

Université de Rennes 1
IFSIC
Master 1 d'informatique
Compilation

2ème session 2005-2006
14-06-06, 10h30-12h30h

Ce sujet comporte 2 pages. Tous les documents de cours, TD et TP sont autorisés. Le barème est indicatif. Toutes les réponses doivent être justifiées (de façon concise mais précise).

Partie I - Analyse syntaxique (8 points)

Question 1

Montrer que la grammaire suivante présentée dans la syntaxe de YACC contient exactement un conflit LALR(1).

```
%token op ident int
%start S
Term : ident '[' int ']' ;
Last : Term ;
Next : Term | ;
First : Term ;
Ellipse : '...' | ;
S : '(' op First Next Ellipse Last ')' ;
```

Question 2

Dans le processus de production d'un analyseur à partir d'une grammaire YACC quelle place occupe le fichier y.output ?

Commenter l'extrait suivant de fichier y.output.

```
state 9
S : '(' op First Next . Ellipse Last ')' (8)
Ellipse : . (7)
"..." shift 11
ident reduce 7
Ellipse goto 12
```

Partie II - Génération de code (8)

On considère un langage d'expressions qui permet des constructions elliptiques à l'image de ce qui se fait en mathématiques, ex. $a_1+a_2+\dots+a_n$. qui exprime l'itération d'une somme sur tous les a_i entre a_1 et a_n . La syntaxe formelle de ce langage d'expressions est donnée par la grammaire de la partie I.

L'interprétation en est la suivante :

1. une expression (op a[i] a[j]) exprime que l'opération op est appliquée à a[i] et a[j].
2. une expression (op a[i] a[j] a[k]) exprime que l'opération op est appliquée à a[i], a[j] et a[k]. L'opération op est supposée associative et l'expression peut donc être interprétée

indifféremment comme $((a[i] \text{ op } a[j]) \text{ op } a[k])$ ou $(a[i] \text{ op } (a[j] \text{ op } a[k]))$.

3. une expression $(\text{op } a[i] \dots a[j])$ exprime que l'opération op est itérée de $a[i]$ à $a[j]$ avec un pas de 1 : $a[i] \text{ op } a[i+1] \text{ op } a[i+2] \text{ op } \dots \text{ op } a[j]$.
4. une expression $(\text{op } a[i] \ a[j] \dots a[k])$ exprime que l'opération op est itérée de $a[i]$ à $a[k]$ avec un pas de $j-i$: $a[i] \text{ op } a[j] \text{ op } a[2j-i] \text{ op } a[3j-2i] \text{ op } \dots \text{ op } a[k]$.

Question 3

Quelles sont les vérifications sémantiques qui doivent être faites avant de songer à engendrer du code pour ces expressions ? Toutes ces vérifications peuvent-elles être faites statiquement ?

On se propose d'engendrer du code pour l'évaluation de ces expressions. Ce code devra être compatible avec celui présenté en cours pour l'évaluation des expressions.

On suppose disponibles les attributs `ident` et `ind` pour le non-terminal `Term`, et `neutre` pour le terminal `op`. `ident` représente le nom du tableau : `a` dans les exemples ci-dessus. `ind` représente l'indice : `i`, `j` ou `k` dans les exemples. `neutre` représente l'élément neutre de `op` ; ex. `0` si `op` est l'addition. Ces attributs sont synthétisés.

Question 4

Proposer un code 3-adresses qui implémente correctement le quatrième cas de l'interprétation des expressions elliptiques.

Proposer un jeu d'attributs qui permet la définition de règles sémantiques pour engendrer du code évaluant ces expressions. Bien décrire le rôle de chaque attribut.

Dessiner un arbre de syntaxe abstraite du quatrième cas de l'interprétation des expressions elliptiques, et montrer la circulation des valeurs attribuées.

Question 5

Donner les règles sémantiques pour les non-terminaux `S`, `Next` et `Ellipse`.

Partie III - Analyse statique (4)

Question 6

Soit l'instruction $p[i] = q[j]$. Lesquels de p , i , q et j sont définis ? Lesquels de p , i , q et j sont utilisés ?

Soit l'instruction $*p = *q$. Lesquels de p et q sont définis ? Lesquels de p et q sont utilisés ?

Question 7

Qu'en conclure pour l'analyse d'activité ?