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

TU Kaiserslautern
Fachbereich Informatik
AG Softwaretechnik

Sheet 2: Logik (SS 2017)

Abgabe: Freitag, 28. April, 15:30

Bitte geben Sie **zu dritt** ab (Ausnahmen können Sie mit Ihrem Tutor besprechen).

Die Abgabekästen stehen neben Raum 34-401.7 bei der AG Softwaretechnik. Sie können die Abgabe natürlich auch direkt in der Übung bei Ihrem Tutor abgeben. Die Abgabe kann handschriftlich oder als Ausdruck abgegeben werden, in deutscher oder englischer Sprache.

Wenden Sie sich bei Fragen zum Aufgabenblatt an Ihren Tutor oder besuchen Sie die Fragestunde (siehe Homepage).

Aufgabe 1 Strukturelle Induktion

Let $t(A)$ denote the *depth* of a propositional formula A and be defined as:

- If A is an atomic formula, then $t(A) = 0$.
- If $A \equiv (B * C)$ with binary operator $* \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$, then

$$t(A) = \max\{t(B), t(C)\} + 1.$$

- If $A \equiv \neg(B)$, then $t(A) = t(B) + 1$.

Furthermore let $|A|$ denote the length of formula A, meaning the number of characters in A: $|p_i| = 1$, $\left|(\neg A)\right| = 3 + |A|$ und $|(A * B)| = 3 + |A| + |B|$ für $* \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$.

Use induction over the structure of propositional formulas to prove that for every propositional formula A (with all parenthesis, no short notations) ...

a) $|A| \leq 5 \cdot op(A) + 1$

Here $op(A)$ is the number of operators $(\neg, \vee, \wedge, \rightarrow, \leftrightarrow)$ in A, as defined on the last sheet.

b) $|A| \leq 4 \cdot 2^{t(A)} - 3$

Aufgabe 2 Semantik von Formeln

a) Let ψ be a variable assignment with $\psi(p) = 1$ and $\psi(q) = \psi(r) = 0$. Use the semantics of propositional calculus to calculate

$$\mathcal{B}_\psi\left(\neg(p \wedge q) \rightarrow r\right)$$

step by step.

b) Prove or disprove the following statements. Use the semantics of propositional logic and truth tables.

1. $q \rightarrow \left(r \rightarrow (p \vee q)\right)$ is a tautology.
2. $q \rightarrow p \models p \rightarrow q$.
3. $\neg p \vee \neg q \models\!\!\!\!\!\!=\!\!\! \neg(p \wedge q)$.
4. $(p \rightarrow q) \wedge (p \rightarrow \neg q)$ is unsatisfiable.
5. $\left\{\neg p \vee \neg q, p \vee \neg r, \neg q \vee \neg r\right\}$ is satisfiable.
6. $(p \wedge r) \in Folg\left(\{p, \neg p\}\right)$

Aufgabe 3 Programmverifikation

You are given the following Java Code, which is supposed to calculate the minimum of 3 numbers.

```
1   /*@
2     @ ensures \result <= x
3     @      && \result <= y
4     @      && \result <= z;
5     @*/
6   public static int min3(int x, int y, int z) {
7     if (x < y && x < z) {
8       return x;
9     } else if (y < x && y < z) {
10      return y;
11    } else {
12      return z;
13    }
14  }
```

We now want to attempt to show that a correct result is returned in line 12. Therefore we can use a tool like OpenJML (the example can be tried online at http://www.rise4fun.com/OpenJMLESC/Anr). Tools like OpenJML use logics to check the correctness of programs.

Bei dieser Übung sollen Sie sich überlegen, welche logischen Schlüsse ein Verifikationstool wie OpenJML ziehen muss, um die Korrektheit des Programms zu beweisen. Um das Problem auf die Aussagenlogik zu reduzieren, verwenden wir die folgenden Variablen:

$p_{x<x}, p_{x<y}, p_{x<z},\ \ p_{y<x}, p_{y<y}, p_{y<z},\ \ p_{z<x}, p_{z<y}, p_{z<z}$

$p_{x\leq x}, p_{x\leq y}, p_{x\leq z},\ \ p_{y\leq x}, p_{y\leq y}, p_{y\leq z},\ \ p_{z\leq x}, p_{z\leq y}, p_{z\leq z}$

We can then express our mathematical knowledge in propositional formulas:

For example: $p_{x\leq y} \to \neg p_{y<x}$ (totality) and $p_{x<y} \wedge p_{y<z} \to p_{x<z}$ (transitivity).

Let M be the set of all such formulas, which describe mathematical properties about the variables (totality, transitivity, ...).

a) When reaching line 12 of the program, we know that the previous conditions were not satisfied. Therefore we know: $\Sigma = \left\{\neg(p_{x<y} \wedge p_{x<z}), \neg(p_{y<x} \wedge p_{y<z})\right\}$

   Zeigen Sie, dass daraus nicht folgt, dass das Ergebnis korrekt ist:

$$M \cup \Sigma \not\models p_{z\leq x} \wedge p_{z\leq y} \wedge p_{z\leq z}$$

   To do this, you can give an example with concrete numbers, which yields a wrong result.

b) We now correct the program and change line 9 to:

```
} else if (y < z) {
```

   Show that the result returned in line 12 is correct with this change. Which mathematical properties (formulas from M) can be used to show the correctness?