LINFO1104 TP11: Système de Contrôle d'Ascenseur

Cette session se concentre sur l'étude des diagramme d'éta pour les systèmes concurrents ainsi que le système de contrôle d'ascenseur.

1.**Diagramme d'état** La figure suivante montre la notation pour représenter un diagramme d'état où les composants d'un système reçoivent et envoient des messages et change d'état. Par exemple, imaginez qu'on représente un lave-vaiselle comme un port object. Chaque fois que le le lave-vaisselle reçoit un message ajoutant un objet (une assiette, un verre,. . .), il doit vérifier s'il est plein ou pas. Si c'est le cas, il n'accepte pas le nouvelle objet et attend de recevoir le message pour le lancer le nettoyage. Sinon, il ajoute l'objet à son état.

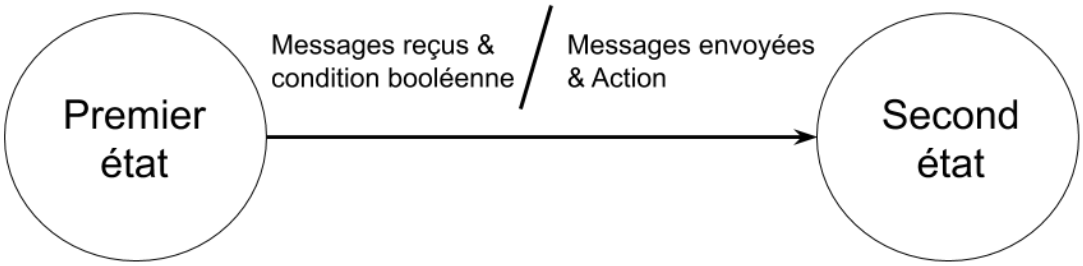

Figure 1: Diagramme d'état

Modélisez le diagramme d'état en utilisant la notation décrite à la Figure 1.

2.**Système de Contrôle d'Ascenseur** Vous trouverez sur Moodle le fichier *lift.oz* qui est l'implémentation du Système Contrôle d'Ascenseur. Supposons qu'on le rédéfinit la procédure `{Bulding}` de la façon suivante:

```
proc {Building FN LN ?Floors ?Lifts} C in
   {Controller C}
   Lifts={MakeTuple lifts LN}
   for I in 1..LN do
      Lifts.I={Lift I state(1 nil false) C Floors}
   end
   Floors={MakeTuple floors FN}
   for I in 1..FN do
      Floors.I={Floor I state(false) Lifts}
   end
end
```

(i.e On associe le même contrôleur à tout les ascenceurs du building). Quel impact aurait ce changement sur le Système de Contrôle d'Ascenseur.

3.**Abstraction**. Supposez qu'on utilise modélise le contrôleur par une simple abstraction à la place d'utiliser un port object.

```
proc {Controller Msg}
   case Msg
   of step(Lid Pos Dest) then
      if Pos<Dest then
     {Delay 1000} {Send Lid 'at'(Pos+1)}
      elseif Pos>Dest then
     {Delay 1000} {Send Lid 'at'(Pos-1)}
      end
   end
end
```

- Changez le code source de façon à ce qu'il corresponde à la nouvelle définition du contrôleur.
- Quel serait l'impact d'un tel changement sur le Système de Contrôle d'Ascenseur?
- Y-a-t'il des changements sur le diagramme d'état?

4.**Indépendance**. Supposez que l'on veuille des ascenseurs autonomes qui n'ont pas besoin d'un contrôleur. Chaque ascenseur, de façon autonome, décide où aller. Avec ce nouveau design, il n'y a plus besoin du message sortant *step* et du message rentrant *at*. Comment implémenteriez-vous un tel système sans perdre les propriétés du Système de Contrôle d'Ascenseur? Par exemple, l'implémentation actuelle permet à un ascenseur de recevoir un message alors qu'il décide où aller. Dessinez le diagramme d'état.

5.**Ordonnanceur amélioré**. L'ordonnanceur du Système de Contrôle d'Ascenseur manque d'efficacité. Par exemple, pour les 3 appels d'ascenseur successives aux étages 1,5 et 3 , l'ascenceur va d'abord à l'étage 1 puis au 5 et enfin au 3 alors qu'il devrait s'arrêter à l'étage 3 avant d'aller au 5. On veut améliorer l'ordonnanceur pour que le système s'effectue de la façon décrite. Comment implémenteriez vous cela? Dessinez le diagramme d'état. (Indice: Séparez le bouton d'appel en appel vers le haut et appel vers le bas)