6.0/4.0 VU Formale Methoden der Informatik

185.291 WS 2011 4 May 2012

Kennzahl (study id)	Matrikelnummer (student id)	Familienname (family name)	Vorname (first name)	Gruppe (version)
				A

1.) We want to prove the *NP-hardness* of **MULTIPROCESSOR SCHEDULING**. Your task is to give a polynomial time reduction from **PARTITION** (which is *NP-complete*) to **MULTIPROCESSOR SCHEDULING**. Note that you have to provide only the reduction and not the proof of correctness of the reduction. The definiton of these two problems is given below:

PARTITION:
Instance: A finite set of n positive integers $S = \{a_1, a_2, \ldots, a_n\}$.
Question: Can the set S be partitioned into two subsets S_1, S_2 such that the sum of the numbers in S_1 equals the sum of the numbers in S_2?

MULTIPROCESSOR SCHEDULING:
Instance: A set J of k jobs where job j_i has length l_i, and m processors.
Question: Can we schedule all jobs in J on m processors such that

a) on each processor the next job in the sequence is started immediately after the preceding job is finished and

b) the total time to execute all jobs on each processor takes the minimum possible time $T_{min} = (\sum_{i=1}^{k} l_i)/m$.

(15 points)

2.) (a) Let φ^E be the following equality logic formula

$$(x_1 = x_2 \wedge x_2 = x_3 \wedge x_3 \neq x_4) \vee x_4 = x_1 \ .$$

Apply the *Sparse Method* and derive a purely propositional formula φ^P, which is equisatisfiable to φ^E. Be verbose and tell us which steps you apply, the result obtained in the each step and the connection between the different partial results. **(5 points)**

(b) We consider a simplified variant of Tseitin's reduction. Let φ be a propositional formula, let $\Sigma(\varphi)$ be the set of all subformulas of φ, and let ℓ_φ be the label for φ. Prove that

$$(\textstyle\bigwedge_{\psi \in \Sigma(\varphi)} (\ell_\psi \equiv \psi)) \rightarrow \ell_\varphi \text{ is valid if and only if } \varphi \text{ is valid.}$$

(10 points)

3.) Extend the toy language presented in the course by assert-statements of the form assert e . When the condition e evaluates to true, the program continues, otherwise the program aborts.

Specify the syntax and semantics of the extended language. Determine the weakest precondition, the weakest liberal precondition, the strongest postcondition, and Hoare rules (partial and total correctness) for assert-statements. Show that they are correct.

Treat the assert-statement as a first-class citizen, i.e., do not refer to other program statements in the final result. However, you may use other statements as intermediate steps when deriving the rules. **(15 points)**

Remember the following properties of wp, wlp, sp, and the Hoare calculus.

$\mathrm{wp}(\mathsf{skip}, G) = G$
$\mathrm{wp}(\mathsf{abort}, G) = \text{false}$
$\mathrm{wp}(v \leftarrow e, G) = G[v/e]$
$\mathrm{wp}(p; q, G) = \mathrm{wp}(p, \mathrm{wp}(q, G))$
$\mathrm{wp}(\mathsf{if}\ e\ \mathsf{then}\ p\ \mathsf{else}\ q\ \mathsf{fi}, G)$
$\quad = (e \wedge \mathrm{wp}(p, G)) \vee (\neg e \wedge \mathrm{wp}(q, G))$

wlp behaves like wp except:
$\mathrm{wlp}(\mathsf{abort}, G) = \text{true}$

$\mathrm{sp}(\mathsf{skip}, F) = F$
$\mathrm{sp}(\mathsf{abort}, F) = \text{false}$
$\mathrm{sp}(v \leftarrow e, F) = \exists v' \, (F[v/v'] \wedge v = e[v/v'])$
$\mathrm{sp}(p; q, F) = \mathrm{sp}(q, \mathrm{sp}(p, F))$
$\mathrm{sp}(\mathsf{if}\ e\ \mathsf{then}\ p\ \mathsf{else}\ q\ \mathsf{fi}, F)$
$\quad = \mathrm{sp}(p, F \wedge e) \vee \mathrm{sp}(q, F \wedge \neg e)$

$\{\, F \,\}\ \mathsf{skip}\ \{\, F \,\}$

$\{\, F \,\}\ \mathsf{abort}\ \{\, G \,\}$ partial correctness

$\{\, \text{false} \,\}\ \mathsf{abort}\ \{\, G \,\}$ total correctness

$\{\, F[v/e] \,\}\ v \leftarrow e\ \{\, F \,\}$

$$\frac{\{\, F \,\}\, p\, \{\, G \,\} \quad \{\, G \,\}\, q\, \{\, H \,\}}{\{\, F \,\}\, p; q\, \{\, H \,\}}$$

$$\frac{\{\, F \wedge e \,\}\, p\, \{\, G \,\} \quad \{\, F \wedge \neg e \,\}\, q\, \{\, G \,\}}{\{\, F \,\}\ \mathsf{if}\ e\ \mathsf{then}\ p\ \mathsf{else}\ q\ \mathsf{fi}\ \{\, G \,\}}$$

$$\frac{F \Rightarrow F' \quad \{\, F' \,\}\, p\, \{\, G' \,\} \quad G' \Rightarrow G}{\{\, F \,\}\, p\, \{\, G \,\}}$$

4.) Bisimulation

Let $M_1 = (S_1, I_1, R_1, L_1)$ and $M_2 = (S_2, I_2, R_2, L_2)$ be two Kripke structures.
Remember, a relation $H' \subseteq S_1 \times S_2$ is a bisimulation relation if for each $(s, s') \in H'$ holds:

- $L_1(s) = L_2(s')$,
- for each $(s, t) \in R_1$ there is a $(s', t') \in R_2$ such that $(t, t') \in H'$, and
- for each $(s', t') \in R_2$ there is a $(s, t) \in R_1$ such that $(t, t') \in H'$.

Let $M_1 = (S_1, \{s_0\}, R_1, L_1)$ and $M_2 = (S_2, \{t_0\}, R_2, L_2)$ be two (finite) Kripke structures, where s_0 is the single initial state of M_1 and t_0 is the single initial state of M_2. Consider the following sequence of sets $P_i \subseteq (S_1 \times S_2)$ and $N_i \subseteq (S_1 \times S_2)$:

$$\begin{aligned}
N_0 &= \emptyset \\
P_0 &= \{(s_0, t_0)\} \\
N_{i+1} &= N_i \cup \\
&\quad \{(s, t) \in P_i \mid L_1(s) \neq L_2(t)\} \cup \\
&\quad \{(s, t) \in P_i \mid \exists (s, s') \in R_1 . \forall (t, t') \in R_2 . (s', t') \in N_i\} \cup \\
&\quad \{(s, t) \in P_i \mid \exists (t, t') \in R_2 . \forall (s, s') \in R_1 . (s', t') \in N_i\} \\
P_{i+1} &= (P_i \setminus N_{i+1}) \cup \\
&\quad \mathbf{choose}(\{(s', t') \notin (P_i \cup N_i) \mid \exists (s, t) \in (P_i \setminus N_{i+1}) . (s, s') \in R_1 \wedge (t, t') \in R_2\}),
\end{aligned}$$

where $\mathbf{choose}(S)$ randomly returns an element of the set S, if S is not empty, and the empty set, otherwise.

(a) Show that, for all $i \in \mathbb{N}$, P_i and N_i are disjoint, i.e., $P_i \cap N_i = \emptyset$.

(2 points)

(b) Show that $N_i \cup P_i \subseteq N_{i+1} \cup P_{i+1}$ holds for all $i \geq 0$. *Hint:* You might use the fact, that $[(s, t) \in P_j \wedge (s, t) \notin N_{j+1}] \Rightarrow (s, t) \in P_{j+1}$ holds for all $j \in \mathbb{N}$.

(4 points)

(c) Show that there is a $n_0 \in \mathbb{N}$ such that $N_{n_0} = N_n$ and $P_{n_0} = P_n$ for all $n > n_0$.

(4 points)

(d) Show that the P_{n_0} from the preceding question is a bisimulation relation.

(5 points)