Université d'Orléans — Maîtrise d'Informatique

Année 2002-03

Partiel de compilation

– durée : 1h30 –

Barème à titre indicatif : 12 + 8

Exercice 1 Nous voulons écrire un compilateur pour le langage SQL. Rappelons qu'une base de données relationnelle est formée de *tables*, ayant chacune un certain nombre de *champs*. A titre d'exemple, nous utiliserons une base de données ayant trois tables[1] :

```
Etudiants(numero_etudiant, nom_etudiant, prenom_etudiant)
Cours(code_cours, descriptif)
InscriptionsCours(numero_etudiant, code_cours, note)
```

La grammaire (restreinte) des requêtes SQL qui nous intéressent est la suivante :

```
Requete     : select ListeChamps
              from ListeTables
              where Terme = Terme ;
ListeChamps : IdChamp
            | ListeChamps, IdChamp
IdChamp     : id
            | id.id
Terme       : const
            | IdChamp
ListeTables : id
            | ListeTables, id
```

Les symboles terminaux de la grammaire sont les caractères ';', '.', ',','=', les mots-clés `select`, `from`, `where`, ainsi que `const` qui désigne une constante et l'identificateur `id` qui désigne un nom de champ ou de table.

1. Proposer une structure de données pour la table des symboles, afin de pouvoir stocker les noms des tables de la base de données, ainsi que les noms des champs de chaque table. Donner le contenu de votre TDS pour la base de données ci-dessus.
2. Attribuer la grammaire afin de vérifier qu'une requête respecte les critères suivants :
   (a) Chaque table de la partie `from` doit exister dans la base.
   (b) Chaque champ utilisé dans la requête doit exister dans l'une des tables listées dans le `from`.
   (c) Chaque champ doit être identifié de manière non ambiguë : la requête

[1] excuser le manque d'originalité...

```
select numero_etudiant, nom_etudiant, note
from Etudiants, InscriptionsCours
where Etudiants.numero_etudiant =
      InscriptionsCours.numero_etudiant
```

est incorrecte parce que l'identificateur `numero_etudiant` de la première ligne correspond à deux mêmes noms de champ des tables dans la deuxième ligne. Si l'on remplace, dans la partie `select`, l'identificateur de champ `numero_etudiant` par `Etudiants.numero_etudiant`, la requête devient correcte.

Préciser, en plus des règles de calcul des attributs et des autres actions sémantiques, le rôle, le type et la nature de chaque attribut, ainsi que l'ordre d'évaluation des attributs. Spécifier les fonctions de manipulation de la TDS dont vous vous servez.

Exercice 2 Soient les déclarations suivantes :

```
type t = record
           a : array [4..7, 0..5] of integer;
           suite : ^t
         end;
var elt : t;
    i : ^t;
```

Les tableaux sont rangés par lignes, les entiers et les pointeurs occupent **quatre** octets. En supposant que l'allocation mémoire soit effectuée à partir de l'adresse 20, préciser l'adresse de chaque variable et traduire cet extrait de programme :

```
i := elt.suite;
while i <> NIL do
begin
    i^.a[5,3] := 0;
    i := i^.suite;
end
```