G52DOA - Derivation of Algorithms
Exercises

Venanzio Capretta

Exercise 1: Derivations and Graph Models

For each of the following propositions, give either a derivation using Fitch-style natural deduction or a graph countermodel:

1. $\forall x, \forall y, \forall z, (\mathsf{E}(x,y) \wedge \mathsf{E}(x,z) \rightarrow \mathsf{E}(y,x)) \rightarrow \exists x, \mathsf{E}(x,x)$;
2. $\forall x, \forall y, \forall z, (\mathsf{E}(x,y) \wedge \mathsf{E}(x,z) \rightarrow y = z) \rightarrow$
   $\forall x, (\mathsf{E}(x,x) \rightarrow \forall y, (\neg x = y \rightarrow \neg \mathsf{E}(x,y)))$;
3. $\forall x, (\mathsf{G}(x) \rightarrow \exists y, (\mathsf{G}(y) \wedge \forall z, (\mathsf{E}(x,z) \rightarrow z = y))) \rightarrow$
   $\forall x, \forall y, (\mathsf{E}(x,y) \rightarrow \mathsf{G}(y))$;
4. $\forall x, \forall y, \forall z, (\mathsf{E}(x,y) \wedge \mathsf{E}(x,z) \rightarrow \mathsf{G}(y) \vee \mathsf{G}(z)) \rightarrow$
   $\forall x, (\neg \exists y, (\mathsf{E}(x,y) \wedge \neg \mathsf{G}(y)))$;
5. $\forall x, \forall y, (\neg \exists z, (\mathsf{E}(z,x) \wedge \mathsf{E}(z,y)) \rightarrow x = y) \rightarrow \forall x, \exists y, \mathsf{E}(x,y) \rightarrow$
   $\exists x, \forall y, \mathsf{E}(y,x)$;
6. $\forall x, \exists y, \mathsf{E}(x,y) \wedge \exists x, \exists y, \exists z, (E(x,y) \wedge \mathsf{E}(x,z) \wedge \neg y = z) \rightarrow$
   $\exists y, \exists u, \exists v, (\mathsf{E}(u,y) \wedge \mathsf{E}(v,y) \wedge \neg u = v)$.

Exercise 2: Hoare Logic

Use the proof rules for assignment and logical implication to show that the following Hoare triples are true:

1. $\{\mathsf{x} > 0\}\ \mathsf{y} := \mathsf{x} + 1\ \{\mathsf{y} > 1\}$;
2. $\{\top\}\ \mathsf{y} := \mathsf{x}; \mathsf{y} := \mathsf{x} + \mathsf{x} + \mathsf{y}\ \{\mathsf{y} = 3\mathsf{x}\}$;
3. $\{x > 1\}\ \mathsf{a} := 1; \mathsf{y} := \mathsf{x}; \mathsf{y} := \mathsf{y} - \mathsf{a}\ \{\mathsf{y} > 0 \wedge \mathsf{x} > \mathsf{y}\}$;
4. $\{\mathsf{x} = x_0 \wedge \mathsf{y} = y_0 \wedge \mathsf{z} = z_0 \wedge\}\ \mathsf{x} := \mathsf{y} + \mathsf{z}; \mathsf{y} := \mathsf{x} - \mathsf{z}; \mathsf{z} := \mathsf{x} - \mathsf{y}\ \{\mathsf{x} \geq \mathsf{y} \wedge \mathsf{x} \geq \mathsf{z} \wedge$
   $\mathsf{y} = y_0 \wedge \mathsf{z} = z_0\}$;
5. $\{z = z_0\}\ \mathsf{x} := \mathsf{x} - \mathsf{y}; \mathsf{y} := \mathsf{y} - \mathsf{z}; \mathsf{z} := \mathsf{z} - \mathsf{x}; \mathsf{x} := \mathsf{x} + \mathsf{y}; \mathsf{y} := \mathsf{y} + \mathsf{z}; \mathsf{z} := \mathsf{z} + \mathsf{x}$
   $\{2\mathsf{z} - \mathsf{x} - \mathsf{y} = z_0\}$.

Write a program P satisfying the given specifications (use only assignment statements and composition and use only addition and subtraction as operations) and write a Hoare logic derivation for the Hoare triple:

1. $\{\top\}\ P\ \{\mathsf{y} = \mathsf{x} + 2\}$;
2. $\{\top\}\ P\ \{\mathsf{z} > \mathsf{x} + \mathsf{y} + 4\}$;
3. $\{\mathsf{x} = x_0 \wedge \mathsf{y} = y_0 \wedge \mathsf{z} = \mathsf{x} \cdot \mathsf{y}\}\ P\ \{\mathsf{x} = x_0 + 1 \wedge \mathsf{y} = y_0 + 1 \wedge \mathsf{z} = \mathsf{x} \cdot \mathsf{y}\}$;
4. $\{\mathsf{x} = x_0 \wedge \mathsf{y} = \mathsf{x}^2 \wedge \mathsf{z} = \mathsf{x}^3\}\ P\ \{\mathsf{x} = x_0 + 1 \wedge \mathsf{y} = \mathsf{x}^2 \wedge \mathsf{z} = \mathsf{x}^3\}$.

Exercise 3

Find the weakest precondition of the following program fragment with respect to the given postcondition and simplify it as much as possible:

```
x := x + 1;
y := y + 4 · x;
x := x + 1;
```

$\{\mathsf{y} = \mathsf{x}^2\}$

Exercise 4

Find the weakest precondition of the following program fragment with respect to the given postcondition and simplify it as much as possible:

```
if y < x then
   u := z − y
else
   if y > z then
      u := y − x
   else
      u := z − x
```

$\{\mathsf{u} = \max(\mathsf{x}, \mathsf{y}, \mathsf{z}) - \min(\mathsf{x}, \mathsf{y}, \mathsf{z})\}$

Exercise 5

You are going to construct a program that computes an approximation to the logarithm of an input value x, that is, it must satisfy the following specification:

$$\{\mathsf{x} > 0\}\ p\ \{2^{\mathsf{y}} \leq \mathsf{x} < 2^{\mathsf{y}+1}\}$$

The program is not allowed to modify the value of x

(a) What is the invariant that you are going to use in your program?

(b) Write down the program and complete the proof tableau for its proof of correctness;

(c) What are the extra implications that you need to prove to complete the proof?

(d) What variant would you use to show the termination of the program?

Exercise 6

You are going to derive an algorithm that searches into an array a of length n the closest element to a given input x. The specification of the algorithm is the following Hoare triple:

$$\{\top\}\ p\ \{\forall i, 0 \le i \le \mathsf{n} \to |\mathsf{a}[\mathsf{k}] - \mathsf{x}| \le |\mathsf{a}[i] - \mathsf{x}|\}$$

(The post-condition states that the kth element of a is the closest one to x. The operator $|\cdot|$ gives the absolute value of its argument.)

(a) What invariant are you going to use in your algorithm?

(b) Write down the algorithm and complete the proof tableau for its correctness.

(c) What are the extra implications that you need to prove to complete the proof?

(d) What variant would you use to show the termination of the program?

Exercise 7

(a) Complete the following proof tableau:

$\{\mathsf{k} > 0 \land \forall j, 0 \le j < \mathsf{k} \to \mathsf{a}[j] = \binom{\mathsf{k}}{j}\}$

```
x := a[0];
a[0] := 1;
i := 1;
while i < k do (
   a[i] := x + a[i];
   x := a[i] − x
)
```

$\{\mathsf{k} > 0 \land \forall j, 0 \le j \le \mathsf{k} \to \mathsf{a}[j] = \binom{\mathsf{k}+1}{j}\}$

Remember that the notation $\binom{n}{m}$ denotes the binomial coefficient:

$$\binom{n}{m} = \frac{n!}{m! \cdot (n-m)!}$$

for $0 \le m \le n$ and that it satisfies the equalities:

$$\binom{n+1}{m+1} = \binom{n}{m} + \binom{n}{m+1} \qquad \binom{n}{0} = \binom{n}{n} = 1.$$

(b) Now write just $\mathsf{binomstep}(\mathsf{a}, \mathsf{k})$ for the program fragment in part (a). Use it to construct an algorithm that, given n, computes the binomial coefficients for degree n:

$$\{\mathsf{n} > 0\}\ p\ \{\forall j, 0 \leq j \leq \mathsf{n} \rightarrow \mathsf{a}[j] = \binom{\mathsf{n}}{j}\}$$

without changing the value of n.