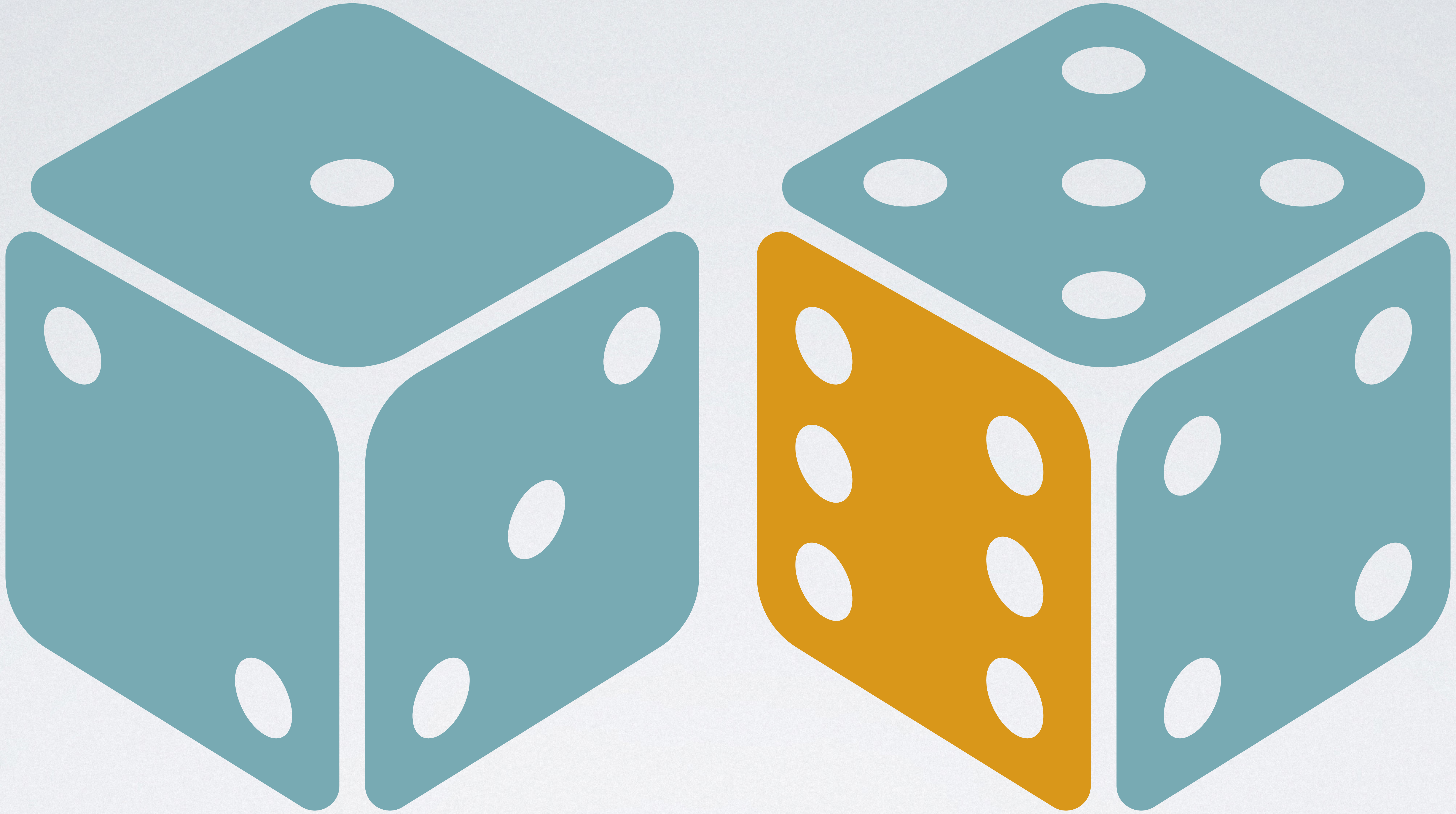
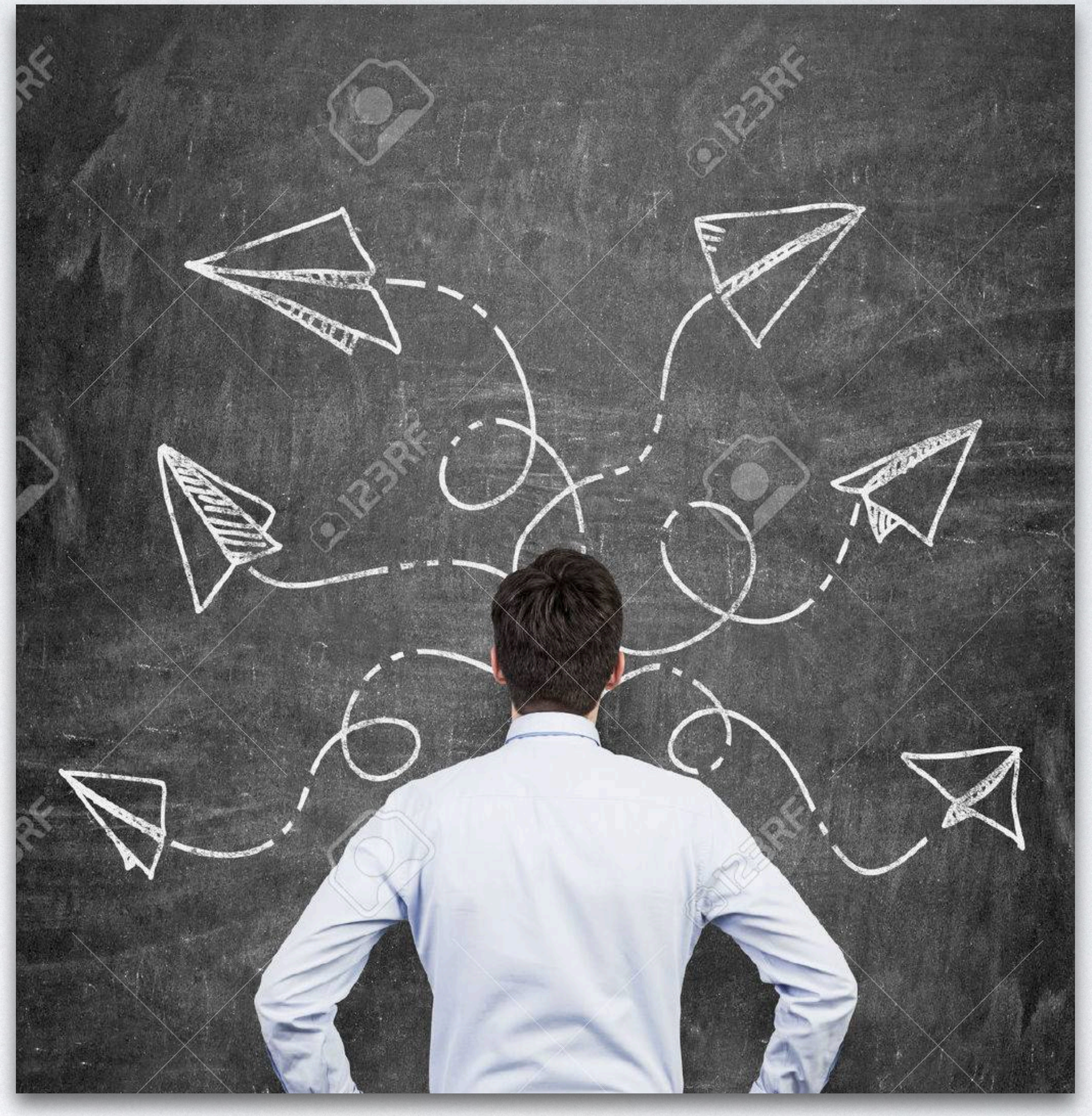


PMAF: AN ALGEBRAIC FRAMEWORK FOR STATIC ANALYSIS OF PROBABILISTIC PROGRAMS

Di Wang[1], Jan Hoffmann[1], Thomas Reps[2]

[1] Carnegie Mellon University
[2] University of Wisconsin; GrammaTech, Inc.

PROBABILISTIC PROGRAMS



Draw random **data** from distributions

Condition **control-flow** at random

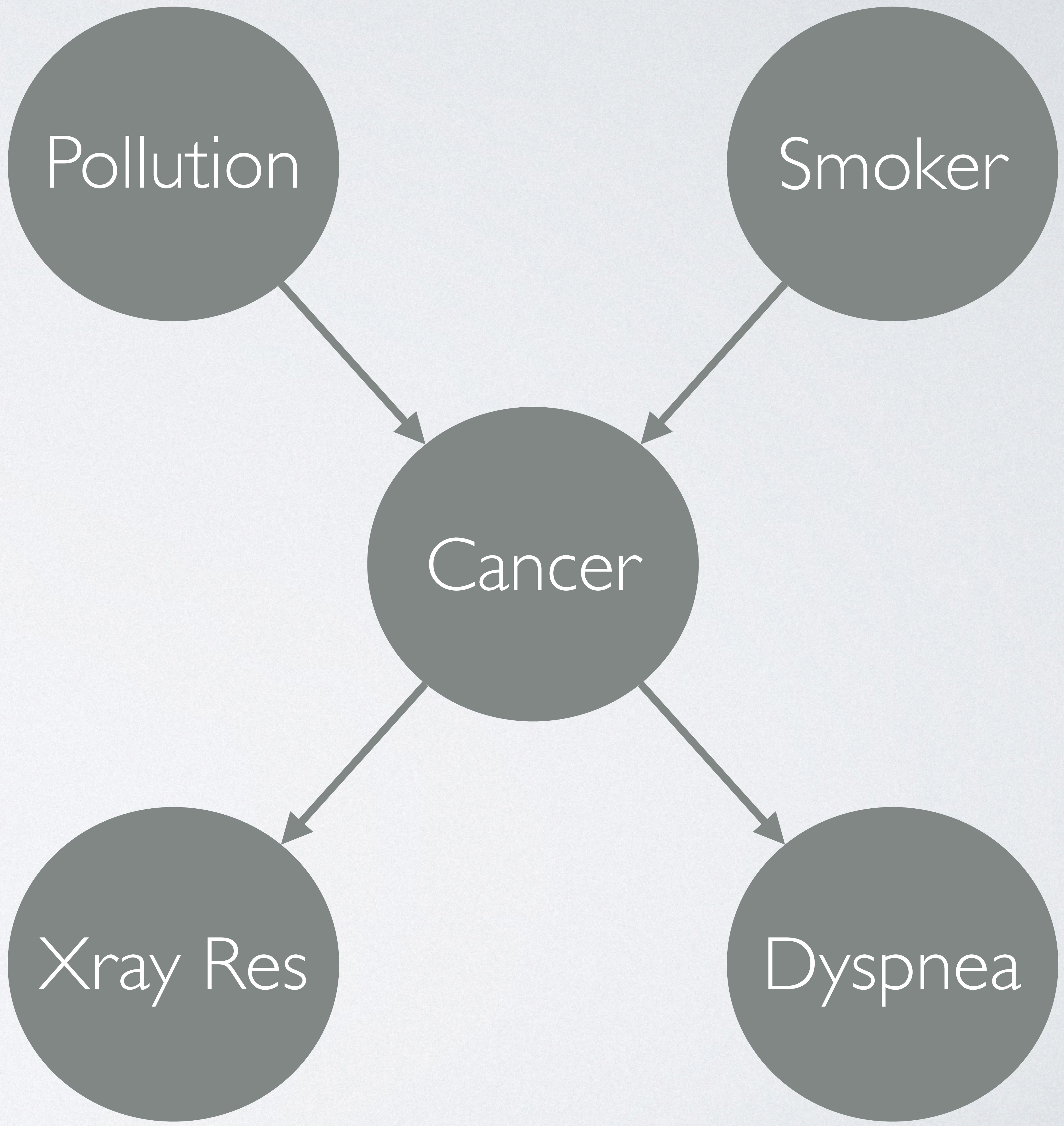
PROBABILISTIC PROGRAMS

- True randomness
- Distributions on executions

```
b1 ~ Bernoulli(0.5);
b2 ~ Bernoulli(0.7);
while (b1 && b2) do
   if prob(0.6) then
     b1 ~ Bernoulli(0.5)
   else
     b2 ~ Bernoulli(0.7)
   fi;
   tick(1.0)
od;
return (b1, b2)
```

BAYESIAN NETWORKS

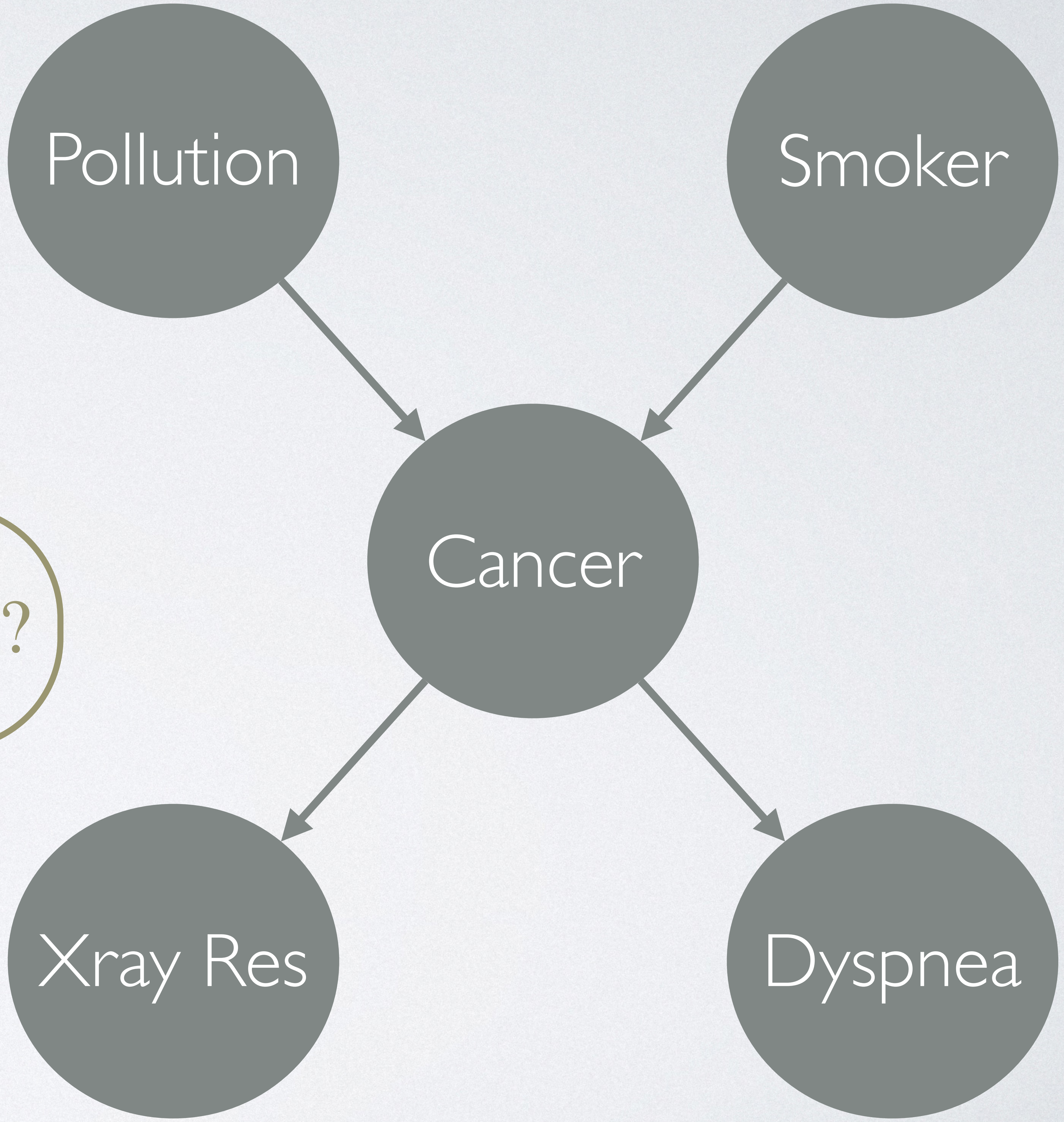
- Conditional distributions
- Query about the posterior



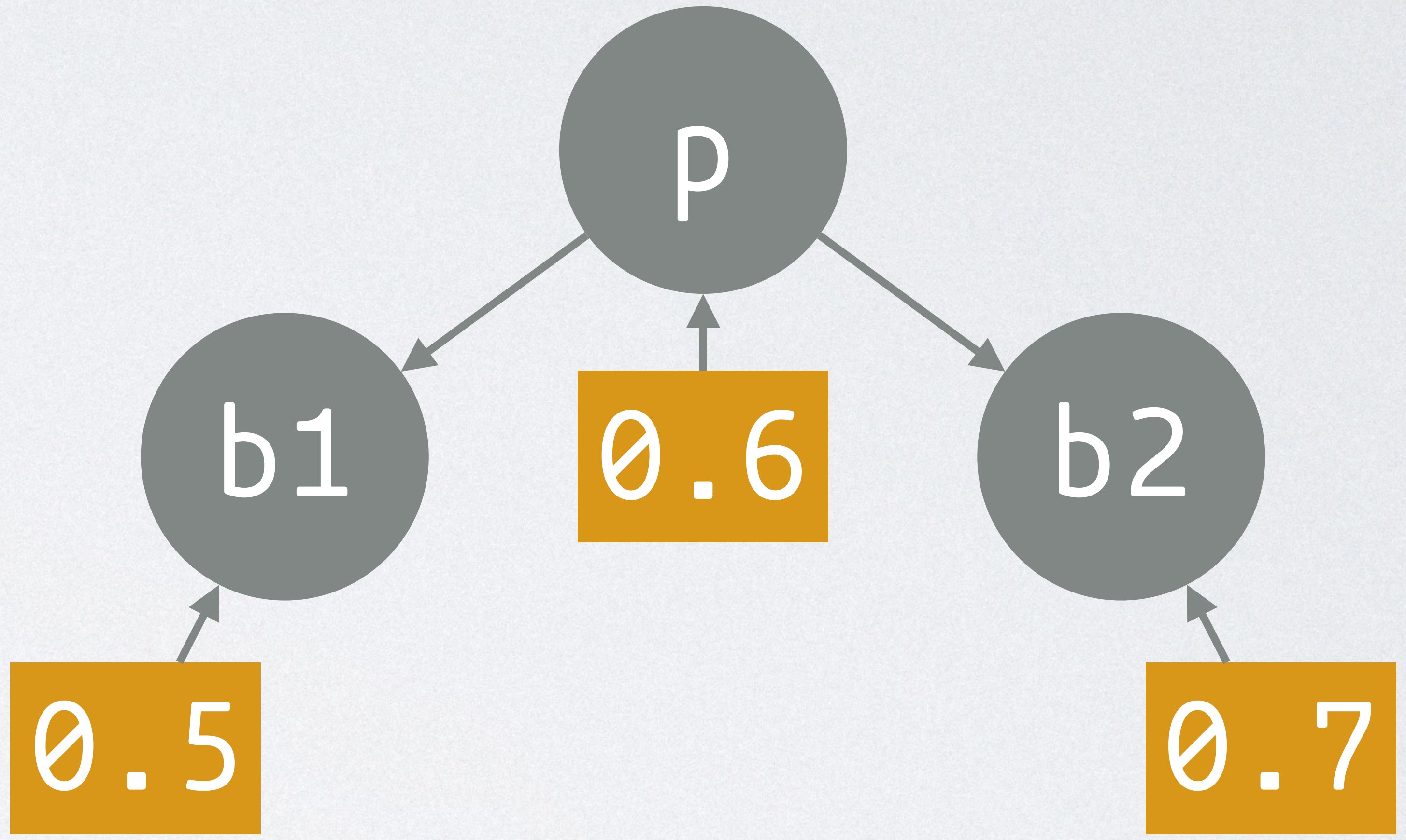
BAYESIAN NETWORKS

- Conditional distributions
- Query about the posterior

Prob[Cancer | Smoker ∧ Xray Res] = ?

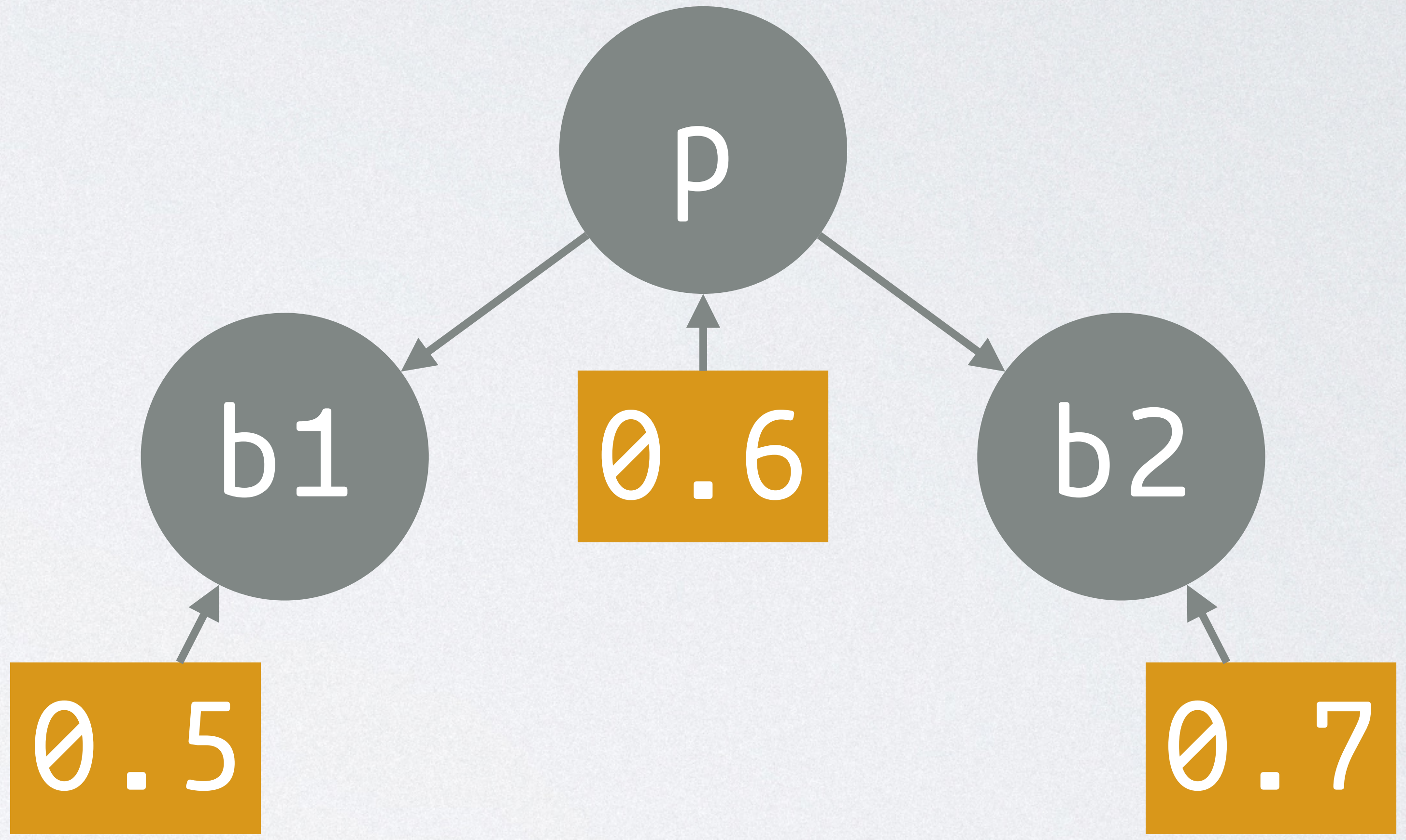


BAYESIAN NETWORKS AS PROB. PROG.



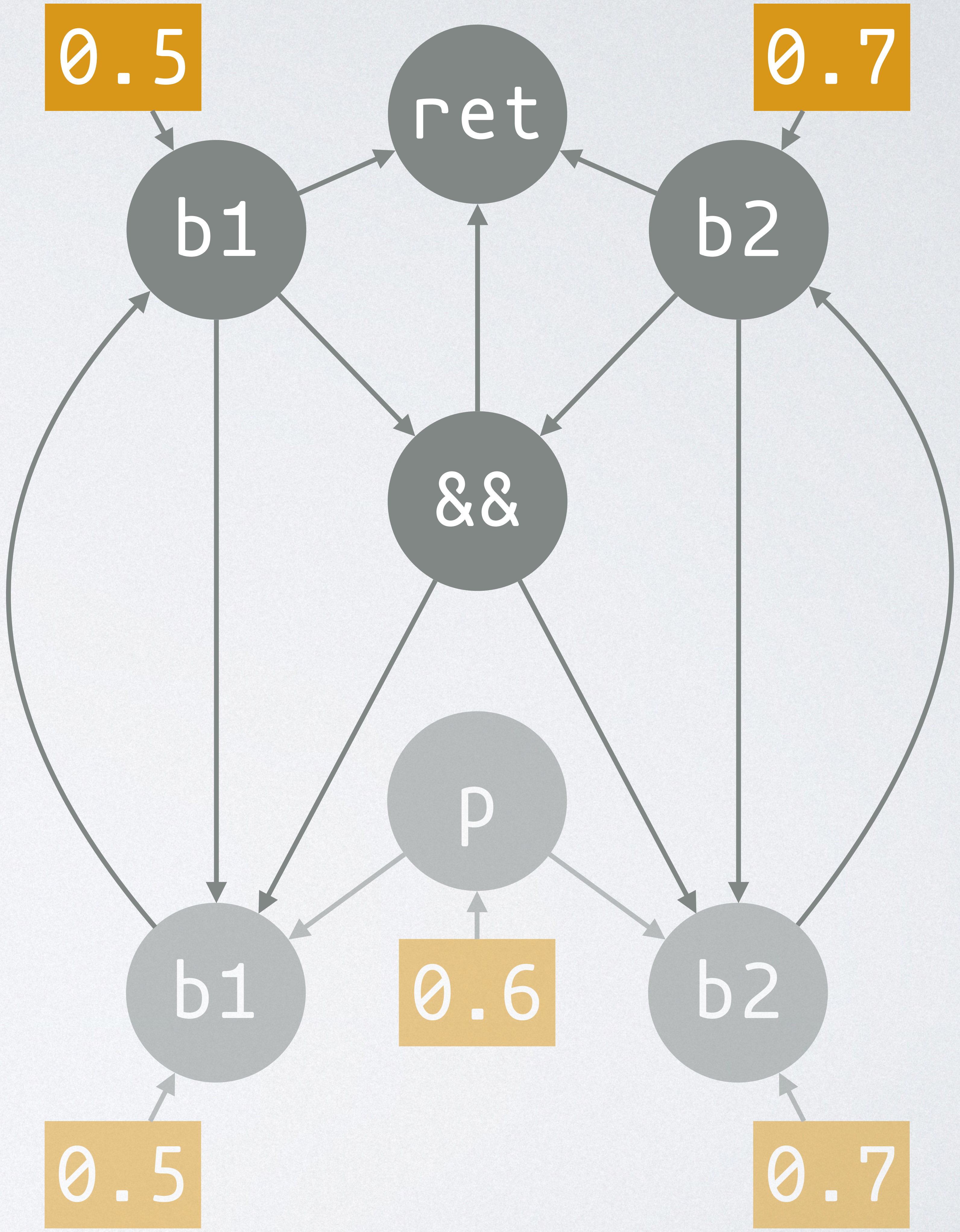
BAYESIAN NETWORKS AS PROB. PROG.

```
if prob(0.6) then
  b1 ~ Bernoulli(0.5)
else
  b2 ~ Bernoulli(0.7)
fi
```



BAYESIAN NETWORKS AS PROB. PROG.

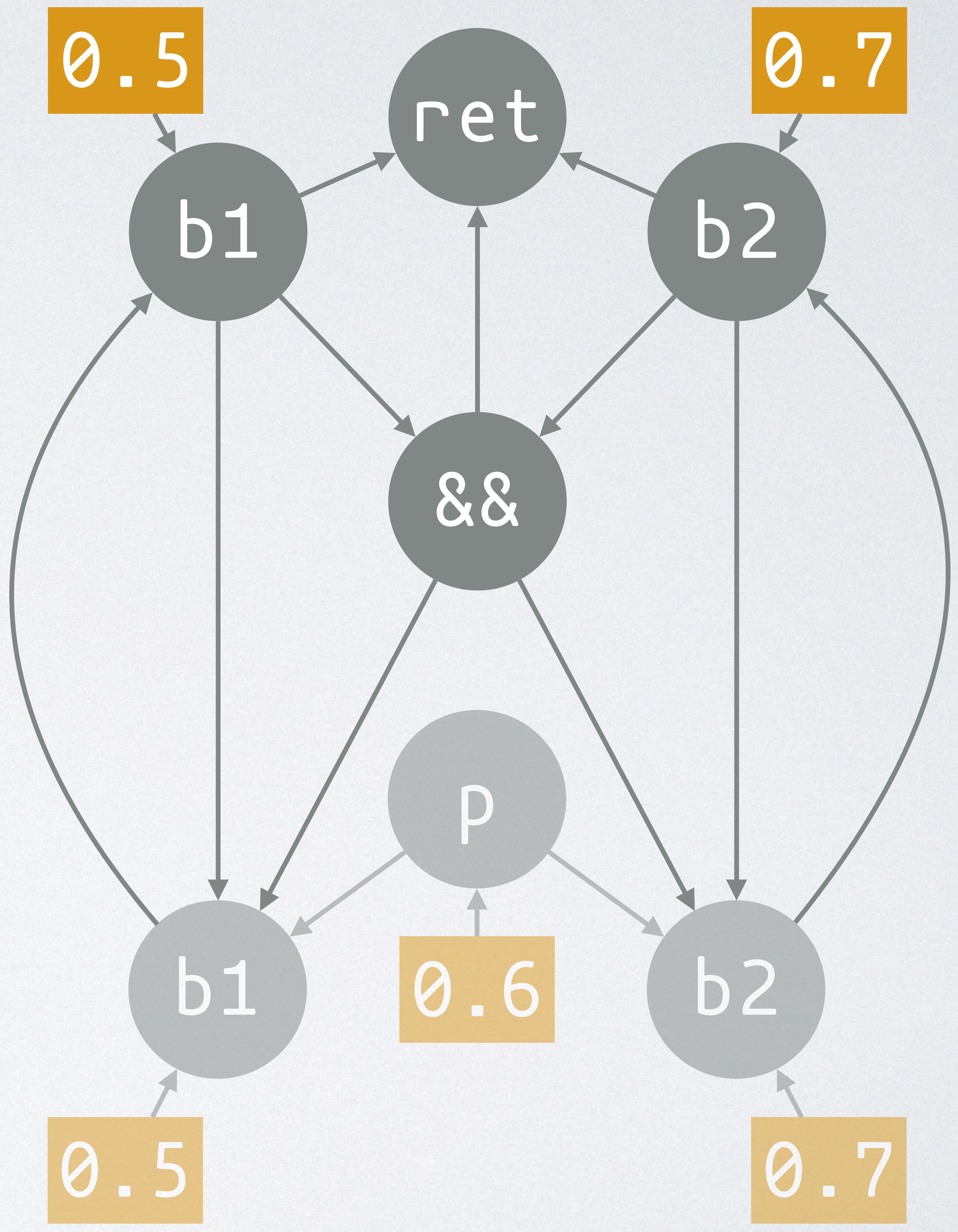
```
b1 ~ Bernoulli(0.5);
b2 ~ Bernoulli(0.7);
while (b1 && b2) do
   if prob(0.6) then
      b1 ~ Bernoulli(0.5)
   else
      b2 ~ Bernoulli(0.7)
   fi;
   tick(1.0)
od;
return (b1, b2)
```



BAYESIAN NETWORKS AS PROB. PROG.

```
b1 ~ Bernoulli(0.5);
b2 ~ Bernoulli(0.7);
while (b1 && b2) do
   if prob(0.6) then
      b1 ~ Bernoulli(0.5)
   else
      b2 ~ Bernoulli(0.7)
   fi;
   tick(1.0)
od;
return (b1, b2)
```

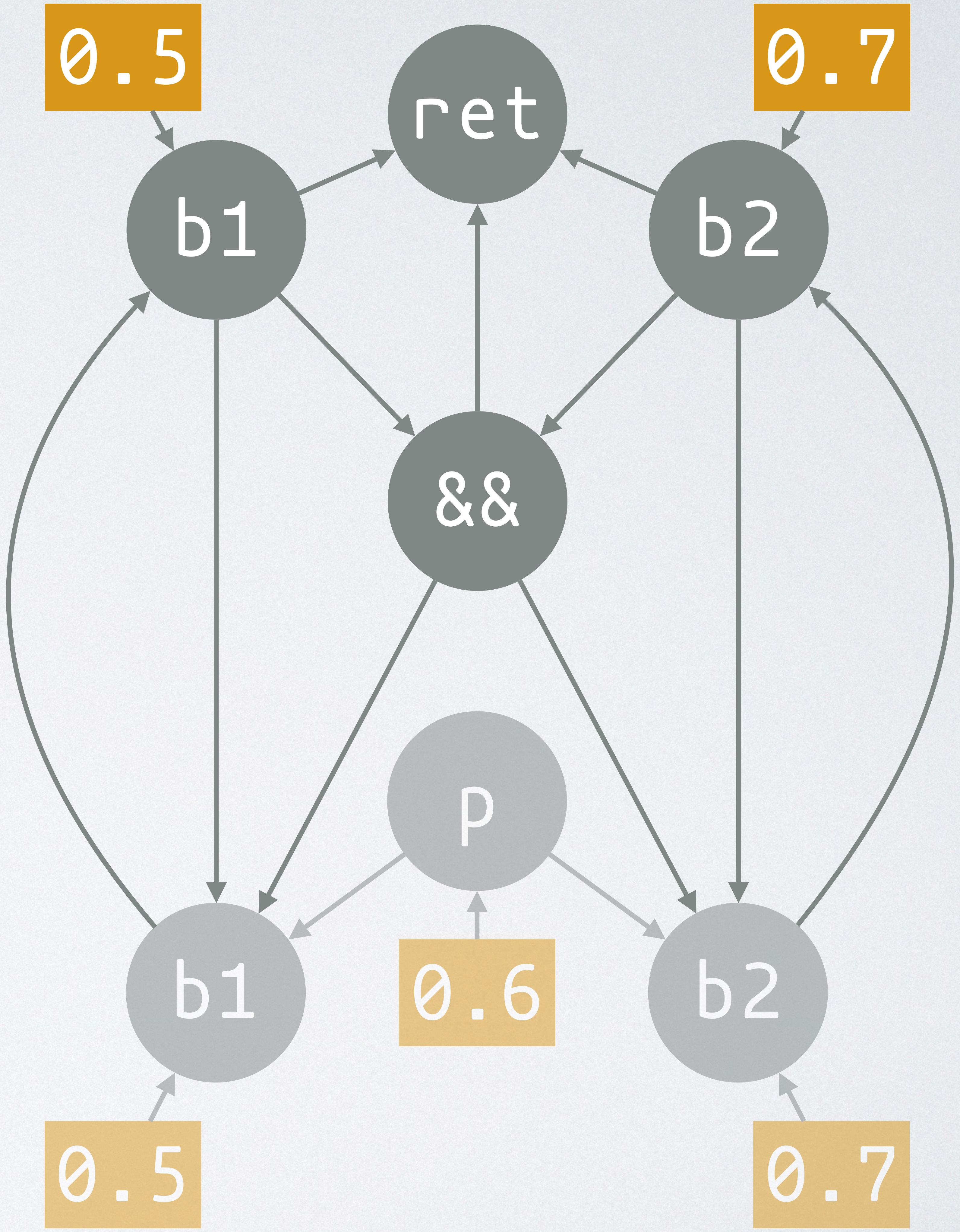
- **Query:** probability that `b1` and `b2` are both `false`?



BAYESIAN NETWORKS AS PROB. PROG.

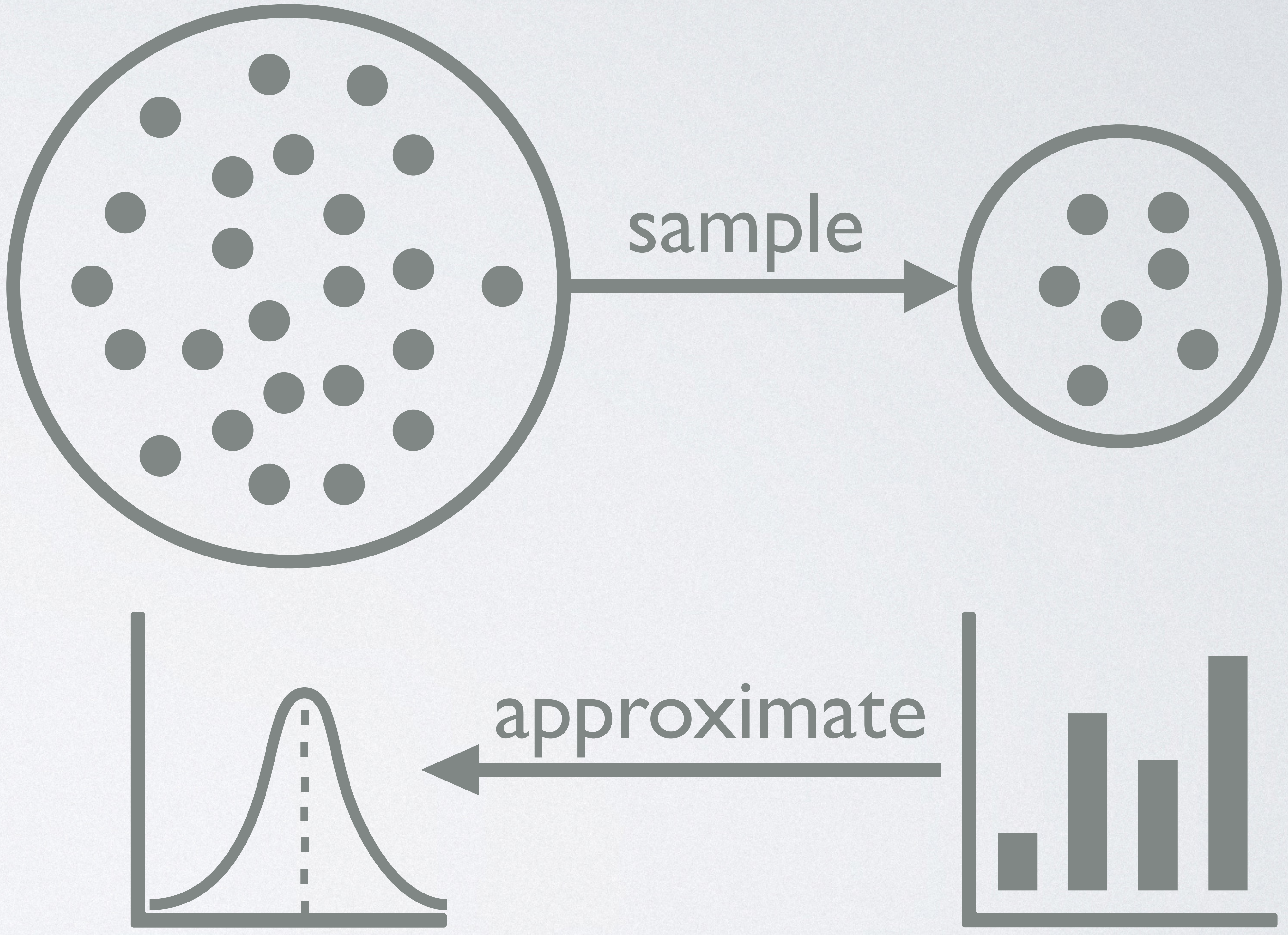
```
b1 ~ Bernoulli(0.5);
b2 ~ Bernoulli(0.7);
while (b1 && b2) do
   if prob(0.6) then
      b1 ~ Bernoulli(0.5)
   else
      b2 ~ Bernoulli(0.7)
   fi;
   tick(1.0)
od;
return (b1, b2)
```

- Query: expected termination time?



SAMPLING-BASED TECHNIQUES

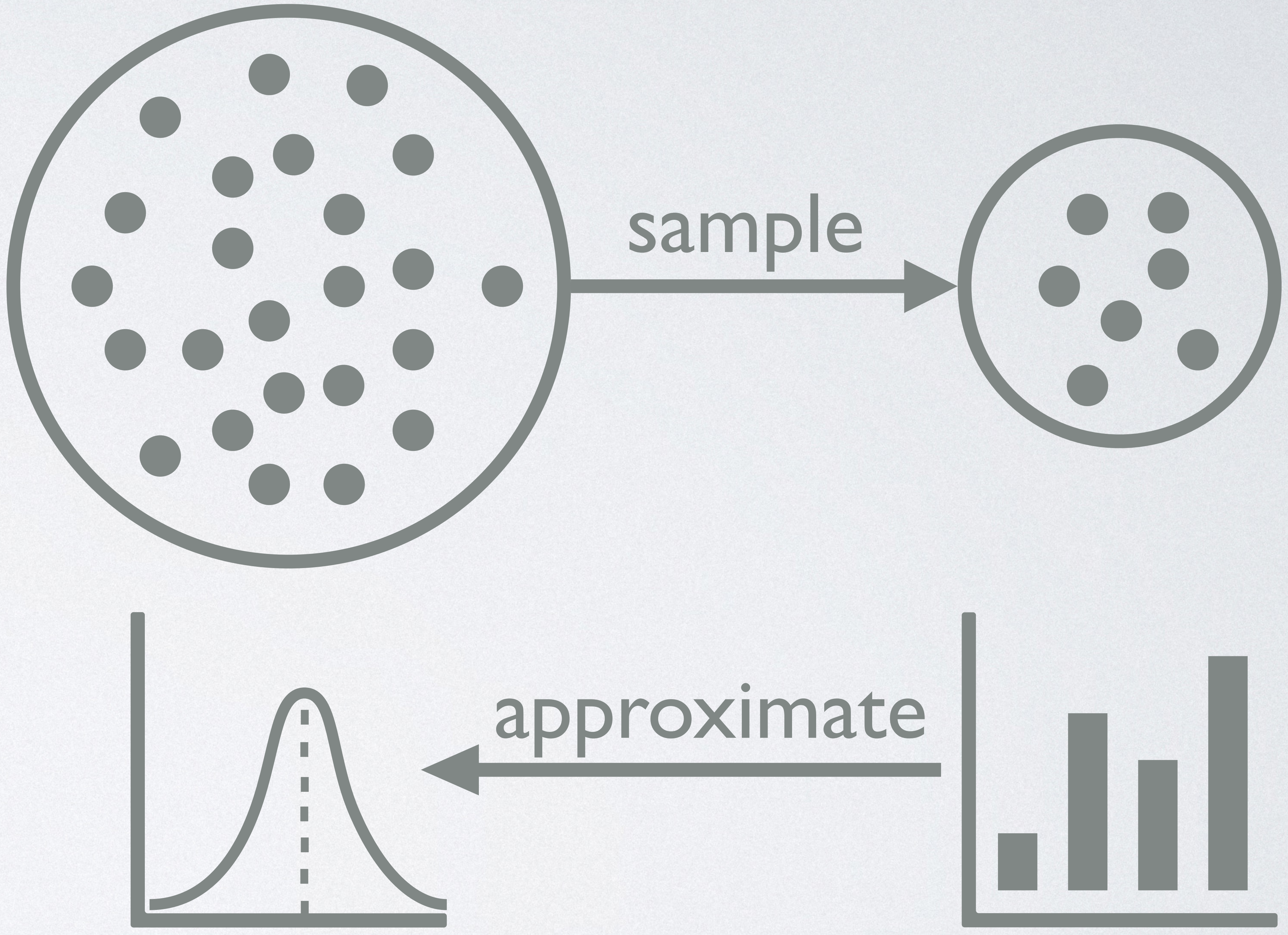
- Simulation & frequency count
- Flexible & universal
- Potentially unsound & inefficient



SAMPLING-BASED TECHNIQUES

- Simulation & frequency count
- Flexible & universal
- Potentially unsound & inefficient

What about **static analysis**?



ABSTRACT INTERPRETATION

- Cousot et al. proposed **Probabilistic Abstract Interpretation**[1]
- **Sound**, flexible, and universal

[1] P. Cousot and M. Monerau. Probabilistic Abstract Interpretation. In *ESOP'12*.

ABSTRACT INTERPRETATION

- Cousot et al. proposed **Probabilistic Abstract Interpretation**[1]
- **Sound**, flexible, and universal
- Their concrete semantics resolves probabilities ***prior* to** nondeterminism

[1] P. Cousot and M. Monerau. Probabilistic Abstract Interpretation. In *ESOP'12*.

ABSTRACT INTERPRETATION

- Cousot et al. proposed **Probabilistic Abstract Interpretation**[1]
- **Sound**, flexible, and universal
- Their concrete semantics resolves probabilities ***prior* to** nondeterminism
- Sometimes desirable to revolve nondeterminism ***prior to*** probabilities

[1] P. Cousot and M. Monerau. Probabilistic Abstract Interpretation. In *ESOP'12*.

COUSOT ET AL.'S SEMANTICS

`*` denotes nondeterministic choice

`tick(q)` increases T by `q`

```
if * then
  if prob(0.5) then tick(1.0) else tick(2.0) fi
else
  if prob(0.5) then tick(1.0) else tick(2.0) fi
fi
```

COUSOT ET AL.'S SEMANTICS

* denotes nondeterministic choice

`tick(q)` increases T by `q`

```
if * then
  if prob(0.5) then tick(1.0) else tick(2.0) fi
else
  if prob(0.5) then tick(1.0) else tick(2.0) fi
fi
```

with prob. $\frac{1}{4}$

COUSOT ET AL.'S SEMANTICS

* denotes nondeterministic choice

`tick(q)` increases T by `q`

```
if * then
  if prob(0.5) then tick(1.0) else tick(2.0) fi
else
  if prob(0.5) then tick(1.0) else tick(2.0) fi
fi
```

COUSOT ET AL.'S SEMANTICS

* denotes nondeterministic choice

tick(q) increases T by q

```
if * then
  if prob(0.5) then tick(1.0) else tick(2.0) fi
else
  if prob(0.5) then tick(1.0) else tick(2.0) fi
fi
```

with prob. $\frac{1}{4}$

COUSOT ET AL.'S SEMANTICS

* denotes nondeterministic choice

tick(q) increases T by q

```
if * then
  if prob(0.5) then tick(1.0) else tick(2.0) fi
else
  if prob(0.5) then tick(1.0) else tick(2.0) fi
fi
```

COUSOT ET AL.'S SEMANTICS

* denotes nondeterministic choice

tick(q) increases T by q

```
if * then
  if prob(0.5) then tick(1.0) else tick(2.0) fi
else
  if prob(0.5) then tick(1.0) else tick(2.0) fi
fi
```

with prob. $\frac{1}{4}$

COUSOT ET AL.'S SEMANTICS

* denotes nondeterministic choice

`tick(q)` increases T by `q`

```
if * then
  if prob(0.5) then tick(1.0) else tick(2.0) fi
else
  if prob(0.5) then tick(1.0) else tick(2.0) fi
fi
```

COUSOT ET AL.'S SEMANTICS

* denotes nondeterministic choice

`tick(q)` increases T by `q`

```
if * then
  if prob(0.5) then tick(1.0) else tick(2.0) fi
else
  if prob(0.5) then tick(1.0) else tick(2.0) fi
fi
```

with prob. $\frac{1}{4}$

COUSOT ET AL.'S SEMANTICS

* denotes nondeterministic choice

`tick(q)` increases T by `q`

```
if * then
  if prob(0.5) then tick(1.0) else tick(2.0) fi
else
  if prob(0.5) then tick(1.0) else tick(2.0) fi
fi
```

COUSOT ET AL.'S SEMANTICS

`*` denotes nondeterministic choice

`tick(q)` increases T by `q`

```
if * then
  if prob(0.5) then tick(1.0) else tick(2.0) fi
else
  if prob(0.5) then tick(1.0) else tick(2.0) fi
fi
```

- Their concrete semantics yields

$$\mathbb{E}[T] \in \frac{1}{4} \cdot \{1\} + \frac{1}{4} \cdot \{2\} + \frac{1}{4} \cdot \{1,2\} + \frac{1}{4} \cdot \{1,2\} = \{1.25, 1.5, 1.75\}$$

COUSOT ET AL.'S SEMANTICS

* denotes nondeterministic choice

tick(q) increases T by q

```
if * then
  if prob(0.5) then tick(1.0) else tick(2.0) fi
else
  if prob(0.5) then tick(1.0) else tick(2.0) fi
fi
```

Identical!

- Their concrete semantics yields

$$\mathbb{E}[T] \in \frac{1}{4} \cdot \{1\} + \frac{1}{4} \cdot \{2\} + \frac{1}{4} \cdot \{1,2\} + \frac{1}{4} \cdot \{1,2\} = \{1.25, 1.5, 1.75\}$$

COUSOT ET AL.'S SEMANTICS

$\star$ denotes nondeterministic choice

`tick(q)` increases T by `q`

```
if ⋆ then
  if prob(0.5) then tick(1.0) else tick(2.0) fi
else
  if prob(0.5) then tick(1.0) else tick(2.0) fi
fi
```

Identical!

- Their concrete semantics yields

$$\mathbb{E}[T] \in \frac{1}{4} \cdot \{1\} + \frac{1}{4} \cdot \{2\} + \frac{1}{4} \cdot \{1,2\} + \frac{1}{4} \cdot \{1,2\} = \{1.25, 1.5, 1.75\}$$

while our semantics yields $\mathbb{E}[T] = 1.5$

CONTRIBUTIONS

- A denotational semantics with nondeterminism resolved first
- An algebraic framework for interprocedural dataflow analysis of first-order probabilistic programs

PMAF

Recursion
Unstructured control-flow
Divergence
Nondeterminism
…

CONTRIBUTIONS

- A denotational semantics with nondeterminism resolved first
- An algebraic framework for interprocedural dataflow analysis of first-order probabilistic programs

PMAF

CONTRIBUTIONS

- A denotational semantics with nondeterminism resolved first
- An algebraic framework for interprocedural dataflow analysis of first-order probabilistic programs

PMAF

Design → Prove → Implement

CONTRIBUTIONS

- A denotational semantics with nondeterminism resolved first
- An algebraic framework for interprocedural dataflow analysis of first-order probabilistic programs

PMAF

CONTRIBUTIONS

- A denotational semantics with nondeterminism resolved first
- An **algebraic framework** for **interprocedural dataflow analysis** of **first-order probabilistic programs**

PMAF

Existing

- Bayesian Inference
- Markov Decision Problem

CONTRIBUTIONS

- A denotational semantics with nondeterminism resolved first
- An algebraic framework for interprocedural dataflow analysis of first-order probabilistic programs

PMAF

Existing

- Bayesian Inference
- Markov Decision Problem

New

- Expectation-Invariant Analysis

EXAMPLE ANALYSES

- Our framework can be instantiated to prove:
- the probability that `b1` and `b2` are both **`false`** at the end of the program = 0.15
- the expected termination time (ticks) = 5/6

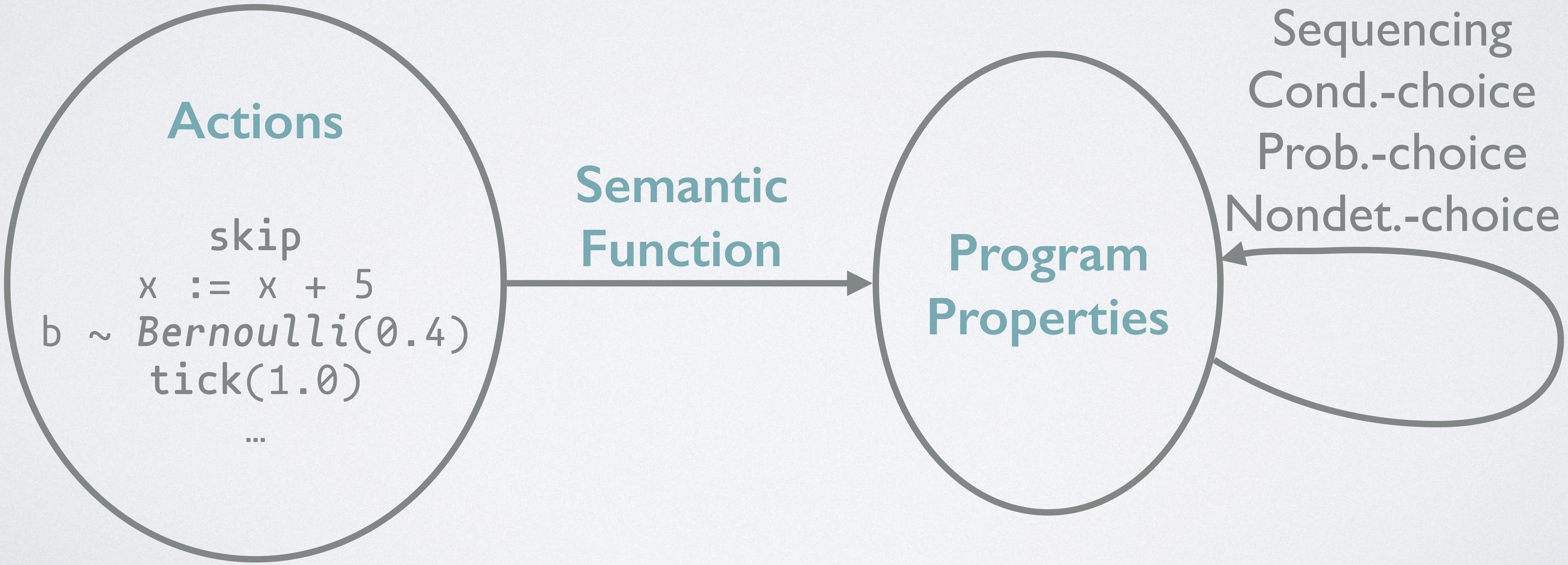
```
b1 ~ Bernoulli(0.5);
b2 ~ Bernoulli(0.7);
while (b1 && b2) do
  if prob(0.6) then
    b1 ~ Bernoulli(0.5)
  else
    b2 ~ Bernoulli(0.7)
  fi;
  tick(1.0)
od;
return (b1, b2)
```

OVERVIEW

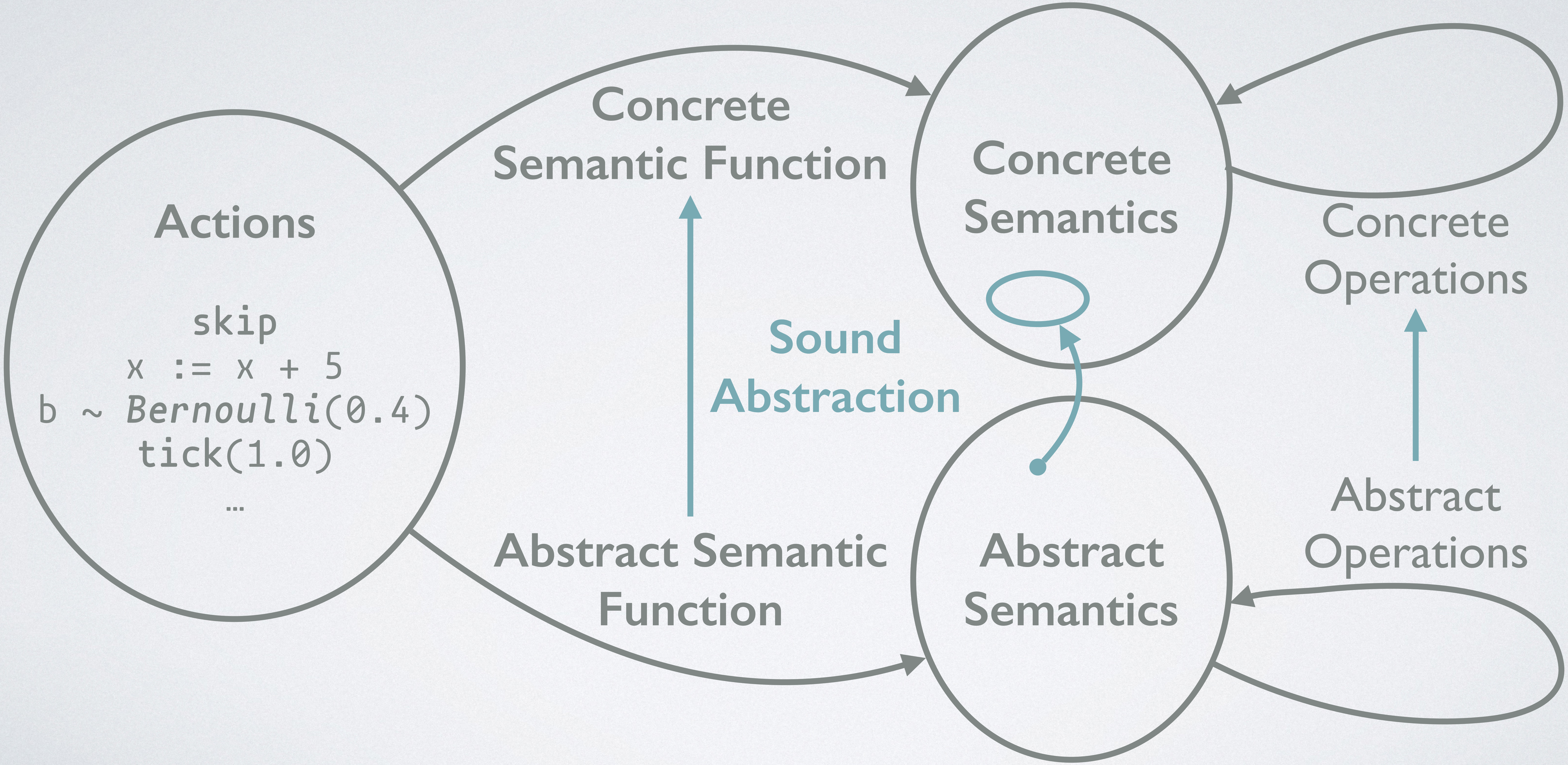
- [x] Motivation
- [ ] The Algebraic Framework
- [ ] Hyper-Graph Analysis
- [ ] Evaluation

THE ALGEBRAIC FRAMEWORK

Any static analysis method performs reasoning in some space of **program properties** and **property operations**



THE ALGEBRAIC FRAMEWORK



THE ALGEBRAIC FRAMEWORK

- Characterize program properties and property operations by **algebraic laws**

THE ALGEBRAIC FRAMEWORK

- Characterize program properties and property operations by **algebraic laws**

$$\langle M, \sqsubseteq, \otimes, {}_{\varphi}\diamond, {}_{p}\oplus, \Cup, \bot, \underline{1} \rangle$$

THE ALGEBRAIC FRAMEWORK

- Characterize program properties and property operations by **algebraic laws**

$$\langle M, \sqsubseteq, \otimes, {}_{\varphi}\!\diamond, {}_{p}\!\oplus, \Cup, \underline{\perp}, \underline{1} \rangle$$

Program properties
and approximation order

THE ALGEBRAIC FRAMEWORK

- Characterize program properties and property operations by **algebraic laws**

$$\langle M, \sqsubseteq, \otimes, {}_{\varphi}\diamond, {}_{p}\oplus, \Cup, \underline{\bot}, \underline{1} \rangle$$

Program properties and approximation order

Sequencing, cond.-choice, prob.-choice, and nondet.-choice

THE ALGEBRAIC FRAMEWORK

- Characterize program properties and property operations by **algebraic laws**

$$\langle M, \sqsubseteq, \otimes, {}_{\varphi}\diamond, {}_{p}\oplus, \Cup, \underline{\bot}, \underline{1}\rangle$$

- $M, \sqsubseteq$: Program properties and approximation order
- $\otimes, {}_{\varphi}\diamond, {}_{p}\oplus, \Cup$: Sequencing, cond.-choice, prob.-choice, and nondet.-choice
- $\underline{\bot}, \underline{1}$: The bottom element and the identity element

THE ALGEBRAIC FRAMEWORK

- Characterize program properties and property operations by **algebraic laws**

$$\langle M, \sqsubseteq, \otimes, {}_{\varphi}\diamond, {}_{p}\oplus, \Cup, \underline{\bot}, \underline{1} \rangle$$

Program properties and approximation order

Sequencing, cond.-choice, prob.-choice, and nondet.-choice

THE ALGEBRAIC FRAMEWORK

- Characterize program properties and property operations by **algebraic laws**

$$\langle M, \sqsubseteq, \otimes, {}_{\varphi}\diamond, {}_{p}\oplus, \Cup, \underline{\bot}, \underline{1} \rangle$$

Program properties and approximation order

Sequencing, cond.-choice, prob.-choice, and nondet.-choice

$$a \;{}_{p}\!\oplus b = b \;{}_{1-p}\!\oplus a$$

$$(a \otimes b) \otimes c = a \otimes (b \otimes c)$$

$$a \otimes \underline{1} = \underline{1} \otimes a = a$$

$$a \Cup a = a$$

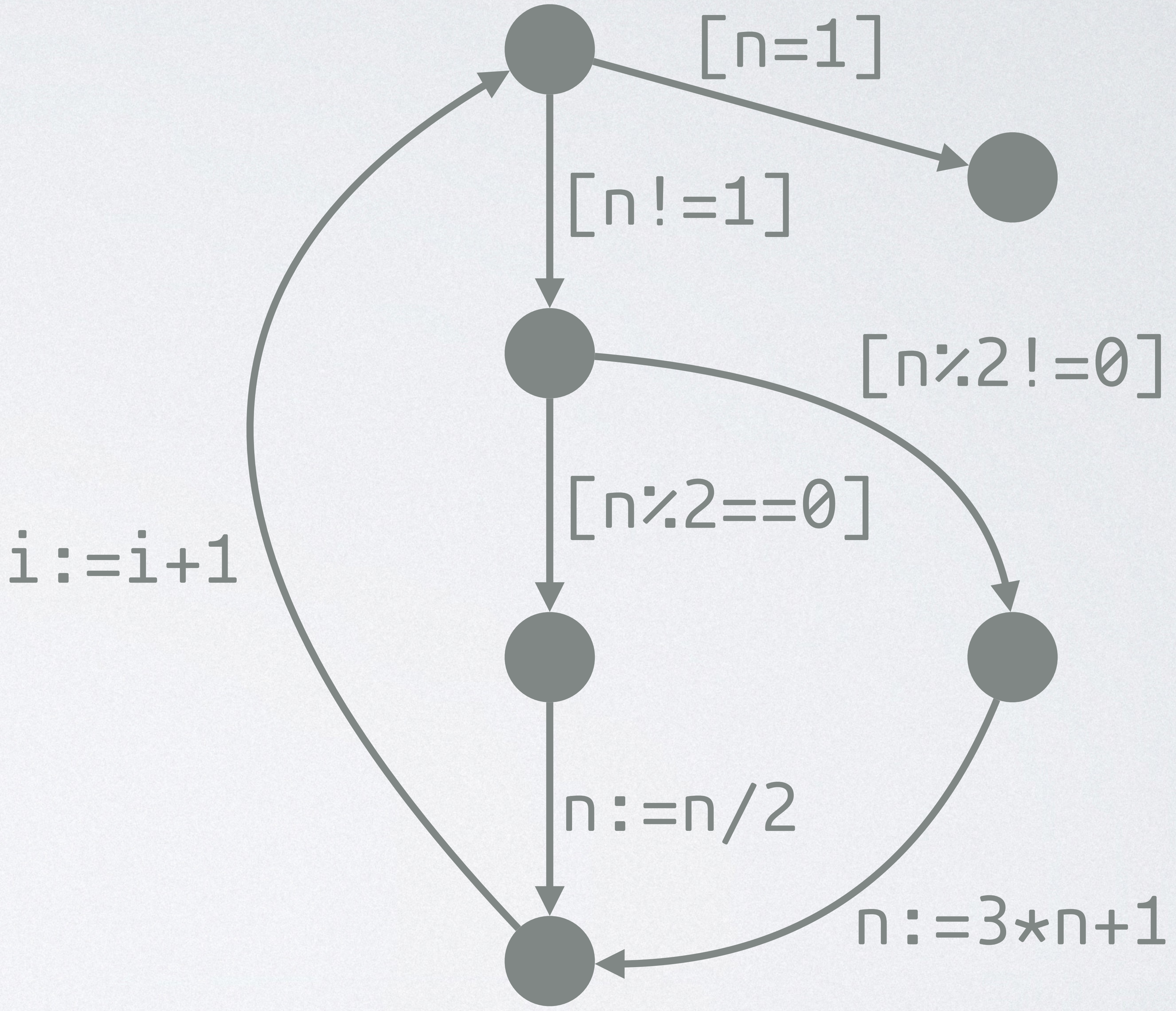
$$\cdots$$

OVERVIEW

- [x] Motivation
- [x] The Algebraic Framework
- [ ] Hyper-Graph Analysis
- [ ] Evaluation

PROGRAM SEMANTICS

- Control-flow graphs
- Reason about **paths**
- Paths are **independent**

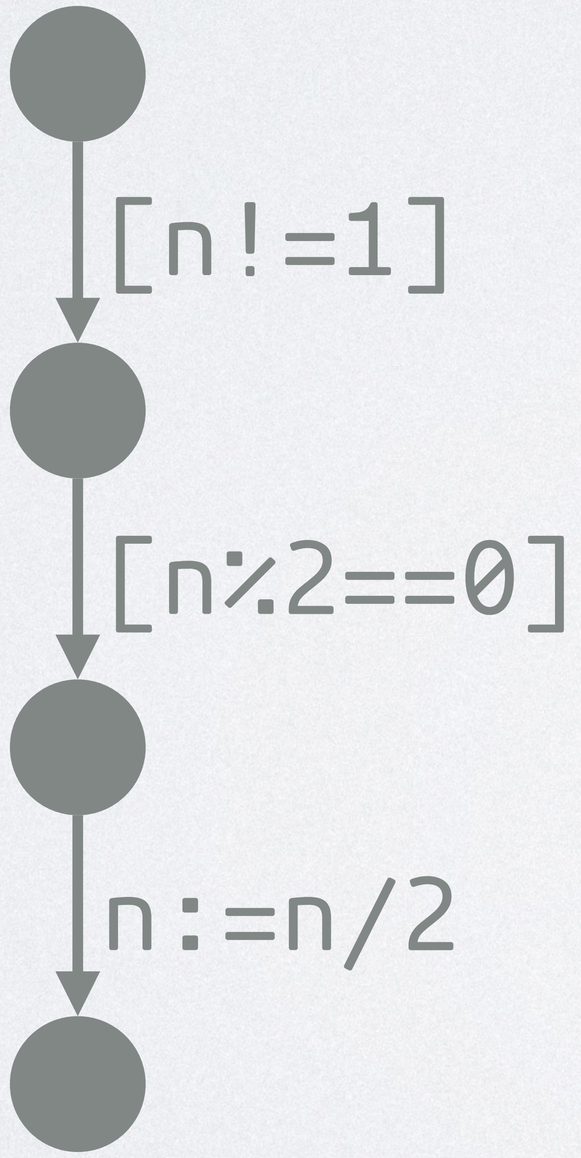


PROGRAM SEMANTICS

- Reason about **distributions over paths**
- Paths are ***not*** **independent**

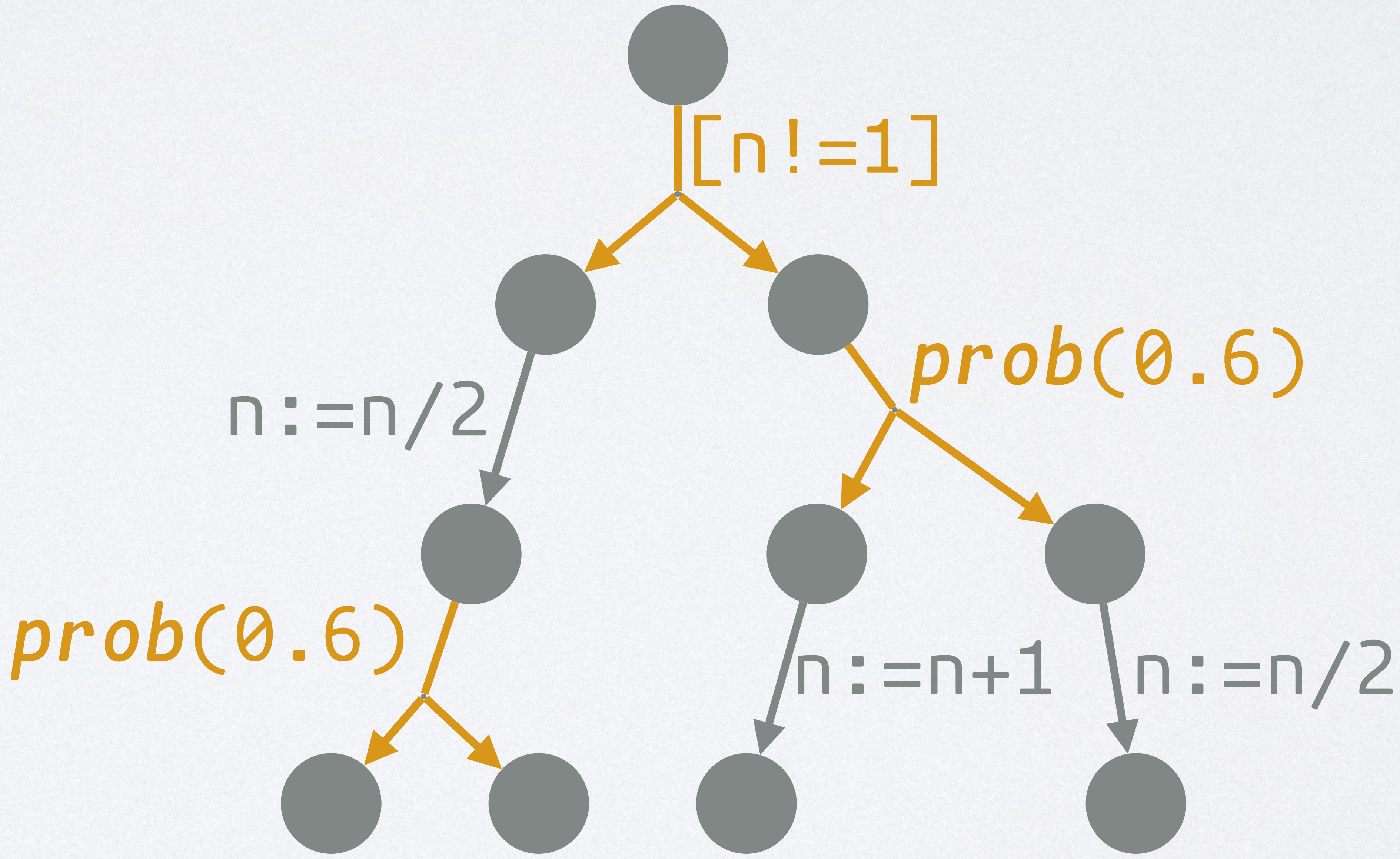
PROGRAM SEMANTICS

- Reason about **distributions over paths**
- Paths are ***not*** **independent**



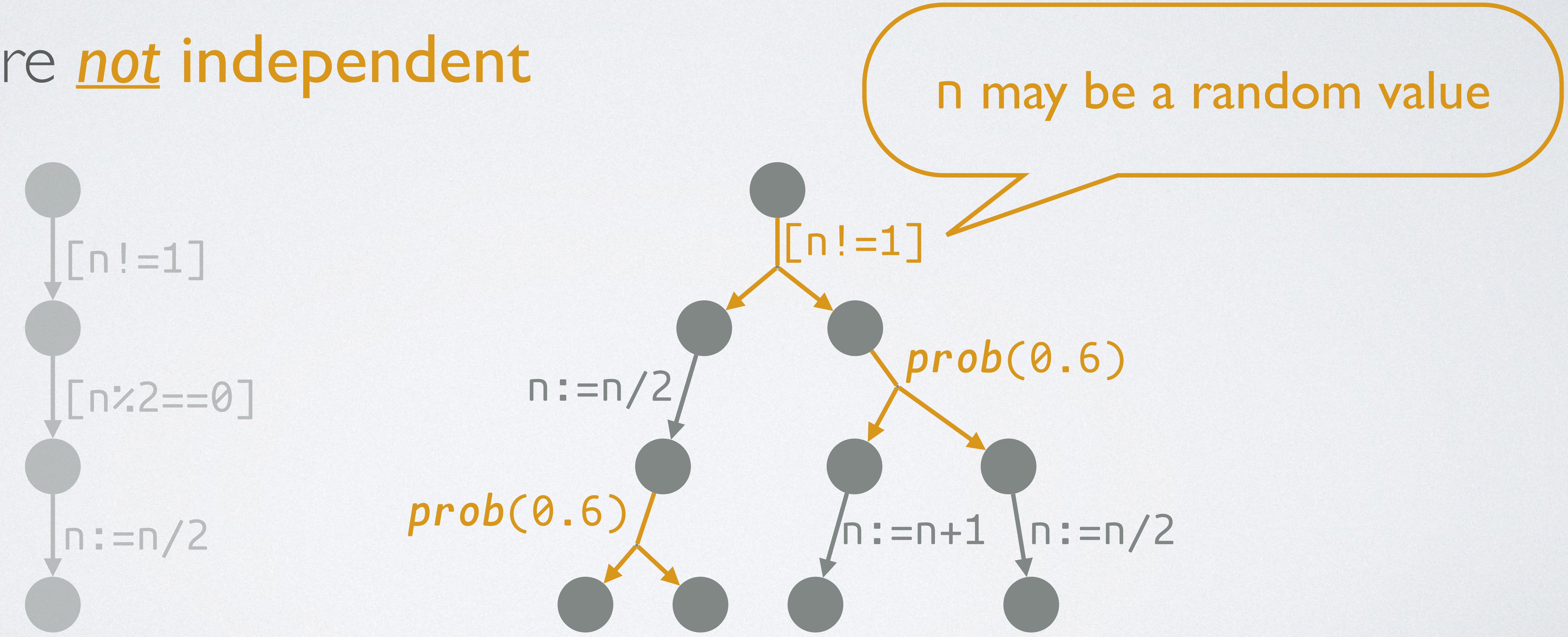
PROGRAM SEMANTICS

- Reason about **distributions over paths**
- Paths are ***not*** **independent**



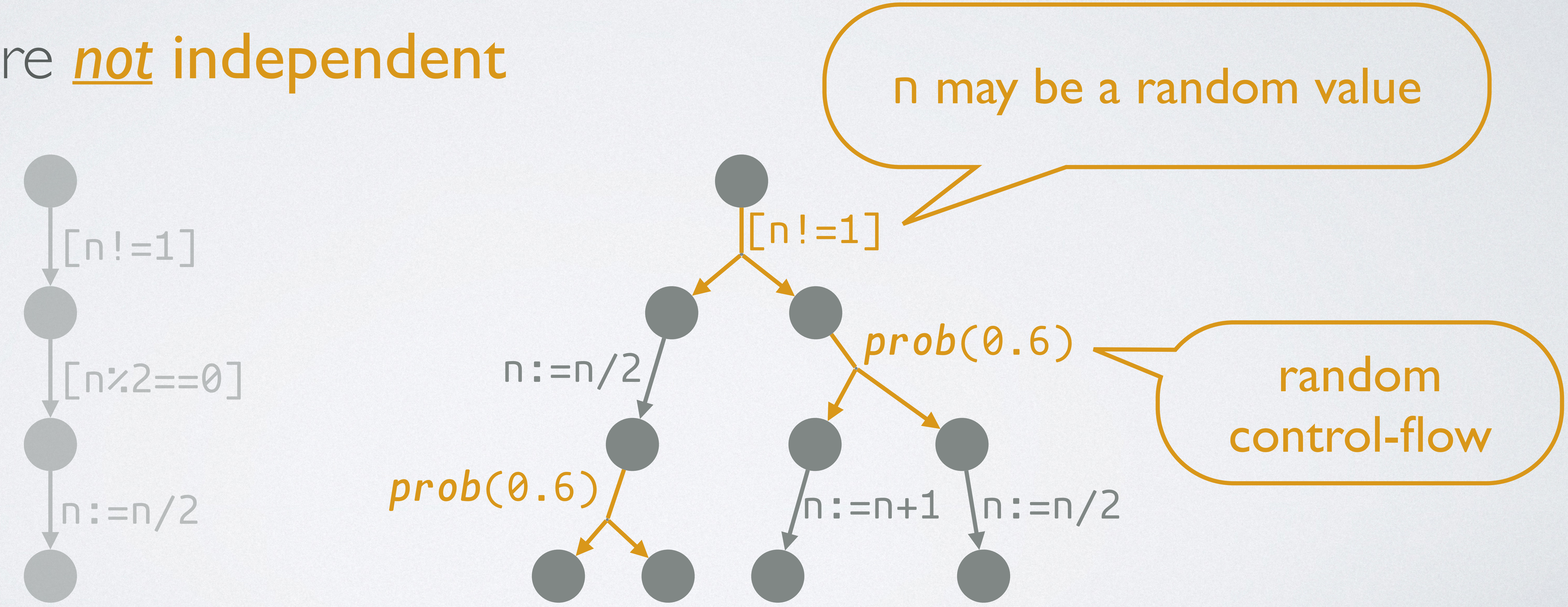
PROGRAM SEMANTICS

- Reason about **distributions over paths**
- Paths are ***not*** **independent**



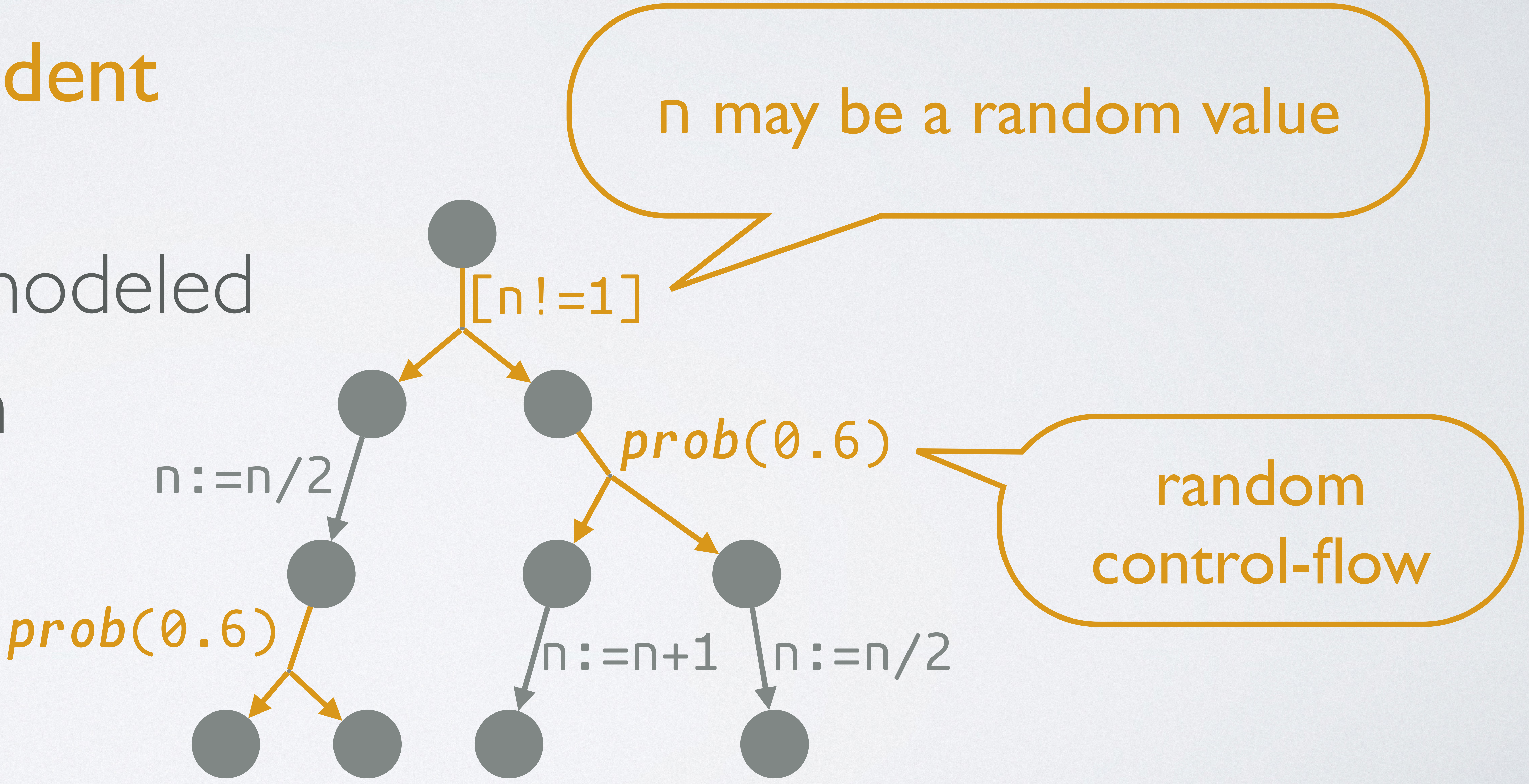
PROGRAM SEMANTICS

- Reason about **distributions over paths**
- Paths are ***not* independent**



PROGRAM SEMANTICS

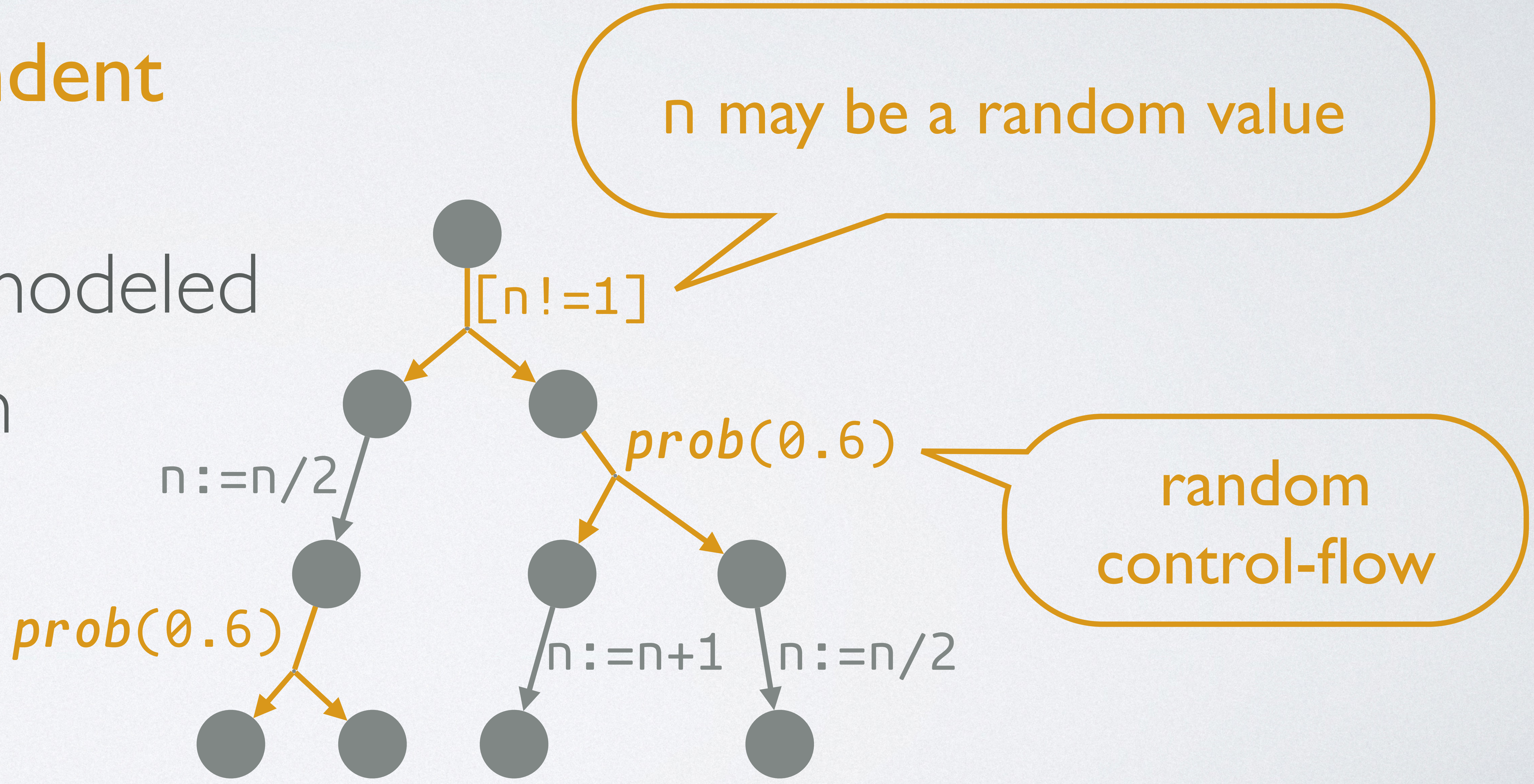
- Reason about **distributions over paths**
- Paths are ***not*** **independent**
- **Nondeterminism** is modeled by **collections** of such distributions



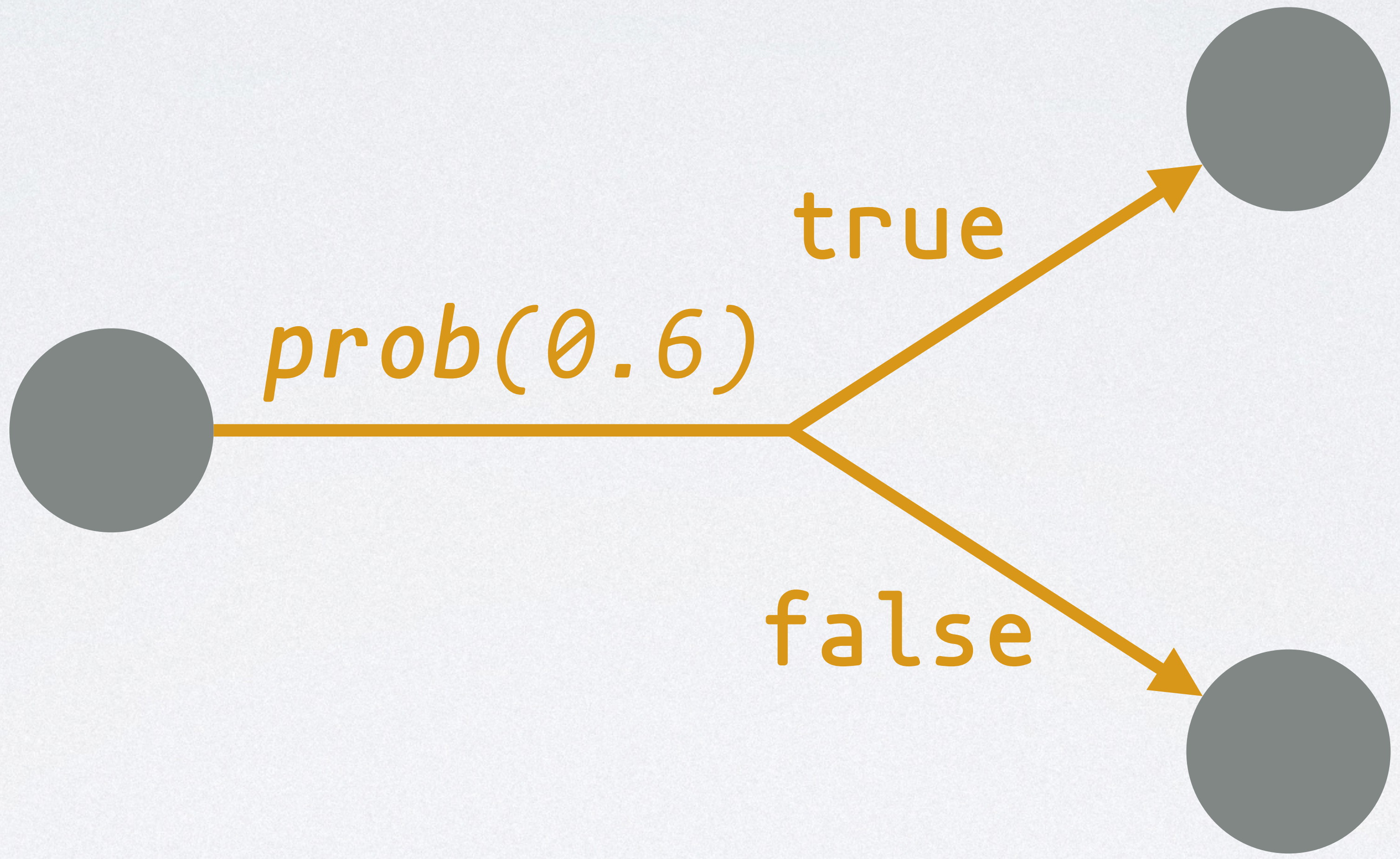
PROGRAM SEMANTICS

- Reason about **distributions over paths**
- Paths are ***not*** **independent**
- **Nondeterminism** is modeled by **collections** of such distributions

Resolve nondeterminism first!

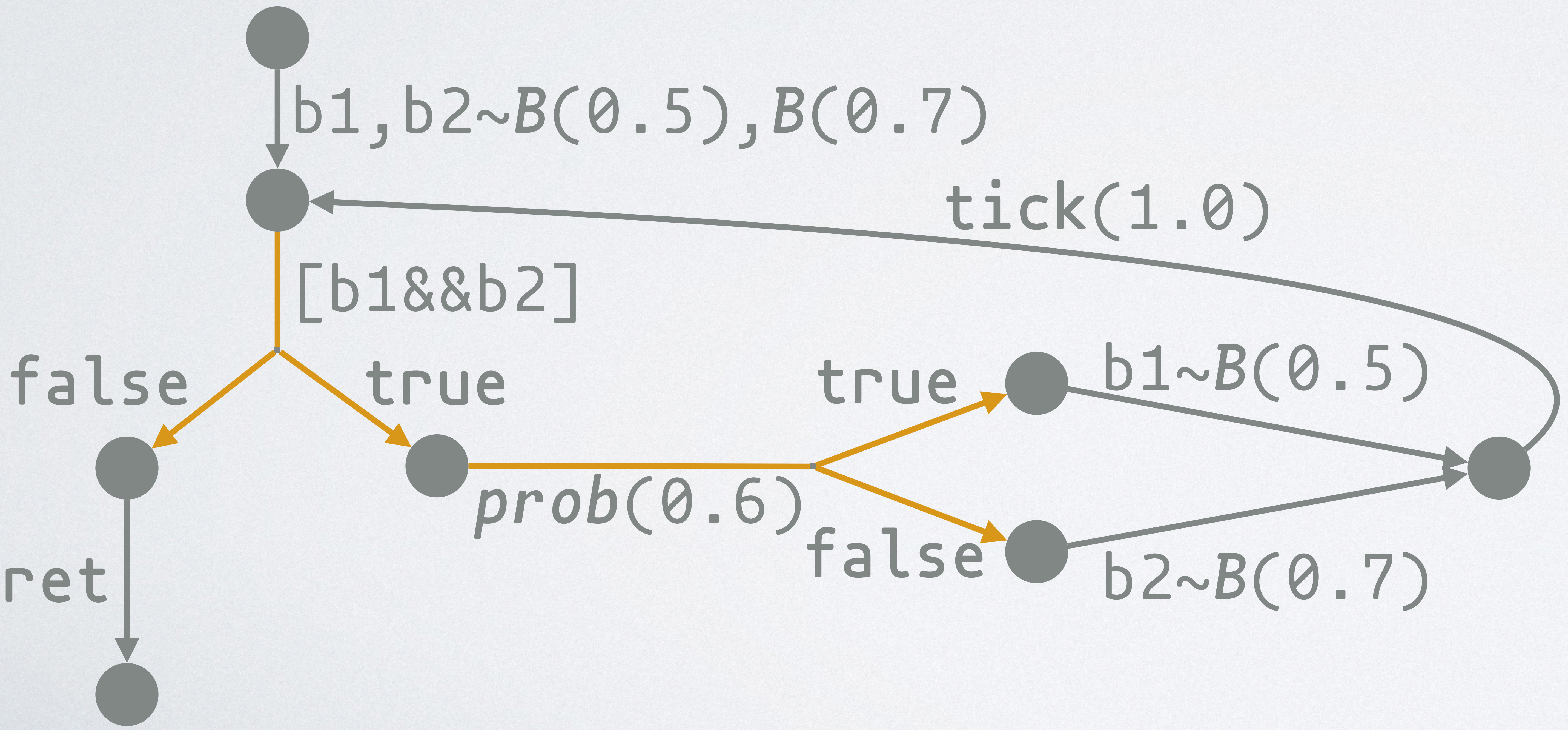


PROGRAM SEMANTICS



PROGRAM SEMANTICS

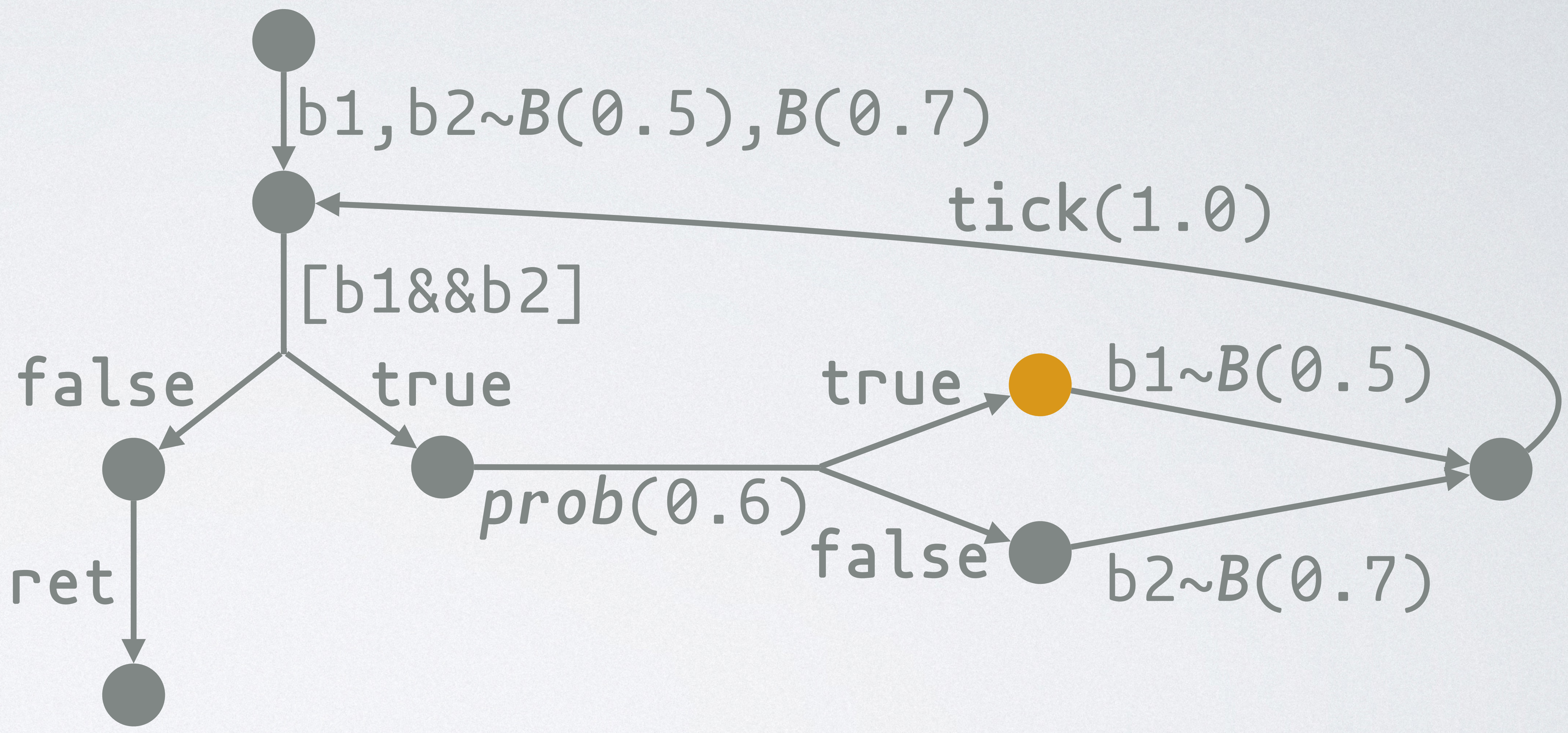
- Control-flow **hyper**-graphs
- Branching are **hyper**-edges




```
b1 ~ Bernoulli(0.5);
b2 ~ Bernoulli(0.7);
while (b1 && b2) do
  if prob(0.6) then
    b1 ~ Bernoulli(0.5)
  else
    b2 ~ Bernoulli(0.7)
  fi;
  tick(1.0)
od;
return (b1, b2)
```

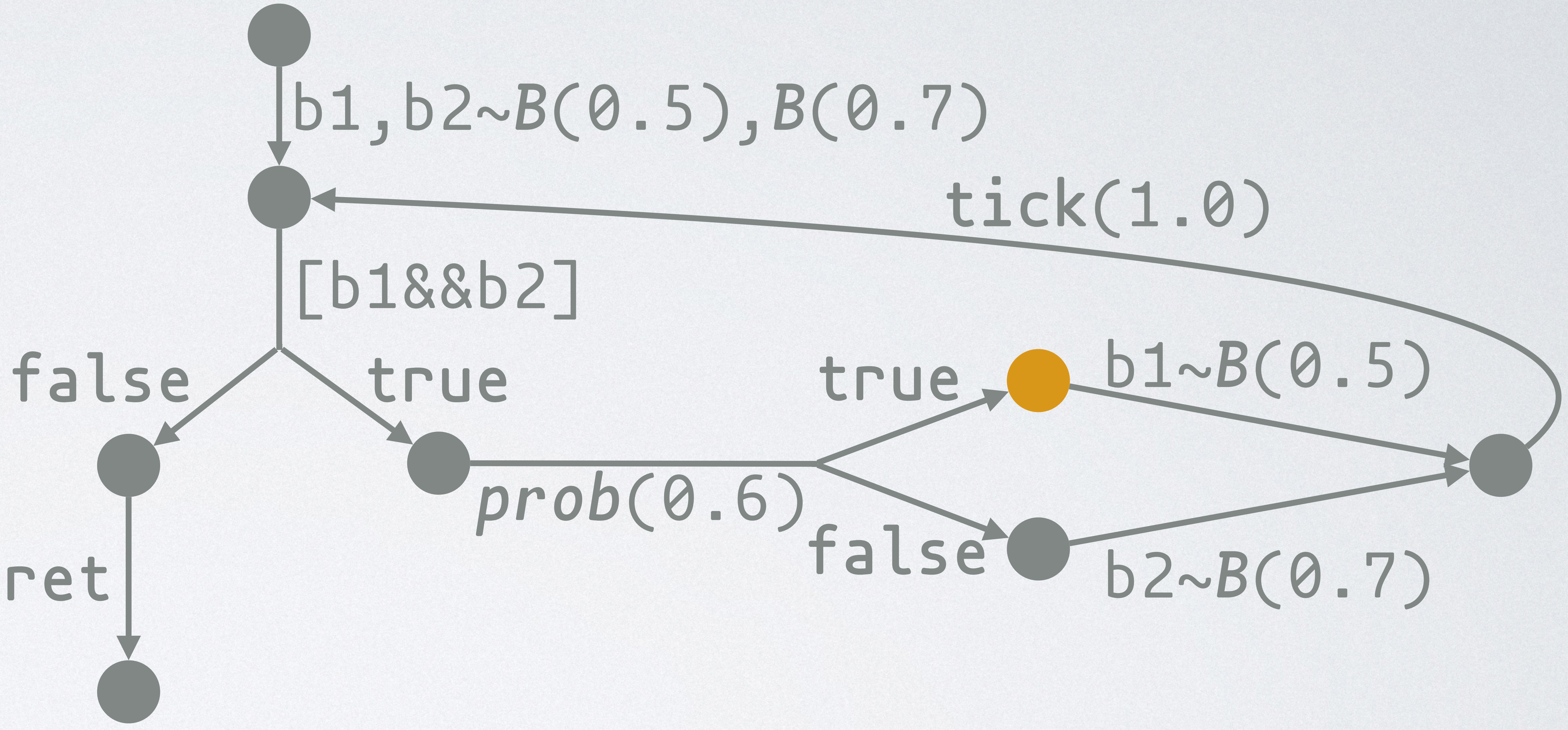
HYPER-GRAPH ANALYSIS

- **Forward assertions**
- The semantics of a node is a summary of computation that **continues from** the node



HYPER-GRAPH ANALYSIS

- **Forward assertions**
- The semantics of a node is a summary of computation that **continues from** the node

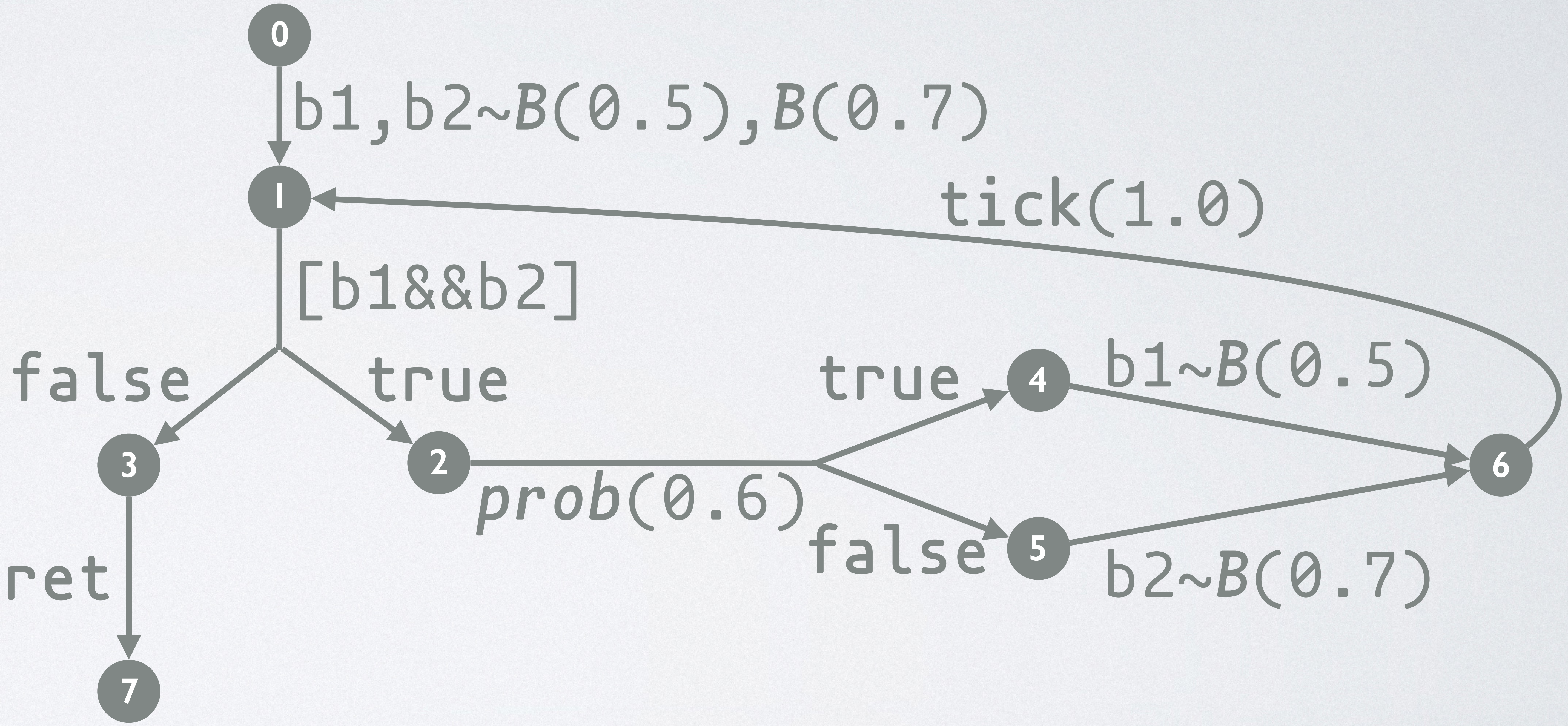


E.g. the semantics of the **node** is

$$\lambda(\mathsf{b1},\mathsf{b2})\,.\,\textbf{if}\ \mathsf{b2}\ \textbf{then}\ \frac{1}{7}[\mathsf{b1}'=\mathsf{T},\mathsf{b2}'=\mathsf{F}]+\frac{6}{7}[\mathsf{b1}'=\mathsf{F},\mathsf{b2}'=\mathsf{T}]$$
$$\textbf{else}\ \frac{1}{2}[\mathsf{b1}'=\mathsf{T},\mathsf{b2}'=\mathsf{F}]+\frac{1}{2}[\mathsf{b1}'=\mathsf{F},\mathsf{b2}'=\mathsf{F}]$$

HYPER-GRAPH ANALYSIS

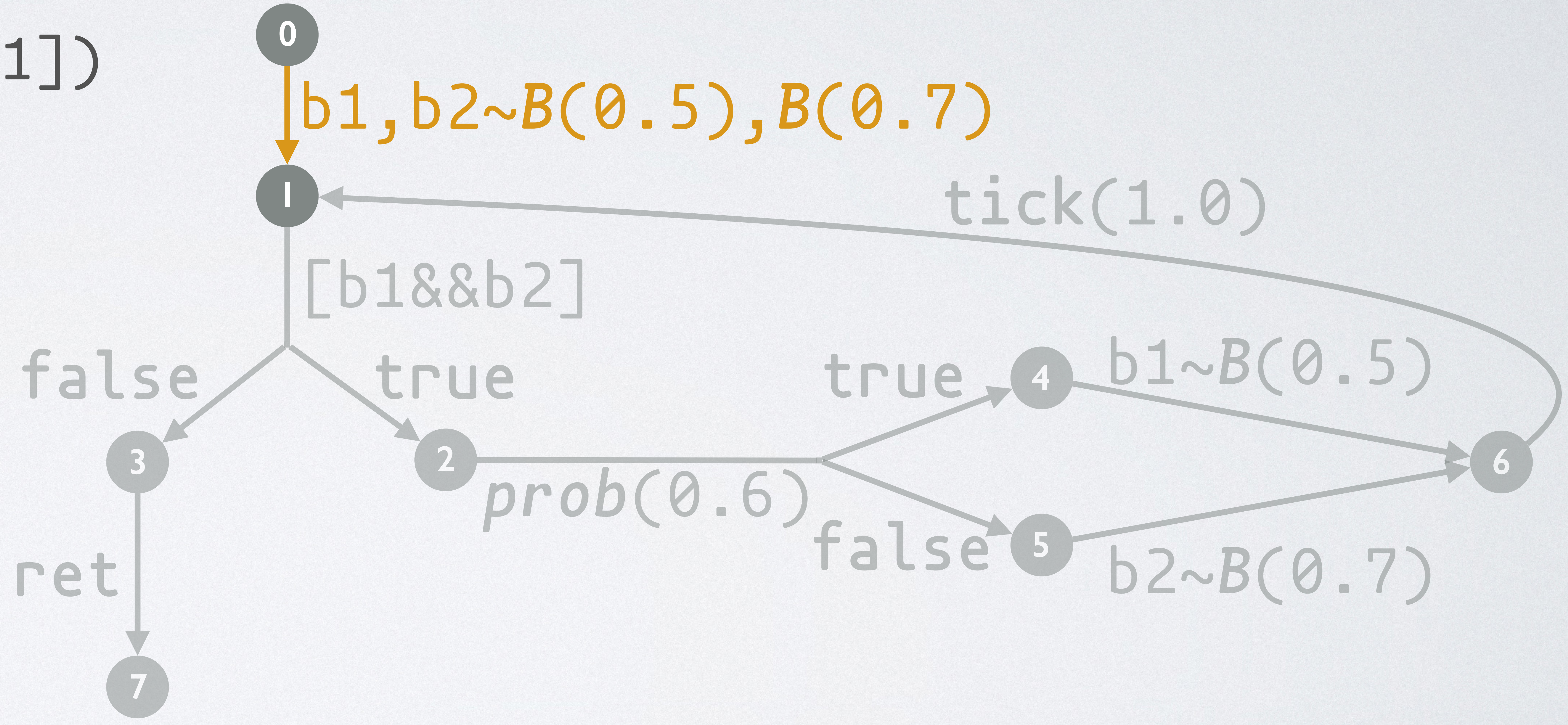
- The hyper-graph analysis is formulated by an equation system



HYPER-GRAPH ANALYSIS

- The hyper-graph analysis is formulated by an equation system

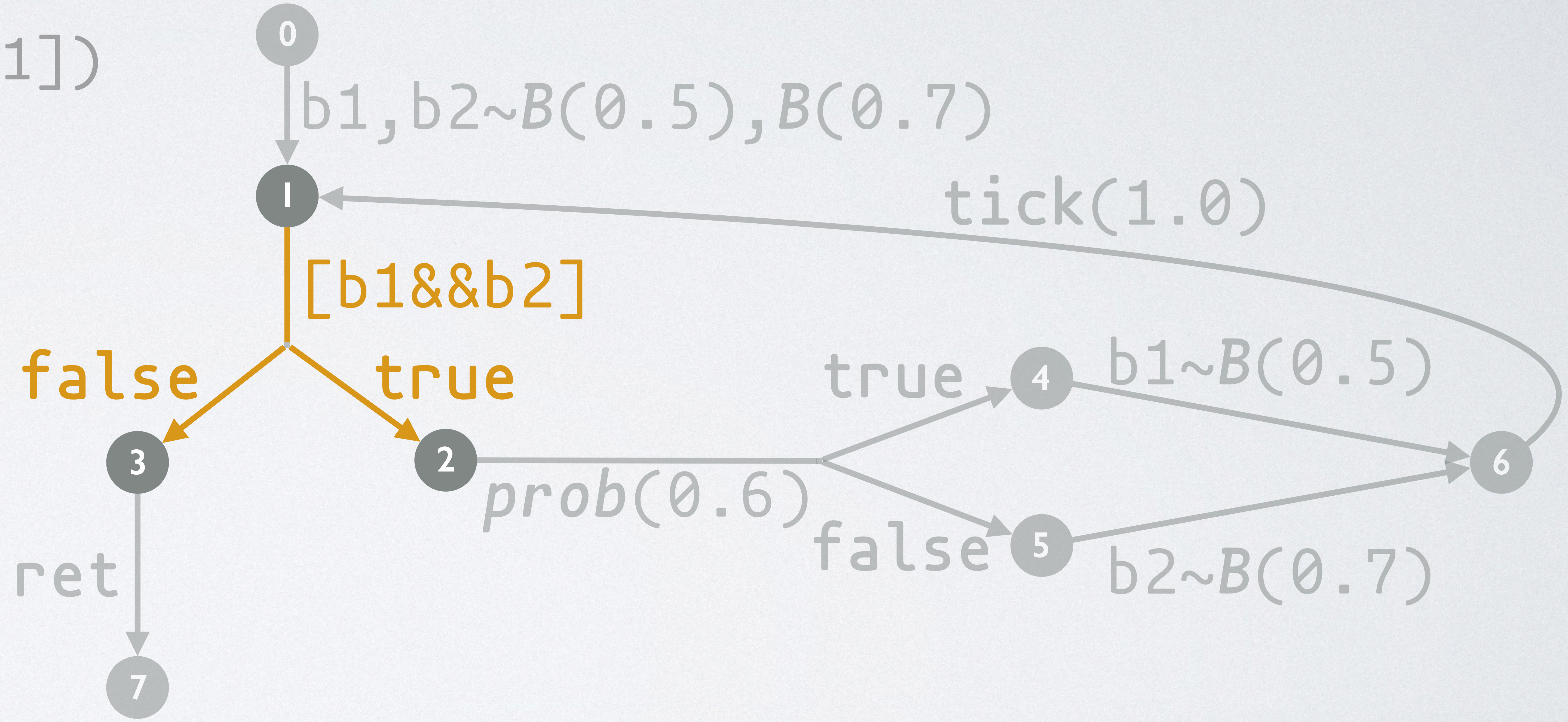
```
S[0]=seq[b1,b2~B(0.5),B(0.7)](S[1])
```



HYPER-GRAPH ANALYSIS

- The hyper-graph analysis is formulated by an equation system

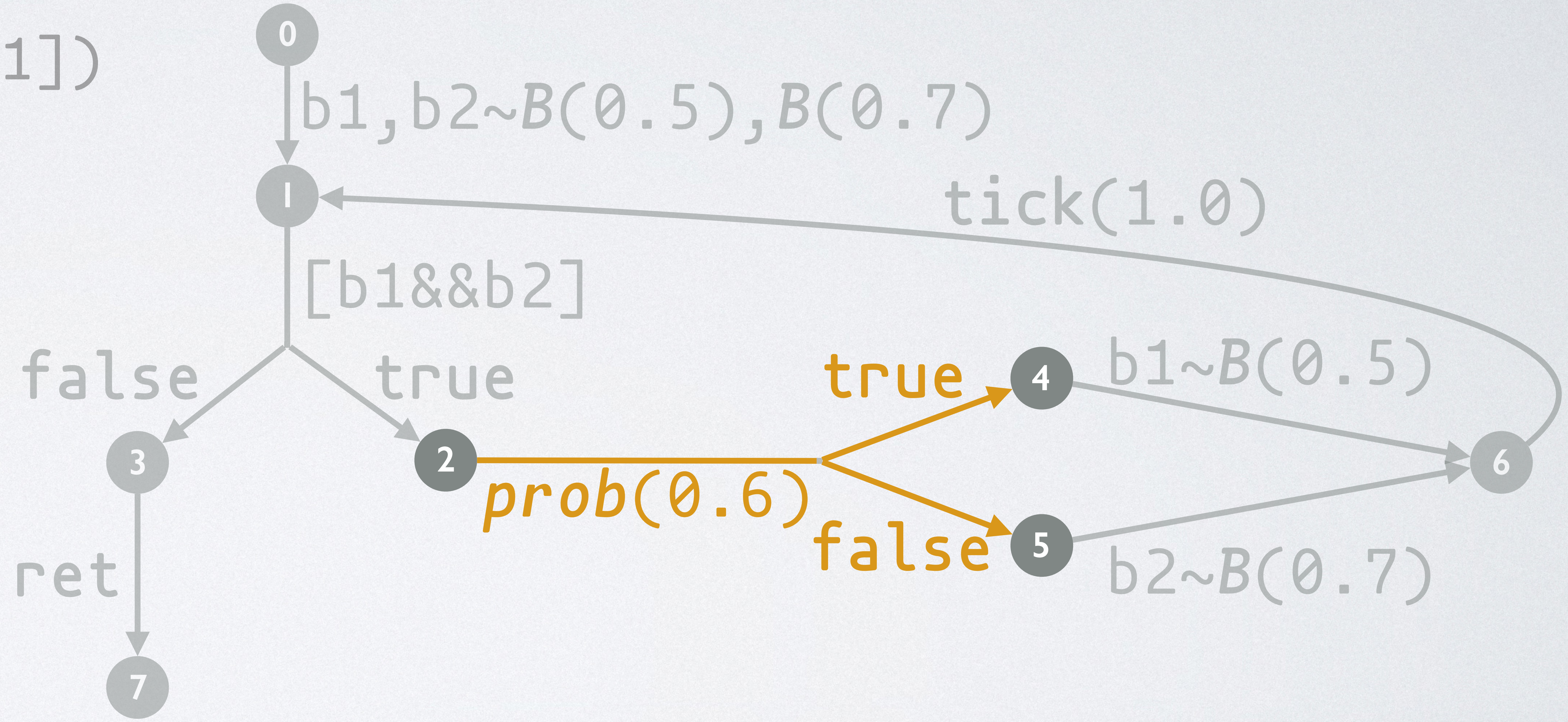
```
S[0]=seq[b1,b2~B(0.5),B(0.7)](S[1])
S[1]=cond[b1&&b2](S[2],S[3])
```



HYPER-GRAPH ANALYSIS

- The hyper-graph analysis is formulated by an equation system

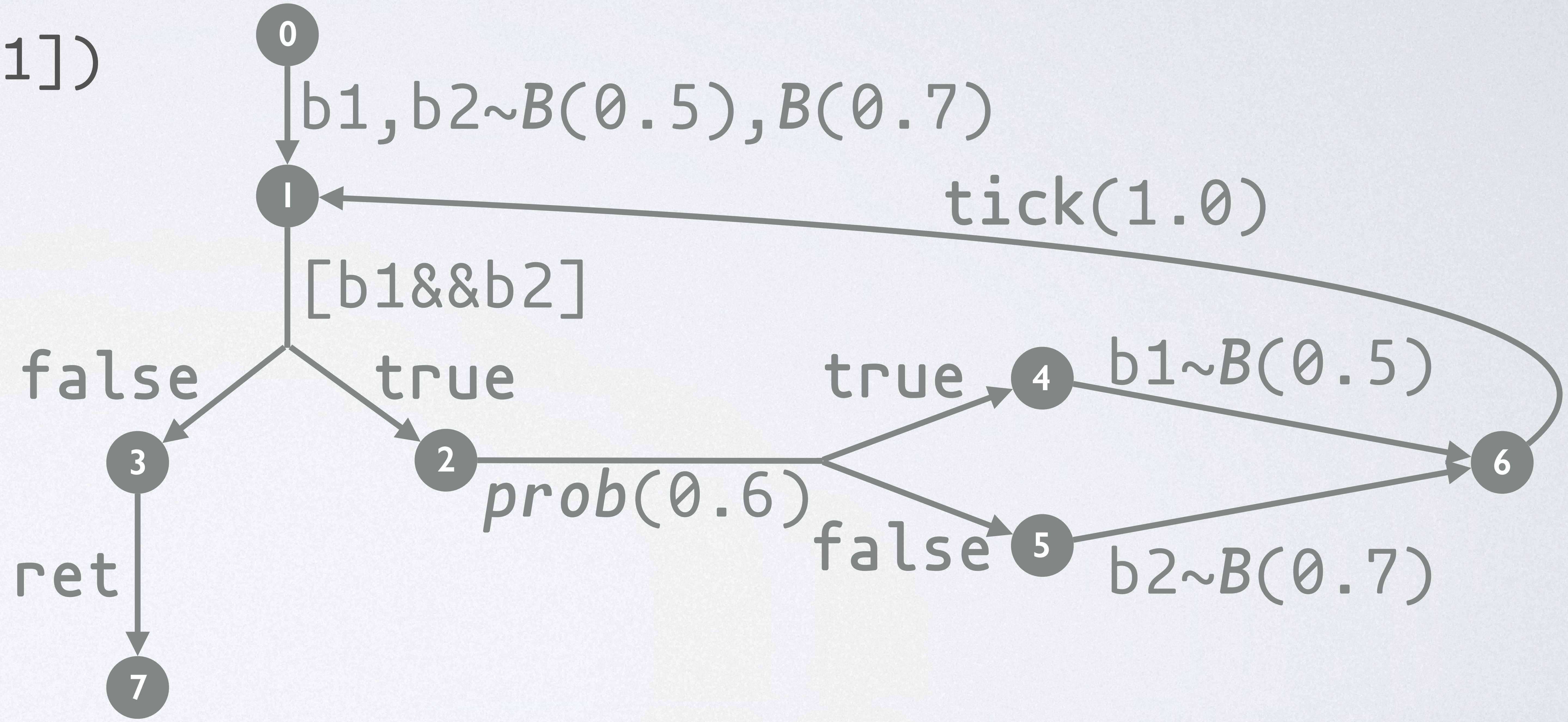
```
S[0]=seq[b1,b2~B(0.5),B(0.7)](S[1])
S[1]=cond[b1&&b2](S[2],S[3])
S[2]=prob[0.6](S[4],S[5])
```



HYPER-GRAPH ANALYSIS

- The hyper-graph analysis is formulated by an equation system

```
S[0]=seq[b1,b2~B(0.5),B(0.7)](S[1])
S[1]=cond[b1&&b2](S[2],S[3])
S[2]=prob[0.6](S[4],S[5])
```

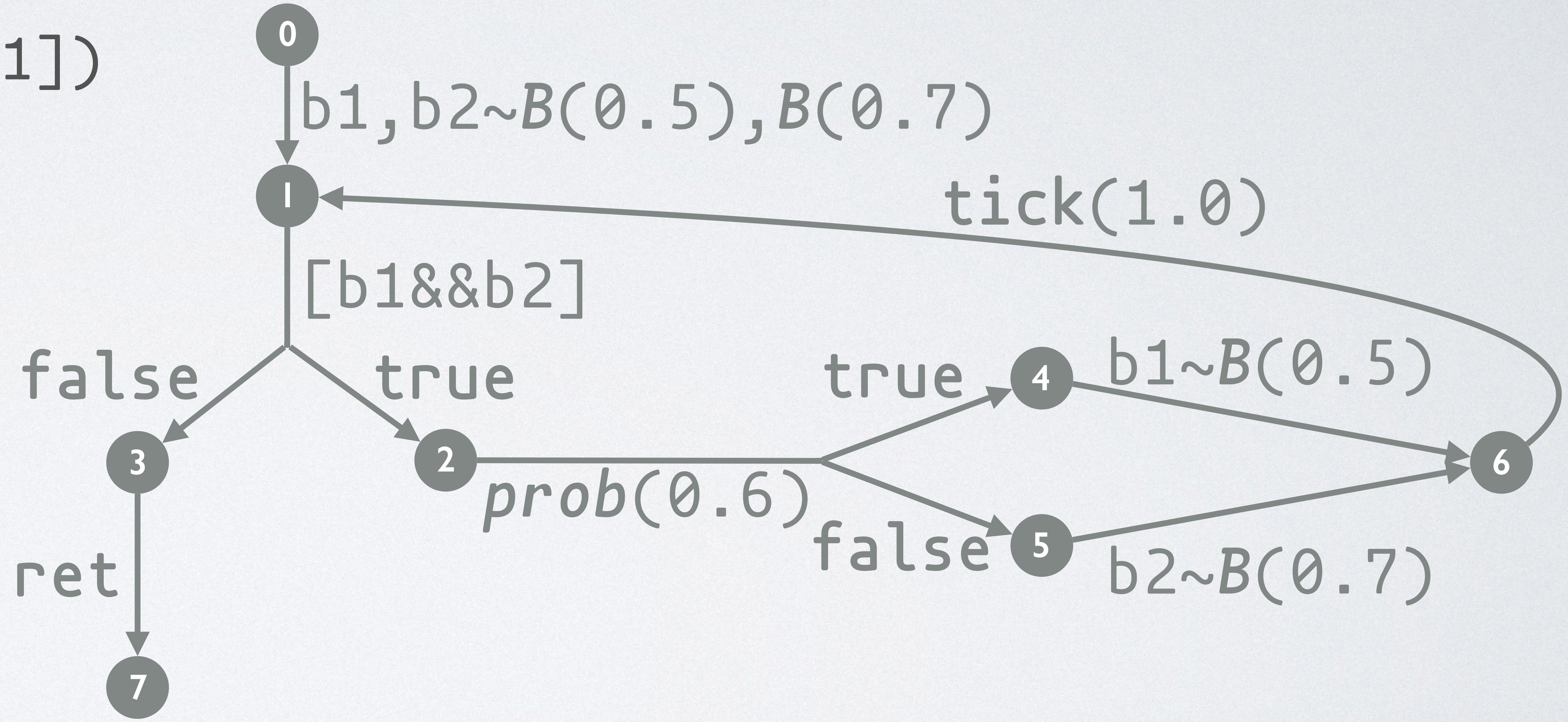


HYPER-GRAPH ANALYSIS

- The hyper-graph analysis is formulated by an equation system

```
S[0]=seq[b1,b2~B(0.5),B(0.7)](S[1])
S[1]=cond[b1&&b2](S[2],S[3])
S[2]=prob[0.6](S[4],S[5])
```

Use the semantic **algebra** to interpret ***seq***, ***cond***, ***prob***



HYPER-GRAPH ANALYSIS

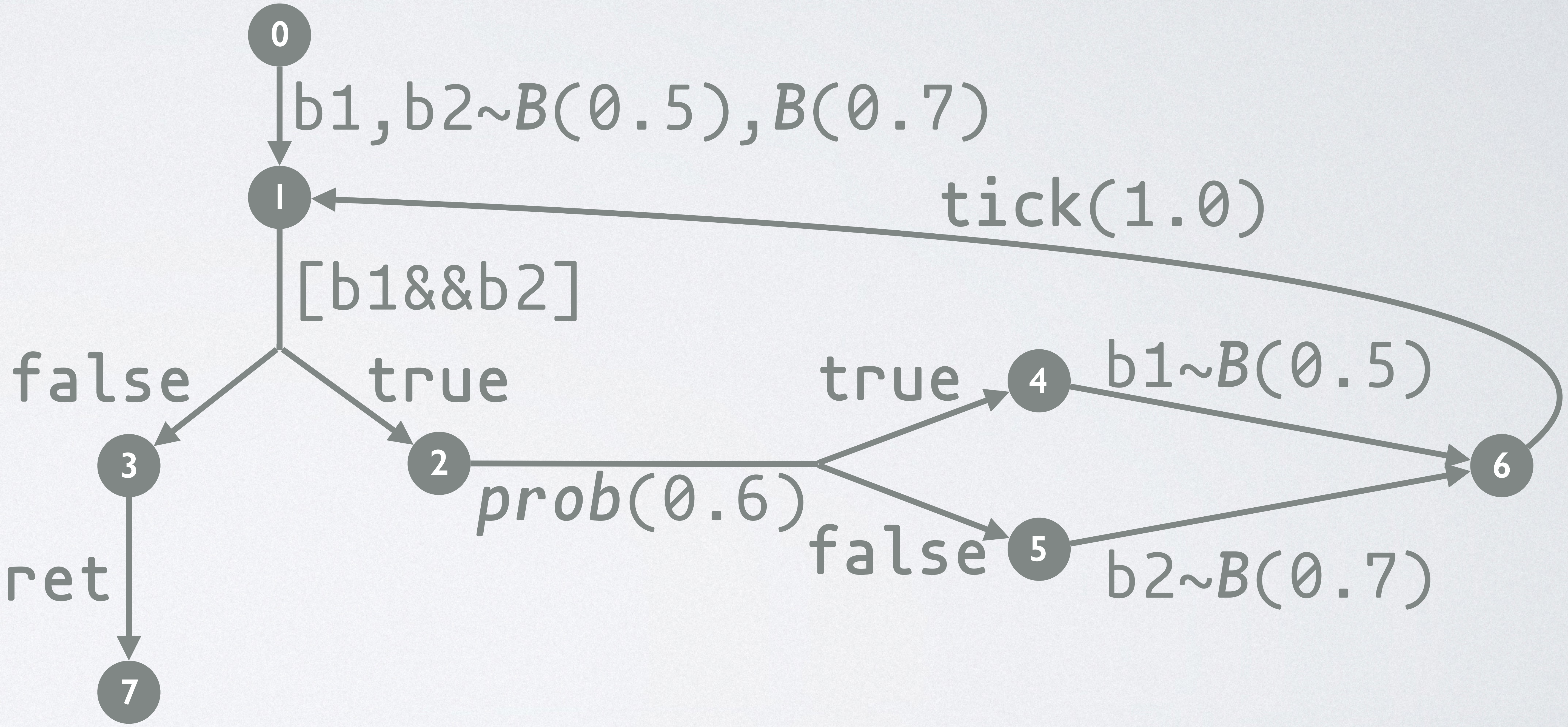
- The hyper-graph analysis is formulated by an equation system

$S[0] = [\mathrm{b1}, \mathrm{b2} \sim \mathrm{B}(0.5), \mathrm{B}(0.7)] \otimes S[1]$

$S[1] = S[2]_{\mathrm{b1\&\&b2}} \diamond S[3]$

$S[2] = S[4]_{0.6} \oplus S[5]$

Use the semantic **algebra** to interpret ***seq***, ***cond***, ***prob***



HYPER-GRAPH ANALYSIS

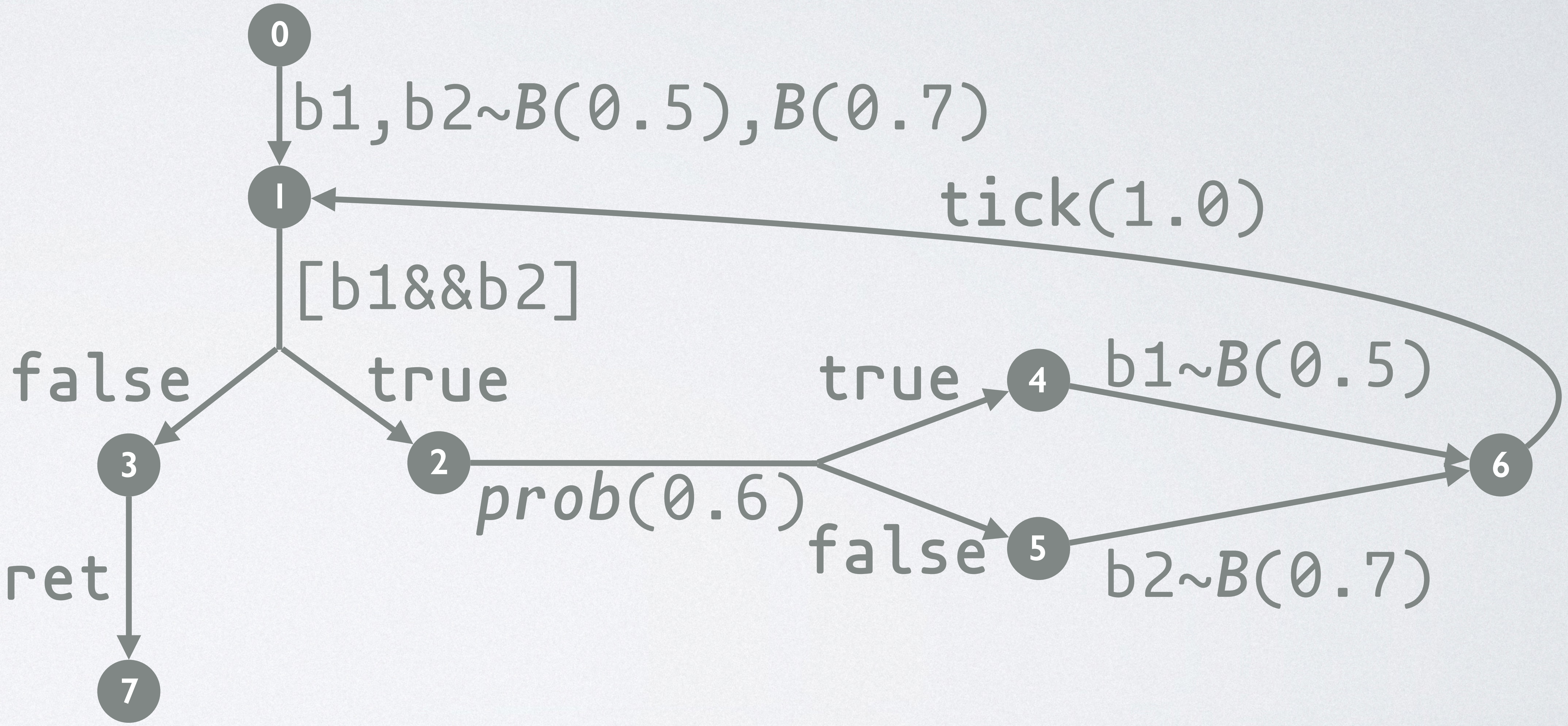
- The hyper-graph analysis is formulated by an equation system

$S[0] = [\mathsf{b1}, \mathsf{b2} \sim \mathsf{B}(0.5), \mathsf{B}(0.7)] \otimes S[1]$

$S[1] = S[2]_{\mathsf{b1\&\&b2}} \diamond S[3]$

$S[2] = S[4]_{0.6} \oplus S[5]$

If using **abstract** semantics, we obtain an equation system for **static analysis**



OVERVIEW

- [x] Motivation
- [x] The Algebraic Framework
- [x] Hyper-Graph Analysis
- [ ] Evaluation

INSTANTIATIONS

PMAF

INSTANTIATIONS

Existing

PMAF

- Bayesian Inference
- Markov Decision Problem

INSTANTIATIONS

PMAF

Existing

- Bayesian Inference
- Markov Decision Problem

New

Interprocedural analyzers!

INSTANTIATIONS

Existing

PMAF

- Bayesian Inference
- Markov Decision Problem

INSTANTIATIONS

PMAF

Existing

- Bayesian Inference
- Markov Decision Problem

New

- Expectation-Invariant Analysis

INSTANTIATIONS

PMAF

Existing

- Bayesian Inference
- Markov Decision Problem

New

- Expectation-Invariant Analysis

Prove invariants among **initial values** and **expected final values**

PROBABILISTIC MODEL ANALYSES

- Benchmark collected from PReMo[1]
- Achieve the same precision

Bayesian Inference (Table 2)

Program	#loc	time (sec)
compare	17	2.22
dice	12	0.02
eg1	10	0.02
eg2	16	0.01
recursive	14	0.01

Markov Decision Problem (Table 2)

Program	#loc	time (sec)
binary10	184	0.03
loop	10	0.03
quicksort7	109	0.03
recursive	13	0.03
student	43	0.03

[1] D. Wojtczak and K. Etessami. PReMo - Probabilistic Recursive Models analyzer. Available at groups.inf.ed.ac.uk/premo/.

EXPECTATION-INVARIANT ANALYSIS

- Benchmark collected from the literature[1,2] and also handcrafted by us
- Derive expectation invariants as least as precise as them in most case

Expectation-Invariant Analysis (Table 1)

Program	#loc	time (sec)	Expectation Invariants
binom-update	14	0.06	E[4x'-n']=4x-n, E[x']<=x+1/4
eg	8	0.89	E[x'+y']=x+y+4, E[z']=1/4z+3/4
recursive	13	0.37	E[x']=x+9
mot-ex	16	0.06	E[2x'-y']=2x-y, E[4x'-3c']=4x-3c, E[x']<=x+3/4

[1] A. Chakarov and S. Sankaranarayanan. Expectation Invariants for Probabilistic Loops as Fixed Points. In *SAS'14*.

[2] J.-P. Katoen, A. K. McIver, L. A. Meinicke, and C. C. Morgan. Linear-Invariant Generation for Probabilistic Programs. In *SAS'10*.

SUMMARY

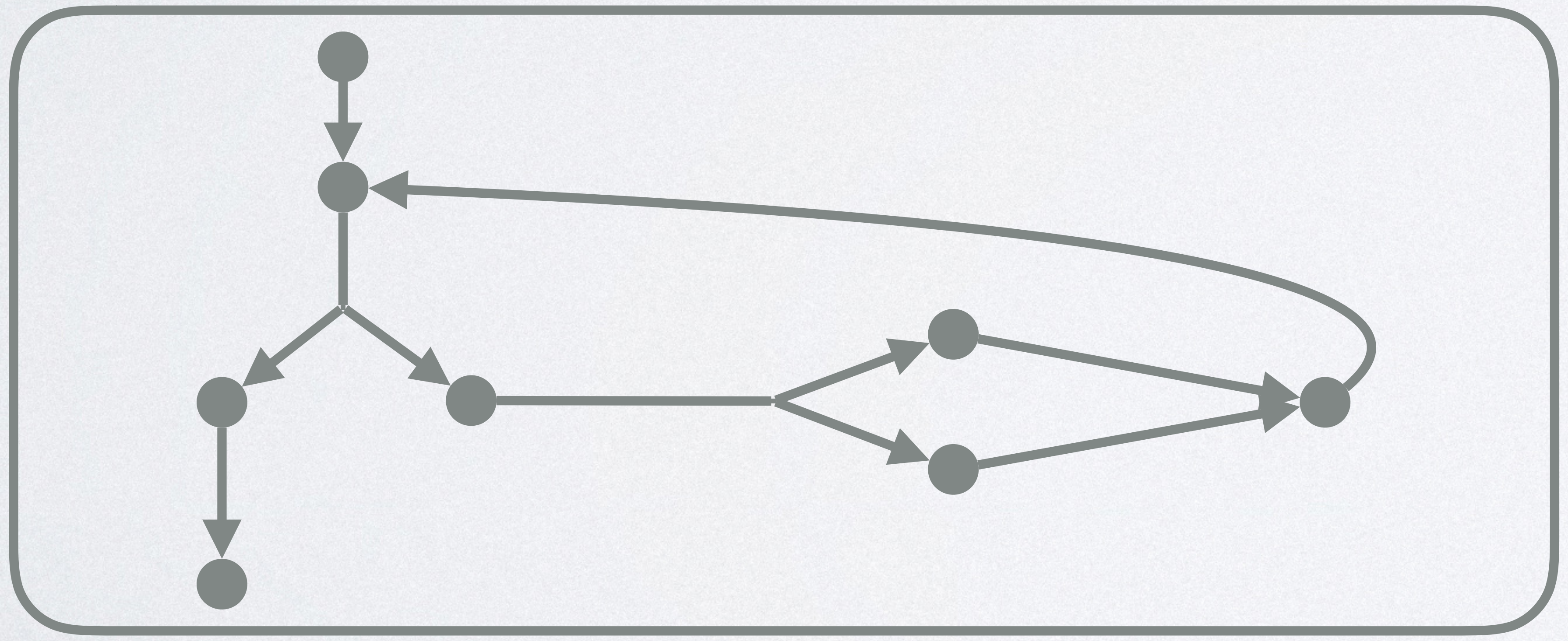
PMAF

Design → Prove → Implement

SUMMARY

PMAF

Design → **Prove** → **Implement**

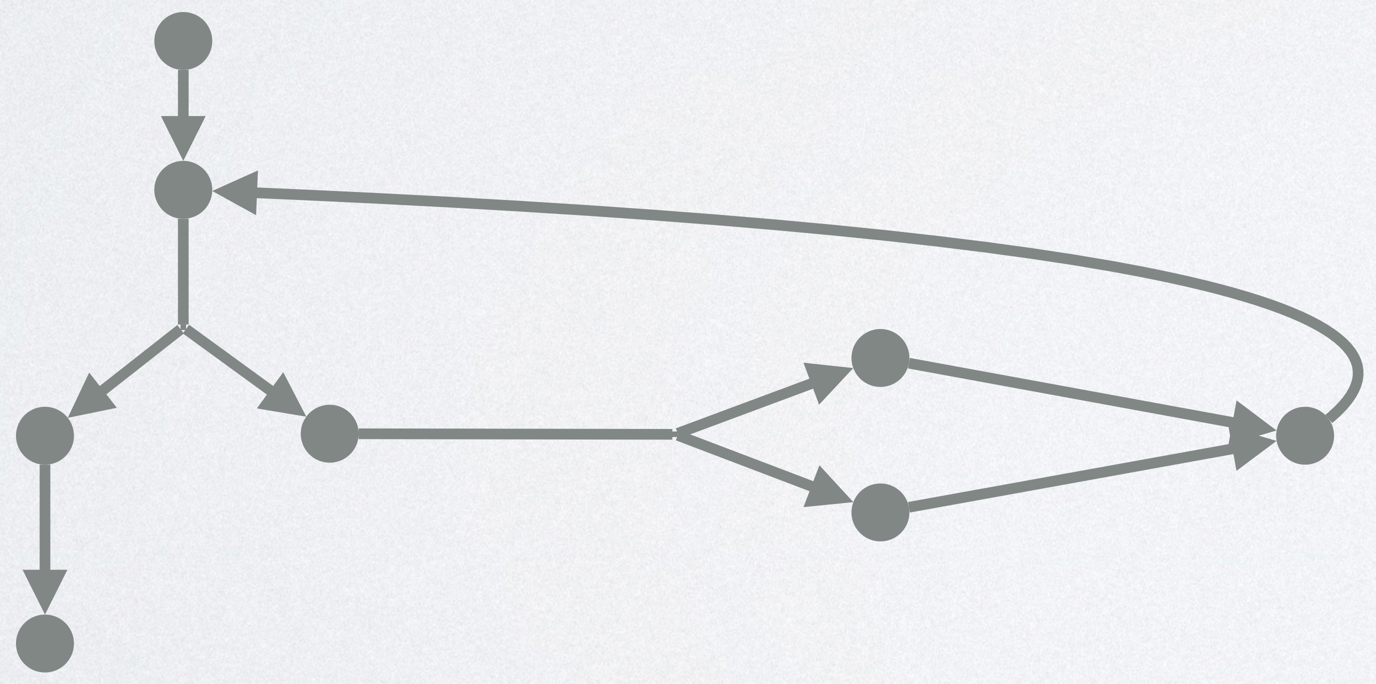


Hyper-Graph Semantics

SUMMARY

PMAF

Design → **Prove** → **Implement**



Hyper-Graph Semantics

- Bayesian Inference
- Markov Decision Problem
- Expectation-Invariant Analysis

Instantiations

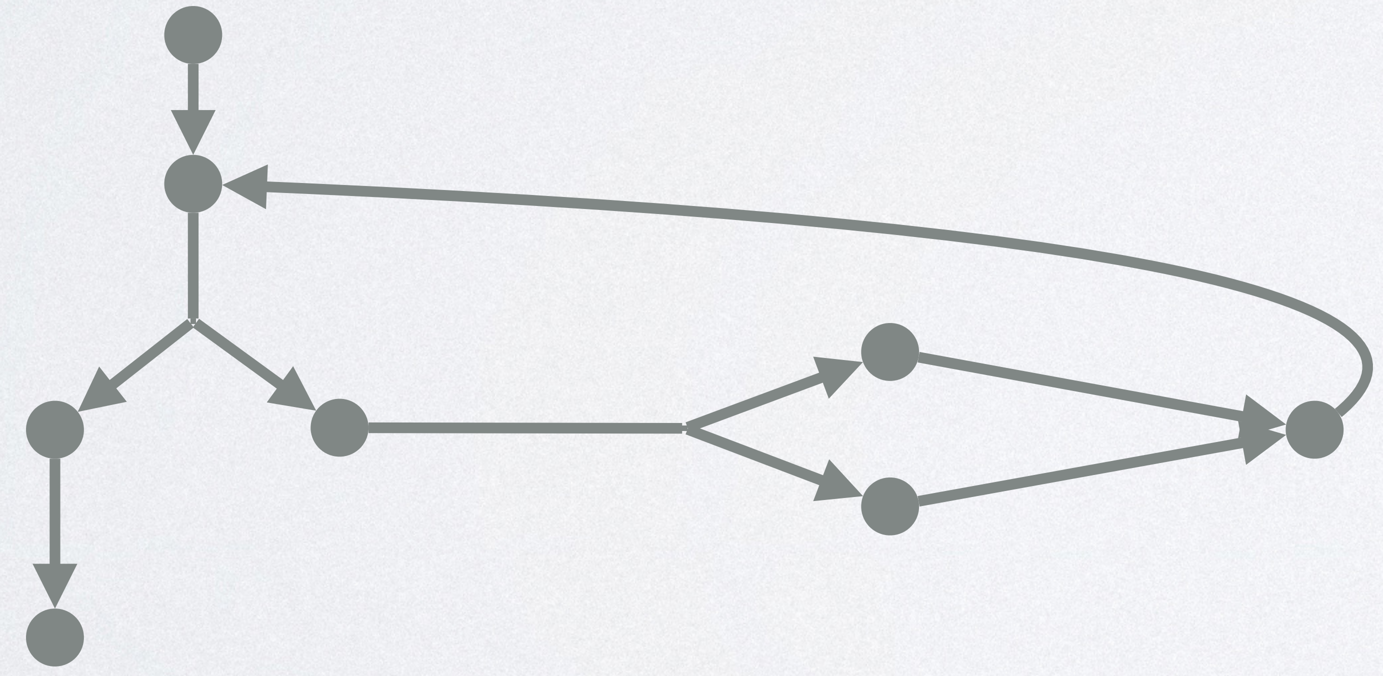
SUMMARY

Limitations:

- Only first-order programs
- No function pointers
- Not Galois connections

PMAF

Design → **Prove** → **Implement**



Hyper-Graph Semantics

- Bayesian Inference
- Markov Decision Problem
- Expectation-Invariant Analysis

Instantiations

SUMMARY

Limitations:

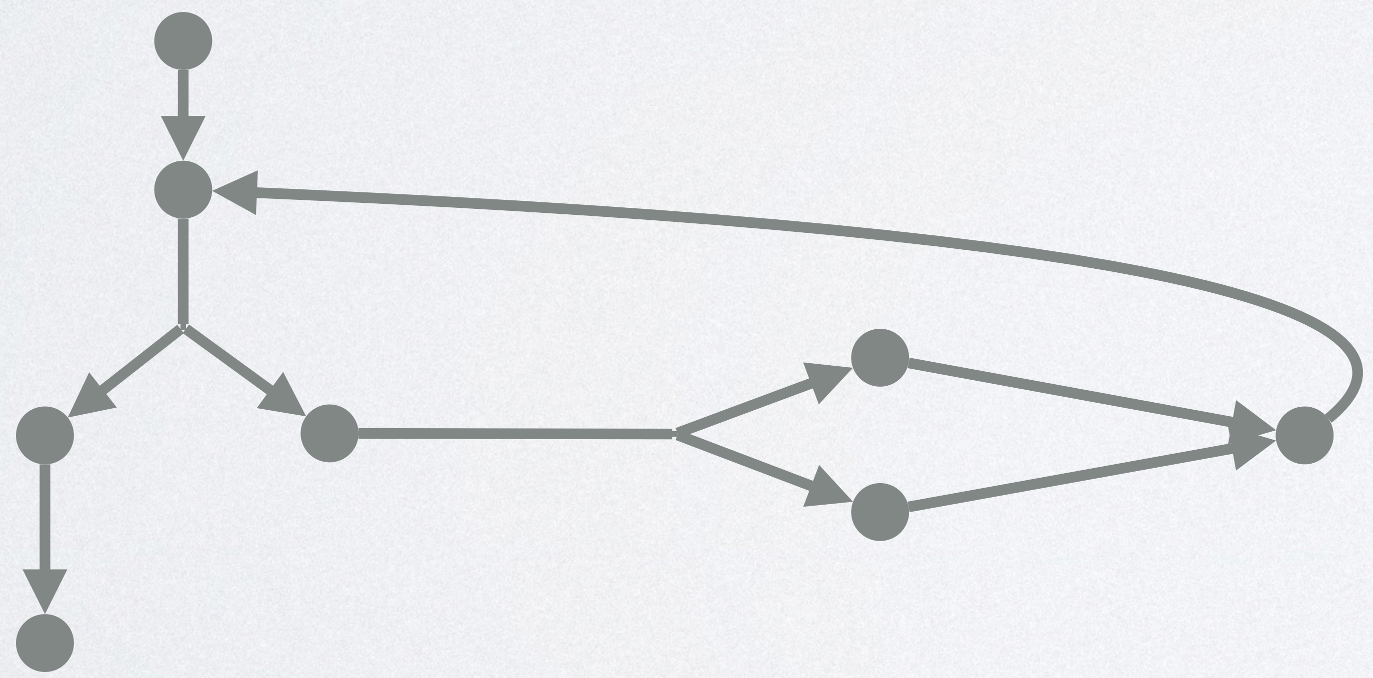
- Only first-order programs
- No function pointers
- Not Galois connections

Future work:

- Higher-order programs
- More efficient algorithm
- New instantiations

PMAF

Design → **Prove** → **Implement**



Hyper-Graph Semantics

- Bayesian Inference
- Markov Decision Problem
- Expectation-Invariant Analysis

Instantiations