INFO–F–203 — Algorithmique 2
Projet 1: ADT d'un graphe

Gilles GEERAERTS Frédéric PLUQUET

Année académique 2006–2007

1 Présentation de la structure

Dans le cadre de ce projet, nous considérons la structure de *graphe dirigé non-pondéré*, que vous avez déjà étudiée au cours théorique. Nous ferons l'hypothèse qu'il n'y a pas de sommet isolé. Nous considérons qu'un graphe est défini par son *ensemble d'arcs* E, chaque arc e étant défini comme une paire de sommets (n_1, n_2). Remarquez que, sous l'hypothèse qu'il n'y a pas de sommet isolé, il n'est pas nécessaire de donner explicitement l'ensemble de sommets, car celui-ci peut se déduire de l'ensemble d'arcs. Par exemple, considérez le graphe dont l'ensemble des sommets est $V = \{a, b, c, d\}$ et l'ensemble d'arcs est $E = \{(a, b), (b, c), (c, d), (d, a), (c, a)\}$. On voit que seul l'ensemble d'arcs E suffit à spécifier l'ensemble du graphe, car il contient toute l'information nécessaire pour retrouver l'ensemble de sommets. Tout graphe peut donc être construit à partir de l'ensemble (d'arcs) vide, en ajoutant successivement des sommets.

Nous considérerons les opérations suivantes :

L'accessibilité entre deux sommets qui consiste à déterminer, étant donné deux sommets n_1 et n_2 s'il existe un chemin dans le graphe dont le premier sommet est n_1 et le dernier est n_2. Plus précisément, n_2 est accessible à partir de n_1 si et seulement si : soit $n_1 = n_2$, soit on trouve dans le graphe un ensemble (non-vide) d'arcs $\{(e_1, e'_1), (e_2, e'_2), \ldots, (e_k, e'_k)\}$ tel que $e_1 = n_1$, $e'_k = n_2$ et, pour tout $1 \leq i \leq k - 1 : e'_i = e_{i+1}$.

L'ensemble de sommets accessibles à partir d'un nœud qui consiste à calculer, étant donné un nœud n, l'ensemble des sommets n' tel qu'il existe un chemin entre n et n'.

Le calcul du graphe inverse qui renvoie un nouveau graphe dans lequel il y a un arc entre n_1 et n_2 si et seulement s'il y a un arc entre n_2 et n_1 dans le graphe donné. Cette opération correspond donc en quelque sort à « retourner les flèches ».

2 Votre travail

On vous demande de réaliser le type de données abstrait d'un *graphe dirigé non-pondéré* tel que décrit ci-dessus (et représenté dans la suite par Graphe).

Pour réalisez ce type de données abstrait, vous vous servirez des constructeurs minimaux $\mathrm{AjouterArc}$: Graphe $\times$ Nœud $\times$ Nœud $\mapsto$ Graphe et $\mathrm{GrapheVide} : \emptyset \mapsto$ Graphe. Les primitives décrites devront impérativement comprendre les deux observateurs et l'opération décrits ci-dessus. En outre, vous fournirez un opérateur qui renvoie l'ensemble des nœuds d'un graphe donné, et un opérateur qui teste s'il existe un arc entre deux nœuds donnés.

Considérez ensuite le graphe cité en exemple dans le paragraphe introductif, et décrivez-le en terme de constructeurs minimaux. Déterminez ensuite s'il existe un chemin entre les sommets a et d, à l'aide de vos axiomes (veillez à justifier chaque étape). De la même manière, calculez l'ensemble de sommets de ce graphe ainsi que son inverse.

On vous demande enfin de fournir une classe C++ qui implémente la structure de graphe en essayant de respecter au mieux les concepts de l'orienté objet. Cette implémentation doit être conforme à votre type de données abstrait (vous utiliserez donc celui-ci comme spécification de votre classe). N'hésitez pas à écrire des tests afin de vérifier l'exactitude de votre solution.

Votre ADT, la manipulation demandée ainsi que votre code devront être présentés dans un court rapport à la présentation soignée. Le tout est à déposer au secrétariat étudiant du département d'informatique (2N8.104) le mardi 6 mars avant 15h30.