Cours d'interprétation abstraite : application à la vérification et à l'analyse statique

partiel

7 décembre 2012
8h45-11h45

Résumé

Cet examen consistera en une épreuve écrite de 3 heures. Les seuls documents autorisés sont les transparents du cours et vos notes de cours personnelles. L'utilisation d'ordinateur ou téléphonne sera bien sûr interdite.

Cette épreuve est composée de deux parties indépendantes. Il est demandé de traiter ces deux parties sur des copies séparées.

Première partie

Abstraction des grammaires

Étant donné un ensemble $\mathcal{X}$, on notera $\mathcal{X}^\star$ l'ensemble des suites finies d'éléments de $\mathcal{X}$ (ou mots). On suppose qu'un alphabet fini $\mathbb{A} = \{a, b, c, \ldots\}$ est fixé.

Une *grammaire* est définie par un triplet $\mathcal{G} = (\mathcal{N}, \mathcal{P}, S)$, où :

- $\mathcal{N} = \{X, Y, Z, \ldots\}$ est l'ensemble des *symboles non terminaux* ($\mathbb{A} \cap \mathcal{N} = \emptyset$) ;
- $\mathcal{P}$ décrit l'ensemble des *productions*, où une production est définie par une paire (X, u) où $X \in \mathcal{N}$ et $u \in (\mathbb{A} \cup \mathcal{N})^\star$ (une telle production est généralement notée $X \to u$) ;
- $S \in \mathcal{N}$ est le *symbole initial*.

À titre d'exemple, nous définissons les deux grammaires $\mathcal{G}_0 = (\mathcal{N}_0, \mathcal{P}_0, S_0)$ et $\mathcal{G}_1 = (\mathcal{N}_1, \mathcal{P}_1, S_1)$ où $\mathcal{N}_0 = \{S_0, T_0, U_0\}$ et $\mathcal{N}_1 = \{S_1, T_1\}$ et dont les ensembles de productions sont définis comme suit :

Définition de $\mathcal{P}_0$:

$$\begin{array}{lll} S_0 & \to_0 & aT_0b \\ S_0 & \to_0 & bT_0a \\ T_0 & \to_0 & c \\ T_0 & \to_0 & aU_0b \\ U_0 & \to_0 & c \end{array}$$

Définition de $\mathcal{P}_1$:

$$\begin{array}{lll} S_1 & \to_1 & S_1cT_1 \\ S_1 & \to_1 & T_1 \\ T_1 & \to_1 & d \\ T_1 & \to_1 & e \\ T_1 & \to_1 & aS_1b \end{array}$$

Intuitivement, une grammaire définit un ensemble de mots sur $\mathbb{A}$, en décrivant comment un mot de l'ensemble peut être engendré en partant du symbole initial, et en appliquant des productions une à une ; à chaque application d'une production $X \to u$, une occurrence de X est remplacée par u (on appelle le processus de générer un tel mot une *dérivation*, et on note $S \to^\star u$). Par exemple, pour $\mathcal{G}_0$, on a la dérivation :

$$S_0 \to_0 aT_0b \to_0 aaU_0bb \to_0 aacbb$$

1 Sémantique standard

On définit la *sémantique standard* $[\![\mathcal{G}]\!]$ de la grammaire $\mathcal{G} = (\mathcal{N}, \mathcal{P}, S)$ comme étant la fonction :

$$\begin{array}{rccl} [\![\mathcal{G}]\!] : & \mathcal{N} & \longrightarrow & \mathcal{P}(\mathbb{A}^\star) \\ & X & \longmapsto & \{u \in \mathbb{A}^\star \mid X \to^\star u\} \end{array}$$

Question 1

- *donner la sémantique standard de $\mathcal{G}_0$;*
- *montrer que $[\![\mathcal{G}_1]\!](S_1)$ contient des mots de longueur arbitrairement longue.*

On définit la notation suivante : si on a une fonction $\Phi : \mathcal{N} \longrightarrow \mathcal{P}(\mathbb{A}^\star)$, alors, on écrit :

$$\overline{\Phi}(a) = a \qquad \overline{\Phi}(X) = \Phi(X) \qquad \overline{\Phi}(u_0 u_1) = \overline{\Phi}(u_0)\overline{\Phi}(u_1)$$

Pour toute grammaire $\mathcal{G} = (\mathcal{N}, \mathcal{P}, S)$, on définit l'opérateur ci-dessous :

$$\begin{array}{rccl} \mathcal{F}_\mathcal{G} : & (\mathcal{N} \to \mathcal{P}(\mathbb{A}^\star)) & \longrightarrow & (\mathcal{N} \to \mathcal{P}(\mathbb{A}^\star)) \\ & \Phi & \longmapsto & \lambda(X \in \mathcal{N}) \cdot \Phi(X) \cup \left(\bigcup_{(X,u)\in\mathcal{P}} \overline{\Phi}(u) \right) \end{array}$$

Question 2

- *prouver que $\mathcal{F}_\mathcal{G}$ est continu sur le treillis $(\mathcal{N} \to \mathcal{P}(\mathbb{A}^\star), \dot{\subseteq})$;*
- *en déduire l'existence d'un plus petit point-fixe $\mathbf{lfp}\mathcal{F}_\mathcal{G}$ de $\mathcal{F}_\mathcal{G}$;*
- *prouver que*

$$[\![\mathcal{G}]\!] = \mathbf{lfp}\mathcal{F}_\mathcal{G} = \bigcup_{n\in\mathbb{N}} \mathcal{F}_\mathcal{G}^n(\lambda(X \in \mathcal{N}) \cdot \emptyset)$$

  (on pourra procéder par double inclusion)
- *donner les trois premiers itérés ($\mathcal{F}_\mathcal{G}^i(\lambda(X \in \mathcal{N}) \cdot \emptyset)$ où $i = 1, 2, 3$) dans le cas de $\mathcal{G}_0$, et dans le cas de $\mathcal{G}_1$; que constate-t'on ?*

2 Vecteur de Parikh et abstraction

Soit un mot $u = a_0 \dots a_{n-1} \in \mathbb{A}^\star$. On appelle *vecteur de Parikh* de u la fonction $\mathfrak{P}_u$ qui associe à chaque lettre $a \in \mathbb{A}$ son nombre d'occurence dans u (i.e., $\mathbf{card}\{i \mid 0 \leq i < n \wedge a_i = a\}$ où $\mathbf{card}E$ est le nombre d'éléments de l'ensemble E).

Question 3

On souhaite définir une abstraction pour les ensembles de mots, fondée sur le vecteur de Parikh :

- *définir un treillis dont les éléments sont les ensembles de vecteurs de Parikh (définir l'ensemble des éléments et la relation d'ordre) ;*
- *définir les fonctions d'abstraction et de concrétisation $\alpha_\mathfrak{P}$ et $\gamma_\mathfrak{P}$ qui correspondent à cette notion d'abstraction ;*
- *montrer que ces fonctions définissent une correspondance de Galois ;*
- *écrire la sémantique abstraite de Parikh $[\![\mathcal{G}]\!]_\mathfrak{P}$ d'une grammaire $\mathcal{G}$ comme étant l'application de l'abstraction de Parikh à la sémantique standard de $\mathcal{G}$ (plus précisément, $[\![\mathcal{G}]\!]_\mathfrak{P}$ associera à chaque symbole non terminal l'ensemble des vecteurs de Parikh des mots qui peuvent être dérivés de ce symbole) ;*
- *donner la sémantique abstraite de Parikh de la grammaire $\mathcal{G}_0$ donnée en exemple (inutile d'expliquer le calcul) ;*
- *déduire de ce qui précède une définition de la sémantique abstraite de Parikh sous la forme d'un plus petit point fixe.*

3 Abstractions numériques du vecteur de Parikh

L'avantage de la construction du vecteur de Parikh est qu'elle permet ensuite d'appliquer de nombreuses applications numériques et de calculer des propriétés sur le langage défini par la grammaire, par interprétation abstraite de la sémantique standard. Nous nous proposons maintenant de définir quelques abstractions d'ensembles de vecteurs de Parikh (essentiellement des abstractions du treillis $(\mathcal{P}(\mathbb{A} \to \mathbb{N}), \subseteq)$).

3.1 Analyse d'occurrence

Un premier exemple d'analyse vise à déterminer quelles sont les lettres qui peuvent apparaître dans un mot, et celles qui sont sûres d'apparaître. Ainsi, aucun mot généré à partir de U_0 ne contient de a alors que tous les mots générés à partir de S_0 en contiennent. On souhaite formaliser cette abstraction :

Question 4
- *proposer un treillis abstrait permettant de représenter cette propriété ;*
- *déduire de ce qui précède une analyse qui associe à chaque symbole non terminal, et à chaque lettre une propriété permettant de savoir si celle-ci peut / doit apparaître dans chaque mot dérivé de ce symbole ;*
- *donner les résultats pour $\mathcal{G}_0$ et $\mathcal{G}_1$ (inutile d'expliquer le calcul).*

3.2 Analyse d'égalités

Une second analyse permet d'établir des relations d'égalités entre nombres d'occurrences de lettres distinctes (par exemple a et b apparaissent le même nombre de fois dans tout mot généré par la grammaire à un seul non terminal S, avec pour seules productions $S \to aSb$ et $S \to c$). Ainsi, les nombres d'occurrences de a et de b sont égaux dans tout mot généré à partir de tout symbole de $\mathcal{G}_1$. On souhaite formaliser cette abstraction :

Question 5
- *proposer un treillis abstrait permettant de représenter cette propriété ;*
- *déduire de ce qui précède une analyse qui associe à chaque symbole non terminal une valeur abstraite décrivant un ensemble d'égalités entre nombres d'occurrences ;*
- *donner les résultats pour $\mathcal{G}_0$ et $\mathcal{G}_1$ (inutile d'expliquer le calcul).*

3.3 "a Followed by b"

On se fixe deux lettres $a, b \in \mathbb{A}$. On s'intéresse à présent à la propriété suivante sur les langages : est il possible de décomposer *tout* mot du langage, sous la forme $u_0 a u_1 b u_2$? Autrement dit, cette propriété est vraie si et seulement si tout mot du langage contient au moins un a et un b et qu'une occurrence de a précède une occurence de b.

Question 6

- *cette propriété peut elle être déterminée de manière raisonnablement précise en utilisant une abstraction du vecteur de Parikh ? (on justifiera à l'aide d'un argument simple ou d'un contre-exemple)*
- *proposer une analyse permettant de déterminer si cette propriété est vraie.*

Deuxième partie

Abstraction de calculs sur les nombres complexes

Résumé

Dans ce problème, nous esquissons la conception d'un domaine numérique pour borner le module des nombres complexes qui apparaissent dans les programmes.

La correction des fonctions de transfert et autres primitives proposées sera prouvée. Par ailleurs, même si aucun critère de précision n'est requis, la note attribuée tiendra compte de la précision des fonctions de transfert et des primitives proposées. De plus, seules des définitions mathématiques sont demandées. Aussi, aucune information sur les structures de données n'est attendue. De manière générale, nous ne nous occupons pas du temps de calcul, ni de la représentabilité. De même, les questions de terminaison de l'analyse d'un programme (opérateurs d'élargissement) ne sont pas abordées dans ce problème.

4 Pré-requis sur les nombres complexes

Un nombre complexe z est un couple (a, b) de nombres réels. L'ensemble des nombres complexes est noté $\mathbb{C}$. Étant donnés des nombres complexes $z \triangleq (a, b)$, $z_1 \triangleq (a_1, b_1)$, et $z_2 \triangleq (a_2, b_2)$ et $\lambda \in \mathbb{R}$ un nombre réel, nous noterons $\overline{z}$, $\lambda.z$, $z_1 + z_2$, et $z_1 \cdot z_2$, les nombres complexes suivants :

- $\overline{z} \triangleq (a, -b)$;
- $\lambda.z \triangleq (\lambda \cdot a, \lambda \cdot b)$;
- $z_1 + z_2 \triangleq (a_1 + a_2, b_1 + b_2)$;
- $z_1 \cdot z_2 \triangleq (a_1 \cdot a_2 - b_1 \cdot b_2, a_1 \cdot b_2 + a_2 \cdot b_1)$.

Le complexe $\overline{z}$ est appelé le conjugué du nombre complexe z ; le complexe $\lambda.z$ est appelé le produit externe du nombre complexe z par le nombre réel λ ; les complexes $z_1 + z_2$ et $z_1 \cdot z_2$ sont respectivement la somme et le produit entre les nombres complexes z_1 et z_2.

Enfin, étant donné un nombre complexe $z \triangleq (a, b)$, nous définissons le module $|z|$ de z de la manière suivante :

$$|z| = \sqrt{a^2 + b^2}.$$

Nous rappelons les propriétés suivantes, pour tous complexes $z, z_1, z_2 \in \mathbb{X}$ et tout réel $\lambda \in \mathbb{R}$:

- $|\overline{z}| = |z|$;
- $|\lambda.z| = |\lambda| \cdot |z|$;
- $||z_1| - |z_2|| \leq |z_1 + z_2| \leq |z_1| + |z_2|$;

– $|z_1.z_2| = |z_1| \cdot |z_2|$.

5 Domaine abstrait

Nous considérons un ensemble de variables V. Nous voulons abstraire un ensemble de fonctions (ou environnements) de l'ensemble des variables V dans l'ensemble des nombres réels $\mathbb{R}$ par un ensemble de contraintes portant sur le module des nombres complexes formés à partir des valeurs associées aux variables de l'ensemble V.

Pour commencer, nous introduisons un ensemble d'expressions pour désigner un nombre complexe :

$$\begin{array}{rcl} v_1, v_2 & \in & V \\ \lambda & \in & \mathbb{R} \\ E_1, E_2, E & \stackrel{\Delta}{=} & C(v_1, v_2) \mid \overline{E} \mid (\lambda.E) \mid (E_1 + E_2) \mid (E_1 \cdot E_2). \end{array}$$

L'expression $C(v_1, v_2)$ représente le nombre complexe dont la première composante est la valeur de la variable v_1 et la second la valeur de la variable v_2 ; l'expression $\overline{E}$ représente le conjugué du nombre complexe représenté par l'expression E ; l'expression $(\lambda.E)$ représente le produit externe entre le nombre réel $\lambda \in \mathbb{R}$ et le nombre complexe représenté par l'expression E ; l'expression $(E_1 + E_2)$ représente la somme des nombres complexes représentés par l'expression E_1 et l'expression E_2 ; enfin l'expression $(E_1 \cdot E_2)$ représente le produit entre les deux nombres complexes représentés par l'expression E_1 et l'expression E_2.

Nous définissons l'ensemble I des intervalles de réels positifs (c'est à dire l'ensemble des parties convexes de $\mathbb{R}^+$). Nous donnons maintenant la syntaxe de nos contraintes :

$$\begin{array}{rcl} i & \in & I \\ c & \stackrel{\Delta}{=} & P(v_1, v_2, i) \mid \mathit{fst}(V, E) \mid \mathit{snd}(V, E) \end{array}$$

Nous distinguons en fait deux types de contraintes :

– Le premier type de contraintes encadre le module des nombres complexes. Si v_1 et v_2 sont deux variables et I un intervalle, la contrainte $P(v_1, v_2, i)$ spécifie que le module du nombre complexe dont la première composante est la valeur de la variable v_1 et la seconde la valeur de la variable v_2, doit être dans l'intervalle i.

– Le second type relie les variables aux expressions : la contrainte $\mathit{fst}(V, E)$ spécifie que la valeur de la variable V est la première composante du nombre complexe représenté par l'expression E ; la contrainte $\mathit{snd}(V, E)$ spécifie que la valeur de la variable V est la seconde composante du nombre complexe représenté par l'expression E.

Notre domaine concret D est l'ensemble des ensembles de fonctions entre V et $\mathbb{R}$, alors que notre domaine abstrait $D^\sharp$ est l'ensemble des ensembles de contraintes.

Question 1
Définir une fonction de concrétisation γ qui associe à un ensemble de contraintes $e^\sharp$, l'ensemble des environnements de V dans $\mathbb{R}$ qui satisfont toutes ces contraintes.

Nous ordonnons le domaine concret D par l'inclusion (des ensembles de fonctions entre V et $\mathbb{R}$) et le domaine abstrait $D^\sharp$ par l'inclusion inverse (sur les ensembles de contraintes).

Question 2
Montrer que la fonction γ est monotone.

6 Fonction de transfert

Nous nous intéressons maintenant aux fonctions de transfert qui permettent de sur-approximer dans l'abstrait le calcul des affectations.

La première étape est de proposer une primitive qui permette d'oublier une variable.

Question 3
Définir une fonction d'oubli qui associe à un ensemble de contraintes $e^\sharp$ un autre ensemble de contraintes de sorte que :

$$\{\rho[v \mapsto a] \mid \rho \in \gamma(e^\sharp), a \in \mathbb{R}\} \subseteq \gamma(\mathit{forget}_v(e^\sharp)).$$

Nous pouvons maintenant définir les fonctions de transfert pour les affectations. Nous considérons pour simplifier la conception du domaine abstrait que toute composante d'un nombre complexe est formée en une seule étape de calcul (ou une seule affectation) et que la sémantique manipule des nombres réels.

Question 4
Définir des fonctions de transferts :

$$\begin{cases} \mathit{scal} \ : \ V \times \mathbb{R} \times V \times D^\sharp \to D^\sharp, \\ \mathit{add} \ : \ V \times V \times V \times D^\sharp \to D^\sharp, \\ \mathit{poly}_2 \ : \ V \times V \times V \times V \times V \times D^\sharp \to D^\sharp, \end{cases}$$

qui simulent dans l'abstrait certains types d'affectations.
Ces fonctions de transferts devront vérifier les contraintes de corrections suivantes :[1]

- $\{\rho[v \mapsto a \times \rho(v')] \mid \rho \in \gamma(e^\sharp), a \in \mathbb{R}\} \subseteq \gamma(\mathit{scal}(v, a, v', e^\sharp))$;
- $\{\rho[v \mapsto \rho(v_1) + \rho(v_2)] \mid \rho \in \gamma(e^\sharp), v_1, v_2 \in V\} \subseteq \gamma(\mathit{add}(v, v_1, v_2, e^\sharp))$;
- $\{\rho[v \mapsto \rho(v_1) \times \rho(v_2) + \rho(v_3) \times \rho(v_4)] \mid \rho \in \gamma(e^\sharp), v_1, v_2, v_3, v_4 \in V\} \subseteq \gamma(\mathit{poly}_2(v, v_1, v_2, v_3, v_4, e^\sharp))$.

1. Si f est une fonction de V dans $\mathbb{R}$, $\lambda \in \mathbb{R}$ est un nombre réel, et $v \in V$ est une variable, alors $f[v \mapsto \lambda]$ est la fonction de V dans $\mathbb{R}$ qui associe à toute variable $v' \in V$ le nombre réel $f(v')$ lorsque $v' \neq v$ et le nombre réel λ sinon.

7 Réduction

Nous nous intéressons maintenant aux opérateurs de réductions. Nous distinguerons trois sortes de réduction.

Connaissant l'intervalle de variation de deux variables, il est possible de déduire un intervalle de variation pour le module du nombre complexe formé par la valeur de ces deux variables.

Question 5
Proposer une primitive **import-interv** : $V \times I \times V \times I \times D^\sharp \to D^\sharp$ *qui prend en compte les intervalles de variations, pour introduire de nouvelles contraintes, et qui vériefiera la propriété suivante :*

$$\{\rho \mid \rho \in \gamma(e^\sharp), |\rho(v_1)| \in i_1, |\rho(v_2)| \in i_2\} \subseteq \gamma(\textbf{import-interv}(v_1, i_1, v_2, i_2, e^\sharp)).$$

Réciproquement, connaissant l'intervalle de variation du module d'un nombre complexe, il est possible de déduire un intervalle de variation pour chacune de ses composantes.

Question 6
Proposer une primitive **export-interv** : $V \times D^\sharp \to I$, *et qui vériefiera la propriété suivante :*

$$\rho \in \gamma(e^\sharp) \Rightarrow |\rho(v)| \in \textbf{export-interv}(v, e^\sharp).$$

Lorsque les deux composantes d'un nombre complexe sont connues, il est possible de former une nouvelle contrainte de type $P(v_1, v_2, I)$.

Question 7
Proposer un opérateur de clôture inférieure **combine** *sur* $(D^\sharp, \supseteq)$ *qui ajoute autant de contraintes de type* $P(v_1, v_2, I)$ *que possible, et qui vériefiera la propriété suivante :*

$$\gamma(e^\sharp) \subseteq \gamma(\textbf{combine}(e^\sharp)).$$

8 Pour aller plus loin

Nous supposons désormais que l'évaluation des expressions se fait en arithmétique flottante. Pour simplifier, nous supposerons connue une borne sur les erreurs d'arrondis pour l'évaluation de chaque expression.

Question 8
Proposer un nouveau domaine abstrait $D'^\sharp$, muni d'une fonction de concrétisation $\gamma' : D'^\sharp \to \wp(V \to \mathbb{R})$ pour définir des fonctions de transferts précises :

$$\begin{cases} \texttt{scal}' : V \times \mathbb{R} \times V \times \mathbb{R}^+ \times D'^\sharp \to D'^\sharp, \\ \texttt{add}' : V \times V \times V \times \mathbb{R}^+ \times D'^\sharp \to D'^\sharp, \\ \texttt{poly}_2' : V \times V \times V \times V \times V \times \mathbb{R}^+ \times D'^\sharp \to D'^\sharp, \end{cases}$$

qui simulent dans l'abstrait certains types d'affectations.
Ces fonctions de transferts devront vérifier les contraintes de corrections suivantes :
- $\{\rho[v \mapsto \epsilon + a \times \rho(v')] \mid \rho \in \gamma'(e^\sharp), a \in \mathbb{R}, |\epsilon| \leq \varepsilon\} \subseteq \gamma'(\texttt{scal}'(v, a, v', \varepsilon, e^\sharp))$;
- $\{\rho[v \mapsto \epsilon + \rho(v_1) + \rho(v_2)] \mid \rho \in \gamma'(e^\sharp), v_1, v_2 \in V, |\epsilon| \leq \varepsilon\} \subseteq \gamma'(\texttt{add}'(v, v_1, v_2, \varepsilon, e^\sharp))$;
- $\{\rho[v \mapsto \epsilon + \rho(v_1) \times \rho(v_2) + \rho(v_3) \times \rho(v_4)] \mid \rho \in \gamma'(e^\sharp), v_1, v_2, v_3, v_4 \in V, |\epsilon| \leq \varepsilon\} \subseteq \gamma'(\texttt{poly}_2'(v, v_1, v_2, v_3, v_4, \varepsilon, e^\sharp))$.

Question 9
Mettre à jour la définition des opérateurs de réduction en conséquence.