

1 Introduction

Dans ce projet, je vous introduis à mon implémentation du DM de méthode formelle. Celui-ci se décline trois parties : présentations des propositions atomiques, introduction au modèle et pour terminer la liste des labels utilisés lors de la proposition atomique.

Projet réalisé par Mehdi BOUCHET en master MIC en 2022-2023.

2 Propositions atomiques

wait : *wait* est un bool. Elle permet de définir si l'utilisateur est autorisé à agir. Si la PA est à true, l'automate est dans l'attente d'une action utilisateur. Elle nous est utile afin de simplifier les transitions du modèle et l'expression de nos propositions. En effet, en poussant le sujet un peu plus en profondeur, si l'on souhaite implémenter la notion d'erreur utilisateur dans les propriétés du modèle en vérifiant s'il est autorisé ou non à appuyer sur un bouton, il nous suffira de vérifier la valeur de la PA *wait*. Elle est, par défaut, initialisée à true.

B_caf : *B_caf* est un bool. Elle permet de définir si l'utilisateur vient d'appuyer sur le bouton café. Le bouton est initialisé à false.

B_the : *B_the* est un bool. Elle permet de définir si l'utilisateur vient d'appuyer sur le bouton thé. Le bouton est initialisé à false.

B_ann : *B_ann* est un bool. Elle permet de définir si l'utilisateur vient d'appuyer sur le bouton annulé. Le bouton est initialisé à false.

B_suc : *B_suc* est un bool. Elle permet de définir si l'utilisateur vient d'appuyer sur le bouton sucre. Le bouton est initialisé à false.

Voy : *Voy* est une proposition atomique qui prend 3 valeurs numériques. Elle permet de définir quel est le voyant allumé. (*Voy* = 0) si le voyant prêt est allumé, (*Voy* = 1) si le voyant "préparation" est allumé et (*Voy* = 2) si le voyant "fin" est allumé. Le voyant est initialisé à 0.

Nous avons fait le choix de définir une proposition atomique numérique afin de simplifier l'écriture de notre modèle. Il est à noter qu'il n'est pas possible dans notre conception d'allumer deux voyants à la fois, auquel cas, j'aurais défini trois PA booléennes bien distinctes ou aurais agrandi le domaine des valeurs de *Voy* et défini (*Voy* = 3) comme le voyant "prêt" et "préparation" allumé et ainsi de suite... De plus, on peut définir (*Voy* = 3) "tous les voyants s'allument" afin de définir aisément le changement d'état des voyants lors d'une erreur machine.

P_50 : *P_50* est un bool. Elle permet de définir si la pièce vient d'être insérée par l'utilisateur. Elle est initialisée à false.

R_50 : *R_50* est un bool. Elle permet de définir si la pièce vient d'être rendue à l'utilisateur. Elle est initialisée à false.

À noter que ces deux propositions définissent une action de l'utilisateur, elles ne doivent pas persister dans le temps dans la représentation de notre automate.

G_out : *G_out* est un bool. Elle permet de définir si l'utilisateur vient de prendre son Gobelet. Elle est initialisée à false.

P_suc : *P* est un bool. Elle définit une proposition atomique pouvant prendre 3 valeurs permettant de définir la quantité de sucre demandé par l'utilisateur. (*P_suc* = 0) si l'utilisateur n'a pas demandé de sucre. (*P_suc* = 1) si l'utilisateur a demandé une quantité de sucre. (*P_suc* = 2) si l'utilisateur a demandé deux quantités de sucre. Elle est initialisée à 0.

Mode : Il est défini une proposition atomique pouvant prendre 4 valeurs permettant de définir le mode du distributeur. (*Mode* = 0) si le distributeur est dans le mode "disponible". (*Mode* = 1) si le distributeur est dans le mode "choix". (*Mode* = 2) si le distributeur est dans le mode "préparation". (*Mode* = 3) si le distributeur est dans le mode "fin". Il est initialisé à 0.

Type : Il est défini une proposition atomique pouvant prendre 3 valeurs permettant de définir le type de la boisson sélectionné par l'utilisateur. (*Type* = 0) si l'utilisateur n'a rien sélectionné. (*Type* = 1) si l'utilisateur a sélectionné du café, (*Type* = 2) si l'utilisateur a sélectionné du thé. Il est initialisé à 0.

Note : L'utilisateur ne pouvant être que dans un type par défaut, nous avons défini une propriété numérique pour définir les trois types plutôt que booléenne.

3 Le modèle

3.1 Mode 0

|| Mode= 0 & wait -> (P_50' = true) & (wait' = false)

Cette instruction représente l'état initial de l'automate. En effet, le distributeur est dans l'attente d'une pièce de la part de l'utilisateur (d'où le *wait*). Lorsque la pièce est insérée, *wait* passe à false et P_50 à true.

|| Mode= 0 & P_50 & !wait-> (Mode' = 1) & (P_50' = false) & (wait' = true);

Cette instruction définit un état par lequel passe l'automate lorsque l'utilisateur insère une pièce. Lorsque *wait* passe à false, l'automate transite par cet état avant de définir le mode du distributeur à 1. D'après notre définition de P_50, cette dernière repasse à false. De même pour *wait* : l'automate est dans l'attente d'une action de l'utilisateur

3.2 Mode 1

|| Mode= 1 & wait -> (Mode' = 0) & (B_ann' = true) & (R_50' = true) & (P_suc' = 0) & (wait' = false);

Lorsque l'automate est dans le (*Mode* = 1), l'utilisateur a 4 possibilités :

1) La possibilité annuler son choix et de reprendre sa pièce.

Cette instruction définit cette action.

|| Mode= 0 & !wait & B_ann -> (B_ann' = false) & (R_50' = false) & (wait' = true);

Lorsque la pièce a été rendue, on repasse à l'état initial.

|| Mode= 1 & wait -> (B_suc' = true) & (wait' = false);

2) L'utilisateur a aussi la possibilité d'ajouter du sucre. Si l'utilisateur appuie sur le bouton *B_suc*, on applique les conditions d'ajout décrites dans le projet. Cela se fait en définissant trois scénarios dépendant de la valeur de *B_suc* pour la suite de nos transitions.

|| Mode= 1 & !wait & B_suc & (P_suc = 0) -> (P_suc' = 1) & (B_suc' = false) & (wait' = true);

2) a) Si la boisson a préparé ne devait pas contenir de sucre, elle doit en contenir une quantité. L'utilisateur peut désormais choisir une nouvelle action.

|| Mode= 1 & !wait & B_suc & (P_suc = 1) -> (P_suc' = 2) & (B_suc' = false) & (wait' = true);

2) b) Si la boisson a préparé devait contenir une quantité de sucre, elle doit en contenir deux. L'utilisateur peut désormais choisir une nouvelle action.

|| Mode= 1 & !wait & B_suc & (P_suc = 2) -> (P_suc' = 0) & (B_suc' = false) & (wait' = true);

2) c) Si la boisson a préparé devait contenir deux quantités de sucre, elle ne doit plus en contenir. L'utilisateur peut désormais choisir une nouvelle action.

|| Mode= 1 & wait -> (B_caf' = true) & (wait' = false);

3) L'utilisateur a la possibilité d'appuyer sur un bouton de boisson Café *B_caf*.

|| Mode= 1 & wait -> (B_the' = true) & (wait' = false);

4) L'utilisateur a finalement la possibilité d'appuyer sur un bouton de boisson Thé *B_the*.

|| Mode= 1 & !wait & B_caf -> (Mode' = 2) & (Voy' = 1) & (B_caf' = false) & (Type' = 1) & (wait' = false);

Si *B_caf* a été appuyé, on définit le type Type de boisson à 1, on prépare le café et l'utilisateur doit attendre la machine (aucune action possible en Mode 2 d'où *Mode* = 2 et *wait* = false). On définit ainsi le type de boisson a préparé (décrit par la PA Type)

|| Mode= 1 & !wait & B_the -> (Mode' = 2) & (Voy' = 1) & (B_the' = false) & (Type' = 2) & (wait' = false);

Si *B_the* a été appuyé, on définit le type Type de boisson à 2, on prépare le thé et l'utilisateur doit attendre la machine.

On considère que l'automate reste bloqué dans ces deux modes tant que la boisson n'est pas terminée. Cela aurait pu être plus explicite avec l'aide d'une PA (*boisson_load* par exemple) qui passe à vrai en (*Mode* = 2) tant qu'elle est en préparation et à faux en (*Mode* = 2) lorsqu'elle est prête. Cela rajouterait un état en plus et nous permettrait par exemple de jouer le son "bip" avant de transiter à l'état 3 avec la prochaine instruction.

3.3 Mode 2

|| Mode= 2 & !wait -> (Mode'= 3) & (Voy' = 2) & (wait'= true);

Lorsque la boisson est prête, on passe au (*Mode* = 3).

Ps: le "bip" peut se jouer ici par exemple.

3.4 Mode 3

|| Mode= 3 & wait & !G_out -> (G_out'= true) & (Type'= 0) & (P_suc'= 0);

On attend que l'utilisateur prenne sa boisson. Dès que l'utilisateur prend sa boisson, on réinitialise les variables *Type* et *P_suc* à 0. L'automate n'est donc plus en position d'indiquer ni la nature de la boisson ni sa quantité de sucre. Il est à noter que j'ai choisi de définir *P_cafe* et *P_the* en tant que label, car nous n'avons pas besoin de ceux-là pour notre représentation. Voir plus loin **Les labels**.

|| Mode= 3 & G_out -> (Mode'= 0) & (Voy'= 0) & (G_out'= false);

Dès que l'utilisateur a pris sa boisson, on repasse à l'état initial.

4 Les labels

4.1 Le sucre

P_pas_sucre définit qu'il n'y a pas de sucre.

P_sucre définit qu'il y a une quantité de sucre.

P_tres_sucre définit qu'il y a deux quantités de sucre.

4.2 Le mode

M_dispo Le distributeur est en mode "disponible".

M_choix Le distributeur est en mode "choix".

M_prepa Le distributeur est en mode "préparation".

M_fin Le distributeur est en mode "fin".

4.3 Le type de boisson sélectionné

T_caf L'utilisateur a sélectionné du café.

T_the L'utilisateur a sélectionné du thé.

4.4 Le type de boisson préparé

Par esprit de simplicité, j'ai fait le choix de décrire les PA du sujet *P_cafe* et *P_the* par des labels, au travers la *Type*. Cela permet de suivre tout au long du processus le type de la boisson sélectionné au lieu de juste définir le type de la boisson préparé à terme, dans le (*Mode* = 3). Lors du (*Mode* = 3), il nous suffit de vérifier si (*Type* = 1) ou (*Type* = 2) afin d'avoir cette information. D'où la rédaction de cette information sous forme de label et non de proposition.

P_cafe Le café est préparé. Il est prêt à être retiré.

P_the Le thé est préparé. Il est prêt à être retiré.

4.5 Les voyants

V_rdy Le voyant "prêt" est allumé.

V_prp Le voyant "préparation" est allumé.

V_fin Le voyant "fin" est allumé.

4.6 La pièce

P_50 La pièce vient d'être insérée par l'utilisateur.

R_50 La pièce vient d'être rendue à l'utilisateur.

4.7 L'utilisateur

G_out L'utilisateur vient de prendre le gobelet de boisson.

4.8 Les boutons

B_the Le bouton "The" vient d'être appuyé.

B_caf Le bouton "Café" vient d'être appuyé.

B_ann Le bouton "Annuler" vient d'être appuyé.

B_suc Le bouton "Sucre" vient d'être appuyé.