee (\neg\neg p \vee r \vee q) && \text{(distribution of negation)}\\
&\equiv (\neg\neg q \wedge \neg\neg p \vee \neg\neg p \wedge \neg r \wedge \neg q) && \text{(distribution of negation)}\\
&\equiv (q \wedge p \vee p \wedge \neg r \wedge \neg q) && \text{(elimination of double negation)}
\end{aligned}
$$

This is the DNF (Disjunctive Normal Form) of the initial formula. We then build the following partial truth table for the formula:

p	q	r	$q \wedge p$	$p \wedge \neg r \wedge \neg q$	$(q \wedge p \vee p \wedge \neg r \wedge \neg q)$
T	T	T	T	F	T
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
F	T	T	F	F	F
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

With only those two lines of the truth table for the semantically equivalent formula $(q \wedge p \vee p \wedge \neg r \wedge \neg q)$ we can say that $(\neg\,(q \to \neg p \wedge p \to (r \vee q)))$ is satisfiable, but not valid:

- **It is satisfiable**, because there exists an evaluation, namely $(p, q, r) = (T, T, T)$ for which the formula evaluates as T.

- **It is not valid**, and the proof for it is by contradiction. When we assume it is valid, it should follow that for all evaluations of the atoms the formula should evaluate to T. However, for the evaluation $(p, q, r) = (F, T, T)$, the formula evaluates as F, which is a contradiction.

2.6 Exercise 1.5.4

Soundness or completeness is to be used to show that a sequent $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$ has a proof iff $\phi_1 \to \phi_2 \to \cdots \to \phi_n \to \psi$ is a tautology.

The statement is not even correct, and we can show that using soundness and completeness and a proof by contradiction.

Assume that the statement is true. The statement should be true for all n, so let $n = 2$. We then have a statement saying a sequent $\phi_1, \phi_2 \vdash \psi$ has a proof iff $\phi_1 \to \phi_2 \to \psi$ is a tautology.

By Remark 1.12, we can transform the original sequent $\phi_1, \phi_2 \vdash \psi$ as follows: $\vdash \phi_1 \to (\phi_2 \to \psi)$. Let us now transform them both using semantic equivalence rules:

$$\begin{aligned} \phi_1 \to (\phi_2 \to \psi) &\equiv \neg\phi_1 \lor (\neg\phi_2 \lor \psi) \\ &\equiv \neg\phi_1 \lor \neg\phi_2 \lor \psi \end{aligned} \qquad \begin{aligned} \phi_1 \to \phi_2 \to \psi &\equiv \neg\phi_1 \lor \phi_2 \to \psi \\ &\equiv \neg(\neg\phi_1 \lor \phi_2) \lor \psi \\ &\equiv (\neg\neg\phi_1 \land \neg\phi_2) \lor \psi \\ &\equiv (\phi_1 \land \neg\phi_2) \lor \psi \\ &\equiv (\phi_1 \lor \psi) \land (\neg\phi_2 \lor \psi) \end{aligned}$$

By the statement assumed to be true, if the original sequent, or equivalently its transformed form, or equivalently its semantic equivalent has a proof, then the second formula, or its semantic equivalent should be a tautology. Let us now prepare their truth tables:

ϕ_1	ϕ_2	ψ	$\neg\phi_1 \vee \neg\phi_2 \vee \psi$	$(\phi_1 \vee \psi) \wedge (\neg\phi_2 \vee \psi)$
T	T	T	T	T
T	T	F	F	F
T	F	T	T	T
T	F	F	T	T
F	T	T	T	T
F	T	F	T	F
F	F	T	T	T
F	F	F	T	F

On the truth table, we can clearly see that there are cases (highlighted in red) where sequent has a proof, yet the formula is not a tautology. This is a contradiction, meaning that our assumption was wrong. Statement that a sequent $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$ has a proof iff $\phi_1 \to \phi_2 \to \cdots \to \phi_n \to \psi$ is a tautology, is not true.

3 Classification of formulas

Following formulas are to be classified as valid, satisfiable, or not satisfiable, using the semantic method.

3.1 An implication

The formula is as follows:

$$\neg q \to q$$

We can transform it into the CNF form:

$$\equiv \neg\neg q \vee q$$

$$\equiv q \vee q$$

By Lemma 1.43, a disjunction of literals $L_1 \vee L_2 \vee \cdots \vee L_m$ is valid iff there are $1 \leq i, j \leq m$ such that L_i is $\neg L_j$. Since $q \vee q$ has no such pair of literals, this disjunction of literals is **not valid**.

Proposition 1.45 states that a formula is satisfiable iff its negation is not valid. Negation of our formula is $(\neg q \vee q)$, and in CNF form:

$$\equiv \neg q \wedge \neg q$$

The resulting formula is the conjunction of two identical disjunction of literals that consist of a single literal. Since there is no pair of literals of form $L_i = \neg L_j$ in $\neg q$, the negated formula is not valid by Lemma 1.43, and therefore our original formula is **satisfiable** by Proposition 1.45.

3.2 A conjunction

The formula is as follows:

$$(\neg q \rightarrow q \wedge q \rightarrow \neg q)$$

We can transform it into the CNF form:

$$\equiv (\neg\neg q \vee q \wedge \neg q \vee \neg q)$$
$$\equiv (q \vee q \wedge \neg q \vee \neg q)$$

For the formula to be valid, both the $q \vee q$ and $\neg q \vee \neg q$ needs to be valid. However, as proven in the previous exercise, former is not valid, therefore the formula is also **not valid**.

The negation of the formula is:

$$(\neg (q \vee q \wedge \neg q \vee \neg q)).$$

We can transform it into the CNF form:

$$\begin{aligned}
&\equiv (\neg q \lor q) \lor (\neg\neg q \lor \neg q) \\
&\equiv (\neg q \land \neg q \lor \neg\neg q \land \neg\neg q) \\
&\equiv (\neg q \land \neg q \lor q \land q) \\
&\equiv ((\neg q \land \neg q) \lor q \land (\neg q \land \neg q) \lor q) \\
&\equiv ((\neg q \lor q \land \neg q \lor q) \land (\neg q \lor q \land \neg q \lor q)) \\
&\equiv (\neg q \lor q \land \neg q \lor q) \land (\neg q \lor q \land \neg q \lor q)
\end{aligned}$$

The resulting formula is the conjunction of 4 identical disjunction of literals that consist of q and $\neg q$. Since $L_i = q$ and $L_j = \neg q$ is of the form $L_i = \neg L_j$ it is valid, and therefore all 4 of them are valid, and consequently the negated formula is also valid.

By Proposition 1.45, negated formula being valid means that the original formula is **not satisfiable**.