Solutions to COMP9020 Problem Set 8

Kai Engelhardt

October 6, 2016

1 State Machines

Solution to Exercise 1 In C this could be the following, with absolutely no regard for efficiency — the compiler can take care of, for instance, eliminating the variable r.

```
#include <stdio.h>
#include <stdlib.h>

double fexp(double, unsigned long);

int main(int argc, char* argv[]) {
  if (argc == 3) {
    double a = strtod(argv[1], NULL);
    unsigned long b = strtoul(argv[2], NULL, 10);
    printf("%f\n", fexp(a,b));
    return EXIT_SUCCESS;
  } else
    return EXIT_FAILURE;
}

double fexp(double a, unsigned long b) {
  double x = a, y = 1;
  unsigned long z = b;

  while (z != 0) {
    unsigned long r = z % 2;
    z = z / 2;
    if (r == 1)
      y *= x;
    x *= x;
  }
  return y;
}
```

Timing is somewhat unconvincing because this is so fast that we hardly see any effects.

```
bash-3.2$ time ./fexp 3.1415926 2
9.869604

real 0m0.004s
```

```
user 0m0.001s
sys 0m0.002s
bash-3.2$ time ./fexp 3.1415926 4
97.409084

real 0m0.005s
user 0m0.001s
sys 0m0.002s
bash-3.2$ time ./fexp 3.1415926 8
9488.529721

real 0m0.005s
user 0m0.001s
sys 0m0.002s
bash-3.2$ time ./fexp 3.1415926 16
90032196.270391

real 0m0.004s
user 0m0.001s
sys 0m0.002s
bash-3.2$ time ./fexp 3.1415926 32
8105796365270132.000000

real 0m0.004s
user 0m0.001s
sys 0m0.002s
bash-3.2$ time ./fexp 3.1415926 64
65703934715226483612547435986944.000000

real 0m0.003s
user 0m0.001s
sys 0m0.002s
bash-3.2$ time ./fexp 3.1415926 128
4317007037062743694739503457778810722165267526429904568062050304.000000

real 0m0.004s
user 0m0.001s
sys 0m0.002s
bash-3.2$ time ./fexp 3.1415926 256
186365497580492493096957817280341286183659326201865618593954759087583932324077393390347106664

real 0m0.005s
user 0m0.001s
sys 0m0.002s
bash-3.2$ time ./fexp 3.1415926 512
347320986884245510254782833283848876954179261505017778554709406978733617473703252549909516855

real 0m0.004s
user 0m0.001s
sys 0m0.002s
bash-3.2$ time ./fexp 3.1415926 1024
```

```
inf

real 0m0.005s
user 0m0.001s
sys 0m0.002s
bash-3.2$ time ./fexp 3.1415926 1
3.141593

real 0m0.004s
user 0m0.001s
sys 0m0.002s
bash-3.2$ time ./fexp 3.1415926 3
31.006275

real 0m0.004s
user 0m0.001s
sys 0m0.002s
bash-3.2$ time ./fexp 3.1415926 7
3020.292867

real 0m0.005s
user 0m0.001s
sys 0m0.002s
bash-3.2$ time ./fexp 3.1415926 15
28658138.636560

real 0m0.004s
user 0m0.001s
sys 0m0.002s
bash-3.2$ time ./fexp 3.1415926 31
2580155162470822.500000

real 0m0.005s
user 0m0.001s
sys 0m0.002s
bash-3.2$ time ./fexp 3.1415926 63
20914212337788959751588271882240.000000

real 0m0.005s
user 0m0.001s
sys 0m0.002s
bash-3.2$ time ./fexp 3.1415926 127
137414604206247012028112260012716683619360292321972921456970956 8.000000

real 0m0.005s
user 0m0.002s
sys 0m0.002s
bash-3.2$ time ./fexp 3.1415926 255
59321981335356004660115835867455828378782809474274933310138929069098498921280252155296496746

real 0m0.003s
```

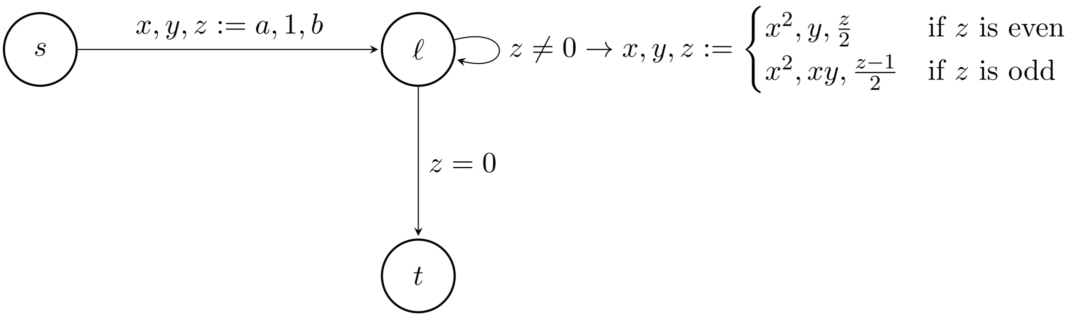
```
user 0m0.001s
sys 0m0.002s
bash-3.2$ time ./fexp 3.1415926 511
11055570569024310636753288902698382576023471794520852903470160359651741896020034159061300045

real 0m0.004s
user 0m0.001s
sys 0m0.002s
bash-3.2$ time ./fexp 3.1415926 1023
inf

real 0m0.006s
user 0m0.002s
sys 0m0.002s
bash-3.2$
```

The next four answers presume familarity with slides 8a on transition diagrams.

Solution to Exercise 2 There a numerous ways to translate such a program into a transition diagram. My choice is guided by the desire to have a node where the loop invariant holds but not many more, if I can avoid it.



Solution to Exercise 3 Precondition $\phi \stackrel{\text{def}}{=} a \in \mathbb{R} \wedge b \in \mathbb{N}$; postcondition $\psi \stackrel{\text{def}}{=} y = a^b$.

Solution to Exercise 4 Define the assertion network Q by:

$$Q_s \stackrel{\text{def}}{=} \phi$$
$$Q_\ell \stackrel{\text{def}}{=} z \in \mathbb{N} \wedge yx^z = a^b$$
$$Q_t \stackrel{\text{def}}{=} \psi$$

To prove that Q is inductive we check the three verification conditions relating to the transitions—the two relating to pre- and postconditions are vacuously true due to the definition of Q_s and Q_t.

$s \to \ell$**:** We need to show that

$$Q_s \Rightarrow Q_\ell \circ [x, y, z := a, 1, b]$$

is valid. This is equivalent to checking that $a \in \mathbb{R} \wedge b \in \mathbb{N} \Rightarrow b \in \mathbb{N} \wedge 1 \cdot a^b = a^b$, which in turn is immediate.

$\ell \to \ell$**:** We need to show that

$$Q_s \wedge z \neq 0 \Rightarrow Q_\ell \circ \left[x, y, z := \begin{cases} x^2, y, \frac{z}{2} & \text{if } z \text{ is even} \\ x^2, xy, \frac{z-1}{2} & \text{if } z \text{ is odd} \end{cases} \right]$$

is valid. We split this into two cases.

$$Q_\ell \wedge z \neq 0 \wedge 2|z \Rightarrow Q_\ell \circ \left[x, y, z := x^2, y, \frac{z}{2}\right] \tag{1}$$

$$Q_\ell \wedge z \neq 0 \wedge 2 \not| z \Rightarrow Q_\ell \circ \left[x, y, z := x^2, xy, \frac{z-1}{2}\right] \tag{2}$$

To show validity of (1) we prove

$$z \in \mathbb{N} \wedge yx^z = a^b \wedge z \neq 0 \wedge 2|z \Rightarrow (z \in \mathbb{N} \wedge yx^z = a^b) \circ \left[x, y, z := x^2, y, \frac{z}{2}\right]$$

which we attack using that, for a predicate $\phi : \Sigma \longrightarrow \mathbb{B}$ and state update function $f : \Sigma \longrightarrow \Sigma$ given as (the meaning of) an assignment statement $[x := e]$, the predicate $\phi \circ f$ is equivalent to $\phi[^e/_x]$, that is, ϕ with all free occurrences of x replaced by e.

$$z \in \mathbb{N} \wedge yx^z = a^b \wedge z \neq 0 \wedge 2|z \Rightarrow \frac{z}{2} \in \mathbb{N} \wedge y(x^2)^{\frac{z}{2}} = a^b \ ,$$

which follows by simple math. Similarly, to show (2), we inspect

$$z \in \mathbb{N} \wedge yx^z = a^b \wedge z \neq 0 \wedge 2 \not| z \Rightarrow \frac{z-1}{2} \in \mathbb{N} \wedge xy(x^2)^{\frac{z-1}{2}} = a^b$$

The crucial bit here is that $x^z = x(x^2)^{\frac{z-1}{2}}$ when $z \in \mathbb{N}$ is odd.

$\ell \to t$: We need to show that

$$Q_\ell \wedge z = 0 \Rightarrow Q_t$$

is valid. This is equivalent to checking that $z \in \mathbb{N} \wedge yx^z = a^b \wedge z = 0 \Rightarrow y = a^b$, which again is immediate.

Solution to Exercise 5 Define the ranking functions

$$\begin{aligned} \rho_s &= (2, 0) \\ \rho_\ell &= (1, \lceil \log_2 z \rceil + 1) \\ \rho_t &= (0, 0) \end{aligned}$$

and use lexicographic ordering on these pairs. The first component of the ranking functions ensures that the two transitions $s \to \ell$ and $\ell \to t$ trivially satisfy the verification condition. The only interesting transition here is

$\ell \to \ell$: for which we need to show that

$$Q_\ell \wedge z \neq 0 \Rightarrow \rho_\ell \circ \left[x, y, z := \begin{cases} x^2, y, \frac{z}{2} & \text{if } z \text{ is even} \\ x^2, xy, \frac{z-1}{2} & \text{if } z \text{ is odd} \end{cases}\right] <_{\text{lex}} \rho_\ell$$

which boils down to proving that $\lceil \log_2 \frac{z}{2} \rceil < \lceil \log_2 z \rceil$ if $z \in \mathbb{N}_{>0}$ is even, and $\lceil \log_2 \frac{z-1}{2} \rceil < \lceil \log_2 z \rceil$ if $z \in \mathbb{N}_{>0}$ is odd. Both follow from elementary properties (namely $\log_2(2k) = 1 + \log_2 k$ and $\log_2(2k+1) > 1 + \log_2 k$) of $\log_2$.

Solution to Exercise 6 This is trivial because the only guards in the transition diagrams are those on the transitions starting at location ℓ, and the disjunction $z \neq 0 \vee z = 0$ of the two guards is a valid proposition.