Prof. Dr. A. Poetzsch-Heffter
Peter Zeller, M. Sc.

TU Kaiserslautern
Fachbereich Informatik
AG Softwaretechnik

Sheet 4: Logik (SS 2017)

Abgabe: Freitag, 12. Mai, 15:30
Abgabekästen neben Raum 34-401.7 (bei AG Softwaretechnik)
Bitte geben Sie zu dritt ab.

Aufgabe 1 Vollständige Operatorenmengen

a) We want to prove that the set $F_{\{\vee,\wedge\}}$ only contains satisfiable formulas. Explain why it is difficult to show this with a straight forward induction.

b) Instead we have to prove a stronger statement using structural induction:
Find a variable assignment ψ, which satisfies all formulas in $F_{\{\vee,\wedge\}}$ and prove this property by induction.

c) Conclude from b) that $\{\vee, \wedge\}$ is not a functionally complete set of operators.

d) Let $\bar{\wedge}$ (**NAND**) be an operator, so that for all formulas A,B and all variable assignments ψ:

$$\mathcal{B}_\psi\left(A \mathbin{\bar{\wedge}} B\right) = 1 - \min\left(\mathcal{B}_\psi\left(A\right), \mathcal{B}_\psi\left(B\right)\right) = \mathcal{B}_\psi\left(\neg(A \wedge B)\right)$$

Use structural induction to prove that $\left\{\bar{\wedge}\right\}$ is a functionally complete set of operators. For the proof you can assume that $\{\neg, \vee\}$ is a functionally complete set of operators.

Aufgabe 2 Kompaktheitssatz

Let $\Sigma_0 \subseteq \Sigma_1 \subseteq \Sigma_2 \subseteq \dots$ be a chain of increasing sets of formulas.

a) Use the compactness theorem to show that the union $\Sigma = \bigcup_{i\in\mathbb{N}} \Sigma_i$ is satisfiable if and only if all Σ_i are satisfiable ($i \in \mathbb{N}$).

b) Is the statement from a) also valid, if $\Sigma_0, \Sigma_1, \dots$ is simply a chain of formulas (i.e. it does not necessarily hold that $\Sigma_i \subseteq \Sigma_{i+1}$ for all $i \in \mathbb{N}$)?

Aufgabe 3 Normalformen und Erfüllbarkeit

A *literal* L is a formula of the shape p or $\neg p$, where p is a propositional variable. A *co-clause* K is a conjunction of literals, $K \equiv L_1 \wedge \cdots \wedge L_k$. A formula F is in *disjunctive normal form (DNF)*, if it is a disjunction of co-clauses, i.e. $F \equiv K_1 \vee \cdots \vee K_m$.

a) Describe how one can construct an equivalent formula in DNF from a given formula $A \in F$.

b) Checking satisfiability is the most important problem of propositional logic. Describe an algorithm that takes a formula A in DNF and decides whether it is satisfiable. Your algorithm should have a complexity in $O\left(|A|^2\right)$.

c) The following algorithm takes a formula $A \in F$ and decides whether it is satisfiable. Why it the running time of this algorithm not polynomial, i.e. why is it not in $O\left(|A|^k\right)$ for some fixed k?

   1. Use the algorithm from part a) to transform the input to DNF.
   2. Use the algorithm from part b) to check satisfiability of the formula in DNF.

Aufgabe 4 Beweise im deduktivem System $\mathcal{F}_0$

For this exercise you should precisely consider the definition of proofs in deductive systems. In particular you should ensure that you know the difference between a proof in a deductive system, a proof with shortcuts in a deductive system, and a proof outside of the deductive system.

In this exercise you must only use the methods and assumptions given in the respective task. In particular, you should not argue using the semantics of propositional logic.

a) Check if the following proof in $\mathcal{F}_0$ is correct. Give an explanation for each correct step, i.e. state which axiom or rule has been used.

$$B_1 \equiv (\neg A \to \neg B) \to (B \to A)$$
$$B_2 \equiv \Big((\neg A \to \neg B) \to (B \to A)\Big) \to \Big(\big((\neg A \to \neg B) \to B\big) \to \big((\neg A \to \neg B) \to A\big)\Big)$$
$$B_3 \equiv \big((\neg A \to \neg B) \to B\big) \to \big((\neg A \to \neg B) \to A\big)$$

b) Show that there is a proof $(A \to B) \to \big((B \to C) \to (A \to C)\big)$ in the deductive system $\mathcal{F}_0$. You can use the deduction theorem.

c) Give a proof with shortcuts for $B \to \neg\neg B$ in the deductive system $\mathcal{F}_0$. In your proof you can assume that $\vdash_{\mathcal{F}_0} \neg\neg A \to A$ for all formulas A and use this fact as a shortcut.

d) Show that there is a proof for $(A \to B) \to \big((A \to \neg B) \to (A \to C)\big)$ in the deductive system $\mathcal{F}_0$. You can use the deduction theorem and you can assume that statements 1-5 from lemma 2.13 are already proven.

e) *(Additional exercise, no points)* Transform your proofs from the previous tasks to proofs in the *Isabelle* tool. You can find a tutorial (in German) with the material for this exercise.