

Programmation Fonctionnelle Avancée —

Projet de fin de semestre n° 4

Avril 2019

Ce projet peut être réalisé en **Scheme** ou en **Ruby**. Les indications données ci-après s'appliquent à la version en **Scheme**, mais avec quelques modifications syntaxiques de l'interface, une version en **Ruby** devrait pouvoir fonctionner de façon analogue. Une version en **Java** peut aussi s'envisager, quoique difficilement.

Description

Le but du jeu est de réaliser la simulation d'un changeur automatique de devises permettant d'obtenir des francs suisses en donnant des euros, et *vice versa*. Pour ne pas compliquer outre mesure le projet, nous conviendrons une fois pour toutes de l'équivalence suivante :

$$1.00 \text{ €} = 1.12 \text{ CHF}$$

Voici les pièces et billets qui sont acceptés par l'automate — le nombre qui suit entre crochets droits indique la quantité, pour la pièce ou le billet correspondant, que possède l'automate à sa mise en service — :

- pour la monnaie européenne : 10 cents [10], 20 cents [10], 50 cents [10], 1 € [10], 2 € [5], 5 € [5], 10 € [5], 20 € [5], 50 € [4] ;
- pour l'argent suisse : 50 centimes [10], 1 franc [10], 2 francs [10], 5 francs [5], 10 francs [5], 20 francs [5], 50 francs [5], 100 francs [4].

Le *modus operandi* est le suivant : l'utilisateur indique au moyen du message **get-chf** (resp. **get-euros**) s'il veut des francs suisses (resp. des euros). Ensuite, il introduit des pièces et/ou billets de la monnaie qu'il possède. Puis il signale par le message **end** que le montant total est introduit. Il dispose également d'un message **cancel** pour invalider une transaction et récupérer son argent.

Dès que l'utilisateur introduit des pièces et/ou billets inconnus, la transaction est invalidée et la totalité de ce qu'il a introduit lui est restituée¹. La réponse de l'automate à une pièce connue ou un billet connue est la valeur logique *vrai* (**#t**). Lorsque le montant total est connu, il est converti dans l'autre devise et les règles suivantes sont appliquées :

- le résultat est arrondi au dixième d'euro inférieur ou au demi-franc suisse inférieur² ;
- si c'est possible — c'est-à-dire, si la monnaie nécessaire est disponible —, délivrer ce montant arrondi ;

1. Ce qui peut se produire avant que l'utilisateur donne le message **end**.

2. Qui a appelé les premières machines à sous des *bandits manchots* ?

- sinon, considérer le cas où nous sommes à moins de 2 cents du dixième d’euro supérieur ou à moins de 10 centimes du demi-franc suisse supérieur : délivrer ce montant si c’est possible ;
- sinon, invalider la transaction et rendre ses pièces et billets à l’utilisateur.

Le fait que l’automate donne des pièces et/ou billets à l’utilisateur, que ce soit dans la monnaie souhaitée ou parce qu’une transaction vient d’être invalidée est traduit dans notre simulation par une liste linéaire dont l’élément de tête est le nom de la monnaie, suivi des montants correspondant aux pièces et/ou billets fournis. On n’impose pas de contrainte sur l’ordre d’apparition des pièces et/ou billets concernés.

Voici un exemple commenté de première session³. Notre automate est réalisé par une fonction `automaton` dont l’argument est un message ou un nombre entier correspondant au monant d’une pièce ou d’un billet.

```
(automaton 10)      => #f      ; Message non compris : la monnaie n'a pas été précisée.
(automaton 'get-chf) => #t      ; OK. Ce sont des euros qui vont être introduits par l'utilisateur.
(automaton 10)      => #t      ; Introduction d'un billet de 10 €.
(automaton 5)       => #t      ; Puis d'un billet de 5 €.
(automaton 9)       => (euro 10 5) ; Mauvais montant — il n'existe ni pièce ni billet de 9 €
                                ; — : invalidation de la transaction depuis le début.
(automaton 10)      => #f      ; À nouveau, pas de monnaie précisée.
(automaton 'get-chf) => #t      ; OK.
(automaton 10)      => #t
(automaton 2)       => #t
(automaton 2)       => #t
(automaton 0.50)    => #t
(automaton 'end)     => (CHF 10 5 1) ; Après conversion et arrondi, nous devons délivrer
                                ; 16 francs suisses : un billet de 10 CHF, un billet de 5 CHF et
                                ; une pièce de 1 CHF.
(automaton 'thanks) => #f      ; Les symboles acceptés comme messages sont cancel, end,
                                ; get-chf, get-euros. Les autres symboles ne sont pas
                                ; reconnus, ils sont invalidés.
```

3. Par « première session », nous entendons qu’il s’agit des premiers appels de la fonction `automaton`.