

Grammaires attribuées

Licence info S5
TD COMPIL – 2011-2012

Exercice 1 : Des listes et des logs

On s'intéresse à un fichier de log contenant les événements générés par un capteur. Pour simplifier, ces événements sont réduits à l'émission d'une valeur de type entier positif.

On considère dans un premier temps la grammaire G_1 suivante, d'axiome $listeEvt$ et de terminaux $\{ \text{EMIT_INT} \}$: $listeEvt \rightarrow \text{EMIT_INT } listeEvt \mid \epsilon$

Q 1.1 : Donner un arbre syntaxique pour le mot `EMIT_INT EMIT_INT` dans la grammaire G_1 . □

Q 1.2 : Attribuer G_1 pour associer à l'axiome le nombre de valeurs émises. Indiquer clairement, pour chaque attribut, à quel terminal ou non-terminal il est associé et s'il est synthétisé ou hérité. □

Q 1.3 : Attribuer G_1 pour associer à l'axiome la plus grande valeur émise. □

Q 1.4 : Attribuer G_1 pour associer à l'axiome la liste des valeurs émises, dans l'ordre d'émission. □

On considère maintenant la grammaire G_2 suivante, d'axiome log et de terminaux $\{ \text{EMIT_INT} \}$:

$$\begin{aligned} log &\rightarrow listeEvt \\ listeEvt &\rightarrow evt \ listeEvt \mid \epsilon \\ evt &\rightarrow \text{EMIT_INT} \end{aligned}$$

Q 1.5 : G_2 est-elle équivalente à G_1 ? Recommencer l'attribution de la question 3. Indiquer clairement, pour chaque attribut, à quel terminal ou non-terminal il est associé et s'il est synthétisé ou hérité. □

On rajoute maintenant dans le système un contrôleur qui indique quand les données émises par le capteur ne sont pas fiables. L'événement `SKIP` indique qu'il faut passer en mode "ignorer les données", l'événement `RESUME` qu'il faut passer en mode "prise en compte des données", qui est le mode initial du système. Dû à l'asynchronisme des échanges, il est possible que le log contienne d'abord un `RESUME` (alors ignoré), ou plusieurs `SKIP` qui se suivent (seul le premier a alors un effet).

Par exemple, si le fichier de log contient les événements
`RESUME EMIT_INT(1) SKIP EMIT_INT(2) EMIT_INT(3) SKIP EMIT_INT(3) RESUME EMIT_INT(5) RESUME EMIT_INT(6) SKIP EMIT_INT(2)`
alors la liste extraite du log doit contenir, dans l'ordre, les valeurs 1, 5, 6.

On considère donc maintenant la grammaire G_3 suivante, d'axiome log et de terminaux $\{ \text{EMIT_INT}, \text{INTEGER}, \text{SKIP}, \text{RESUME} \}$:

$$\begin{aligned} log &\rightarrow listeEvt \\ listeEvt &\rightarrow evt \ listeEvt \mid \epsilon \\ evt &\rightarrow \text{EMIT_INT} \\ &\quad \mid \text{SKIP} \\ &\quad \mid \text{RESUME} \end{aligned}$$

Q 1.6 : Donner un arbre syntaxique pour le mot `EMIT_INT SKIP EMIT_INT RESUME EMIT_INT` dans la grammaire G_3 . □

Q 1.7 : Attribuer la grammaire G_3 pour associer à l'axiome la liste des entiers à prendre en compte par le système, en proposant une attribution avec attribut hérité, et une sans. Indiquer clairement, pour chaque attribut, à quel terminal ou non-terminal il est associé et s'il est synthétisé ou hérité. □

Exercice 2 : Évaluation d'entiers et bases

On considère la grammaire suivante qui décrit des nombres octaux et décimaux¹ : son axiome est

1. En pratique, un nombre est une entité lexicale décrite au niveau lexical par une description régulière, et n'est donc pas traité par une grammaire attribuée comme le fait cet exercice.

nombre, ses terminaux sont $\{\text{chiffre}, \text{o}, \text{d}\}$, ses non-terminaux sont $\{\text{nombre}, \text{base}, \text{suiteChiffre}\}$:

$$\begin{aligned} \text{nombre} &\rightarrow \text{suiteChiffre base} & \text{base} &\rightarrow \text{o} \mid \text{d} \\ \text{suiteChiffre} &\rightarrow \text{chiffre} \mid \text{suiteChiffre chiffre} \end{aligned}$$

Q 2.1 : Donner un arbre syntaxique pour le mot **chiffre chiffre d**. □

La valeur du nombre est par exemple 17 pour 21o ($17 = 2 \cdot 8 + 1$) et 21 pour 21d. Noter que 324o a pour valeur $(3 \cdot 8 + 2) \cdot 8 + 4 = 3 \cdot 8^2 + 2 \cdot 8^1 + 4 \cdot 8^0 = 212$.

Q 2.2 : Attribuer la grammaire de telle sorte qu'un attribut de l'axiome donne la valeur du nombre reconnu. Donner une attribution avec uniquement des attributs synthétisés, et une attribution avec attribut hérité. □

Q 2.3 : Quel contrôle sémantique manque-t-il dans cette grammaire attribuée (en quoi est-elle trop permissive)? □

Q 2.4 : On remplace la production $\text{suiteChiffre} \rightarrow \text{suiteChiffre chiffre}$ par la production $\text{suiteChiffre} \rightarrow \text{chiffre suiteChiffre}$. Le langage engendré est-il le même que précédemment? Est-il nécessaire de modifier l'attribution pour attacher à l'axiome la valeur du nombre reconnu? □

Exercice 3 : Boîtes Tex

Dans un langage de formatage à la Latex on utilise des boîtes pour positionner les caractères les uns par rapport aux autres. Il y a deux types de boîtes :

- les boîtes *horizontales* : les éléments y sont disposés en ligne les uns derrière les autres, alignés sur leur bord supérieur ;
- les boîtes *verticales* : les éléments y sont disposés en colonne les uns au dessus des autres, alignés à gauche.

Un élément d'une boîte peut être un caractère ou une boîte. On utilise la grammaire suivante :

$G = (V_T, V_N, \text{boite}, P)$ avec $V_T = \{\mathbf{h}, \mathbf{v}, (,), \mathbf{x}\}$, $V_N = \{\text{boite}, \text{listeb}\}$ et P contient les productions :

$$\begin{aligned} \text{boite} &\rightarrow \mathbf{x} \mid \mathbf{h} (\text{listeb}) \mid \mathbf{v} (\text{listeb}) \\ \text{listeb} &\rightarrow \text{boite} \mid \text{boite listeb} \end{aligned}$$

Par exemple la description de boîte $\mathbf{v}(\mathbf{x} \ \mathbf{h}(\mathbf{x} \ \mathbf{v}(\mathbf{x} \ \mathbf{x}) \ \mathbf{x}) \ \mathbf{x})$ correspond à :

```
x
x x x
  x
x
```

On souhaite attribuer G pour connaître les dimensions d'une boîte, cad sa largeur et sa hauteur, sachant qu'un caractère a comme largeur et hauteur 1. On suppose données les classes Java **Dimension** et **Orientation** :

```
public class Dimension {
    ...
    public int getH(){...};
    public int getL(){...};
    public void setH(int h){...};
    public void setL(int l){...};
}

public class Orientation {
    ...
    public static final HORZ = ...;
    public static final VERT = ...;
}
```

Q 3.1 : Attribuer G (en utilisant des instructions Java) pour calculer la dimension d'une boîte, préciser pour chaque attribut à quel terminal/non terminal il est attaché, s'il est hérité ou synthétisé et quel est son type. Discuter les solutions avec et sans hérité. □