MPRI - Examen Langage de Programmation Probabiliste
Mars 2022

Consignes Vous avez 3h pour compléter l'examen.
Il y a 3 exercices.
Merci de respecter les consignes suivantes :

- Tous les appareils électroniques doivent être éteints et rangés dans votre sac.
- Documentation autorisée: polycopié de cours.
- Le barème est donné à titre indicatif et peut être modifié.

Exercice 1 : Modélisation - 6 points

On modélise l'arrivée d'étudiants au restaurant universitaire à l'aide de processus de Poisson de paramètre L.
On note $N(s+t)-N(s)$ le nombre d'arrivées entre les temps s et $s+t$. On suppose que $N(s+t)-N(s)$ est une variable aléatoire discrète dont la fonction de masse sachant que le paramètre L est λ est :

$$\mathbb{P}(N(s+t)-N(s)=k|L=\lambda)=\frac{(\lambda t)^k}{k!}e^{-\lambda t}.$$

On note $U(n)$ le laps de temps écoulé entre les arrivées du n-ème étudiant et du $n+1$-ème étudiant. On suppose que $U(n)$ est une variable aléatoire continue à densité dont la fonction de densité sachant que le paramètre est λ est

$$f_{U(n)|L=\lambda}(x)=\lambda e^{-\lambda x}\mathbb{1}_{x\geqslant 0}.$$

Le paramètre L du Processus de Poisson dépend du service. À midi, le paramètre L vaut 20. Le soir, L vaut 10.
À chaque service, le nombre d'étudiants est noté afin de prévoir les quantités. Ces données sont notées à la main, scannées et analysées par un algorithme de reconnaissance de caractères.
Sur une des feuilles, l'horaire indiqué pour le début du service est peu lisible. L'algorithme de reconnaissance de caractères indique 12h avec probabilité 0.6 et 19h avec probabilité 0.4. On peut en déduire qu'a priori, $\mathbb{P}(L=20)=0.6$ et $\mathbb{P}(L=10)=0.4$. On cherche à affiner notre connaissance sur cette probabilité à partir des observations notées, en utilisant l'inférence bayésienne.

1. Quelle est la probabilité que l'heure notée sur la feuille soit 12h sachant que l'on a observé $k=100$ personnes en un temps $t=10$ minutes. Vous utiliserez la formule de Bayes.
2. Choisir un langage de programmation probabiliste et implémenter un programme permettant d'approximer cette probabilité.
3. Décrire un algorithme d'inférence qui peut être utilisé pour approcher cette probabilité.
4. À partir de cette même feuille, on cherche maintenant à approcher la probabilité que l'heure notée soit 12h sachant que deux arrivées consécutives ont été enregistrées à 6 secondes d'intervalle. La vraissemblance est supposée proportionnelle à la fonction de densité du temps d'attente entre deux arrivées sachant le paramètre du processus de Poisson.

   Implémenter un programme permettant de calculer cette probabilité. Est-ce que l'on va trouver la même valeur que dans le modèle précédent, pourquoi ?

Exercice 2 : Simuler une pièce équilibrée - 6 points

On considère une pièce de monnaie biaisée qui, quand on la lance, renvoie face avec probabilité $p \in]0,1[$ et pile avec probabilité $1-p$.
On considère l'algorithme suivant:

- Lancer successivement la pièce deux fois, les lancers étant indépendants.
- Ensuite,
  - retourner 1 si la pièce a renvoyé pile puis face,
  - retourner 0 si la pièce a renvoyé face puis pile,
  - dans les autres cas, recommencer.

1. Donner un programme P implémentant cet algorithme dans le langage:

$$\begin{aligned} e :=& x \mid \lambda x\, e \mid (e)\, e \mid \texttt{fix}\, e \mid \texttt{true} \mid \texttt{false} \mid \texttt{e} \neq \texttt{e} \mid \texttt{if}\, e\, \texttt{then}\, e\, \texttt{else}\, e \\ & \mid \texttt{let x} = e\, \texttt{in}\, e \quad \texttt{sample(bernoulli}\ p) \forall p \in [0,1] \end{aligned}$$

ainsi que sa dérivation de typage dans le système de type :

$t_1, t_2 := \texttt{int} \mid \texttt{bool} \mid t_1 \to t_2$

$$\textsc{Var}\ \frac{}{G, x : t \vdash x : t} \qquad \textsc{App}\ \frac{G \vdash e_1 : t_1 \qquad G \vdash e_2 : t_1 \to t_2}{G \vdash (e_2) e_1 : t_2} \qquad \textsc{Abs}\ \frac{G, x : t_1 \vdash e : t_2}{G \vdash \lambda x.e : t_1 \to t_2}$$

$$\textsc{Bern}\ \frac{}{G \vdash \texttt{sample bernoulli(p)} : \texttt{bool}} \qquad \textsc{True}\ \frac{}{G \vdash \texttt{true} : \texttt{bool}} \qquad \textsc{False}\ \frac{}{G \vdash \texttt{false} : \texttt{bool}}$$

$$\textsc{Neq}\ \frac{G \vdash e_1 : \texttt{bool} \qquad G \vdash e_2 : \texttt{bool}}{G \vdash e_1 \neq e_2 : \texttt{bool}} \qquad \textsc{If}\ \frac{G \vdash e_0 : \texttt{bool} \qquad G \vdash e_1 : t \qquad G \vdash e_2 : t}{G \vdash \texttt{if e}_0\ \texttt{then e}_1\ \texttt{else e}_2 : t}$$

$$\textsc{Let}\ \frac{G \vdash e_1 : t_1 \qquad G, x : t_1 \vdash e_2 : t_2}{G \vdash \texttt{let x} = \texttt{e}_1\ \texttt{in e}_2 : t_2} \qquad \textsc{Fix}\ \frac{G \vdash e : t \to t}{G \vdash \texttt{fix}\, e : t}$$

2. On rappelle les règles de la sémantique opérationnelle:

$$\frac{}{(\lambda x\, M)N \xrightarrow{1} M\,[N/x]} \qquad \frac{M \xrightarrow{p} M'}{(\ M)N \xrightarrow{p} (\ M')N} \qquad \frac{}{\texttt{fix}\, M \xrightarrow{1} (M)(\texttt{fix}M)}$$

$$\frac{}{\texttt{if true then}\, N\, \texttt{else}\, P \xrightarrow{1} N} \qquad \frac{}{\texttt{if false then}\, N\, \texttt{else}\, P \xrightarrow{1} P}$$

$$\frac{M \xrightarrow{p} M'}{\texttt{succ}\ M \xrightarrow{p} \texttt{succ}\ M'}$$

$$\frac{M \xrightarrow{p} M'}{\texttt{if}\ M\, \texttt{then}\, N\, \texttt{else}\, P \xrightarrow{p} \texttt{if}\ M'\, \texttt{then}\, N\, \texttt{else}\, P}$$

$$\frac{}{\texttt{sample(bernoulli}\ p) \xrightarrow{p} \texttt{true}} \qquad \frac{}{\texttt{sample(bernoulli}\ p) \xrightarrow{1-p} \texttt{false}}$$

$$\frac{M \xrightarrow{p} M'}{\texttt{let}\, x = M\, \texttt{in}\, N \xrightarrow{p} \texttt{let}\, x = M'\, \texttt{in}\, N} \qquad \frac{\text{v is a constant}}{\texttt{let}\, x = v\, \texttt{in}\, M \xrightarrow{1} M\,[v/x]}$$

Donner la règle pour l'opérateur $\neq$, puis décrire la sémantique opérationnelle de votre programme P.

3. On rappelle le théorème de correction de la sémantique probabiliste:

$$[\![M]\!] = \sum_N \mathbf{Proba}(M, N)\, [\![N]\!]$$

où $\mathbf{Proba}(M, N)$ est déterminé par la sémantique opérationnelle.
Montrer que si $P = \texttt{fix}\ \lambda x.M$ alors

$$[\![P]\!] = [\![M\,[P/x]]\!]$$

4. On rappelle quelques règles (simplifiées) de la sémantique dénotationnelle :

$$[\![\texttt{sample bernoulli}(p)]\!] = p\,[\![\texttt{true}]\!] + (1-p)\,[\![\texttt{false}]\!]$$

$$[\![\texttt{let}\ x = e_1\ \texttt{in}\ e_2]\!] = \sum_a [\![e_1]\!]_a\, [\![e_2]\!]_a$$

$$[\![\texttt{true}]\!] =: \begin{cases} \texttt{true} \mapsto 1 \\ \texttt{false} \mapsto 0 \end{cases}$$

$$[\![\texttt{if}\ M\ \texttt{then}\ N\ \texttt{else}\ P]\!] = [\![M]\!]_{\texttt{true}}\, [\![N]\!] + [\![M]\!]_{\texttt{false}}\, [\![P]\!]$$

Montrer en justifiant chacune des étapes de votre calcul que:

$$[\![P]\!] = p(1-p)\,[\![\texttt{true}]\!] + p(1-p)\,[\![\texttt{false}]\!] + (p^2 + (1-p)^2)\,[\![P]\!]$$

Puis résoudre cette équation pour calculer $[\![P]\!]$.

5. On rappelle le lemme d'adéquation au type `bool`:

$$[\![P]\!]_{\texttt{true}} = \mathbf{Proba}(P, \texttt{true}) \qquad \text{et} \qquad [\![\texttt{P}]\!]_{\texttt{false}} = \mathbf{Proba}(\texttt{P}, \texttt{false})$$

Montrer que votre programme simule une pièce équilibrée.

Exercice 3 : Correction du calcul de plus faible espérance[1] **- 8 points**

On considère le langage probabiliste suivant:

$$\begin{array}{rcl} E & ::= & \texttt{true} \mid \texttt{false} \mid n \mid x \mid E\ op\ E \mid \texttt{bernoulli p} \\ op & ::= & + \mid - \mid * \mid = \mid \leqslant \mid \mathit{and} \mid \mathit{not} \\ P & ::= & \texttt{skip} \mid x := E \mid P; P \mid \texttt{if}\ E\ \texttt{then}\ P\ \texttt{else}\ P \mid \texttt{while}\ E\ \texttt{do}\ P \end{array}$$

Les **valeurs** sont $Val = \{\texttt{true}, \texttt{false}, n\}$.
Un **état** est une fonction des variables libres d'un programme dans les valeurs: $s : Var \to Val$.
On définit la mise à jour de la valeur de x par v dans l'état s par:
$s[x \mapsto v]$ est l'état dont la variable x est associée à v et les autres variables sont associées à la même valeur que dans s.
On note S l'ensemble des états et $\mathcal{V}(S) = (\mathbb{R}_+^\infty)^S$ l'ensemble des **facteurs**, c'est-à-dire des fonctions de S dans les réels positifs ou infinis.

1. Issu de la thèse de Claire Jones, 1989

Sémantique. On note $\mathbb{P}(E(s), v)$ la probabilité que l'expression E dans l'environnement s s'évalue vers la valeur v.
Par exemple,

- Si $E = (n + m)$ et $s = [n \mapsto 2; m \mapsto 3]$, alors on a $\mathbb{P}(E(s), 5) = 1$ et $\mathbb{P}(E(s), 6) = 0$
- Si $E = (\texttt{bernoulli p or x})$ et $s = [x \mapsto \texttt{false}]$, alors on a $\mathbb{P}(E(s), \texttt{true}) = p$ et $\mathbb{P}(E(s), \texttt{false}) = 1 - p$

On note $\chi_s : S \to \mathbb{R}^\infty_+$ le facteur défini par: $\chi_s(s') = \begin{cases} 1 & \text{si } s = s' \\ 0 & \text{sinon} \end{cases}$
La sémantique $[\![P]\!] : S \to \mathcal{V}(S)$ d'un programme P est définie par induction sur la structure de P de la façon suivante: pour tous états s et s',

$$
\begin{aligned}
[\![\texttt{skip}]\!](s)(s') &= \chi_s(s') \\
[\![\texttt{x:=E}]\!](s)(s') &= \sum_{v \in Val} \chi_{s'}(s[x \mapsto v])\mathbb{P}(E(s), v) \\
[\![\texttt{P;Q}]\!](s)(s') &= \sum_{t \in S} [\![P]\!](s)(t)\ [\![Q]\!](t)(s') \\
[\![\texttt{if E then P else Q}]\!](s)(s') &= \mathbb{P}(E(s), \texttt{true})\,[\![P]\!](s)(s') + \mathbb{P}(E(s), \texttt{false})\,[\![Q]\!](s)(s') \\
[\![\texttt{while E do P}]\!](s)(s') &= \bigcup_n f_n(s)(s')
\end{aligned}
$$

avec $f_0(s)(s') = 0$ et $f_{n+1}(s)(s') = \mathbb{P}(E(s), \texttt{true}) \sum_{t \in S} [\![P]\!](s)(t)\, f_n(t, s') + \mathbb{P}(E(s), \texttt{false})\chi_s(s')$

Logique de Hoare probabiliste. Étant donnés des facteurs $f, g : S \to \mathbb{R}^\infty_+$ et programme P, on définit le système de preuve des triplets de Hoare probabilistes $f[P]g$ par les règles suivantes:

SKIP RULE

$$\frac{}{f\ \texttt{[skip]}\ f}$$

ASSIGN RULE

$$\frac{}{h\ \texttt{[x:=E]}\ g} \text{ where } h(s) = \sum_{v \in Val} g(s[x \mapsto v]\mathbb{P}(E(s), v)$$

SEQUENCE RULE

$$\frac{f\ \texttt{[P]}\ g \qquad h\ \texttt{[Q]}\ h}{f\ \texttt{[P;Q]}\ h}$$

IF RULE

$$\frac{f\ \texttt{[P]}\ g \qquad f'\ \texttt{[Q]}\ g}{\mathbb{P}(E, \texttt{true})f + \mathbb{P}(E, \texttt{false})f'\ \texttt{[if E then P else Q]}\ h}$$

WHILE RULE

$$\frac{f_{n+1}\ \texttt{[P]}\ \mathbb{P}(E, \texttt{true})f_n + \mathbb{P}(E, \texttt{false})k +}{\mathbb{P}(E, \texttt{true})\bigvee_n f_n + \mathbb{P}(E, \texttt{false})k\ \texttt{[while E do P]}\ k} \text{ where } \begin{cases} f_0 = \lambda s.0 \\ \bigvee_n f_n = \lambda s. \sup_n f_n(s) \end{cases}$$

CONSEQUENCE RULE

$$\frac{f\ \texttt{[P]}\ g}{f'\ \texttt{[P]}\ g'} \qquad \text{where } \forall s,\ f'(s) \leqslant f(s) \text{ and } g(s) \leqslant g'(s)$$

Calcul de plus faible préfacteur Étant donnés un programme `P` et un facteur g, on définit le facteur $\texttt{wp}(P, g)$ par induction sur la structure de `P`:

$$\texttt{wp}(\texttt{skip}, g) = g$$
$$\texttt{wp}(\texttt{x:=E}, g) = \lambda s. \sum_{v \in Val} g(s[x \mapsto v])\mathbb{P}(E(s), v)$$
$$\texttt{wp}(\texttt{P;Q}, g) = \texttt{wp}(\texttt{Q}, \texttt{wp}(\texttt{P}, g))$$
$$\texttt{wp}(\texttt{if E then P else Q}, g) = \mathbb{P}(E, \texttt{true})\texttt{wp}(P, g) + \mathbb{P}(E, \texttt{false})\texttt{wp}(P, g)$$
$$\texttt{wp}(\texttt{while E do P}, g) = \bigvee f_n \text{ where } \begin{cases} f_0 = \lambda s.0 \\ f_{n+1} = \mathbb{P}(E, \texttt{true})\texttt{wp}(\texttt{P}, f_n) + \mathbb{P}(E, \texttt{false})g \end{cases}$$

Questions

1. Démontrer la **correction**:

   Si $f \,\texttt{[P]}\, g$ est dérivable dans la logique de Hoare, alors $\forall s \in S,\ \sum_{s' \in S} g(s') \left[\!\left[P\right]\!\right](s)(s') \geqslant f(s)$.

   Vous raisonnerez par induction sur la structure de la preuve de $f[P]g$. Vous indiquerez l'hypothèse d'induction et traiterez trois cas au choix.

2. Démontrer la **complétude**:

   Si $\forall s \in S,\ \sum_{s' \in S} g(s') \left[\!\left[P\right]\!\right](s)(s') \geqslant f(s)$, alors $f \,\texttt{[P]}\, g$ est dérivable dans la logique de Hoare.

   Vous raisonnerez par induction sur la structure du programme P. Vous indiquerez l'hypothèse d'induction et traiterez trois cas au choix.

3. Montrer que
   $$\texttt{wp}(P, g) = \sup\{h \mid h[P]g\}.$$
   Vous raisonnerez par induction sur la structure du programme P. Vous traiterez trois cas au choix.

4. En déduire le **théorème de dualité**:
   $$\texttt{wp}(\texttt{P}, g) = \lambda s. \sum_{s' \in S} g(s') \left[\!\left[P\right]\!\right](s)(s')$$

5. On note $\lambda s.1$ le facteur constant qui associe 1 à tout environnement s. Montrer en utilisant le théorème de dualité que $\texttt{wp}(P, \lambda s.1)(s)$ est la probabilité que le programme P termine en partant d'un état s.

6. On considère le programme `P`, défini par
   ```
   n:=1 ; while (bernoulli p) do n:=n+1
   ```
   Étant donné un facteur g, calculer $\texttt{wp}(\texttt{P}, g)$.
   Quelle est la probabilité que ce programme termine ?

7. Donner une implémentation de l'algorithme de l'exercice 2 et calculer la probabilité que ce programme termine.