Software Verification: Contracts, Trusted Components and Patterns

ETH Zürich

Date: 15 December 2008

Surname, first name: ...........................................................................................................

Student number: ...............................................................................................................  

I confirm with my signature, that I was able to take this exam under regular circumstances and that I have read and understood the directions below.

Signature: ...........................................................................................................................

Directions:

• Exam duration: 1 hour 45 minutes.

• Except for a dictionary you are not allowed to use any supplementary material.

• All solutions can be written directly on the exam sheets. If you need more space for your solution ask the supervisors for a sheet of official paper. You are not allowed to use other paper. Please write your student number on each additional sheet.

• Only one solution can be handed in per question. Invalid solutions need to be crossed out clearly.

• Please write legibly! We will only correct solutions that we can read.

• Manage your time carefully (take into account the number of points for each question).

• Don’t forget to include header comments in features.

• Please immediately tell the exam supervisors if you feel disturbed during the exam.

Good luck!
<table>
<thead>
<tr>
<th>Question</th>
<th>Number of possible points</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>70</td>
<td></td>
</tr>
</tbody>
</table>
1 Axiomatic semantics (20 points)

Consider the following Hoare triple:

\{ x > 0 \}
y := 1;
z := 0;
while (z != x) do
    z := z + 1;
y := y \cdot z
end
\{ y = x! \}

The ! in the postcondition denotes the factorial function, i.e. \( x! = x \cdot (x-1) \cdot (x-2) \cdot \ldots \cdot 1 \) and \( 0! = 1 \). Prove that this triple is a theorem of Hoare’s axiomatic system for partial correctness. The proof should be a sequence of lines with three elements on each line: line number; proposition; justification.
2 Program analysis (15 Points)

The assignment to variable $v$ by statement $S$ of program $Prog$ reaches a point $p$ in $Prog$ if there exists a control-flow path from $S$ to $p$ on which no statement reassigns $v$. This can be formulated as a labelling scheme on control-flow graphs:

- A label is a pair $(\text{varname} : \text{statementnumber})$, where $\text{varname}$ is a variable of $Prog$ and $\text{statementnumber}$ the number of a node in the control-flow graph of $Prog$. Each node $S$ is numbered with a unique positive integer, $\text{number}(S)$.

- Each node $S$ has two sets of labels: the incoming label set $\text{In}(S)$ and the outgoing label set $\text{Out}(S)$:
  \[
  \text{In}(S) = \begin{cases} 
  \emptyset & \text{if } S \text{ is the node in } Prog \text{ at which control-flow starts}. \\
  \bigcup_{S_0 \in \text{pred}(S)} \text{Out}(S_0) & \text{otherwise, where } \text{pred}(S) \text{ denotes the set of all nodes with edges pointing to } S. 
  \end{cases}
  \]
  \[
  \text{Out}(S) = \begin{cases} 
  (\text{In}(S) - \{(\text{varname} : n)|n \in \mathbb{N}\}) \cup \{(\text{varname} : \text{number}(S))\} & \text{if } S \text{ is of the form } \text{varname} := \text{expression}. \\
  \text{In}(S) & \text{otherwise.}
  \end{cases}
  \]

Draw the control-flow graph of the following program fragment and annotate its nodes with reachability labels:

```plaintext
a := 2
b := -a
if b <= a then
  a := b * 2
  b := a
else
  b := b + 4
end
b := b + 1
```

\[6\]
3 Separation logic (15 Points)

1. (8 points) Consider program states A and B in the following figure:

\[ \begin{array}{c}
\text{Stack} \\
\begin{array}{c}
| & 2 \\
X & 2 \\
Y & 2 \\
\end{array}
\end{array} \quad \begin{array}{c}
\text{Heap} \\
\begin{array}{c}
| & 2 \\
X & 2 \\
Y & 2 \\
\end{array}
\end{array} \]

Indicate in the table whether or not a given assertion is satisfied by states A and B respectively. Indicate satisfaction with a T and non-satisfaction with an F.

\[
\begin{array}{|c|c|c|}
\hline
x \rightarrow 2 & A & B \\
\hline
y \rightarrow 2 \ast \text{true} & & \\
\hline
x \rightarrow 2 \ast y \rightarrow 2 & & \\
\hline
x \rightarrow 2 \land y \rightarrow 2 & & \\
\hline
\end{array}
\]

2. (4 points) Do the following implications hold? If an implication holds, explain why. If it does not hold, provide a counterexample.

\[
(\neg P \land Q) \Rightarrow (P \ast Q) \quad (1)
\]

\[
(P \ast Q) \Rightarrow (P \land Q) \quad (2)
\]
3. (3 points) Consider the following derivation attempt:

\[
\{a \mapsto 30\} b := [a] \{a \mapsto 30 \land b = 30\} \\
\{(a \mapsto 30) \ast b \mapsto 45\} b := [a] \{((a \mapsto 30 \land b = 30) \ast b \mapsto 45)\}
\]

Explain why the frame rule was wrongly applied.
4 Abstract interpretation (10 Points)

Consider the grammar of integer expressions

\[ e ::= i \mid e_1 + e_2 \mid e_1 - e_2 \mid e_1 \times e_2 \]

where \( i \in I \) and \( I = \{-1000, -999, \ldots, 999, 1000\} \).

Devise an abstract interpretation scheme to determine whether a given \( e \) represents an even or odd integer. You may assume the existence of a function \( f : I \to \{\text{even}, \text{odd}\} \) that maps \( i \) to \text{even} if \( i \) is even and \( i \) to \text{odd} if \( i \) is odd.
5 Model checking (10 Points)

Here is the semantics of a subset of LTL formulas:

For a path \(\pi = s_1 \rightarrow s_2 \rightarrow \ldots\) in a model \(M = (S, \rightarrow, L)\) and an LTL formula \(\phi\):

- \(\pi \models true\)
- \(\pi \not\models false\)
- \(\pi \models p\) iff \(p \in L(s_1)\)
- \(\pi \models \neg \phi\) iff \(\pi \not\models \phi\)
- \(\pi \models \phi_1 \land \phi_2\) iff \(\pi \models \phi_1\) and \(\pi \models \phi_2\)
- \(\pi \models \phi_1 \lor \phi_2\) iff \(\pi \models \phi_1\) or \(\pi \models \phi_2\)
- \(\pi \models \phi_1 \Rightarrow \phi_2\) iff \(\pi \models \phi_2\) whenever \(\pi \models \phi_1\)
- \(\pi \models X \phi\) iff \(\pi^2 \models \phi\) \((\pi^1 = s_i \rightarrow s_{i+1} \rightarrow \ldots)\)
- \(\pi \models G \phi\) iff for all \(i \geq 1\), \(\pi^i \models \phi\)
- \(\pi \models F \phi\) iff there is some \(i \geq 1\) such that \(\pi^i \models \phi\)
- \(\pi \models \phi_1 \cup \phi_2\) iff there is some \(i \geq 1\) such that \(\pi^i \models \phi_2\) and for all \(1 \leq j < i\), \(\pi^j \models \phi_1\)

\(M, s \models \phi\) for a state \(s \in S\) iff for every path \(\pi\) in \(M\) starting at \(s\) we have \(\pi \models \phi\).

1. (6 points) Consider the transition system \(M\):

Do the following statements hold? If yes, provide a brief justification, if no, provide a counterexample path.

(a) \(M, s_0 \models X (q \land r)\)

................................................................................................................

................................................................................................................

................................................................................................................

................................................................................................................
2. (4 points) Express the following specifications as LTL formulas:

(a) A certain process will eventually be permanently *deadlocked*.

\( M, s_0 \models G \neg(p \land r) \)

(b) A downwards travelling lift at the fifth floor with passengers wishing to go to the second floor does not change its direction until it reaches the second floor.

\( M, s_0 \models G F p \)