

TP2 : AUTOMATES ET LANGAGES

Session	Automne 2019
Pondération	10 % de la note finale
Lieu de réalisation	L-4708
Taille des équipes	3 étudiants
Date de remise du projet	3 Décembre 2019
Directives particulières	Soumission du livrable par moodle uniquement (https://moodle.polymtl.ca).
	Toute soumission du livrable en retard est pénalisée à raison de 10% par jour de retard.
Les questions sont les bienvenues et peuvent être envoyées à: Stéphane Burwash (stephane.burwash@polymtl.ca), Jean-Philippe Anctil (Jean-Philippe.Anctil@polymtl.ca), Paulina Stevia Nouwou Mindom (Paulina_stevia.nouwou_mindom@polymtl.ca),	

1 Connaissances requises

- Notions d'algorithmique et de programmation C++, Python, Java ou autre (voir chargé).
- Notions d'automates et de langages.

2 Objectif

L'objectif de ce travail pratique est de vous permettre d'appliquer les notions théoriques sur les automates que nous avons vues en cours, sur des cas concrets mais hypothétiques, tirés de votre quotidien. A cet effet, il est question dans ce travail d'utiliser des automates pour encoder un lexique, faire de la complétion et de l'ordonnancement de mots.

3 Mise en situation

Le personnel en charge de faire les requêtes d'objets dans votre entrepôt demande au département d'informatique (vous, chers étudiants de LOG2810) de développer une méthode plus précise pour aller chercher des objets particuliers. Vous décidez donc d'attribuer à chaque objet un nom, un type (A, B ou C) et un code identificateur (hexadécimal à 6 caractères). Le code identificateur est unique aux objets, alors que plusieurs objets peuvent avoir le même nom ou le même type. Comme dans le TP1, les objets d'un certain type sont associés à un certain poids (A=1Kg, B=3Kg, C=6Kg). Vous écrivez ensuite un programme qui permettra au personnel de facilement pouvoir faire l'inventaire de leur stock en plus de pouvoir effectuer une commande d'objets aux robots. Voici les requis que vous établissez avec le personnel pour votre projet:

1. Nous pouvons effectuer une recherche selon 3 critères différents (sous forme de filtres) :
 - Type
 - Code identificateur
 - Nom
2. Lorsqu'une recherche par rapport à un certain critère est effectuée, tous les choix possibles sont affichés à la console. Dans le cas du code identificateur et du nom, il faudra afficher tous les choix possibles à partir du terme que vous avez écrit (complétion du mot).
 - Exemple 1 :
Recherche pour un objet de type A : retourne une liste de tous les objets dont le type est A et les affiche avec tous leurs propriétés.
 - Exemple 2 :
Recherche pour un objet dont le nom est che : retourne une liste de tous les objets dont le nom commence par les lettres che et les affiche avec tous leurs propriétés
3. Il doit être possible d'appliquer plusieurs filtres de recherche en même temps. Ainsi, nous pourrions chercher pour un objet dont le nom commence par abei et dont le code commence par A19 .
4. À chaque recherche, il doit être affiché le nombre d'éléments dans notre inventaire qui respectent les critères de recherche.
 - Exemple:
Recherche pour un objet de type A : Affiche le nombre d'éléments qui sont de type A dans notre inventaire présentement.
5. Il est possible d'ajouter des éléments à un panier pour effectuer une commande. Lorsqu'un élément est ajouté au panier, il est enlevé de l'inventaire.
6. Lorsque nous avons tous les éléments que nous voulons dans notre panier, nous pouvons passer une commande.
 - Si la commande est trop lourde (Poids > 25Kg), elle est refusée par le programme.
 - Sinon, la commande est passé. Les robots iront chercher vos objets, ce qui veut dire que ceux-ci ne seront plus dans votre inventaire. Il faudra prendre ça en compte lors des appels subséquents de vos fonctions.
7. L'application aussi longtemps que nous n'avons pas appelé la commande exit .

4 Tâches à réaliser

- C1. Créer un ou plusieurs automates à partir d'un lexique. Le lexique est donné sous la forme d'un fichier .txt.
- C2. Implémenter la fonctionnalité de suggestion qui permettra de suggérer à l'utilisateur des objets à partir du nom, code, type ou une combinaison de ces éléments qu'il a saisi à la console.
- C3. Créer une interface graphique simple (**oui, elle peut être écrite à la console**) qui permet à l'utilisateur de saisir différents éléments. Au fur et à mesure que l'utilisateur écrit un paramètre (par exemple le nom), votre application doit lui afficher, sur la base du lexique (automate), toutes les objets qui remplissent ce critère (dont le nom commence par les lettres écrites). Par exemple, si nous saisissons a pour le nom, H pour le code et A pour le type, votre programme devra afficher tous les objets dont le nom commence par A, le code commence par H et dont le type est A.

C4. Créer un système de commande d'objets

- Nous pouvons sélectionner des objets pour les ajouter à notre liste de commande
- Les contenus du panier doivent toujours être visible à l'écran ou être visibles à l'appel d'une fonction (libre à vous)
- Ajouter un objet au panier l'enlève de nos objets potentiels (donc il est non-recherchable)
- Implémenter une fonction qui vide le panier, remettant tous les objets dans notre inventaire (ils sont de nouveau recherchables)
- Implémenter une fonction de commande. Si la commande est au-dessus de 25Kg, il faut retirer des éléments avant de passer la commande (vous pouvez tout simplement vider le panier ou implémenter une fonction pour enlever des objets individuellement, libre à vous). Si la commande est valide (en dessous de 25Kg), la commande est passé et les objets sont enlevés de façon permanente.

C5. Faire une interface qui affiche le menu suivant :

- Initialiser le programme (qui prend en entrée le lexique fournit par le laboratoire et construit l'automate minimal associé).
- Fonctionnalité suggestion (qui permet à l'utilisateur de saisir du texte et de voir apparaître les suggestions d'objets lorsqu'il écrit). Pour les lexiques les plus importants, vous vous limiterez à l'affichage des 10 premiers objets que votre programme trouvera.
- Fonctionnalité d'ajout d'objet au panier
- Fonctionnalité de retrait d'objets du panier ou de vidage de panier
- Fonctionnalité de passage de commande
- Retour (qui permet à l'utilisateur de revenir au menu principal à partir d'une des trois interfaces précédemment citées).
- Quitter (qui permet à l'utilisateur de quitter l'application à partir du menu principal).

5 Livrable

Le livrable attendu est constitué du code source et du rapport de laboratoire. Le livrable est une archive (ZIP ou RAR) dont le nom est formé des numéros de matricule des membres de l'équipe, séparés par un trait de soulignement (_). L'archive contiendra les fichiers suivants :

- les fichiers .cpp;
- les fichiers .h le cas échéant;
- le rapport au format PDF.

L'archive ne doit pas contenir de programme exécutable, de fichier de projet ou solution de Visual Studio, de répertoire Debug ou Release, etc. Les fichiers .cpp et .h suffiront pour l'évaluation du travail.

5.1 Rapport

Un rapport de laboratoire rédigé avec soin est requis à la soumission (format .pdf, maximum 8 pages). Sinon, votre travail ne sera pas corrigé (aussi bien le code source que l'exécutable). Le rapport doit obligatoirement inclure les éléments ou sections suivantes :

1. Page présentation : elle doit contenir le libellé du cours, le numéro et l'identification du TP, la date de remise, les matricules et noms des membres de l'équipe. Vous pouvez compléter la page présentation qui vous est fournie.

2. Introduction avec vos propres mots pour mettre en évidence le contexte et les objectifs du TP.
3. Présentation de vos travaux : une explication de votre solution.
4. Difficultés rencontrées lors de l'élaboration du TP et les éventuelles solutions apportées.
5. Conclusion : expliquez en quoi ce laboratoire vous a été utile, ce que vous avez appris, ce que vous voudriez approfondir dans vos projets futurs, etc.

Notez que vous ne devez pas mettre le code source dans le rapport.

5.2 Soumission du livrable

La soumission doit se faire uniquement par moodle.

6 évaluation

éléments évalués	Points
Qualité du rapport : respect des exigences du rapport, qualité de la présentation des solutions	1.5
Qualité du programme : compilation, structures de données, gestion adéquate des variables et de toute ressource (création, utilisation, libération), passage d'arguments, gestion des erreurs, documentation du code, etc.	1.5
Composants implémentés : respect des requis, logique de développement, etc.	
C1	3
C2	4
C3	4
C4	3
C5	3
Total de points	20

7 Documentation.txt

- <http://www.cplusplus.com/doc/tutorial/>
- <http://public.enst-bretagne.fr/~brunet/tutcpp/Tutoriel%20de%20C++.pdf>
- Automate de Mealy : https://fr.wikipedia.org/wiki/Machine_de_Mealy