

INF889J Semaine 2 (Solutionnaire)

Indice. L'ordre des questions n'est pas accidentel ! Utilisez les propriétés énoncées par les questions précédentes pour résoudre les questions subséquentes.

1 Clôture réflexive et transitive

Lorsqu'on raisonne au sujet de la sémantique à petites étapes, on souhaite souvent raisonner au sujet de séquences de réductions $e_0 \mapsto e_1 \mapsto \dots \mapsto e_n$ où $n \geq 0$, c'est-à-dire, de la clôture réflexive et transitive $\mapsto^*$ de $\mapsto$. Pour faciliter le raisonnement inductif, on peut la définir par induction par les règles suivantes :

$e \mapsto^* e'$ e se réduit à e' en zéro ou plusieurs étapes

$$\frac{}{e \mapsto^* e} (\mapsto^*/\text{REFL}) \quad \frac{e \mapsto e'' \quad e'' \mapsto^* e'}{e \mapsto^* e'} (\mapsto^*/\mapsto)$$

(Ces règles définissent la clôture réflexive et transitive de $\mapsto$, mais elles s'adaptent à la clôture réflexive et transitive $\mathcal{R}^*$ de n'importe quelle relation $\mathcal{R}$.)

PROBLÈME 1. *Démontrez par induction sur la dérivation que le jugement $e \mapsto^* e'$ représente une relation transitive : si $e_1 \mapsto^* e_2$ et $e_2 \mapsto^* e_3$, alors $e_1 \mapsto^* e_3$.*

Solution 0: Nous procédons par induction sur la dérivation de $e_1 \mapsto^* e_2$ pour démontrer que pour tout e_3 , si $e_2 \mapsto^* e_3$, alors $e_1 \mapsto^* e_3$.

Cas où la dérivation de $e_1 \mapsto^* e_2$ est formée par $(\mapsto^*/\text{REFL})$. Dans ce cas, $e_1 \mapsto^* e_2$ est de la forme $e_1 \mapsto^* e_1$ (conclusion de la règle). Nous devons démontrer que $e_1 \mapsto^* e_3$. Puisque $e_1 = e_2$, on a $e_1 \mapsto^* e_3$ par l'hypothèse $e_2 \mapsto^* e_3$.

Cas où la dérivation de $e_1 \mapsto^* e_2$ est formée par $(\mapsto^*/\mapsto)$. On déduit :

- (1) $e_1 \mapsto e$ prémisse de la règle
- (2) $e \mapsto^* e_2$ prémisse de la règle
- (3) $e_2 \mapsto^* e_3$ hypothèse
- (4) pour tout e' , si $e_2 \mapsto^* e'$, alors $e \mapsto^* e'$ H.R. de 2
- (5) $e \mapsto^* e_3$ 4 et 3
- (6) $e_1 \mapsto^* e_3$ $(\mapsto^*/\mapsto)$, 1, 5

□

2 Langage d'expressions binaire

On reprend le langage d'expressions binaires de la semaine dernière :

$$b ::= 0 \mid 1 \mid b1 \mid b0 \mid \text{succ}(b)$$

Les valeurs étaient les expressions sans l'opérateur $\text{succ}(\cdot)$. On admettait plusieurs valeurs pour le même nombre naturel, par exemple, 001 val, 01 val et 1 val sont trois valeurs qui dénotent le nombre naturel 1. On raffine la définition de valeur pour aussi exclure les zéros non significatifs. Pour nous aider à définir le jugement de valeur, on utilise un jugement auxiliaire « b commParUn » qui signifie que b est une valeur qui commence par un 1. Les deux jugements sont définis par induction par les règles suivantes :

$\boxed{b \text{ commParUn}}$ b commence par un 1

$$\frac{}{1 \text{ commParUn}} \text{ (NORM/1)} \quad \frac{b \text{ commParUn}}{b0 \text{ commParUn}} \text{ (NORM/B0)} \quad \frac{b \text{ commParUn}}{b1 \text{ commParUn}} \text{ (NORM/B1)}$$

$\boxed{b \text{ val}}$ b est une valeur

$$\frac{}{0 \text{ val}} \text{ (VAL/0)} \quad \frac{b \text{ commParUn}}{b \text{ val}} \text{ (VAL/NORM)}$$

On modifie aussi notre sémantique à petites étapes pour éliminer les zéros non significatifs :

$\boxed{b \mapsto b'}$ b se réduit à b' en une étape

$$\begin{array}{c} \frac{b \mapsto b'}{b0 \mapsto b'0} \text{ (}\mapsto/0\text{)} \quad \frac{b \mapsto b'}{b1 \mapsto b'1} \text{ (}\mapsto/1\text{)} \quad \frac{b \mapsto b'}{\text{succ}(b) \mapsto \text{succ}(b')} \text{ (}\mapsto/\text{succ}\text{)} \\ \frac{}{\text{succ}(0) \mapsto 1} \text{ (}\mapsto/0\text{)} \quad \frac{}{\text{succ}(1) \mapsto 10} \text{ (}\mapsto/1\text{)} \\ \frac{}{\text{succ}(b0) \mapsto b1} \text{ (}\mapsto/b0\text{)} \quad \frac{}{\text{succ}(b1) \mapsto \text{succ}(b)0} \text{ (}\mapsto/b1\text{)} \\ \frac{}{00 \mapsto 0} \text{ (}\mapsto/00\text{)} \quad \frac{}{01 \mapsto 1} \text{ (}\mapsto/01\text{)} \end{array}$$

La question suivante nous permet de vérifier la correction partielle de notre sémantique à petites étapes, pour assurer que la réduction ne change pas le sens de l'expression :

PROBLÈME 2. Démontrez que si $b \mapsto b'$, alors $\llbracket b \rrbracket = \llbracket b' \rrbracket$. Supposons qu'on s'était trompé et qu'on avait utilisé la règle

$$\frac{}{\text{succ}(b1) \mapsto b0} \text{ (}\mapsto/b1/\text{oops}\text{)}$$

au lieu de la règle $(\mapsto/b1)$. Auriez-vous pu démontrer la propriété de correction ?

Solution 0: Soit une dérivation arbitraire de $\beta \mapsto \beta'$. Nous procédons par induction sur cette dérivation pour démontrer que $\llbracket \beta \rrbracket = \llbracket \beta' \rrbracket$.

Cas $(\mapsto/0)$. Dans ce cas, $\beta \mapsto \beta'$ a la forme $b0 \mapsto b'0$ où $b \mapsto b'$. Nous avons comme hypothèse de récurrence que $\llbracket b \rrbracket = \llbracket b' \rrbracket$. Nous calculons :

$$\llbracket \beta \rrbracket = \llbracket 0b \rrbracket = 2\llbracket b \rrbracket = 2\llbracket b' \rrbracket = \llbracket 0b' \rrbracket = \llbracket \beta' \rrbracket.$$

Cas $(\mapsto/1)$. Dans ce cas, $\beta \mapsto \beta'$ a la forme $b1 \mapsto b'1$ où $b \mapsto b'$. Nous avons comme hypothèse de récurrence que $\llbracket b \rrbracket = \llbracket b' \rrbracket$. Nous calculons :

$$\llbracket \beta \rrbracket = \llbracket 1b \rrbracket = 2\llbracket b \rrbracket + 1 = 2\llbracket b' \rrbracket + 1 = \llbracket 1b' \rrbracket = \llbracket \beta' \rrbracket.$$

Cas $(\mapsto/\text{succ})$. Dans ce cas, $\beta \mapsto \beta'$ a la forme $\text{succ}(b) \mapsto \text{succ}(b')$ où $b \mapsto b'$. Nous avons comme hypothèse de récurrence que $\llbracket b \rrbracket = \llbracket b' \rrbracket$. Nous calculons :

$$\llbracket \beta \rrbracket = \llbracket \text{succ}(b) \rrbracket = \llbracket b \rrbracket + 1 = \llbracket b' \rrbracket + 1 = \llbracket \text{succ}(b') \rrbracket = \llbracket \beta' \rrbracket.$$

Cas $(\mapsto/0)$. Dans ce cas, $\beta \mapsto \beta'$ a la forme $\text{succ}(0) \mapsto 1$. Nous calculons :

$$\llbracket \beta \rrbracket = \llbracket \text{succ}(0) \rrbracket = \llbracket 0 \rrbracket + 1 = 1 = \llbracket 1 \rrbracket = \llbracket \beta' \rrbracket.$$

Cas $(\mapsto/1)$. Dans ce cas, $\beta \mapsto \beta'$ a la forme $\text{succ}(1) \mapsto 10$. Nous calculons :

$$\llbracket \beta \rrbracket = \llbracket \text{succ}(1) \rrbracket = \llbracket 1 \rrbracket + 1 = 2 = \llbracket 10 \rrbracket = \llbracket \beta' \rrbracket.$$

Cas $(\mapsto/b0)$. Dans ce cas, $\beta \mapsto \beta'$ a la forme $\text{succ}(b0) \mapsto b1$. Nous calculons :

$$\llbracket \beta \rrbracket = \llbracket \text{succ}(b0) \rrbracket = \llbracket b0 \rrbracket + 1 = 2\llbracket b \rrbracket + 1 = 2\llbracket b' \rrbracket + 1 = \llbracket b'1 \rrbracket = \llbracket \beta' \rrbracket.$$

Cas $(\mapsto/b1)$. Dans ce cas, $\beta \mapsto \beta'$ a la forme $\text{succ}(b1) \mapsto \text{succ}(b)0$. Nous calculons :

$$\llbracket \beta \rrbracket = \llbracket \text{succ}(b1) \rrbracket = \llbracket b1 \rrbracket + 1 = (2\llbracket b \rrbracket + 1) + 1 = 2(\llbracket b \rrbracket + 1) = 2\llbracket \text{succ}(b) \rrbracket = \llbracket \text{succ}(b)0 \rrbracket.$$

Cas $(\mapsto/00)$. Dans ce cas, $\beta \mapsto \beta'$ a la forme $00 \mapsto 0$. Nous calculons :

$$\llbracket \beta \rrbracket = \llbracket 00 \rrbracket = 2\llbracket 0 \rrbracket = 0 = \llbracket 0 \rrbracket = \llbracket \beta' \rrbracket.$$

Cas $(\mapsto/01)$. Dans ce cas, $\beta \mapsto \beta'$ a la forme $01 \mapsto 1$. Nous calculons :

$$\llbracket \beta \rrbracket = \llbracket 01 \rrbracket = 2\llbracket 0 \rrbracket + 1 = 0 + 1 = \llbracket 1 \rrbracket = \llbracket \beta' \rrbracket.$$

Nous n'aurions pas pu démontrer la propriété avec la règle $(\mapsto/b1/\text{oops})$, car elle permet de déduire $\text{succ}(b1) \mapsto b0$, mais :

$$\llbracket \text{succ}(b1) \rrbracket = \llbracket b1 \rrbracket + 1 = 2\llbracket b \rrbracket + 1 \neq 2\llbracket b \rrbracket = \llbracket b0 \rrbracket. \quad \square$$

COROLLAIRE 1. Si $b \mapsto^* b'$, alors $\llbracket b \rrbracket = \llbracket b' \rrbracket$.

Observez que votre sémantique dénotationnelle détermine une bijection entre les valeurs et les nombres naturels :

- elle est injective quand restreinte aux valeurs : pour tout b_1 et tout b_2 tels que b_1 val et b_2 val, si $\llbracket b_1 \rrbracket = \llbracket b_2 \rrbracket$, alors $b_1 = b_2$;
- elle est surjective quand restreinte aux valeurs : pour tout $n \in \mathbb{N}$, il existe un b tel que b val et $\llbracket b \rrbracket = n$.

En effet, chaque b tel que b val est la représentation binaire canonique (sans zéro non significatifs) d'un nombre naturel, et nous savons que chaque nombre naturel a une représentation binaire canonique.

On peut utiliser notre sémantique dénotationnelle pour démontrer que la valeur obtenue suite à une séquence de réductions est unique :

PROBLÈME 3. Dédisez que pour tout b, b_1 et b_2 , si $b \mapsto^* b_1$, b_1 val, $b \mapsto^* b_2$ et b_2 val, alors $b_1 = b_2$.

Solution 0: Par le corollaire 1, $\llbracket b_1 \rrbracket = \llbracket b \rrbracket = \llbracket b_2 \rrbracket$. Par l'observation et le fait que $\llbracket \cdot \rrbracket$ est injectif sur les valeurs, $b_1 = b_2$. $\square$

Vous pouvez tenir pour acquis que :

PROPOSITION 2. Pour tout b , il existe un b' tel que $b \mapsto^* b'$ et b' val.

On peut déduire que le b' donné par la proposition 2 est unique grâce au problème 3. Pour nos applications, on s'intéresse seulement à observer le résultat de la simplification d'une expression binaire. On écrira $\{b\}$ pour cette **observation** b' , où b' est l'unique expression telle que $b \mapsto^* b'$ et b' val.

On dit que deux expressions binaires b_1 et b_2 sont **opérationnellement équivalentes**, $b_1 \simeq b_2$, si $\{C[b]\} = \{C[b']\}$ pour tout contexte $C[\cdot]$.¹ Cela signifie qu'aucun contexte ne peut les différencier en termes de réduction à une valeur. Rappelez-vous que deux expressions b_1 et b_2 sont **dénotationnellement équivalentes**, $b_1 \equiv b_2$, si $\llbracket b_1 \rrbracket = \llbracket b_2 \rrbracket$, et qu'une sémantique dénotationnelle est **compositionnelle**, donc si $b_1 \equiv b_2$, alors $C[b_1] \equiv C[b_2]$ pour tout contexte $C[\cdot]$.

1. Les contextes $C[\cdot]$ pour notre langage sont générés par la grammaire $C[\cdot] ::= [\cdot] \mid C[\cdot]0 \mid C[\cdot]1 \mid \text{succ}(C[\cdot])$. On écrit $C[b]$ pour le résultat de "remplir le trou" du contexte par b .

On dit qu'une sémantique dénotationnelle est **pleinement abstraite** (*fully abstract*) par rapport à une sémantique opérationnelle si l'équivalence dénotationnelle coïncide avec l'équivalence opérationnelle.

PROBLÈME 4 (PLEINE ABSTRACTION). *Démontrez que pour tout b_1 et b_2 , $b_1 \equiv b_2$ si et seulement si $b_1 \simeq b_2$.*

Solution 0: Supposons que $b_1 \equiv b_2$. Nous démontrons que $b_1 \simeq b_2$. Soit un contexte $C[\cdot]$ arbitraire. Nous devons démontrer que $\llbracket C[b_1] \rrbracket = \llbracket C[b_2] \rrbracket$. Observons que si $b_1 \equiv b_2$, alors $\llbracket C[b_1] \rrbracket = \llbracket C[b_2] \rrbracket$ par la compositionnalité. De plus,

$$\llbracket \llbracket C[b_1] \rrbracket \rrbracket = \llbracket C[b_1] \rrbracket = \llbracket C[b_2] \rrbracket = \llbracket \llbracket C[b_2] \rrbracket \rrbracket$$

par le corollaire. Cependant, notre sémantique dénotationnelle est injective sur les valeurs, donc $\llbracket C[b_1] \rrbracket = \llbracket C[b_2] \rrbracket$. On conclut que $b_1 \simeq b_2$.

Supposons que $b_1 \simeq b_2$. Nous démontrons que $b_1 \equiv b_2$. Par la définition de $\simeq$ avec le contexte $[\cdot]$, $\llbracket b_1 \rrbracket = \llbracket b_2 \rrbracket$, et par le corollaire, $\llbracket b_1 \rrbracket = \llbracket \llbracket b_1 \rrbracket \rrbracket = \llbracket \llbracket b_2 \rrbracket \rrbracket = \llbracket b_2 \rrbracket$. On conclut que $b_1 \equiv b_2$. $\square$

PROBLÈME 5. *Démontrez que les deux expressions suivantes sont opérationnellement équivalentes, c'est-à-dire, que $b_1 \simeq b_2$, où :*

$$b_1 = \text{succ}(\text{succ}(\text{succ}(\text{succ}(\text{succ}(10)100)1001)00)0)10$$

$$b_2 = \text{succ}(\text{succ}(\text{succ}(\text{succ}(11\ 100)1001)00)0)10$$

(Les espaces sont là uniquement pour faciliter la lecture.)

Solution 0: On démontre en premier que $\text{succ}(10) \equiv 11 : \llbracket \text{succ}(10) \rrbracket = \llbracket 10 \rrbracket + 1 = 2\llbracket 1 \rrbracket + 1 = \llbracket 11 \rrbracket$. Soit le contexte $C[\cdot] = \text{succ}(\text{succ}(\text{succ}(\text{succ}([\cdot]100)1001)00)0)10$. Par la compositionnalité de la sémantique dénotationnelle, on peut déduire que $b_1 = C[\text{succ}(10)] \equiv C[11] = b_2$, donc $b_1 \simeq b_2$ par la pleine abstraction. $\square$