COMS22202: 2015/16

Language Engineering

Dr Oliver Ray
(csxor@Bristol.ac.uk)

Department of Computer Science
University of Bristol

Tuesday 8th March, 2016

Question 1(a): proof tree

$$\text{cons}_1\ \dfrac{\text{loop}\ \dfrac{\text{cons}_2\ \dfrac{\text{ass}\ \dfrac{}{\left\{\begin{matrix}(x-2)\%2=n\%2\\ x\geq 0\end{matrix}\right\}\ \texttt{x:=x-2}\ \left\{\begin{matrix}x\%2=n\%2\\ x\geq 0\end{matrix}\right\}}}{\left\{\begin{matrix}x\%2=n\%2\\ x\geq 0\\ 2\leq x\end{matrix}\right\}\ \texttt{x:=x-2}\ \left\{\begin{matrix}x\%2=n\%2\\ x\geq 0\end{matrix}\right\}}}{\left\{\begin{matrix}x\%2=n\%2\\ x\geq 0\end{matrix}\right\}\ \texttt{while 2}\leq\texttt{x do x:=x-2}\ \left\{\begin{matrix}x\%2=n\%2\\ x\geq 0\\ \neg(2\leq x)\end{matrix}\right\}}}{\left\{\begin{matrix}x=n\\ n\geq 0\end{matrix}\right\}\ \texttt{while 2}\leq\texttt{x do x:=x-2}\ \left\{\begin{matrix}x=n\%2\end{matrix}\right\}}$$

Question 1(a): proof obligations

Pre-1

(1)	$x = n$	given
(2)	$n \geq 0$	given
(3)	$x\%2 = n\%2$	taking modulus of both sides of (1)
(4)	$x \geq 0$	substituting x for n in (2) using (1)

Post-1

(1)	$x\%2 = n\%2$	given
(2)	$x \geq 0$	given
(3)	$\neg(2 \leq x)$	given
(4)	$\forall a, b : 0 \leq a < b \rightarrow a\%b = a$	lemma
(5)	$2 > x$	from (3) by properties of $>$
(6)	$0 \leq x < 2$	$\wedge$-introduction on (2) and (5)
(7)	$x = x\%2$	from (4) using (6) with $a = x$ and $b = 2$
(8)	$x = n\%2$	from (7) and (1) by transitivity of $=$

Pre-2

(1)	$x\%2 = n\%2$	given
(2)	$x \geq 0$	given
(3)	$2 \leq x$	given
(4)	$\forall a, b : a\%b = (a - b)\%b$	lemma
(5)	$x \geq 0$	from (2)
(6)	$(x - 2)\%2 = x\%2$	from (4) with $a = x$ and $b = 2$
(7)	$(x - 2)\%2 = n\%2$	from (6) and (1) by transitivity of $=$

Post-2

trivial

Here are the proofs of the obligations generated by the consequence rule. In each case, the conjunction of red formulae entail the conjunction of green ones - assuming the truth of the lemmas in blue (that we could also prove in this way if we wished).

Question 1(a): tableau format

$\{x{=}n \wedge n \geq 0\}$

Pre-1

$\{x\%2{=}n\%2 \wedge x \geq 0\}$

while 2≤x do (

$\{x\%2{=}n\%2 \wedge x \geq 0 \wedge 2 \leq x\}$

Pre-2

$\{(x-2)\%2{=}n\%2 \wedge (x-2) \geq 0\}$

x:=x-2

Post-2

$\{x\%2{=}n\%2 \wedge x \geq 0\}$

)

$\{x\%2{=}n\%2 \wedge x \geq 0 \wedge \neg(2 \leq x)\}$

$0 \leq x < 2$

Post-1

$x = x\%2$

$\{x{=}n\%2\}$

This is the layout I recommend for axiomatic proofs. It has exactly the same information as a tree but avoids much of the redundancy (so is much easier to use in an exam).

This tableau generates the same obligations as the earlier tree, so the proofs can be re-used. The arrows are useful to highlight key intermediate results and their dependencies.

Question 1(b):

The answer to this question is given in Example 6.9 on pages 181-2 of the text book (using the invariant INV= x>0 → y.x!=n! ^ n>=x)

Note that while the book uses a more verbose notation, its proof can be used very easily to construct a tree-based or tableau-based proof of the form used in the lectures (which are both much easier to read!).

Note that Neilson’s notation is meant to emphasise the fact that pre- and post-conditions can be formally regarded as functions that map states to truth values (which return true iff the logical formula is true in a given state). This is the basis of their so-called “extensional” approach and why they end up embedding the denotational semantics of arithmetic and Boolean expression into their presentation of the axiom schemata. But while these details are important for proving soundness and completeness properties of the axiomatic semantics, they are not needed to actually prove the correctness of the programs we will be considering on this unit. That is why I encourage you to use the simplified notation I have presented (and which Nielson do also often end up using through the use of the conventions described on p.177).

It is worth noting the distinction between the symbols for logical entailment |= and material implication → used in the lecture notes and the symbols |- and => used in the book. The symbol |- means provability. They write |- {P}S{Q} to denote the fact that the assertion {P}S{Q} can be derived using the axiomatic calculus. They subscript the |- symbol with a P or T to denote the partial or total semantics. They use this to break up a axiomatic proofs into simpler parts to avoid having to represent the entire proof tree as done in the slides. Similarly their use of => in proof obligations can be reduced to |= in the way shown in the slides. This is how we avoided these symbols along with the embedding of the denotational semantics into the proof rules.

But if we were to use the conventions described in the book then we have the following soundness and completeness results (as proven in the book):

|- {P}S{Q} iff for all states s and s’ it holds that (P s)=tt and [[S]]s=s’ implies (Q s’)=tt

Question 1(c):

The answer to this question is closely related to Example 6.8 on pages 180 of the text book.

Note that you are not required to understand the whole of Chapter 6, but only Section 6.2 pp.175-186 (excluding the technical details of the assertion language on p.177) and Section 6.4 pp.191-4.