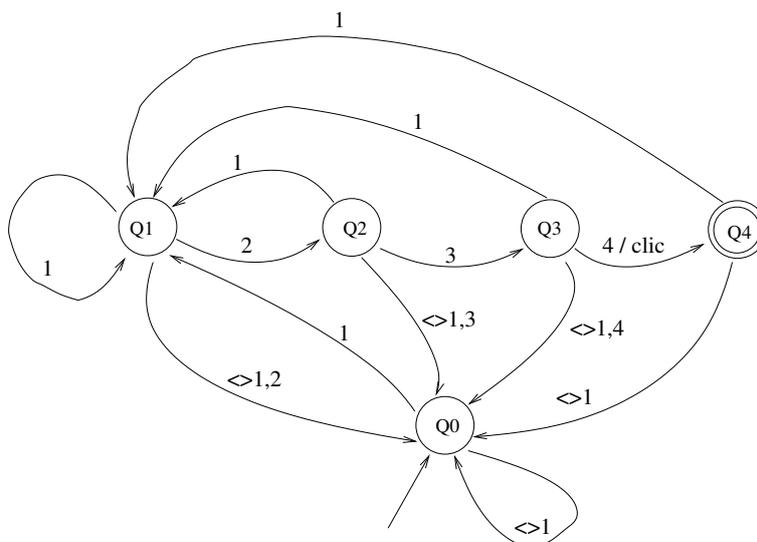


## INF203 - Travaux pratiques, séance 10

### Encore des automates ...

#### 1 Digicode

On s'intéresse à l'automate d'un digicode. Les entrées de cet automate sont les chiffres de 0 à 9. Il émet un message de sortie lors de l'ouverture de la porte.



[a] Quel est le code de la porte? ■

##### 1.1 Version programmée

[Digicode\_V1] Lisez et comprenez les fichiers fournis. A l'aide de l'automate ci-dessus, complétez la fonction `simule_automate` sur le modèle de ce qui est fourni : complétez le "case Q0", traitez le cas des états Q2, Q3 et Q4. N'oubliez pas d'afficher un message lorsque la porte s'ouvre. Compilez et testez votre programme.

[b] Que faut-il faire pour arrêter la simulation? (la réponse n'est pas CTRL-C ...). ■

Modifiez la condition de sortie de la boucle afin que la simulation s'arrête dans les conditions actuelles, mais aussi lorsque l'état final est atteint.

[c] Comment testez vous si la simulation est parvenue à un état final? ■

##### 1.2 Version programmée avec fonction

Lorsque votre digicode fonctionne, placez-vous dans votre répertoire TP10, et créez un répertoire `Digicode_V2`, puis copiez l'ensemble des fichiers de `Digicode_V1` dans `Digicode_V2`. Enfin, placez-vous dans `Digicode_V2`.

[Digicode\_V2] Modifiez `automate.c` de la façon suivante :

- ajoutez une fonction de profil `etat transition(etat Q, int e)` qui calcule et retourne l'état atteint à partir de `Q` avec l'entree `e` : pour cela, copiez-collez le `switch` de `simule_automate`, et adaptez les déclarations et noms de variables.

- dans `simule_automate`, remplacez le `switch` par un appel à la fonction `transition`.

Compilez, testez : cette version 2 doit se comporter comme la version 1.

### 1.3 Version tabulée

Créez un répertoire `Digicode_V3`, copiez-y les fichiers de `Digicode_V2`, et placez-vous dans `Digicode_V3`.

[`Digicode_V3`] Dans cette version, la fonction de transition est tabulée, c'est-à-dire codée dans un tableau.

[d] Ecrivez dans votre compte rendu la fonction de transition de l'automate sous la forme d'un tableau dont les lignes sont indicées par les numéros d'états (de 0 à 4), et les colonnes par les entrées (de 0 à 9). ■

Une fois cela fait :

- Supprimez la fonction `transition`.
- En vous aidant du tableau que vous avez écrit, déclarez et initialisez un tableau `transition` avec les bonnes valeurs.
- Modifiez la ligne qui calcule l'état suivant.

Testez!

[e] Quelle solution avez-vous adoptée pour afficher un message lorsque la porte s'ouvre? ■

Exercice complémentaire :

Modifiez le code de la porte ...

[f] (question difficile) Le code peut-il comporter plusieurs fois le même chiffre? ■

[g] Quels sont les avantages et les inconvénients des 3 versions dans le cas où on veut modifier le code? ■

### 1.4 Version tabulée avec fichier

Créez un répertoire `Digicode_V4` etc.

[h] Sur le dessin de votre tableau, entourez les valeurs différentes de 0.

- quel est leur rapport avec le code de la porte?
- à quelle(s) transition(s) correspond(ent) chacun des 4 chiffres du code?

■

Dans cette version, le code de la porte est lu dans un fichier, et sert, par conséquent, à initialiser le tableau `transition`.

1. dans `automate.c`, ajoutez une fonction de profil `void init_par_defaut()` qui initialise toutes les "cases" du tableau `transition` à 0.
2. choisissez un format pour le fichier contenant le code : la lecture de ce code dépendra du format choisi, choisissez donc un format que vous saurez lire!
3. dans `automate.c`, ajoutez une fonction de profil `int init_automate(char* nom_fich)` qui devra ouvrir en lecture le fichier de nom `nom_fich`, et :
  - retourner 1 si l'ouverture s'est mal passée;
  - sinon
    - initialiser l'automate aux valeurs par défaut;
    - lire le code et modifier le tableau `transition` en conséquence;
    - fermer le fichier et renvoyer 0.
4. modifiez le programme principal pour y intégrer la vérification du nombre d'arguments, l'appel de `init_automate`, la vérification de la valeur du code de retour (cf ci-dessus), puis la simulation de l'automate.
5. un autre fichier doit être modifié ... faites-le!

Testez cette nouvelle version.

[i] Quel format avez-vous choisi pour le fichier de code? Joignez le texte de votre fonction `init_automate` à votre compte rendu. ■