Licence d'informatique
Module de C/Unix

Examen – Partie C

Jean-Luc LEVAIRE

Septembre 1999

Les documents de cours et TD sont autorisés.

On rendra deux copies séparées pour la partie Unix et pour la partie langage C. La partie C est à rendre sur une *copie de couleur blanche.*

1 Questions de cours

Question 1
Donner la représentation schématique de `i` et de `p` avant exécution de la boucle `for`. Qu'affiche le programme ? Quel est le contenu du tableau `i` à la fin du programme ?

```
void
f(int *i)
{
  (*(i++))++;
}

void
main(void)
{
  int i[5] = {0, 1, 2, 3, 4}, *p = i;
  int j;

  for (j=0; j < 10; j++)
    f(p);
  printf("%d %d\n", i[0], *p);
}
```

Question 2
Dans le système de fichiers Unix, quelle est la différence entre un lien physique et un lien symbolique ?

Question 3
Que signifient les déclarations suivantes :
- `char *(*d1[2])(int);`
- `int (*d2(int (*d[])(int), int))(int);`

2 Manipulation de graphes

Le but de ce problème est d'implémenter des graphes dirigés. On se propose d'utiliser la structure présentée à la Figure 1. Un graphe G est représenté sous la forme d'une structure contenant trois champs : le nombre de nœuds de G, un tableau *suivant* contenant pour chaque nœud, sa

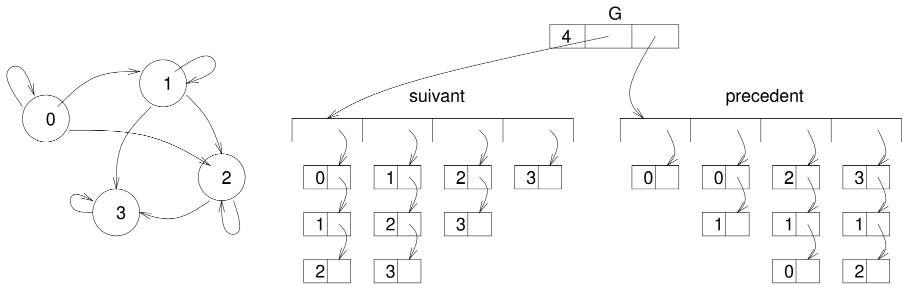

FIG. 1 – Un graphe G et sa représentation interne.

liste de nœuds suivants, et un tableau *precedent* contenant pour chaque nœud, sa liste de nœuds précédents. Ces deux tableaux, ainsi que les différentes listes de nœuds sont alloués *dynamiquement* lors de la création du graphe. Enfin les nœuds sont référencés dans les listes par un numéro d'ordre choisi quelconque à la création du graphe.

Question 4
Proposer un type `Noeud` pour représenter un noeud dans une liste, et un type `ListeNoeud` pour représenter une liste de nœuds. Proposer alors un type `Graphe` pour représenter un graphe. Proposer enfin un type `Bool` pour représenter le type booléen

Question 5
Écrire une fonction

```
Bool Reflexif(Graphe G) ;
```

qui teste si la relation associée à un graphe `G` est réflexive[1].

Question 6
Écrire une fonction

```
Bool Coherent(Graphe G) ;
```

qui teste si la représentation G d'un graphe est cohérente. Une représentation d'un graphe est cohérente, si les listes des suivants et des précédents sont cohérentes.

Question 7
À l'origine le graphe est représenté sous la forme d'une matrice d'adjacence `M`. `M` est une matrice carrée d'entiers, et un élément `M[i][j]` différent de `0` indique qu'il y a une arête du nœud `i` vers le nœud `j`. Ecrire une fonction

```
Graphe Convert(int M[][], int nbnoeuds) ;
```

qui convertit une matrice `M` en un objet de type `Graphe`. Le paramètre `nbnoeuds` est le nombre de nœuds du graphe.

[1]Un nœud a est en relation avec un nœud b si et seulement si il existe une arête entre a et b dans G. Une relation est réflexive si tout nœud est en relation avec lui-même.