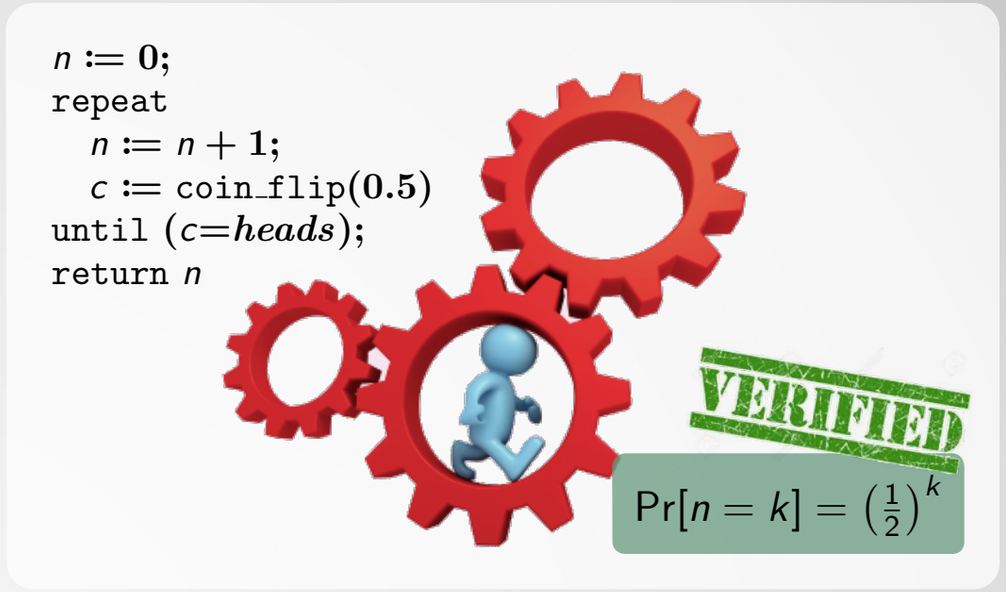
Seminar on

"Verification of Probabilistic Programs"



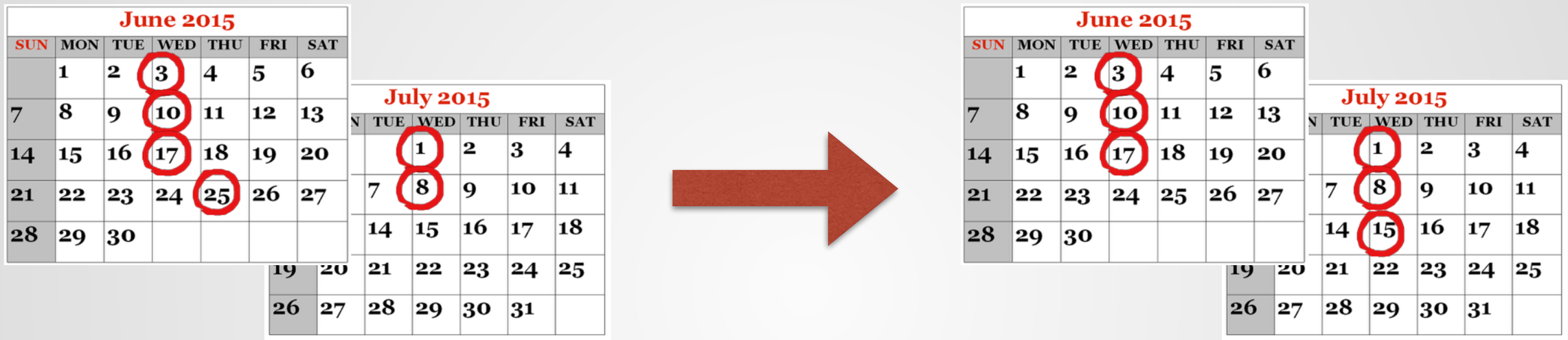
Lecture 3:

Probabilisitic Predicate Transformers II

Federico Olmedo

Software Modeling and Verification Group
RWTH AACHEN UNIVERSITY

Lecture Reschedule



- Lecture on 25th June cancelled
- New lecture on 15th July

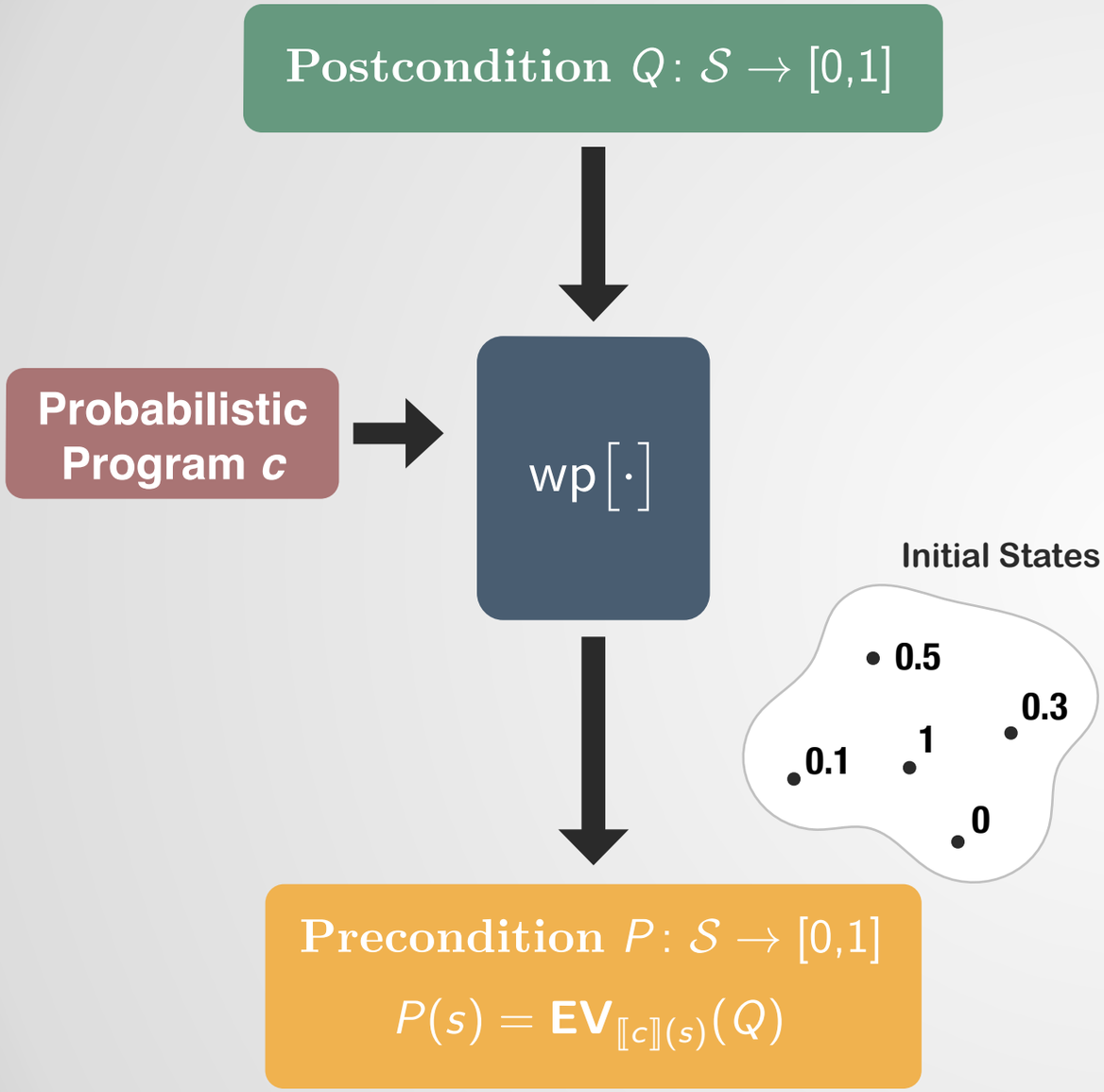
Agenda

- Recap on previous lecture
- Algebraic properties
- Extension to unbounded expectations
- Connection to relational semantics
- Extension to non-deterministic programs
- Summary

Agenda

- Recap on previous lecture
- Algebraic properties
- Extension to unbounded expectations
- Connection to relational semantics
- Extension to non-deterministic programs
- Summary

Expectation Transformers — Recap



(post-)expectation (pre-)expectation

$$\mathsf{wp}[c]\colon (\mathcal{S} \to [0,1]) \to (\mathcal{S} \to [0,1])$$

Characteristic function of event E

- $\mathsf{wp}[c]([E])$ → probability of terminating and establishing E.

- $\mathsf{wlp}[c]([E])$ → probability of diverging or establishing E.

Expectation Transformers — Recap

The pGCL language

$\mathcal{C}$	:=	`skip`	nop
	|	`abort`	abortion
	|	$x := E$	assignment
	|	`if` G `then` $\mathcal{C}$ `else` $\mathcal{C}$	conditional
	|	$\{\mathcal{C}\}\ [\mathbf{p}]\ \{\mathcal{C}\}$	**probabilistic choice**
	|	`while` G `do` $\mathcal{C}$	while loop
	|	$\mathcal{C}; \mathcal{C}$	sequence

Theorem (McIver & Morgan '96)

For **pGCL** program c, expectation transformer $wp[c]$ can be defined by induction on c structure.

$$
\begin{aligned}
\mathrm{wp}[\texttt{skip}](f) &= f \\
\mathrm{wp}[\texttt{abort}](f) &= \underline{0} \\
\mathrm{wp}[x := E](f) &= f[E/x] \\
\mathrm{wp}[\texttt{if } G \texttt{ then } c_1 \texttt{ else } c_2](f) &= [G] \cdot \mathrm{wp}[c_1](f) + [\neg G] \cdot \mathrm{wp}[c_2](f) \\
\mathrm{wp}[\{c_1\}\ [p]\ \{c_2\}](f) &= p \cdot \mathrm{wp}[c_1](f) + (1-p) \cdot \mathrm{wp}[c_2](f) \\
\mathrm{wp}[c_1; c_2](f) &= (\mathrm{wp}[c_1] \circ \mathrm{wp}[c_2])(f) \\
\mathrm{wp}[\texttt{while } G \texttt{ do } c](f) &= \mu h \bullet ([G] \cdot \mathrm{wp}[c](h) + [\neg G] \cdot f)
\end{aligned}
$$

Expectation Transformers — Recap

Example

$$c_1: \quad \{x := 0\}\ [p]\ \{x := 1\};\ \{y := 0\}\ [q]\ \{y := 1\}$$

$$\begin{aligned}
\mathsf{wp}[x := E](f) &= f[E/x] \\
\mathsf{wp}[\{c_1\}\ [p]\ \{c_2\}](f) &= p \cdot \mathsf{wp}[c_1](f) + \bar{p} \cdot \mathsf{wp}[c_2](f) \\
\mathsf{wp}[c_1; c_2](f) &= \mathsf{wp}[c_1](\mathsf{wp}[c_2](f))
\end{aligned}$$

$$\begin{aligned}
& \mathsf{wp}[c_1]([x{\leq}y]) \\
= \quad & \langle \text{rule for sequential composition} \rangle \\
& \mathsf{wp}\big[\{x := 0\}\ [p]\ \{x := 1\}\big]\big(\mathsf{wp}\big[\{y := 0\}\ [q]\ \{y := 1\}\big]([x{\leq}y])\big) \\
= \quad & \langle \text{rule for probabilistic choice} \rangle \\
& \mathsf{wp}\big[\{x := 0\}\ [p]\ \{x := 1\}\big]\big(q \cdot \mathsf{wp}\big[y := 0\big]([x{\leq}y]) + \bar{q} \cdot \mathsf{wp}\big[y := 1\big]([x{\leq}y])\big) \\
= \quad & \langle \text{rule for assignment, twice} \rangle \\
& \mathsf{wp}\big[\{x := 0\}\ [p]\ \{x := 1\}\big]\big(q \cdot [x{\leq}0] + \bar{q} \cdot [x{\leq}1]\big) \\
= \quad & \langle \text{rule for probabilistic choice} \rangle \\
& p \cdot \mathsf{wp}\big[x := 0\big]\big(q \cdot [x{\leq}0] + \bar{q} \cdot [x{\leq}1]\big) + \bar{p} \cdot \mathsf{wp}\big[x := 1\big]\big(q \cdot [x{\leq}0] + \bar{q} \cdot [x{\leq}1]\big) \\
= \quad & \langle \text{rule for assignment, twice} \rangle \\
& p \cdot \big(q \cdot [0{\leq}0] + \bar{q} \cdot [0{\leq}1]\big) + \bar{p} \cdot \big(q \cdot [1{\leq}0] + \bar{q} \cdot [1{\leq}1]\big) \\
= \quad & \langle \text{algebra} \rangle \\
& \underline{p + \bar{p}\bar{q}}
\end{aligned}$$

Expectation Transformers — Recap

Loop Rules

Consider loop `while` G `do` c and post-expectation f. Assume that

(1) there exists a standard (ie a predicate) loop invariant I, which restricted to $\neg G$ is stronger than the post-expectation f, and

(2) there exists a bounded variant e, which in each iteration decreases with at least a fixed probability $\epsilon > 0$.

Then, $[\![I]\!]$ is a valid pre-expectation of the loop w.r.t. post-expectation f (but not necessarily the weakest).

(1) ⇒ **partial correctness (wlp)** (1) + (2) ⇒ **total correctness (wp)**

Agenda

- Recap on previous lecture
- Algebraic properties
- Extension to unbounded expectations
- Connection to relational semantics
- Extension to non-deterministic programs
- Summary

Healthiness Conditions

For any (purely probabilistic) **pGCL** program c, it holds

- Monotonicity

$$\mathsf{wp}[c](f) \leq \mathsf{wp}[c](g) \quad \text{and} \quad \mathsf{wlp}[c](f) \leq \mathsf{wlp}[c](g) \qquad \text{if} \quad f \leq g$$

- Feasibility

$$\mathsf{wp}[c](f) \leq \underline{u} \qquad \text{if} \quad \forall s \bullet f(s) \leq u$$
$$\mathsf{wlp}[c](f) \geq \underline{u} \qquad \text{if} \quad \forall s \bullet f(s) \geq u$$

- Linearity

$$\mathsf{wp}[c](\alpha \cdot f + \beta \cdot g) = \alpha \cdot \mathsf{wp}[c](f) + \beta \cdot \mathsf{wp}[c](g)$$

- Strictness / Co-strictness

$$\mathsf{wp}[c](\underline{0}) = \underline{0} \quad \text{and} \quad \mathsf{wlp}[c](\underline{1}) = \underline{1}$$

- Connection between both transformers

$$\mathsf{wlp}[c](f) = \underline{1} - \mathsf{wp}[c](\underline{1} - f)$$

Agenda

- Recap on previous lecture
- Algebraic properties
- Extension to unbounded expectations
- Connection to relational semantics
- Extension to non-deterministic programs
- Summary

Generalizing Expectations

$$
\begin{array}{ll}
c: & n := 0; \\
& b := \mathsf{true}; \\
& \mathtt{while}\,(b{=}\mathsf{true})\,\mathtt{do} \\
& \quad \{b := \mathsf{true}\}\ [1/2]\ \{b := \mathsf{false}\}; \\
& \quad n := n{+}1;
\end{array}
$$

Probability that the program terminates within 5 iterations → $\mathrm{wp}[c]([n{\leq}5])$

Expected number of iterations to terminate → $\mathrm{wp}[c](\lambda s \bullet s(n))$

Generalizing Expectations

$$
\begin{aligned}
c:\quad & n := 0;\\
& b := \mathsf{true};\\
& \texttt{while}\,(b{=}\mathsf{true})\,\texttt{do}\\
& \quad \{b := \mathsf{true}\}\ [1/2]\ \{b := \mathsf{false}\};\\
& \quad n := n{+}1;
\end{aligned}
$$

Probability that the program terminates within 5 iterations → $\mathrm{wp}[c]([n{\leq}5])$

Expected number of iterations to terminate → $\mathrm{wp}[c](\lambda s \bullet s(n))$

$\notin \mathcal{S} \to [0,1]$

Generalizing Expectations

- If post-expectation f is bounded by some constant $M>1$, ie

$$\forall s \bullet\ f(s) \in [0,M]$$

we can exploit the linearity of wp[c]:

$$\mathsf{wp}[c](f) \;=\; M \cdot \mathsf{wp}[c]\left(\tfrac{1}{M} \cdot f\right)$$

$\in \mathcal{S} \to [0,1]$ GO

- If post-expectation f is unbounded, the technique can be adapted:
  - the range of expectations must also include ∞, ie $\mathbb{E} \triangleq \mathcal{S} \to \mathbb{R}^{\geq 0}_{\infty}$
  - the inductive definition of $\mathsf{wp}[\cdot]$ remains the same
  - the loop rule must be adapted

- If post-expectation f takes both positive and negative values, pre-expectation is not guaranteed to exist

(post-)expectation (pre-)expectation

$$\mathsf{wp}[c] : (\mathcal{S} \to [0,1]) \to (\mathcal{S} \to [0,1])$$

$$\mathsf{wp}[c](f) \;=\; \lambda s \bullet \mathbf{EV}_{[\![c]\!](s)}(f)$$

c: $n := 0$;
$b :=$ true;
while $(b{=}\text{true})$ do
$\{b := \text{true}\}\ [1/2]\ \{b := \text{false}\}$;
$n := n{+}1$;

$$\mathsf{wp}[c](f)(s) \;=\; \sum_{i\geq 1} \left(\tfrac{1}{2}\right)^i f(s[b, n/\text{false}, i])$$

$$\mathsf{wp}[c]\left(2^n\right)(s) \;=\; \sum_{i\geq 1} 1 = \infty$$

$$\begin{aligned}\mathsf{wp}[c]\left((-2)^n\right)(s) \;&=\; \sum_{i\geq 1} (-1)^n \\ &=\; -1+1-1+1\cdots\end{aligned}$$

Generalizing Expectations

Loop Rule for Unbounded Expectations

Consider loop `while` G `do` c. Assume that

(1) $T \subseteq \mathcal{S}$ is a set of initial states that guarantee almost sure termination of the loop, ie

$$[T] \Rrightarrow \mathrm{wp}[\texttt{while}\ G\ \texttt{do}\ c](\underline{1})$$

(Use the original loop rule to determine T.)

(2) $I : \mathcal{S} \to \mathbb{R}^{\geq 0}_{\infty}$ is a loop invariant, ie

$$I \cdot [G] \Rrightarrow \mathrm{wp}[c](I)$$

(3) the expected value of $I \cdot [G]$ upon the entry of the loop body tends to zero as the loop "continues to execute" (ie in the long run).

Then

$$I \cdot [T] \Rrightarrow \mathrm{wp}[\texttt{while}\ G\ \texttt{do}\ c](I \cdot [\neg G])$$

Generalizing Expectations

Application of loop rule for unbounded expectations

We want to prove that the program terminates **in average** in two iterations, i.e.

$$\mathsf{wp}[c](n) \;=\; \underline{2}$$

c :
$n := 0;$
$b := \mathsf{true};$
$\mathtt{while}\,(b{=}\mathsf{true})\,\mathtt{do}$
$\quad \{b := \mathsf{true}\}\ [1/2]\ \{b := \mathsf{false}\};$
$\quad n := n{+}1;$

$\mathsf{wp}[c](n)$

$=$ $\quad \langle$rule for seq. comp.$\rangle$

$\mathsf{wp}\big[n := 0;\ b := \mathsf{true}\big]\big(\mathsf{wp}\big[\mathtt{while}\,(b{=}\mathsf{true})\,\mathtt{do}\ldots\big](n)\big)$

$\Leftarrow$ $\quad \langle$monoton. of $\mathsf{wp}[\cdot]$; let $I := [b]\cdot(n+2) + [\neg b]\cdot n\rangle$

$\mathsf{wp}\big[n := 0;\ b := \mathsf{true}\big]\big(\mathsf{wp}\big[\mathtt{while}\ (b{=}\mathsf{true})\ \mathtt{do}\ \mathit{body}\big]\big(I\cdot[\neg b]\big)\big)$

$\Leftarrow$ $\quad \langle$monoton. of $\mathsf{wp}[\cdot]$, loop rule with invariant I
and set of initial terminating states $\mathcal{S}\rangle$

$\mathsf{wp}\big[n := 0;\ b := \mathsf{true}\big]\big([b]\cdot(n+2) + [\neg b]\cdot n\big)$

$=$ $\quad \langle$rule for seq. comp. and assign.$\rangle$

$\underline{2}$

(1) $I\cdot[\neg b] \;\Rrightarrow\; n$

(2) $\mathsf{true} \;\Rrightarrow\; \mathsf{wp}\big[\mathtt{while}\ b\ \mathtt{do}\ \mathit{body}\big](\underline{1})$

(3) $I\cdot[b] \Rrightarrow \mathsf{wp}\big[\mathit{body}\big](I)$

(4) $\lim\limits_{i\to\infty} \mathbf{EV}_i\big[I\cdot[b]\big] = 0$

Generalizing Expectations

Application of loop rule for unbounded expectations

We want to prove that the program terminates **in average** in two iterations, i.e.

$$\mathsf{wp}[c](n) \;=\; \underline{2}$$

$$
\begin{array}{ll}
c: & n := 0; \\
& b := \mathsf{true}; \\
& \mathtt{while}\,(b{=}\mathsf{true})\,\mathtt{do} \\
& \quad \{b := \mathsf{true}\}\ [1/2]\ \{b := \mathsf{false}\}; \\
& \quad n := n{+}1;
\end{array}
$$

$\mathsf{wp}[c](n)$

$=$ $\quad \langle$rule for seq. comp.$\rangle$

$\mathsf{wp}\big[n := 0;\ b := \mathsf{true}\big]\big(\mathsf{wp}\big[\mathtt{while}\,(b{=}\mathsf{true})\,\mathtt{do}\ldots\big](n)\big)$

$\Leftarrow$ $\quad \langle$monoton. of $\mathsf{wp}[\cdot]$; let $I := [b]\cdot(n+2) + [\neg b]\cdot n\rangle$ **(1)** $I \cdot [\neg b] \;\Rrightarrow\; n$

$\mathsf{wp}\big[n := 0;\ b := \mathsf{true}\big]\big(\mathsf{wp}\big[\mathtt{while}\ (b{=}\mathsf{true})\ \mathtt{do}\ \mathit{body}\big](I \cdot [\neg b])\big)$

$\Leftarrow$ $\quad \langle$monoton. of $\mathsf{wp}[\cdot]$, loop rule with invariant I
and set of initial terminating states $\mathcal{S}\rangle$

$\mathsf{wp}\big[n := 0;\ b := \mathsf{true}\big]\big([b]\cdot(n+2) + [\neg b]\cdot n\big)$

$=$ $\quad \langle$rule for seq. comp. and assign.$\rangle$

$\underline{2}$

Generalizing Expectations

Application of loop rule for unbounded expectations

We want to prove that the program terminates **in average** in two iterations, i.e.

$$\mathsf{wp}[c](n) \;=\; \underline{2}$$

c :
$n := 0;$
$b := \mathsf{true};$
$\mathtt{while}\,(b{=}\mathsf{true})\,\mathtt{do}$
$\quad \{b := \mathsf{true}\}\ [1/2]\ \{b := \mathsf{false}\};$
$\quad n := n{+}1;$

$\mathsf{wp}[c](n)$

$=$ $\quad \langle$rule for seq. comp.$\rangle$

$\mathsf{wp}\big[n := 0;\ b := \mathsf{true}\big]\big(\mathsf{wp}\big[\mathtt{while}\,(b{=}\mathsf{true})\,\mathtt{do}\ldots\big](n)\big)$

$\Leftarrow$ $\quad \langle$monoton. of $\mathsf{wp}[\cdot]$; let $I := [b]\cdot(n+2) + [\neg b]\cdot n\rangle$

$\mathsf{wp}\big[n := 0;\ b := \mathsf{true}\big]\big(\mathsf{wp}\big[\mathtt{while}\ (b{=}\mathsf{true})\ \mathtt{do}\ \mathit{body}\big]\big(I\cdot[\neg b]\big)\big)$

$\Leftarrow$ $\quad \langle$monoton. of $\mathsf{wp}[\cdot]$, loop rule with invariant I
and set of initial terminating states $\mathcal{S}\rangle$

$\mathsf{wp}\big[n := 0;\ b := \mathsf{true}\big]\big([b]\cdot(n+2) + [\neg b]\cdot n\big)$

$=$ $\quad \langle$rule for seq. comp. and assign.$\rangle$

$\underline{2}$

(2) $\mathsf{true} \;\Rrightarrow\; \mathsf{wp}\big[\mathtt{while}\ b\ \mathtt{do}\ \mathit{body}\big](\underline{1})$

Generalizing Expectations

Application of loop rule for unbounded expectations

We want to prove that the program terminates **in average** in two iterations, i.e.

$$\mathsf{wp}[c](n) \;=\; \underline{2}$$

c :
$n := 0;$
$b := \mathsf{true};$
$\mathtt{while}\,(b{=}\mathsf{true})\,\mathtt{do}$
$\quad \{b := \mathsf{true}\}\ [1/2]\ \{b := \mathsf{false}\};$
$\quad n := n{+}1;$

$\mathsf{wp}[c](n)$

$=$ $\quad \langle$rule for seq. comp.$\rangle$

$\mathsf{wp}\big[n := 0;\ b := \mathsf{true}\big]\big(\mathsf{wp}\big[\mathtt{while}\,(b{=}\mathsf{true})\,\mathtt{do}\ldots\big](n)\big)$

$\Leftarrow$ $\quad \langle$monoton. of $\mathsf{wp}[\cdot]$; let $I := [b]\cdot(n+2) + [\neg b]\cdot n\rangle$

$\mathsf{wp}\big[n := 0;\ b := \mathsf{true}\big]\big(\mathsf{wp}\big[\mathtt{while}\ (b{=}\mathsf{true})\ \mathtt{do}\ \mathit{body}\big]\big(I\cdot[\neg b]\big)\big)$

$\Leftarrow$ $\quad \langle$monoton. of $\mathsf{wp}[\cdot]$, loop rule with invariant I
and set of initial terminating states $\mathcal{S}\rangle$

$\mathsf{wp}\big[n := 0;\ b := \mathsf{true}\big]\big([b]\cdot(n+2) + [\neg b]\cdot n\big)$

$=$ $\quad \langle$rule for seq. comp. and assign.$\rangle$

$\underline{2}$

Variant $[b]$ decreases with probability $1/2$ in each iteration

(2) $\mathtt{true} \;\Rrightarrow\; \mathsf{wp}\big[\mathtt{while}\ b\ \mathtt{do}\ \mathit{body}\big](\underline{1})$

Generalizing Expectations

Application of loop rule for unbounded expectations

We want to prove that the program terminates **in average** in two iterations, i.e.

$$\mathsf{wp}[c](n) \;=\; \underline{2}$$

c :
$n := 0;$
$b := \mathsf{true};$
$\texttt{while}\,(b{=}\mathsf{true})\,\texttt{do}$
$\quad \{b := \mathsf{true}\}\ [1/2]\ \{b := \mathsf{false}\};$
$\quad n := n{+}1;$

$\mathsf{wp}[c](n)$

$=$ $\quad \langle$rule for seq. comp.$\rangle$

$\mathsf{wp}\big[n := 0;\ b := \mathsf{true}\big]\big(\mathsf{wp}\big[\texttt{while}\,(b{=}\mathsf{true})\,\texttt{do}\ldots\big](n)\big)$

$\Leftarrow$ $\quad \langle$monoton. of $\mathsf{wp}[\cdot]$; let $I := [b]\cdot(n+2)+[\neg b]\cdot n\rangle$

$\mathsf{wp}\big[n := 0;\ b := \mathsf{true}\big]\big(\mathsf{wp}\big[\texttt{while}\ (b{=}\mathsf{true})\ \texttt{do}\ \mathit{body}\big]\big(I\cdot[\neg b]\big)\big)$

$\Leftarrow$ $\quad \langle$monoton. of $\mathsf{wp}[\cdot]$, loop rule with invariant I
and set of initial terminating states $\mathcal{S}\rangle$

$\mathsf{wp}\big[n := 0;\ b := \mathsf{true}\big]\big([b]\cdot(n+2)+[\neg b]\cdot n\big)$

$=$ $\quad \langle$rule for seq. comp. and assign.$\rangle$

$\underline{2}$

(3) $I\cdot[b] \Rrightarrow \mathsf{wp}\big[\mathit{body}\big](I)$

Generalizing Expectations

Application of loop rule for unbounded expectations

We want to prove that the program terminates **in average** in two iterations, i.e.

$$\mathsf{wp}[c](n) \;=\; \underline{2}$$

$$\begin{array}{ll}
c: & n := 0; \\
& b := \mathsf{true}; \\
& \mathtt{while}\,(b{=}\mathsf{true})\,\mathtt{do} \\
& \quad \{b := \mathsf{true}\}\ [1/2]\ \{b := \mathsf{false}\}; \\
& \quad n := n{+}1;
\end{array}$$

$\mathsf{wp}[c](n)$

$=$ $\quad \langle$rule for seq. comp.$\rangle$

$\mathsf{wp}\big[n := 0;\ b := \mathsf{true}\big]\big(\mathsf{wp}\big[\mathtt{while}\,(b{=}\mathsf{true})\,\mathtt{do}\ldots\big](n)\big)$

$\Leftarrow$ $\quad \langle$monoton. of $\mathsf{wp}[\cdot]$; let $I := [b]\cdot(n+2) + [\neg b]\cdot n\rangle$

$\mathsf{wp}\big[n := 0;\ b := \mathsf{true}\big]\big(\mathsf{wp}\big[\mathtt{while}\ (b{=}\mathsf{true})\ \mathtt{do}\ \mathit{body}\big](I\cdot[\neg b])\big)$

$\Leftarrow$ $\quad \langle$monoton. of $\mathsf{wp}[\cdot]$, loop rule with invariant I
and set of initial terminating states $\mathcal{S}\rangle$

$\mathsf{wp}\big[n := 0;\ b := \mathsf{true}\big]\big([b]\cdot(n+2) + [\neg b]\cdot n\big)$

$=$ $\quad \langle$rule for seq. comp. and assign.$\rangle$

$\underline{2}$

(4) $\lim\limits_{i\to\infty} \mathbf{EV}_i\big[I\cdot[b]\big] = 0$

beginning of i-th itaration: $\overbrace{\Big(\big(\tfrac{1}{2}\big)^{i-1}\cdot 1 + \overline{\big(\tfrac{1}{2}\big)^{i-1}}\cdot 0\Big)}^{[b]} \cdot \underbrace{(n+2)}_{\leq i+2}$

$\lim_{i\to\infty}\big(\tfrac{1}{2}\big)^{i-1}\cdot(i{+}2) = 0$

Generalizing Expectations

Application of loop rule for unbounded expectations

We want to prove that the program terminates **in average** in two iterations, i.e.

$$\mathsf{wp}[c](n) \;=\; \underline{2}$$

$$\begin{array}{ll}
c: & n := 0; \\
& b := \mathsf{true}; \\
& \mathtt{while}\,(b{=}\mathsf{true})\,\mathtt{do} \\
& \quad \{b := \mathsf{true}\}\ [1/2]\ \{b := \mathsf{false}\}; \\
& \quad n := n{+}1;
\end{array}$$

$\mathsf{wp}[c](n)$

$=$ $\quad \langle$rule for seq. comp.$\rangle$

$\mathsf{wp}\big[n := 0;\ b := \mathsf{true}\big]\big(\mathsf{wp}\big[\mathtt{while}\,(b{=}\mathsf{true})\,\mathtt{do}\ldots\big](n)\big)$

$\Leftarrow$ $\quad \langle$monoton. of $\mathsf{wp}[\cdot]$; let $I := [b]\cdot(n+2) + [\neg b]\cdot n\rangle$

$\mathsf{wp}\big[n := 0;\ b := \mathsf{true}\big]\big(\mathsf{wp}\big[\mathtt{while}\ (b{=}\mathsf{true})\ \mathtt{do}\ \mathit{body}\big]\big(I\cdot[\neg b]\big)\big)$

$\Leftarrow$ $\quad \langle$monoton. of $\mathsf{wp}[\cdot]$, loop rule with invariant I
and set of initial terminating states $\mathcal{S}\rangle$

$\mathsf{wp}\big[n := 0;\ b := \mathsf{true}\big]\big([b]\cdot(n+2) + [\neg b]\cdot n\big)$

$=$ $\quad \langle$rule for seq. comp. and assign.$\rangle$

$\underline{2}$

Generalizing Expectations

Application of loop rule for unbounded expectations

We want to prove that the program terminates **in average** in two iterations, i.e.

$$\mathsf{wp}[c](n) \;=\; \underline{2}$$

c :
$n := 0;$
$b := \mathsf{true};$
$\mathtt{while}\,(b{=}\mathsf{true})\,\mathtt{do}$
$\quad \{b := \mathsf{true}\}\ [1/2]\ \{b := \mathsf{false}\};$
$\quad n := n{+}1;$

$\mathsf{wp}[c](n)$

$=$ $\quad \langle$rule for seq. comp.$\rangle$

$\mathsf{wp}\big[n := 0;\ b := \mathsf{true}\big]\big(\mathsf{wp}\big[\mathtt{while}\,(b{=}\mathsf{true})\,\mathtt{do}\ldots\big](n)\big)$

$\Leftarrow$ $\quad \langle$monoton. of $\mathsf{wp}[\cdot]$; let $I := [b]\cdot(n+2) + [\neg b]\cdot n\rangle$

$\mathsf{wp}\big[n := 0;\ b := \mathsf{true}\big]\big(\mathsf{wp}\big[\mathtt{while}\ (b{=}\mathsf{true})\ \mathtt{do}\ \mathit{body}\big]\big(I\cdot[\neg b]\big)\big)$

$\Leftarrow$ $\quad \langle$monoton. of $\mathsf{wp}[\cdot]$, loop rule with invariant I
and set of initial terminating states $\mathcal{S}\rangle$

$\mathsf{wp}\big[n := 0;\ b := \mathsf{true}\big]\big([b]\cdot(n+2) + [\neg b]\cdot n\big)$

$=$ $\quad \langle$rule for seq. comp. and assign.$\rangle$

$\underline{2}$

$\therefore\ \ \mathsf{wp}[c](n) \geq \underline{2}$

Generalizing Expectations

We can show that

$$\mathsf{wp}[c](n) \leq \underline{2}$$

by proving that

$$\underline{0} \;\Rrightarrow\; \mathsf{wp}[c](2{-}n)$$

since

$$\underline{0} \;\leq\; \mathsf{wp}[c](2{-}n) \;=\; \underline{2} - \mathsf{wp}[c](n)$$

$$\begin{array}{ll} c: & n := 0; \\ & b := \mathsf{true}; \\ & \mathtt{while}\,(b{=}\mathsf{true})\,\mathtt{do} \\ & \quad \{b := \mathsf{true}\}\ [1/2]\ \{b := \mathsf{false}\}; \\ & \quad n := n{+}1; \end{array}$$

$$\mathsf{wp}[c](f)(s) \;=\; \sum_{i \geq 1} \left(\tfrac{1}{2}\right)^i f(s[b, n/\mathsf{false}, i])$$

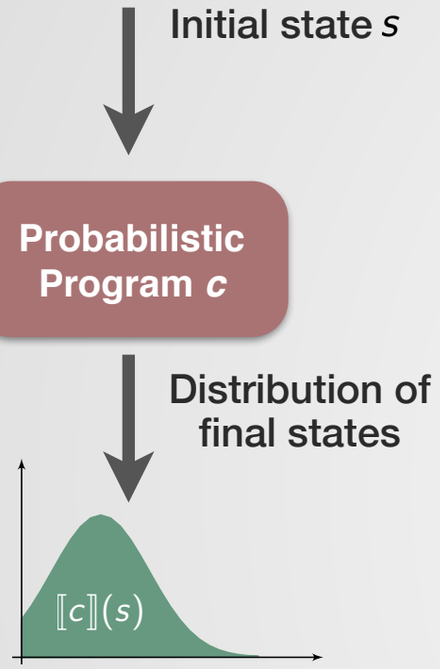
Agenda

- Recap on previous lecture
- Algebraic properties
- Extension to unbounded expectations
- Connection to relational semantics
- Extension to non-deterministic programs
- Summary

Two Semantical Views of Probabilistic Programs

Relational view

$$[\![c]\!] : \mathcal{S} \to \mathcal{D}(\mathcal{S})$$



Expectation transformer view

$$\mathrm{wp}[c] : \mathbb{E} \to \mathbb{E}$$

$$
\begin{aligned}
\mathrm{wp}[\texttt{skip}](f) &= f \\
\mathrm{wp}[\texttt{abort}](f) &= \underline{0} \\
\mathrm{wp}[x \coloneqq E](f) &= f[E/x] \\
\mathrm{wp}[\texttt{if } G \texttt{ then } c_1 \texttt{ else } c_2](f) &= [G] \cdot \mathrm{wp}[c_1](f) + [\neg G] \cdot \mathrm{wp}[c_2](f) \\
\mathrm{wp}[\{c_1\}\ [p]\ \{c_2\}](f) &= p \cdot \mathrm{wp}[c_1](f) + (1-p) \cdot \mathrm{wp}[c_2](f) \\
\mathrm{wp}[c_1; c_2](f) &= (\mathrm{wp}[c_1] \circ \mathrm{wp}[c_2])(f) \\
\mathrm{wp}[\texttt{while } G \texttt{ do } c](f) &= \mu h \bullet ([G] \cdot \mathrm{wp}[c](h) + [\neg G] \cdot f)
\end{aligned}
$$

Connection between the two views

$$\mathrm{wp}[c](f) \;=\; \lambda s \bullet \mathbf{EV}_{[\![c]\!](s)}(f)$$

Relational Semantics — Preliminaries

Distributions over program states

We let $\mathcal{D}(\mathcal{S})$ be the set of *sub-probability distributions* over program states.

missing probability
$1 - w([\![c]\!](s))$

Total mass of $[\![c]\!](s)$

→ **probability of non-termination** from initial state s

Related notation and operations

η_s	Dirac distribution $\quad (s \in \mathcal{S})$
$\mathbf{0}$	null distribution
$p \cdot \mu_1 + (1-p) \cdot \mu_2$	convex combination between distributions μ_1 and μ_2 $\quad (p \in [0,1],\ \mu_1, \mu_2 \in \mathcal{D}(\mathcal{S}))$
$\mu \blacktriangleright f$	distribution $\sum_{a \in A} \Pr[\mu{=}a] \cdot f(a)$ $(\mu \in \mathcal{D}(A),\ f : A \to \mathcal{D}(\mathcal{S}))$

Relational Semantics — Definition

For pGCL program c, the semantic function

$$[\![c]\!] : \mathcal{S} \to \mathcal{D}(\mathcal{S})$$

is defined by induction on c structure as follows:

$$
\begin{aligned}
[\![\texttt{skip}]\!] &= \lambda s \bullet \eta_s \\
[\![\texttt{abort}]\!] &= \lambda s \bullet \mathbf{0} \\
[\![x \coloneqq E]\!] &= \lambda s \bullet \eta_{s'} \quad \text{where } s' = s[E/x] \\
[\![\texttt{if } G \texttt{ then } c_1 \texttt{ else } c_2]\!] &= \lambda s \bullet [\![c_1]\!](s) \lhd [\![G]\!](s) \rhd [\![c_2]\!](s) \\
[\![\{c_1\}\ [p]\ \{c_2\}]\!] &= \lambda s \bullet p \cdot [\![c_1]\!](s) + (1-p) \cdot [\![c_2]\!](s) \\
[\![c_1; c_2]\!] &= \lambda s \bullet [\![c_1]\!](s) \blacktriangleright [\![c_2]\!] \\
[\![\texttt{while } G \texttt{ do } c]\!] &= \mathit{lfp}(F) \quad \text{where } F(f) = \lambda s \bullet ([\![c]\!](s) \blacktriangleright f) \lhd [\![G]\!](s) \rhd \eta_s
\end{aligned}
$$

$: (\mathcal{S} \to \mathcal{D}(\mathcal{S})) \to (\mathcal{S} \to \mathcal{D}(\mathcal{S}))$

Connection between Relational and Expectation Transformer Semantics

Theorem

For any (purely probabilistic) pGCL program c and post-expectation f,

$$\mathsf{wp}[c](f) \;=\; \lambda s \bullet \mathbf{EV}_{[\![c]\!](s)}(f)$$

- This theorem proves both the soundness and the completeness of the expectation transformer semantics wrt the relational semantics:

SOUNDNESS ➡ $\mathsf{wp}[c]([P])(s) = \alpha \implies \Pr\big[P \in [\![c]\!](s)\big] = \alpha$

COMPLETENESS ➡ $\Pr\big[P \in [\![c]\!](s)\big] = \alpha \implies \mathsf{wp}[c]([P])(s) = \alpha$

Connection between Relational and Expectation Transformer Semantics

Theorem

For any (purely probabilistic) pGCL program c and post-expectation f,

$$\mathsf{wp}[c](f) \;=\; \lambda s \bullet \mathbf{EV}_{[\![c]\!](s)}(f)$$

- This theorem proves both the soundness and the completeness of the expectation transformer semantics wrt the relational semantics:

SOUNDNESS ➡ $\mathsf{wp}[c]([P])(s) = \alpha \implies \Pr\big[P \in [\![c]\!](s)\big] = \alpha$

COMPLETENESS ➡ $\Pr\big[P \in [\![c]\!](s)\big] = \alpha \implies \mathsf{wp}[c]([P])(s) = \alpha$

- The connection between the two semantics can be recast in a more uniform manner:

$$g = \mathsf{wp}[c](f) \quad \text{iff} \quad \forall \mu \in \mathcal{D}(\mathcal{S}) \bullet \; \mathbf{EV}_{\mu}(g) \;=\; \mathbf{EV}_{\mu \blacktriangleright [\![c]\!]}(f)$$

Connection between Relational and Expectation Transformer Semantics

Theorem

For any (purely probabilistic) pGCL program c and post-expectation f,

$$\mathsf{wp}[c](f) \;=\; \lambda s \bullet \mathbf{EV}_{[\![c]\!](s)}(f)$$

- This theorem proves both the soundness and the completeness of the expectation transformer semantics wrt the relational semantics:

SOUNDNESS ➡ $\mathsf{wp}[c]([P])(s) = \alpha \implies \Pr\big[P \in [\![c]\!](s)\big] = \alpha$

COMPLETENESS ➡ $\Pr\big[P \in [\![c]\!](s)\big] = \alpha \implies \mathsf{wp}[c]([P])(s) = \alpha$

- The connection between the two semantics can be recast in a more uniform manner:

$$g = \mathsf{wp}[c](f) \quad \text{iff} \quad \forall \mu \in \mathcal{D}(\mathcal{S}) \bullet \mathbf{EV}_{\mu}(g) \;=\; \mathbf{EV}_{\mu \blacktriangleright [\![c]\!]}(f)$$

$$P = \mathsf{wp}[c](Q) \quad \text{iff} \quad \forall s \in \mathcal{S} \;\bullet\; P(s) \;=\; Q([\![c]\!](s)) \quad \left(\begin{smallmatrix}\text{deterministic}\\ \text{counterpart}\end{smallmatrix}\right)$$

Agenda

- Recap on previous lecture
- Algebraic properties
- Extension to unbounded expectations
- Connection to relational semantics
- Extension to non-deterministic programs
- Summary

Non-deterministic Programs

Syntax

The pGCL language

$\mathcal{C}$	:=	`skip`	nop
	|	`abort`	abortion
	|	$x := E$	assignment
	|	`if` G `then` $\mathcal{C}$ `else` $\mathcal{C}$	conditional
	|	$\{\mathcal{C}\}\ [p]\ \{\mathcal{C}\}$	probabilistic choice
	|	$\{\mathcal{C}\} \Box \{\mathcal{C}\}$	**non-deterministic choice**
	|	`while` G `do` $\mathcal{C}$	while loop
	|	$\mathcal{C}; \mathcal{C}$	sequence

Semantics

- Non-determinism is resolved by means of a scheduler who decides, on each occurrence of a non-deterministic choice, which branch (left or right) to execute.

Scheduler		Induced Prob.
$\{\mathcal{L}, \mathcal{L}\}$	$\rightsquigarrow$	Pr[A]=0.20
$\{\mathcal{L}, \mathcal{R}\}$	$\rightsquigarrow$	Pr[A]=0.80
$\{\mathcal{R}, \mathcal{L}\}$	$\rightsquigarrow$	Pr[A]=0.05
$\{\mathcal{R}, \mathcal{R}\}$	$\rightsquigarrow$	Pr[A]=0.60

- **Demonic model:** we adopt the scheduler that minimizes the probability of the event at stake (scheduler varies according to the post-expectation and initial state)

Non-deterministic Programs

Examples

- $\{x := \mathsf{true}\} \,\square\, \{x := \mathsf{false}\};$
  $\{y := \mathsf{true}\}\ [1/3]\ \{y := \mathsf{false}\};$

$$\begin{aligned}\Pr[x{=}y] &= \min\left\{\Pr^{\mathcal{L}}[x{=}y], \Pr^{\mathcal{R}}[x{=}y]\right\} \\ &= \min\left\{\tfrac{1}{3}, \tfrac{2}{3}\right\} = \tfrac{1}{3}\end{aligned}$$

- $b := \mathsf{true};$
  $\mathtt{while}\,(b{=}\mathsf{true})\,\mathtt{do}$
  $\quad \{b := \mathsf{true}\}\ [1/2]\ \{b := \mathsf{false}\};$

$$\Pr[\mathsf{true}] \;=\; 1$$

Non-deterministic Programs

Examples

- $\{x := \mathsf{true}\} \,\square\, \{x := \mathsf{false}\};$
  $\{y := \mathsf{true}\}\ [1/3]\ \{y := \mathsf{false}\};$

$$\begin{aligned} \Pr[x{=}y] &= \min\left\{\Pr^{\mathcal{L}}[x{=}y], \Pr^{\mathcal{R}}[x{=}y]\right\} \\ &= \min\left\{\tfrac{1}{3}, \tfrac{2}{3}\right\} = \tfrac{1}{3} \end{aligned}$$

- $b := \mathsf{true};$
  $\mathtt{while}\,(b{=}\mathsf{true})\,\mathtt{do}$
  $\quad \{b := \mathsf{true}\}\ [1/2]\ \{b := \mathsf{false}\};$
  $\quad \{\mathtt{skip}\} \,\square\, \{b := \neg b\}$

$$\Pr[\mathsf{true}] \;=$$

Non-deterministic Programs

Examples

- $\{x := \mathsf{true}\} \,\square\, \{x := \mathsf{false}\};$
  $\{y := \mathsf{true}\}\ [1/3]\ \{y := \mathsf{false}\};$

$$\begin{aligned} \Pr[x{=}y] &= \min\left\{\Pr^{\mathcal{L}}[x{=}y], \Pr^{\mathcal{R}}[x{=}y]\right\} \\ &= \min\left\{\tfrac{1}{3}, \tfrac{2}{3}\right\} = \tfrac{1}{3} \end{aligned}$$

- $b := \mathsf{true};$
  $\mathtt{while}\,(b{=}\mathsf{true})\,\mathtt{do}$
  $\quad \{b := \mathsf{true}\}\ [1/2]\ \{b := \mathsf{false}\};$
  $\quad \{\mathtt{skip}\} \,\square\, \{b := \neg b\}$

$$\Pr[\mathsf{true}] \;=\; 0$$

Non-deterministic Programs

Examples

- $\{x := \text{true}\} \,\square\, \{x := \text{false}\};$
  $\{y := \text{true}\}\ [1/3]\ \{y := \text{false}\};$

$$\begin{aligned} \Pr[x{=}y] &= \min\left\{\Pr^{\mathcal{L}}[x{=}y], \Pr^{\mathcal{R}}[x{=}y]\right\} \\ &= \min\left\{\tfrac{1}{3}, \tfrac{2}{3}\right\} = \tfrac{1}{3} \end{aligned}$$

- $b := \text{true};$
  $\texttt{while}\,(b{=}\text{true})\,\texttt{do}$
  $\quad \{b := \text{true}\}\ [1/2]\ \{b := \text{false}\};$
  $\quad \{\texttt{skip}\} \,\square\, \{b := \neg b\}$

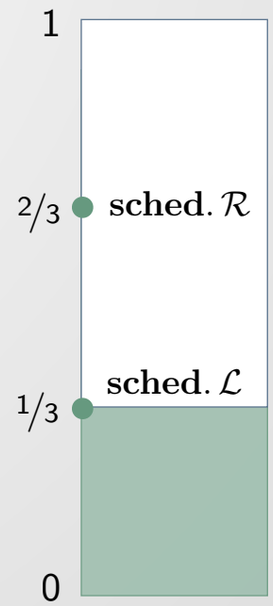
$$\Pr[\text{true}] = 0$$

Extension of expectation transformers

$$\text{wp}\left[\{c_1\} \,\square\, \{c_2\}\right](f) = \min\left\{\text{wp}[c_1](f), \text{wp}[c_2](f)\right\}$$

$\text{wp}\left[c\right](f)(s)$

- *c* purely prob. → expected value of *f* from state *s*
- *c* non-det. → greatest lower bound for the expected value of *f* from state *s*



Non-deterministic Programs

Examples

- $\{x := \text{true}\} \,\square\, \{x := \text{false}\};$
  $\{y := \text{true}\}\ [1/3]\ \{y := \text{false}\};$

$$\begin{aligned} \Pr[x{=}y] &= \min\left\{\Pr^{\mathcal{L}}[x{=}y], \Pr^{\mathcal{R}}[x{=}y]\right\} \\ &= \min\left\{\tfrac{1}{3}, \tfrac{2}{3}\right\} = \tfrac{1}{3} \end{aligned}$$

- $b := \text{true};$
  $\texttt{while}\,(b{=}\text{true})\,\texttt{do}$
  $\quad \{b := \text{true}\}\ [1/2]\ \{b := \text{false}\};$
  $\quad \{\texttt{skip}\} \,\square\, \{b := \neg b\}$

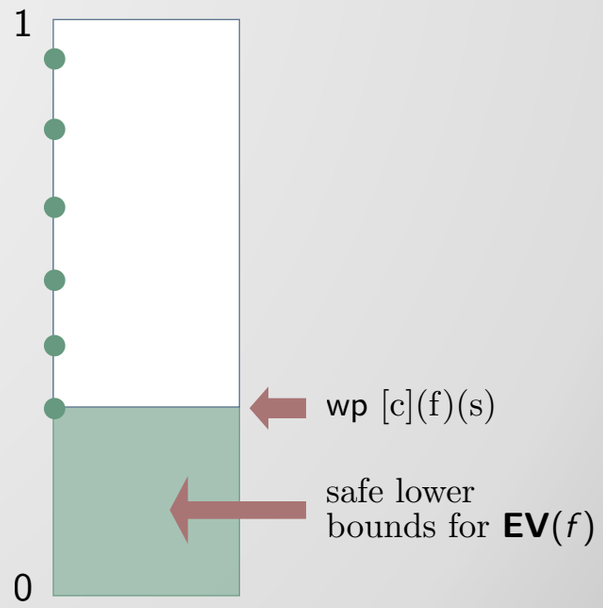
$$\Pr[\text{true}] = 0$$

Extension of expectation transformers

$$\text{wp}\left[\{c_1\} \,\square\, \{c_2\}\right](f) = \min\left\{\text{wp}[c_1](f), \text{wp}[c_2](f)\right\}$$

$\text{wp}\left[c\right](f)(s)$

- *c* purely prob. → expected value of *f* from state *s*
- *c* non-det. → greatest lower bound for the expected value of *f* from state *s*



Interaction between Non-deterministic and Probabilistic Choice

c_1 : $\{x := \mathsf{t}\} \,\square\, \{x := \mathsf{f}\};$
$\{y := \mathsf{t}\}\ [1/3]\ \{y := \mathsf{f}\}$

$$
\begin{aligned}
\mathsf{wp}[x := E](f) &= f[E/x] \\
\mathsf{wp}[\{c_1\}\ [p]\ \{c_2\}](f) &= p \cdot \mathsf{wp}[c_1](f) + \bar{p} \cdot \mathsf{wp}[c_2](f) \\
\mathsf{wp}[\{c_1\} \,\square\, \{c_2\}](f) &= \min\{\mathsf{wp}[c_1](f), \mathsf{wp}[c_2](f)\} \\
\mathsf{wp}[c_1; c_2](f) &= \mathsf{wp}[c_1](\mathsf{wp}[c_2](f))
\end{aligned}
$$

$$
\begin{aligned}
& \mathsf{wp}\big[c_1\big]([x{=}y]) \\
= \quad & \langle \text{rule for seq. comp.} \rangle \\
& \mathsf{wp}\big[\{x{:=}\mathsf{t}\} \,\square\, \{x{:=}\mathsf{f}\}\big]\big(\mathsf{wp}\big[\{y{:=}\mathsf{t}\}\ [1/3]\ \{y{:=}\mathsf{f}\}\big]([x{=}y])\big) \\
= \quad & \langle \text{rule for prob. choice and assgn.} \rangle \\
& \mathsf{wp}\big[\{x{:=}\mathsf{t}\} \,\square\, \{x{:=}\mathsf{f}\}\big]\big(\tfrac{1}{3} \cdot [x{=}\mathsf{t}] + \tfrac{2}{3} \cdot [x{=}\mathsf{f}]\big) \\
= \quad & \langle \text{rule for non-det. choice and assgn.} \rangle \\
& \min\big\{\tfrac{1}{3} \cdot [\mathsf{t}{=}\mathsf{t}] + \tfrac{2}{3} \cdot [\mathsf{t}{=}\mathsf{f}], \tfrac{1}{3} \cdot [\mathsf{f}{=}\mathsf{t}] + \tfrac{2}{3} \cdot [\mathsf{f}{=}\mathsf{f}]\big\} \\
= \quad & \langle \text{simplification} \rangle \\
& \underline{\tfrac{1}{3}}
\end{aligned}
$$

Interaction between Non-deterministic and Probabilistic Choice

c_1 : $\{x := \mathsf{t}\} \,\square\, \{x := \mathsf{f}\};$
$\{y := \mathsf{t}\}\ [1/3]\ \{y := \mathsf{f}\}$

$$
\begin{aligned}
\mathsf{wp}[x := E](f) &= f[E/x] \\
\mathsf{wp}[\{c_1\}\ [p]\ \{c_2\}](f) &= p \cdot \mathsf{wp}[c_1](f) + \bar{p} \cdot \mathsf{wp}[c_2](f) \\
\mathsf{wp}[\{c_1\} \,\square\, \{c_2\}](f) &= \min\{\mathsf{wp}[c_1](f), \mathsf{wp}[c_2](f)\} \\
\mathsf{wp}[c_1; c_2](f) &= \mathsf{wp}[c_1](\mathsf{wp}[c_2](f))
\end{aligned}
$$

c_2 : $\{y := \mathsf{t}\}\ [1/3]\ \{y := \mathsf{f}\};$
$\{x := \mathsf{t}\} \,\square\, \{x := \mathsf{f}\}$

$$
\begin{aligned}
& \mathsf{wp}\big[c_1\big]([x{=}y]) \\
= \quad & \langle \text{rule for seq. comp.} \rangle \\
& \mathsf{wp}\big[\{x{:=}\mathsf{t}\} \,\square\, \{x{:=}\mathsf{f}\}\big]\big(\mathsf{wp}\big[\{y{:=}\mathsf{t}\}\ [1/3]\ \{y{:=}\mathsf{f}\}\big]([x{=}y])\big) \\
= \quad & \langle \text{rule for prob. choice and assgn.} \rangle \\
& \mathsf{wp}\big[\{x{:=}\mathsf{t}\} \,\square\, \{x{:=}\mathsf{f}\}\big]\big(\tfrac{1}{3} \cdot [x{=}\mathsf{t}] + \tfrac{2}{3} \cdot [x{=}\mathsf{f}]\big) \\
= \quad & \langle \text{rule for non-det. choice and assgn.} \rangle \\
& \min\big\{\tfrac{1}{3} \cdot [\mathsf{t}{=}\mathsf{t}] + \tfrac{2}{3} \cdot [\mathsf{t}{=}\mathsf{f}], \tfrac{1}{3} \cdot [\mathsf{f}{=}\mathsf{t}] + \tfrac{2}{3} \cdot [\mathsf{f}{=}\mathsf{f}]\big\} \\
= \quad & \langle \text{simplification} \rangle \\
& \underline{\tfrac{1}{3}}
\end{aligned}
$$

Interaction between Non-deterministic and Probabilistic Choice

c_1 : $\{x := \mathsf{t}\} \,\square\, \{x := \mathsf{f}\};$
$\{y := \mathsf{t}\} \;[1/3]\; \{y := \mathsf{f}\}$

$$
\begin{aligned}
\mathsf{wp}[x := E](f) &= f[E/x] \\
\mathsf{wp}[\{c_1\} \;[p]\; \{c_2\}](f) &= p \cdot \mathsf{wp}[c_1](f) + \bar{p} \cdot \mathsf{wp}[c_2](f) \\
\mathsf{wp}[\{c_1\} \,\square\, \{c_2\}](f) &= \min\{\mathsf{wp}[c_1](f), \mathsf{wp}[c_2](f)\} \\
\mathsf{wp}[c_1; c_2](f) &= \mathsf{wp}[c_1](\mathsf{wp}[c_2](f))
\end{aligned}
$$

c_2 : $\{y := \mathsf{t}\} \;[1/3]\; \{y := \mathsf{f}\};$
$\{x := \mathsf{t}\} \,\square\, \{x := \mathsf{f}\}$

$$
\begin{aligned}
& \mathsf{wp}\big[c_1\big]([x{=}y]) \\
= \quad & \langle \text{rule for seq. comp.} \rangle \\
& \mathsf{wp}\big[\{x{:=}\mathsf{t}\} \,\square\, \{x{:=}\mathsf{f}\}\big]\big(\mathsf{wp}\big[\{y{:=}\mathsf{t}\} \;[1/3]\; \{y{:=}\mathsf{f}\}\big]([x{=}y])\big) \\
= \quad & \langle \text{rule for prob. choice and assgn.} \rangle \\
& \mathsf{wp}\big[\{x{:=}\mathsf{t}\} \,\square\, \{x{:=}\mathsf{f}\}\big]\big(\tfrac{1}{3} \cdot [x{=}\mathsf{t}] + \tfrac{2}{3} \cdot [x{=}\mathsf{f}]\big) \\
= \quad & \langle \text{rule for non-det. choice and assgn.} \rangle \\
& \min\big\{\tfrac{1}{3} \cdot [\mathsf{t}{=}\mathsf{t}] + \tfrac{2}{3} \cdot [\mathsf{t}{=}\mathsf{f}], \tfrac{1}{3} \cdot [\mathsf{f}{=}\mathsf{t}] + \tfrac{2}{3} \cdot [\mathsf{f}{=}\mathsf{f}]\big\} \\
= \quad & \langle \text{simplification} \rangle \\
& \underline{\tfrac{1}{3}}
\end{aligned}
$$

$$
\begin{aligned}
& \mathsf{wp}\big[c_2\big]([x{=}y]) \\
= \quad & \langle \text{rule for seq. comp.} \rangle \\
& \mathsf{wp}\big[\{y{:=}\mathsf{t}\} \;[1/3]\; \{y{:=}\mathsf{f}\}\big]\big(\mathsf{wp}\big[\{x{:=}\mathsf{t}\} \,\square\, \{x{:=}\mathsf{f}\}\big]([x{=}y])\big) \\
= \quad & \langle \text{rule for non-det. choice and assgn.} \rangle \\
& \mathsf{wp}\big[\{y{:=}\mathsf{t}\} \;[1/3]\; \{y{:=}\mathsf{f}\}\big]\big(\min\big\{[\mathsf{t}{=}y], [\mathsf{f}{=}y]\big\}\big) \\
= \quad & \langle \text{rule for prob. choice and assgn.} \rangle \\
& \tfrac{1}{3} \cdot \min\big\{[\mathsf{t}{=}\mathsf{t}], [\mathsf{f}{=}\mathsf{t}]\big\} + \tfrac{2}{3} \cdot \min\big\{[\mathsf{t}{=}\mathsf{f}], [\mathsf{f}{=}\mathsf{f}]\big\} \\
= \quad & \langle \text{simplification} \rangle \\
& \underline{0}
\end{aligned}
$$

- A scheduler can “look” the past, but not the future.
- In c_2 the scheduler has access to the value of y, and can then assign x the opposite value, yielding a zero probability that $x{=}y$.
- In c_1 the scheduler must choose a single value for x, which will be the same for both branches of the probabilistic choice assigning the value to y.

Algebraic Properties of Non-deterministic Programs

The *Law of Total Probabilities* holds for purely probabilistic programs, ie

$$\mathsf{wp}[c]([P]) + \mathsf{wp}[c]([\neg P]) \;=\; \mathsf{wp}[c](\underline{1})$$

It follows from the linearity of wp[c], ie

$$\alpha\cdot\mathsf{wp}[c](f) + \beta\cdot\mathsf{wp}[c](g) \;=\; \mathsf{wp}[c](\alpha\cdot f + \beta\cdot g)$$

For non-deterministic programs we only have

$$\mathsf{wp}[c]([P]) + \mathsf{wp}[c]([\neg P]) \;\leq\; \mathsf{wp}[c](\underline{1})$$

since only sub-linearity of wp[c] holds, ie

$$\alpha\cdot\mathsf{wp}[c](f) + \beta\cdot\mathsf{wp}[c](g) \;\leq\; \mathsf{wp}[c](\alpha\cdot f + \beta\cdot g)$$

c_2 : $\{y := \mathsf{t}\}\ [1/3]\ \{y := \mathsf{f}\};$
$\{x := \mathsf{t}\} \,\square\, \{x := \mathsf{f}\}$

$\mathsf{wp}\big[c_2\big]([x{=}y]) = 0$

$\mathsf{wp}\big[c_2\big]([x{\neq}y]) = 0$

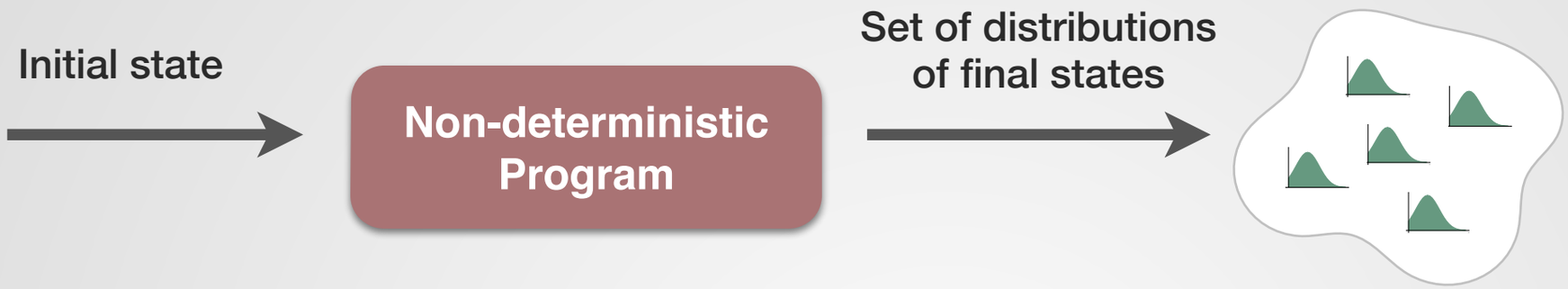
Connection between the two Semantics for Non-deterministic Programs

Theorem

For any (purely probabilistic) pGCL program c and post-expectation f,

$$\mathrm{wp}[c](f) \;=\; \lambda s \bullet \mathbf{EV}_{[\![c]\!](s)}(f)$$

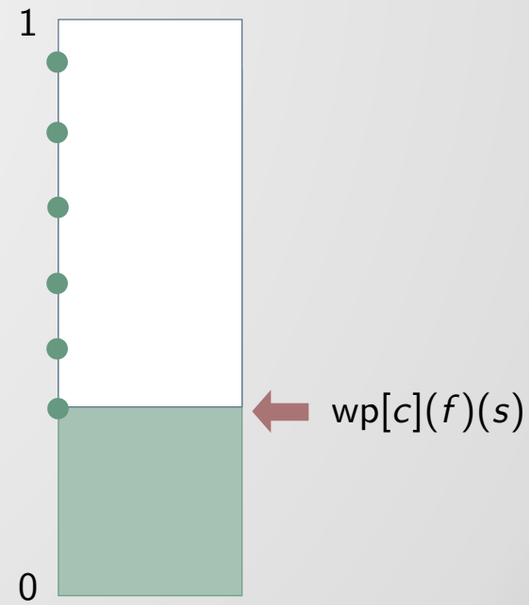
Connection between the two Semantics for Non-deterministic Programs



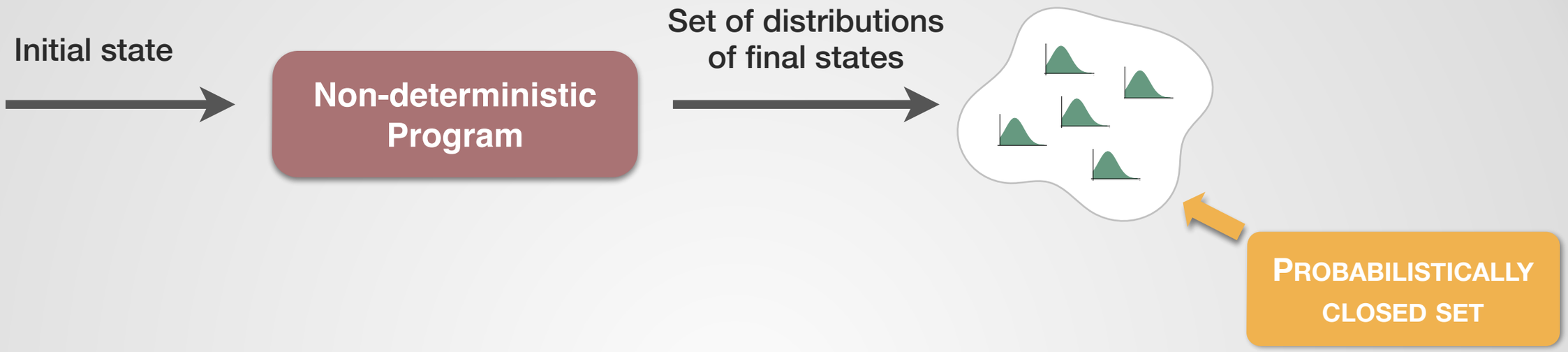
Theorem

For any (possibly non-deterministic) pGCL program c and post-expectation f,

$$\mathrm{wp}[c](f) \;=\; \lambda s \bullet\; \inf \left\{ \mathbf{EV}_{\mu'}(f) \mid \mu' \in [\![c]\!](s) \right\}$$



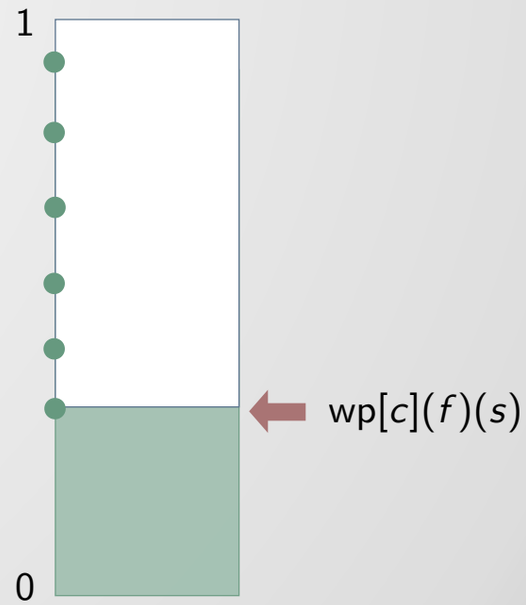
Connection between the two Semantics for Non-deterministic Programs



Theorem

For any (possibly non-deterministic) pGCL program c and post-expectation f,

$$\mathrm{wp}[c](f) \;=\; \lambda s \bullet\; \inf \left\{ \mathbf{EV}_{\mu'}(f) \mid \mu' \in [\![c]\!](s) \right\}$$



Probabilistic Powerdomain

Probabilistically closed subsets

A set of distributions $P \subseteq \mathcal{D}(\mathcal{S})$ is **probabilistically closed** iff it is

- up-closed: $\mu \in P \implies \mu' \in P \quad \text{for all } \mu' \geq \mu$
- non-empty
- convex: $\mu, \mu' \in P \implies \lambda \cdot \mu + (1-\lambda) \cdot \mu' \in P \quad \text{for all } \lambda \in [0,1]$
- Cauchy-closed: $\text{closed in the topological space } \mathbb{R}^{\mathcal{S}}_{\geq 0}$

We let $\mathbb{C}(\mathcal{S})$ be the family of all probabilistically closed subsets.

$(\mathbb{C}(\mathcal{S}), \sqsubseteq, \bigsqcup)$ defines a a complete partial order with bottom element, being

$$\theta_1 \sqsubseteq \theta_2 \triangleq \theta_1 \supseteq \theta_2 \qquad \bigsqcup_{i \in \mathbb{N}} \theta_i \triangleq \bigcap_{i \in \mathbb{N}} \theta_i \qquad \bot \triangleq \mathcal{D}(\mathcal{S})$$

* $\mu \leq \mu' \triangleq \forall s \in \mathcal{S} \bullet \Pr[\mu = s] \leq \Pr[\mu' = s]$

Relational Semantics of Non-deterministic Programs — Definition

The relational semantics of a possibly non-deterministic pGCL program c, is given by function

$$[\![c]\!] : \mathcal{S} \to \mathbb{C}(\mathcal{S})$$

defined by induction on c structure as follows:

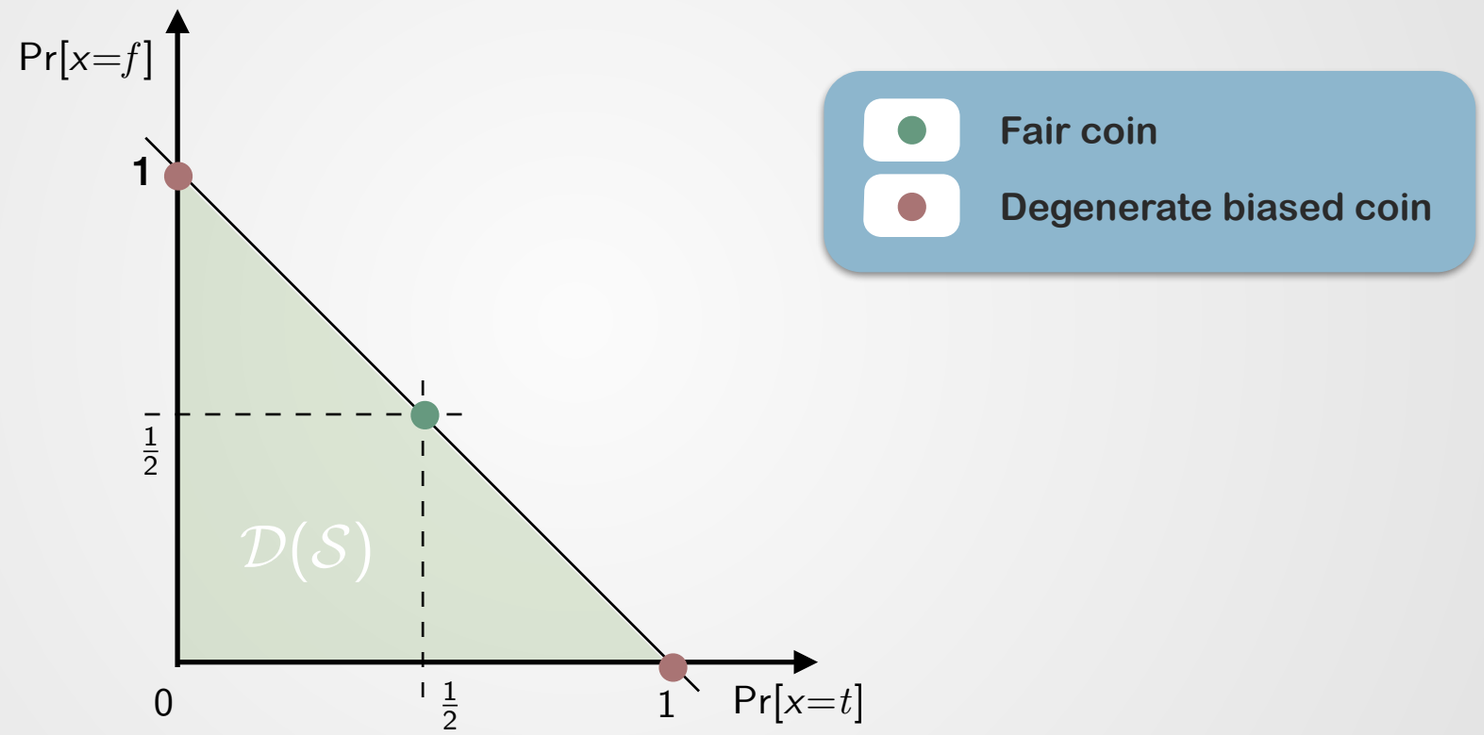
$$
\begin{aligned}
[\![\texttt{skip}]\!] &= \lambda s \bullet \{\eta_s\} \\
[\![\texttt{abort}]\!] &= \lambda s \bullet \mathcal{D}(\mathcal{S}) \\
[\![x \coloneqq E]\!] &= \lambda s \bullet \{\eta_{s'}\} \quad \text{where } s' = s[E/x] \\
[\![\texttt{if } G \texttt{ then } c_1 \texttt{ else } c_2]\!] &= \lambda s \bullet [\![c_1]\!](s) \lhd [\![G]\!](s) \rhd [\![c_2]\!](s) \\
[\![\{c_1\}\ [p]\ \{c_2\}]\!] &= \lambda s \bullet [\![c_1]\!](s) \oplus_p [\![c_2]\!](s) \\
[\![\{c_1\} \Box \{c_2\}]\!] &= \lambda s \bullet \bigcup_{p \in [0,1]} [\![c_1]\!](s) \oplus_p [\![c_2]\!](s) \\
[\![c_1; c_2]\!] &= \lambda s \bullet \{\mu \blacktriangleright f \mid \mu \in [\![c_1]\!](s) \text{ and } f \in \mathit{det\_ref}([\![c_2]\!])\} \\
[\![\texttt{while } G \texttt{ do } c]\!] &= \mathit{lfp}(F) \quad \text{where} \\
F(f) &= \lambda s \bullet \{\mu \blacktriangleright h \mid \mu \in [\![c]\!](s) \text{ and } h \in \mathit{det\_ref}(f)\} \lhd [\![G]\!](s) \rhd \{\eta_s\}
\end{aligned}
$$

$: (\mathcal{S} \to \mathbb{C}(\mathcal{S})) \to (\mathcal{S} \to \mathbb{C}(\mathcal{S}))$

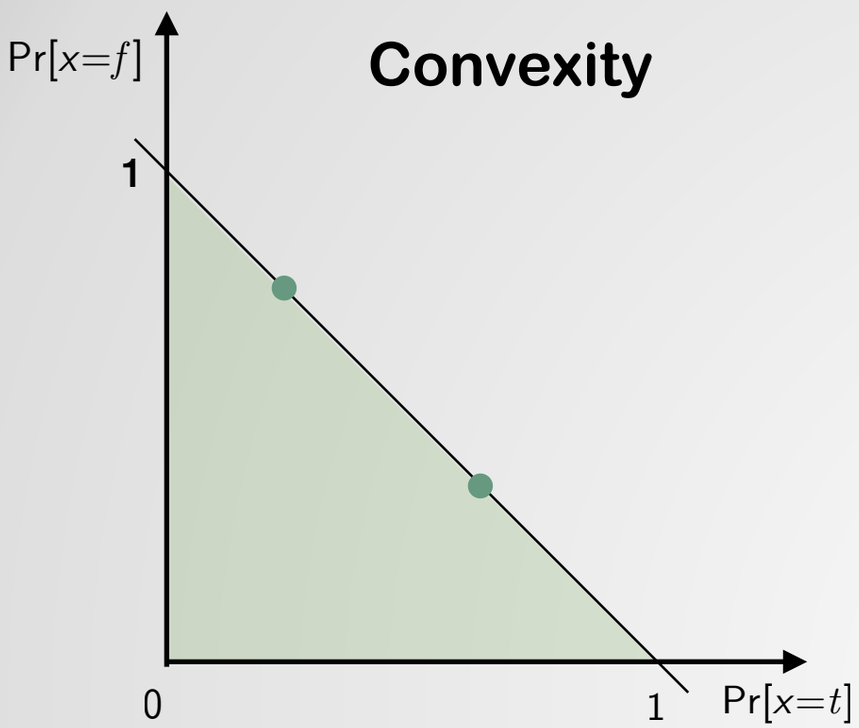
* $\theta_1 \oplus_p \theta_2 \triangleq \{p \cdot \mu_1 + (1-p) \cdot \mu_2 \mid \mu_1 \in \theta_1 \wedge \mu_2 \in \theta_2\}$

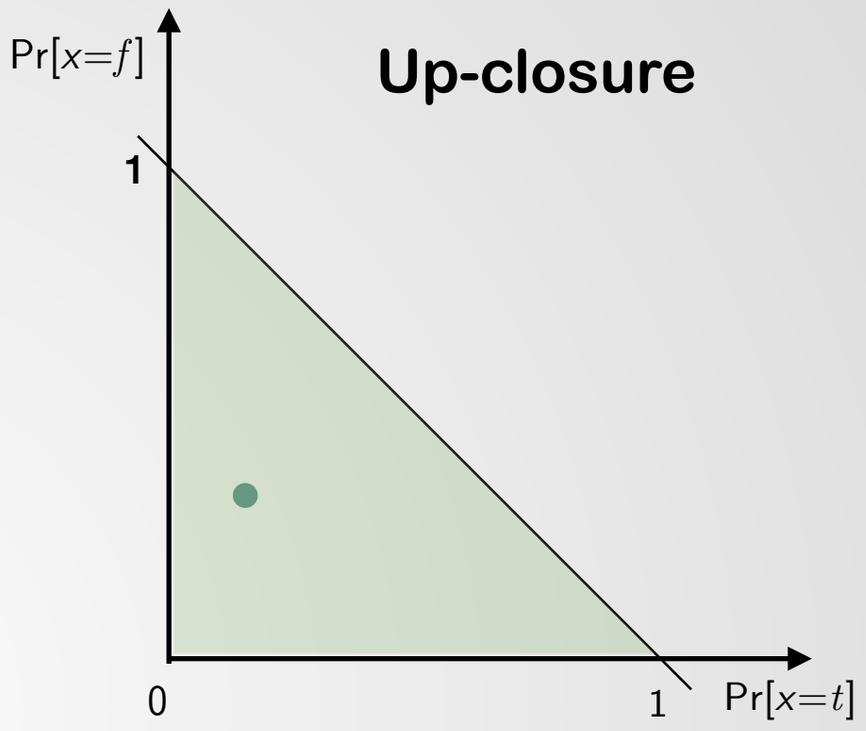
Relational Semantics of Non-det. Programs — Geometric Interpretation

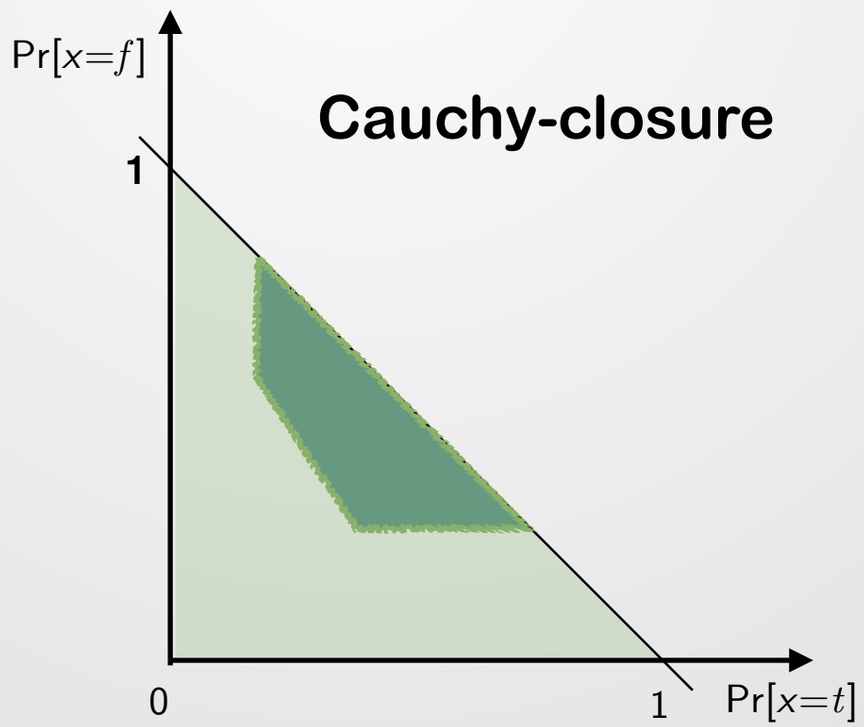
Assume we have only one program variable x, which is Boolean. Then the set of distributions over program states $\mathcal{D}(\mathcal{S})$ can be represented in the cartesian plane.



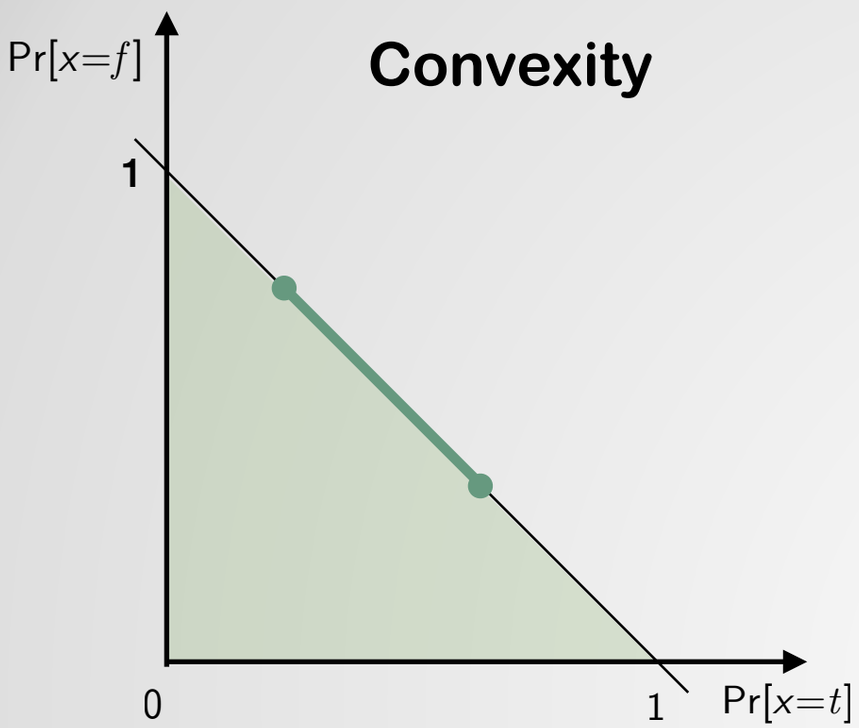
Relational Semantics of Non-det. Programs — Geometric Interpretation

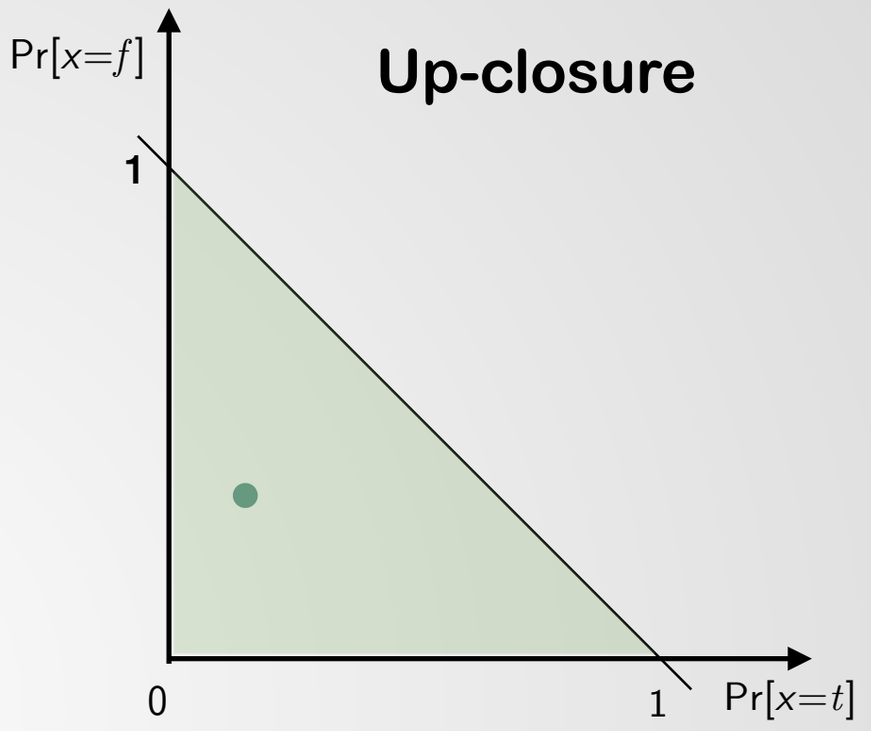


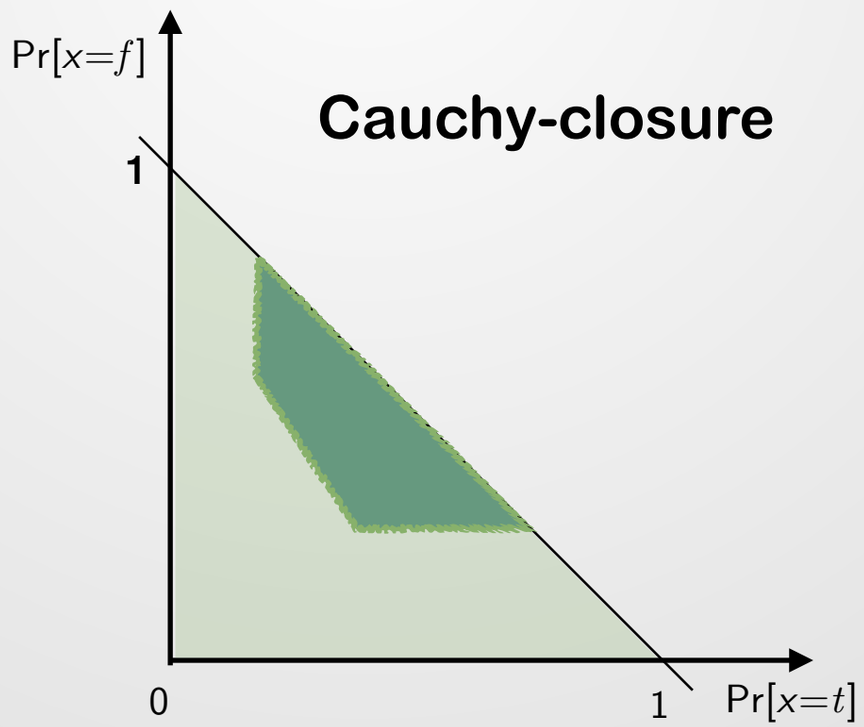




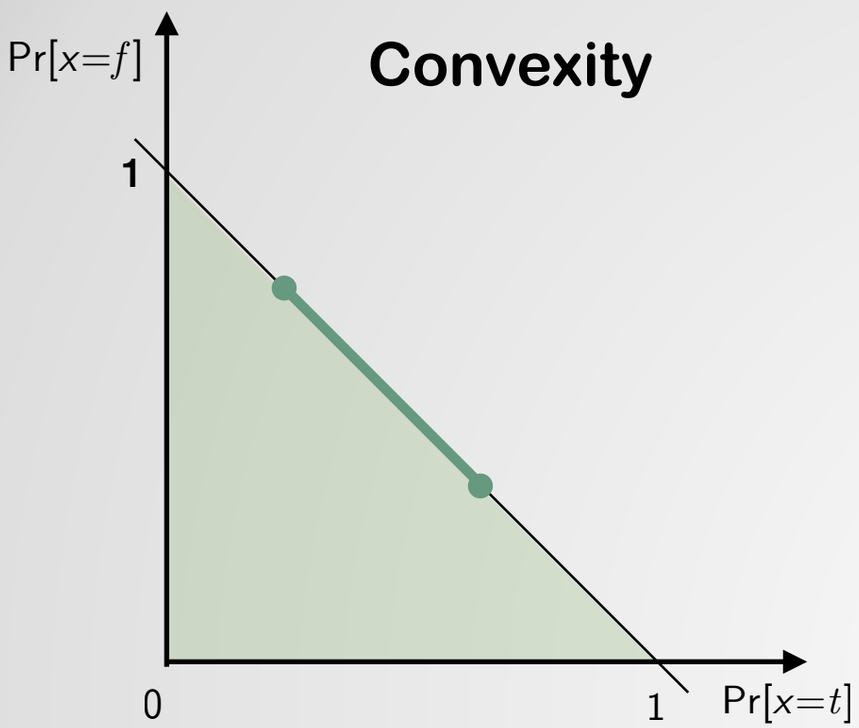
Relational Semantics of Non-det. Programs — Geometric Interpretation

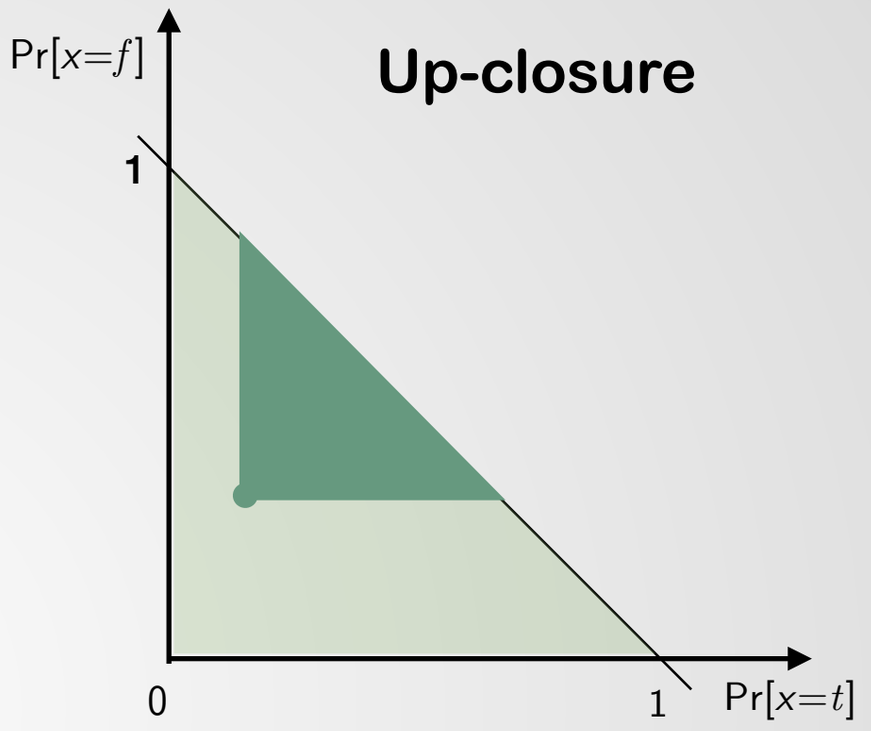


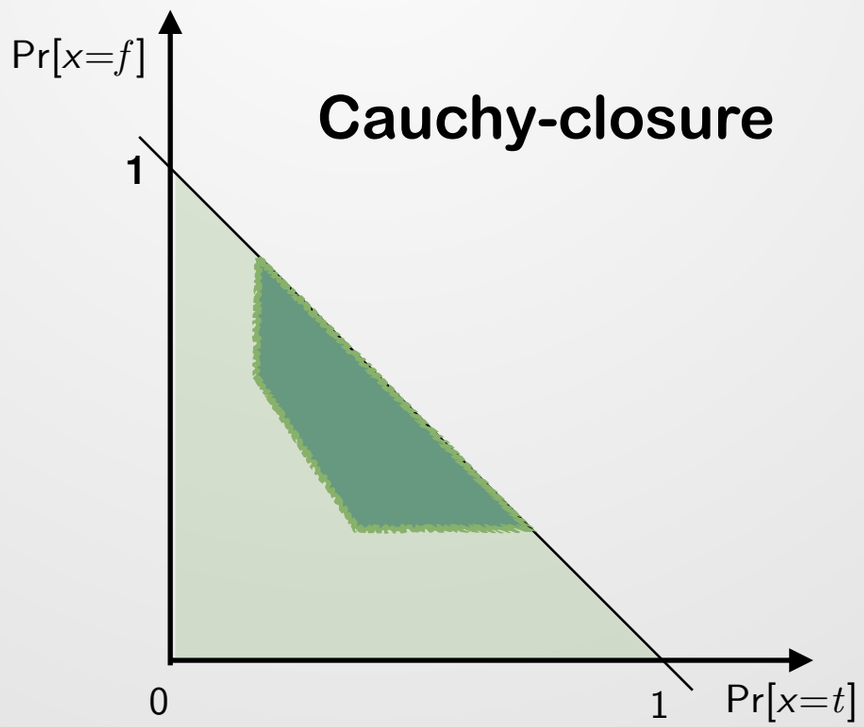




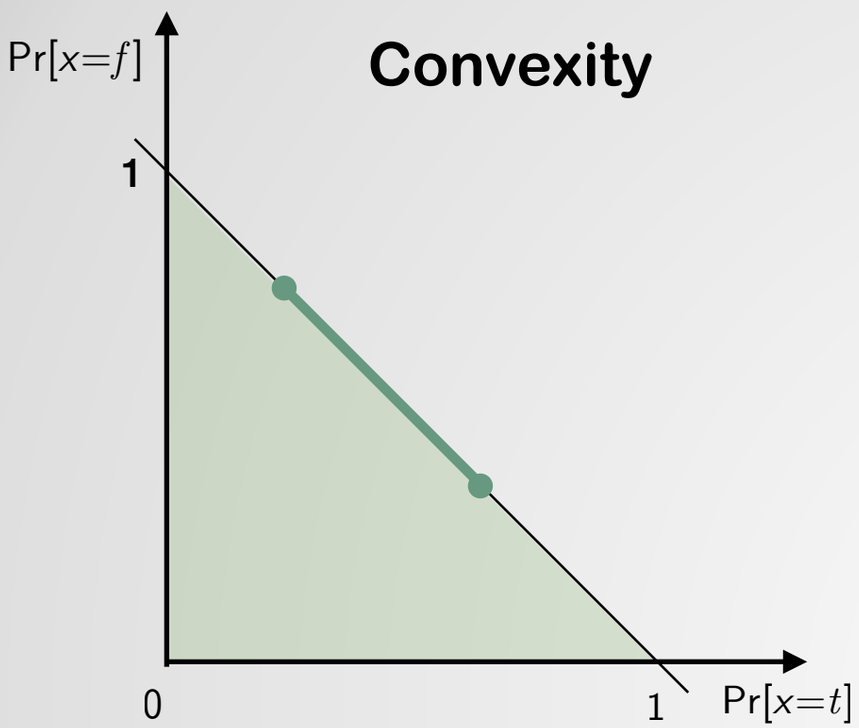
Relational Semantics of Non-det. Programs — Geometric Interpretation

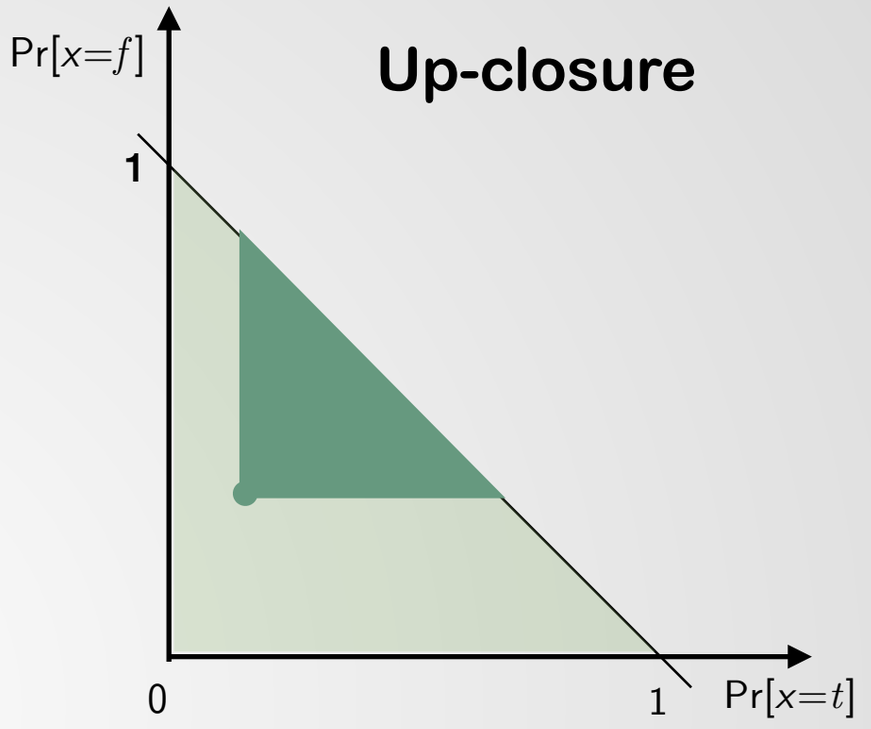


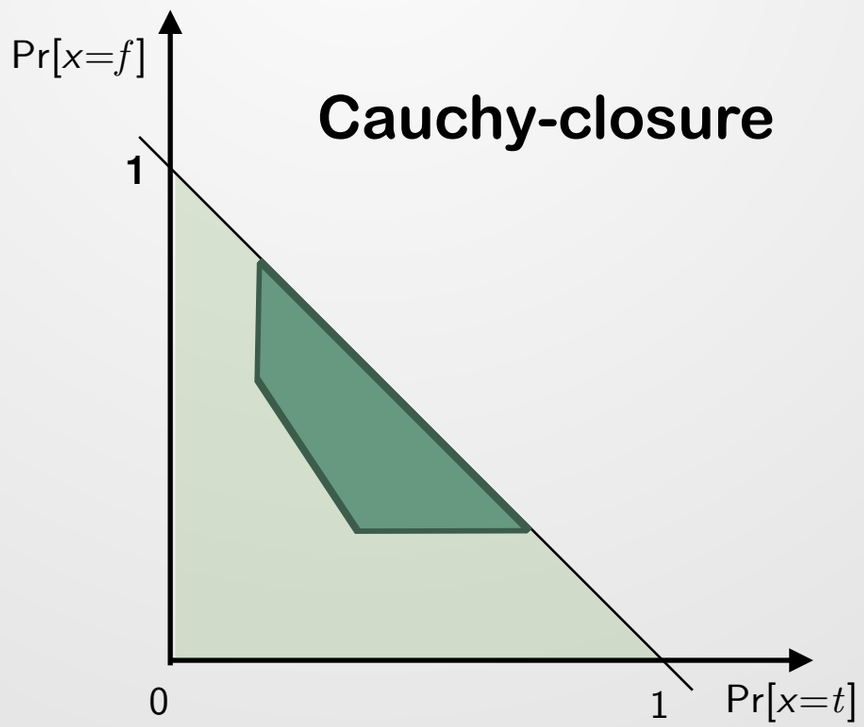




Relational Semantics of Non-det. Programs — Geometric Interpretation

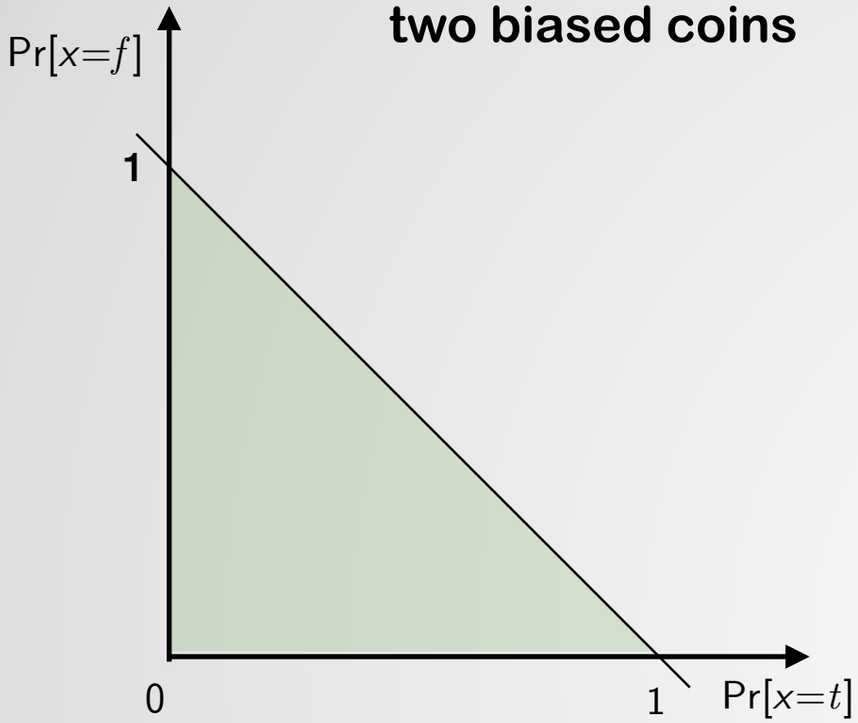


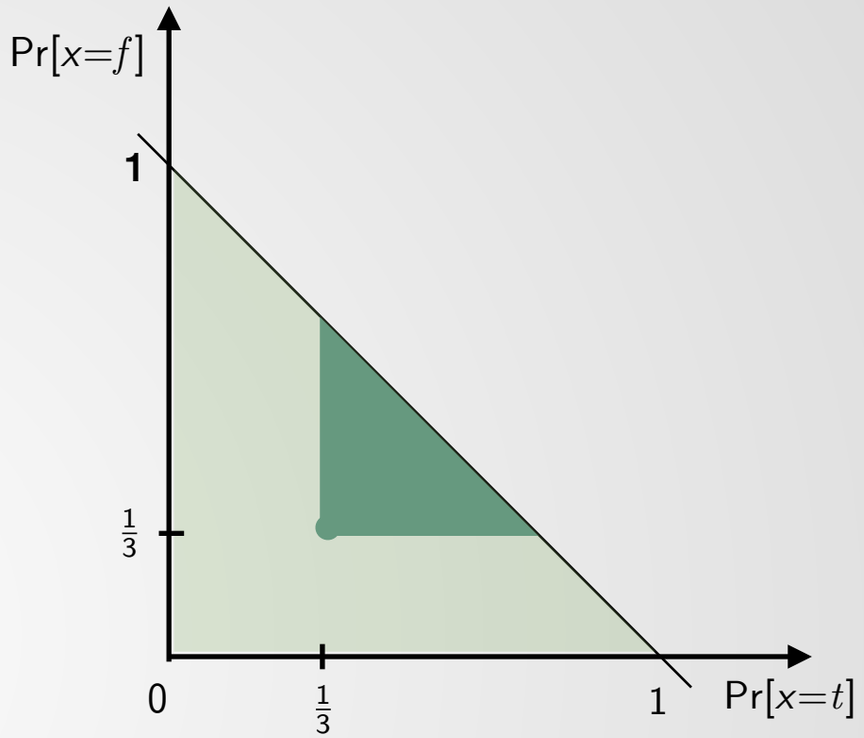




Relational Semantics of Non-det. Programs — Geometric Interpretation

Demonic choice between two biased coins



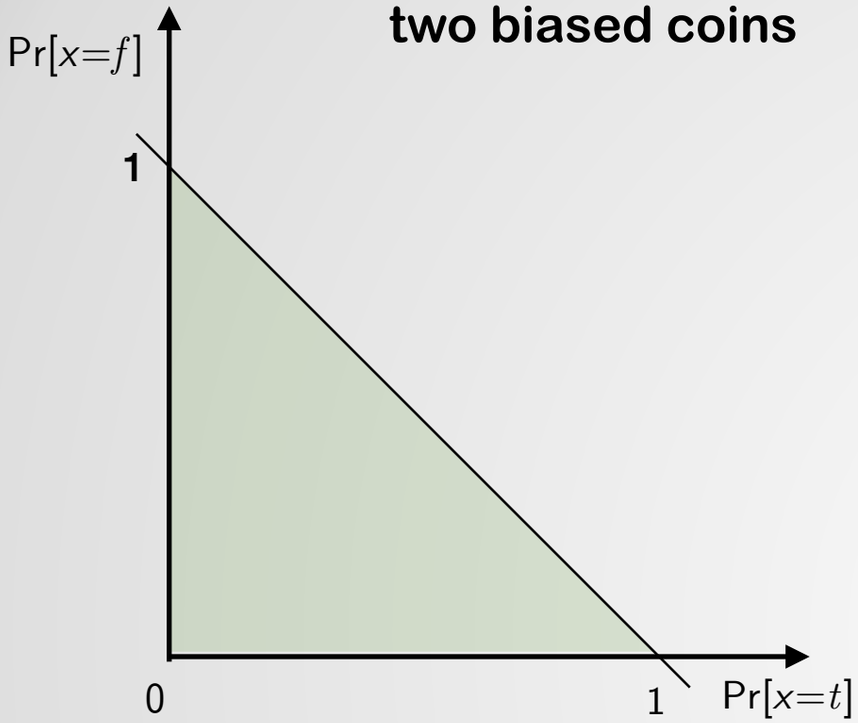


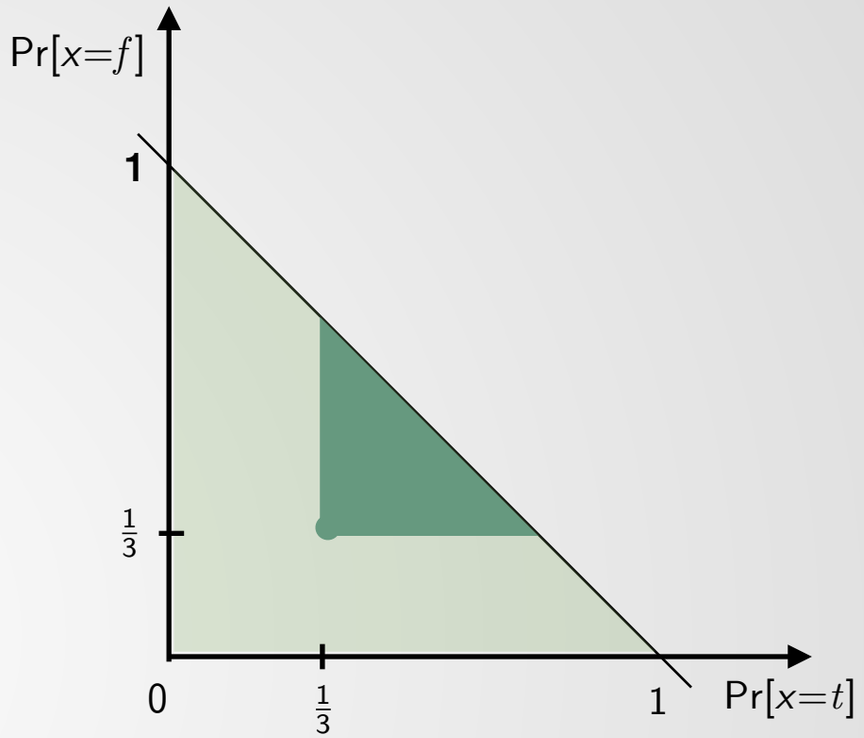
$$\{\ x := t\ \ [p]\ \ x := f\ \} \ \square\ \{\ x := t\ \ [q]\ \ x := f\ \}$$

$$
\begin{aligned}
[\![x := E]\!] &= \lambda s \bullet \{\eta_{s'}\} \ \text{ where } s' = s[E/x] \\
[\![\{c_1\}\ [p]\ \{c_2\}]\!] &= \lambda s \bullet [\![c_1]\!](s) \oplus_p [\![c_2]\!](s) \\
[\![\{c_1\} \square \{c_2\}]\!] &= \lambda s \bullet \bigcup_{p \in [0,1]} [\![c_1]\!](s) \oplus_p [\![c_2]\!](s)
\end{aligned}
$$

Relational Semantics of Non-det. Programs — Geometric Interpretation

Demonic choice between two biased coins

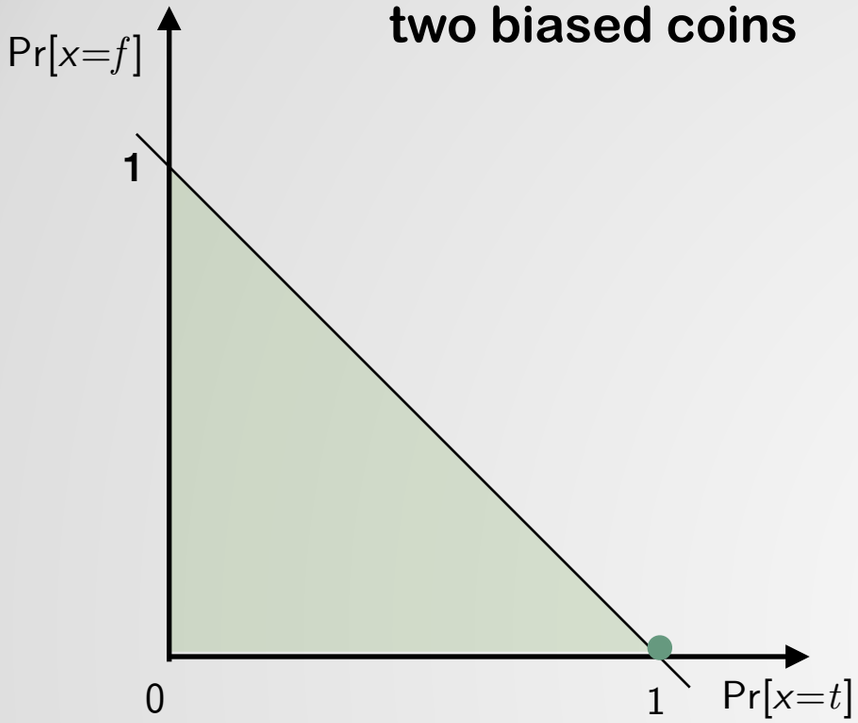


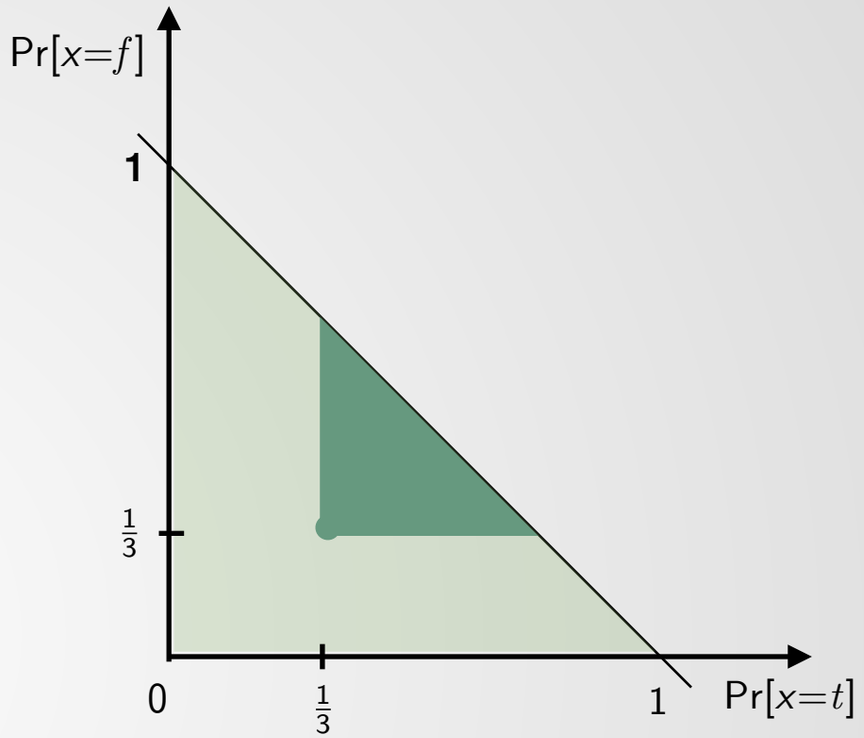


$$
\begin{aligned}
[\![x := E]\!] &= \lambda s \bullet \{\eta_{s'}\} \quad \text{where } s' = s[E/x] \\
[\![\{c_1\}\ [p]\ \{c_2\}]\!] &= \lambda s \bullet [\![c_1]\!](s) \oplus_p [\![c_2]\!](s) \\
[\![\{c_1\} \Box \{c_2\}]\!] &= \lambda s \bullet \bigcup_{p \in [0,1]} [\![c_1]\!](s) \oplus_p [\![c_2]\!](s)
\end{aligned}
$$

Relational Semantics of Non-det. Programs — Geometric Interpretation

Demonic choice between two biased coins



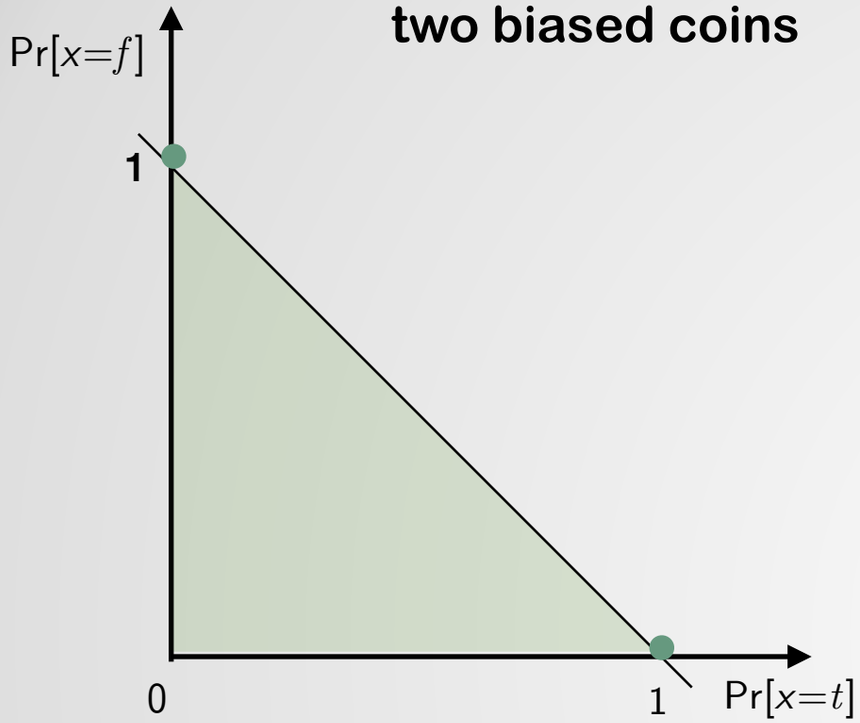


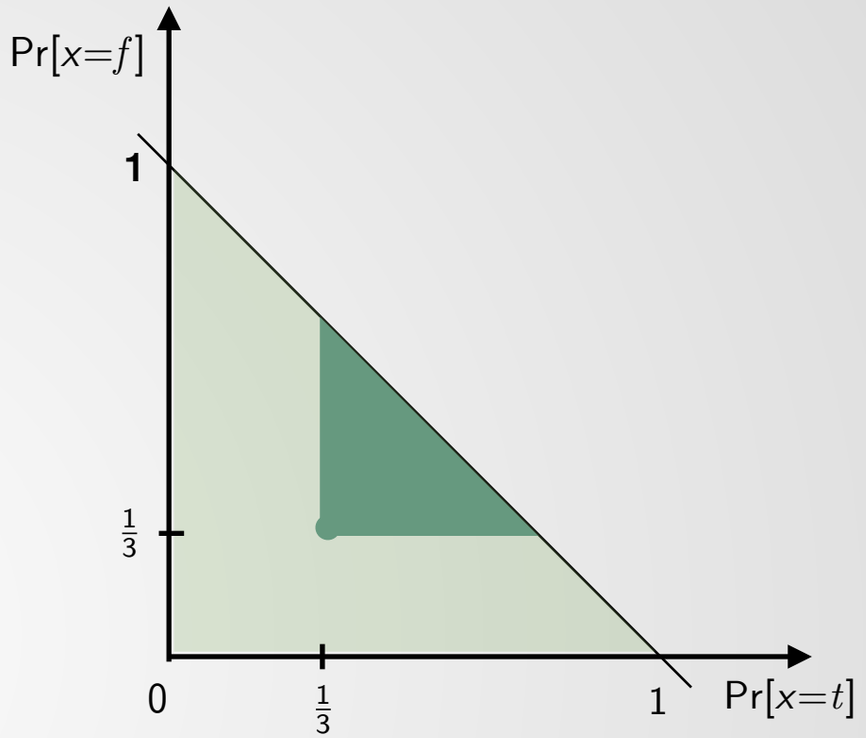
$x := t$

$$
\begin{aligned}
[\![x := E]\!] &= \lambda s \bullet \{\eta_{s'}\} \quad \text{where } s' = s[E/x] \\
[\![\{c_1\}\ [p]\ \{c_2\}]\!] &= \lambda s \bullet [\![c_1]\!](s) \oplus_p [\![c_2]\!](s) \\
[\![\{c_1\} \Box \{c_2\}]\!] &= \lambda s \bullet \bigcup_{p \in [0,1]} [\![c_1]\!](s) \oplus_p [\![c_2]\!](s)
\end{aligned}
$$

Relational Semantics of Non-det. Programs — Geometric Interpretation

Demonic choice between two biased coins

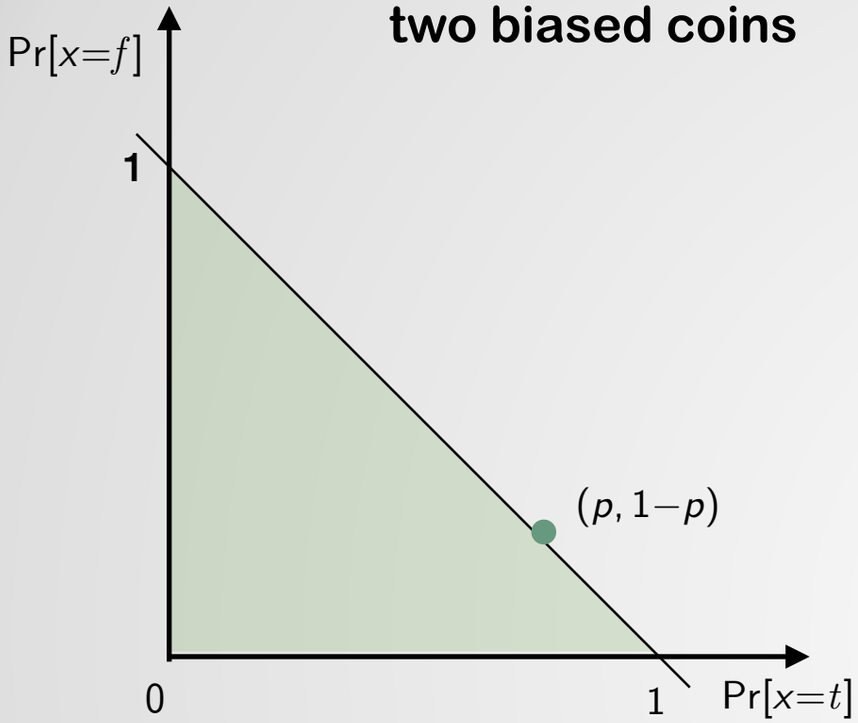


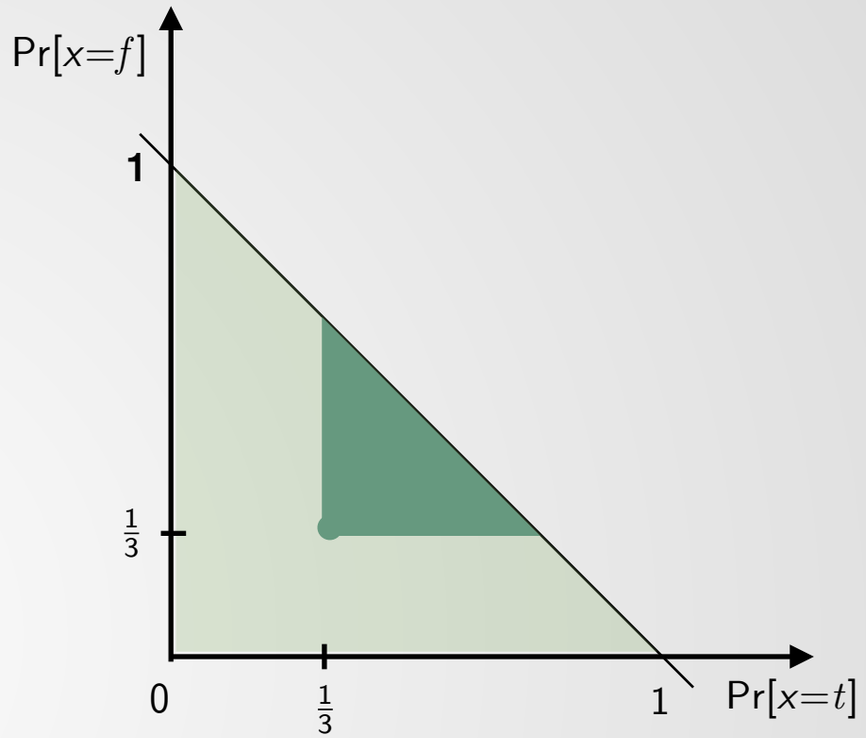


$x := t$ $\quad$ $x := f$

$$
\begin{aligned}
[\![x := E]\!] &= \lambda s \bullet \{\eta_{s'}\} \text{ where } s' = s[E/x] \\
[\![\{c_1\}\ [p]\ \{c_2\}]\!] &= \lambda s \bullet [\![c_1]\!](s) \oplus_p [\![c_2]\!](s) \\
[\![\{c_1\} \Box \{c_2\}]\!] &= \lambda s \bullet \bigcup_{p \in [0,1]} [\![c_1]\!](s) \oplus_p [\![c_2]\!](s)
\end{aligned}
$$

Relational Semantics of Non-det. Programs — Geometric Interpretation



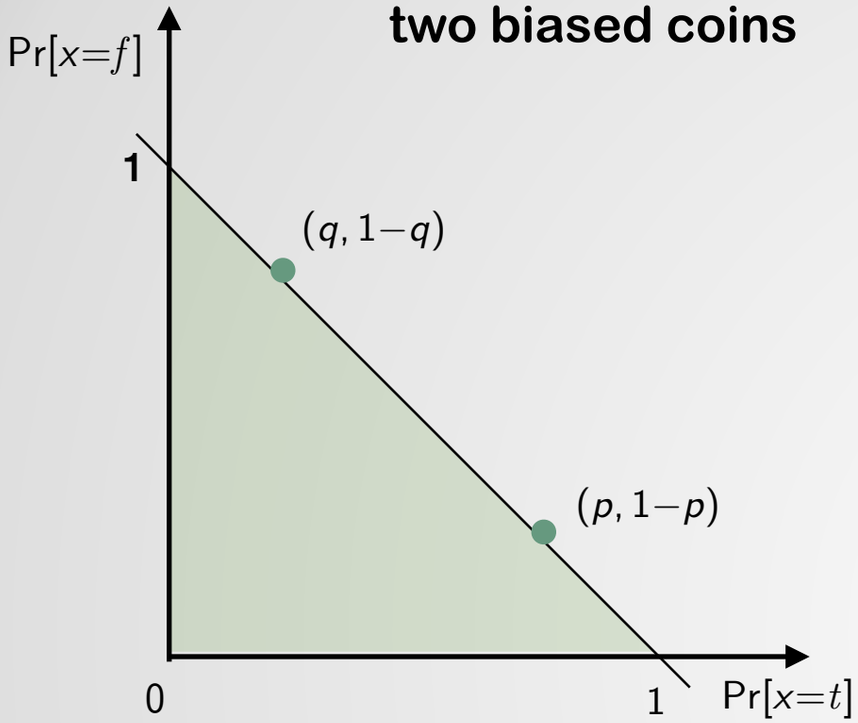


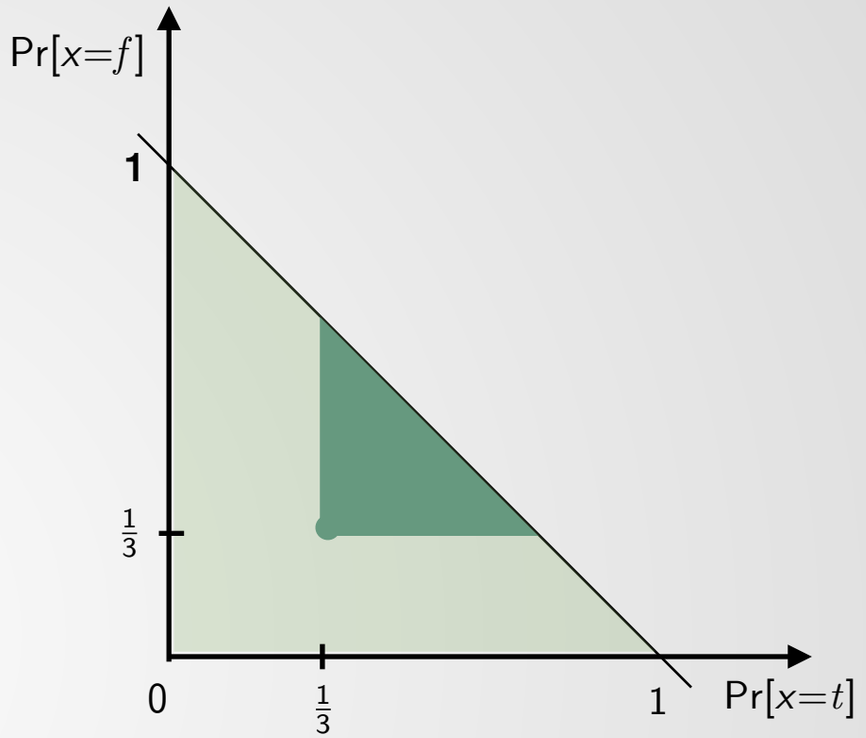
$\{\ x := t\ \ [p]\ \ x := f\ \}$

$$
\begin{aligned}
[\![x := E]\!] &= \lambda s \bullet \{\eta_{s'}\} \ \text{ where } s' = s[E/x] \\
[\![\{c_1\}\ [p]\ \{c_2\}]\!] &= \lambda s \bullet [\![c_1]\!](s) \oplus_p [\![c_2]\!](s) \\
[\![\{c_1\} \Box \{c_2\}]\!] &= \lambda s \bullet \bigcup_{p \in [0,1]} [\![c_1]\!](s) \oplus_p [\![c_2]\!](s)
\end{aligned}
$$

Relational Semantics of Non-det. Programs — Geometric Interpretation

Demonic choice between two biased coins

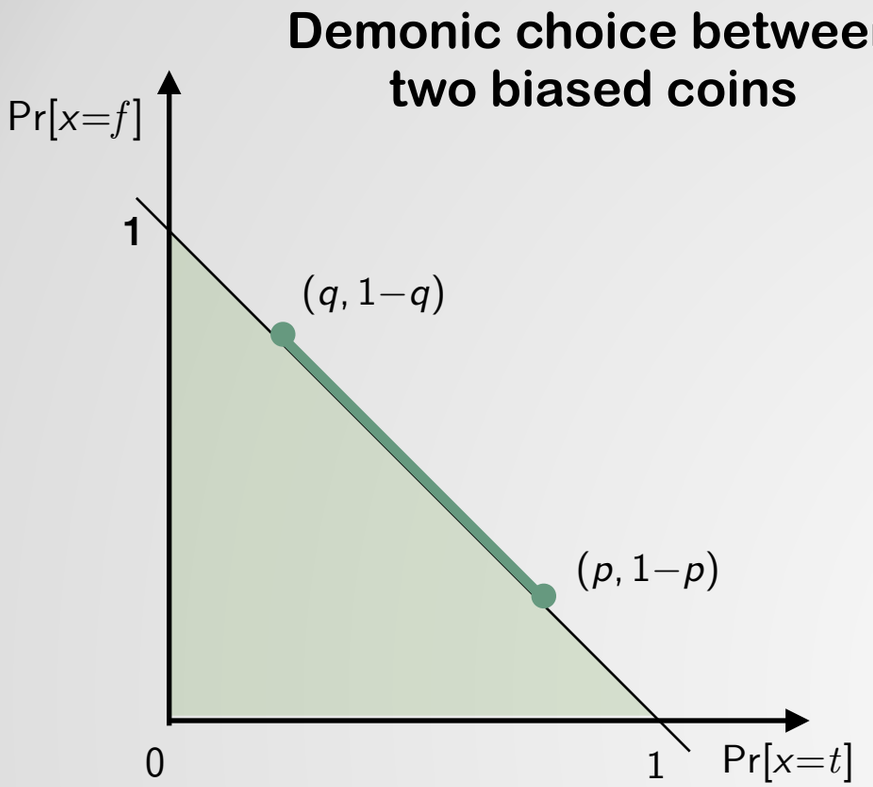


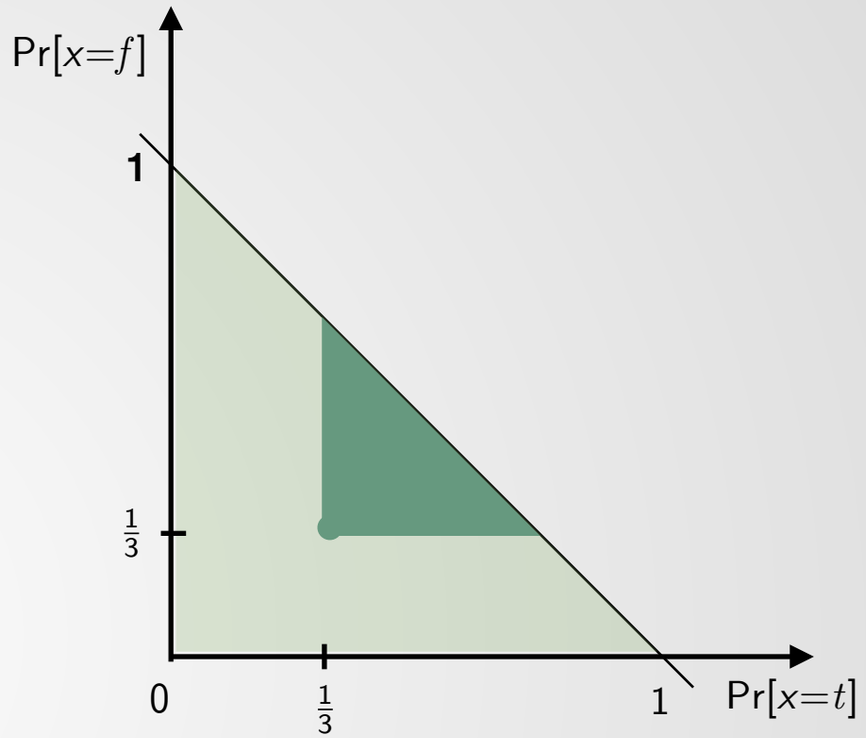


$\{\ x := t\ \ [p]\ \ x := f\ \}$ $\quad$ $\{\ x := t\ \ [q]\ \ x := f\ \}$

$$
\begin{aligned}
[\![x := E]\!] &= \lambda s \bullet \{\eta_{s'}\} \ \text{ where } s' = s[E/x] \\
[\![\{c_1\}\ [p]\ \{c_2\}]\!] &= \lambda s \bullet [\![c_1]\!](s) \oplus_p [\![c_2]\!](s) \\
[\![\{c_1\} \Box \{c_2\}]\!] &= \lambda s \bullet \bigcup_{p \in [0,1]} [\![c_1]\!](s) \oplus_p [\![c_2]\!](s)
\end{aligned}
$$

Relational Semantics of Non-det. Programs — Geometric Interpretation



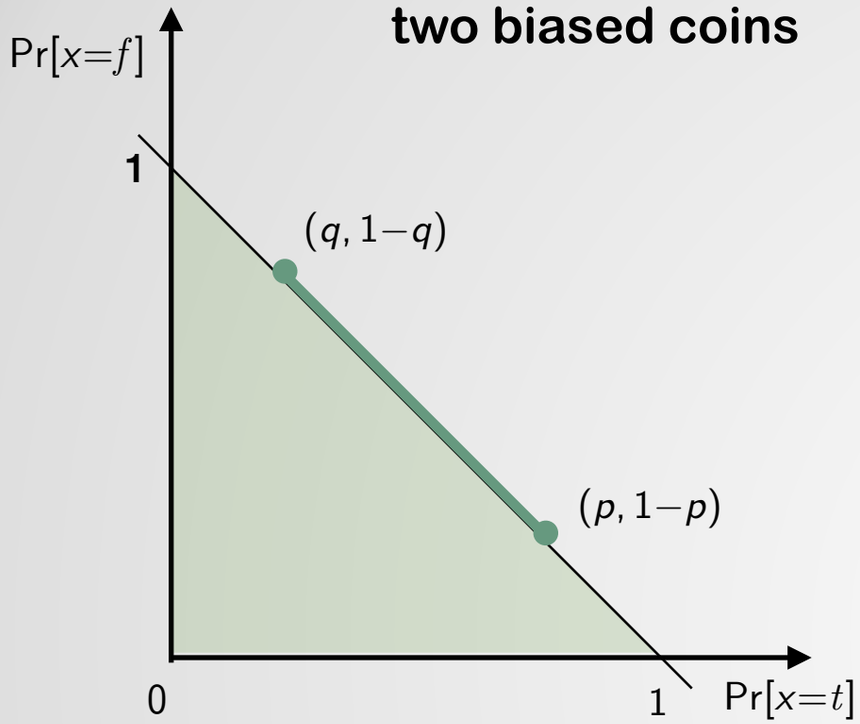


$\{\ x := t\ \ [p]\ \ x := f\ \} \ \square\ \{\ x := t\ \ [q]\ \ x := f\ \}$

$$
\begin{aligned}
[\![x := E]\!] &= \lambda s \bullet \{\eta_{s'}\} \ \text{ where } s' = s[E/x] \\
[\![\{c_1\}\ [p]\ \{c_2\}]\!] &= \lambda s \bullet [\![c_1]\!](s) \oplus_p [\![c_2]\!](s) \\
[\![\{c_1\} \square \{c_2\}]\!] &= \lambda s \bullet \bigcup_{p \in [0,1]} [\![c_1]\!](s) \oplus_p [\![c_2]\!](s)
\end{aligned}
$$

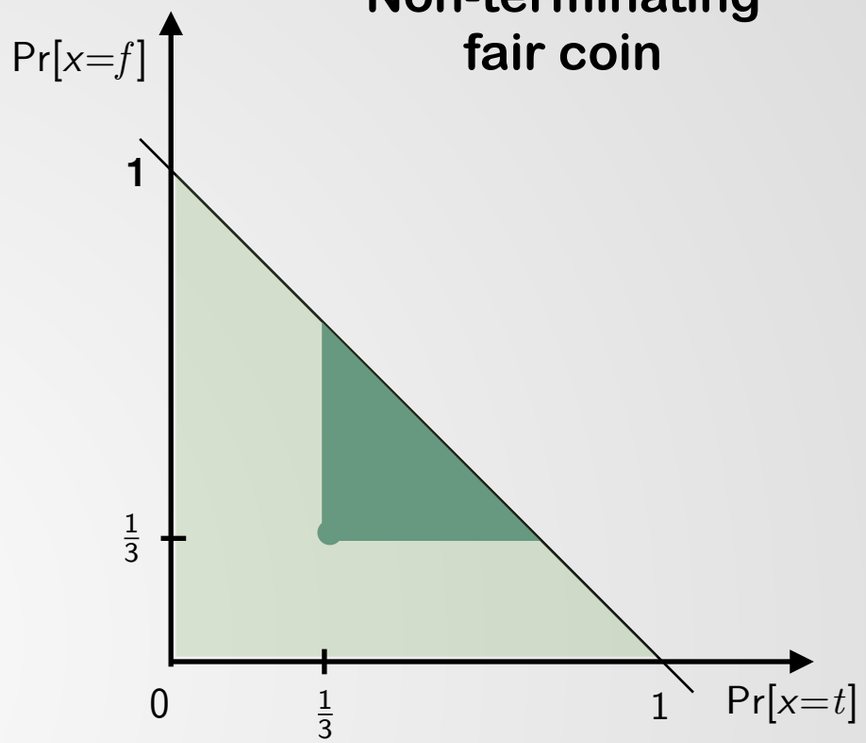
Relational Semantics of Non-det. Programs — Geometric Interpretation

Demonic choice between two biased coins



$\{\ x := t\ \ [p]\ \ x := f\ \} \ \square\ \{\ x := t\ \ [q]\ \ x := f\ \}$

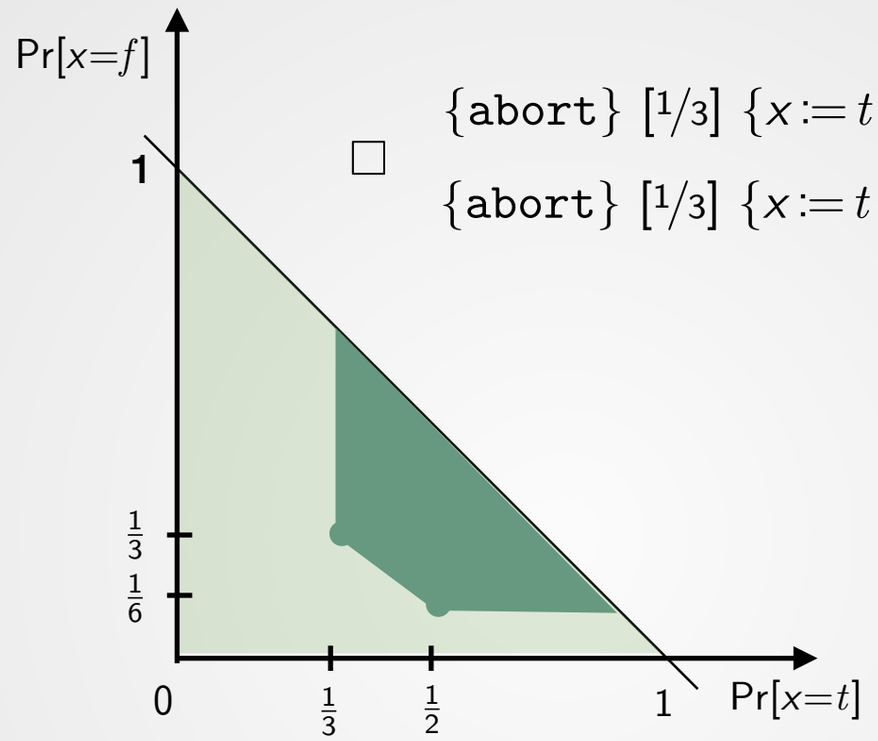
Non-terminating fair coin



$\{\texttt{abort}\}\ [1/3]\ \{x := t\ \ [1/2]\ \ x := f\}$

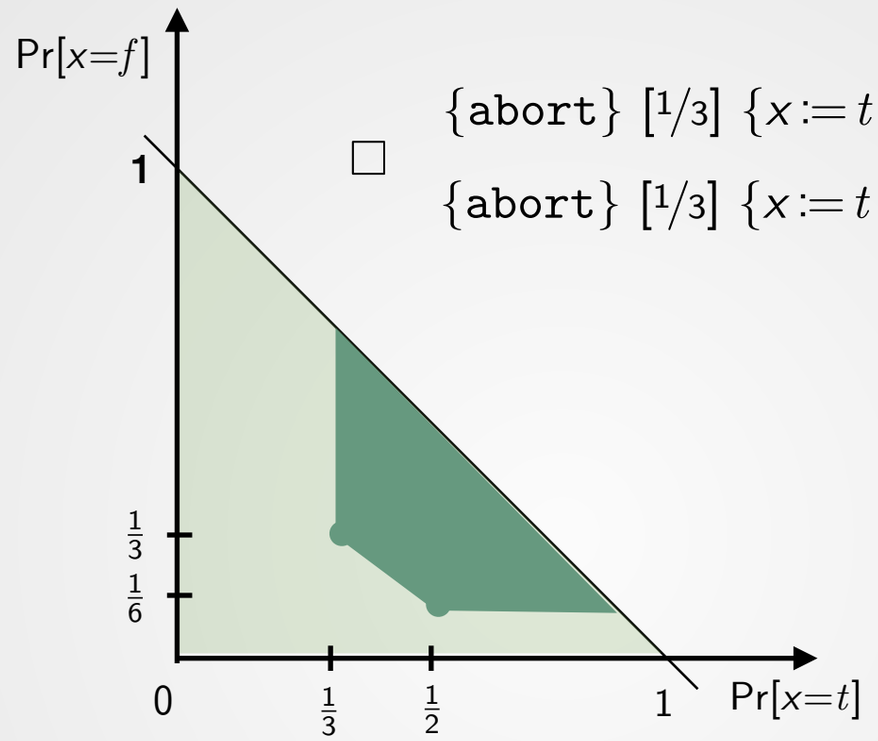
$$
\begin{aligned}
[\![x := E]\!] &= \lambda s \bullet \{\eta_{s'}\} \ \text{ where } s' = s[E/x] \\
[\![\{c_1\}\ [p]\ \{c_2\}]\!] &= \lambda s \bullet [\![c_1]\!](s) \oplus_p [\![c_2]\!](s) \\
[\![\{c_1\} \square \{c_2\}]\!] &= \lambda s \bullet \bigcup_{p \in [0,1]} [\![c_1]\!](s) \oplus_p [\![c_2]\!](s)
\end{aligned}
$$

Relational Semantics of Non-det. Programs — Geometric Interpretation



$$\{\texttt{abort}\}\ [1/3]\ \{x := t\ \ [1/2]\ \ x := f\}$$
$$\square$$
$$\{\texttt{abort}\}\ [1/3]\ \{x := t\ \ [3/4]\ \ x := f\}$$

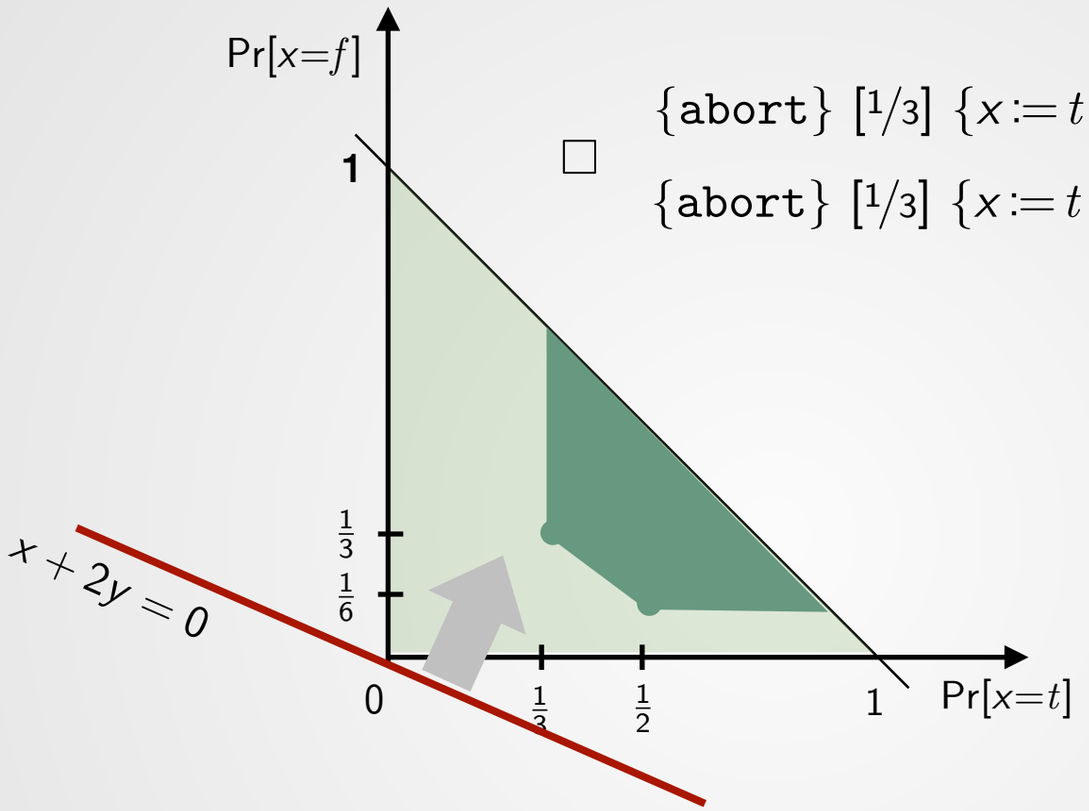
Relational Semantics of Non-det. Programs — Geometric Interpretation



$$\{\texttt{abort}\}\ [1/3]\ \{x \coloneqq t\ \ [1/2]\ \ x \coloneqq f\}$$
$$\square$$
$$\{\texttt{abort}\}\ [1/3]\ \{x \coloneqq t\ \ [3/4]\ \ x \coloneqq f\}$$

$$\mathsf{wp}\big[c\big]\big([x{=}t] + 2 \cdot [x{=}f]\big)$$

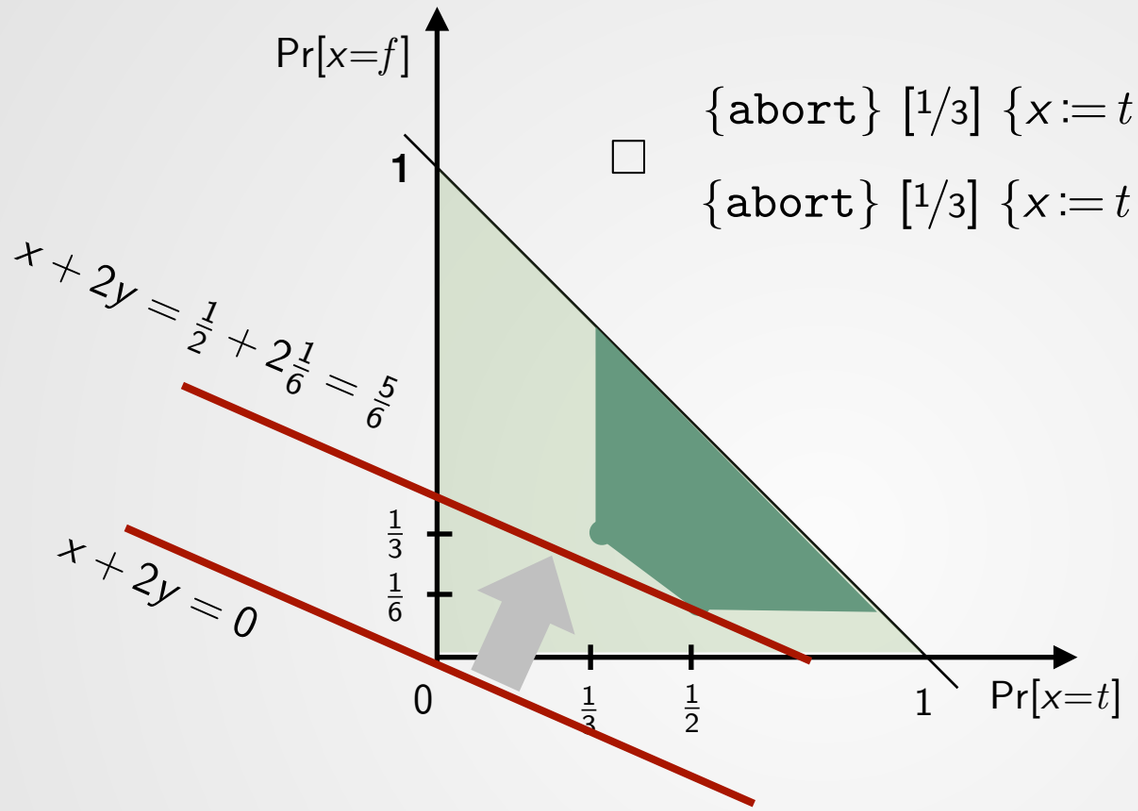
Relational Semantics of Non-det. Programs — Geometric Interpretation



$$\{\texttt{abort}\}\ [1/3]\ \{x := t\ \ [1/2]\ \ x := f\}$$
$$\square$$
$$\{\texttt{abort}\}\ [1/3]\ \{x := t\ \ [3/4]\ \ x := f\}$$

$$\mathsf{wp}\big[c\big]\big([x{=}t] + 2 \cdot [x{=}f]\big)$$

Relational Semantics of Non-det. Programs — Geometric Interpretation



$$\square \quad \begin{array}{l} \{\texttt{abort}\}\ [1/3]\ \{x := t\ \ [1/2]\ \ x := f\} \\ \{\texttt{abort}\}\ [1/3]\ \{x := t\ \ [3/4]\ \ x := f\} \end{array}$$

$$\mathsf{wp}\big[c\big]\big([x{=}t] + 2 \cdot [x{=}f]\big) \ = \underline{\tfrac{5}{6}}$$

Agenda

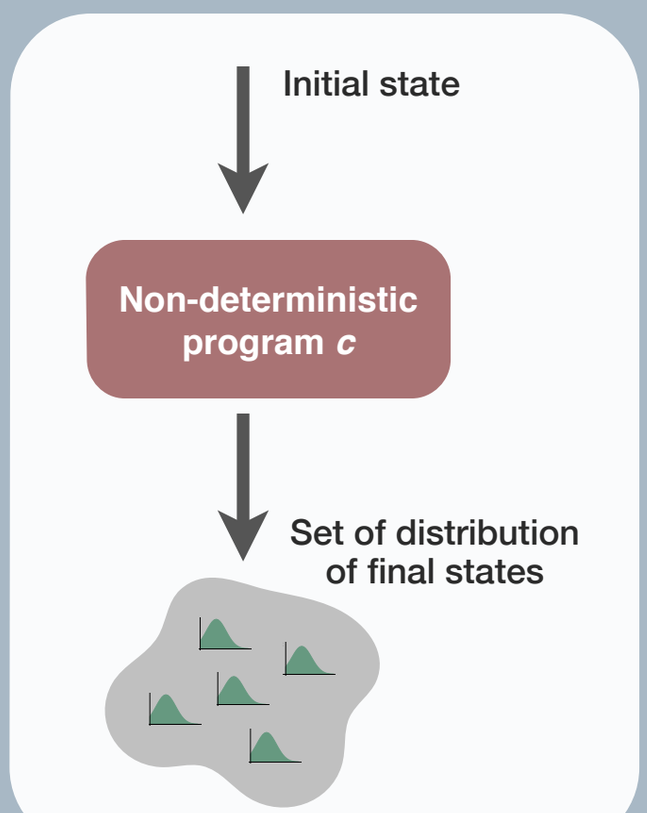
- Recap on previous lecture
- Algebraic properties
- Extension to unbounded expectations
- Connection to relational semantics
- Extension to non-deterministic programs
- Summary

Summary

Algebraic Properties

- Monotonicity
- Feasibility
- Linearity
- (Co-)Strictness

Relational Semantics

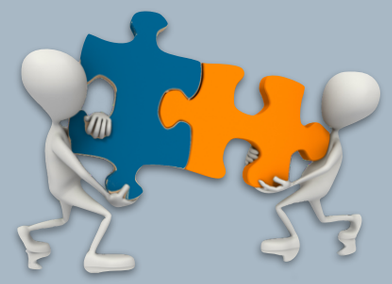


Extensions

Unbounded expectations

Non-deterministic programs

Connection between the two semantical views



$$\mathrm{wp}[c](f) \;=\; \lambda s \bullet\; \inf \left\{ \mathbf{EV}_{\mu'}(f) \mid \mu' \in [\![c]\!](s) \right\}$$