Verifikation probalistischer Programme

Im Folgenden sei `random`(a, b) ein Zufallszahlengenerator, der auf $\{a, \ldots, b\}$ gleichverteilte Zufallszahlen liefert.

Im Folgenden sei `random`(a, b) ein Zufallszahlengenerator, der auf $\{a, \ldots, b\}$ gleichverteilte Zufallszahlen liefert.

Betrachten wir hierzu nun das Programm:

```
while not x <= 2 do
   x := random(1, 6)
end
```

Bei Terminierung des Programms haben wir $P(\mathrm{x} = 1) = \frac{1}{2}$ und $P(\mathrm{x} = 2) = \frac{1}{2}$.

Aber wie wird dies formal bewiesen? Es mag ja Programme geben, die so kompliziert sind, dass nicht mehr klar ersichtlich ist, welche Verteilung ihre Ausgaben haben.

Probalistische denotationelle Semantik

Bei einem gewöhnlichen Kommando c gab es einen deterministischen Übergang vom Zustand $s \in S$ in den Zustand $s' = C[\![c]\!](s)$. Bei der Zuweisung einer zufällig gewählten Zahl trennt sich der Pfad nun aber auf, wobei den erreichbaren Zielzuständen eine positive Wahrscheinlichkeit zukommt. Ziel des Übergangs ist so gesehen eine Zufallsgröße Y, wobei ein Zustand s' mit der Wahrscheinlichkeit $P(Y = s')$ erreicht wird.

Bei einem gewöhnlichen Kommando c gab es einen deterministischen Übergang vom Zustand $s \in S$ in den Zustand $s' = C[\![c]\!](s)$. Bei der Zuweisung einer zufällig gewählten Zahl trennt sich der Pfad nun aber auf, wobei den erreichbaren Zielzuständen eine positive Wahrscheinlichkeit zukommt. Ziel des Übergangs ist so gesehen eine Zufallsgröße Y, wobei ein Zustand s' mit der Wahrscheinlichkeit $P(Y = s')$ erreicht wird.

Sagen wir, bereits der Startzustand X_0 sei eine Zufallsgröße, die zunächst die Einheitsmasse in einem Zustand $s \in S$ haben kann. Zu einem gewöhnlichen Kommando c gehört die Funktion $f : S \to S$ mit $f(s) := C[\![c]\!](s)$, das die Transformation $Y = f(X)$ für Zufallsgrößen X, Y induziert.

Bei einem gewöhnlichen Kommando c gab es einen deterministischen Übergang vom Zustand $s \in S$ in den Zustand $s' = C[\![c]\!](s)$. Bei der Zuweisung einer zufällig gewählten Zahl trennt sich der Pfad nun aber auf, wobei den erreichbaren Zielzuständen eine positive Wahrscheinlichkeit zukommt. Ziel des Übergangs ist so gesehen eine Zufallsgröße Y, wobei ein Zustand s' mit der Wahrscheinlichkeit $P(Y = s')$ erreicht wird.

Sagen wir, bereits der Startzustand X_0 sei eine Zufallsgröße, die zunächst die Einheitsmasse in einem Zustand $s \in S$ haben kann. Zu einem gewöhnlichen Kommando c gehört die Funktion $f : S \to S$ mit $f(s) := C[\![c]\!](s)$, das die Transformation $Y = f(X)$ für Zufallsgrößen X, Y induziert.

Wir definieren daher die Menge der Massefunktionen als

$$M := \{p : S \cup \{\bot\} \to \mathbb{R} \mid \sum_{s \in S \cup \{\bot\}} p(s) = 1\}.$$

Der probalistische Übergang zu einem Kommando erklärt sich nun als Funktion

$$D : \mathrm{Com} \to (M \to M),$$

die die Transformation der Massefunktion durch das Kommando beschreibt.

Zur Transformation einer Zufallsgröße X gilt die allgemeine Beziehung

$$P(f(X) = y) = \sum_{x \in f^{-1}(\{y\})} P(X = x) = \sum_{x\,:\, f(x) = y} P(X = x).$$

Zur Transformation einer Zufallsgröße X gilt die allgemeine Beziehung

$$P(f(X) = y) = \sum_{x \in f^{-1}(\{y\})} P(X = x) = \sum_{x\,:\, f(x)=y} P(X = x).$$

Demnach ergibt sich für jedes Kommando c mit deterministischer Übergangsfunktion $C[\![c]\!]$ bezüglich $p(s) = P(X = s)$ der induzierte probalistische Übergang

$$D[\![c]\!](p)(s') = P(C[\![c]\!](X) = s') = \sum_{s\,:\, C[\![c]\!](s)=s'} P(X = s) = \sum_{s\,:\, C[\![c]\!](s)=s'} p(s).$$

Zur Transformation einer Zufallsgröße X gilt die allgemeine Beziehung

$$P(f(X) = y) = \sum_{x \in f^{-1}(\{y\})} P(X = x) = \sum_{x\,:\,f(x)=y} P(X = x).$$

Demnach ergibt sich für jedes Kommando c mit deterministischer Übergangsfunktion $C[\![c]\!]$ bezüglich $p(s) = P(X = s)$ der induzierte probalistische Übergang

$$D[\![c]\!](p)(s') = P(C[\![c]\!](X) = s') = \sum_{s\,:\,C[\![c]\!](s)=s'} P(X = s) = \sum_{s\,:\,C[\![c]\!](s)=s'} p(s).$$

Die Beziehung gilt, wie gesagt, allerdings nur für die Kommandos, die `random` nicht enthalten. Wir müssen daher noch den Übergang bei allgemeinen Kommandos klären.

Zu einer Sequenz $c; c'$ gilt offenbar analog

$$D[\![c; c']\!](p) = D[\![c']\!](D[\![c]\!](p)).$$

Zu einer Sequenz $c; c'$ gilt offenbar analog

$$D[\![c; c']\!](p) = D[\![c']\!](D[\![c]\!](p)).$$

Wir können nachrechnen, dass dies nicht der zuvor gemachten Feststellung im Fall deterministischer Kommandos c, c' widerspricht.

Dazu sei $f := C[\![c]\!]$ und $g := C[\![c']\!]$. Zu zeigen gilt

$$D[\![c; c']\!](p)(s'') \stackrel{\text{def}}{=} \sum_{s \in (g \circ f)^{-1}(\{s''\})} p(s) = \sum_{s' \in g^{-1}(\{s''\})} \sum_{s \in f^{-1}(\{s'\})} p(s) \stackrel{\text{def}}{=} D[\![c']\!](D[\![c]\!](p))(s'').$$

Es findet sich

$$\sum_{s\in(g\circ f)^{-1}(\{s''\})} p(s) = P((g\circ f)(X)=s'') = P_X((g\circ f)^{-1}(\{s''\})) = P_X(f^{-1}(g^{-1}(\{s''\})))$$
$$= P_X(f^{-1}(\bigcup_{s'\in g^{-1}(\{s''\})}\{s'\})) = P_X(\bigcup_{s'\in g^{-1}(\{s''\})} f^{-1}(\{s'\})) = \sum_{s'\in g^{-1}(\{s''\})} P_X(f^{-1}(\{s'\})),$$

weil Urbilder disjunkter Zerlegungen wieder solche sind. Und weiter

$$P_X(f^{-1}(\{s'\})) = P_X(\bigcup_{s\in f^{-1}(\{s'\})}\{s\}) = \sum_{s\in f^{-1}(\{s'\})} P_X(\{s\})$$

mit $P_X(\{s\}) = P(X^{-1}(\{s\})) = P(X=s) = p(s)$.

Bezüglich einem booleschen Ausdruck b sei

$$(b?p)(s) := [B[\![b]\!](s)]p(s) = \begin{cases} p(s), & \text{wenn } B[\![b]\!](s) = \mathbf{true}, \\ 0 & \text{sonst.} \end{cases}$$

Es sollte gelten

$$D[\![\,\mathbf{if}\ b\ \mathbf{then}\ c\ \mathbf{else}\ c'\ \mathbf{end}\,]\!](p) = D[\![c]\!](b?p) + D[\![c']\!](\neg b?p).$$

Bezüglich einem booleschen Ausdruck b sei

$$(b?p)(s) := [B[\![b]\!](s)]p(s) = \begin{cases} p(s), & \text{wenn } B[\![b]\!](s) = \textbf{true}, \\ 0 & \text{sonst.} \end{cases}$$

Es sollte gelten

$$D[\![\textbf{if } b \textbf{ then } c \textbf{ else } c' \textbf{ end}]\!](p) = D[\![c]\!](b?p) + D[\![c']\!](\neg b?p).$$

Rechnen wir wieder nach, dass dies nicht dem Sachverhalt für deterministische Kommandos c, c' widerspricht. Zu zeigen ist also

$$\sum_{s \in f^{-1}(\{s'\})} p(s) = \sum_{s \in C[\![c]\!]^{-1}(\{s'\})} [B[\![b]\!](s)]p(s) + \sum_{s \in C[\![c']\!]^{-1}(\{s'\})} [B[\![\neg b]\!](s)]p(s)$$

bezüglich $f(s) := C[\![\textbf{if } b \textbf{ then } c \textbf{ else } c' \textbf{ end}]\!](s)$.

Es findet sich

$$\sum_{s\in f^{-1}(s')} p(s) = \sum_s [f(s)=s']p(s)$$

$$= \sum_s \Big[(C[\![c]\!](s)=s' \wedge B[\![b]\!](s)) \vee (C[\![c']\!](s)=s' \wedge \neg B[\![b]\!](s)) \Big] p(s)$$

$$= \sum_s \Big([C[\![c]\!](s)=s'][B[\![b]\!](s)] + [C[\![c']\!](s)=s'][\neg B[\![b]\!](s)] \Big) p(s)$$

$$= \sum_s [C[\![c]\!](s)=s'][B[\![b]\!](s)]p(s) + \sum_s [C[\![c']\!](s)=s'][\neg B[\![b]\!](s)]p(s)$$

$$= \sum_{s\in C[\![c]\!]^{-1}(\{s'\})} [B[\![b]\!](s)]p(s) + \sum_{s\in C[\![c']\!]^{-1}(\{s'\})} [\neg B[\![b]\!](s)]p(s).$$

Zur randomisierten Zuweisung überlegt man sich

$$D[\![X := \texttt{random}(a, b)]\!](p) = \frac{1}{b-a+1} \sum_{k=a}^{b} D[\![X := k]\!](p).$$

Zur randomisierten Zuweisung überlegt man sich

$$D[\![X := \texttt{random}(a,b)]\!](p) = \frac{1}{b-a+1} \sum_{k=a}^{b} D[\![X := k]\!](p).$$

Wir prüfen, ob dies plausibel ist. Ist p die Einheitsmasse im Zustand s_0, also $p(s) := [s = s_0]$, muss sich die Massefunktion der Gleichverteilung auf den möglichen Zuweisungen ergeben.

Zur randomisierten Zuweisung überlegt man sich

$$D[\![X := \mathtt{random}(a,b)]\!](p) = \frac{1}{b-a+1}\sum_{k=a}^{b} D[\![X := k]\!](p).$$

Wir prüfen, ob dies plausibel ist. Ist p die Einheitsmasse im Zustand s_0, also $p(s) := [s = s_0]$, muss sich die Massefunktion der Gleichverteilung auf den möglichen Zuweisungen ergeben. Es findet sich

$$\begin{aligned}
&\frac{1}{b-a+1}\sum_{k=a}^{b} D[\![X := k]\!](p)(s') = \frac{1}{b-a+1}\sum_{k=a}^{b}\sum_{s}[C[\![X := k]\!](s) = s']p(s)\\
=&\frac{1}{b-a+1}\sum_{k=a}^{b}[C[\![X := k]\!](s_0) = s'] = \frac{1}{b-a+1}\sum_{k=a}^{b}[s_0[X := k] = s']\\
=&\frac{1}{b-a+1}[s' \in \{s_0[X := k] \mid a \le k \le b\}].
\end{aligned}$$

Die while-Schleife mag man als iterierte if-Anweisung auffassen, also

$$\textbf{while}\ b\ \textbf{do}\ c\ \textbf{end}$$

als Lösung c' der Rekurrenz

$$c' = \textbf{if}\ b\ \textbf{then}\ c; c'\ \textbf{else skip end}.$$

Die while-Schleife mag man als iterierte if-Anweisung auffassen, also

$$\textbf{while}\ b\ \textbf{do}\ c\ \textbf{end}$$

als Lösung c' der Rekurrenz

$$c' = \textbf{if}\ b\ \textbf{then}\ c; c'\ \textbf{else skip end}.$$

Diesbezüglich erhält man die Rekurrenz

$$\begin{aligned} D[\![c']\!](p) &= D[\![c; c']\!](b?p) + D[\![\textbf{skip}]\!](\neg b?p) \\ &= D[\![c']\!](D[\![c]\!](b?p)) + (\neg b?p). \end{aligned}$$

Ende.

Januar 2025