

**Rapport TP 2 - Automates**

Dans le cadre du cours

Structures Discrètes – LOG2810

Marc-André Jolicoeur - 1846304

Alexandre Boudreault - 1486776

Mathieu Bélanger - 1850591

28 novembre 2017

# Introduction

L’objectif de ce travail est de nous familiariser avec les notions d’automates, de structures de données et la théorie des langages que nous avons vus en cours. Le même stagiaire a encore besoin de notre aide, cette fois-ci pour un programme d’une compagnie de drone. Le travail est d’optimiser le parcours d’une flotte de drone de livraison à travers Montréal, du point de vue de la logistique.

L’application est implantée en Python, et il s’agit d’une application console sur MS-DOS. Elle gère les requêtes de livraison et s’assure d’optimiser leur acheminement. Elle implémente aussi un automate qui ne reconnait que les codes postaux spécifiques à Montréal.

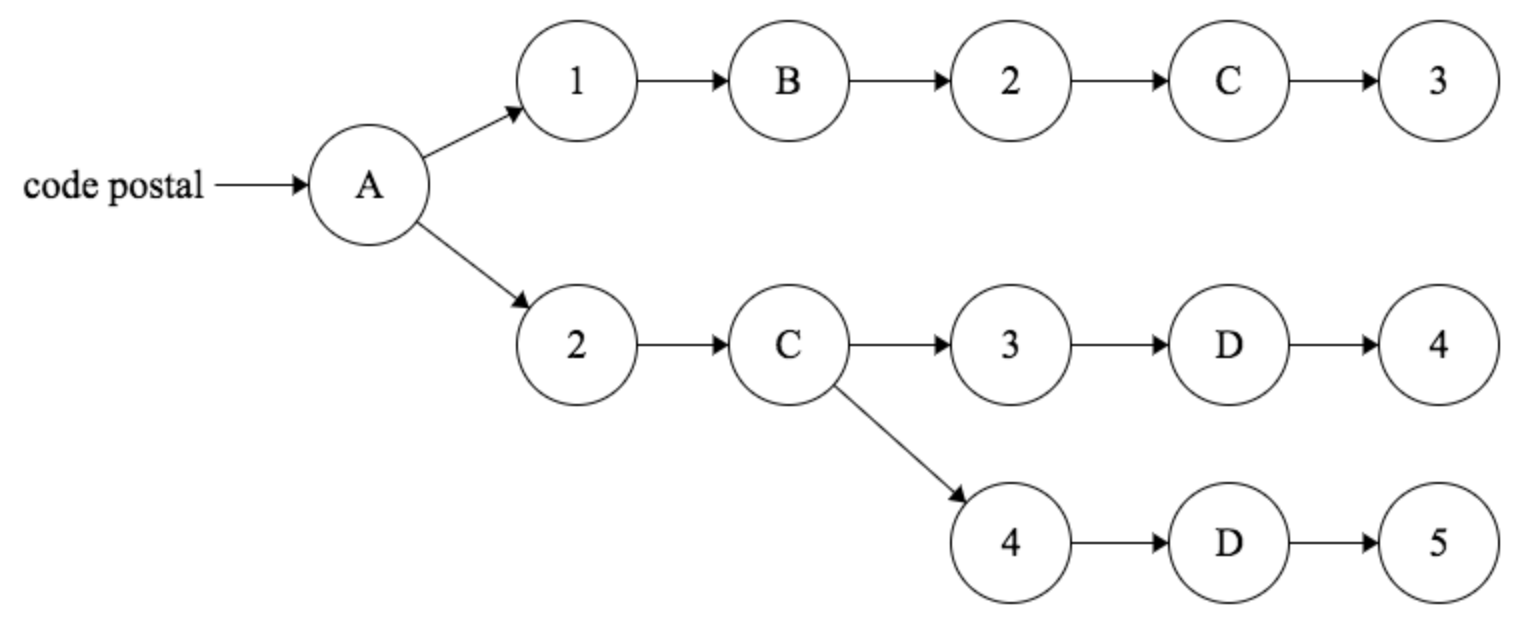
# Présentation de l’application produite

L’interface offre d’abord 4 choix à l’usager : (a) Créer l’automate, (b) Traiter des requêtes, (c) Afficher les statistiques et (d) Quitter.

L’option **Créer l’automate** crée une banque de codes postaux à partir d’un fichier texte. D’abord, on vérifie si les caractères du code sont bien disposés selon les critères d’un code postal : 6 caractères et une alternance de lettres et de chiffres (A1B 2C3). Si les critères sont respectés, le code peut être enregistré.

La façon dont ont été enregistré les données est particulière : elle consiste en une architecture à 6 niveaux, un pour chaque caractère du code postal. Ainsi, supposons les codes « A1B 2C3 », « A2C 3D4 » et « A2C 4D5 », « A » se retrouve au premier niveau et contient les chiffres qui lui sont associés : « A » : {1,2}. Ensuite, « 1 » qui se situe au deuxième niveau contiendra « B » et le « 2 » contiendra « C » et ainsi de suite.

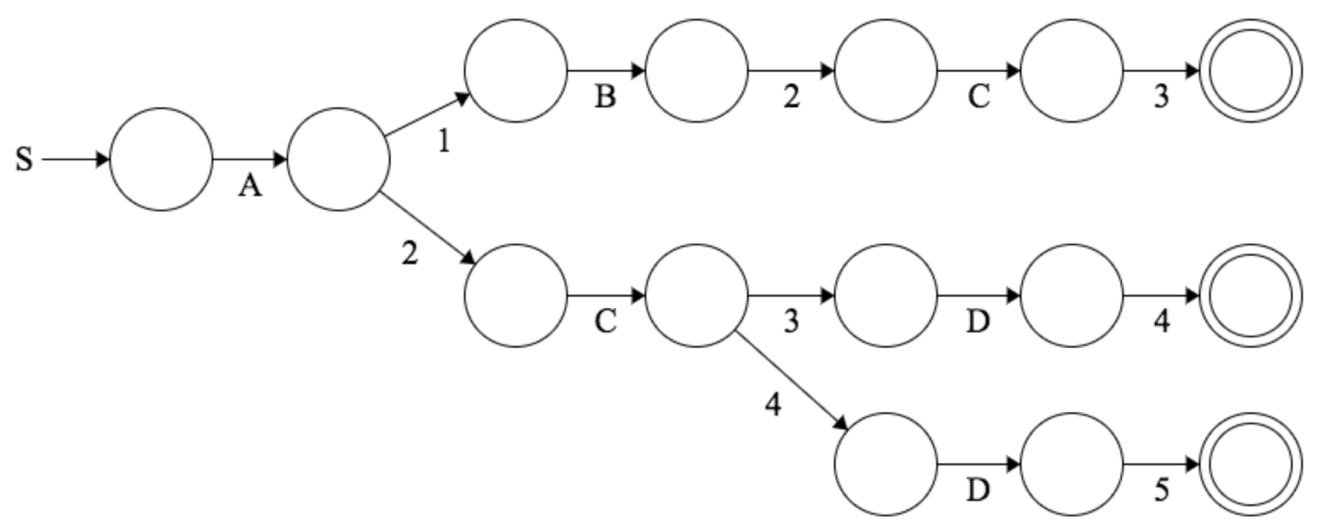
**Exemple de banque de code postaux**



Ceci nous permet de transformer la recherche des codes postaux en automate. En effet, chaque lettre du mot permet à l’automate de changer d’état. Si le caractère n’est pas reconnu, le mot est refusé. C’est seulement lorsque le mot est rendu au 6e état qu’il est validé par l’automate. Il s’agit en fait du principe de reconnaissance d’un langage, notion que nous avons abordée dans la théorie des langages. Ainsi, l’automate créé ne reconnait que les codes postaux spécifiques, et aucun autre mot.

**Exemple d’automate qui reconnait les codes postaux**

**« A1B 2C3 », « A2C 3D4 » et « A2C 4D5 »**



A

A1

A2

A1B

A2C

A1B2

A2C3

A2C4

A2C3D

A2C4D

A1B2C

A1B2C3

A2C3D4

A2C4D5

Une fois l’automate établi, la fonction procède en créant une flotte de drone à un endroit choisi, comme le garage de la compagnie par exemple. Ensuite, deux types de drones sont créés : les drones de catégorie 1 qui peuvent transporter jusqu’à 1kg et ceux de catégorie 2 qui transporte jusqu’à 5kg. Dix drones de catégorie 1 et cinq de catégorie 2 sont ajoutés à la flotte de drones. Les drones sont ensuite envoyés vers des centres de dépôt aléatoires pour mieux servir les futures requêtes. Ainsi, il est plus probable qu’un drone se trouve déjà à l’origine d’une commande au moment de traiter les requêtes.

L’option **Traiter des requêtes**, elle, coordonne les commandes de livraison qui se trouve sur un fichier texte. Les commandes sont sous la forme : *codePostalOrigine codePostalDestination poidDuColis*. Les deux codes sont envoyés dans l’automate, qui en vérifie l’appartenance au langage créé précédemment. Si les codes sont reconnus par l’automate, un objet de type *Delivery* est créé avec les arguments origine, destination et poids, puis est ajouté à la file d’attente. L’opération est exécutée pour chaque ligne du fichier texte. Les commandes qui ne sont pas sous la bonne forme ou qui comporte des codes postaux invalides sont ignorées. Un compte des commandes ignorées est maintenu. À noté que l’automate doit être créé avant de traiter les données, sinon un message d’erreur s’affiche.

Au début du cycle, un rééquilibrage des drones est effectué en fonction des commandes dans la file d’attente. Les drones se dirigent d’abord vers les origines des livraisons. Une fois à l’origine, il prend le colis. Ayant une charge limite, la capacité du drone est réduite de la masse du colis. Si un autre colis à cette origine doit être livré à la même destination et que la capacité du drone lui permet, le deuxième colis est pris par le drone. Si aucun autre colis n’est associé à la même destination ou si la charge du drone est pleine, le drone achemine le colis à l’endroit désiré et redevient disponible.

Si aucun drone n’est disponible pour se rendre à l’origine d’une commande, la commande est reportée au cycle suivant, avec priorité par rapport aux nouvelles commandes.

Si toutes les commandes sont traitées et des drones sont disponibles, ils sont répartis vers les centres de dépôts qui n’ont pas été visité dernièrement. Ainsi, un objet garde en mémoire le nombre de cycles depuis la dernière visite à chaque centre de dépôt. Si le centre n’a pas été visité depuis plusieurs cycle, un drone y sera envoyé lors du rééquilibrage. Ainsi, tout les centres de dépôt sont visités de temps en temps.

L’option **Afficher les statistiques**, comme son nom l’indique, permet d’afficher différentes informations sur la flotte de drones et les commandes traitées depuis le lancement du programme.

* Le nombre de requêtes traitées
* Le nombre de requêtes invalides
* Le nombre de drones dans chaque quartier suite au rééquilibrage
* Le nombre moyen de colis transportés par un drone à petite capacité
* Le nombre moyen de colis transportés par un drone à grande capacité

Pour arriver à garder ces données en mémoire, un objet de type « *RecordKeeper* » est créé et mis à jour pendant le traitement des requêtes.

Le menu est affiché tant que l’option **Quitter** n’est pas sélectionné.

# Difficultés rencontrées

Une difficulté majeure du travail pratique est le grand nombre de conditions qui sont prises en compte à chaque opération. En effet, il faut d’abord vérifier si les codes postaux de la commande sont valides. Ensuite, il faut vérifier si un des drones de la flotte est disponible pour la livraison. Il faut vérifier si d’autres colis ont la même destination afin d’optimiser les déplacements d’un drone. Il faut ensuite vérifier si la capacité du drone permet l’ajout du nouveau colis et ainsi de suite.

De plus, le fait qu’un drone puisse transporter plusieurs colis à la fois rend le rééquilibrage plus complexe. En effet, si deux colis ont les mêmes origine et destination, il faut envoyer qu’un drone qui pourra transporter les deux colis. Ainsi, il faut que lors du traitement des requêtes, le programme n’envoie pas bêtement un drone par origine de colis. Plutôt, pour minimiser les déplacements des drones, il faut regrouper les livraisons avec la même origine et destination et les assigner à un seul drone, en tenant compte des deux masses de colis.

# Conclusion

Nous sommes encore venus en aide à notre collègue. Le programme gère de façon autonome la flotte de drone de la compagnie. En fonction d’une banque de codes postaux, les requêtes sont validées et traitées par la flotte de drones, puis des statistiques peuvent être affichées.

De plus, le programme est très flexible en permettant notamment de modifier le nombre de drones, la capacité des drones, la banque de codes postaux reconnus et la localisation du garage de la compagnie. Ainsi, en cas d’expansion, le start-up pourra augmenter le nombre de drone dans sa flotte, ou changer la localisation du garage de la compagnie. De nouvelles catégories de drones peuvent aussi être ajoutées. Finalement, de nouveaux centres de dépôt peuvent être ajoutés à la banque de données. L’automate sera automatiquement modifié pour permettre la reconnaissance des nouveaux codes ajoutés.