Formal Model and Verification

Exercise 8: Symbolic preconditions of rules

1. Please compute the following preconditions.

(a) pre(x=3y+z; , x < y+3 && y>2)

(b) pre(x = y+3*x; , x < y+3 && y > 2)

(c) pre(x = x+z+3; , x < y+3 && y > 2)

2. Please compute the following preconditions.

(a) pre(if (x <= z && z > 8) x = x+1; else z = 3; , x < y+3 && y > 2)

(b) pre(y = 3*z+x; x = x+1; z = 3; , x < y+3 && y > 2)

(c) pre(if (x <= z && z > 8) x = x+1; else z = 3;
y = 3*z+x;
x = x+1;
z = 3;,
x < y+3 && y > 2
)

3. Please compute the following preconditions.

(a) pre(while (x >= 3 && y < 8) { x = x+1; y = y-1; }, x < 3 && y > 2)

(b) pre(while (x >= y && y < 8) y = y+1;, x > 3 && y > 2)

4. Given a set of rules r_1, r_2, …, r_m of the form

pre(while (K) s, b)

L_2 represents those states that never leave K with steps of s. In the class, we have the following two algorithms for computing L_2.

```
w0 = K; k = 1;
repeat
    wk = K∧pre(s, wk-1);
    k = k +1;
  until wk ≡ wk-1;
return wk;
```

```
w0 = K; k = 1;
repeat
    wk = wk-1∧pre(s, wk-1);
    k = k +1;
  until wk ≡ wk-1;
return wk;
```

Please prove that the two algorithms are equivalent, that is, the two algorithms return formulas that represent the same state space.

5. Continued from problem 4, please answer the following two questions.

(a) Please write down an example verification task for which calculating L_2 is needed.

(b) Please write down an example verification task for which calculating L_2 is not needed.