# TD 7

## Test Structurel

**Bu 26.11.20**

- Exo 1:
  - Code

```
boolean comp_occurrences(char s[], char a, char b) {
    int res = 0;
    int i = 0;
    while(i < s.length) {
        if(s[i] == a) { res++; }
        if(s[i] == b) { res--; }
        i++;
    }
    return (res > 0);
}
```

**Figure 1:** Code comp_occurrences

  - CFG

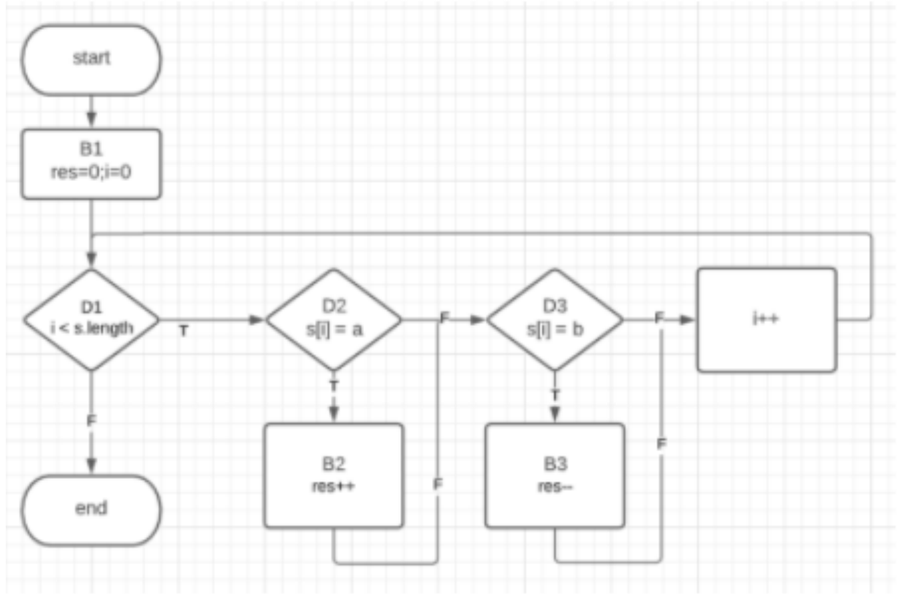


**Figure 2:** CFG comp-resoccurrences

- Exo 1:
  - Specification:

pre (s,a,b) ≡ true

post(s,a,b,result) ≡ ( |[x : s | x = a]| > |[x : s | x = b]| )

  - Coverage Criterion:

AllInstructions(CFG) =
{ [start, B1, D1,D2,B2,D3,B3,B4,D1, end ]}

  - Exec Symbolique:

| | | s | a | b | res | i |
|---|---|---|---|---|---|---|
| start, | true | $s_0$ | $a_0$ | $b_0$ | $res_0$ | $i_0$ |
| B1, | true | $s_0$ | $a_0$ | $b_0$ | 0 | 0 |
| D1, | $(i < s.length)[i \mapsto 0, s \mapsto s_0]$ | | " | | | |
| D2, | $0<s_0.length \wedge (s[0] = a)\ [i \mapsto 0, s \mapsto s_0]$ | | " | | | |
| B2, | $0<s_0.length \wedge (s_0[0] = a_0)$ | $s_0$ | $a_0$ | $b_0$ | 1 | 0 |
| D3, | $0<s_0.length \wedge s_0[0] = a_0 \wedge s_0[0] = b_0$ | | | | | |
| B3, | " | $s_0$ | $a_0$ | $b_0$ | 0 | 0 |
| B4, | " | $s_0$ | $a_0$ | $b_0$ | 0 | 1 |
| D1, | " $\wedge$ $1 \geqslant s_0.length$ | | " | | | |
| end | " | | " | | | |

result = false

- Exo 1:
  - Condition de Chemin: $0<s_0.length \wedge s_0[0] = a_0 \wedge s_0[0] = b_0 \wedge \wedge 1 \geqslant s_0.length$

    Faisable ? Oui si a = b

  - Jeu de Test: s = ['z'] , a = 'z' , b = ‘z’ resultat attendu : false

  - Verdikt: resultat du programme: false, ce qui correspond a la spec. Donc programme correct pour ce cas.

- Exo 2:
  - Programme:

```
boolean palindrome(char[] s, int n) {
  int i = 0;
  boolean b = true;
  while (i < (n div 2) && b)  {
    if (s[i] == s[n-i-1]) {
      i = i+1;
    } else {
      b = false;
    }
  }
  return b;
}
```

**Figure 3:** code "palindrome"

  - Specification:

definition $pre_{palindrome}(s, a) \equiv$ $n \leqslant s.length$ || $valid(s,n)$

definition $post_{palindrome}(s, a, r) \equiv$ $(r = (\forall\ i\ .\ i \leqslant s.length / 2 \longrightarrow s[i] = s[s.length-i-1]))$

- Exo 1:
  - CFG

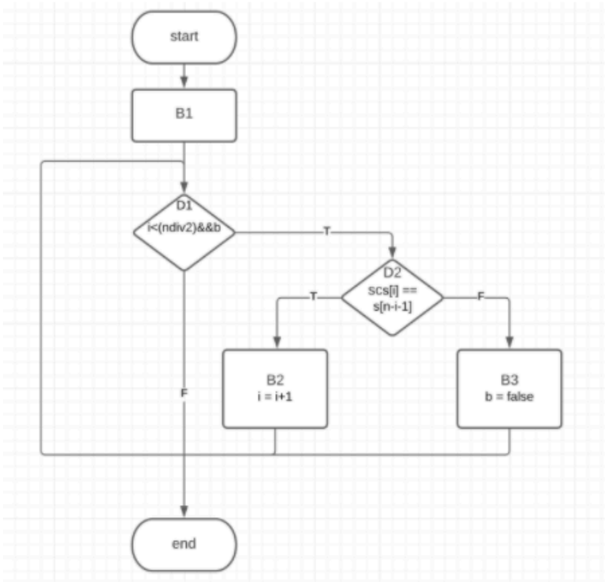


**Figure 4:** CFG palindrome

  - Dailleurs:
    - Combien de chemins est-ce qu'il y a dans AllPath(2)(CFG), donc |AllPath(2)(CFG)| = ??? (réponse: 1+2+4)
    - Combien des chemins dans le cas générale: (réponse: 2^(n+1)-1)

- Exo 1:
  - Execution symbolique de [Start, B1, D1, D2, B3, D1, D2, B2, D1, End]

| | CondChemin | i | b | result |
|---|---|---|---|---|
| Start | $n \leqslant s.length$ | $i_0$ | $b_0$ | $result_0$ |
| B1 | " | 0 | true | " |
| D1 | " $\wedge$ $0 < (n_0 \text{ div } 2)$ | " | " | " |
| D2 | " $\wedge$ $s_0[0] <> s_0[n_0-1]$ | " | " | " |
| B3 | " | " | false | " |
| D1 | " $\wedge$ $i_0<(n_0 \text{ div } 2)$&&false | " | " | " |
| D2 | false .... | | | |
| ... | false | | | |
| ... | false | | | |

  - La condition de chemin va etre: false. Cela veut dire: le chemin est infaisable. Donc il n'y a pas de tests pour ce cas (le cas est "vide").

  - On cherche un autre chemin pour AllTransitions, faisable cette fois ci.

- Exo 1:
  - Execution symbolique de [Start, B1, D1, D2, B2, D1, D2, B3, D1, End]

| | CondChemin | i | b | result |
|---|---|---|---|---|
| Start | $n_0 \leqslant s_0.length$ | $i_0$ | $b_0$ | $result_0$ |
| B1 | " | 0 | true | " |
| D1 | " $\wedge$ $0 < (n_0 \text{ div } 2)$ | " | " | " |
| D2 | " $\wedge$ $s_0[0] = s_0[n_0-1]$ | " | " | " |
| B2 | " | 1 | " | " |
| D1 | " $\wedge$ $1<(n_0 \text{ div } 2)$ | " | " | " |
| D2 | " $\wedge$ $s_0[1] <> s_0[n_0-2]$ | " | " | " |
| B3 | " | " | false | " |
| D1 | " $\wedge$ $\neg(1<(n_0 \text{ div } 2) \wedge \text{false})$ | " | " | " |
| end | | | | |

  - Condition de Chemin:
  - Faisable cette fois ci.

$$n_0 \leqslant s_0.length \wedge 0 < (n_0 \text{ div } 2) \wedge s_0[0] = s_0[n_0-1] \wedge 1<(n_0 \text{ div } 2) \wedge s_0[1] <> s_0[n_0-2] \wedge \neg(1<(n_0 \text{ div } 2) \wedge \text{false})$$

- Exo 1:
  - Jeu de test:

n = 4
s = ['a', '1', '2', 'a']
result expected false,

On note que n=4, s=[‘a’,’1’,’2’,’a’,’x’,’y’] est egalement possible est a le meme chemin. (Pourquoi ?)

  - Verdikt: Le programme est conforme a la spec dans le cas:

[start,B1,D1,D2,B2,D1,D2,B3,D1,end]