Université de Yaoundé I
The University of Yaounde I

Faculté des Sciences
Faculty of Sciences

Département d'Informatique

INFO304 (Langage Formel et Compilation) : Fiche de TD N°1

19 avril 2021

Etienne Kouokam

Exercice 1: Erreurs de compilation.
On considère les programmes suivants :

```
class A {
int x?;
public static void main(){
}
}
```

```
class B {
    int x;
    public static void main(){
    String s = "bonjour;
    System.out.println(s);
  }
}
```

```
class C {
  public static void main(){
    int y=3;
    System.out.println(x+y);
  }
}
```

```
class D {
  static void m(int x){
    System.out.println(x+1);
  }
  public static void main(){
    int y=3;
    m(y=y+4);
    m(y==y+4);
  }
}
```

```
class E {
  static int m(int x){
    if (x>0) return 1;
    else
      if (x<=0) return 2;
      else return 3;
    return 4;
  }
  public static void main(){
  m(42);
  }
}
```

```
class F {
  static int m(int x){
    return (x+1);
  }
  public static void main(){
    int final=42;
    m(final+1);
  }
}
```

```
class G {
  public static void main(){
    short s = 35000;
    System.out.println(s);
  }
}
```

```
class H {
static int m(int x, boolean b,
                    double f){
return b?x+(int)f:3;
}
public static void main(){
m(4.0,true,3.14);
}
}
```

```
class I {
  static int m(int x){
    final int y=x+1;
    y++;
    return y;
       }
  public static void main(){
    m(4);
  }
}
```

1.1 Dire si les programmes du tableau ci-dessus sont corrects. Quand un programme ne l'est pas, indiquer à quel moment de la compilation l'erreur est détectée.

1.2 Dire à quelle phase de la compilation on peut détecter les erreurs suivantes ?
a. Identificateur mal formé : 12K3 en C, en CAML ? **b.** Conflit de type sin('a') **c.** Instruction non atteignable **d.** Variable non déclarée **e.** Commentaire non fermé **f.** Parenthèse non fermée **g.** BEGIN non fermé **h.** Mot clé utilisé comme identificateur **i.** Non conformité entre le nombre de paramètres de définition et d'appel d'une procédure. **j.** Tentative de modifier une constante

Exercice 2: Notion de longueur.

a. Soit $\Sigma = \{a, b\}$ et les langages $L = \{abb, b, a, \epsilon\}$ et $L' = \{ba, baa\}$.
Calculer les langages suivants : $L \cup L'$, $L \cap L'$, LL', $L'L$, L^0, L^2.

b. Étant donné un alphabet Σ, définir par induction sur la structure des mots la fonction $|w|_a$ qui calcule le nombre d'occurrences d'une lettre $a \in \Sigma$ dans le mot $w \in \Sigma^*$.

c. Vérifier que : $|u.v| = |u| + |v|$; $|u.v|_a = |u|_a + |v|_a$.

Exercice 3: Propriétés de l'itération.

a. Terminer la démonstration des propriétés 8) de l'itération, c'est-à-dire, montrer que, quels que soient $L, M \subseteq \Sigma^*$:

1. $(L + M)^* \subseteq (L^* + M^*)^*$
2. $(L^* + M^*)^* \subseteq (L^*M^*)^*$
3. $(L^*M^*)^* \subseteq L^*(ML^*)^*$
4. $L^*(ML^*)^* \subseteq (L + M)^*$

b. Montrer que, quels que soient $L, M \subseteq \Sigma^*$:

1. $(L^*M)^* = \epsilon + (L + M)^*M$
2. $(LM^*)^* = \epsilon + L(L + M)^*$

Exercice 4: Facteurs gauches.

L'ensemble $fg(u)$ des facteurs gauches d'un mot $u \in \Sigma^*$ se définit par la récurrence suivante : $fg(\epsilon) = \epsilon \quad fg(ux) = fg(u) + ux \ \forall u \in \Sigma^*$ et tout $x \in \Sigma$.

a. Vérifier que cette définition est bien la bonne, c'est-à-dire que

$$(v \in fg(u)) \Leftrightarrow (\exists w \in \Sigma^* u = vw)$$

b. Vérifier que $fg(uv) = fg(u) + ufg(v) \ \forall u \in \Sigma^*$ et tout $v \in \Sigma^*$

c. Exprimer $fg(LM)$ et $fg(L^*)$ pour $L, M \subseteq \Sigma^*$ quelconques.

Exercice 5: Facteurs droits.

En mettant la définition de l'ensemble $fg(u)$ des facteurs gauches du mot $u \in \Sigma^*$ devant un miroir on peut obtenir facilement une définition de l'ensemble $fd(u)$ de ses facteurs droits, par une récurrence basée sur l'ajout des lettres à gauche : $fd(\epsilon) = \epsilon \quad fd(xu) = xu + fd(u) \ \forall u \in \Sigma^*$ et tout $x \in \Sigma$.
Si l'on s'obstine à vouloir ajouter les lettres à droite, on est conduit à poser la définition par récurrence suivante :

$$fd(\epsilon) = \epsilon \quad fd(ux) = \epsilon + fd(u)x \ \forall u \in \Sigma^* \text{ et tout } x \in \Sigma$$

a. Vérifier que cette définition est bien la bonne, c'est-à-dire que

$$(v \in fg(u)) \Leftrightarrow (\exists w \in \Sigma^* u = wv)$$

b. Vérifier que $fd(uv) = fd(v) + fd(u)v \ \forall u \in \Sigma^*$ et tout $v \in \Sigma^*$

c. Exprimer $fd(LM)$ et $fd(L^*)$ pour $L, M \subseteq \Sigma^*$ quelconques.

Exercice 6: Facteurs.

L'ensemble $fact(u)$ des facteurs du mot $u \in \Sigma^*$ peut se définir par récurrence de la faÃĄon suivante, si l'on ajoute des lettres "Ãă droite et à gauche" :
$fact(\epsilon) = \epsilon \quad fact(x) = \epsilon + x$ pour tout $x \in \Sigma$,
$fact(xuy) = fact(xu) + fact(uy) + xuy\ \forall x, y \in \Sigma$ et tout $u \in \Sigma^*$. Si l'on s'obstine à vouloir ajouter les lettres à droite, on est conduit à poser la définition par récurrence suivante :

$$fact(\epsilon) = \epsilon \quad fact(ux) = fact(u) + fd(u)x\ \forall u \in \Sigma^* \text{ et tout } x \in \Sigma$$

a. Vérifier que cette définition est bien la bonne, c'est-à-dire que

$$(v \in fact(u)) \Leftrightarrow \left(\exists w \in \Sigma^* u = wvw'\right)$$

b. Vérifier que $fact(uv) = fact(u) + fd(u)fg(v) + fact(v)\ \forall u \in \Sigma^*$ et tout $v \in \Sigma^*$

c. Exprimer $fd(LM)$ et $fd(L^*)$ pour $L, M \subseteq \Sigma^*$ quelconques.

Exercice 7: Application.

Soient a et b des symboles distincts l'un de l'autre.
Calculer $fg(L_i)$, $fd(L_i)$ et $fact(L_i)$ $(1 \le i \le 6)$ dans chacun des cas suivants :

$L_1 = a^*b^* = \{a^m b^n \mid 0 \le n, m\}$ $\qquad$ $L_4 = \{a^m b^n \mid 0 \le m < n\}$
$L_2 = \{a^m b^m \mid 0 \le m\}$ $\qquad$ $L_5 = \{a^m b^n \mid 0 \le n \le m\}$
$L_3 = \{a^m b^n \mid 0 \le m \le n\}$ $\qquad$ $L_6 = \{a^m b^n \mid 0 \le n < m\}$

Exercice 8: Système d'équations linéaires en langages.

Résoudre les Systèmes d'équations suivants :

$$\begin{cases} X_0 &= bX_0 + aX_1 \\ X_1 &= aX_1 + bX_3 \\ X_2 &= aX_1 + bX_3 + \epsilon \\ X_3 &= bX_1 + aX_3 \end{cases} \quad \begin{cases} X_0 &= aX_1 + nX_2 \\ X_1 &= aX_1 + nX_3 \\ X_2 &= aX_4 + \epsilon \\ X_3 &= X_3 + \epsilon \\ X_4 &= aX_4 + \epsilon \end{cases} \quad \begin{cases} X_1 &= bX_2 + aX_3 \\ X_2 &= bX_1 + aX_4 \\ X_3 &= \epsilon + aX_4 + bX_2 \\ X_4 &= \epsilon + (a+b)X_4 \end{cases} \quad \begin{cases} S &= 0 + X_0 \\ X_0 &= 0X_0 + 1X_1 \\ X_1 &= 0X_2 + 1X_0 + 1 \\ X_2 &= 0X_1 + 1X_2 \end{cases}$$

Le dernier Système d'équations correspondant au langages des multiples de 3 écrits dans l'ordre inverse. Pourriez-vous en déduire un système d'équations du même langage dans l'ordre normal ?

Exercice 9: Mélange de mots.

Cette opération consiste à insérer de nouvelles occurrences de caractères à un mot.

a) Mélange de deux mots. Soit $mel(u, v)$ l'ensemble des mélanges de $u \in A^*$. et $v \in A^*$ défini par la (double) récurrence suivante :

$$mel(u, \epsilon) = u \quad mel(\epsilon, v) = v \quad mel(ux, vy) = mel(u, vy)x + mel(ux, v)y$$

1. Intuitivement, $w \in mel(u, v)$ est un mot de longueur $|u| + |v|$ obtenu en faisant "glisser" les caractères de v dans u, en respectant leur ordre relatif. Pour s'en persuader, calculer $mel(abc, def)$.
2. Vérifier que $mel(u, v) = mel(v, u)$.

b) Mélange de deux langages. L'application $mel : A^* \times A^* \to \mathcal{P}(\mathcal{A}^*)$ se prolonge aux langages par :

$$mel(L, M) = \sum_{(v,w) \in L \times M} mel(v, w)$$

c'est-à-dire : $u \in mel(L, M)$ ssi il existe $v \in L$ et $w \in M$ tels que $u \in mel(v, w)$.
Calculer $mel(a^*, b^*)$ et $mel(ab, a^*b^*)$.