CS 620 – Theory of Computation – Fall 2009

Instructor: Marc Pomplun

Assignment #1

Sample Solutions

Question 1:

Consider the following predicate $P(x, y)$:

$$P(x,y) = \begin{cases} 1 & \text{if} \quad x \geq 2y + 3 \\ 0 & \text{otherwise} \end{cases}$$

a) Write a program in the language $\mathcal{L}$ that computes $P(x, y)$. You cannot use any macros at all.

b) Write down the complete list of snapshots that occur during the computation of $P(5, 2)$. Use the format shown on page 29 in the textbook.

Answer a): The following program computes $f(X_1, X_2)$, which is equivalent to P(x, y):

```
(1)           Z1 ← Z1 + 1
(2)           Z1 ← Z1 + 1
(3)           Z1 ← Z1 + 1
(4)           if X2≠0 GOTO A
(5)           if Z1≠0 GOTO B
(6)     [A]   X2 ← X2 - 1
(7)           Z1 ← Z1 + 1
(8)           Z1 ← Z1 + 1
(9)           if X2≠0 GOTO A
(10)    [B]   if X1≠0 GOTO C
(11)          if Z1≠0 GOTO E
(12)    [C]   X1 ← X1 - 1
(13)          Z1 ← Z1 - 1
(14)          if X1≠0 GOTO C
(15)          if Z1≠0 GOTO E
(16)          Y ← Y + 1
```

Answer b):

(1, {$X_1 = 5$, $X_2 = 2$, $Y = 0$, $Z_1 = 0$})
(2, {$X_1 = 5$, $X_2 = 2$, $Y = 0$, $Z_1 = 1$})
(3, {$X_1 = 5$, $X_2 = 2$, $Y = 0$, $Z_1 = 2$})
(4, {$X_1 = 5$, $X_2 = 2$, $Y = 0$, $Z_1 = 3$})
(6, {$X_1 = 5$, $X_2 = 2$, $Y = 0$, $Z_1 = 3$})
(7, {$X_1 = 5$, $X_2 = 1$, $Y = 0$, $Z_1 = 3$})
(8, {$X_1 = 5$, $X_2 = 1$, $Y = 0$, $Z_1 = 4$})
(9, {$X_1 = 5$, $X_2 = 1$, $Y = 0$, $Z_1 = 5$})
(6, {$X_1 = 5$, $X_2 = 1$, $Y = 0$, $Z_1 = 5$})
(7, {$X_1 = 5$, $X_2 = 0$, $Y = 0$, $Z_1 = 5$})
(8, {$X_1 = 5$, $X_2 = 0$, $Y = 0$, $Z_1 = 6$})
(9, {$X_1 = 5$, $X_2 = 0$, $Y = 0$, $Z_1 = 7$})
(10, {$X_1 = 5$, $X_2 = 0$, $Y = 0$, $Z_1 = 7$})
(12, {$X_1 = 5$, $X_2 = 0$, $Y = 0$, $Z_1 = 7$})
(13, {$X_1 = 4$, $X_2 = 0$, $Y = 0$, $Z_1 = 7$})
(14, {$X_1 = 4$, $X_2 = 0$, $Y = 0$, $Z_1 = 6$})
(12, {$X_1 = 4$, $X_2 = 0$, $Y = 0$, $Z_1 = 6$})
(13, {$X_1 = 3$, $X_2 = 0$, $Y = 0$, $Z_1 = 6$})
(14, {$X_1 = 3$, $X_2 = 0$, $Y = 0$, $Z_1 = 5$})
(12, {$X_1 = 3$, $X_2 = 0$, $Y = 0$, $Z_1 = 5$})
(13, {$X_1 = 2$, $X_2 = 0$, $Y = 0$, $Z_1 = 5$})
(14, {$X_1 = 2$, $X_2 = 0$, $Y = 0$, $Z_1 = 4$})
(12, {$X_1 = 2$, $X_2 = 0$, $Y = 0$, $Z_1 = 4$})
(13, {$X_1 = 1$, $X_2 = 0$, $Y = 0$, $Z_1 = 4$})
(14, {$X_1 = 1$, $X_2 = 0$, $Y = 0$, $Z_1 = 3$})
(12, {$X_1 = 1$, $X_2 = 0$, $Y = 0$, $Z_1 = 3$})
(13, {$X_1 = 0$, $X_2 = 0$, $Y = 0$, $Z_1 = 3$})
(14, {$X_1 = 0$, $X_2 = 0$, $Y = 0$, $Z_1 = 2$})
(15, {$X_1 = 0$, $X_2 = 0$, $Y = 0$, $Z_1 = 2$})
Output: 0

Question 2:

Now you want to write a program in your favorite language $\mathcal{L}$ that computes the square root function $f(x) = \sqrt{x}$. Remember that we only consider natural numbers; for example, $f(49) = 7$, but $f(17) = \uparrow$.

a) Write down the program using macros. You are allowed to use the macros $V \leftarrow V'$, $V \leftarrow m$, GOTO L, and $W \leftarrow f(V_1, \ldots, V_n)$. Use at least one macro of the form $W \leftarrow f(V_1, \ldots, V_n)$, and whenever you do so, provide a program that computes $f(V_1, \ldots, V_n)$.

b) In your program, expand all the macros of the form $W \leftarrow f(V_1, \ldots, V_n)$ using the convention given in Section 2.5 in the textbook. Write down the expanded version of your program, which should now contain no macros except (possibly) those of the forms $V \leftarrow V'$, $V \leftarrow m$, and GOTO L

Answer a):

```
        IF X≠0 GOTO A
        GOTO E
[A]     Z1 ← X
        Y ← Y + 1
        Z2 ← SQ(Y)    // SQ(Y) = Y·Y
[B]     Z1 ← Z1 - 1
        Z2 ← Z2 - 1
        IF Z2≠0 GOTO C
        IF Z1≠0 GOTO A
        GOTO E
[C]     IF Z1≠0 GOTO B
        GOTO C
```

Macro SQ(X) = X·X:

```
        IF X≠0 GOTO A
        GOTO E
[A]     Z1 ← X
[B]     Z2 ← X
[C]     Z2 ← Z2 - 1
        Y ← Y + 1
        IF Z2≠0 GOTO C
        Z1 ← Z1 - 1
        IF Z1≠0 GOTO B
```

Answer b):

```
        IF X≠0 GOTO A
        GOTO E
[A]     Z1 ← X
        Y ← Y + 1
        Z3 ← 0
        Z4 ← Y
        Z5 ← 0
        Z6 ← 0
        IF Z4≠0 GOTO A4
        GOTO E3
[A4]    Z5 ← Z4
[A5]    Z6 ← Z4
[A6]    Z6 ← Z6 - 1
        Z3 ← Z3 + 1
        IF Z6≠0 GOTO A6
        Z5 ← Z5 - 1
        IF Z5≠0 GOTO A5
[E3]    Z2 ← Z3
[B]     Z1 ← Z1 - 1
        Z2 ← Z2 - 1
        IF Z2≠0 GOTO C
        IF Z1≠0 GOTO A
        GOTO E
[C]     IF Z1≠0 GOTO B
        GOTO C
```

Question 3:

Give a detailed argument that the following functions are primitive recursive:

a) x^y

b) the predecessor function $p(x)$

These arguments should be of the same form as those provided in Examples 1, 2, and 3 of Section 3.4 in the textbook.

a) $h(x, y) = x^y$

Step 1: Write recursive equations

$h(x, 0) = 1$
$h(x, y+1) = h(x, y) \cdot x$

Step 2: Write recursive equations in terms of initial functions

$h(x, 0) = s(n(x))$
$h(x, y+1) = g(y, h(x,y), x)$
where
$g(x_1, x_2, x_3) = u_2^3(x_1, x_2, x_3) \cdot u_3^3(x_1, x_2, x_3)$

b) $h(x) = p(x)$, 'the predecessor function'

Step 1: Write recursive equations

$h(0) = 0$
$h(t+1) = t$

Step 2: Write recursive equations in terms of initial functions

$h(0) = n(t)$
$h(t+1) = g(t, h(t))$
where
$g(x_1, x_2) = u_1^2(x_1, x_2)$

Question 4:

Show that the class of all total functions is a PRC class.

Definition of a total function (from page 30 of the text):
A function g of m variables is called total if $g(r_1,..., r_m)$ is defined for all $r_1,..., r_m$.

Let $\mathcal{T}$ be the class of all total functions.
On the basis of the Definition on page 42 of the text, $\mathcal{T}$ is called a PRC class if
1. The initial functions belong to $\mathcal{T}$.
2. A function obtained from functions belonging to $\mathcal{T}$ by either composition or recursion also belongs to $\mathcal{T}$.

Requirement 1 is fulfilled, because all three initial functions are total functions, so they belong to $\mathcal{T}$.

Requirement 2 is also fulfilled, and we will show this separately for composition and recursion:

Composition: Let $h(x_1,..., x_n) = f(g_1(x_1,..., x_n),..., g_k(x_1,..., x_n))$, where f and the functions g_i are members of $\mathcal{T}$. The domain of h is the domain of all g_i. Because all functions g_i are total, they will yield k input values to f for any given $x_1,..., x_n$. Since f is total, it is defined for any $g_1,..., g_k$. Therefore, h is defined for all $x_1,..., x_n$, which means that h is total.

Recursion: Let
$h(x_1,..., x_n, 0) = f(x_1,..., x_n)$
$h(x_1,..., x_n, t + 1) = g(t, h(x_1,..., x_n, t), x_1,..., x_n)$,
where f and g are in $\mathcal{T}$. Then we can show by induction that h is total:
Basis step: For t = 0, $h(x_1,..., x_n, 0)$ is defined.
Induction step: Suppose that $h(x_1,..., x_n, t)$ is defined. Then, because g is total,
$g(t, h(x_1,..., x_n, t), x_1,..., x_n)$ is defined, and therefore, $h(x_1,..., x_n, t + 1)$ is also defined.
Conclusion: h is total.

Question 5:

Let $f(x)$ be the sum of the divisors of x if $x \neq 0$; $f(0) = 0$. For example, $f(6) = 1 + 2 + 3 + 6 = 12$. Show that $f(x)$ is primitive recursive.

f(x) can be written as

$f(x) = \sum_{t \leq x} (t \mid x) \cdot t$

We know from the theorems in the textbook that the operations used in this equation are primitive recursive. Since we use composition of these functions to obtain f, f must also be primitive recursive.

Question 6:

Let $u(n)$ be the n-th number in order of size which is the sum of two squares. Show that $u(n)$ is primitive recursive.

u(n) can be defined as follows:

$u(0) = 0$
$u(n+1) = \min_{t \leq n^2} [(\exists x)_{\leq t}(\exists y)_{\leq t}(x^2 + y^2 = t) \;\&\; (t > u(n) \vee n = 0)]$

We know from the theorems in the textbook that the operations used in this equation are primitive recursive. Since we apply composition and recursion to these functions to obtain u, u must also be primitive recursive.

However, we still have to show that $u(n+1) \leq n^2$ for all n.

Let us take a look at the sequence s(n) of perfect squares:

$s(0) = 0$
$s(1) = 0^2 = 0$
$s(2) = 1^2 = 1$
$s(3) = 2^2 = 4$
$s(4) = 3^2 = 9$
.
.
.

Obviously, we have $s(n+1) \leq n^2$ for all n. Now if we consider u(n), it contains all the numbers in s(n), but also additional ones that need to be listed (such as $1^2 + 1^2 = 2$), so with larger n the values in the sequence u(n) increase more slowly than those in s(n).

Therefore, we have $u(n) \leq s(n)$ for all n, and since $s(n+1) \leq n^2$, we finally get

$u(n+1) \leq n^2$.