Projet de Programmation Comparée : Analyse de Wikipedia

Université Paris Diderot – Master 2 SRI/LP

11 février 2014

1 Principe du projet

Ce projet de programmation comparée a pour objectif de vous faire réfléchir à la pertinence des choix de mécanismes calculatoires utilisés pour développer un logiciel. Ces choix ont en e□et des conséquences sur la capacité à résoudre le problème de façon succinte et e□cace, à raisonner sur la solution obtenue pour vérifier sa correction et pour se préparer à des extensions futures.

Il n'existe malheureusement pas un ordre total permettant d'évaluer objectivement et de façon définitive, un choix de conception vis-à-vis d'un autre. Il existe cependant des critères objectifs (modularité, e□cacité, déclarativité, portabilité, extensibilité, généralité, ...) fournissant une grille permettant de comparer et d'argumenter de façon à défendre un choix plutôt qu'un autre. Ce projet vise à vous faire développer une telle comparaison entre deux versions di□érentes d'un même logiciel.

2 Sujet

Le problème de ce projet est d'écrire un solveur de requêtes sur le contenu de Wikipedia vu comme un graphe dont les nœuds sont les pages et les arêtes sont les hyperliens entre ces pages. Le contenu de la version française des pages de Wikipedia est disponible ici :

`http://dumps.wikimedia.org/frwiki/latest/frwiki-latest-pages-articles.xml.bz2` [1]

Une requête calcule un ensemble de nœuds du graphe vérifiant une conjonction de critères basés sur deux mesures :

La distance en largeur entre deux nœuds Soient u et v deux nœuds. On appelle occurrences d'un noeud n, noté $\mathcal{O}(n)$ l'ensemble des positions des hyperliens pointant vers n dans les textes des autres nœuds. La distance en largeur entre u et v, notée $L(u, v)$, est définie par :

$$L(u, v) = min\{k \mid u \in V_k(v)\}$$

où

$$V_k(v) = \{w \mid \exists p_v \in \mathcal{O}(v), \exists p_w \in \mathcal{O}(w), |p_v - p_w| \leq k\}$$

La distance en profondeur entre deux nœuds Soient u et v deux nœuds. On dit qu'il existe un chemin entre u et v si il existe un chemin dans le graphe *non orienté* induit par le graphe orienté représenté par le contenu de Wikipedia. La distance en profondeur entre u et v, notée $P(u, v)$ est la longueur du plus petit chemin dont u et v sont les extrémités.

Les requêtes sont des conjonctions de prédicats de la forme $P_i(w) \leq k$ où P_i vaut $\lambda w.L(w, u_i)$ ou $\lambda w.P(w, u_i)$ et k est un entier positif ou nul. On se donne une syntaxe concrète pour ces requêtes dont voici la grammaire BNF :

1. Ce fichier est disponible sur la partie commune du serveur de fichier de l'UFR (`/info/master2/Public`). Ne le recopiez pas sur vos comptes personnels !

R	::=		*Requête*
	|	*C*	*Un seul critère*
	|	*C* ‘,’ *R*	*Un critère suivi d’une virgule puis de la suite de la requête*
C	::=	*D* ‘ <= ‘ k	*Un critère est une borne sur une mesure*
D	::=		*Distance*
	|	L ‘ < ‘ u ‘ > ‘	*La distance en largeur entre* w *et* u
	|	P ‘ < ‘ u ‘ > ‘	*La distance en profondeur entre* w *et* u

Les espaces et tabulations sont ignorés par cette grammaire sauf entre les chevrons. Les mots u sont des séquences de caractères qui ne sont pas des chevrons et ils correspondent aux titres des pages Wikipedia. Par exemple, la requête suivante cherche tous les mots à une distance en profondeur inférieure à 2 de “Denis Diderot” et à une distance en largeur inférieure à 10 de “Paris” :

```
P <Denis Diderot> <= 2, L <Paris> <= 10
```

La réponse à cette requête doit suivre la grammaire suivante :

S	::=		*Réponse à requête*
	|	*W*	*Un seul mot*
	|	*W* ‘,’ *S*	*Un mot suivi de la liste des mots*
W	::=	‘ < ‘ u ‘ > ‘	*Un mot entre chevrons*

Par ailleurs, les mots des réponses devront être triés par ordre alphabétique.

Votre programme devra attendre un fichier `requests.txt` contenant une requête par ligne et produire un fichier `answers.txt` contenant les réponses à chacune de ces requêtes dans le même ordre.

3 Travail demandé

Ce projet est un travail à faire en binôme et une soutenance aura pour objectif de démontrer que vous êtes réellement les auteurs du projet fourni.

Le 20 février Vous recevrez une liste de requêtes dans un fichier `requests.txt` suivant le format indiqué ci-dessus.

Avant le 10 mars Vous devrez rendre votre projet en l’envoyant par email à l’adresse suivante :

`yrg@pps.univ-paris-diderot.fr`

Il devra être accompagné du fichier `answers.txt` correspondant aux réponses (que vous aurez pu calculer) aux requêtes contenues dans le fichier `requests.txt` de votre groupe.

Il sera aussi accompagné d’un rapport décrivant votre architecture, vos choix de conception et les résultats obtenus (en particulier, le temps mis pour répondre aux requêtes).

Le 11 mars Vous recevrez le code source et le rapport d’un autre groupe (tiré au hasard et anonymisé).

Avant le 18 mars Vous devez produire un rapport comparant votre travail à celui de l’autre groupe du point de vue des choix d’implémentation e□ectués. Ce rapport devra être structuré à l’aide de critères de comparaison que vous aurez vous-mêmes choisis.

Bon courage !