Two hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Understanding Programming Languages

Date: Wednesday 23rd January 2013
Time: 09:45 - 11:45

Please answer any THREE Questions from the FIVE Questions provided

Use a SEPARATE answerbook for each QUESTION.

This is a CLOSED book examination

The use of electronic calculators is NOT permitted

[PTO]
Answer *three* of the five questions. All questions refer to the **Repeat**
language of the Appendix

1. a) Give a natural semantics for the statement part of the **Repeat**
language.
   (5 marks)

   b) Prove that the semantics of **Repeat** in the Appendix is equivalent to your natural
   semantics.
   (10 marks)

   c) Extend your natural semantics to handle non-deterministic or statements.
   (5 marks)
2. a) Extend the semantics of Repeat in the Appendix to include the abort statement.  
   (5 marks)

b) Extend the semantics of Repeat in the Appendix to include the non-deterministic or statements.  
   (5 marks)

c) Extend the semantics of Repeat in the Appendix to include concurrent parallel statements, or par statements.  
   (5 marks)

d) Modify your answer to part (c) so that critical sections can be protected by a protect/end pair.  
   (5 marks)
3. a) Give a compiler and abstract machine for the statement part of the Repeat language.  
(5 marks)

b) Prove that the semantics of Repeat in the Appendix is equivalent to your compiler and abstract machine.  
(10 marks)

c) Extend your compiler and abstract machine to handle non-deterministic Abort statements.  
(5 marks)
4. a) Give a denotational semantics for the statement part of the Repeat language. (5 marks)

b) Prove that the semantics of Repeat in the Appendix is equivalent to your denotational semantics. (10 marks)

c) Extend your denotational semantics to handle while statements. (5 marks)
5. a) Give a axiomatic semantics for partial correctness for the statement part of the Repeat language. (5 marks)

b) Prove that the semantics of Repeat in the Appendix is sound and complete with respect to your axiomatic semantics. (10 marks)

c) Extend your axiomatic semantics to handle non-deterministic or statements. (5 marks)
\[ a ::= n \mid x \mid a_1 + a_2 \mid a_1 \ast a_2 \mid a_1 - a_2 \]
\[ b ::= \text{true} \mid \text{false} \mid a_1 = a_2 \mid a_1 \leq a_2 \mid \neg b \mid b_1 \land b_2 \]
\[ S ::= x := a \mid \text{skip} \mid S_1 ; S_2 \mid \text{if } b \text{ then } S_1 \text{ else } S_2 \mid \text{repeat } S \text{ until } b \]

Table 1: Syntax of \textbf{Repeat}

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{ass} ___</td>
<td>\texttt{&lt; x := a, s &gt;} \implies \texttt{s[x \mapsto A[![a]!] s]}</td>
</tr>
<tr>
<td>\texttt{skip} ___</td>
<td>\texttt{&lt; skip, s &gt;} \implies \texttt{s}</td>
</tr>
<tr>
<td>\texttt{comp} ___</td>
<td>\texttt{&lt; S_1, s &gt;} \implies \texttt{&lt; S'_1, s' &gt;}</td>
</tr>
<tr>
<td>\texttt{comp} ___</td>
<td>\texttt{&lt; S_1; S_2, s &gt;} \implies \texttt{&lt; S'_1; S_2, s' &gt;}</td>
</tr>
<tr>
<td>\texttt{if} ___</td>
<td>\texttt{&lt; if } b \texttt{ then } S_1 \texttt{ else } S_2, s &gt; \implies \texttt{&lt; S_1, s &gt;} \quad \text{if } B[![b]!] s = \texttt{tt}</td>
</tr>
<tr>
<td>\texttt{if} ___</td>
<td>\texttt{&lt; if } b \texttt{ then } S_1 \texttt{ else } S_2, s &gt; \implies \texttt{&lt; S_2, s &gt;} \quad \text{if } B[![b]!] s = \texttt{ff}</td>
</tr>
<tr>
<td>\texttt{repeat} ___</td>
<td>\texttt{&lt; repeat } S \texttt{ until } b, s &gt; \implies \texttt{&lt; S; if } b \texttt{ then skip else repeat } S \texttt{ until } b, s &gt;</td>
</tr>
</tbody>
</table>

Table 2: Structural Operational Semantics for \textbf{Repeat}

\[
\begin{align*}
B[\![\text{true}]\!] s &= \texttt{tt} \\
B[\![\text{false}]\!] s &= \texttt{ff} \\
B[\![a_1 = a_2]\!] s &= \begin{cases} 
\texttt{tt} & \text{if } A[\![a_1]\!] s = A[\![a_2]\!] s \\
\texttt{ff} & \text{if } A[\![a_1]\!] s \neq A[\![a_2]\!] s 
\end{cases} \\
B[\![a_1 \leq a_2]\!] s &= \begin{cases} 
\texttt{tt} & \text{if } A[\![a_1]\!] s \leq A[\![a_2]\!] s \\
\texttt{ff} & \text{if } A[\![a_1]\!] s > A[\![a_2]\!] s 
\end{cases} \\
B[\![\neg b]\!] s &= \begin{cases} 
\texttt{tt} & \text{if } B[\![b]\!] s = \texttt{ff} \\
\texttt{ff} & \text{if } B[\![b]\!] s = \texttt{tt} 
\end{cases} \\
B[\![b_1 \land b_2]\!] s &= \begin{cases} 
\texttt{tt} & \text{if } B[\![b_1]\!] s \text{ \textbf{and} } B[\![b_2]\!] s \\
\texttt{ff} & \text{if } \textbf{not} (B[\![b_1]\!] s \text{ \textbf{and} } B[\![b_2]\!] s) 
\end{cases}
\end{align*}
\]

Table 3: The Semantics of Boolean Expressions

\[
\begin{align*}
A[\![n]\!] s &= A[\![n]\!] \\
A[\![x]\!] s &= x s \\
A[\![a_1 + a_2]\!] s &= A[\![a_1]\!] s + A[\![a_2]\!] s \\
A[\![a_1 \ast a_2]\!] s &= A[\![a_1]\!] s \ast A[\![a_2]\!] s \\
A[\![a_1 - a_2]\!] s &= A[\![a_1]\!] s - A[\![a_2]\!] s
\end{align*}
\]

Table 4: The Semantics of Arithmetic Expressions