Modèles de calcul et Validation d'algorithmes raphael.charrondiere@inria.fr
L3 MIAGE, UGA
2019/2020

DM : IMP parallèle

*On rappelle que les réponses doivent être **justifiées**, cependant une justification n'a pas à être longue pour être de qualité. Ce devoir maison est individuel, doit être rendu sous forme manuscrite. Il sera noté. En cas de non rendu la note sera automatiquement nulle, et chaque jour de retard entraînera une pénalité.*

Conseil : N'hésitez pas à utiliser des exemples sur votre brouillon.

Important : Un exemple n'est qu'un cas particulier. Pour la question 1.3.2, si la réponse est non, un contre exemple suffit, mais si la réponse est oui, un exemple ne sert à rien.

On propose d'étendre le langage IMP avec du parallélisme. Ainsi on ajoute une commande $c_1 || \ c_2$. Il faudra alors définir la sémantique à grand pas et à petit pas. Après avoir vu quelques propriétés, on comparera les deux sémantiques.

1 Le cas où || isole

1.1 Règles d'inférence

Commençons par la sémantique à grands pas pour la commande $c_1 || \ c_2$. Les prémisses utilisables sont de la forme $\sigma, c \Downarrow \sigma'$, on ne peut donc pas décomposer les commandes c_i. On considère alors que l'une des deux commandes est exécutée d'un bloc, avant l'autre.

Donner les règles d'inférences correspondantes.

1.2 Déterminisme

A priori le parallélisme introduit du non-déterminisme comme vous l'avez sans doute déjà vu avec d'autres langages.

Pouvez-vous le vérifier ?

1.3 Propriétés mathématiques

Commençons par définir une relation binaire $\equiv$. Celle-ci traduira le fait que deux commandes se comportent de la même façon. De manière formelle, $c_1 \equiv c_2$ si et seulement si, pour tout états mémoires σ, σ' on a $\sigma, c_1 \Downarrow \sigma' \Leftrightarrow \sigma, c_2 \Downarrow \sigma'$.

1.3.1 Commutativité

Montrez que l'opérateur || est commutatif, c'est-à-dire, pour toutes commandes c_1, c_2 on a $c_1 || \ c_2 \equiv c_2 || \ c_1$

1.3.2 Associativité

Idéalement || devrait être aussi associatif, c'est-à-dire $(c_1 || \ c_2) || \ c_3 \equiv c_1 || \ (c_2 || \ c_3)$, ce qui signifierait que l'on parallélise bien toutes les commandes.

Mais est-ce bien le cas ?

2 Pas à pas

La sémantique à grand pas ne permettait pas d'entrelacement, l'exécution de $c_1 || \ c_2$ doit terminer c_1 avant de commencer c_2 où vice-versa. La sémantique à petits pas permet de suivre plus finement le déroulement du programmes. On n'introduira pas de vrai parallélisme dans cette partie. Virtuellement, des parties des commandes c_1 et c_2 seront exécutées l'une après l'autre.

2.1 Règles d'inférences

Donnez les règles d'inférences nécessaires pour définir ||.

Attention ! Un programme qui a terminé ne contient plus de || !

2.2 Comparaison de la sortie

On a vu en TD, $\sigma, c \Downarrow \sigma' \Leftrightarrow \sigma, c \rightarrow^* \sigma', \textbf{skip}$, c'est-à-dire que grand pas vaut petit pas. Par commodité, on définira la relation $\searrow$ par $\sigma, c \searrow \sigma'$ ssi $\sigma, c \rightarrow^* \sigma', \textbf{skip}$

La question est alors de savoir si $\Downarrow$ et $\searrow$ sont équivalents : est-ce-que ce que nous avons vu en TD est toujours vrai ?

2.3 Commandes équivalentes

De même que l'on avait définit $\equiv$, nous définissons $\approx$: $c_1 \approx c_2$ si et seulement si, pour tout états mémoires σ, σ' on a $\sigma, c_1 \searrow \sigma' \Leftrightarrow \sigma, c_2 \searrow \sigma'$.

Les opérateurs sont-ils équivalents ? (Les propriétés ci-dessous sont-elles vraies ?)

— $\sigma, c \Downarrow \sigma' \Rightarrow \sigma, c \searrow \sigma'$

— $\sigma, c \searrow \sigma' \Rightarrow \sigma, c \Downarrow \sigma'$

2.4 Propriétés mathématiques de || en petits pas

Attention Les questions suivantes sont complexes ! La notation sera donc moins stricte, on n'attend pas une réponse parfaitement bien justifiée, les bonnes idées seront récompensées. Si vous avez eu des difficultés avant, il est déconseillé d'entamer cette partie et il vaut mieux revoir les questions précédentes.

2.4.1 Commutativité

L'opérateur || est-il commutatif en petit pas, c'est-à-dire, a-t-on pour toutes commandes c_1, c_2, $c_1 || \ c_2 \approx c_2 || \ c_1$

2.4.2 Associativité

A-t-on $(c_1 || \ c_2) || \ c_3 \approx c_1 || \ (c_2 || \ c_3)$?