Université Paris 7 DESS Logiciels Fondamentaux
U.F.R d'informatique Année 2002-2003

TD 13 – Model Checking Symbolique - Automates Temporises

1 Model Checking Symbolique

On s'interesse au systeme suivant à deux compteurs :

- $c_r = 0 \wedge c_l = 0 \rightarrow^{g_r} c_r + +$
- $c_r = 0 \rightarrow^{g_l} c_l + +$
- $c_l > 0 \rightarrow^{r_l} c_l - -$
- $c_r > 0 \rightarrow^{r_r} c_r - -$

Peut on utiliser SMV pour ce systeme ? Spin ?

Ce systeme decrit l'acces à une ressource ou une personne peut etre en train d'ecrire, et que plusieurs peuvent lire. Pour garder une coherence on souhaite qu'au plus une personne puisse ecrire en meme temps, et qu'il n'y a ait pas de lecteur au moment ou quelqu'un ecrit.

Donc en particulier : $(c_r = 0 \wedge c_l = 0) \Rightarrow AG(\neg(c_r > 0 \wedge c_l > 0)) = \psi$

Avec les techniques d'analyse en arriere, verifier cette formule.

2 Automates Temporises

- Rappels et definition - Automates finis ?
- Modeliser les 4 composants du Box-sorter Unit

3 Pour aller plus loin

Parfois les modeles etudies ne rentrent pas dans le cadre imposé. Supposons que le cahier des charges impose qu'il y ait une occurence unique hor boucle d'une operation $x \leftarrow x - c$ sur une horloge. Peut on tout de meme utiliser UPPAAL ? Meme question avec $x \leftarrow y - c$. Meme question quand on peut borner a priori le nombre d'operation $x \leftarrow x - c$ le long d'une execution. Meme question quand on peut borner a priori le nb de $x \leftarrow x - c$ entre 2 remises a zero de x.

4 UPPAAL - Glance

Nous allons utiliser UPPAAL sur l'exemple du tapis roulant.

Le programme `xuppaal` est disponible sur `cunillet` dans le répertoire `/usr/local/Uppaal/bin` que vous pouvez ajouter à votre `PATH`.

4.1 La Représentation des Automates

Dans son menu `Run`, `xuppall` permet de faire appel à `autograph` pour dessiner les automates. La première chose à faire est d'y créer une fenêtre : `Window - Create`.

Reprenez l'exemple du piston.

Pour créer les états de l'automates, on construit les sommets du graphe associé (puisque `autograph` est à l'origine dédié aux graphes) : dans le menu `ObjectEdit`, sélectionner `vertice` puis `create`. Vous pouvez ensuite dessiner un certain nombre d'états de l'automate dans votre fenêtre. Dans ce même menu, vous pouvez choisir des options pour supprimer les noeuds, ou les déplacer, ou définir les états qui sont initiaux. Le bouton du milieu de la souris permet de déplacer un noeud.

Les noms des états et les invariants sont des étiquettes attachés aux états. Les noms doivent être sans espaces, et les invariants présentés entre parenthèse, sous forme de liste séparée par des virgules. Dans le menu `label` sélectionnez `create default` puis cliquez sur un état.

Pour créer les transitions (arêtes), `objectsEdit`, `Edges`, `create`. Puis choisissez un noeud de depart. Gardez le bouton de la souris enfoncé, eventuellement relachez le bouton sur un point intermédiaire, jusqu'au choix de l'état d'arrivée.

Vous étiquetterez les transitions de la même facon que les états. Les gardes sur une ligne, les actions de synchronisation sur une autre. Le test de l'égalité se fait avec `==`, et l'affectation avec `:=`.

Pour définir un automate, il reste à englober votre dessin dans une boite. `ObjectsEdit`,`Boxes`, `Create` et selectionner la partie qui vous interesse, tout en gardant le bouton appuyé du coin supérieur gauche au coins inférieur droit.

Donnez un nom à votre automate en lui attibuant un label.

Vous pouvez le sauver, en sélectionnant `Files`, `Save atg as` puis en cliquant sur la partie que vous voulez sauvegarder.

On peut aussi crér un réseau d'automate, en dessinant d'autres automates sur le même fichier. Pour coordonner ces automates, on crée une boite, d'étiquette `Config`. Qui contiendra les informations nécessaires : horloges, variables et canaux de synchronisation.

Un exemple sera trouve dans /usr/local/Uppaal/exemples/boxes.atg. Vous pourrez constater l'interet d'utiliser `autograph`, en sélectionnant dans `uppaal Run - Show Model`.

Un exemple de propriétés à vérifier est donné dans le fichier `/usr/local/Uppaal/examples/boxes.q`, que vous pouvez afficher dans `uppal` en sélectionnant `Run - Show Req. Spec.`

Pour faire apparaitre tout les labels dans `autograph`, selectionnez `Label - Struct - Show/hide all` puis cliquez dans la fenêtre. Les noms des états apparaissent, onpeut à présent interpréter le fichie de spécification.

Vous pouvez alors lancer la vérification dans `uppaal` avec `Run - Verification`

Vous pouvez proposer une formule qui ne soit pas vérifiée pour noter la réaction de l'outil.

Correction 1 :

On transforme $A \Rightarrow B$ en $\neg A \vee B$, puis $AG\phi$ en $\neg EF(\neg\phi)$ $(c_r = 0 \wedge c_l = 0) \Rightarrow AG(\neg(c_r > 0 \wedge c_l > 0))$
$\neg(c_r = 0 \wedge c_l = 0) \vee AG(\neg(c_r > 0 \wedge c_l > 0))$
$\neg(c_r = 0 \wedge c_l = 0) \vee \neg EF(c_r > 0 \wedge c_l > 0) = \psi$

Puis on calcule le EF comme un point fixe. Pour retrouver la methode on ecrit $EF\phi = \phi \vee EXEF\phi$. Donc on part de ϕ, on calcule $pred(\phi)$, on ajoute etc... tant que c'est possible.

On rapelle comment on calcule le pred d'un ensemble S pour une transition τ de la forme $g(x) \rightarrow \vec{x} := f(\vec{x})$:

$pred_\tau(S) := \exists X', S(X') \wedge X' = f(X) \wedge g(X)$, on peut eliminer la quantification :

$pred_\tau(S) := g(X) \wedge S[X/f(X)]$

maintenant on reprend le calcul de $EF(c_r > 0 \wedge c_l > 0)$:

$pred^*_{g_r}$: $(c_r > 0 \wedge c_l > 0) \vee$ $(c_r = 0 \wedge c_l = 0) \wedge ((c_r + 1) > 0 \wedge c_l > 0)$. Stop, la derniere formule est fausse.

$pred^*_{g_l}$: $(c_r > 0 \wedge c_l > 0) \vee$ $(c_r = 0) \wedge (c_r > 0 \wedge (c_l + 1) > 0)$. Stop, la derniere formule est fausse.

$pred^*_{r_l}$: $(c_r > 0 \wedge c_l > 0) \vee$ $(c_l > 0) \wedge (c_r > 0 \wedge (c_l - 1) > 0)$. La derniere formule est subsumée par la premiere, le point fixe est atteind.

$pred^*_{r_r}$: $(c_r > 0 \wedge c_l > 0) \vee$ $(c_r > 0) \wedge ((c_r - 1) > 0 \wedge c_l > 0)$. La derniere formule est subsumée par la premiere, le point fixe est atteind.

Bref $EF(c_r > 0 \wedge c_l > 0)$ c'est $c_r > 0 \wedge c_l > 0$, et

$psi := \neg(c_r = 0 \wedge c_l = 0) \vee \neg(c_r > 0 \wedge c_l > 0)$ c'est donc tout les etats. Ma formule etait vraie.

Vous pouvez essayer aussi avec $AG(c_r \leq 1)$ (ca devrait marcher j'espere...)