Fiche de TD no. 10

Types et typage

Exercice 1. Considérez le langage suivant qui permet de définir des graphes :

$$\begin{aligned} GRAPHE &\rightarrow \text{'\{'}\ INSTRUCTIONS\ \text{'\}'} \\ INSTRUCTIONS &\rightarrow INSTRUCTION\ \text{';'}\ INSTRUCTIONS \mid \epsilon \\ INSTRUCTION &\rightarrow SOMMET \mid ARETE \\ SOMMET &\rightarrow ID \\ ARETE &\rightarrow ID\ DIRECTION\ ID \\ DIRECTION &\rightarrow \text{'}->\text{'} \mid \text{'}--\text{'} \end{aligned}$$

Voici un système de types permettant de dire si un graphe défini par cette syntaxe est orienté ou non :

$$\frac{is : \text{non-orienté}}{\{\ is\ \} : \text{non-orienté}} \qquad \frac{is : \text{orienté}}{\{\ is\ \} : \text{orienté}}$$

$$\frac{a : \text{non-orienté} \quad is : \text{non-orienté}}{a\ ;\ is : \text{non-orienté}} \qquad \frac{a : \text{orienté} \quad is : \text{orienté}}{a\ ;\ is : \text{orienté}}$$

$$\frac{is : \text{non-orienté}}{s\ ;\ is : \text{non-orienté}} \qquad \frac{is : \text{orienté}}{s\ ;\ is : \text{orienté}}$$

$$\frac{}{x\ --\ y : \text{non-orienté}} \qquad \frac{}{x\ ->\ y : \text{orienté}}$$

1. Donnez un exemple d'un graphe (décrit par cette grammaire) qui n'est pas orienté ni non-orienté.
2. Donnez un exemple d'un graphe qui est à la fois orienté et non-orienté.
3. Ajoutez des actions sémantiques à la grammaire (modifiez éventuellement cette grammaire) de façon à faire le calcul du type d'un graphe défini par cette syntaxe.
4. Modifiez le règles de typages afin qu'un graphe orienté puisse se considérer comme un graphe sans orientation.

Exercice 2. Pour chaque couple d'expressions de type qui suit, trouvez un unificateur :

1. $tableau(10, \alpha)$, $tableau(10, pointeur(\beta))$
2. $tableau(10, \alpha)$, $pointeur(\beta)$
3. $pointeur(\alpha) \times (\beta \rightarrow \gamma)$, $\beta \times (\gamma \rightarrow \delta)$.

Table des symboles, langages à blocs, portée

Exercice 3. Expliquez pourquoi, dans l'implantation de la table de symboles, nous n'avons pas fait recours à des systèmes d'allocation dynamique (tels que `malloc`).

Exercice 4. Modifiez la grammaire de l'exercice afin qu'on puisse définir des sous-graphes d'un graphe donné.

Exercice 5. Considérez le programme C suivant :

```
#define IMPRIMER printf("%d : %d %d %d\n",++i,a,b,++c)
int main()
{
  int i=0,a=0,b=1,c=2; IMPRIMER;
  {
    int a=1,c=1; IMPRIMER;
    {
      int a=3,b=2; IMPRIMER;
    }
    {
      int a=2; IMPRIMER;
    }
    {
      int b=3,c=0; IMPRIMER;
    }
  }
  {
    int b=2,c=3; IMPRIMER;
  }
}
```

1. Que s'affiche à l'écran ?
2. Représentez cette structure de blocs par un arbre.

Exercice 6.

1. Décrivez comment une table d'adressage à liste évolue à fur et mesure que l'on insère les symboles déclarés dans le programme de l'exercice précédent.
2. Soit h une fonction d'hachage telle que $h(a) = h(c) = 1$ et $h(b) = 0$. Décrivez comment une table d'adressage dispersée évolue à fur et mesure que l'on insère les symboles déclarés dans le programme de l'exercice précédent.

Exercice 7. Un intervalle est un sous-ensemble $I \subseteq \mathbb{N}$ de la forme

$$I = [a, b] = \{\, x \mid a \leq x \,\text{et}\, x \leq b \,\}\,.$$

Soit $\mathfrak{I}$ une collection d'intervalles telle que, pour $I, J \in \mathfrak{I}$ on a :

$$I \supseteq J \,\text{ou}\, J \supseteq I \,\text{ou}\, I \cap J = \emptyset\,.$$

Supposons que $I_0 \in \mathfrak{I}$ soit tel que $I_0 \supseteq J$ pour tout $J \in \mathfrak{I}$. Soit

$$I \to J \,\text{ssi}\;\; \{\, K \in \mathfrak{I} \;\mid\; I \supseteq K \supseteq J \,\} = \{\, I, J \,\}\,.$$

Montrez que le graphe dirigé enraciné $(\mathfrak{I}, \to, I_0)$ est un arbre : pour tout $I \in \mathfrak{I}$ il existe un et un seul chemin de I_0 vers I.