Méthodes formelles - Vérification probabiliste

Sebastien Patte
Naïm Moussaoui-Remil

March 26, 2023

1 Modélisation

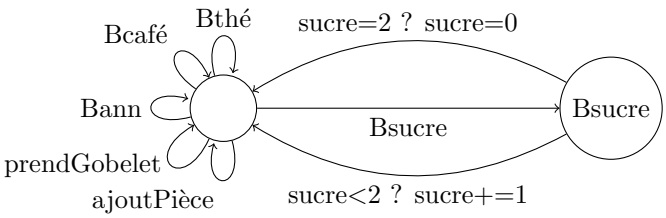
Le modèle est composé de 2 modules : Utilisateur et Contrôleur. L'utilisateur peut faire les actions `ajoutPièce`, `prendGobelet`, $\texttt{B}_{\texttt{ann}}$, $\texttt{B}_{\texttt{café}}$, $\texttt{B}_{\texttt{thé}}$ et $\texttt{B}_{\texttt{sucre}}$, mais uniquement quand ces actions actions sont simultanément possibles dans le modèle du Contrôleur.

1.1 Utilisateur

Dans le modèle de l'utilisateur, on s'occupe pricipalement de la gestion du niveau de sucre avec le bouton $\texttt{B}_{\texttt{sucre}}$, les autres actions étant gérées dans le module Contrôleur.

Les actions `ajoutPièce`, `prendGobelet`, $\texttt{B}_{\texttt{ann}}$, $\texttt{B}_{\texttt{café}}$, $\texttt{B}_{\texttt{thé}}$ bouclent sur l'état initial. Mais l'action $\texttt{B}_{\texttt{sucre}}$ fait passer le système dans un état où la proposition $\texttt{B}_{\texttt{sucre}}$ est vraie, et depuis lequel on peut seulement revenir à l'état initial par une des 2 actions internes en fonction du niveau de sucre actuel.

Le niveau de sucre : 0 (pas sucré), 1 (sucré), et 2 (très sucré), est incrémenté si le niveau actuel est < 2, et est réinitialisé à 0 si il était à 2.



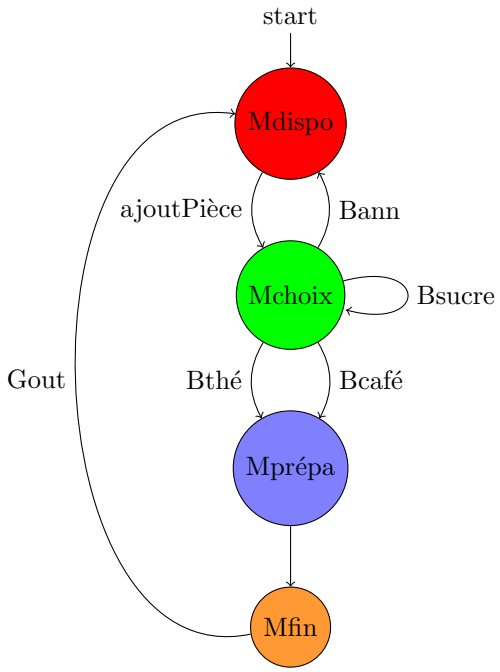
1.2 Contrôleur

La machine a café peut être dans 4 modes différents : $\texttt{M}_{\texttt{dispo}}$ $\texttt{M}_{\texttt{choix}}$, $\texttt{M}_{\texttt{prépa}}$ et $\texttt{M}_{\texttt{fin}}$, représentés respectivement par les couleurs rouge, vert, bleu, et orange.

1.2.1 Fonctionnement global

Depuis l'état initial, quand l'Utilisateur ajoute une pièce on passe dans le mode $\texttt{M}_{\texttt{choix}}$. Quand la machine est dans ce mode, l'utilisateur peut : règler le niveau de sucre avec l'action $\texttt{B}_{\texttt{sucre}}$, appuyer sur $\texttt{B}_{\texttt{ann}}$ pour retourner dans $\texttt{M}_{\texttt{dispo}}$, ou bien sur $\texttt{B}_{\texttt{café}}$ ou $\texttt{B}_{\texttt{thé}}$ pour aller dans le mode $\texttt{M}_{\texttt{prépa}}$.

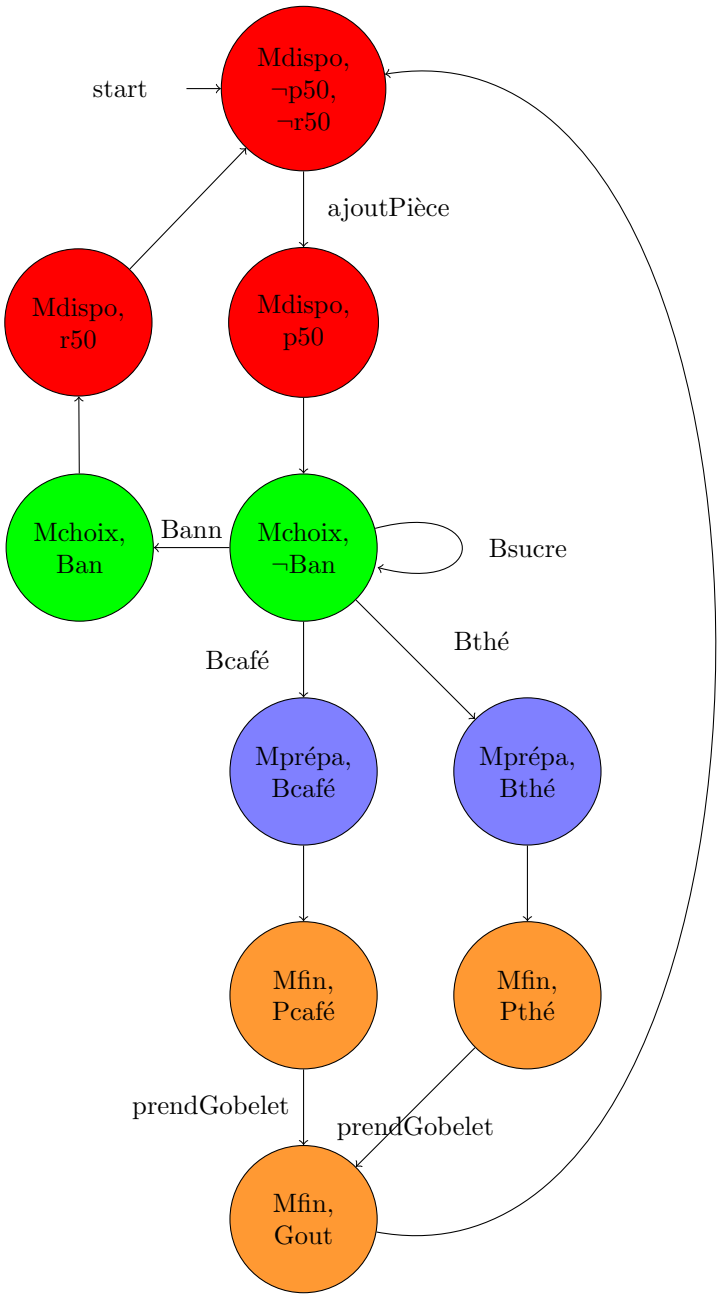
La machine passe du mode $\texttt{M}_{\texttt{prépa}}$ à $\texttt{M}_{\texttt{fin}}$ quand elle a terminé de servir la boisson, puis l'utilisateur peut alors prendre le gobelet pour faire retourner la machine dans son état initial.



1.2.2 États intermédiaires

Pour pouvoir utiliser les propositions atomiques $\texttt{B}_{\texttt{ann}}$, $\texttt{B}_{\texttt{café}}$, $\texttt{B}_{\texttt{thé}}$, $\texttt{P}_{50}$ $\texttt{R}_{50}$, et $\texttt{G}_{\texttt{out}}$, on a en fait un état intermédiaire pour chaque action de l'utilisateur.

Une action (`Bann`, `Bcafé`, `Bthé`, `ajoutePièce`, `prendGobelet`), fait atterir le système dans un état ou la proposition correspondante ($\texttt{B}_{\texttt{ann}}$, $\texttt{B}_{\texttt{café}}$, $\texttt{B}_{\texttt{thé}}$, $\texttt{P}_{50}$, $\texttt{G}_{\texttt{out}}$) est vraie, puis cette proposition passe à faux dans l'état suivant. De plus, depuis l'état où $\texttt{B}_{\texttt{ann}}$ est vraie, une action interne amène le système dans le mode $\texttt{M}_{\texttt{dispo}}$ avec la proposition $\texttt{R}_{50}$ vraie, puis directement dans l'état initial.


start
Mdispo,
¬p50,
¬r50
ajoutPièce
Mdispo,
r50
Mdispo,
p50
Mchoix,
Ban
Bann
Mchoix,
¬Ban
Bsucre
Bcafé
Bthé
Mprépa,
Bcafé
Mprépa,
Bthé
Mfin,
Pcafé
Mfin,
Pthé
prendGobelet
prendGobelet
Mfin,
Gout

2 Spécification

Nous allons reprendre quelques propriétés où une petite explication est nécessaire.

1. Ici la seul difficulté et d'exprimer un "xor" pour chaque cas d'où la présence de mutiple négation.

   ```
   Prop: A [ G (("Mdispo"&!("Mfin")&!("Mprepa")&!("Mchoix")) ....]
   ```

2. Dans le fichier `.props`, cette proporiété est séparée en 3. Rien de spécial à préciser.

3. Pour indiquer le fait que le changement d'état arrivera un temps plus tard en utilisant l'implication et l'opérateur next.

   ```
   Prop :A [ G ("Mdispo"&"P50" $\longrightarrow$ (X ("Mchoix"))) ].
   ```

4. Rien de particulier. Le "mais si" est traité comme une disjonction donc on obtient deux sous propriétés.

5. RAS.

6. Dans la première propriété, il est précisé que l'évenement arriverra "éventuellement". Ce qui nous indique qu'il faut utiliser l'opérateur F.

   ```
   Exemple prop 1: A [ G ("Mdispo"&(X (!"Mdispo"))=>(X(F"Mchoix"))) ].
   ```

   Dans les 3 autres, c'est directement à l'état d'après donc on utilise un next uniquement.

7. Le défi est de traduire "infiniment souvent". Ce que l'on traduit par "toujours un jours" soit "G F".

8. Dans le fichier `.props`, cette proporiété est séparée en 3. Rien de spécial à préciser.

9. En appuyant sur le bouton café on aura un jour un café. Donc utilise l'opérateur F encore.