Composition d'Informatique
Les Principes des Langages de Programmation (INF 321)
Promotion 2011

Sujet proposé par Eric Goubault

12 juillet 2012

Les questions qui suivent sont independantes et peuvent être traitees dans n'importe quel ordre. Les questions 2.d et 4.h sont optionnelles. Il est conseille aux etudiants de lire rapidement tout le sujet avant de demarrer. En n, on attachera une grande importance a la clarte, a la precision et a la concision de la redaction et des explications.

On considere dans ce probleme l'implementation d'une feuille de calcul a la Excel. D'un point de vue logique, une feuille de calcul est un tableau bi-dimensionnel

$$C = (c_{i,j})_{i=0,\ldots,n-1;j=0,\ldots,m-1}$$

dont les elements sont des formules arithmetiques ou des valeurs numeriques (rentrees par un utilisateur) ou des cellules non de nies (on utilise alors le symbole $\perp$). Le but du probleme est de donner une semantique aux calculs sur cette feuille, et d'implementer ces calculs en Java.

Pour xer les idees, voici un exemple de feuille simple, avec $n = 2$ est $m = 2$:

$$\mathcal{C} = \begin{array}{|c|c|}\hline 1 & c_{1,1} \\ \hline \perp & c_{0,0} - c_{0,1} \\ \hline \end{array}$$

ainsi $c_{0,0}$ est l'expression 1, $c_{0,1}$ est l'expression $c_{1,1}$, $c_{1,0}$ est l'expression $\perp$ et $c_{1,1}$ est l'expression $c_{0,0} - c_{0,1}$.

1 Types de données et constructeurs

Les formules arithmetiques que l'on souhaite entrer dans chaque cellule ont pour grammaire:

$$\begin{array}{rl} AExpr ::= & \perp \mid n \mid c_{i,j} \mid (AExpr) \mid AExpr + AExpr \mid \\ & AExpr - AExpr \mid -AExpr \mid AExpr * AExpr \mid AExpr/AExpr \end{array}$$

ou n est une constante ottante ($n \in Float$, de type `float` donc), $c_{i,j}$ denote la valeur de la case numero (i, j), $i = 0, \ldots, n-1$, $j = 0, \ldots, m-1$. Ainsi, les expressions:

$$\begin{array}{l} 3.1415 \\ c_{1,2} \\ (c_{1,2} + c_{2,3}) * c_{0,0} \end{array}$$

sont des expressions arithmetiques possibles, de même que les expressions apparaissant dans l'exemple $\mathcal{C}$ de l'introduction.

- (1.a) De nir une classe Java *concrète* `AExpr` qui implemente le type correspondant aux expressions arithmetiques (on ne cherchera pas forcement a être e cace en memoire). On utilisera en particulier un champ `operator kind` ou `operator` est le type enumere (somme) suivant:

```
public enum operator {plus, times, div}
```

qui de nit le type des operateurs binaires.

- (1.b) De nir les constructeurs:

```
AExpr()
AExpr(float n)
AExpr(int i, int j)
AExpr(operator op, lexpr, rexpr)
AExpr(expr)
```

qui rendent respectivement $\perp$, un AExpr representant le ottant n, $c_{i,j}$, l'operation binaire op applique a une sous-expression gauche `lexpr` et a une sous-expression droite `rexpr`, et le

- (2.b) Ecrire la semantique denotationnelle $[\![expr]\!]$ des expressions arithmetiques $a \in AExpr$, par recurrence structurelle sur $AExpr$. On de nira uniquement cette semantique, dans le cas idealise ou les calculs sont e ectues dans les nombres reels (et non en precision nie). Cette fonction prend un environnement ρ et renvoie $[\![a]\!]\rho$, egal a la valeur (reelle) de l'expression a si a est de nie, $\perp$ sinon. On renverra $\perp$ egalement en cas d'erreur dans une operation arithmetique (une division par zero). Expliquer brievement pourquoi cette recurrence structurelle est bien de nie.

  On cherche maintenant a implementer la fonction semantique de la question (2.b), que l'on nommera `Eval` telle que $[\![a]\!]\rho$=`a.Eval(`ρ`)`. Cette fonction implementera directement la de nition par recurrence structurelle de la question (2.b), en en faisant une fonction recursive.

- (2.c) Ecrire le code de la methode dynamique et recursive `Eval` de la classe `AExpr` prenant en argument une feuille C, et permettant d'evaluer l'expression sur laquelle elle s'applique. On supposera disposer d'une methode `Defined`, que l'on ne cherchera pas a ecrire, qui s'applique a une expression a, prend en argument une feuille C, et renvoie `true` si a est de nie dans C, `false` sinon.

- (2.d - optionnel) Comment faire pour se passer de la methode `Defined` pour programmer `Eval` ? Ecrire le code correspondant.

  Indication: On pourra s'aider d'un tableau de booleens `bool dejavu[n][m]` passe en argument a `Eval`, qui est initialise a faux au moment de l'appel initial a `Eval`. L'entree `dejavu[i][j]` est mis a vrai quand on a besoin d'evaluer (recursivement) $c_{i,j}$.

3 Gestion de la feuille de calcul

- (3.a) Ecrire une methode dynamique `Compute`, de la classe `Feuille`, qui evalue chaque cellule de la feuille courante (la fonction `Eval` du 2.c etant supposee donnee).

- (3.b) Que doit donner `Compute` sur $\mathcal{C}$? et sur $\mathcal{D}$?

- Le defaut de la methode precedente est qu'une cellule peut être a $\perp$ parce qu'elle n'est pas de nie, mais aussi parcequ'une division par zero a pu se produire. On souhaiterait maintenant pouvoir remonter l'information supplementaire qu'une erreur de division par zero s'est produite.

  – (3.c) Par quel mecanisme peut-on traiter simplement ces cas de division par zero ?
  – (3.d) Dire comment modi er le code d'`Eval` (sans tout reecrire) de la question (2.c) et le code de `Compute` de la question (3.a) a n d'implementer ce mecanisme.

4 Méthode alternative d'évaluation

De fait, une feuille de calcul pourrait egalement être vue comme un systeme d'equations de point xe, donne par l'ensemble des expressions arithmetiques $c_{i,j}$. Ainsi, l'exemple de feuille simple $\mathcal{C}$ donne en debut de probleme peut être aussi decrit par le systeme d'equations:

$$\begin{pmatrix} c_{0,0} \\ c_{0,1} \\ c_{1,0} \\ c_{1,1} \end{pmatrix} = F^{\mathcal{C}} \begin{pmatrix} c_{0,0} \\ c_{0,1} \\ c_{1,0} \\ c_{1,1} \end{pmatrix}$$

ou

$$F^{\mathcal{C}} \begin{pmatrix} c_{0,0} \\ c_{0,1} \\ c_{1,0} \\ c_{1,1} \end{pmatrix} = \begin{pmatrix} 1 \\ c_{1,1} \\ \perp \\ c_{0,0} - c_{0,1} \end{pmatrix}$$

et $F^{\mathcal{C}}$ est croissante et continue (on ne cherchera pas a le demontrer) dans le CPO $Float_{\perp}$ ou $\perp$ en est le plus petit element, et aucun couple $x, y \in Float$ avec $x \neq y$ n'est comparable. On notera $F^{\mathcal{C}}_{i,j}$ la composante de la fonction $F^{\mathcal{C}}$ correspondant a l'entree (i, j), i = 0, . . . , $n-1$, j = 0, . . . , $m-1$.

- (4.a) Quels sont les points xes du $F^{\mathcal{C}}$ particulier de ni ci-dessus ? De même pour $F^{\mathcal{D}}$ la fonctionnelle induite par la feuille $\mathcal{D}$ de la question 2.a.
- (4.b) Quel est le plus petit point xe de $F^{\mathcal{C}}$? de $F^{\mathcal{D}}$?
- (4.c) Donner les iterees successives du calcul de plus petit point xe de $F^{\mathcal{C}}$ et de $F^{\mathcal{D}}$ par le theoreme de Kleene.
- Prouver que, dans le cas general (donc pour tout F venant d'une feuille de calcul C quelconque):
  - (4.d) Si $c_{i,j}$ n'est pas de ni dans la feuille ρ, alors pour tout $k \in \mathbb{N}$, $(F^k)_{i,j}(\perp) = \perp$, ou $(F^k)_{i,j}$ est la composante (i, j) de l'iteree $k$ieme de F
  - (4.e) Si tous les $c_{i,j}$ sont de nis dans la feuille ρ, on peut de nir la profondeur de dependance $k^a_{i,j} \in \mathbb{N}$ de toute expression a par rapport a la cellule $c_{i,j}$ dans l'environnement ρ_C comme etant la longueur maximale d'un chemin dans G_{ρ_C} partant des cellules contenues dans a et arrivant en $c_{i,j}$.
    Appelons C^1 l'ensemble des cellules $c_{i,j}$ telles que $\rho(c_{i,j})$ est une constante numerique. Prouver maintenant que si tous les $c_{i,j}$ sont de nis dans la feuille ρ, $F^k(\perp)$, $k \geq 1$, donne la valeur numerique (donc di erente de $\perp$, au sens de la semantique denotationnelle de la question (2.a)) de l'ensemble C^k des cellules $c_{i,j}$ telles que $k > k^{c_{i,j}}_{i',j'}$ pour toutes cellules $c_{i',j'} \in C^1$.
  - (4.f) En deduire que le plus petit point xe de la fonctionnelle F est egal a la semantique denotationnelle donnee a la question (2.a).
- (4.g) Ecrire une version iterative de la methode $Eval$ de la question (2.c), implementant l'iteration de Kleene, que l'on sait donner le même resultat que la methode recursive de la question (2.c), grâce a la question (4.f) (on n'a pas besoin d'avoir repondu a (4.f) pour faire cette question).
- (4.h - optionnel) Soit P le predicat suivant sur les $a \in AExpr$ et les feuilles ρ: $P(a, \rho)$ est vrai si a est de nie dans ρ. Donner les assertions necessaires a la preuve a la Hoare du code iteratif d'`Eval` de la question (4.d), a chaque ligne du code, permettant de prouver

  $$\{\neg P(a, \rho)\}\texttt{x=a.Eval(}\rho\texttt{)}\{x = \perp\}$$

  Veri er les assertions de preuve en utilisant les regles de preuve a la Hoare du cours. On pourra bien sûr s'inspirer de la preuve a la question (4.d).