

TD/TP1 PREMIERES NOTIONS

Notions utilisées

- Graphe d'états et automates
- Sommets, arcs et degrés d'un graphe, nombre chromatique
- Chemin, chaîne, cycle
- Théorème d'Euler
- Modélisation et implémentation en C++ d'un graphe avec matrice d'adjacence

Question 1.1 : Machine de Turing et graphe d'états (extrait du DS 2015)

Une machine de Turing (pas celle du film « Imitation game » sur le génial mathématicien et cryptologue Alan Turing) est réalisée avec les éléments suivants :

- un « ruban » divisé en cases consécutives. Chaque case contient un symbole parmi un alphabet fini. L'alphabet contient un symbole spécial « blanc » ('0' dans les exemples qui suivent), et un ou plusieurs autres symboles ;
- une « tête de lecture/écriture » peut lire et écrire les symboles sur le ruban, et se déplacer vers la gauche ou vers la droite du ruban ;
- un « registre d'état » mémorise l'état courant de la machine de Turing. Le nombre d'états possibles est toujours fini, et il existe un état spécial appelé « état de départ » qui est l'état initial de la machine avant son exécution ;
- une « table d'actions » indique à la machine quel symbole écrire, comment déplacer la tête de lecture (une case vers la gauche ou vers la droite), et quel est le nouvel état, en fonction du symbole lu sur le ruban et de l'état courant de la machine. Si aucune action n'existe pour une combinaison donnée d'un symbole lu et d'un état courant, la machine s'arrête.

La machine de Turing qui suit possède un alphabet $\{0', 1'\}$, '0' étant le « blanc ». On suppose que le ruban contient une série de '1', et que la tête de lecture/écriture se trouve initialement au-dessus du '1' le plus à gauche. Cette machine a pour effet de doubler le nombre de '1', en intercalant un '0' entre les deux séries. Par exemple, « 111 » devient « 1110111 ».

L'ensemble d'états possibles de la machine est $\{e1, e2, e3, e4, e5\}$ et l'état initial est e1.

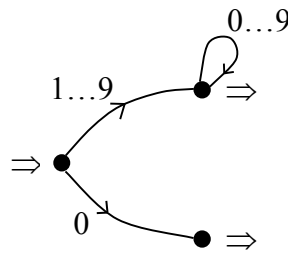
La table d'actions suivante représente les actions possibles (symbole lu, écrit et mouvement) entre 2 états :

Ancien état	Symbole lu	Symbole écrit	Mouvement	Nouvel état
e1	0	(Arrêt)		
	1	0	Droite	e2
e2	1	1	Droite	e2
	0	0	Droite	e3
e3	1	1	Droite	e3
	0	1	Gauche	e4
e4	1	1	Gauche	e4
	0	0	Gauche	e5
e5	1	1	Gauche	e5
	0	1	Droite	e1

Dessinez le graphe d'état-transition à partir de la table d'actions ci-dessus. Chaque sommet du graphe représente un état (e1, e2...) et chaque arc une transition qui dépend du symbole lu et indique les actions correspondantes (symbole écrit et mouvement). A l'aide de ce graphe, déterminez l'exécution de cette machine lorsque le ruban ne contient qu'un seul '1' au début (état initial)

Question 1.2 : des automates déterministes

Le graphe ci-dessous représente un automate déterministe (un seul chemin par solution) qui permet de reconnaître un entier naturel dont l'écriture est normalisée (ne commençant pas par un 0, s'il est non nul) :



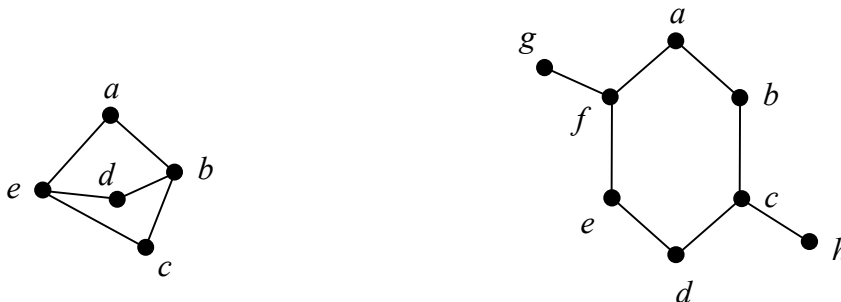
On entre un nombre entier par la gauche, et selon le premier chiffre on se dirige vers la branche du bas (si c'est un zéro), ou celle du haut (sinon). On recommence pour tous les chiffres du nombre. Si le dernier chiffre du nombre emmène sur un état de sortie, le nombre est valide, sinon il est invalide (état d'échec)

- a) Proposez un automate déterministe permettant de reconnaître un entier naturel multiple de 10 (terminant par 0)
- b) Proposez un automate déterministe permettant de reconnaître un horaire donné sous la forme hh:mm (de 00:00 à 23:59)

Question 1.3 : le graphe en couleurs

L'algorithme de Welsh et Powell est un algorithme « glouton » qui permet de trouver un certain nombre de couleurs pouvant colorer un graphe. Les chercheurs cherchent encore un algorithme efficace permettant de minimiser ce nombre de couleurs (le « nombre chromatique » du graphe). Si vous trouvez cet algorithme, à vous la vie multicolore !

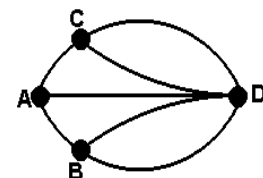
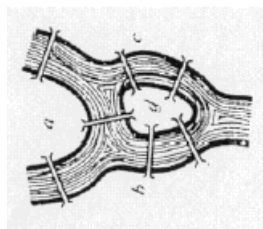
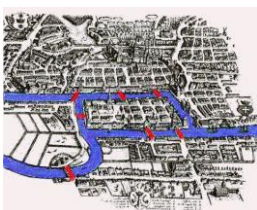
- a) Appliquer l'algorithme de Welsh et Powell aux deux graphes représentés ci-dessous.



- b) Comparer, pour chaque graphe, le nombre de couleurs obtenues par l'algorithme avec son nombre chromatique.

Question 1.4 : Königsberg (1736) et le théorème d'Euler

Sept ponts enjambent la Pregel, reliant quatre quartiers de la ville. Les habitants se demandent s'il existe un trajet leur permettant d'emprunter une seule fois tous les ponts. Les quartiers sont les sommets du graphe, les ponts les arêtes. Il y a quatre sommets (l'ordre du graphe est 4), au sommet A arrivent trois arêtes (le degré de A est 3) :



Euler modélise le problème et ouvre ainsi une nouvelle théorie. Le théorème d'Euler énonce qu'un graphe non orienté admet une chaîne eulérienne si et seulement si il est connexe et admet zéros ou deux sommets impairs. Si tous les sommets sont pairs, il s'agit de cycle eulérien.

- a) D'après le théorème d'Euler, existe-t-il un trajet partant d'un point donné, passant par toutes les arêtes une et une seule fois (chaîne eulérienne) ? Ou bien existe-t-il une chaîne eulérienne revenant au point de départ (cycle eulérien) ?
- b) À Königsberg, rebaptisée depuis Kaliningrad, il y a deux nouveaux ponts, l'un entre B et C et l'autre entre B et A. Y a-t-il une chaîne eulérienne ? Où faudrait-il construire un autre pont pour obtenir un cycle eulérien ?

Question 1.5 : modélisation d'un graphe et code en C++

Le conseil d'administration d'une société secrète (toute situation avec des personnes existantes est purement fortuite) est composé de 6 personnes : Messieurs Emmanuel M, Manuel V, François B, Alain J et Mesdames Marine L et Christiane T. Chacune de ces personnes influence certains de ses collègues conformément au tableau ci-dessous.

M. ou Mme...	influence...
Emmanuel M	B
Manuel V	T, J
François B	M
Alain J	personne
Marine L	V
Christiane T	personne

a) Représentez, au moyen d'un graphe - en précisant ce que représentent ici les sommets et arcs - les jeux d'influence au sein de ce conseil.

b) Ecrivez la matrice d'adjacence et calculez les demi-degrés intérieur et extérieur des sommets. Interprétez.

c) Pour le TP1, la partie qui suit est à coder en C++. Pour cela, inspirez-vous des **codes sources C++ pour implémenter un graphe** sur la page campus <http://campus.ece.fr/course/view.php?id=123> :

- Définir l'interface (fichier header .h) et l'implémentation (fichier source .cpp) de la classe *Sommet* contenant les propriétés publiques (on évitera ainsi les getters et setters pour encapsuler des attributs privés) suivantes :
 - Le nom du sommet ;
 - Un constructeur avec en paramètre le nom qui initialise l'attribut, avec par défaut la chaîne vide ;
 - Un destructeur.
- Définir l'interface (fichier header .h) et l'implémentation (fichier source .cpp) de la classe *Arete* contenant les propriétés publiques (on évitera ainsi les getters et setters pour encapsuler des attributs privés) suivantes d'une arête :
 - Les 2 sommets de la classe *Sommet* ;
 - Un constructeur avec en paramètre le nom qui initialise l'attribut, avec par défaut la chaîne vide ;
 - Un destructeur.
- Définir l'interface et l'implémentation de la classe *Graphe* contenant les propriétés publiques suivantes :
 - Au choix, une matrice d'adjacence dynamique d'entiers (ou de booléens) ou un vecteur d'arêtes ;
 - L'ordre (nombre de sommets du graphe) ;
 - Un vecteur de sommets ;
 - Un constructeur par défaut ;
 - Un destructeur qui détruit les sommets du vecteur ;
 - Une méthode avec en paramètre l'ordre dont l'objectif est d'allouer la matrice d'adjacence du graphe de « ordre » sommets et instancier le vecteur de sommets ;
 - Une méthode dont l'objectif est de récupérer dans un fichier texte les données de ce conseil d'administration pour les mémoriser dans les attributs : lire l'ordre dans le fichier, appeler la méthode précédente, remplir la matrice d'adjacence (avec des 0 et des 1) et les noms des sommets (personnes) dans le vecteur. Cette méthode a pour paramètre le nom du fichier ;
 - Une méthode qui affiche les jeux d'influences des personnes de ce graphe sous la forme « qui influence qui ».
- Écrire le **main** qui effectue les traitements suivants :
 - Instancier un objet dynamique de la classe *Graphe* ;
 - Inviter l'utilisateur à saisir le nom du fichier qui contient les données du conseil d'administration ;
 - Appeler la méthode qui récupère les données du fichier nommé avant dans les attributs de l'objet ;
 - Appeler la méthode qui affiche les jeux d'influence de l'objet ;
 - Détruire l'objet.