apl. Prof. Dr. Thomas Noll

Kevin van der Pol, Hao Wu

Exercise 1 (Axiomatic Equivalence): (4 Points)

Establish the following axiomatic equivalence without using the equivalence of axiomatic and operational/denotational semantics:

$$\textbf{repeat } c \textbf{ until } b \quad \equiv \quad c; \textbf{while } \neg b \textbf{ do } c$$

where the axiomatic rule of the repeat-until-statement is the same as in Exercise 4.4.

Exercise 2 (Weakest precondition): (3 Points)

(a) The weakest precondition $wp(P, R)$ is the weakest assertion Q such that $\{Q\}P\{R\}$ holds (note that termination is not required). Give a definition for the weakest precondition of a given program P and assertion R by induction over the structure of program P.
(b) Use structural induction to prove the soundness of the rules you defined in (a).

Exercise 3 (Total correctness): (5 Points)

Consider the problem of defining an algorithm to compute the exponentiation x^y.

(a) Give the total correctness property that expresses that a program P calculates $r = X^Y$, where X and Y are the initial values of the variables x and y.

A naïve program to calculate the exponentiation could be as follows:
$r := 1;$
while $y > 0$ **do**
    $r := r * x;$
    $y := y - 1$

(b) What is the invariant of this repetition?

(c) Use the identity $x^{2*y} = (x * x)^y$ to give a faster program to calculate the exponentiation. Assume the existence of the boolean expression $even(n)$ to test if a number n is even. What is the invariant of this repetition?

(d) Prove the total correctness of the program you defined in (c). Use the naïve program if you could not find a program in question (c).