Devoir Programmation Logique

Manipulations de graphes étiquetés

Préliminaires

Nous allons travailler sur des graphes non orientés dont les sommets sont étiquetés. L'étiquette d'un sommet est un sous-ensemble d'un ensemble A d'attributs binaires (représentant des propriétés, des mots, etc ...). Par exemple, $A = \{x, y, z, k\}$ et l'étiquette du sommet a est $l(a) = \{x, y, k\}$. Par la suite, on écrira plus simplement $l(a) = xyk$. L'ensemble des étiquettes possibles est donc l'ensemble 2^A des parties de A.

On appellera *motif* un élément de 2^A et on s'intéressera au *support* d'un motif m dans un graphe G, c'est-à-dire au sous-ensemble S_m des sommets du graphe dont l'étiquette contient m. Ainsi, si $l(a) = xyk$ alors a fait partie du support S_{xy} de $m = xy$.

Un graphe est de la forme $G = (V, E, l)$ où l où donne accès à l'étiquette d'un sommet. Nous nous intéresserons au *sous-graphe G_S induit par un sous-ensemble S de sommets.* On a $G_S = (S, E_S, l)$ où E_S est le sous-ensemble des arêtes de G liant les sommets de S. Nous nous intéresserons en particulier aux sous-graphes induits par le support des motifs.

Abstractions de graphes

Nous allons nous intéresser à des opérations appelées *abstractions de graphes.* Pour définir une abstraction de graphe, on considère une propriété P sur un sommet dans un sous-graphe $G_S = (S, E_S, l)$. Par exemple $P = deg \geq 2$ est telle que $P(s, S)$ est vrai si le degré de s dans le sous-graphe engendré par S est supérieur ou égal à 2. Cette propriété doit vérifier la condition suivante :

Définition 1 *Soit* $P : V \times 2^V \to \{\text{vrai}, \text{faux}\}$ *telle que :*

- $s \notin S \Rightarrow P(s, S) = \text{faux}$ *: par convention la propriété est fausse pour les sommets qui ne sont pas dans le sous-graphe induit par S.*
- $S \subseteq S'$ *et* $P(s, S) \Rightarrow P(s, S')$ *: si la propriété est vérifiée dans le sous-graphe induit par S, elle l'est dans le sous-graphe induit par un sur-ensemble de S.*

P est alors la propriété caractéristique d'une abstraction de graphe.

Il est clair que la propriété $P = deg \geq 2$ est caractéristique d'une abstraction : si le degré de s dans le sous-graphe induit par S est supérieur ou égal à 2, alors si on ajoute des sommets à S pour former S', le degré du sommet s dans le graphe induit par S' ne peut qu'augmenter, donc $deg \geq 2(s, S')$ est vrai pour $S' \supseteq S$.

Pour obtenir l'abstraction d'un sous-graphe $G_S = (S, E_S, l)$ relativement à la propriété P, on procède de la manière suivante : en retirant les sommets s de S ne satisfaisant pas $P(s, S)$ on obtient un sous-ensemble S_1 de S. Dans le sous-graphe induit par S_1 certains des sommets restant s ne satisfont pas $P(s, S_1)$, on retire alors ces éléments ce qui produit S_2. On réitère ce procédé jusqu'à obtenir un point fixe S_P dont tous les sommets s satisfont $P(s, S_P)$. Le sous-graphe induit correspondant est l'abstraction de $G_S = (S, E_S, l)$ relativement à P.

Exemple 1 *Considérons la propriété $deg \geq 2$ et le graphe G de la Figure 1-a. Soit un motif t dont le support est $\{a, b, c, d, f\}$. On obtient alors le sous-graphe $G_{(\{a,b,c,d,f\}}$ représenté Figure 1-b. Pour obtenir l'abstraction de ce sous-graphe associée à $deg \geq 2$, on retire les sommets de degré inférieur strictement à 2 (ici f), et on considère le graphe induit par les sommets restant $\{a, b, c, d\}$. On retire alors d dont le degré dans ce nouveau sous-graphe induit est égal à 1, et dans le sous-graphe induit par les sommets restant $\{a, b, c\}$, tous les sommets sont de degré 2. On a donc atteint un point fixe et le graphe correspondant, représenté Figure 1-c est l'abstraction de $G_{(\{a,b,c,d,f\}}$.*

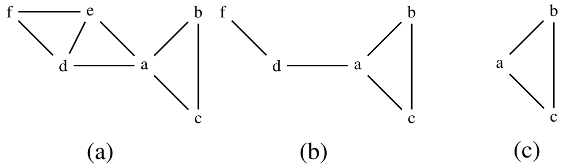

Figure 1: Soit le graphe G dessiné en (a). Les sommets $\{a, b, c, d, f\}$ induisent le sous-graphe dessiné en (b). L'abstraction de ce dernier associé à la propriété $deg \geq 2$ est dessiné en (c).

Projet

Vous devez concevoir un logiciel PROLOG de manipulation de graphes étiquetés, permettant d'entrer des requêtes associées à des motifs et de construire les sous-graphes associés aux requêtes et les abstractions de ces sous-graphes associés à différentes propriétés d'abstraction. Des prédicats permettant de lire et d'écrire un graphe attribué dans un fichier de type `.dot` vous seront fournis. Ils vous permettront de lire et écrire des graphes attribués sous ce format, et de les visualiser.

Les fonctionnalités attendues sont :

1. pouvoir répondre à des requêtes de la forme : "quel est le sous-graphe de G_1 induit par les sommets dont l'étiquette contient un motif donné m" ? Par exemple, quel est le sous graphe G_1 de G induit par un ensemble d'attributs. A partir du graphe G de la figure 2, quel est le graphe G_1 induit par le motif $[rock]$, par le motif $[rock, blues]$? G_1 peut lui-même être un sous-graphe de G induit par les sommets d'une requête précédente.

2. On veut pouvoir construire l'abstraction G_{S_P} d'un sous-graphe G_S d'un graphe G, associée à une propriété des sommets. Typiquement, on commencera par considérer un motif m, on cherchera le sous-graphe induit par le support de m, puis on fera l'abstraction de ce sous-graphe. Par exemple, on peut souhaiter calculer le graphe abstrait défini par la propriété $deg \geq 2$ et le motif $[rock, blues]$.

Votre logiciel doit donc pouvoir traiter un problème donné sous la forme d'un graphe de départ. On doit pouvoir explorer ce graphe et les sous-graphes résultant de requêtes concernant des motifs en utilisant au moins deux formes d'abstractions différentes. Parmi les abstractions possibles, on s'intéressera à

- $deg(y) \geq k$
- $cc(x) \geq k$: x appartient à une composante connexe de taille au moins k),
- $triangle(x)$: x appartient à un triangle.
- $etoile(x) \geq k$: x a un degré au moins k OU est voisin d'un sommet de degré au moins k,
- $coeurPeripherie(x)(c, p)$: $(deg(x) \geq c$ ET x lié à y avec $deg(y) \geq p)$ OU $(deg(x) \geq p$ ET x lié à y avec $deg(y) \geq c)$. On a c (coeur) plus grand que p (périphérie) par exemple $c = 4, p = 2$.

Il est à noter qu'il est possible également de combiner ces propriétés $P(x)$ avec des ET et des OU.

Le rapport contiendra en particulier une partie expérimentale illustrant les possibilités de votre logiciel. Le travail est à faire en binôme (nous contacter pour que nous validions exceptionnellement les monômes ou trinômes). Le projet est à rendre sur l'ENT avant le mercredi 9 janvier 2019 à 20h, délai de rigueur.

En annexe, vous trouverez le fichier `mougel.dot` représentant un graphe étiqueté au format standard `.dot`. Ce graphe est représenté Figure 2 à l'aide de la commande
`sfdp -Tpdf mougel.dot`
Les lignes 2 à 4 donnent une information générale : position du titre, taille de la police du titre, nom du problème (`Goûts Musicaux`) , liste des mots pouvant apparaître dans l'étiquette d'un sommet. Les lignes suivantes représentent les sommets : le nom du sommet (utilisé dans les lignes représentant les arêtes), suivi de l'information qui lui sera associée sur le dessin : le nom du sommet suivi de son étiquette sous forme d'un sous-ensemble de goûts musicaux. Le reste du fichier contient les arêtes.

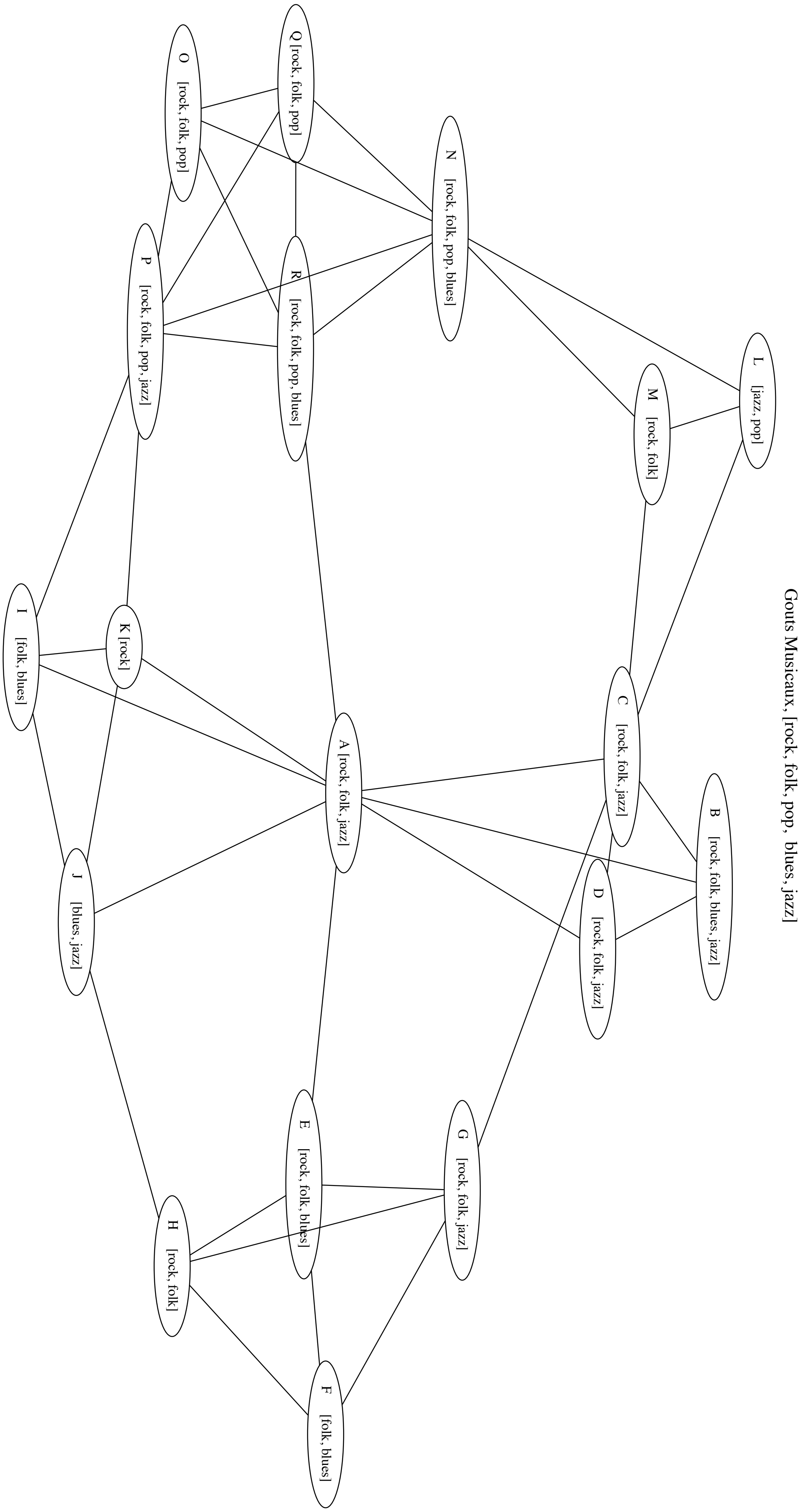

Figure 2: Le graphe étiqueté des goûts musicaux

.

Annexe : format E/S des graphes manipulés

Graph labelloc=top; fontsize=18; label="Gouts musicaux, [rock, folk, pop, blues, jazz]"; a [label="a [rock, folk, jazz]"]; b [label="b [rock, folk, blues, jazz]"]; c [label="c [rock, folk, jazz]"]; d [label="d [rock, folk, jazz]"]; e [label="e [rock, folk, blues]"]; f [label="f [folk, blues]"]; g [label="g [rock, folk, jazz]"]; h [label="h [rock, folk]"]; i [label="i [folk, blues]"]; j [label="j [blues, jazz]"]; k [label="k [rock]"]; l [label="l [jazz, pop]"]; m [label="m [rock, folk]"]; n [label="n [rock, folk, pop, blues]"]; o [label="o [rock, folk, pop]"]; p [label="p [rock, folk, pop, jazz]"]; q [label="q [rock, folk, pop]"]; r [label="r [rock, folk, pop, blues]"]; a – b; a – c; a – d; a – e; a – i; a – j; a – k; a – r; b – c; b – d; c – d; c – g; c – l; c – m; e – f; e – g; e – h; f – g; f – h; g – h; h – j; i – j; i – k; i – p; j – k; k – p; l – m; l – n; m – n; n – o; n – p; n – q; n – r; o – p; o – q; o – r; p – q; p – r; q – r;