PIF1006 – Mathématiques pour informaticiens II

Session Automne 2022

TRAVAIL #1

Indications principals	
Date de présentation de l'énoncé	30 septembre 2022
Date de remise du travail	28 octobre 2022 à 23h59
Politique sur remises tardives	-25% par jour de retard
Éléments et format de remise	Rapport et fichiers complet de la solution ou du projet remis sous forme électronique sur le portail de cours dans un seul fichier compressé (.zip)
Pondération	10%
Nb d'étudiants par équipe	1 à 5, sans exception (une équipe de 2 ou 3 est conseillée)

INTRODUCTION

Dans ce premier travail pratique, il vous sera demandé d'expérimenter avec les notions d'automate.

Vous devrez valider des chaînes données en entrée et indiquer si la chaîne est acceptée ou refusée. Votre application devra pouvoir charger un fichier qui contient l'automate, que vous saurez éditer a priori.

Ainsi, l'objectif principal du travail est de vous faire réviser des notions de structures de données, de pratiquer la gestion des fichiers, la manipulation des chaînes de caractères et certains principes algorithmiques de base, tout en mettant en œuvre la théorie sur les automates telle que vue en classe.

DESCRIPTION DÉTAILLÉE DU TRAVAIL À ACCOMPLIR

Vous devez créer une application console ou avec une interface graphique (à votre guise) dans laquelle un utilisateur se voit offrir différentes options qui lui permettront de valider (accepter ou refuser) des expressions textes entrées par l'utilisateur puis passées en paramètres.

Vous devrez :

- Créer des fichiers textes/JSON/XML ou un format de votre choix dans un format désérialisable/interprétable facilement d'un point de vue algorithmique;
- Permettre à un automate de charger une liste d'états et de transitions;
- Permettre un affichage adéquat d'une représentation de cette automate;
- Permettre la validation (rejet ou acceptation) d'expression données en entrée par un utilisateur ou une utilisatrice.

Une solution de base (avec des classes) vous est fournie et doit être exploitée/ajustée, ce qui constituera votre solution, sachant que :

- Toutes les instructions plus précises s'y trouvent;
- Vous pouvez utiliser Java et copier/coller le code de ces classes et adapter;
- Vous devriez inscrire le nom de vos coéquipiers en entête du fichier Program.cs.

Par ailleurs, lors de l'analyse syntaxique, dans votre stratégie de « parcours » vous devez considérer que votre automate est déterministe, même s'il est non déterministe, c'est-à-dire que vous n'avez pas à gérer des retours arrière. Vous êtes invités à vous créer des fichiers d'automates déterministes, mais pouvez vous amuser avec des automates non déterministes.

AUTRES EXIGENCES/INDICATIONS SUPPLÉMENTAIRES :

- Assurez-vous que vous avez des commentaires pertinents dans votre code.

- Assurez-vous de tester votre application avant de la rendre en vous assurant qu'elle peut rouler sans planter. S'il est impossible pour le correcteur de tester des options dans l'application, il n'aura pas comme mandat de débugger et recompiler votre application et les points liés au bon fonctionnement de tout ce qui en découle seront perdus.

- Assurez-vous de remettre tous les fichiers de projet/solution, les fichiers de code ET les fichiers compilés de telle sorte que le correcteur puisse ouvrir votre projet dans l'environnement de développement intégré sans devoir créer de projet à partir de vos fichiers.

- À moins d'une entente particulière, vous DEVEZ utiliser C# .Net ou Java.

RAPPORT À REMETTRE

En plus de votre solution complète, votre équipe doit remettre un rapport contenant minimalement :

- 1 page de présentation;

- 1 page détaillant le rôle joué par chacun des membres de l'équipe;

- 1 section sur les problèmes et difficultés rencontrées s'il y a lieu;

- 1 section dans laquelle vous présentez un **guide utilisateur**, **imprimes-écran à l'appui**, qui non seulement détaille comment fonctionne votre application sous la forme d'une preuve de tests en décrivant au minimum pour chaque test :

    1. Le test réalisé;
    2. Les étapes réalisées incl. les entrées utilisées
    3. Les résultats tels qu'attendus (si vous n'avez pas réussi, vos hypothèses et la liste de ce que vous avez tentés de faire)

    Les tests demandés sont au min. les suivants :

    1. Un exemple de fichier d'automate conforme et son chargement initial;
    2. Un exemple de fichier d'automate non conforme (transition pour état non existant) et le succès du chargement en ignorant l'action;
    3. Un exemple de fichier d'automate non conforme (action autre que celles prévues) et le succès du chargement en ignorant l'action;
    4. L'affichage de l'automate en console/UI (le faire pour 2 automates différents avec au moins 4 états chacun)
    5. La soumission d'un input qui retourne une acceptation (3 cas de tests)
    6. La soumission d'un input qui retourne un rejet (3 cas de tests)
    7. La soumission d'un input composée d'au moins un caractère qui n'est pas un 0 ou un 1, et qui retourne un rejet sans plantage

GRILLE D'ÉVALUATION

Voici les critères d'évaluation et la pondération associée à chacun d'entre eux (pour chacun des critères sont évalués la fonctionnalité (est-ce que cela fonctionne?) et les structures/algorithmes/formes du code (est-ce bien fait et lisible?):

Sujet d'évaluation	***Pts (/10)***
Rapport complet/guide utilisateur/preuves de tests	2
Menu / interactions utilisateurs/trices et affichage en console	1
Chargement d'un fichier d'automate	2
Validation (acceptation/rejet) des entrées	4
Représentation textuelle d'un automate aux fins d'affichage	1