Théorie de la programmation — DM 2
Transformations vers Fun

à faire seul(e), à rendre le mardi 7 janvier 2020 à 10h15 au plus tard
vous pouvez déposer le devoir dans le casier à courrier de D. Hirschkoff

1 Ajout des couples

On s'intéresse à FUN*, l'extension de FUN définie de la manière suivante :

Expressions :

$$e \quad ::= \quad k \;\big|\; e_1 + e_2 \;\big|\; x \;\big|\; \texttt{fun}\ x\ \texttt{->}\ e \;\big|\; e_1\ e_2 \;\big|\; (e_1, e_2) \;\big|\; \pi_1(e) \;\big|\; \pi_2(e)$$

Valeurs :

$$v \quad ::= \quad k \;\big|\; \texttt{fun}\ x\ \texttt{->}\ e \;\big|\; (v_1, v_2)$$

Réduction :

$$\textsc{Rpk}\ \frac{}{k_1 + k_2 \to k}\ k \text{ est la somme de } k_1 \text{ et } k_2 \qquad \textsc{Rpd}\ \frac{e_2 \to e_2'}{e_1 + e_2 \to e_1 + e_2'} \qquad \textsc{Rpg}\ \frac{e_1 \to e_1'}{e_1 + k_2 \to e_1' + k_2} \qquad \textsc{R}\beta\ \frac{}{(\texttt{fun}\ x\ \texttt{->}\ e)\ v_2 \to e[v_2/x]}$$

$$\textsc{Rad}\ \frac{e_2 \to e_2'}{e_1\ e_2 \to e_1\ e_2'} \qquad \textsc{Rag}\ \frac{e_1 \to e_1'}{e_1\ v_2 \to e_1'\ v_2} \qquad \textsc{Rcd}\ \frac{e_2 \to e_2'}{(e_1, e_2) \to (e_1, e_2')} \qquad \textsc{Rcg}\ \frac{e_1 \to e_1'}{(e_1, v_2) \to (e_1, v_2')}$$

$$\textsc{Rc1}\ \frac{}{\pi_1((v_1, v_2)) \to v_1} \qquad \textsc{Rc2}\ \frac{}{\pi_2((v_1, v_2)) \to v_2} \qquad \textsc{Rcp1}\ \frac{e \to e'}{\pi_1(e) \to \pi_1(e')} \qquad \textsc{Rcp2}\ \frac{e \to e'}{\pi_2(e) \to \pi_2(e')}$$

On admet les propriétés suivantes, qui ne sont pas très difficiles à établir en soi, mais ne constituent pas le sujet central de ce devoir :

- La relation $\to$ est *déterministe* : si $e \to e_1$ et $e \to e_2$, alors $e_1 = e_2$.
- La relation $\to^+$ (clôture transitive de $\to$) satisfait le pendant des règles RPD, RPG, RAD, RAG, RCD, RCG. Ainsi, par exemple, si $e_2 \to^+ e_2'$, alors $e_1\ e_2 \to^+ e_1\ e_2'$ (pendant de la règle RAD).

1. **Traduction de Fun* vers Fun.** Comme en cours, on définit la traduction d'une expression FUN* e, notée $[\![e]\!]$. Le résultat de la traduction, $[\![e]\!]$, est une expression FUN.

   Pour les constructions qui n'utilisent pas les couples, la traduction est "plan plan" :

   $$[\![k]\!] = k \qquad [\![e_1 + e_2]\!] = [\![e_1]\!] + [\![e_2]\!] \qquad [\![\texttt{fun}\ x\ \texttt{->}\ e]\!] = \texttt{fun}\ x\ \texttt{->}\ [\![e]\!]$$

   $$[\![e_1\ e_2]\!] = [\![e_1]\!]\ [\![e_2]\!] \qquad [\![x]\!] = x$$

   La traduction des couples est définie ainsi :

   $$[\![(e_1, e_2)]\!] = (\texttt{fun}\ x\ y\ p\ \texttt{->}\ p\ x\ y)\ [\![e_1]\!]\ [\![e_2]\!] \qquad \text{où } \{x, y, p\} \cap (V\ell(e_1) \cup V\ell(e_2)) = \varnothing$$

   (on rappelle que $(\texttt{fun}\ x\ y\ p\ \texttt{->}\ p\ x\ y) = \texttt{fun}\ x\ \texttt{->}\ (\texttt{fun}\ y\ \texttt{->}\ (\texttt{fun}\ p\ \texttt{->}\ (p\ x\ y)))$).

   Deviner la traduction des projections, $\pi_i(e)$, pour $i = 1, 2$.

2. **Simulation.** On dit qu'une relation binaire $\mathcal{R}$ sur un (unique[1]) SRA $(A, \to)$ est une *simulation* si et seulement si dès que $e_1 \mathcal{R} e_2$ et $e_1 \to e_1'$, il existe e_2' tel que $e_2 \to e_2'$ et $e_1' \mathcal{R} e_2'$.

   Pourquoi ne peut-on pas définir une simulation pour valider la traduction définie ci-dessus ?

3. **Simulation progressante.** Une relation $\mathcal{R}$ est une *simulation progressante* si $e_1 \mathcal{R} e_2$ et $e_1 \to e_1'$ impliquent l'existence de e_2' tel que $e_2 \to^+ e_2'$ et $e_1' \mathcal{R} e_2'$, où $\to^+$ est la clôture transitive de $\to$.

   (a) Définissez $\mathcal{R}_0$, la plus petite relation qui est une simulation progressante et contient le couple $(e_0, [\![\, e_0 \,]\!])$, où $e_0 = (1 + 2, (\texttt{fun}\ x \texttt{->} x + 1)\ 3)$.

   (b) Montrez que $\mathcal{T} = \{(e, [\![\, e \,]\!]) \mid e \text{ une expression } \textsc{Fun}^*\}$ est une simulation progressante.

   Erratum : Comme cela a été signalé par certains d'entre vous, ça n'est pas vrai, à cause du cas particulier où l'on inclut des couples dans des couples (et plus généralement parce que la traduction d'une valeur n'est pas toujours une valeur).

   Vous pouvez montrer où la preuve marche, et où surgit le problème. Si vous le souhaitez (bonus), vous pouvez aussi proposer une solution à ce problème (modifier la traduction, définir la plus petite simulation progressante contenant $\mathcal{T}$, ...).

2 De Fun vers Fun

On définit une traduction de Fun vers lui-même de la manière suivante :

$$[\![\, k \,]\!] = k \qquad [\![\, e_1 + e_2 \,]\!] = [\![\, e_1 \,]\!] + [\![\, e_2 \,]\!] \qquad [\![\, x \,]\!] = x\ (\texttt{fun}\ t \texttt{->} t) \qquad [\![\, \texttt{fun}\ x \texttt{->} e \,]\!] = \texttt{fun}\ x \texttt{->} [\![\, e \,]\!]$$

$$[\![\, e_1\ e_2 \,]\!] = [\![\, e_1 \,]\!]\ \ (\texttt{fun}\ z \texttt{->} [\![\, e_2 \,]\!]) \qquad \text{où } z \notin V\ell(e_2)$$

L'image de la traduction est le langage Fun vu en cours, avec comme sémantique opérationnelle la relation de réduction $\to$ vue en cours.

1. Définir une traduction $[\![\, \tau \,]\!]$ et une traduction $[\![\, \Gamma \,]\!]$ telles que $\Gamma \vdash e : \tau$ implique $[\![\, \Gamma \,]\!] \vdash [\![\, e \,]\!] : [\![\, \tau \,]\!]$.

   Prouver que l'on a cette propriété.

   On souhaite établir un résultat de simulation progressante (au sens de la partie précédente) , et le but est de "deviner" le langage de départ : la grammaire est la même que pour Fun, mais la réduction dans le langage de départ diffère, elle sera notée $\mapsto$.

2. Définir une relation de réduction $\mapsto$, entre expressions Fun, telle que l'on ait une simulation progressante entre e (exécuté selon $\mapsto$) et $[\![\, e \,]\!]$ (exécuté selon $\to$, la sémantique opérationnelle standard de Fun).

   Il est demandé d'être fair play et d'éviter les réponses inintéressantes genre la relation vide, etc.

   NB : On ne demande pas de prouver le résultat de simulation, et lisez la question suivante pour comprendre qu'on *espère* avoir le résultat de simulation, sans forcément être capable de le prouver. L'idée est que $\mapsto$ corresponde à l'exécution des expressions "telle que programmée par la traduction".

3. On souhaite maintenant montrer que $\mathcal{R} = \{(e, [\![\, e \,]\!])\}$ est une simulation progressante modulo simulation progressante :

   Soit $\mathcal{S}$ une simulation progressante. Une relation $\mathcal{R}$ est une *simulation progressante modulo $\mathcal{S}$* si $e_1 \mathcal{R} e_2$ et $e_1 \to e_1'$ impliquent l'existence de e_2' tel que $e_2 \to^+ e_2'$ et $e_1' \mathcal{R}\mathcal{S} e_2'$, où $e_1' \mathcal{R}\mathcal{S} e_2'$ ssi il existe e tel que $e_1' \mathcal{R} e$ et $e \mathcal{S} e_2'$.

[1]En cours on a vu la simulation entre deux SRA.

Indiquez quelles sont les étapes pour démontrer que $\mathcal{R}$ est une simulation progressante modulo $\mathcal{S}$, pour $\mathcal{S}$ une simulation progressante bien choisie.

Inutile d'être trop formels pour cette réponse, mais tâchez tout de même d'être convainquant(e).

4. Définir la relation de sémantique opérationnelle à grands pas correspondant à $\mapsto$. On notera celle-ci $e \downarrow v$.

5. Proposez deux exemples, raisonnablement différents l'un de l'autre, d'expressions Fun qui s'évaluent différemment suivant $\Downarrow$ (la sémantique opérationnelle à grands pas vue en cours) et $\downarrow$. Expliquez, à chaque fois, en quoi les deux évaluations diffèrent.