Rafael Fenner Ortiz

Exercises for Analysis of Distributed and Concurrent Systems

Resource Analysis of Distributed and Concurrent Systems: Lecture 2

Exercise 1: Consider the following ABS code:

```
Unit main (Int a, Int b) {
  C c = new C();
  c!m(a,b);
}
```

```
class C {
  Unit m (Int a, Int b) {
    while (a > 0) {
      this!p(b);
      a = a - 1;
    }
  }
  Unit p (Int b) {
    while (b < 10) {
      b = b + 1;
    }
  }
}
```

Give the ABS code above, please, write the **cost relation system** of this ABS code. Additionally, write the **upper bound** of the program.

(1) while (a > 0) this!p(b); a = a - 1;

(2) while (b < 10) b = b + 1;

In order to know how much (2) cost, we have to know if variable b is modified by external process. In this code doesn't occurs, therefore (2) cost:

$$\text{while}(b<10) = k_1 \quad \{\, b \geq 10 \,\}$$
$$\text{while}(b<10) = k_2 + \text{while}(b+1) \quad \{\, b < 10 \,\}$$

Let us consider $dist : (\mathbb{Z} \times \mathbb{Z}) \longrightarrow \mathbb{N}$ the distance function, so the cost of while(b+1) is $d(b+1, 10)$. Therfore.

$$(z_1, z_2) \longrightarrow |z_1 - z_2|$$

$$\text{while}(b<10) = k_1 \quad \{\, b \geq 10 \,\}$$
$$\text{while}(b<10) = k_2 \cdot |b+1-10| \quad \{\, b < 10 \,\}$$
$$\Rightarrow \text{while}(b<10) = k_1 + k_2 \cdot |b+1-10|$$

On the other hand, we have another while loop that contains the above loop (2) nested. If we analyse the loop perse as follow:

while (a>0) this!p(b); a = a-1;

We obtain:

(1.1) while $(a>0) = k_3 \quad \forall\, a \leq 0 \,\forall$

(1.2) while $(a>0) = k_4 + k_5 + \text{while}(a-1) \quad \forall\, a > 0 \,\forall$

where k_5 = cost of (2). Therefore, we obtain that (1):

$\text{while}(a>0) = k_3 \quad \forall\, a \leq 0 \,\forall$

$\text{while}(a>0) = k_4 + \text{while}(b) + \text{while}(a-1) \quad \forall\, a > 0 \,\forall \iff$

$\iff \text{while}(a>0) = k_4 + \left(k_1 + k_2 + \text{while}(b-1)\right) + \text{while}(a-1) \quad \forall\, a > 0 \,\forall \iff$ Analogous to (2)

$\iff \text{while}(a>0) = \left(k_4 + \left(k_1 + k_2 \cdot |b+1-10|\right)\right) \cdot \text{dist}(0, a-1) \quad \forall\, a > 0 \,\forall \iff$

$\iff \text{while}(a>0) = \left(k_4 + \left(k_1 + k_2 \cdot |b+1-10|\right)\right) \cdot |a-1| \quad \forall\, a > 0 \,\forall$

$\Rightarrow \text{while}(a>0) = k_3 + |a-1| \cdot \left(k_4 + k_1 + k_2 \cdot |b+1-10|\right) \quad \forall\, a > 0 \,\forall$

(1.1), (1.2) $\Rightarrow$ $\underline{\text{while}(a>0) = k_3 + |a-1| \cdot \left(k_4 + k_1 + k_2 \cdot |b+1-10|\right)}$

$\Rightarrow \text{main}(a, b) = k_0 + k_3 + |a-1| \cdot \left(k_4 + k_1 + k_2 \cdot |b+1-10|\right)$

Exercise 2: Consider the following ABS code:

```
1 Unit main (Int a, Int b) {
2   C c = new C();
3   C c2 = new C();
4   c!m(a,b);
5   c2!p(b);
6 }
```

```
7  class C {
8    Unit m (Int a, Int b) {
9      while (a > 0) {
10       this!p(b);
11       a = a - 1;
12     }
13   }
14   Unit p (Int b) {
15     while (b < 10) {
16       b = b + 1;
17     }
18   }
19 }
```

Give the ABS code above, please, write the **object sensitive cost relation system** of this ABS code. Additionally, write the **upper bound** of the program including the corresponding **cost centers**.

We can observe main code has two diffents instances of C, so we could consider that our progams has three cost-center.

Let us consider the code above, we know that for points-to analysis, we have o_2, o_3 where $o_2 \in A(4, main)$ and $o_3 \in A(5, main)$ also o_{main}. By exercise ① we know that

$cost(o_2) = \underbrace{k_3 + |a-1| \cdot (k_4 + k_1 + k_2 \cdot |b+1-10| + k_p)}_{while} + k_m$; and

$cost(o_3) = k_1 + k_2 \cdot |b+1-10| + k_p$

$\Rightarrow U = U_m + U_{o_2} + U_{o_3} \Leftrightarrow$

$$U = C(o_{main}) \cdot k_{main} + C(o_2) \cdot (k_3 + |a-1| \cdot (k_p + k_4 + k_1 + k_2 \cdot |b+1-10|) + k_m) + C(o_3) \cdot (k_p + k_4 + k_1 + k_2 \cdot |b+1-10|)$$

Exercise 3: Consider the following ABS code:

```
1 main (int n) {
2   C z = new C();  // m1
3   C y = new C();  // m1
4   f = y!p(n);     // m1
5   await f?;       // m2
6   ...             // m3
7   y!p(n);         // m3
8   ...             // m4
9   y!q(z,n);       // m4
10  ...             // m5
11 }
```

```
class C {
  void p (Int n) {
    ...           // p1
  }
  void q (C z, Int n) {
    z!s(n);       // q1
    ...           // q2
  }
  void s (Int n) {
    ...           // s1
  }
}
```

Draw the **distributed flow graph** of the program and compute all possible paths to *terminal nodes* from the entry node (m_1).