Prof. Dr. Andreas Podelski
Matthias Heizmann

3.11.2011
Submission: 8.11.2011
at the beginning of the lecture

Tutorials for Program Verification
Exercise sheet 2

Exercise 1: C Programs — some bonus points

Consider the following C program[1]. Find all errors of this program. You get 0.5 bonus points for each error.

```
int main() {
  char *p,*q;
  p = NULL;
  printf("%s",p);
  q = (char *)malloc(100);
  p = q;
  free(q);
  *p = 'x';
  free(p);
  p = (char *)malloc(100);
  p = (char *)malloc(100);
  q = p;
  strcat(p,q);
}
```

Exercise 2: Propositional Logic - Formalization — 5 points

Use ¬,→, ∧ and ∨ to express the following declarartive sentences in propositional logic; in each case state what your respective propositional atoms p, q etc. stand for:

(a) Robert was jealous of Yvonne, or he was not in a good mood.

(b) If the barometer falls, then either it will rain or it will snow.

(c) If a request occurs, then either it will eventually be acknowledged, or the requesting process won't ever be able to make progress.

(d) Cancer will not be cured unless its cause is determined and a new drug for cancer is found.

(e) If interest rates go up, share prices go down.

(f) If Smith has installed central heating, then he has sold his car or he has not paid his mortgage.

[1] This example was taken from Michael Schwarzbach's lecture notes on static analysis `http://lara.epfl.ch/dokuwiki/_media/sav08:schwartzbach.pdf`

(g) Today it will rain of shine, but not both.

(h) If Dick met Jane yesterday, they had a cup of coffee together, or they took a walk in the park.

(i) No shoes, no shirt, no service.[2]

(j) My sister wants a black and white cat.

Example: The sentence "If the sun shines today, then it won't shine tomorrow." can be expressed by the formula $p_{td} \to \neg p_{tm}$, where the propositional variable p_{td} stands for "sun shines today" and the propositional variable p_{tm} stands for "sun shines tomorrow".

Reminder: C.A.R.Hoare developed a logic which can be used to prove partial correctness of programs. Central object used in Hoare logic are triples $\{F\}$ P $\{G\}$, where P is a program, and F and G are assertions over the values of program variables. We call $\{F\}$ P $\{G\}$ a *Hoare triple.*
We call a Hoare triple *valid* if the following holds.

> If the assertion F holds before the execution of program P and program P terminates, then the assertion G holds afterwards.

Exercise 3: Hoare logic 4 + 1 + 1 bonus points

We consider in this exercise very simple Hoare triples, where

- the precondition *precond*(`X1`, ...,`Xn`) is a Boolean expression over the variables `X1`, ...,`Xn` and does not contain the variable `Y`.
- the program consists of the single line

  `Y:=` *expr*(`X1`, ...,`Xn`) ,

  where `Y` is a Boolean variable and *expr*(`X1`, ...,`Xn`) is a Boolean expression over the variables `X1`, ...,`Xn` that does not contain `Y`,
- and the postcondition *postcond*(`X1`, ...,`Xn`) is a Boolean expression over the variables `Y`,`X1`, ...,`Xn`.

(a) State a propositional logical formula

vc(`Y`,`X1`, ...,`Xn`)

that is valid if and only if a Hoare triple that has the following form is valid.

```
{ precond(X1,...,Xn) }
Y:= expr(X1,...,Xn)
{ postcond(Y,X1,...,Xn) }
```

[2] You find this sentence on signs in front of Californian beach restaurants. Think about the real meaning of the sentence before you write down your formula.

(b) Compute your propositional logical formula $vc(\texttt{U},\texttt{V},\texttt{Z})$ for the following concrete program.

```
{ U ↔ V }
Z:= U ∧ V
{ Z ↔ U }
```

Is your formula valid?

(c) Now we drop the restriction that $precond(\texttt{X1}, ..., \texttt{Xn})$ does not contain the variable `Y`. Find a Hoare triple which is not valid, but where your formula $vc(\texttt{U},\texttt{V},\texttt{Z})$ is valid.