Final Exam in Programming Theory

Department of Computer Systems
Uppsala University
1995–10–25
Location: Parosh A. A., M. Kindahl
Plats: Polacksbacken
Time: 9 – 15
No books or calculator allowed

Directions:

1. Each proof step should be motivated formally (no credit for informal proofs)
2. **Simple** arithmetic rules and all the axioms and theorems in the appendix may be used in the proofs
3. Answer only one problem on each sheet of paper
4. Do not write on the back of the paper
5. Write your name on each sheet of paper

Good Luck!

Problem 1 (20 p)

Prove the following:

$$\left(\begin{array}{c} i < j+1 \\ \wedge \\ (\mathcal{N}k \ : \ i \leq k < j \ : \ b[k] \leq x) \ = \ 0 \end{array} \right) \ \Rightarrow \ (\mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x) \ \leq \ 1$$

Solution:

Lemma 1

$$\begin{array}{c} \left(\ i < j+1 \ \wedge \ (\mathcal{N}k \ : \ i \leq k < j \ : \ b[k] \leq x) \ = \ 0 \ \right) \\ \Rightarrow \\ \left(\begin{array}{c} (b[j] \leq x) \ \wedge \ (\mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x) \ \leq \ 1 \\ = \\ b[j] \leq x \end{array} \right) \end{array}$$

Proof:

Assume $\quad i < j+1$

Assume $\quad (\mathcal{N}k \ : \ i \leq k < j \ : \ b[k] \leq x) \ = \ 0$

$(b[j] \leq x) \ \wedge \ (\mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x) \ \leq \ 1$

$=$ { *and-simplification and definition of numerical quantification* }

$$i < j+1 \ \wedge \ b[j] \leq x \ \Rightarrow \ \left(\begin{array}{c} \mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x \\ = \\ (\mathcal{N}k \ : \ i \leq k < j+1-1 \ : \ b[k] \leq x)+1 \end{array} \right)$$
$$\wedge \ (b[j] \leq x) \ \wedge \ (\mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x) \ \leq \ 1$$

$=$ { *Assumption: "$i < j+1$" and and-simplification* }

$$b[j] \leq x \ \Rightarrow \ \left(\begin{array}{c} \mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x \\ = \\ (\mathcal{N}k \ : \ i \leq k < j+1-1 \ : \ b[k] \leq x)+1 \end{array} \right)$$
$$\wedge \ (b[j] \leq x) \ \wedge \ (\mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x) \ \leq \ 1$$

$=$ { *Elimination of* $\Rightarrow$ }

$$\left(\begin{array}{c} \mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x \\ = \\ (\mathcal{N}k \ : \ i \leq k < j+1-1 \ : \ b[k] \leq x)+1 \end{array} \right)$$
$$\wedge \ (b[j] \leq x) \ \wedge \ (\mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x) \ \leq \ 1$$

$=$ { *Substitution* }

$$\left(\begin{array}{c} \mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x \\ = \\ (\mathcal{N}k \ : \ i \leq k < j+1-1 \ : \ b[k] \leq x)+1 \end{array} \right)$$
$$\wedge \ (b[j] \leq x) \ \wedge \ (\mathcal{N}k \ : \ i \leq k < j+1-1 \ : \ b[k] \leq x)+1 \ \leq \ 1$$

$=$ { *Introduction of* $\Rightarrow$ }

$$b[j] \leq x \ \Rightarrow \ \left(\begin{array}{c} \mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x \\ = \\ (\mathcal{N}k \ : \ i \leq k < j+1-1 \ : \ b[k] \leq x)+1 \end{array} \right)$$
$$\wedge \ (b[j] \leq x) \ \wedge \ (\mathcal{N}k \ : \ i \leq k < j+1-1 \ : \ b[k] \leq x)+1 \ \leq \ 1$$

$=$ { *and-simplification and Assumption: "$i < j+1$"* }

$$i < j+1 \ \wedge \ b[j] \leq x \ \Rightarrow \ \left(\begin{array}{c} \mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x \\ = \\ (\mathcal{N}k \ : \ i \leq k < j+1-1 \ : \ b[k] \leq x)+1 \end{array} \right)$$
$$\wedge \ (b[j] \leq x) \ \wedge \ (\mathcal{N}k \ : \ i \leq k < j+1-1 \ : \ b[k] \leq x)+1 \ \leq \ 1$$

$=$ { *Definition of numerical quantification and and-simplification* }

$(b[j] \leq x) \ \wedge \ (\mathcal{N}k \ : \ i \leq k < j+1-1 \ : \ b[k] \leq x)+1 \ \leq \ 1$

$=$ { *Arithmetic* }

$(b[j] \leq x) \ \wedge \ (\mathcal{N}k \ : \ i \leq k < j \ : \ b[k] \leq x)+1 \ \leq \ 1$

$=$ { *Assumption: "$(\mathcal{N}k \ : \ i \leq k < j \ : \ b[k] \leq x) \ = \ 0$"* }

$(b[j] \leq x) \ \wedge \ 0+1 \ \leq \ 1$

$=$ { *Arithmetic and and-simplification* }

$b[j] \leq x \quad \Box$

Lemma 2

$$
\begin{array}{c}
i < j+1 \ \wedge \ (\mathcal{N}k \ : \ i \leq k < j \ : \ b[k] \leq x) \ = \ 0 \\
\Rightarrow \\
\left(\begin{array}{c}
\neg(b[j] \leq x) \ \wedge \ (\mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x) \ \leq \ 1 \\
= \\
\neg(b[j] \leq x)
\end{array} \right)
\end{array}
$$

Proof:

Assume $\quad i < j+1$

Assume $\quad (\mathcal{N}k \ : \ i \leq k < j \ : \ b[k] \leq x) \ = \ 0$

$\neg(b[j] \leq x) \ \wedge \ (\mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x) \ \leq \ 1$

$=$ { *and-simplification and definition of numerical quantification* }

$$i < j+1 \ \wedge \ \neg(b[j] \leq x) \ \Rightarrow \ \left(\begin{array}{c} \mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x \\ = \\ (\mathcal{N}k \ : \ i \leq k < j+1-1 \ : \ b[k] \leq x) \end{array} \right)$$
$$\wedge \ \neg(b[j] \leq x) \ \wedge \ (\mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x) \ \leq \ 1$$

$=$ { *Assumption: "$i < j+1$" and and-simplification* }

$$\neg(b[j] \leq x) \ \Rightarrow \ \left(\begin{array}{c} \mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x \\ = \\ (\mathcal{N}k \ : \ i \leq k < j+1-1 \ : \ b[k] \leq x) \end{array} \right)$$
$$\wedge \ \neg(b[j] \leq x) \ \wedge \ (\mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x) \ \leq \ 1$$

$=$ { *Elimination of* $\Rightarrow$ }

$$\left(\begin{array}{c} \mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x \\ = \\ (\mathcal{N}k \ : \ i \leq k < j+1-1 \ : \ b[k] \leq x) \end{array} \right)$$
$$\wedge \ \neg(b[j] \leq x) \ \wedge \ (\mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x) \ \leq \ 1$$

$=$ { *Substitution* }

$$\left(\begin{array}{c} \mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x \\ = \\ (\mathcal{N}k \ : \ i \leq k < j+1-1 \ : \ b[k] \leq x) \end{array} \right)$$
$$\wedge \ \neg(b[j] \leq x) \ \wedge \ (\mathcal{N}k \ : \ i \leq k < j+1-1 \ : \ b[k] \leq x) \ \leq \ 1$$

$=$ { *Introduction of* $\Rightarrow$ }

$$\neg(b[j] \leq x) \ \Rightarrow \ \left(\begin{array}{c} \mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x \\ = \\ (\mathcal{N}k \ : \ i \leq k < j+1-1 \ : \ b[k] \leq x) \end{array} \right)$$
$$\wedge \ \neg(b[j] \leq x) \ \wedge \ (\mathcal{N}k \ : \ i \leq k < j+1-1 \ : \ b[k] \leq x) \ \leq \ 1$$

$=$ { *and-simplification and Assumption: "$i < j+1$"* }

$$i < j+1 \ \wedge \ \neg(b[j] \leq x) \ \Rightarrow \ \left(\begin{array}{c} \mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x \\ = \\ (\mathcal{N}k \ : \ i \leq k < j+1-1 \ : \ b[k] \leq x) \end{array} \right)$$
$$\wedge \ \neg(b[j] \leq x) \ \wedge \ (\mathcal{N}k \ : \ i \leq k < j+1-1 \ : \ b[k] \leq x) \ \leq \ 1$$

$=$ { *Definition of numerical quantification and and-simplification* }

$\neg(b[j] \leq x) \ \wedge \ (\mathcal{N}k \ : \ i \leq k < j+1-1 \ : \ b[k] \leq x) \ \leq \ 1$

$=$ { *Arithmetic* }

$\neg(b[j] \leq x) \ \wedge \ (\mathcal{N}k \ : \ i \leq k < j \ : \ b[k] \leq x) \ \leq \ 1$

$=$ { *Assumption: "$(\mathcal{N}k \ : \ i \leq k < j \ : \ b[k] \leq x) \ = \ 0$"* }

$\neg(b[j] \leq x) \ \wedge \ 0 \ \leq \ 1$

$=$ { *Arithmetic and and-simplification* }

$\neg(b[j] \leq x) \quad \Box$

Main proof:

Assume $i < j+1$
Assume $(\mathcal{N}k \ : \ i \leq k < j \ : \ b[k] \leq x) \ = \ 0$

$(\mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x) \ \leq \ 1$
$=$ { and-simplification and Excluded Middle }
$((b[j] \leq x) \ \vee \ \neg(b[j] \leq x)) \ \wedge \ (\mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x) \ \leq \ 1$
$=$ { Distributivity }
$(b[j] \leq x) \ \wedge \ (\mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x) \ \leq \ 1$
$\vee$
$\neg(b[j] \leq x) \ \wedge \ (\mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x) \ \leq \ 1$
$=$ { Conditional Substitution: Lemma 1 and
Assumptions: "$i < j+1$" and "$(\mathcal{N}k \ : \ i \leq k < j \ : \ b[k] \leq x) \ = \ 0$" }
$(b[j] \leq x)$
$\vee$
$\neg(b[j] \leq x) \ \wedge \ (\mathcal{N}k \ : \ i \leq k < j+1 \ : \ b[k] \leq x) \ \leq \ 1$
$=$ { Conditional Substitution: Lemma 2 and
Assumptions: "$i < j+1$" and "$(\mathcal{N}k \ : \ i \leq k < j \ : \ b[k] \leq x) \ = \ 0$" }
$(b[j] \leq x) \ \vee \ \neg(b[j] \leq x)$
$=$ { Excluded Middle }
$T \quad \Box$

Problem 2 (12 p)

Two programs P_1 and P_2 are said to be *equivalent* iff $wp\,(P_1,\ R) \ = \ wp\,(P_2,\ R)$, for any predicate R. Show that the programs:

$$
\begin{array}{lll}
IF_1 \ : & \textbf{if} & B_1 \ \rightarrow \ S_1 \\
 & \Box & B_2 \ \rightarrow \ S_2 \\
 & \textbf{fi} &
\end{array}
$$

and

$$
\begin{array}{lll}
IF_2 \ : & \textbf{if} & B_1 \ \rightarrow \ S_1 \\
 & \Box & B_2 \ \rightarrow \ S_3 \\
 & \textbf{fi} &
\end{array}
$$

are equivalent, where S_3 is defined as:

$$\begin{array}{lll} S_3 : & \textbf{if} & B_1 \rightarrow S_1 \\ & \Box & B_2 \rightarrow S_2 \\ & \textbf{fi} & \end{array}$$

Solution:

Lemma 3

$$(p \Rightarrow q \wedge p \Rightarrow r) = (p \Rightarrow (q \wedge r))$$

Proof:

$$\begin{array}{ll} & p \Rightarrow q \wedge p \Rightarrow r \\ = & \{ \textit{ Implication } \} \\ & (\neg p \vee q) \wedge (\neg p \vee r) \\ = & \{ \textit{ Distributivity } \} \\ & \neg p \vee (q \wedge r) \\ = & \{ \textit{ Implication } \} \\ & p \Rightarrow (q \wedge r) \quad \Box \end{array}$$

Main proof:

$wp\,(\mathrm{IF}_1,\ R)$

$=$ { Definition of the alternative command }

$(B_1 \ \vee\ B_2) \ \wedge\ (B_1 \ \Rightarrow\ wp\,(S_1,\ R)) \ \wedge\ (B_2 \ \Rightarrow\ wp\,(S_2,\ R))$

$=$ { and-simplification }

$(B_1 \ \vee\ B_2) \ \wedge\ (B_1 \ \Rightarrow\ wp\,(S_1,\ R)) \ \wedge\ ((B_2 \ \wedge\ B_2) \ \Rightarrow\ wp\,(S_2,\ R))$

$=$ { Shunting }

$(B_1 \ \vee\ B_2) \ \wedge\ (B_1 \ \Rightarrow\ wp\,(S_1,\ R)) \ \wedge\ (B_2 \ \Rightarrow\ (B_2 \ \Rightarrow\ wp\,(S_2,\ R)))$

$=$ { ImpIntro $(p \ \wedge\ (q \ \Rightarrow\ p) \ =\ p)$ }

$(B_1 \ \vee\ B_2) \ \wedge\ (B_1 \ \Rightarrow\ wp\,(S_1,\ R)) \ \wedge$
$(B_2 \ \Rightarrow\ ((B_1 \ \vee\ B_2) \ \wedge\ (B_1 \ \Rightarrow\ wp\,(S_1,\ R))))$
$\wedge\,(B_2 \ \Rightarrow\ (B_2 \ \Rightarrow\ wp\,(S_2,\ R)))$

$=$ { Lemma 3 }

$(B_1 \ \vee\ B_2) \ \wedge\ (B_1 \ \Rightarrow\ wp\,(S_1,\ R)) \ \wedge$
$(B_2 \ \Rightarrow\ ((B_1 \ \vee\ B_2) \ \wedge\ (B_1 \ \Rightarrow\ wp\,(S_1,\ R)) \ \wedge\ (B_2 \ \Rightarrow\ wp\,(S_2,\ R))))$

$=$ { Definition of the alternative command }

$(B_1 \ \vee\ B_2) \ \wedge\ (B_1 \ \Rightarrow\ wp\,(S_1,\ R)) \ \wedge\ (B_2 \ \Rightarrow\ wp\,(S_3,\ R))$

$=$ { Definition of the alternative command }

$wp\,(\mathrm{IF}_2,\ R)$ $\quad\square$

Problem 3 (25 p)

Consider the following program:

$\{Q:\ n \geq 0\}$
$found :=\ F;\ i :=\ 0;$
do $\quad \neg found \ \wedge\ (i \neq n+1) \ \wedge\ (b[i] \neq x) \ \rightarrow\ i :=\ i+1$
$\quad\square\ \neg found \ \wedge\ (i \neq n+1) \ \wedge\ (b[i] \ =\ x) \ \rightarrow\ found :=\ T$
od

The program is used to find the smallest index in the interval 0 to n in an array b such that $b[i] = x$. If x does not occur in that interval then the flag *found* remains equal to F.

Define the predicates P_1, P_2, and P_3 as

$P_1 \ :\ \neg found \implies \forall k \ :\ 0 \leq k < i \ :\ b[k] \neq x$
$P_2 \ :\ found \implies (0 \leq i \leq n) \ \wedge\ (\forall k \ :\ 0 \leq k < i \ :\ b[k] \neq x) \ \wedge\ (b[i] = x)$
$P_3 \ :\ 0 \leq i \leq n+1$

- Show that $P_1 \wedge P_2 \wedge P_3$ is an invariant during the execution of the loop. (20 p)
- Find an appropriate bound function. (5p)

Solution:
To prove that $P = P_1 \wedge P_2 \wedge P_3$ is an invariant, you have to prove

1. $P \wedge \neg found \wedge (i \neq n+1) \wedge (b[i] \neq x) \Rightarrow wp(\text{“}i := i+1\text{”}, P)$
2. $P \wedge \neg found \wedge (i \neq n+1) \wedge (b[i] = x) \Rightarrow wp(\text{“}found := T\text{”}, P)$

□

□

Problem 4 (20p) Betrakta problemet att beräkna *punktprodukten* $p = a \bullet b$ av två vektorer $a[0:n]$ och $b[0:n]$, där

$$a \bullet b = \sum_{i=0}^{n} a[i] * b[i].$$

Hitta slutvillkoret R, invarianten P, gränsfunktionen t, vakten B samt kommandona S_0 och S_1 så att programmet uppfyller specifikationen. Beskriv noga ur du kommer fram till lösningen.

$\{Q\colon\ n > 1\}$
S_0;
$\{\text{inv } P\colon\ ?\}$
$\{\text{bound } t\colon\ ?\}$
do $B \rightarrow S_1$ **od**
$\{R\colon\ ?\}$

Ledtrådar:

- Loopen har endast ett fall.

Problem 4 (20p) Consider the problem of computing the *dot product* $p = a \bullet b$ of two vectors $a[0:n]$ and $b[0:n]$, where

$$a \bullet b = \sum_{i=0}^{n} a[i] * b[i].$$

Develop the post-condition R, the invariant P, the bound function t, the guard B, and the statements S_0 and S_1 such that the program satisfies the specification. Motivate carefully how you find the solution.

$\{Q\colon\ n > 1\}$
S_0;
$\{\text{inv } P\colon\ ?\}$
$\{\text{bound } t\colon\ ?\}$
do $B \to S_1$ **od**
$\{R\colon\ ?\}$

Hints:

- The loop has only one guarded command.

□