

Travaux Dirigés

Séance nr. 4 : Graphes

Dans cet exercice on se propose de spécifier et d'implanter la structure *Graphe*.

Exercice 1: Spécification algébrique des Graphes

Proposer une spécification algébrique pour les **graphes orientés simples finis** avec des sommets de sorte *S*. Les opérations sont les suivantes : graphenouv, adjonction d'un sommet et d'un arc, suppression d'un sommet et d'un arc, existence d'un sommet et d'un arc, demi-degrés interne et externe d'un sommet.

Spécifier ensuite un type : ensemble de sommets (c'est un sous-ensemble de *S*). Les opérations sont : ensnouv, l'insertion d'un élément, la suppression d'un élément, l'appartenance, le test de vacuité, le cardinal et le choix d'un élément « quelconque » de l'ensemble. En utilisant la spécification des ensembles de type *S*, compléter la spécification du graphe avec les opérations suivantes : récupérer l'ensemble des sommets, récupérer l'ensemble des successeurs (resp. prédécesseurs) d'un sommet donné.

Refaire la spécification de ces opérations, en utilisant cette fois-ci une *liste* de sommets, plutôt qu'un *ensemble* de sommets.

Exercice 2: Représentation d'un graphe par liste d'adjacence

Implanter un graphe par liste d'adjacence. On utilisera les types suivants:

```
typedef struct strarc {S v; struct strarc *suc;} Strarc, *ListeSom;  
typedef struct {S v; ListeSom ls;} Couple;  
typedef struct strsom {Couple c; struct strsom *suiv;} Strsom, *Graphe;
```

Le type *S* est le type des sommets.

Proposer une implantation pour les opérations usuelles: graphnouv, adjs, adja, sups, supa, exs, exa, de, di, lsucc et lpred.

Pour cela, on écrira d'abord des opérations sur des listes chaînées de types « Strsom » et « Strarc » : adjtete, sup, recherche, etc.