ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

D. Basin and P. Müller

Formal Methods and Functional Programming

Exercise Sheet 12: Axiomatic Semantics 2 Solutions

Submission deadline: May 31st, 2010

Two Extra Hoare Logic derivation rules (not required for exam)

In order to complete the headache (and possibly assignment 1 of this sheet, depending on your approach) some additional derivation rules are required, which facililate "plugging in" a derivation for one statement into a larger context. For example, imagine that you have proved a certain triple about a loop, and now that loop is to occur in a larger block of code to achieve a particular goal. If we can somehow "plug in" the proof we've already made, this will allow for a kind of modular development of proofs for large programs - we can prove the smaller parts, and then plug them into bigger programs, with a little work. In particular, we can re-use proofs for the same blocks of code used in different contexts.

Two issues arise when plugging a small proof into a larger context. Firstly, it is usually the case that the larger context is concerned with extra properties which are irrelevant for the small proof. Nonetheless, it is typically important that we can hold onto those properties in order to prove the larger program. To do this, we need a rule which allows extra information to be added to assertions, provided that the extra information is not affected by the smaller code. This is achieved by the *frame rule* (which you may have discussed in your sessions):

$$\frac{\{\ P\ \}\ s\ \{\ Q\ \}}{\{\ P \wedge R\ \}\ s\ \{\ Q \wedge R\ \}}\ (\textit{if } \mathit{fv}(s) \cap \mathit{fv}(R) = \emptyset)$$

The side-condition is expressed in terms of fv - the free variables in a statement or predicate, and is there to ensure that the extra predicate R cannot have its validity affected by the statement s (in particular, s does not assign to any of the variables mentioned in R). Using this rule, we can then extend an existing proof for a small piece of code, by adding on whatever extra information the outer context needs to hold onto.

The second issue which arises, is if the small code makes use of logical variables in its pre-condition which are not used in the larger code. For example, such logical variables are typically used to express that a certain variable does not change during execution of a statement, for example in triples of the form $\{\ P \wedge x = X_0\ \}\ s\ \{\ Q \wedge x = X_0\ \}$. Now, if we want to use this triple inside a larger derivation in which the logical variable X_0 is not used, we have a problem

- we do not have any rule for *introducing* a new logical variable halfway through a derivation. Intuitively, what we would like to be able to say is that the first time the new logical variable is encountered in the derivation, say in the pre-condition $x = X_0$, the new logical variable takes whatever value is appropriate at that point (whatever value x has at that point). Of course, we know that x must have *some* value at that point, and whatever it is, we would like the derivation for s to proceed accordingly. This "intuitive" step can be partly supported by the *existential rule* below:

$$\frac{\{\ P\ \}\ s\ \{\ Q\ \}}{\{\ \exists X.P\ \}\ s\ \{\ Q\ \}}\ (\textit{if}\ X \notin \textit{fv}(Q))$$

Note that the side-condition only lets us apply this rule in a very special case - when the logical variable X does not occur in the post-condition Q. This is a requirement for soundness - we cannot make a step which says "whatever the value of X is in the pre-condition" while X is mentioned elsewhere. This rule, in combination with the frame rule, can be used to solve the problem described above, as follows:

To show what these rules can be used for, suppose now that we have proved a triple $\{\ P \wedge x = X_0\ \}\ s\ \{\ Q\ \}$ and we wish to make use of it inside a larger proof for a larger program, in which the logical variable X_0 does not occur. Using the surrounding code, we need to find some alternative expression e for the value of x in the pre-condition of s (this can typically be found by making the derivation up to this point, or by manual inspection). The expression e (since it comes from the larger program) will not mention the variable X_0, and will be such that we can derive that $x = e$ holds when we reach the pre-condition of s. Furthermore, suppose that the expression e does not contain any variables that are modified by s (this restriction is not essential, but will be sufficient for our purposes). Then, by applying the frame rule, we can extend our original triple:

$$\frac{\{\ P \wedge x = X_0\ \}\ s\ \{\ Q\ \}}{\{\ P \wedge x = X_0 \wedge X_0 = e\ \}\ s\ \{\ Q \wedge X_0 = e\ \}}\ (\textit{fv}(s) \cap \textit{fv}(e) = \emptyset)$$

Now, by employing the rule of consequence and the existential rule, we can eliminate the need to mention the variable X_0:

$$\dfrac{\dfrac{\dfrac{\{\ P \wedge x = X_0 \wedge X_0 = e\ \}\ s\ \{\ Q \wedge X_0 = e\ \}}{\{\ P \wedge x = X_0 \wedge X_0 = e\ \}\ s\ \{\ Q[e/X_0]\ \}}\ (\textit{conseq.})}{\{\ \exists X_0.(P \wedge x = X_0 \wedge X_0 = e)\ \}\ s\ \{\ Q[e/X_0]\ \}}\ (\textit{existential})}{\{\ P[x/X_0] \wedge x = e\ \}\ s\ \{\ Q[e/X_0]\ \}}\ (\textit{conseq.})$$

We now have a triple for s which is suitable for insertion into the larger proof (you may wish to check that the rules above satisfy their side-conditions). Note that if the larger proof requires additional information to be preserved across the execution of s, we may need to further apply the frame rule to conjoin this information.

To summarise: if we have proven $\{\ P \wedge x = X_0\ \}\ s\ \{\ Q\ \}$ and want to use it in a larger proof in which the logical variable does not occur, we can employ the following recipe:

1. Find an expression e for the value of x in the pre-condition of x. The expression e comes

from analysis of the code which comes before s - it should not mention X_0 (and, for simplicity, we assume e will also not have variables in common with s).

2. We can use the frame, existential and consequence rules to obtain the triple $\{ P[x/X_0] \wedge x = e \}\ s\ \{ Q[e/X_0] \}$.
3. Use the frame rule further to conjoin any extra information which needs to be preserved across execution of s.
4. Use the resulting triple in the larger proof, when dealing with s.

Let's see this in action on a simple example:

Assignment 1

Let s be the following statement:

```
y := 1;
z := 0;
while z<x do
  y := y * 2;
  z := z + 1
end
```

a) Assuming X_0 is the original value of x, a suitable loop invariant is $x = X_0 \wedge y = 2^z \wedge z \leq x$.

b) A suitable loop variant is $(x - z)$. We can now prove the following triple, characterising the general behaviour of the statement s (using the invariant and variant above): $\{ x = X_0 \wedge X_0 \geq 0 \}\ s\ \{ \Downarrow x = X_0 \wedge y = 2^{X_0} \}$. Here is the proof outline:

$\{x = X_0 \wedge X_0 \geq 0\}$
`y := 1;`
$\{x = X_0 \wedge X_0 \geq 0 \wedge y = 1\}$
`z := 0;`
$\{x = X_0 \wedge X_0 \geq 0 \wedge y = 1 \wedge z = 0\}$
$\Rightarrow$
$\{x = X_0 \wedge y = 2^z \wedge z \leq x\}$
`while z<x do`
  $\{x = X_0 \wedge y = 2^z \wedge z \leq x \wedge z < x \wedge (x - z) = V_0\}^\star$
  $\Rightarrow$
  $\{x = X_0 \wedge y * 2 = 2^{z+1} \wedge z + 1 \leq x \wedge (x - (z + 1)) < V_0\}$
  `y := y * 2;`
  $\{x = X_0 \wedge y = 2^{z+1} \wedge z + 1 \leq x \wedge (x - (z + 1)) < V_0\}$
  `z := z + 1`
  $\{x = X_0 \wedge y = 2^z \wedge z \leq x \wedge (x - z) < V_0\}$
`end`
$\{x = X_0 \wedge y = 2^z \wedge z \leq x \wedge z \geq x\}$
$\Rightarrow$
$\{x = X_0 \wedge y = 2^{X_0}\}$

* *side-condition: this predicate implies* $(x - z) \geq 0$

However, the next part of the question requires us to make use of this loop in a more-specific context (when we know $x = 10$ in the pre-condition, and do not use the logical variable X_0).

c) We wish to prove that $\vdash \{\ x{=}10\ \}\ s\ \{\ \Downarrow y{=}1024\ \}$. We could of course construct an entirely new proof outline to show this triple. However, the recipe above tells us how to adapt our existing proof to this more-specific context: using the recipe we can re-use our previous proof for the loop. Firstly, we need to find an expression e such that $X_0 = e$ in the pre-condition of s. Here, we obviously choose e to be 10. Now, according to the recipe, we can adapt our proven assertion $\{\ x = X_0 \wedge X_0 \geq 0\ \}\ s\ \{\ \Downarrow x = X_0 \wedge y = 2^{X_0}\ \}$ to the new assertion $\{\ (X_0 \geq 0)[x/X_0] \wedge x = 10\ \}\ s\ \{\ \Downarrow (x = X_0 \wedge y = 2^{X_0})[10/X_0]\ \}$, i.e., we obtain the triple $\{\ x \geq 0 \wedge x = 10\ \}\ s\ \{\ \Downarrow x = 10 \wedge y = 2^{10}\ \}$. Finally, we can obtain the desired triple using the rule of consequence:

$$\frac{\{\ x \geq 0 \wedge x = 10\ \}\ s\ \{\ \Downarrow x = 10 \wedge y = 2^{10}\ \}}{\{\ x = 10\ \}\ s\ \{\ \Downarrow y = 1024\ \}}$$

Assignment 2

The right-to-left direction ($\Leftarrow$) can be shown directly: Suppose that there exist P',Q',R' with $P \Rightarrow P'$ and $Q' \Rightarrow Q$ and $\vdash \{\ P'\ \}\ s_1\ \{\ \Downarrow R'\ \}$ and $\vdash \{\ R'\ \}\ s_2\ \{\ \Downarrow Q'\ \}$. Then we can construct the following derivation (using first the rule for sequential composition, and then the

rule of consequence):

$$\dfrac{\dfrac{\{\ P'\ \}\ s_1\ \{\ \Downarrow R'\ \} \quad \{\ R'\ \}\ s_2\ \{\ \Downarrow Q'\ \}}{\{\ P'\ \}\ s_1; s_2\ \{\ \Downarrow Q'\ \}}}{\{\ P\ \}\ s_1; s_2\ \{\ \Downarrow Q\ \}}$$

For the left-to-right direction ($\Rightarrow$) we proceed by induction on the structure of the derivation of $\{\ P\ \}\ s_1; s_2\ \{\ \Downarrow Q\ \}$, considering cases for the last rule applied. Given the form of the statement, there are only two possible cases - either the rule for sequential composition or the rule of consequence was the last rule applied:

Case 1 - sequential composition rule: Then from the form of the rule, there must be some predicate R such that we have derivations for $\{\ P\ \}\ s_1\ \{\ \Downarrow R\ \}$ and $\{\ R\ \}\ s_2\ \{\ Q\ \}$. Choosing P' to be P, Q' to be Q and R' to be R, we have exactly the four properties required.

Case 2 - rule of consequence: Then from the form of the rule, there must be some predicates P'' and Q'' such that $P \Rightarrow P''$ and $Q'' \Rightarrow Q$ and we have a (sub-)derivation for $\{\ P''\ \}\ s_1; s_2\ \{\ \Downarrow Q''\ \}$. By applying the induction hypothesis to this sub-derivation, we know that there exist P',Q' and R' such that $P'' \Rightarrow P'$ and $Q' \Rightarrow Q''$ and $\vdash \{\ P'\ \}\ s_1\ \{\ \Downarrow R'\ \}$ and $\vdash \{\ R'\ \}\ s_2\ \{\ \Downarrow Q'\ \}$. By transitivity of implication, we have $P \Rightarrow P'$ and $Q' \Rightarrow Q$, which concludes the case.

Assignment 3

This question concerns termination and the Zune bug, as discussed in the lectures.

a) If the triple $\{\ \mathit{true}\ \}\ s\ \{\ \Downarrow \mathit{true}\ \}$ can be derived, this means that the statement s is guaranteed to terminate (regardless of the initial state).

b) See the lecture slides, p.210

c) See the lecture slides, p.211

Assignment 4

Note that there are two ways to proceed - we could either apply the result of Sheet 12 Assignment 2 directly (twice), or work by induction on the structure of the assumed derivation, and then use Sheet 12 Assignment 2 (once) during the proof. The latter approach yields a simpler proof, since the induction hypothesis makes everything straightforward in the case that the rule of consequence was the last applied.

We proceed by induction on the structure of the derivation of $\{\ P\ \}\ (s_1; s_2); s_3\ \{\ \Downarrow Q\ \}$, considering cases for the last rule applied. Given the form of the statement, there are only two possible cases - either the rule for sequential composition or the rule of consequence was the last rule applied:

Case 1 - sequential composition rule: Then from the form of the rule, there must be some predicate R such that we have derivations for $\{\ P\ \}\ (s_1; s_2)\ \{\ \Downarrow R\ \}$ and $\{\ R\ \}\ s_3\ \{\ \Downarrow Q\ \}$. By applying the result of Sheet 12 Assignment 2 to the former of these two derivations, we know that there exist predicates P',R',T' such that $P \Rightarrow P'$ and $R' \Rightarrow R$ and $\vdash \{\ P'\ \}\ s_1\ \{\ \Downarrow T'\ \}$ and $\vdash \{\ T'\ \}\ s_2\ \{\ \Downarrow R'\ \}$. Combining all of this information together, we can construct the following derivation (in which $(*)$s mark the uses of the rule of consequence, to aid readability):

$$\dfrac{\{\ P'\ \}\ s_1\ \{\ \Downarrow T'\ \} \qquad \dfrac{\{\ T'\ \}\ s_2\ \{\ \Downarrow R'\ \} \quad \dfrac{\{\ R\ \}\ s_3\ \{\ \Downarrow Q\ \}}{\{\ R'\ \}\ s_3\ \{\ \Downarrow Q\ \}}\ (*)}{\{\ T'\ \}\ s_2; s_3\ \{\ \Downarrow Q\ \}}}{\dfrac{\{\ P'\ \}\ s_1; (s_2; s_3)\ \{\ \Downarrow Q\ \}}{\{\ P\ \}\ s_1; (s_2; s_3)\ \{\ \Downarrow Q\ \}}\ (*)}$$

Case 2 - rule of consequence: Then from the form of the rule, there must be some predicates P' and Q' such that $P \Rightarrow P'$ and $Q' \Rightarrow Q$ and we have a (sub-)derivation for $\{\ P'\ \}\ (s_1; s_2); s_3\ \{\ \Downarrow Q'\ \}$. By applying the induction hypothesis to this sub-derivation, we know that there exists a derivation of $\{\ P'\ \}\ s_1; (s_2; s_3)\ \{\ \Downarrow Q'\ \}$. Extending this new derivation by the rule of consequence, we obtain $\{\ P\ \}\ s_1; (s_2; s_3)\ \{\ \Downarrow Q\ \}$ as required.

Assignment 5 - Headache of the week

The code used for s is (as used in Sheet 7 question 1) as follows:

```
z := 0;
v := 0;
while v<y do
  v := 1;
  i := 0;
  while i<x do
    v := v*(z+1);
    i := i+1
  end;
  if v<=y then
    z := z+1
  else
    skip
  end
end
```

For the outer loop, the invariant used (see Sheet 7 solutions for a discussion of the main idea) is:
$x{=}X_0 \wedge y{=}Y_0 \wedge X_0{>}0 \wedge z{\geq}0 \wedge z^x{\leq}y \wedge (v{\leq}y \Rightarrow v{=}z^x) \wedge (v{>}y \Rightarrow v{=}(z+1)^x)$
The variant used is $\mathit{max}(0,(y-v))$.

For the inner loop, the invariant used is:
$x{=}X_0 \wedge X_0{>}0 \wedge i{\leq}x \wedge v{=}(z{+}1)^i \wedge z{\geq}0 \wedge z^x{<}y \wedge y{=}Y_0 \wedge V_0{=}(y-z^x)$
The variant used is $(x-i)$

The proof outline is as follows:

$\{x{=}X_0 \wedge y{=}Y_0 \wedge X_0{>}0 \wedge Y_0{>}0\}$
`z := 0;`
$\{x{=}X_0 \wedge y{=}Y_0 \wedge X_0{>}0 \wedge Y_0{>}0 \wedge z{=}0\}$
`v := 0;`
$\{x{=}X_0 \wedge y{=}Y_0 \wedge X_0{>}0 \wedge Y_0{>}0 \wedge z{=}0 \wedge v{=}0\}$
$\Rightarrow$
$\{x{=}X_0 \wedge y{=}Y_0 \wedge X_0{>}0 \wedge z{\geq}0 \wedge z^x{\leq}y \wedge (v{\leq}y \Rightarrow v{=}z^x) \wedge (v{>}y \Rightarrow v{=}(z+1)^x)\}$
`while v<y do`
$\{x{=}X_0 \wedge y{=}Y_0 \wedge X_0{>}0 \wedge z{\geq}0 \wedge z^x{\leq}y \wedge (v{\leq}y \Rightarrow v{=}z^x) \wedge (v{>}y \Rightarrow v{=}(z+1)^x) \wedge v{<}y \wedge V_0{=}\mathit{max}(0,(y-v))\}^\star$
$\Rightarrow$
$\{x{=}X_0 \wedge y{=}Y_0 \wedge X_0{>}0 \wedge z{\geq}0 \wedge z^x{<}y \wedge V_0{=}(y{-}z^x)\}$
`v := 1;`
$\{x{=}X_0 \wedge y{=}Y_0 \wedge X_0{>}0 \wedge z{\geq}0 \wedge z^x{<}y \wedge V_0{=}(y{-}z^x) \wedge v = 1\}$
`i := 0;`
$\{x{=}X_0 \wedge y{=}Y_0 \wedge X_0{>}0 \wedge z{\geq}0 \wedge z^x{<}y \wedge V_0{=}(y{-}z^x) \wedge v = 1 \wedge i = 0\}$
$\Rightarrow$
$\{x{=}X_0 \wedge X_0{>}0 \wedge i{\leq}x \wedge v{=}(z{+}1)^i \wedge z{\geq}0 \wedge z^x{<}y \wedge y{=}Y_0 \wedge V_0{=}(y-z^x)\}$
`while i<x do`
$\{x{=}X_0 \wedge X_0{>}0 \wedge i{\leq}x \wedge v{=}(z{+}1)^i \wedge i{<}x \wedge z{\geq}0 \wedge z^x{<}y \wedge y{=}Y_0 \wedge V_0{=}(y-z^x) \wedge V_1{=}(x{-}i)\}^{\star\star}$
$\Rightarrow$
$\{x{=}X_0 \wedge X_0{>}0 \wedge (i{+}1){\leq}x \wedge (v{*}(z{+}1)){=}(z{+}1)^{i+1} \wedge z{\geq}0 \wedge z^x{<}y \wedge y{=}Y_0 \wedge V_0{=}(y-z^x) \wedge (x{-}(i{+}1)){<}V_1\}$
`v := v*(z+1);`
$\{x{=}X_0 \wedge X_0{>}0 \wedge (i{+}1){\leq}x \wedge v{=}(z{+}1)^{i+1} \wedge z{\geq}0 \wedge z^x{<}y \wedge y{=}Y_0 \wedge V_0{=}(y-z^x) \wedge (x{-}(i{+}1)){<}V_1\}$
`i := i+1`
$\{x{=}X_0 \wedge X_0{>}0 \wedge i{\leq}x \wedge v{=}(z{+}1)^i \wedge z{\geq}0 \wedge z^x{<}y \wedge y{=}Y_0 \wedge V_0{=}(y-z^x) \wedge (x{-}i){<}V_1\}$
`end;`
$\{x{=}X_0 \wedge X_0{>}0 \wedge i{\leq}x \wedge v{=}(z{+}1)^i \wedge z{\geq}0 \wedge z^x{<}y \wedge y{=}Y_0 \wedge V_0{=}(y-z^x) \wedge i{\geq}x\}$
$\Rightarrow$
$\{x{=}X_0 \wedge y{=}Y_0 \wedge X_0{>}0 \wedge v{=}(z{+}1)^x \wedge z{\geq}0 \wedge z^x{<}y \wedge V_0{=}(y-z^x)\}$
`if v<=y then`
$\{x{=}X_0 \wedge y{=}Y_0 \wedge X_0{>}0 \wedge v{=}(z{+}1)^x \wedge z{\geq}0 \wedge z^x{<}y \wedge V_0{=}(y-z^x) \wedge v{\leq}y\}$
$\Rightarrow$
$\{x{=}X_0 \wedge y{=}Y_0 \wedge X_0{>}0 \wedge v{=}(z{+}1)^x \wedge v{\leq}y \wedge (z{+}1){\geq}0 \wedge (z{+}1{-}1)^x{<}y \wedge V_0{=}(y-(z{+}1{-}1)^x)\}$
`z := z+1`
$\{x{=}X_0 \wedge y{=}Y_0 \wedge X_0{>}0 \wedge v{=}z^x \wedge v{\leq}y \wedge z{\geq}0 \wedge (z{-}1)^x{<}y \wedge V_0{=}(y-(z{-}1)^x)\}$
$\Rightarrow$
$\{x{=}X_0 \wedge y{=}Y_0 \wedge X_0{>}0 \wedge z{\geq}0 \wedge z^x{\leq}y \wedge (v{\leq}y \Rightarrow v{=}z^x) \wedge (v{>}y \Rightarrow v{=}(z+1)^x) \wedge \mathit{max}(0,(y-v)){<}V_0\}$
`else`
$\{x{=}X_0 \wedge y{=}Y_0 \wedge X_0{>}0 \wedge v{=}(z{+}1)^x \wedge z{\geq}0 \wedge z^x{<}y \wedge V_0{=}(y-z^x) \wedge v{>}y\}$
`skip`
$\{x{=}X_0 \wedge y{=}Y_0 \wedge X_0{>}0 \wedge v{=}(z{+}1)^x \wedge z{\geq}0 \wedge z^x{<}y \wedge V_0{=}(y-z^x) \wedge v{>}y\}$
$\Rightarrow$
$\{x{=}X_0 \wedge y{=}Y_0 \wedge X_0{>}0 \wedge z{\geq}0 \wedge z^x{\leq}y \wedge (v{\leq}y \Rightarrow v{=}z^x) \wedge (v{>}y \Rightarrow v{=}(z+1)^x) \wedge \mathit{max}(0,(y-v)){<}V_0\}$
`end`
$\{x{=}X_0 \wedge y{=}Y_0 \wedge X_0{>}0 \wedge z{\geq}0 \wedge z^x{\leq}y \wedge (v{\leq}y \Rightarrow v{=}z^x) \wedge (v{>}y \Rightarrow v{=}(z+1)^x) \wedge \mathit{max}(0,(y-v)){<}V_0\}$
`end`
$\{x{=}X_0 \wedge y{=}Y_0 \wedge X_0{>}0 \wedge z{\geq}0 \wedge z^x{\leq}y \wedge (v{\leq}y \Rightarrow v{=}z^x) \wedge (v{>}y \Rightarrow v{=}(z+1)^x) \wedge v{\geq}y\}$
$\Rightarrow$
$\{(v{=}Y_0 \wedge v{=}Z^{X_0} \wedge Z^{X_0}{\leq}Y_0 \wedge v{\geq}Y_0) \vee (v{>}Y_0 \wedge v{=}(z{+}1)^{X_0} \wedge z^{X_0}{\leq}Y_0)\}$
$\Rightarrow$
$\{z^{X_0}{\leq}Y_0 \wedge (z{+}1)^{X_0} > Y_0\}$

* *side-condition: this predicate implies* $\mathit{max}(0,(y-v)) \geq 0$

** *side-condition: this predicate implies* $(x{-}i) \geq 0$