Tentamen i Programmeringsteori

Institutionen för datorteknik
Uppsala universitet
1996–08–14
Lärare: Parosh A. A., M. Kindahl
Plats: Polacksbacken
Skrivtid: 9 – 15
Hjälpmedel: Inga

Anvisningar:

1. Varje bevissteg ska motiveras formellt (informella bevis ger 0 poäng)
2. **Enkla** aritmetiska regler samt alla axiom och teorem i bilagan kan användas i bevisen
3. Lös endast en uppgift per blad
4. Skriv inte på baksidan
5. Skriv ditt namn på varje blad

Lycka till!

Final Exam in Programming Theory

Department of Computer Systems
Uppsala University
1996–08–14
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!

Uppgift 1 (20 p)
Visa följande:

$$\left(\begin{array}{c} m_1 \leq m_2 \leq m_3 \leq m_4 \\ \wedge \\ a[m_1 : m_3] \leq x \\ \wedge \\ a[m_2 : m_4] \leq x \end{array}\right) \implies a[m_1 : m_4] \leq x$$

Problem 1 (20 p)
Prove the following:

$$\left(\begin{array}{c} m_1 \leq m_2 \leq m_3 \leq m_4 \\ \wedge \\ a[m_1 : m_3] \leq x \\ \wedge \\ a[m_2 : m_4] \leq x \end{array}\right) \implies a[m_1 : m_4] \leq x$$

Uppgift 2 (15 p)
Två program P_1 och P_2 sägs vara *ekvivalenta* omm $wp(P_1, R) = wp(P_2, R)$, för alla predikat R. Anta att $B_2 \Rightarrow B_1$. Visa att programmen S och S' är ekvivalenta.

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

och

$$\begin{array}{lllll} S': & \textbf{if} & B_1 & \rightarrow & S_1 \\ & \Box & B_2 & \rightarrow & S \\ & \textbf{fi} & & & \end{array}$$

Problem 2 (15 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. Suppose that $B_2 \Rightarrow B_1$. Show that the programs S and S' are equivalent.

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

och

$$\begin{array}{lllll} S': & \textbf{if} & B_1 & \rightarrow & S_1 \\ & \Box & B_2 & \rightarrow & S \\ & \textbf{fi} & & & \end{array}$$

Uppgift 3 (20 p)
Anta att x och y är naturliga tal. Genom att använda teoremet för villkorliga satser, visa att följande program är korrekt.

$$\begin{array}{l} \{(y \;=\; x-2) \;\vee\; (y \;=\; x+2)\} \\ \quad \textbf{if} \;\; y \leq x \;\; \rightarrow \;\; x,\, y := \; y+3,\; x-1 \\ \quad \Box \;\; x \leq y \;\; \rightarrow \;\; y := 4;\; x := 6 \\ \quad \textbf{fi} \\ \{(y \;=\; x-2) \;\vee\; (y \;=\; x+2)\} \end{array}$$

Ledtråd: Använd följande regler:

- $((0 < c) \;\wedge\; (x - c \leq x)) = F$.
- $((0 < c) \;\wedge\; (x + c \geq x)) = F$.

Problem 3 (20 p)
Suppose that x and y are natural numbers. Using the alternative command theorem, show that the following program is correct.

$$\begin{array}{l} \{(y \;=\; x-2) \;\vee\; (y \;=\; x+2)\} \\ \quad \textbf{if} \;\; y \leq x \;\; \rightarrow \;\; x,\, y := \; y+3,\; x-1 \\ \quad \Box \;\; x \leq y \;\; \rightarrow \;\; y := 4;\; x := 6 \\ \quad \textbf{fi} \\ \{(y \;=\; x-2) \;\vee\; (y \;=\; x+2)\} \end{array}$$

Hint: Use the following rules.

- $((0 < c) \;\wedge\; (x - c \leq x)) = F$.
- $((0 < c) \;\wedge\; (x + c \geq x)) = F$.

Uppgift 4 (25p)
Följande program är en mycket enkel variant för att lösa problemet om ömsesidig uteslutning. Variablerna $proc_1$ respektive $proc_2$ säger att process 1 respektive process 2 är inne i den kritiska regionen. För att kontrollera tillträde till den kritiska regionen använder vi en flaggvariabel $flag$. Genom att använda teoremet för iterativa satser, visa att invarianten är korrekt (att den verkligen är invariant).

$$
\begin{array}{l}
flag, proc_1, proc_2 := F, F, F; \\
\{\text{inv } Q: (proc_1 \vee proc_2 \Rightarrow flag) \wedge \neg(proc_1 \wedge proc_2)\} \\
\begin{array}{lll}
\textbf{do} & \neg flag & \rightarrow flag := T; proc_1 := T \\
{[]} & \neg flag & \rightarrow flag := T; proc_2 := T \\
{[]} & flag \wedge proc_1 & \rightarrow \textbf{skip} \\
{[]} & flag \wedge proc_2 & \rightarrow \textbf{skip} \\
{[]} & flag \wedge proc_1 & \rightarrow proc_1 := F; flag := F \\
{[]} & flag \wedge proc_2 & \rightarrow proc_2 := F; flag := F \\
\textbf{od} & &
\end{array}
\end{array}
$$

Problem 4 (25p)
Following is a program to solve the mutual exclusion problem. The variables $proc_1$ and $proc_2$ respectively denotes that process 1 or process 2 respectively is in the critical region. To control access to the critical region we use a flag variable $flag$. Using the iterative command theorem, prove that the invariant is correct (i.e. that it really is invariant).

$$
\begin{array}{l}
flag, proc_1, proc_2 := F, F, F; \\
\{\text{inv } Q: (proc_1 \vee proc_2 \Rightarrow flag) \wedge \neg(proc_1 \wedge proc_2)\} \\
\begin{array}{lll}
\textbf{do} & \neg flag & \rightarrow flag := T; proc_1 := T \\
{[]} & \neg flag & \rightarrow flag := T; proc_2 := T \\
{[]} & flag \wedge proc_1 & \rightarrow \textbf{skip} \\
{[]} & flag \wedge proc_2 & \rightarrow \textbf{skip} \\
{[]} & flag \wedge proc_1 & \rightarrow proc_1 := F; flag := F \\
{[]} & flag \wedge proc_2 & \rightarrow proc_2 := F; flag := F \\
\textbf{od} & &
\end{array}
\end{array}
$$

Uppgift 5 (20p)
Antag att vi representerar polynom som vektorer. Exempelvis så representerar vi polynomet $p(x) = -2x^4 + 3x^3 + x^2 - 2x + 5$ som vektorn $p[0 : 4] = (-2, 3, 1, -2, 5)$. Betrakta nu problemet att *evaluera polynomet*, d.v.s. beräkna polynomet $p[0 : n]$ givet ett värde x. Detta definieras som

$$eval(p, x) = \sum_{i=0}^{n} p[i] * x^{n-i}.$$

Detta kan man göra effektivt med hjälp av *Horners regel* som beräknar polynomet $p(x)$ genom att alternera addition och multiplikation. Exempelvis så beräknas polynomet p ovan i ordningen

$$x(x(x(x(-2) + 3) + 1) - 2) + 5.$$

Utveckla ett program som använder Horners regel fr att evaluera polynom representerade som vektorer. 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.

$\{Q\text{: } n \geq 0\}$
S_0;
$\{\text{inv } P\text{: ?}\}$
$\{\text{bound } t\text{: ?}\}$
do $B \rightarrow S_1$ **od**
$\{R\text{: ?}\}$

Ledtrådar:

- Lagra slutresultatet i exempelvis variabeln r.
- Loopen har endast ett fall.

Problem 5 (20p)
Suppose we represents polynoms as vectors of coefficients, e.g. we represents the polynom $p(x) = -2x^4 + 3x^3 + x^2 - 2x + 5$ as the vector $p[0 : 4] = (-2, 3, 1, -2, 5)$. Consider the problem of *evaluating the polynom*, i.e. compute the polynom $p[0 : n]$ given a value x. This is defined as

$$eval(p, x) = \sum_{i=0}^{n} p[i] * x^{n-i}.$$

This can be done efficiently by using *Horner's rule* which computes the polynom $p(x)$ by alternating addition and multiplication. As an example; the polynom p above is computed in the order

$$x(x(x(x(-2) + 3) + 1) - 2) + 5.$$

Develop a program that uses Horner's rule to evaluate polynoms represented as vectors. 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.

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

Hints:

- Store the result in a variable, e.g. r.
- The loop has only one guarded command.