Exercices de TME

Pour ces exercices nous allons utiliser DEMO-S5, un logiciel permettant de manipuler les notions vues en logique épistémique S5, et en particulier de faire évoluer les structures de Kripke à la suite d'*annonces publiques*. Le logiciel a été développé par Jan van Eijck, chercheur à l'Université d'Amsterdam (`http://homepages.cwi.nl/~jve/software/demo_s5/`).

Créez un répertoire de travail pour ce TME, puis rendez-vous sur Piazza et téléchargez les fichiers `DEMO-S5.hs` et `EREL.hs`. Ce sont des fichiers en Haskell, un langage fonctionnel. Pour lancer l'interpréteur, il suffira d'éxécuter dans un terminal la commande `ghci nomfichier.hs`. Pour les deux exercices qui suivent, nous avons préparé les définitions (création des structures, etc.) utiles dans les fichiers respectifs `tme4-hexa.hs` et `tme4-cheryl.hs`.
Pour information : le fichier `DEMO-S5.pdf` donne les définitions de base utilisables, ainsi que plusieurs autres exemples de modélisation de problèmes (dont les "enfants sales" vus en cours). Un petit mémo rappelle les principales commandes à la fin de ce sujet.

Exercice 1 – Le modèle Hexa – *inspiré de Hans van Ditmarsch.*

On considère ici le simple jeu suivant : il y a trois cartes : une rouge (r), une jaune (j), une verte (v), et trois joueurs a, b, et c. Chaque joueur reçoit une carte qu'il ne montre pas aux autres joueurs.

En chargeant `tme4-hexa.hs` dans l'interpréteur Haskell (commande `ghci tme4-hexa.hs`), vous avez accès à la définition de la structure de Kripke initiale et à plusieurs définitions utiles. En particulier, on utilisera des propositions de type `holds_b_verte` pour exprimer que l'agent b détient la carte verte.

1. Représentez la structure de Kripke représentant cette situation. Vérifiez votre structure en affichant le modèle initial (`model0`) dans l'interpréteur.
2. En partant de la structure initiale à chaque fois, testez les effets des annonces publiques suivantes :
   (i) L'agent a annonce : "je sais que b possède la carte verte"
   `upd_pa model0 (Kn a holds_b_verte)`
   (ii) L'agent a annonce : "je ne sais pas que b possède la carte verte"
   `upd_pa model0 (Ng (Kn a holds_b_verte))`
   Expliquez les modifications observées dans la structure de Kripke.

3* Les séquences d'annonces suivantes sont envisagées.
   (i) la première séquence est la suivante :
   - le joueur b annonce qu'il ne possède pas la carte jaune ;
   - le joueur a annonce qu'il ne sait pas que b possède la carte rouge ;
   - le joueur a annonce qu'il ne possède pas la carte jaune.

   (ii) la deuxième séquence est la suivante :
   - le joueur c annonce qu'il ne possède pas la carte jaune ;
   - le joueur a annonce qu'il ne sait pas que le joueur b sait qu'il possède la carte jaune ;
   - le joueur c annonce qu'il sait que le joueur b ne possède pas la carte verte.

   Considérez ces deux séquences et indiquez si à la fin les joueurs peuvent savoir quelle carte possède chacun. Pour la deuxième séquence, indiquez les modifications qui sont apportées à chaque étape.
4. Selon vous, le résultat final dépend-il de l'ordre dans lequel les annonces sont effectuées ? Vérifiez votre réponse en testant d'autres séquences.

Exercice 2 – L'anniversaire de Cheryl – *Examen 1ère session 2015-2016*

Cet exercice a été posé en 2015 lors d'un examen de mathématiques "Singapore and Asian Schools Math Olympiad" (voir par exemple `https://en.wikipedia.org/wiki/Cheryl's_Birthday`*) et il fut largement repris par la suite dans les médias.*

Cheryl ne souhaite pas révéler sa date d'anniversaire à ses deux nouveaux amis, Albert et Bernard. Par contre, elle leur donne 10 dates (jour/mois) possibles, qui sont indiquées dans la table suivante :

Mai			15		16					19
Juin							17		18	
Juillet	14				16					
Août	14		15				17			

Cheryl parle ensuite séparément à Albert, à qui elle révèle seulement le mois de son anniversaire, puis à Bernard à qui elle révèle seulement le jour de son anniversaire.

1. Donnez la structure de Kripke où les mondes possibles sont les dates d'anniversaires possibles, avec les relations d'accessibilité correspondant à Albert et à Bernard. Chargez `tme4-cheryl.hs` et vérifiez votre structure en affichant le modèle initial (`model0`) dans l'interpréteur.

2* Comment définiriez-vous le fait, pour un agent i, de connaitre la date d'anniversaire ? Inspectez le code de `tme4-cheryl.hs` pour vérifier que la définition est conforme à votre intuition.

3. Albert et Bernard font les déclarations publiques suivantes :

   (i) Albert : "Je ne sais pas quelle est la date d'anniversaire de Cheryl, mais je sais que Bernard ne le sait pas non plus".

   (ii) Bernard : "Maintenant je sais quelle est la date d'anniversaire de Cheryl".

   (iii) Albert : "Alors je sais moi aussi quelle est la date d'anniversaire de Cheryl".

   Sur le modèle de l'exercice précédent, modélisez les annonces publiques réalisées et observez comment la structure de Kripke évolue après chacune. Quelle est la date d'anniversaire de Cheryl ?

4. Certaines parties des annonces (i), (ii) et (iii) sont-elles superflues ?

Récréation

Exercice 3 *Les as et les huits.*

Placez vous en groupes de 3 joueurs et jouez au jeu des as et des huits. Après la distribution des cartes, chaque joueur doit annoncer à tour de rôle (de manière sincère) : soit (1) qu'il ne sait pas quelles cartes il possède, soit (2) qu'il sait qu'il a une paire (et dire laquelle) ou qu'il sait qu'il a un as et un huit. La partie se poursuit jusqu'à ce qu'un joueur puisse annoncer (2), ou que chaque joueur ait parlé deux fois.

1* Répétez le jeu au moins 9 fois en modifiant à chaque le joueur qui débute les annonces. Notez à la fin de chaque partie la donne, et les annonces effectuées ;

2. Pour les parties qui se sont terminées par un échec (aucun joueur n'a su déterminer ses cartes après deux tours), essayez de voir rétrospectivement ensemble si vous auriez pu déterminer ces cartes ;

3. Essayez de suivre le déroulement d'une ou plusieurs parties en vous appuyant sur la structure de Kripke construite en TD.

Haskell et DEMO	
`ghci nomFichier.hs`	lance l'interpréteur haskell et permet d'entrer des commandes interactives
`:quit`	permet de quitter l'interpréteur
`x = expression`	permet de déclarer et initialiser la variable x
`x == y`	teste l'égalité des valeurs des variables x et y
`x`	affiche la valeur de x —lorsque x est un modèle de Kripke, l'affichage débute par *Mo* puis est suivi par la liste des mondes possibles
`(Kn a formule)`	pour représenter que l'agent a sait que *formule*
`(Ng formule)`	construit la négation de *formule*
`upb_a model0 formule`	construit un nouveau modèle, obtenu par mise à jour de *model0* après l'annonce publique de *formule*