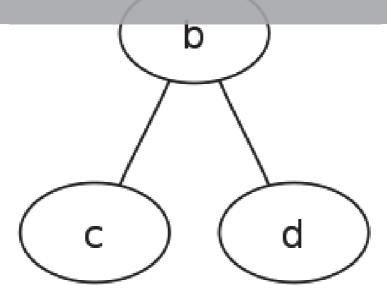


RAPPORT PROJET PROLOG



Travail encadré par :

M. SOLDANO Henry

Document réalisé par :

Nidam BENSAOULA Valentin OLIVE

Licence Informatique

Année universitaire 2013/2014

Table des matières

Introduction	. 2
Le menu d'interaction	. 3
Prédicat utiliser pour le menu	. 4
Prédicat utiliser pour la manipulation de graphe	. 6
Problème rencontré	. 8
Conclusion	. 9

Introduction

Après les différents cours magistraux et des travaux dirigés qui nous ont été dispensés tout au long du semestre, nous avons dû réaliser un projet en prolog.

Afin de le mener à bien, nous devions gérer plusieurs fonctionnalités. Il fallait tout d'abord être capable de gérer des entrées et sorties avec des fichiers .txt ou des fichiers .dot.

Plus important encore, le traitement des graphes : nous devions être capables de récupérer un sous graphe contenant tous les sommets ayant les étiquettes passées en paramètres. A cela s'ajoute la réalisation d'une abstraction d'un graphe suivant le degré donné en paramètre.

Dans la suite du document, nous allons vous présenter notre programme, en commençant par le menu d'interaction avec l'utilisateur, puis nous présenterons les fonctionnalités de notre programme et pour finir ce qui ne fonctionne pas comme nous le souhaiterions.

Le menu d'interaction

Notre menu d'interaction comporte 5 options : Chargement fichier, Sauvegarde fichier, Manipulation graphe, Affichage graphe, Quitter.

Le chargement d'un fichier est l'option 1. Lorsque nous choisissons cette option, nous devons entrer le nom du fichier .txt que nous voulons charger. Par exemple : pour un fichier graphe.txt on renseigne seulement graphe. Dans ce fichier nous devons avoir le predicat graphe(nom_du_graphe, definition_du_graphe). Et definition_du_graphe doit être de la forme [ListeSommet] ou ListeSommet doit être une liste d'élément de la forme [ListeEtiquette,[NomSommet|SuccesseurSommet]].

Une fois que nous chargeons ce fichier, nous enregistrons le graphe dans la base prolog à l'aide d'un assert.

La sauvegarde est l'option 2. Lorsque nous choisissons cette option, nous enregistrons le graphe que nous avons dans la base sous la forme de graphe(nom_du_graphe, definition_du_graphe). Nous l'enregistrons dans un fichier txt du nom nom_du_graphe et nous l'enregistrons également dans un fichier .dot du même nom.

Pour manipuler le graphe, nous devons choisir l'option 3. Si cette option est choisie nous quittons le menu et nous pouvons utiliser les différents prédicats de manipulation de graphe. Pour sauvegarder le graphe dans un fichier, nous devons d'abord utiliser le prédicat sauvegardeGraphe(NomDeSauvegarde,Graphe). Ce prédicat permet d'enregistrer le graphe obtenu après manipulation. Pour sauvergarder celui-ci nous devons relancer le prédicat menu et choisir l'option 2. Si l'option 1 est choisie, nous supprimons de la base le graphe obtenu précédement.

L'option 4 de notre menu permet d'afficher un graphe. De même que lors du choix de l'option 1, nous devons renseigner un nom de fichier.dot. Par exemple : pour un fichier graphe.dot on renseigne seulement graphe. Le graphe est alors affiché sous la forme du fichier.dot c'est-à-dire que le contenu du fichier.dot est affiché.

L'option 5 permet de quitter le menu. Lorsque cette option est choisie, nous supprimons tout ce qui se trouve dans les prédicats dynamiques de la base de connaissance.

Prédicat utiliser pour le menu

Voici la liste des prédicats que nous utilisons lors de l'interaction avec l'utilisateur grâce au menu :

- <u>menu/0</u>: Ce prédicat permet d'afficher le menu principal du programme.
- <u>menuChargement/0</u>: Ce prédicat permet d'afficher le menu pour le chargement du fichier.
- <u>menuAffichage/0</u>: Ce prédicat permet d'afficher le menu pour l'affichage d'un graphe.
- **choix/1**: Ce prédicat permet de récupérer le choix saisi par l'utilisateur dans le menu principal. L'argument est le choix saisi par l'utilisateur.
- **choix/2**: Ce prédicat est utilisé par les sous-menus. Il permet de lancer la suite du programme (la récupération du contenu d'un fichier). Le premier argument est défini par le programme et le second est ce que l'utilisateur a saisi.
- <u>finimenu/1</u>: Ce prédicat permet de faire un break dans la boucle infini (donc de revenir sur le menu principal). L'argument est défini par le programme.
- <u>charger/1</u>: Ce prédicat permet de charger un fichier donné en paramètre. Le paramètre est le nom du fichier à charger.
- <u>chargerTxt/2</u>: Ce prédicat permet récupérer le nom_du_fichier en nom_du_fichier.txt. Le premier argument est le nom_du_fichier et le second argument est le nom du fichier.txt retourné par le prédicat.
- <u>sauverTxt/0</u>: Ce prédicat permet de sauvegarder un graphe dans un fichier .txt.
- <u>sauvegarderTxt/3</u>: Ce prédicat permet de récupérer le nom du graphe et sa réprésentation dans le prédicat dynamique sauvegarde/3. Le premier argument est la représentation du graphe qui est récupéré par le prédicat. Le second argument est le nom du fichier dans lequel nous allons sauvegarder le graphe et le troisième argument est le nom du graphe.
- <u>sauverDot/0</u>: Ce prédicat permet de sauvegarder un graphe dans un fichier .dot.
- <u>sauvegarderDot/2</u>: Ce prédicat permet de récupérer le nom du graphe et sa réprésentation dans le prédicat dynamique sauvegarde/3. Le premier argument est le nom du fichier dans lequel nous allons sauvegarder le graphe et le second argument est le contenu à ajouter au fichier.
- <u>recupererFormatDotSucc/3</u>: Ce prédicat permet de récupérer la chaîne de caractères associé à un sommet et ses successeurs. Le premier argument est le sommet dont on veut récupérer la chaîne de caractères associé à ses successeurs. Le second argument est les successeurs de ce sommet et le troisième argument est le résultat contenant cette chaîne de caractères

- <u>recupererFormatDot/2</u>: Ce prédicat permet de récupérer la chaîne de caractères qui est au format pour un fichier.dot. Le premier argument est le graphe dont nous voulons récupérer la représentation en format .dot et le second argument est le contenu à ajouter au fichier.
- <u>supprimer liste/3</u>: Ce prédicat permet de supprimer un sommet d'une liste. Le premier argument est le sommet à supprimer, le second argument est une liste et le troisième argument est le résultat qui contient la liste sans le sommet.
- <u>supprimer sommet succ/3</u>: Ce prédicat permet de supprimer un sommet dans la liste des successeurs de chacun des autres sommets du graphe. Le premier argument est le sommet à supprimer, le second argument est un graphe et le troisième argument est le résultat qui contient le graphe sans le sommet.
- <u>afficher graphe/1</u>: Ce prédicat permet d'afficher un graphe associe à un fichier .dot. L'argument est le nom du fichier à afficher
- <u>afficherDot/2</u>: Ce prédicat permet de récupérer le nom du fichier et qui le formate en NomFichier.dot pour pouvoir l'ouvrir ensuite. Le premier argument est le nom_du_fichier et le second argument est le résultat qui contient le nom du fichier.dot.

Nous utilisons également les prédicats de manipulation de chaîne de caractères tels que : string concat/3, string to atom/2 et name/2. Et nous utilisons également les prédicats de flux tels que : open/3, read/2, write/2 et close/1. Enfin pour le menu nous utilisons le prédicat repeat/0 pour faire un menu avec une boucle infini et les prédicats read/1 et between/3 pour lire ce que l'utilisateur nous renseigne et vérifié qu'il ne nous a pas donné un mauvais choix.

Prédicat utiliser pour la manipulation de graphe

Voici la liste des prédicats que l'utilisateur pourra utiliser pour manipuler un graphe :

- <u>contientEtiquette/2</u>: Ce prédicat permet de vérifier que la liste des étiquettes du sommet contient une sous liste de la liste des étiquettes demandées. Le premier argument est la liste des étiquettes demandées et le second argument est la liste des étiquettes d'un sommet.
- <u>listeSommets/4</u>: Ce prédicat permet de récupérer dans une liste les sommets qui contiennent une sous liste de la liste des étiquettes demandées. Le premier argument est un graphe, le second argument est une liste d'étiquette. Le troisième argument est le résultat qui contient la liste des sommets contenant la liste des étiquettes données. Le quatrième argument est le résultat qui contient la liste des sommets ne contenant pas la liste des étiquettes données.
- <u>memberListe/2</u>: Ce prédicat permet de vérifier qu'au moins un élément de la liste est contenu dans la seconde. Le premier argument est une liste de sommets et le second argument est une liste de successeurs d'un sommet.
- parcours graphe/3 : Ce prédicat permet de parcourir le graphe pour récupérer les sommets du graphe qui sont contenu dans la liste de sommet demandé (contenant les étiquettes voulu) et qui construit le graphe induit. Le premier argument est un graphe, le second argument est la liste des sommets demandé et le troisième argument est le résultat contenant le graphe réduit.
- <u>sous graphe/3</u>: Ce prédicat permet de créer le sous graphe induit par les étiquettes demandées. Le premier argument est un graphe, le second argument est une liste de sommet et le troisième argument est le graphe réduit contenant uniquement les sommets dans la liste des sommets demandés.
- <u>sauvegardeGraphe/2</u>: Ce prédicat permet de sauvegarder un graphe à l'aide d'un prédicat dynamique. Pour pouvoir faire une sauvegarde dans un fichier. Le premier argument est le nom du graphe et le second argument est le graphe que l'on doit sauvegarder.
- <u>degreSommet/3</u>: Ce prédicat permet de récupérer le degré d'un sommet donné. Le premier argument est le sommet, le second argument est la liste des successeurs de ce sommet et le troisième argument est le degré de ce sommet.
- <u>abstraction graphe/3</u>: Ce prédicat permet, à partir d'un graphe, de créer un graphe contenant uniquement les sommets d'un degré supérieur ou égal à un degré donné. Le premier argument est un graphe, le second argument est un degré et le troisième argument est le graphe contenant uniquement les sommets d'un degré supérieur ou égal au degré donné en second paramètre.
- <u>listeSommetsDegre/4</u>: Ce prédicat permet de récupérer dans une liste les sommets qui ont un degré supérieur ou égal à un degré donné. Le premier argument est un graphe, le second argument est un degré. Le troisième argument est la liste des

sommets qui ont un degré supérieur ou égal au degré qui est donné en second argument. Le quatrième argument est la liste des sommets qui ont un degré strictement inférieur au degré qui est donné en second argument.

- <u>supprimer sommet succ/3</u>: Ce prédicat permet de supprimer un sommet dans la liste des successeurs de chacun des autres sommets du graphe. Le premier argument est le sommet à supprimer, le second argument est un graphe et le troisième argument est le résultat qui contient le graphe sans le sommet.

Nous avons utilisé des prédicats dynamiques qui sont : <u>graphe/2</u> et <u>sauvegarde/3</u>. Nous avons également utilisé les prédicats <u>assert/1</u> et <u>retractall/1</u>.

Problème rencontré

L'unique problème rencontré et qui ne fonctionne pas dans notre programme est lors du choix numéro 4 de notre menu principal. Le programme nous transmet en effet une erreur qui est erreur end_of_file dans le fichier. Nous ne pouvons donc pas afficher le contenu de ce fichier.

L'amélioration que nous pouvons apporter à notre programme concerne le menu. En effet, lorsque nous devons manipuler le graphe, nous sortons du menu et nous devons relancer le prédicat menu. pour pouvoir sauvegarder le graphe dans un fichier .txt et dans un fichier .dot.

Conclusion

Réaliser le projet fut ardu. Nous voulions nous y mettre le plus tôt possible, cependant, après plusieurs lectures du sujet de notre part, plusieurs points restaient toujours flous.

Après les explications données par le professeur, nous avons enfin été en mesure de faire un point sur ce que nous devions faire. C'est ainsi que nous avons été à même d'établir une liste des prédicats à coder et réaliser le programme de manipulation de graphe.

Pour ce faire, nous avons utilisés toutes les ressources mis à notre disposition durant ce semestre par nos professeurs.