Probabilistic program annotations

JP Katoen[a] AK McIver[b] LA Meinicke[b] CC Morgan[c]

[a] RWTH Aachen University, Germany

[b] Macquarie University, Australia

[c] University of New South Wales, Australia

Qualitative programs

We can describe properties using Hoare triples:

$\{0 \leq N\}$ $\leftarrow$ if this holds before
$\mathsf{x}, \mathsf{n} := 0, 0;$

while $\mathsf{n} < N$ do

$\quad (\mathsf{x} := \mathsf{x} + 1 \sqcap \mathsf{skip}); \mathsf{n} := \mathsf{n} + 1$
od
$\{0 \leq \mathsf{x} \leq \mathsf{n} = N\}$ $\leftarrow$ then this holds after

Qualitative programs

$\{P\}$ *prog* $\{Q\}$

when $\sigma \longrightarrow$ *prog* $\longrightarrow \sigma'$

then $P.\sigma \quad \Rightarrow \quad Q.\sigma'$

Qualitative programs

$\{P\}$ *prog* $\{Q\}$ or equivalently $P \Rightarrow$ *wp.prog.Q*

when $\sigma \longrightarrow$ *prog* $\longrightarrow \sigma'$

then $P.\sigma \quad \Rightarrow \quad Q.\sigma'$

Qualitative programs

If we want to verify a given Hoare triple:

$\{0 \leq N\}$
$\mathsf{x}, \mathsf{n} := 0, 0;$

while $\mathsf{n} < N$ do

    $(\mathsf{x} := \mathsf{x}+1 \sqcap \mathsf{skip}); \mathsf{n} := \mathsf{n}+1$
od
$\{0 \leq \mathsf{x} \leq \mathsf{n} = N\}$

Qualitative programs

It is enough to find a valid program annotation from which it can be inferred:

$\{0 \le N\}$
$\mathsf{x}, \mathsf{n} := 0, 0;$
$\{0 \le \mathsf{x} \le \mathsf{n} \le N\}$ $\leftarrow$ these extra annotations
$\mathsf{while}\ \mathsf{n} < N\ \mathsf{do}$
$\{0 \le \mathsf{x} \le \mathsf{n} < N\}$ $\leftarrow$ allow us to prove it
$\quad (\mathsf{x} := \mathsf{x}+1 \sqcap \mathsf{skip}); \mathsf{n} := \mathsf{n}+1$
od
$\{0 \le \mathsf{x} \le \mathsf{n} = N\}$

Valid program annotations

For all annotations P, Q separated by annotation-free *path_prog*:

$\{P\}$ *path_prog* $\{Q\}$

Valid program annotations

For example, for

$$\ldots \quad \{P\} \text{ if } G \text{ then } \mathit{prog}_1 \ \{Q\} \text{ else } \mathit{prog}_2 \text{ fi } \quad \ldots$$

we require

$$\{P\} \text{ ``if } G \text{ then } \mathit{prog}_1 \text{''} \ \{Q\}.$$

Valid program annotations

For example, for

$$\ldots \quad \{P\}\ \textsf{if}\ G\ \textsf{then}\ \textit{prog}_1\ \{Q\}\ \textsf{else}\ \textit{prog}_2\ \textsf{fi} \quad \ldots$$

we require

$$\{P\} \quad \underset{\substack{\uparrow \\ \text{program guard}}}{[G]};\textit{prog}_1 \quad \{Q\}$$

where $\quad [G] \quad \triangleq \quad \textsf{if}\ G\ \textsf{then}\ \textsf{skip}\ \textsf{else}\ \textsf{magic}\ \textsf{fi} \quad .$

Quantitative programs

$\mathsf{x}, \mathsf{n} := 0, 0;$

while $\mathsf{n} < N$ do

$\quad (\mathsf{x} := \mathsf{x} + 1 \;{}_p\!\oplus\, \mathsf{skip}); \mathsf{n} := \mathsf{n} + 1 \quad \leftarrow$ probabilistic choice

od

Quantitative programs

$\mathsf{x}, \mathsf{n} := 0, 0;$

$\mathsf{while}\ \mathsf{n} < N\ \mathsf{do}$

$\quad (\mathsf{x} := \mathsf{x} + 1\ {}_p\!\oplus \mathsf{skip}); \mathsf{n} := \mathsf{n} + 1 \quad \leftarrow$ probabilistic choice

od

This program sets variable x to a value in $[0, N]$ according to the binomial distribution with probability p .

Quantitative programs

Probabilistic programs map each initial state σ to (possibly a set of) final distribution(s) δ' on states.

σ
$\downarrow$

$\mathsf{x}, \mathsf{n} := 0, 0;$

while $\mathsf{n} < N$ do

$\quad (\mathsf{x} := \mathsf{x}+1 \;{}_p\!\oplus\; \mathsf{skip});\ \mathsf{n} := \mathsf{n}+1$

od

$\downarrow$
δ'

Quantitative programs

Probabilistic programs map each initial state σ to (possibly a set of) final distribution(s) δ' on states.

$(\mathsf{x} \mapsto x, \mathsf{n} \mapsto n)$
$\downarrow$

$\mathsf{x}, \mathsf{n} := 0, 0;$

while $\mathsf{n} < N$ do

    $(\mathsf{x} := \mathsf{x} + 1 \ {}_p\!\oplus \ \mathsf{skip}); \ \mathsf{n} := \mathsf{n} + 1$
od

$\downarrow$
$[\mathsf{x} = 0 \wedge \mathsf{n} = N] \times (1-p)^N \quad + \quad [\mathsf{x} = 1 \wedge \mathsf{n} = N] \times \binom{N}{1} (p)(1-p)^{N-1} \quad + \quad \ldots$

Quantitative Hoare logic (McIver, Morgan 2005)

Annotations P, Q are real-valued expressions on the program state.

$\{P\}$ *prog* $\{Q\}$

when $\sigma \longrightarrow$ *prog* $\longrightarrow \delta'$

then $P.\sigma \quad \leq \quad$ expected value of Q over δ'

Quantitative Hoare logic (McIver, Morgan 2005)

Annotations P, Q are real-valued expressions on the program state.

$\{P\}$ *prog* $\{Q\}$ or equivalently $P \leq$ *wp.prog.*Q

when $\sigma \rightarrow$ *prog* $\rightarrow \delta'$

then $P.\sigma \quad \leq \quad$ expected value of Q over δ'

Quantitative Hoare logic (McIver, Morgan 2005)

Annotations P, Q are real-valued expressions on the program state.

$\{P\}$ $\rightarrow$ this is a lower bound on
prog
$\{Q\}$ $\rightarrow$ the least expected value of this

when $\sigma \rightarrow \textit{prog} \rightarrow \delta'$

then $P.\sigma \leq$ expected value of Q over δ'

Quantitative Hoare logic (McIver, Morgan 2005)

$\{\ [\mathsf{true}]\ \}$
$\mathsf{x}, \mathsf{n} := 0, 0;$

$\mathsf{while}\ \mathsf{n} < N\ \mathsf{do}$

$\quad (\mathsf{x} := \mathsf{x} + 1\ {}_{p}\!\oplus\ \mathsf{skip});\ \mathsf{n} := \mathsf{n} + 1$
od
$\{\ [\mathsf{n} \geq N]\ \}$

Quantitative Hoare logic (McIver, Morgan 2005)

$\{ \; [0 \leq N] \times pN \; \}$
$\mathsf{x}, \mathsf{n} := 0, 0;$

while $\mathsf{n} < N$ do

$\quad (\mathsf{x} := \mathsf{x} + 1 \; {}_p\!\oplus \; \mathsf{skip}); \; \mathsf{n} := \mathsf{n} + 1$
od
$\{ \; \mathsf{x} \; \}$

Quantitative Hoare logic (McIver, Morgan 2005)

$\{\ [0 \leq N] \times pN\ \}$
x, n := 0, 0;

while n < N do

(x := x+1 $_p\oplus$ skip); n := n+1
od
$\{\ x\ \}$

How do we prove it?

We need to find a valid quantitative program annotation !

Valid probabilistic program annotations

What's that!?

Valid probabilistic program annotations

1. Annotation at start and end of program.
   - so that we can reason about the correctness of the program as a whole

2. At least one annotation along every program path.
   - verification conditions involve only cycle free program fragments

Valid probabilistic program annotations

What else? Can we put annotations wherever we want?

Valid probabilistic program annotations

What else? Can we put annotations wherever we want?

$$\ldots \quad \{P\}\ (\mathsf{x}{:=}\,0\ \{Q\}\ {}_{\frac{1}{2}}\!\oplus \mathsf{x}{:=}\,1) \quad \ldots$$

Valid probabilistic program annotations

What else? Can we put annotations wherever we want?

$$\ldots \quad \{P\}\ (\mathsf{x}{:=}0\ \{Q\}\ {}_{\frac{1}{2}}\!\oplus \mathsf{x}{:=}1) \quad \ldots$$

What does $\{P\}$ "$\mathsf{x}{:=}0\ {}_{\frac{1}{2}}\!\oplus$" $\{Q\}$ mean?

Valid probabilistic program annotations

What else? Can we put annotations wherever we want?

$$\ldots \quad \{P\}\ (\mathsf{x}{:=}0\ \{Q\}\ {}_{\frac{1}{2}}\!\oplus \mathsf{x}{:=}1) \quad \ldots$$

What does $\{P\}$ "$\mathsf{x}{:=}0\ {}_{\frac{1}{2}}\!\oplus$" $\{Q\}$ mean?

Does it mean $\{\frac{1}{2}{\times}P\}\ \mathsf{x}{:=}0\ \{Q\}$?

Valid probabilistic program annotations

But that is too weak . . .

$$\{1\} \quad (x{:=}\,0 \ \{\tfrac{1}{2}\} \ {}_{\frac{1}{2}}\!\oplus \ x{:=}\,1 \ \{\tfrac{1}{2}\}) \quad \{\tfrac{1}{2}\}$$

Valid probabilistic program annotations

But that is too weak . . .

$$\{1\}\quad (\mathrm{x}{:=}\,0\ \{\tfrac{1}{2}\}\ {}_{\frac{1}{2}}\!\oplus\ \mathrm{x}{:=}\,1\ \{\tfrac{1}{2}\})\quad \{\tfrac{1}{2}\}$$

We have

$$\{\tfrac{1}{2}\times 1\}\quad \mathrm{x}{:=}\,0\quad \{\tfrac{1}{2}\}$$

Valid probabilistic program annotations

But that is too weak . . .

$$\{1\} \quad (x{:=}0\ \{\tfrac{1}{2}\}\ {}_{\frac{1}{2}}\!\oplus x{:=}1\ \{\tfrac{1}{2}\}) \quad \{\tfrac{1}{2}\}$$

We have

$$\{\tfrac{1}{2}\times 1\} \quad x{:=}0 \quad \{\tfrac{1}{2}\}$$
$$\{\tfrac{1}{2}\times 1\} \quad x{:=}1 \quad \{\tfrac{1}{2}\}$$

Valid probabilistic program annotations

But that is too weak . . .

$$\{1\} \quad (\mathsf{x}{:=}\,0\ \{\tfrac{1}{2}\}\ {}_{\frac{1}{2}}\!\oplus \mathsf{x}{:=}\,1\ \{\tfrac{1}{2}\}) \quad \{\tfrac{1}{2}\}$$

We have

$$\begin{array}{lll} \{\tfrac{1}{2}\times 1\} & \mathsf{x}{:=}\,0 & \{\tfrac{1}{2}\} \\ \{\tfrac{1}{2}\times 1\} & \mathsf{x}{:=}\,1 & \{\tfrac{1}{2}\} \\ \{\tfrac{1}{2}\} & \mathsf{skip} & \{\tfrac{1}{2}\} \end{array}$$

Valid probabilistic program annotations

But that is too weak . . .

$$\{1\} \quad (\mathsf{x}{:=}\,0\ \{\tfrac{1}{2}\}\ {}_{\frac{1}{2}}\!\oplus \mathsf{x}{:=}\,1\ \{\tfrac{1}{2}\}) \quad \{\tfrac{1}{2}\}$$

We have

$$\begin{array}{lll}
\{\frac{1}{2}\times 1\} & \mathsf{x}{:=}\,0 & \{\frac{1}{2}\} \\
\{\frac{1}{2}\times 1\} & \mathsf{x}{:=}\,1 & \{\frac{1}{2}\} \\
\{\frac{1}{2}\} & \mathsf{skip} & \{\frac{1}{2}\} \\
\{\frac{1}{2}\} & \mathsf{skip} & \{\frac{1}{2}\}
\end{array}$$

Valid probabilistic program annotations

But that is too weak . . .

$$\{1\} \quad (x{:=}0\ \{\tfrac{1}{2}\}\ {}_{\frac{1}{2}}\!\oplus x{:=}1\ \{\tfrac{1}{2}\}) \quad \{\tfrac{1}{2}\}$$

We have

$$\begin{array}{lll}
\{\frac{1}{2}\times 1\} & x{:=}0 & \{\frac{1}{2}\} \\
\{\frac{1}{2}\times 1\} & x{:=}1 & \{\frac{1}{2}\} \\
\{\frac{1}{2}\} & \mathsf{skip} & \{\frac{1}{2}\} \\
\{\frac{1}{2}\} & \mathsf{skip} & \{\frac{1}{2}\}
\end{array}$$

but $\{1\} \quad (x{:=}0\ {}_{\frac{1}{2}}\!\oplus x{:=}1) \quad \{\frac{1}{2}\}$?

Valid probabilistic program annotations

What else? Can we put annotations wherever we want?

$$\ldots \quad \{P\}\ (\mathsf{x}{:=}0\ \{Q\}\ {}_{\frac{1}{2}}\!\oplus \mathsf{x}{:=}1) \quad \ldots$$

What does $\{P\}$ “$\mathsf{x}{:=}0\ {}_{\frac{1}{2}}\!\oplus$” $\{Q\}$ mean?

Valid probabilistic program annotations

What else? Can we put annotations wherever we want?

$$\ldots \quad \{P\}\ (\mathsf{x}{:=}0\ \{Q\}\ {}_{\frac{1}{2}}{\oplus}\ \mathsf{x}{:=}1) \quad \ldots$$

What does $\{P\}$ "$\mathsf{x}{:=}0\ {}_{\frac{1}{2}}{\oplus}$" $\{Q\}$ mean?

Does it mean $\{P\}\ \mathsf{x}{:=}0\ \{Q\}$?

Valid probabilistic program annotations

But that is too strong . . .

$$\{\tfrac{1}{2}\} \quad (x{:=}\,0\ \{[x = 0]\}\ {}_{\frac{1}{2}}\!\oplus x{:=}\,1\ \{[x = 0]\}) \quad \{[x = 0]\}$$

Valid probabilistic program annotations

But that is too strong . . .

$$\{\tfrac{1}{2}\} \quad (x := 0 \ \{[x = 0]\} \ {}_{\frac{1}{2}}\!\oplus \ x := 1 \ \{[x = 0]\}) \quad \{[x = 0]\}$$

We have

$$\{\tfrac{1}{2}\} \quad x := 0 \quad \{[x = 0]\}$$

Valid probabilistic program annotations

But that is too strong ...

$$\{\tfrac{1}{2}\} \quad (x{:=}\,0 \ \{[x = 0]\} \ {}_{\frac{1}{2}}\!\oplus x{:=}\,1 \ \{[x = 0]\}) \quad \{[x = 0]\}$$

We have

$$\{\tfrac{1}{2}\} \quad x{:=}\,0 \quad \{[x = 0]\}$$
$$\{\tfrac{1}{2}\} \quad x{:=}\,1 \quad \{[x = 0]\}$$

Valid probabilistic program annotations

But that is too strong . . .

$$\{\tfrac{1}{2}\} \quad (x{:=}\,0\ \{[x = 0]\}\ {}_{\frac{1}{2}}\!\oplus\ x{:=}\,1\ \{[x = 0]\}) \quad \{[x = 0]\}$$

We have

$$\{\tfrac{1}{2}\} \quad x{:=}\,0 \quad \{[x = 0]\}$$
$$\cancel{\{\tfrac{1}{2}\} \quad x{:=}\,1 \quad \{[x = 0]\}}$$

Valid probabilistic program annotations

But that is too strong . . .

$$\{\tfrac{1}{2}\} \quad (\mathsf{x}{:=}0 \ \{[\mathsf{x}=0]\} \ {}_{\frac{1}{2}}\!\oplus \mathsf{x}{:=}1 \ \{[\mathsf{x}=0]\}) \quad \{[\mathsf{x}=0]\}$$

We have

$$\{\tfrac{1}{2}\} \quad \mathsf{x}{:=}0 \quad \{[\mathsf{x}=0]\}$$
$$\cancel{\{\tfrac{1}{2}\} \quad \mathsf{x}{:=}1 \quad \{[\mathsf{x}=0]\}}$$

but $\{\tfrac{1}{2}\} \quad (\mathsf{x}{:=}0 \ {}_{\frac{1}{2}}\!\oplus \mathsf{x}{:=}1) \quad \{[x=0]\}$ . . .

Valid probabilistic program annotations

Clearly we need to treat branching paths together somehow . . .

Valid probabilistic program annotations

But even if we try and treat them together . . .

$$
\begin{array}{l}
\quad\vdots \\
\{P\} \\
(\mathsf{x}{:=}\,0 \;{}_{\frac{1}{2}}{\oplus}\; \mathsf{x}{:=}\,1); \\
\textsf{if } \mathsf{x} = 0 \textsf{ then } \mathsf{y}{:=}\,0 \;\{Q\}\; \textsf{else } \mathsf{y}{:=}\,1 \textsf{ fi} \\
\quad\vdots
\end{array}
$$

Valid probabilistic program annotations

But even if we try and treat them together . . .

$$\begin{array}{l} \quad\vdots \\ \{P\} \\ (\mathsf{x}{:=}\,0 \;{}_{\frac{1}{2}}\!\oplus\, \mathsf{x}{:=}\,1); \\ \mathsf{if}\ \mathsf{x} = 0\ \mathsf{then}\ \mathsf{y}{:=}\,0\ \ \{Q\}\ \mathsf{else}\ \mathsf{y}{:=}\,1\ \mathsf{fi} \\ \quad\vdots \end{array}$$

What does $\{P\}$ "$(\mathsf{x}{:=}\,0 \;{}_{\frac{1}{2}}\!\oplus\, \mathsf{x}{:=}\,1);\mathsf{if}\ \mathsf{x} = 0\ \mathsf{then}\ \mathsf{y}{:=}\,0$" $\{Q\}$ mean?

Valid probabilistic program annotations

But even if we try and treat them together . . .

$$
\begin{array}{l}
\quad\vdots \\
\{P\} \\
(\mathsf{x}{:=}\,0 \ {}_{\frac{1}{2}}\!\oplus\, \mathsf{x}{:=}\,1); \\
\mathsf{if}\ \mathsf{x}=0\ \mathsf{then}\ \mathsf{y}{:=}\,0\ \{Q\}\ \mathsf{else}\ \mathsf{y}{:=}\,1\ \mathsf{fi} \\
\quad\vdots
\end{array}
$$

Does it mean $\{P\}\ (\mathsf{x}{:=}\,0 \ {}_{\frac{1}{2}}\!\oplus\, \mathsf{x}{:=}\,1); [\mathsf{x}=0]\ \mathsf{then}\ \mathsf{y}{:=}\,0\ \ \{Q\}$?

where $[\mathsf{x}=0] \ \triangleq\ \mathsf{if}\ \mathsf{x}=0\ \mathsf{then}\ \mathsf{skip}\ \mathsf{else}\ \mathsf{magic}\ \mathsf{fi}$.

Valid probabilistic program annotations

But even if we try and treat them together . . .

$\{\frac{1}{2}\}$
$(x{:=}\,0\ {}_{\frac{1}{2}}\!\oplus x{:=}\,1);$
if $x = 0$ then $y{:=}\,0$ $\{[y{=}3]\}$ else $y{:=}\,1$ $\{[y{=}3]\}$ fi
$\{[y{=}3]\}$

We have

$$\{\tfrac{1}{2}\}\ (x{:=}\,0\ {}_{\frac{1}{2}}\!\oplus x{:=}\,1);\,[x = 0];\,y{:=}\,0 \quad \{[y{=}3]\}$$

Valid probabilistic program annotations

But even if we try and treat them together . . .

$\{\frac{1}{2}\}$
$(x{:=}\,0 \;{}_{\frac{1}{2}}{\oplus}\; x{:=}\,1);$
$\text{if } x = 0 \text{ then } y{:=}\,0 \;\{[y{=}3]\}\; \text{else } y{:=}\,1 \;\{[y{=}3]\}\; \text{fi}$
$\{[y{=}3]\}$

We have

$$\{\tfrac{1}{2}\}\ (x{:=}\,0; y{:=}\,0 \;{}_{\frac{1}{2}}{\oplus}\; \text{magic}) \quad \{[y{=}3]\}$$

Valid probabilistic program annotations

But even if we try and treat them together . . .

$\{\frac{1}{2}\}$
$(x{:=}\,0 \;{}_{\frac{1}{2}}\!\oplus\; x{:=}\,1)$;
if $x = 0$ then $y{:=}\,0$ $\{[y{=}3]\}$ else $y{:=}\,1$ $\{[y{=}3]\}$ fi
$\{[y{=}3]\}$

We have

$\{\frac{1}{2}\}$ $(x{:=}\,0 \;{}_{\frac{1}{2}}\!\oplus\; x{:=}\,1); [x = 0]; y{:=}\,0$ $\{[y{=}3]\}$
$\{\frac{1}{2}\}$ $(x{:=}\,0 \;{}_{\frac{1}{2}}\!\oplus\; x{:=}\,1); [x \neq 0]; y{:=}\,1$ $\{[y{=}3]\}$
$\{[y = 3]\}$ skip $\{[y = 3]\}$
$\{[y = 3]\}$ skip $\{[y = 3]\}$

but $\{\frac{1}{2}\}$ $(x{:=}\,0 \;{}_{\frac{1}{2}}\!\oplus\; x{:=}\,1)$; if $x = 0$ then $y{:=}\,0$ else $y{:=}\,1$ fi $\{[y{=}3]\}$?

Valid probabilistic program annotations

We don't constrain where you can put annotations . . .

But we do sometimes require you to add more!

Valid probabilistic program annotations

3. If there is an interior annotation following a choice point but before the choice rejoins:

$$\ldots \quad \mathit{prog}_1 \; ; \; \underset{\substack{\uparrow \\ \text{choice point}}}{\mathsf{if}} \; G \; \mathsf{then} \; \mathit{prog}_2 \;\; \underset{\substack{\uparrow \\ \text{interior annotation}}}{\{P\}} \; \mathit{prog}_3 \; \mathsf{else} \; \mathit{prog}_4 \; \mathsf{fi} \quad \underset{\substack{\uparrow \\ \text{choice rejoins}}}{\ldots}$$

Valid probabilistic program annotations

3. If there is an interior annotation following a choice point but before the choice rejoins:

$$\begin{array}{ccccccccc} \ldots & \mathit{prog}_1 & ; & \text{if } G \text{ then } \mathit{prog}_2 & \{P\} & \mathit{prog}_3 \text{ else } \mathit{prog}_4 \text{ fi} & \ldots \\ & & \uparrow & & \uparrow & & \uparrow \\ & & \text{choice point} & & \text{interior annotation} & & \text{choice rejoins} \end{array}$$

Then we must add extra annotations:

$$\begin{array}{cccccccccc} \ldots & \mathit{prog}_1 & \{Q\} & \text{if } G \text{ then} & \{R\} & \mathit{prog}_2 \ \{P\} \ \mathit{prog}_3 \text{ else} & \{S\} & \mathit{prog}_4 \text{ fi} & \ldots \\ & & \uparrow & & \uparrow & & \uparrow \\ & & \text{at the choice point} & & \text{at the first exit} & & \text{at the second exit} \end{array}$$

Valid probabilistic program annotations

4. For all annotations P, Q separated by annotation-free *path_prog*:

   If P is not at a choice point:

   $$\{P\} \quad \textit{path\_prog} \quad \{Q\}$$

Valid probabilistic program annotations

4. If annotation P is at a choice point:

$$\ldots \quad \{P\} \; (\{Q\}; \mathit{prog}_1 \;\; {}_{\frac{1}{2}}\!\oplus \;\; \{R\} \; \mathit{prog}_2) \quad \ldots$$

Condition $\{P\}$ "${}_{\frac{1}{2}}\!\oplus$" $\{Q\}$ does not make sense . . .

Valid probabilistic program annotations

4. If annotation P is at a choice point:

$$\ldots \quad \{P\}\ (\{Q\};\, \mathit{prog}_1 \ {}_{\frac{1}{2}}\!\oplus\ \{R\}\ \mathit{prog}_2) \quad \ldots$$

We require $\quad P \leq \frac{1}{2}{\times}Q + \frac{1}{2}{\times}R \quad .$

Valid probabilistic program annotations

4. If annotation P is at a choice point:

$$\ldots \quad \{P\} \text{ if } G \text{ then } \{Q\}; \textit{prog}_1 \text{ else } \{R\} \ \textit{prog}_2 \text{ fi} \quad \ldots$$

We require $P \leq [G]{\times}Q + [\neg G]{\times}R$.

Valid probabilistic program annotations

4. If annotation P is at a choice point:

$$\ldots \quad \{P\} \text{ while } \; G \text{ then } \{Q\} \;\; \textit{body} \;\; \text{od } \{R\} \quad \ldots$$

We require $P \leq [G] \times Q + [\neg G] \times R$.

So . . .

Now we know what a probabilistic program annotation is.

How do we find them?

Generating qualitative program annotations

Possible for annotations of restricted forms for restricted types of programs.

Methods include

- iterative fixed-point methods like abstract interpretation (Cousot and Cousot, 1977), and
- constraint-based approaches (e.g., Colón et al., 2003; Podelski and Rybalchenko, 2004; Cousot, 2005; Monniaux, 2000; Gulwani et al., 2008).

Constraint-based approaches

1. Speculatively annotate with parametrised expressions of a given form.

2. Calculate verification conditions.

3. Translate them to machine solvable form.

4. Solve for the annotation parameters using a constraint solver.

How can we adapt these methods to our needs?

How can we adapt these methods to our needs?

Define a constraint-based method for generating linear program annotations for linear probabilistic programs.

It generalises the approach of Colón et al. (2003)

A constraint-based approach

For linear programs ...

$\mathsf{x}, \mathsf{n} := 0, 0;$ $\leftarrow$ variables are real-valued

while $\mathsf{n} < N$ do $\leftarrow$ guards are linear constraints

$(\mathsf{x} := \mathsf{x}+1 \; {}_p\!\oplus \; \mathsf{skip}); \mathsf{n} := \mathsf{n}+1$ $\leftarrow$ updates are linear expressions
od probabilities are constant

A constraint-based approach

```
x, n := 0, 0;

while n < N do

    (x := x+1 p⊕ skip); n := n+1
od;

skip
```

1. Speculatively annotate with propositionally linear expressions

$$\textstyle\sum_{m:\,[1..M]}[\bigwedge_{n:\,[1..N]} \alpha_{(mn,1)}\mathsf{x}_1 + \ldots + \alpha_{(mn,X)}\mathsf{x}_X + \alpha_{(mn,X+1)} \ll 0] \\ \times (\beta_{(m,1)}\mathsf{x}_1 + \ldots + \beta_{(m,X)}\mathsf{x}_X + \beta_{(m,X+1)})$$

   where x_i are program variables, α_i, β_i are parameters and $\ll$ is $\leq$ or $<$.

A constraint-based approach

$\{ \ [0 \leq N] \times pN \ \}$
x, n:= 0, 0;
$\{ \ [0 \leq \mathsf{x} \leq \mathsf{n} \leq N{+}1] \times (\alpha\mathsf{x} - \beta\mathsf{n} + \gamma) \ \}$
while n<N do
$\{ \ [0 \leq \mathsf{x} \leq \mathsf{n} < N] \times (\alpha\mathsf{x} - \beta\mathsf{n} + \gamma) \ \}$
    (x:= x+1 $_p\oplus$ skip); n:= n+1
od;
$\{ \ [0 \leq \mathsf{x} \leq \mathsf{n} \leq N{+}1 \wedge \mathsf{n} \geq N] \times (\alpha\mathsf{x} - \beta\mathsf{n} + \gamma) \ \}$
skip
$\{ \ \mathsf{x} \ \}$

1. Speculatively annotate with propositionally linear expressions

$$\sum_{m:\,[1..M]} [\bigwedge_{n:\,[1..N]} \alpha_{(mn,1)}\mathsf{x}_1 + \ldots + \alpha_{(mn,X)}\mathsf{x}_X + \alpha_{(mn,X+1)} \ll 0] \\ \times (\beta_{(m,1)}\mathsf{x}_1 + \ldots + \beta_{(m,X)}\mathsf{x}_X + \beta_{(m,X+1)})$$

   where x_i are program variables, α_i, β_i are parameters and $\ll$ is $\leq$ or $<$.

A constraint-based approach

$\{\ [0 \leq N]\times pN\ \}$
x, n:= 0, 0;
$\{\ [0 \leq \mathsf{x} \leq \mathsf{n} \leq N{+}1]\times(\alpha\mathsf{x} - \beta\mathsf{n} + \gamma)\ \}$
while n$<N$ do
$\{\ [0 \leq \mathsf{x} \leq \mathsf{n} < N]\times(\alpha\mathsf{x} - \beta\mathsf{n} + \gamma)\ \}$
    (x:= x+1 $_p\oplus$ skip); n:= n+1
od;
$\{\ [0 \leq \mathsf{x} \leq \mathsf{n} \leq N{+}1 \wedge \mathsf{n} \geq N]\times(\alpha\mathsf{x} - \beta\mathsf{n} + \gamma)\ \}$
skip
$\{\ \mathsf{x}\ \}$

2. Calculate verification conditions

   [Inequalities between propositionally linear expressions.] E.g.

$$
\begin{array}{ll}
 & [0 \leq \mathsf{x} \leq \mathsf{n} < N]\times(\alpha\mathsf{x} + \beta\mathsf{n} + \gamma) \\
\leq & \\
 & [0 \leq \mathsf{x}{+}1 \leq \mathsf{n}{+}1 \leq N{+}1]\times(p\alpha(\mathsf{x}{+}1) + p\beta(\mathsf{n}{+}1) + p\gamma) + \\
 & [0 \leq \mathsf{x} \leq \mathsf{n}{+}1 \leq N{+}1]\times((1{-}p)\alpha\mathsf{x} + (1{-}p)\beta(\mathsf{n}{+}1) + (1{-}p)\gamma)
\end{array}
$$

A constraint-based approach

$\{ \ [0 \leq N] \times pN \ \}$
$\mathsf{x}, \mathsf{n} := 0, 0;$
$\{ \ [0 \leq \mathsf{x} \leq \mathsf{n} \leq N+1] \times (\alpha\mathsf{x} - \beta\mathsf{n} + \gamma) \ \}$
while $\mathsf{n} < N$ do
$\{ \ [0 \leq \mathsf{x} \leq \mathsf{n} < N] \times (\alpha\mathsf{x} - \beta\mathsf{n} + \gamma) \ \}$
    $(\mathsf{x} := \mathsf{x}+1 \ {}_p\!\oplus \mathsf{skip}); \mathsf{n} := \mathsf{n}+1$
od;
$\{ \ [0 \leq \mathsf{x} \leq \mathsf{n} \leq N+1 \wedge \mathsf{n} \geq N] \times (\alpha\mathsf{x} - \beta\mathsf{n} + \gamma) \ \}$
skip
$\{ \ \mathsf{x} \ \}$

3. Translate them to universally quantified Boolean expressions on linear inequalities

   E.g. (assuming non-negativity of expressions on previous slide)

$$[0 \leq \mathsf{x} \leq \mathsf{n} < N] \Rightarrow [0 \leq p\alpha + \beta]$$

A constraint-based approach

Then we proceed just as Colón et al (2003) for qualitative case.

4. Translate them to machine-solvable form.
   [Use Farka's lemma to translate to existentially quantified, polynomial constraints on annotation parameters.]

5. Solve for annotation parameters using constraint solvers.
   [E.g. REDLOG]

A constraint-based approach

$\{\ [0 \leq N]\times pN\ \}$
$\mathsf{x}, \mathsf{n} := 0, 0;$
$\{\ [0 \leq \mathsf{x} \leq \mathsf{n} \leq N{+}1]\times(\mathsf{x} - p\mathsf{n} + pN)\ \}$
while $\mathsf{n} < N$ do
$\{\ [0 \leq \mathsf{x} \leq \mathsf{n} < N]\times(\mathsf{x} - p\mathsf{n} + pN)\ \}$
    $(\mathsf{x} := \mathsf{x}+1\ {}_p\!\oplus\ \mathsf{skip});\ \mathsf{n} := \mathsf{n}+1$
od;
$\{\ [0 \leq \mathsf{x} \leq \mathsf{n} \leq N{+}1 \wedge \mathsf{n} \geq N]\times(\mathsf{x} - p\mathsf{n} + pN)\ \}$
skip
$\{\ \mathsf{x}\ \}$

“Fully automated” method?

“Fully automated” method?

We envisage ourselves using it interactively.

Conclusion

We have defined what valid probabilistic program annotations are.

And we have proposed a contraint-based method of generating linear forms of annotations for linear probabilistic programs.

Alternative automated methods: comparison

We could use model checking:

- Apply MDP model checking. [LiQuor, PRISM]
  - works for program instances, but no general solution
- Use abstraction-refinement techniques. [PASS, PRISM]
  - loop analysis with real variables does not work well
- Check language equivalence. [APEX]
  - cannot deal with parametrised probabilistic programs