The Complexity of Analyzing Infinite-State Markov Chains, Markov Decision Processes, and Stochastic Games

Kousha Etessami

University of Edinburgh

STACS'13
Kiel, March 1st, 2013

Overview

- A number of important countable infinite-state stochastic processes can be captured by adding recursion to finite-state Markov chains.

Overview

- A number of important countable infinite-state stochastic processes can be captured by adding recursion to finite-state Markov chains.
- Adding recursion to MCs also provides a natural abstract model of probabilistic procedural programs (useful in verification).

Overview

- A number of important countable infinite-state stochastic processes can be captured by adding recursion to finite-state Markov chains.
- Adding recursion to MCs also provides a natural abstract model of probabilistic procedural programs (useful in verification).
- Such models can also be captured by probabilistic extensions to classic infinite-state automata-theoretic models, like context-free grammars, pushdown automata, and one-counter automata.

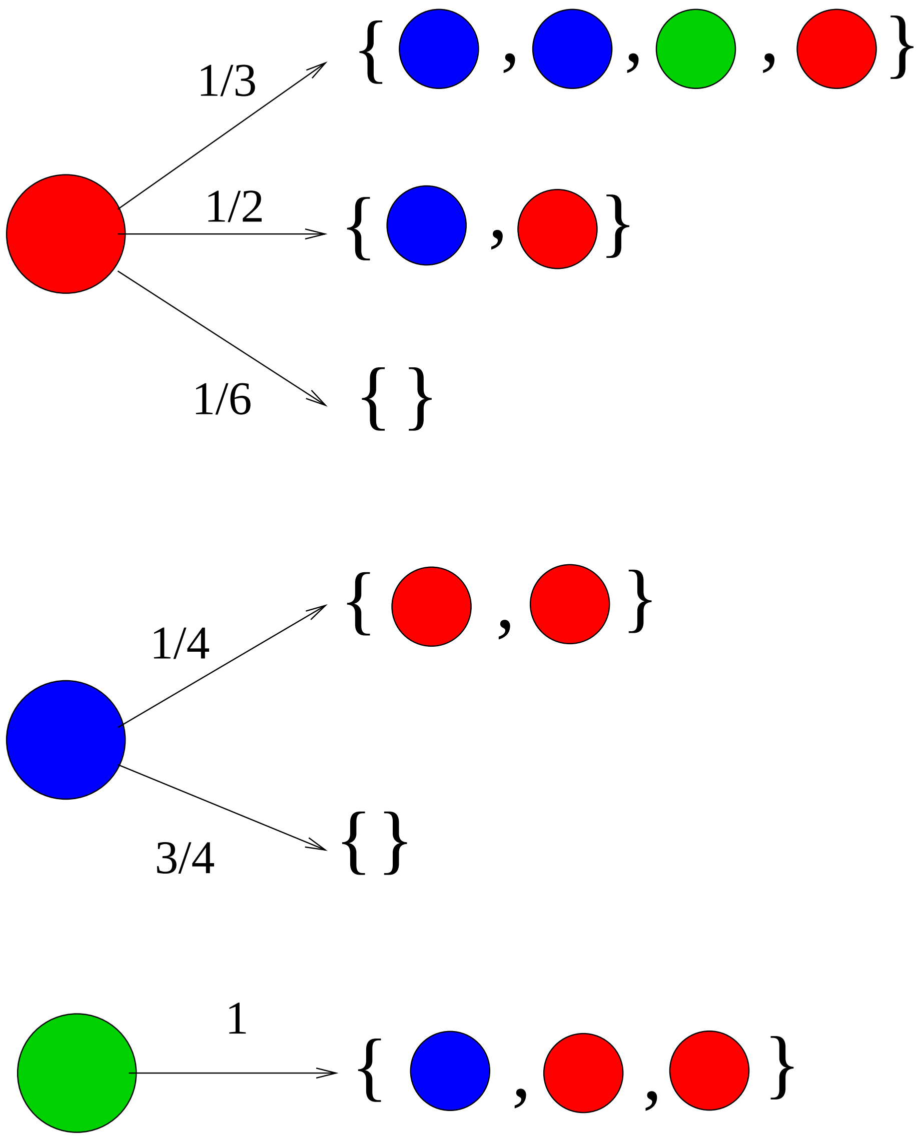
Overview

- A number of important countable infinite-state stochastic processes can be captured by adding recursion to finite-state Markov chains.
- Adding recursion to MCs also provides a natural abstract model of probabilistic procedural programs (useful in verification).
- Such models can also be captured by probabilistic extensions to classic infinite-state automata-theoretic models, like context-free grammars, pushdown automata, and one-counter automata.
- The algorithmic theory, and complexity, of analyzing such recursive MCs and their extension to Markov decision processes and stochastic games, has turned out to be an extremely rich subject.

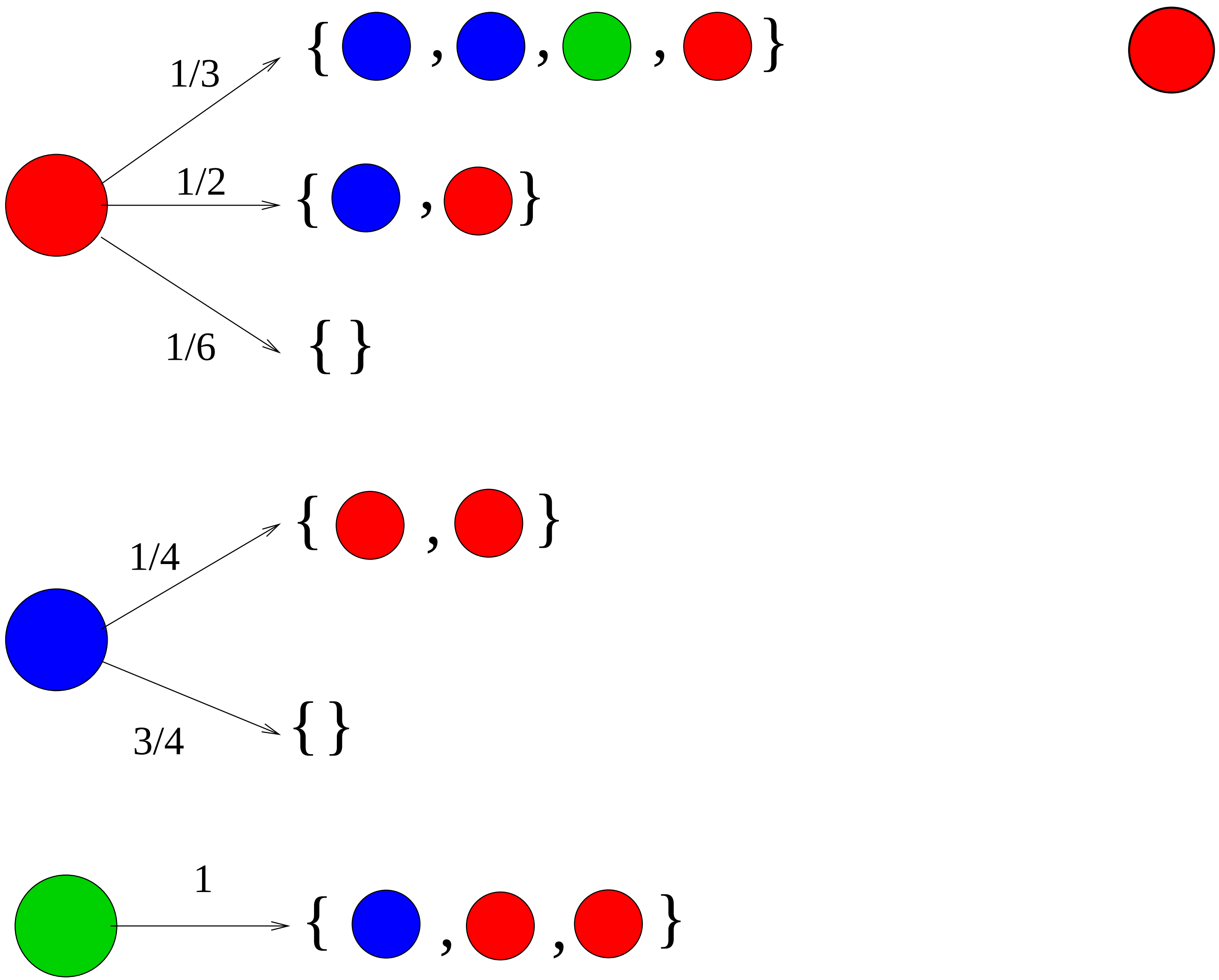
Overview

- A number of important countable infinite-state stochastic processes can be captured by adding recursion to finite-state Markov chains.
- Adding recursion to MCs also provides a natural abstract model of probabilistic procedural programs (useful in verification).
- Such models can also be captured by probabilistic extensions to classic infinite-state automata-theoretic models, like context-free grammars, pushdown automata, and one-counter automata.
- The algorithmic theory, and complexity, of analyzing such recursive MCs and their extension to Markov decision processes and stochastic games, has turned out to be an extremely rich subject.
- In this talk, I will survey only one fragment of this theory (focusing mainly on recent joint work with Alistair Stewart and Mihalis Yannakakis).

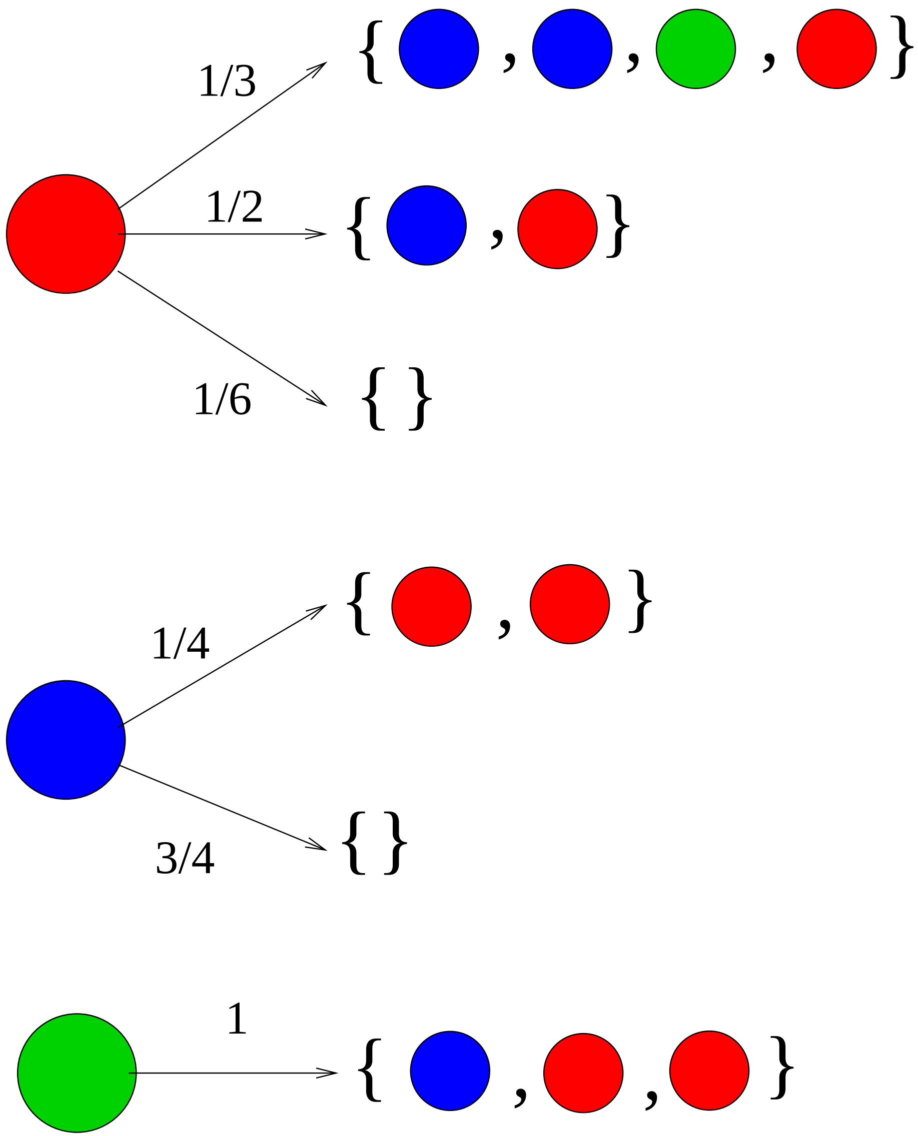
Multi-type Branching Processes (Kolmogorov,1940s)

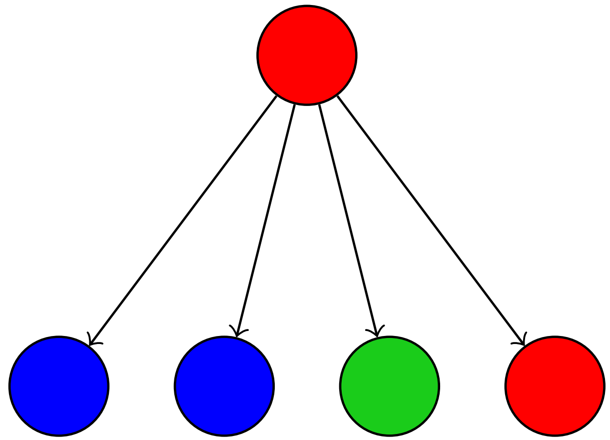


Multi-type Branching Processes (Kolmogorov,1940s)

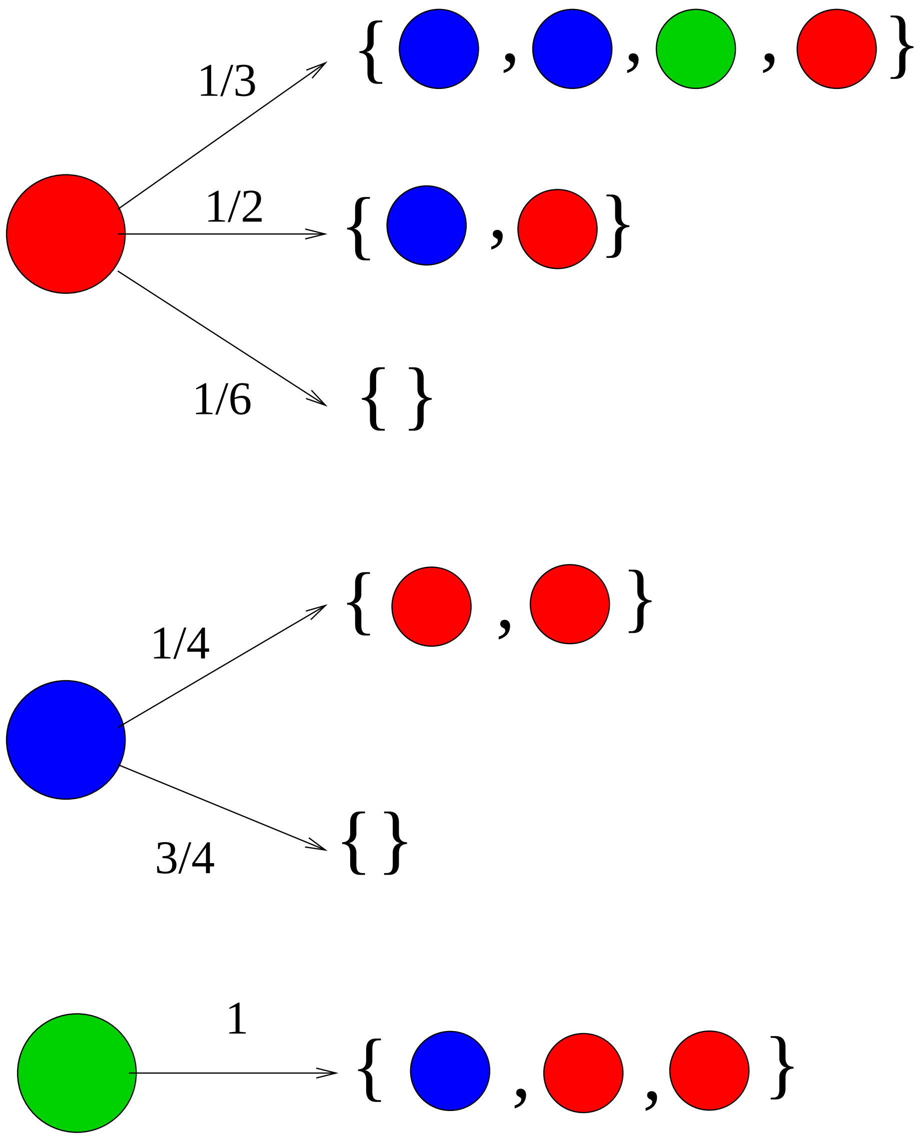


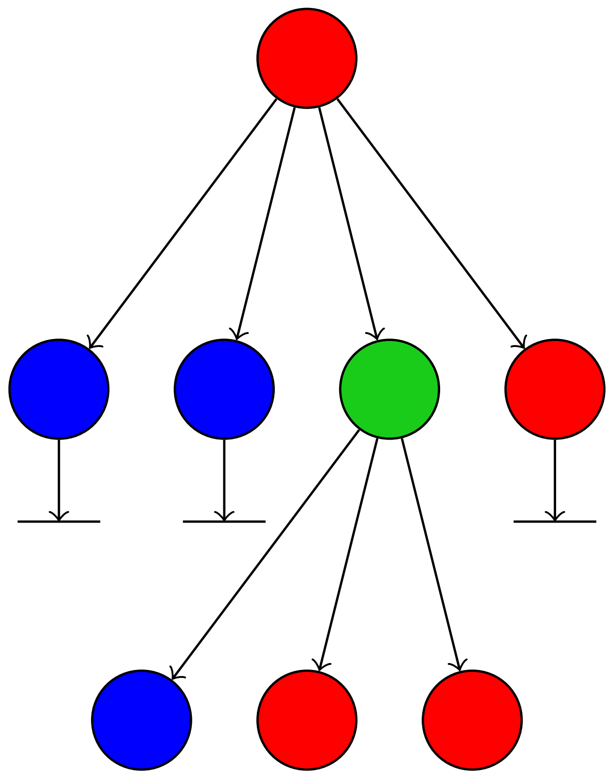
Multi-type Branching Processes (Kolmogorov,1940s)



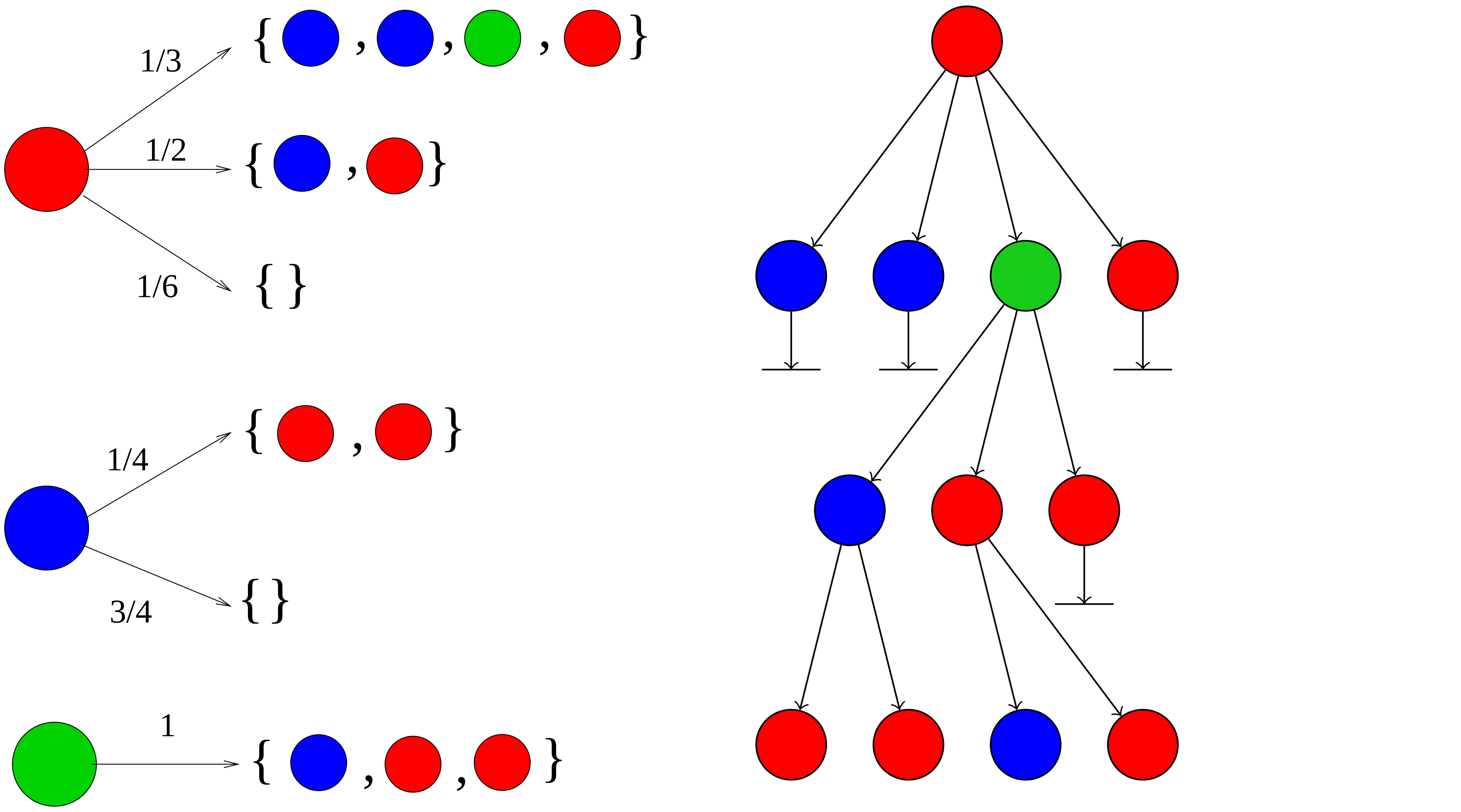


Multi-type Branching Processes (Kolmogorov,1940s)

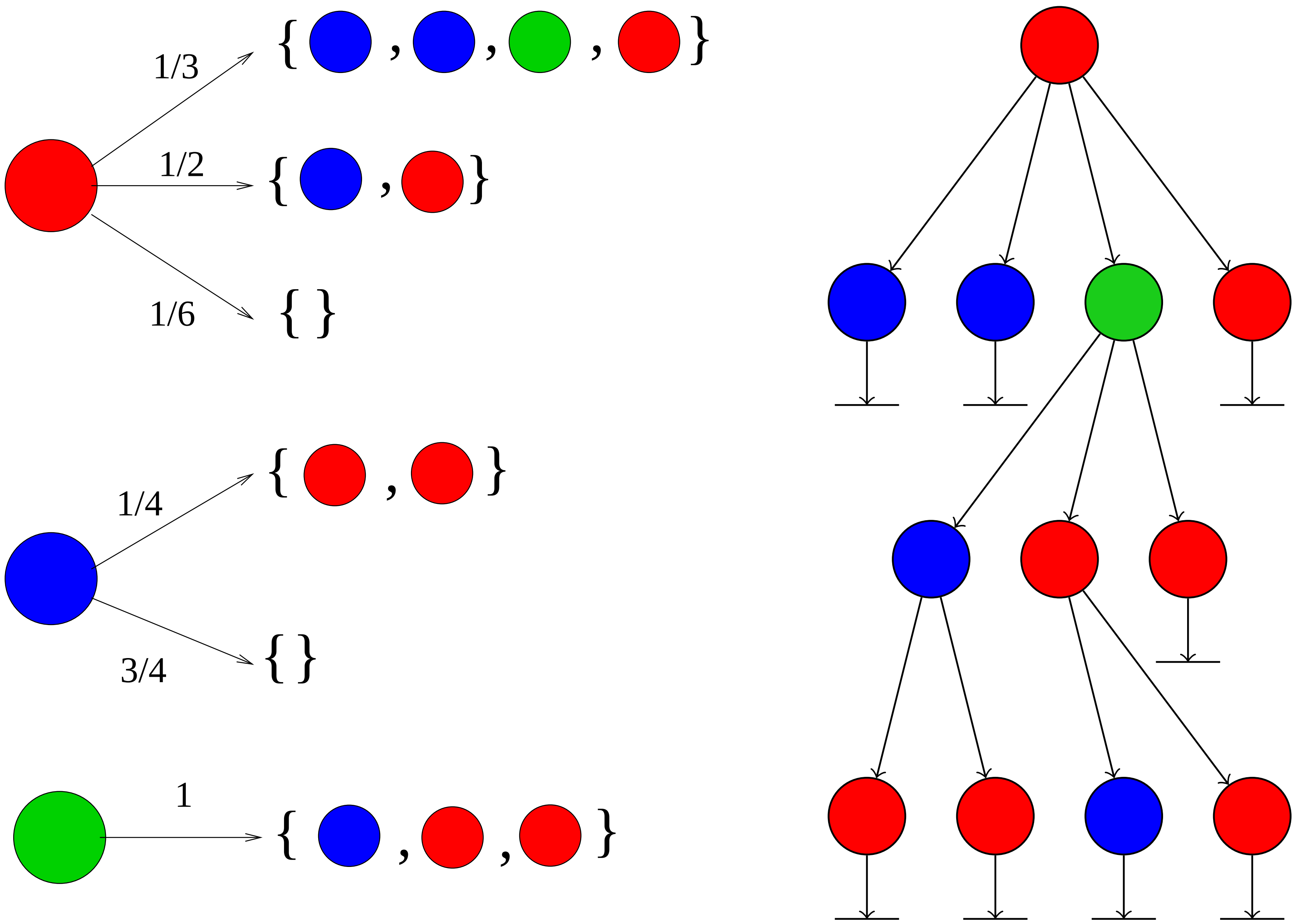




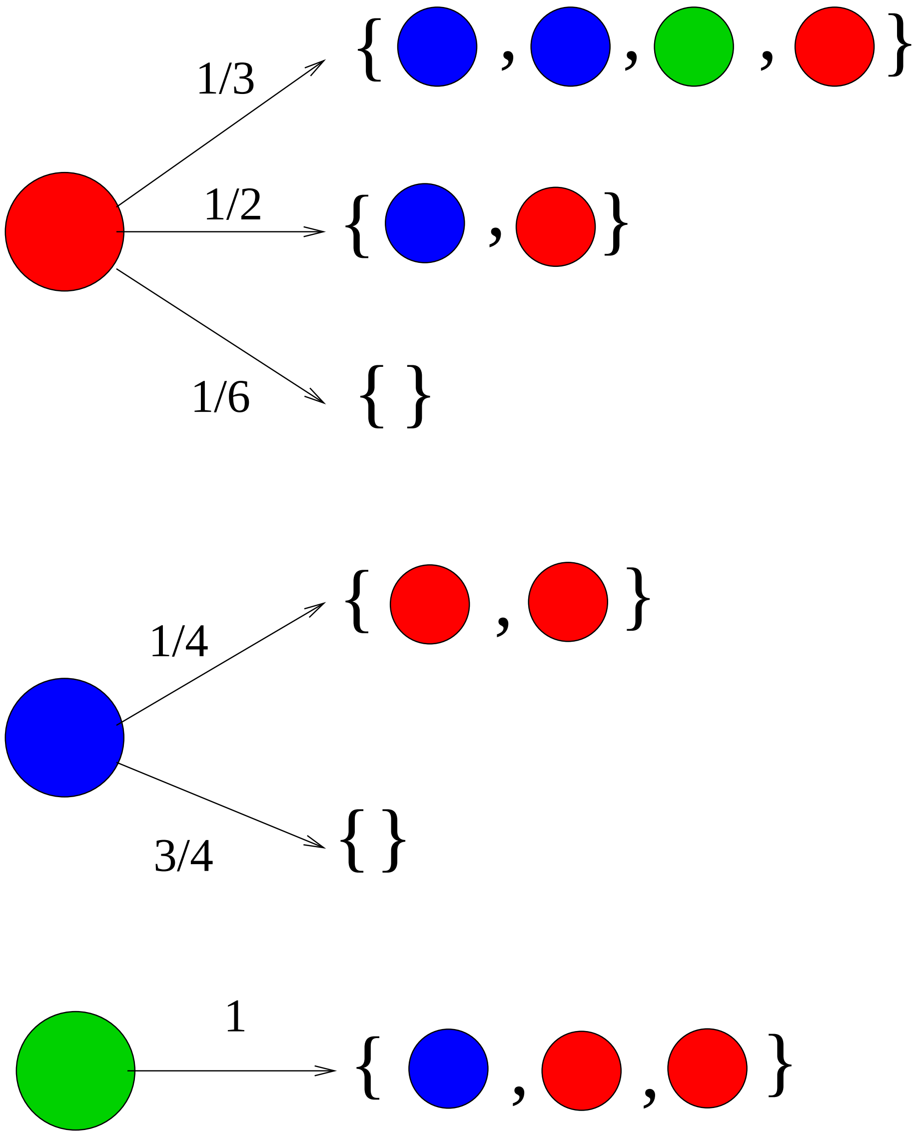
Multi-type Branching Processes (Kolmogorov,1940s)

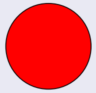


Multi-type Branching Processes (Kolmogorov,1940s)

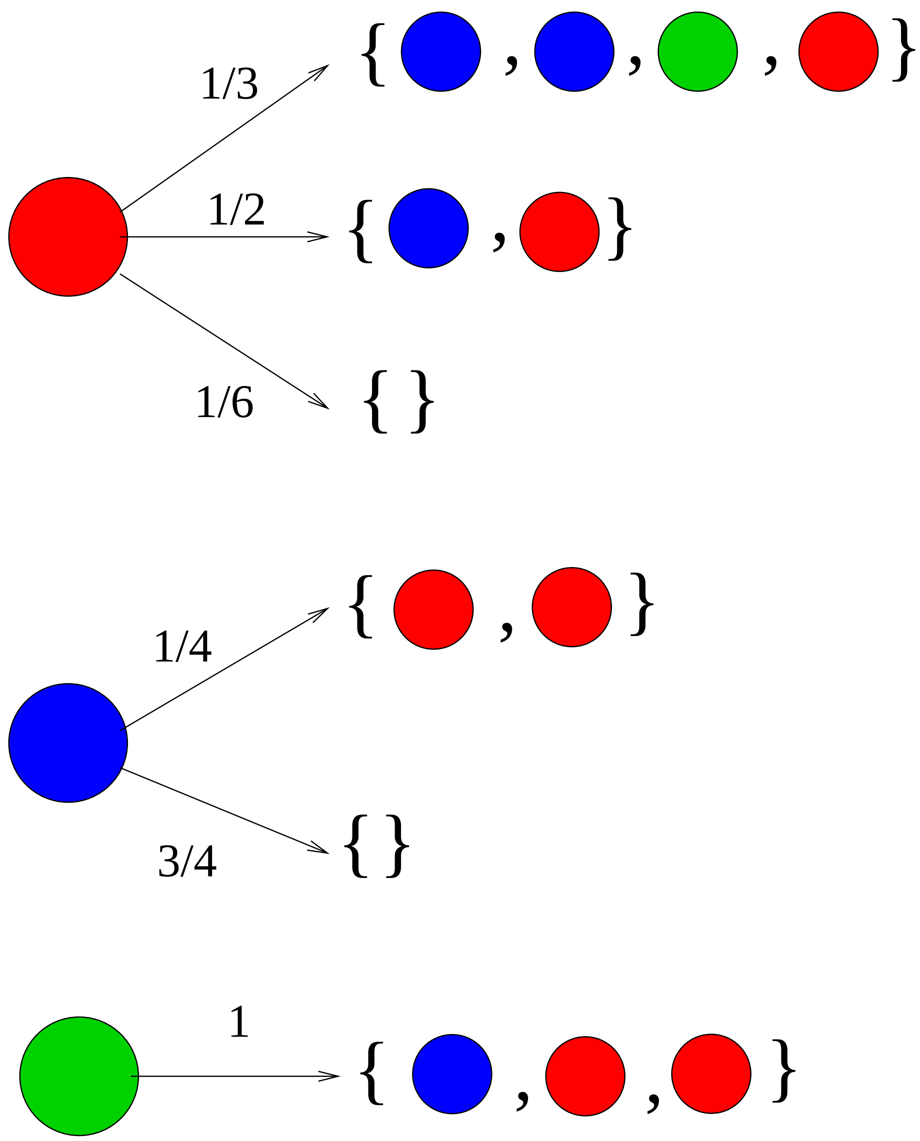


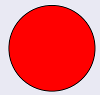
Multi-type Branching Processes (Kolmogorov 1940s)



Question: What is the probability of eventual extinction, starting with one  ?

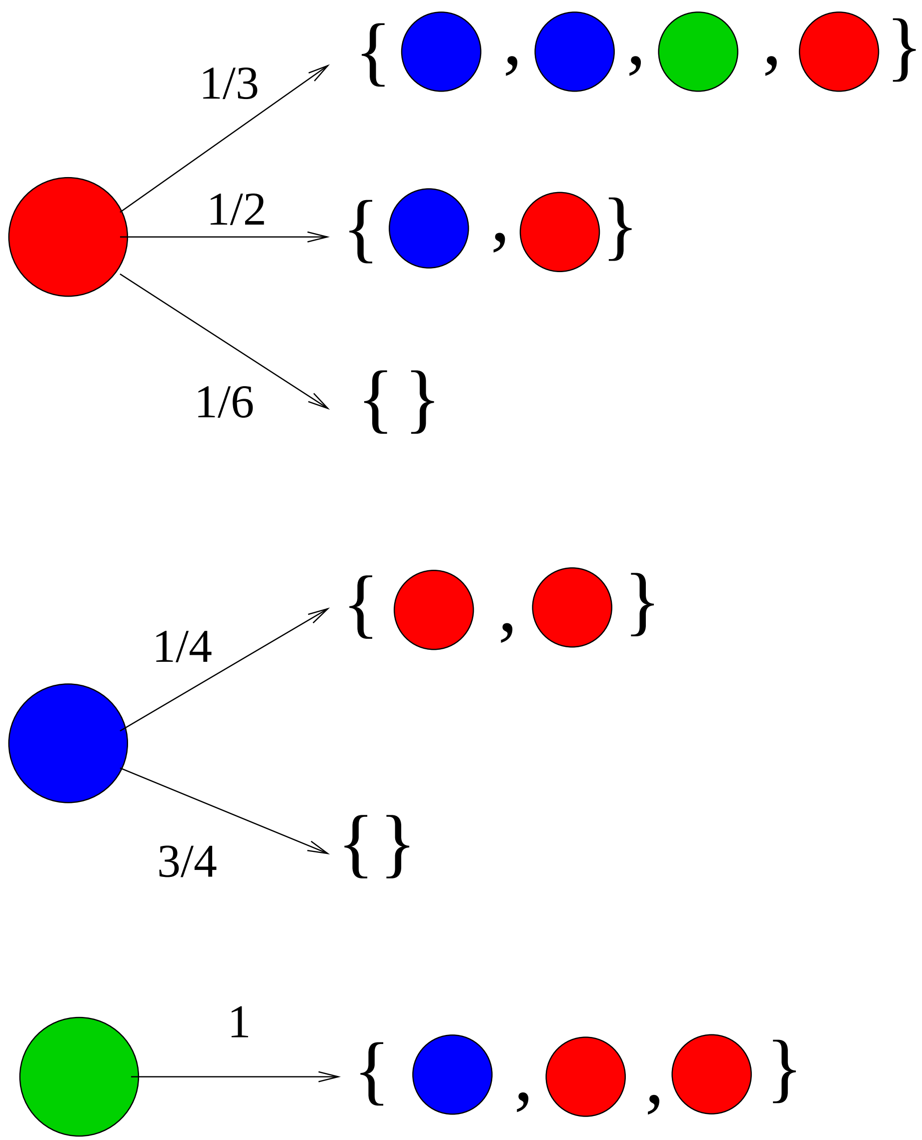
Multi-type Branching Processes (Kolmogorov 1940s)



Question: What is the probability of eventual extinction, starting with one  ?

$x_R \;=$

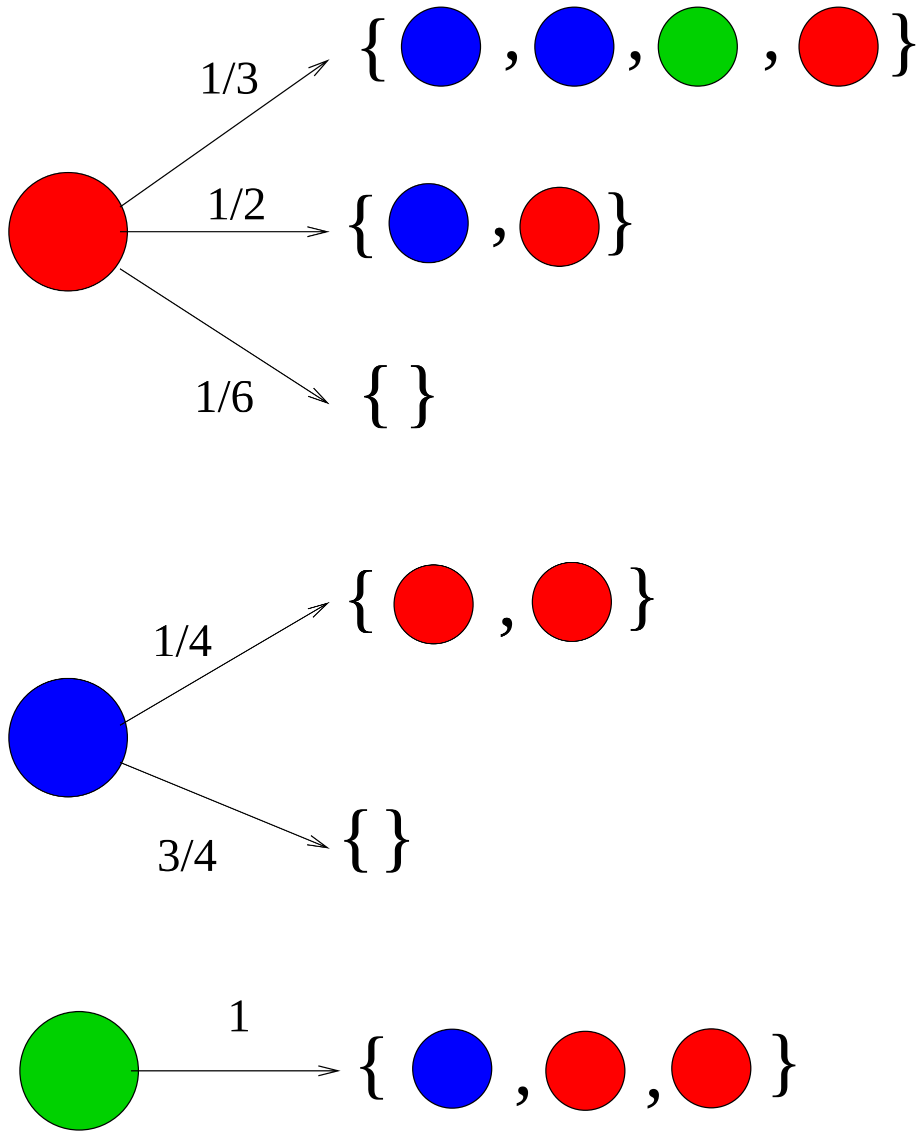
Multi-type Branching Processes (Kolmogorov 1940s)

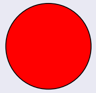


Question: What is the probability of eventual extinction, starting with one ?

$$x_R = \frac{1}{3} x_B^2 x_G x_R + \frac{1}{2} x_B x_R + \frac{1}{6}$$

Multi-type Branching Processes (Kolmogorov 1940s)



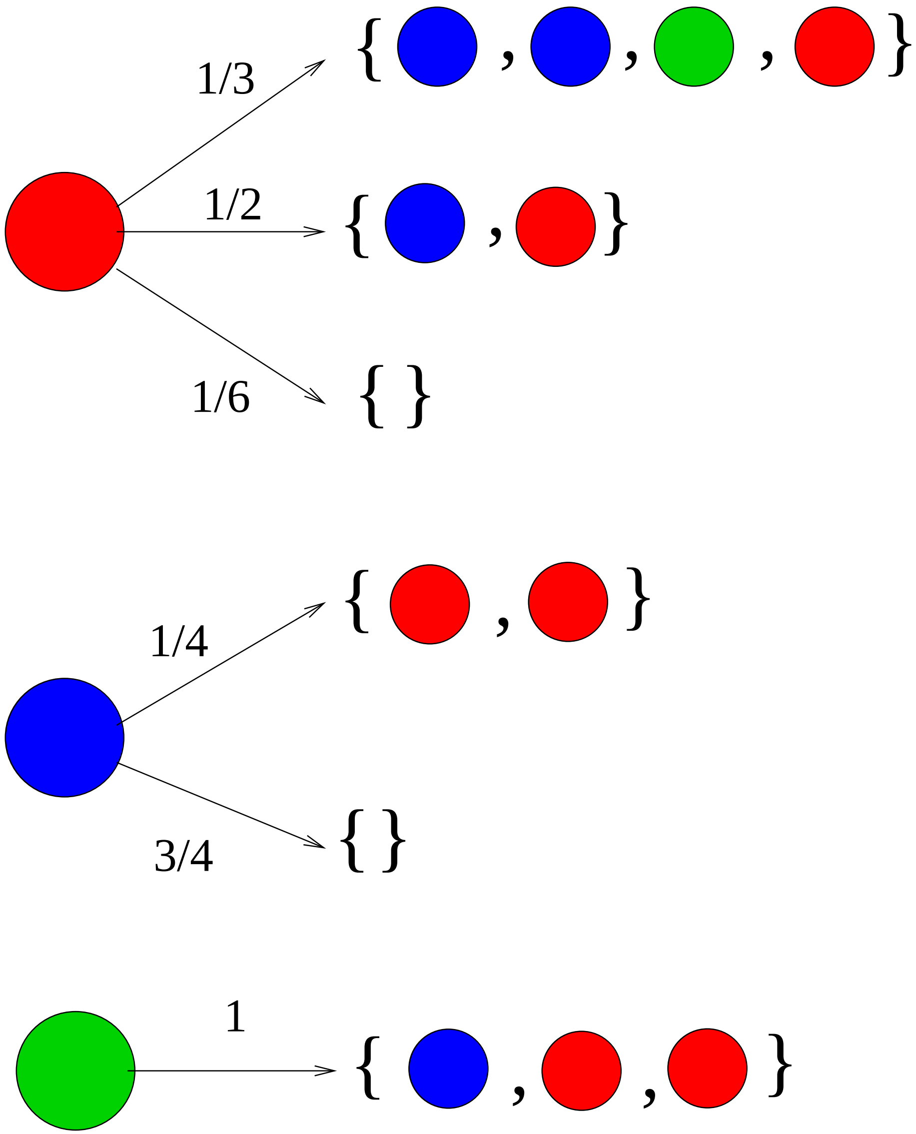
Question: What is the probability of eventual extinction, starting with one  ?

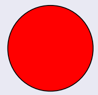
$$x_R = \frac{1}{3}x_B^2 x_G x_R + \frac{1}{2}x_B x_R + \frac{1}{6}$$

$$x_B = \frac{1}{4}x_R^2 + \frac{3}{4}$$

$$x_G = x_B x_R^2$$

Multi-type Branching Processes (Kolmogorov 1940s)



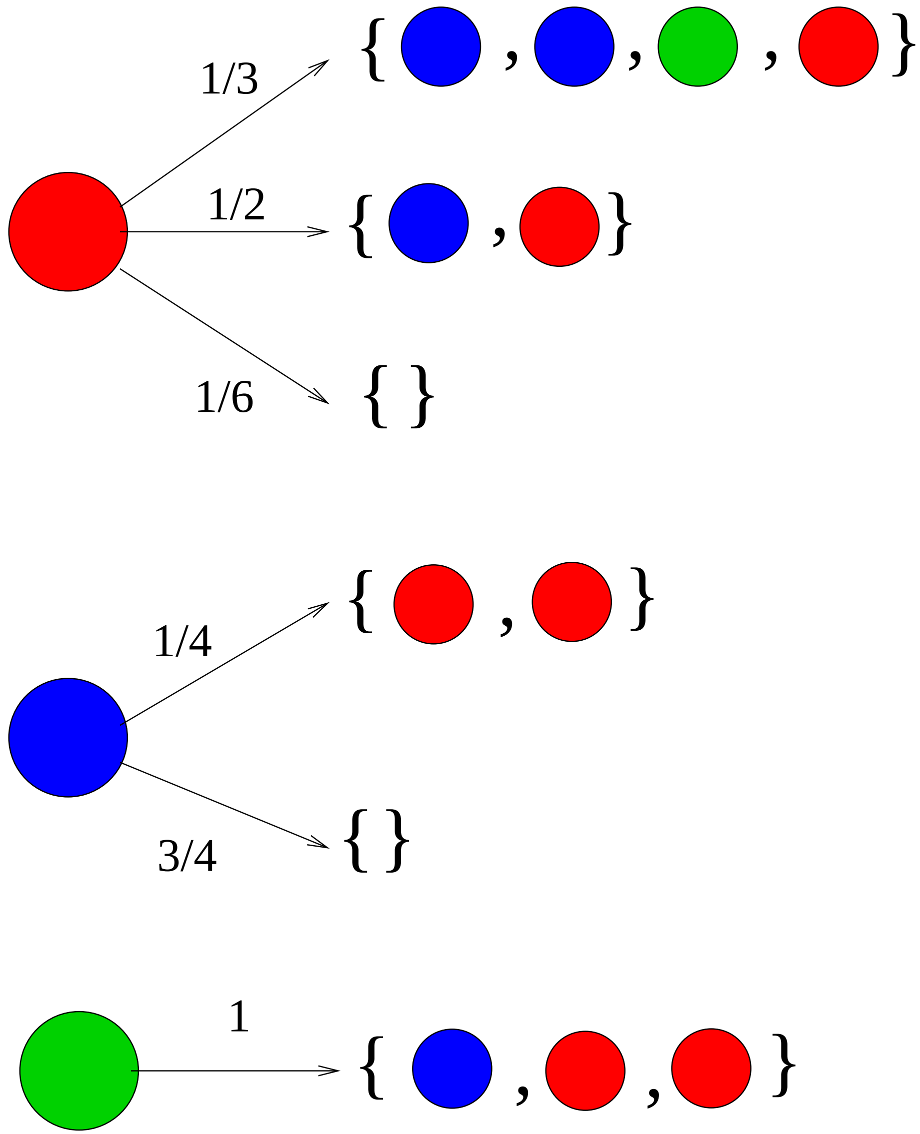
Question: What is the probability of eventual extinction, starting with one  ?

$$
\begin{aligned}
x_R &= \frac{1}{3}x_B^2 x_G x_R + \frac{1}{2}x_B x_R + \frac{1}{6} \\
x_B &= \frac{1}{4}x_R^2 + \frac{3}{4} \\
x_G &= x_B x_R^2
\end{aligned}
$$

We get nonlinear fixed point equations:

$$\bar{\mathbf{x}} = P(\bar{\mathbf{x}}).$$

Multi-type Branching Processes (Kolmogorov 1940s)



Question: What is the probability of eventual extinction, starting with one ?

$$
\begin{aligned}
x_R &= \frac{1}{3}x_B^2 x_G x_R + \frac{1}{2}x_B x_R + \frac{1}{6}\\
x_B &= \frac{1}{4}x_R^2 + \frac{3}{4}\\
x_G &= x_B x_R^2
\end{aligned}
$$

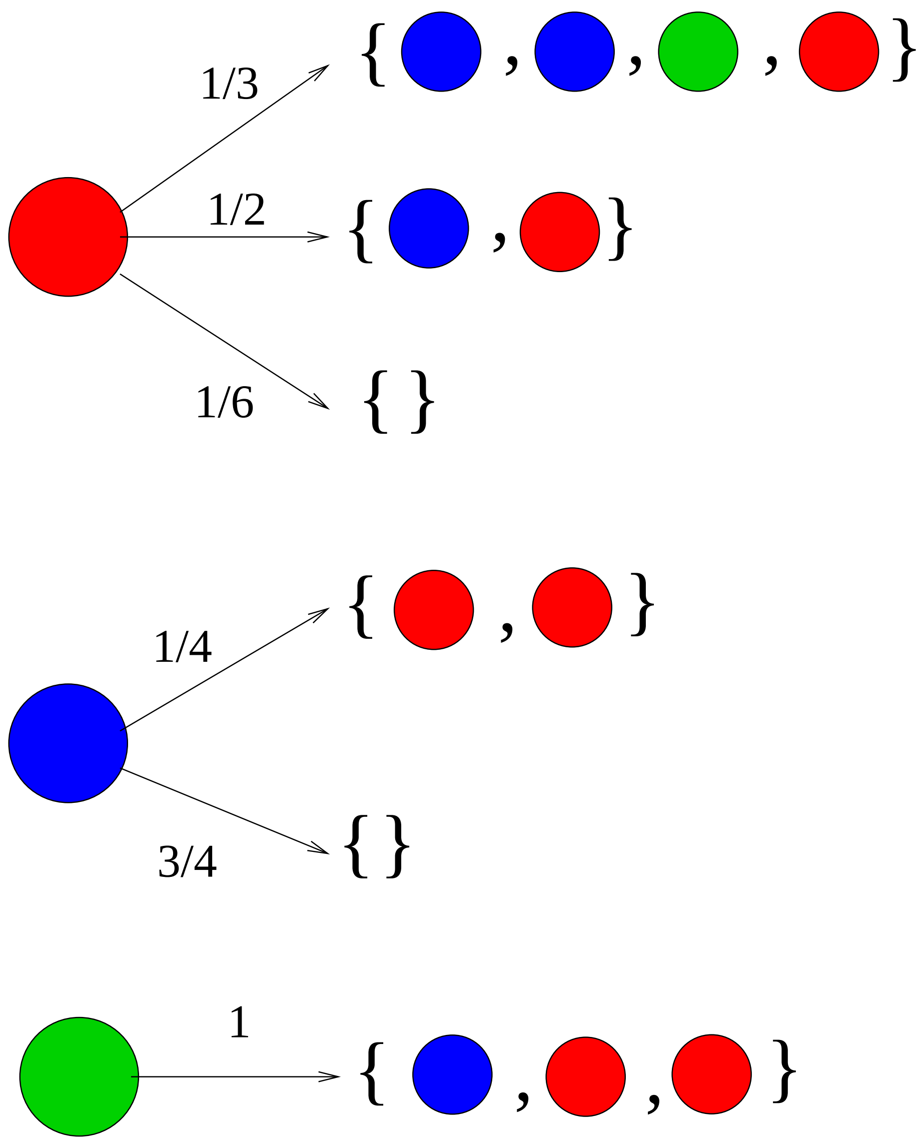
We get nonlinear fixed point equations:

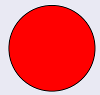
$$\bar{\mathbf{x}} = P(\bar{\mathbf{x}}).$$

Fact

The extinction probabilities are the least fixed point, $\mathbf{q}^* \in [0,1]^3$, of $\bar{\mathbf{x}} = P(\bar{\mathbf{x}})$.

Multi-type Branching Processes (Kolmogorov 1940s)



Question: What is the probability of eventual extinction, starting with one  ?

$$
\begin{aligned}
x_R &= \frac{1}{3}x_B^2 x_G x_R + \frac{1}{2}x_B x_R + \frac{1}{6} \\
x_B &= \frac{1}{4}x_R^2 + \frac{3}{4} \\
x_G &= x_B x_R^2
\end{aligned}
$$

We get nonlinear fixed point equations:

$$\bar{\mathbf{x}} = P(\bar{\mathbf{x}}).$$

Fact

The extinction probabilities are the least fixed point, $\mathbf{q}^* \in [0,1]^3$, of $\bar{\mathbf{x}} = P(\bar{\mathbf{x}})$.
$q_R^* = 0.276$; $q_B^* = 0.769$; $q_G^* = 0.059$.

Stochastic Context-Free Grammars

$$R \xrightarrow{1/3} a\,B\,B\,c\,G\,d\,R$$

$$R \xrightarrow{1/2} b\,c\,B\,b\,R$$

$$R \xrightarrow{1/6} d$$

$$B \xrightarrow{1/4} e\,e\,R\,R\,f$$

$$B \xrightarrow{3/4} g$$

$$G \xrightarrow{1} a\,B\,c\,R\,R\,b$$

Stochastic Context-Free Grammars

$$R \xrightarrow{1/3} aBBcGdR$$

$$R \xrightarrow{1/2} bcBbR$$

$$R \xrightarrow{1/6} d$$

$$B \xrightarrow{1/4} eeRRf$$

$$B \xrightarrow{3/4} g$$

$$G \xrightarrow{1} aBcRRb$$

Leftmost derivation

$\underline{R}$

Stochastic Context-Free Grammars

$$R \xrightarrow{1/3} aBBcGdR$$

$$R \xrightarrow{1/2} bcBbR$$

$$R \xrightarrow{1/6} d$$

$$B \xrightarrow{1/4} eeRRf$$

$$B \xrightarrow{3/4} g$$

$$G \xrightarrow{1} aBcRRb$$

Leftmost derivation

$$\underline{R}$$

$$\xrightarrow{1/2} \quad bc\underline{B}bR$$

Stochastic Context-Free Grammars

$$R \xrightarrow{1/3} aBBcGdR$$

$$R \xrightarrow{1/2} bcBbR$$

$$R \xrightarrow{1/6} d$$

$$B \xrightarrow{1/4} eeRRf$$

$$B \xrightarrow{3/4} g$$

$$G \xrightarrow{1} aBcRRb$$

Leftmost derivation

$$\underline{R}$$

$$\xrightarrow{1/2} bc\underline{B}bR$$

$$\xrightarrow{1/4} bcee\underline{R}RfbR$$

Stochastic Context-Free Grammars

$$R \xrightarrow{1/3} aBBcGdR$$

$$R \xrightarrow{1/2} bcBbR$$

$$R \xrightarrow{1/6} d$$

$$B \xrightarrow{1/4} eeRRf$$

$$B \xrightarrow{3/4} g$$

$$G \xrightarrow{1} aBcRRb$$

Leftmost derivation

$$\underline{R}$$

$$\xrightarrow{1/2} bc\underline{B}bR$$

$$\xrightarrow{1/4} bcee\underline{R}RfbR$$

$$\xrightarrow{1/6} bceed\underline{R}fbR$$

Stochastic Context-Free Grammars

$$R \xrightarrow{1/3} aBBcGdR$$

$$R \xrightarrow{1/2} bcBbR$$

$$R \xrightarrow{1/6} d$$

$$B \xrightarrow{1/4} eeRRf$$

$$B \xrightarrow{3/4} g$$

$$G \xrightarrow{1} aBcRRb$$

Leftmost derivation

$$\underline{R}$$

$$\xrightarrow{1/2} bc\underline{B}bR$$

$$\xrightarrow{1/4} bcee\underline{R}RfbR$$

$$\xrightarrow{1/6} bceed\underline{R}fbR$$

$$\xrightarrow{1/6} bceeddcb\underline{R}$$

Stochastic Context-Free Grammars

$R \xrightarrow{1/3} aBBcGdR$

$R \xrightarrow{1/2} bcBbR$

$R \xrightarrow{1/6} d$

$B \xrightarrow{1/4} eeRRf$

$B \xrightarrow{3/4} g$

$G \xrightarrow{1} aBcRRb$

Leftmost derivation

$$\begin{aligned}
& \underline{R} \\
\xrightarrow{1/2} \quad & bc\underline{B}bR \\
\xrightarrow{1/4} \quad & bcee\underline{R}RfbR \\
\xrightarrow{1/6} \quad & bceed\underline{R}fbR \\
\xrightarrow{1/6} \quad & bceeddcb\underline{R} \\
\xrightarrow{1/6} \quad & bceeddcbd
\end{aligned}$$

probability of this derivation: $\frac{1}{2} \cdot \frac{1}{4} \cdot \frac{1}{6}^3$

Stochastic Context-Free Grammars

$$R \xrightarrow{1/3} aBBcGaabR$$

$$R \xrightarrow{1/2} bcBbR$$

$$R \xrightarrow{1/6} cb$$

$$B \xrightarrow{1/4} bbRRc$$

$$B \xrightarrow{3/4} a$$

$$G \xrightarrow{1} aBcRRb$$

Question

What is the probability of termination, i.e., eventually generating a finite string, starting with one non-terminal, R ?

Stochastic Context-Free Grammars

$$R \xrightarrow{1/3} aBBcGaabR$$

$$R \xrightarrow{1/2} bcBbR$$

$$R \xrightarrow{1/6} cb$$

$$B \xrightarrow{1/4} bbRRc$$

$$B \xrightarrow{3/4} a$$

$$G \xrightarrow{1} aBcRRb$$

Question

What is the probability of termination, i.e., eventually generating a finite string, starting with one non-terminal, R ?

$$\begin{aligned} x_R &= \frac{1}{3}x_B^2 x_G x_R + \frac{1}{2}x_B x_R + \frac{1}{6} \\ x_B &= \frac{1}{4}x_R^2 + \frac{3}{4} \\ x_G &= x_B x_R^2 \end{aligned}$$

Fact

Termination probabilities (also called the partition function of the SCFG) are the least fixed point, $\mathbf{q}^* \in [0,1]^3$, of $\bar{\mathbf{x}} = P(\bar{\mathbf{x}})$.

Probabilistic Polynomial Systems of Equations

$$\frac{1}{3}x_B^2 x_G x_R + \frac{1}{2}x_B x_R + \frac{1}{6}$$

is a Probabilistic Polynomial: the coefficients are positive and sum to 1.

A Probabilistic Polynomial System (PPS), is a system of n equations

$$\mathbf{x} = P(\mathbf{x})$$

in n variables where each $P_i(x)$ is a probabilistic polynomial.

Every multi-type Branching Process (BP) with n types, and every SCFG with n nonterminals, corresponds to a PPS, and vice-versa.

Basic properties of PPSs, $\mathbf{x} = P(\mathbf{x})$

For every PPS, $P : [0,1]^n \rightarrow [0,1]^n$ defines a monotone map on $[0,1]^n$.

Proposition

- *A PPS, $x = P(x)$ has a least fixed point, $q^* \in [0,1]^n$.*
  (q^* can be irrational.)
- $q^* = \lim_{k \rightarrow \infty} P^k(\mathbf{0})$.
- *q^* is vector of extinction/termination probabilities for the BP (SCFG).*

Question

Can we compute the probabilities q^* efficiently (in P-time)?

First considered by Kolmogorov & Sevastyanov (1940s).

Newton's method

Newton's method

Seeking a solution to $F(\mathbf{x}) = 0$, we start at a guess $\mathbf{x}^{(0)}$, and iterate:

$$\mathbf{x}^{(k+1)} := \mathbf{x}^{(k)} - (F'(\mathbf{x}^{(k)}))^{-1} F(\mathbf{x}^{(k)})$$

Here $F'(\mathbf{x})$, is the **Jacobian matrix**:

$$F'(\mathbf{x}) = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} \dots \frac{\partial F_1}{\partial x_n} \\ \vdots \vdots \vdots \\ \frac{\partial F_n}{\partial x_1} \dots \frac{\partial F_n}{\partial x_n} \end{bmatrix}$$

For PPSs, $F(x) \equiv (P(x) - x)$, and Newton iteration looks like this:

$$\mathbf{x}^{(k+1)} := \mathbf{x}^{(k)} + (I - P'(\mathbf{x}^{(k)}))^{-1}(P(\mathbf{x}^{(k)}) - \mathbf{x}^{(k)})$$

where $P'(\mathbf{x})$ is the Jacobian of $P(\mathbf{x})$.

Newton on PPSs

We can decompose $\mathbf{x} = P(\mathbf{x})$ into its strongly connected components (SCCs), based on variable dependencies, and eliminate "0" variables.

Theorem [E.-Yannakakis'05]

Decomposed Newton's method converges monotonically to the LFP $\mathbf{q}^*$ for PPSs, and for more general Monotone Polynomial Systems (MPSs).

But...

- In [E.-Yannakakis'05,'09], we gave no upper bounds on # of iterations needed for PPSs or MPSs.
- We proved hardness results (PosSLP-hardness) for obtaining any nontrivial approximation of the LFP of MPSs for recursive Markov chains.

What is Newton's worst case behavior for PPSs?

[Esparza,Kiefer,Luttenberger,'10] studied Newton's method on MPSs further:

- Gave bad examples of PPSs, $\mathbf{x} = P(\mathbf{x})$, where $q^* = 1$, requiring exponentially many iterations, as a function of the encoding size $|P|$ of the equations, to converge to within additive error $< 1/2$.
- For strongly-connected equation systems they gave an exponential upper bound in $|P|$.
- But they gave no upper bounds for arbitrary PPSs or MPSs in terms of $|P|$.

What is Newton's worst case behavior for PPSs?

[Esparza,Kiefer,Luttenberger,'10] studied Newton's method on MPSs further:

- Gave bad examples of PPSs, $\mathbf{x} = P(\mathbf{x})$, where $q^* = 1$, requiring exponentially many iterations, as a function of the encoding size $|P|$ of the equations, to converge to within additive error $< 1/2$.
- For strongly-connected equation systems they gave an exponential upper bound in $|P|$.
- But they gave no upper bounds for arbitrary PPSs or MPSs in terms of $|P|$.
- (Recently [Stewart-E.-Yannakakis'13], we give a matching exponential upper bound in $|P|$ for arbitrary PPSs and MPSs.)

P-time approximation for PPSs

Theorem ([E.-Stewart-Yannakakis,STOC'12])

Given a PPS, $\mathbf{x} = P(\mathbf{x})$, with LFP $\mathbf{q}^* \in [0,1]^n$, we can compute a rational vector $\mathbf{v} \in [0,1]^n$ such that

$$\|\mathbf{v} - \mathbf{q}^*\|_\infty \leq 2^{-j}$$

in time polynomial in both the encoding size $|P|$ of the equations and in j (the number of "bits of precision").

We use Newton's method..... but how?

Qualitative decision problems for PPSs are in P-time

Theorem ([Kolmogorov-Sevastyanov'47,Harris'63])

For certain classes of strongly-connected PPSs, $q_i^* = 1$ for all i iff the spectral radius $\varrho(P'(\mathbf{1}))$ for the moment matrix $P'(\mathbf{1})$ is ≤ 1, and otherwise $q_i^* < 1$ for all i.

Theorem ([E.-Yannakakis'05])

Given a PPS, $\mathbf{x} = P(\mathbf{x})$, deciding whether $q_i^* = 1$ is in P-time.

(Deciding whether $q_i^* = 0$ is also in P-time (and a lot easier).)

Algorithm for approximating the LFP q^* for PPSs

1. Find and remove all variables x_i such that $q_i^* = 0$ or $q_i^* = 1$.
2. On the resulting system of equations, run Newton's method starting from $\mathbf{0}$.

Algorithm for approximating the LFP q^* for PPSs

1. Find and remove all variables x_i such that $q_i^* = 0$ or $q_i^* = 1$.
2. On the resulting system of equations, run Newton's method starting from $\mathbf{0}$.

Theorem ([ESY'12])

Given a PPS $\mathbf{x} = P(\mathbf{x})$ with LFP $\mathbf{0} < \mathbf{q}^* < \mathbf{1}$, if we apply Newton starting at $\mathbf{x}^{(0)} = \mathbf{0}$, then

$$\|\mathbf{q}^* - \mathbf{x}^{(4|P|+j)}\|_\infty \leq 2^{-j}$$

Algorithm for approximating the LFP q^* for PPSs

1. Find and remove all variables x_i such that $q_i^* = 0$ or $q_i^* = 1$.
2. On the resulting system of equations, run Newton's method starting from $\mathbf{0}$.

Theorem ([ESY'12])

Given a PPS $\mathbf{x} = P(\mathbf{x})$ with LFP $\mathbf{0} < \mathbf{q}^* < \mathbf{1}$, if we apply Newton starting at $\mathbf{x}^{(0)} = \mathbf{0}$, then

$$\|\mathbf{q}^* - \mathbf{x}^{(4|P|+j)}\|_\infty \leq 2^{-j}$$

Theorem ([ESY'12])

Given a PPS $\mathbf{x} = P(\mathbf{x})$ with LFP $\mathbf{0} < \mathbf{q}^* < \mathbf{1}$, if we apply Newton starting at $\mathbf{x}^{(0)} = \mathbf{0}$, then

$$\|\mathbf{q}^* - \mathbf{x}^{(32|P|+2j+2)}\|_\infty \leq 2^{-2^j}$$

Algorithm with rounding

1. Find and remove all variables x_i such that $q_i^* = 0$ or $q_i^* = 1$.
2. On the resulting system of equations, run Newton's method starting from $\mathbf{0}$.
3. After each iteration, round down to a multiple of 2^{-h}

Theorem ([ESY'12])

If, after each Newton iteration, we round down to a multiple of 2^{-h} where $h := 4|P| + j + 2$, then after h iterations $\|\mathbf{q}^* - \mathbf{x}^{(h)}\|_\infty \leq 2^{-j}$.

Thus, we obtain a P-time algorithm (in the standard Turing model) for approximating q^*.

High level picture of proof

- For a PPS, $x = P(x)$, with LFP $\mathbf{0} < \mathbf{q}^* < \mathbf{1}$, $P'(q^*)$ is a non-negative square matrix, and (we show)

$$\text{(spectral radius of } P'(q^*) \text{)} \equiv \varrho(P'(q^*)) < 1$$

- So, $(I - P'(q^*))$ is non-singular, and $(I - P'(q^*))^{-1} = \sum_{i=0}^{\infty}(P'(q^*))^i$.

- We can show the $\#$ of Newton iterations needed to get within $\epsilon > 0$ is

$$\approx\approx \log \|(I - P'(q^*))^{-1}\|_\infty + \log \frac{1}{\epsilon}$$

- $\|(I - P'(q^*))^{-1}\|_\infty$ is tied to the distance $|1 - \varrho(P'(q^*))|$, which in turn is related to $\min_i(1 - q_i^*)$, which we can lower bound.

- Uses lots of Perron-Frobenius theory.

Proof outline: some key lemmas

$(\mathbf{1} - \mathbf{q}^*)$ is the vector of survival probabilities.

Lemma

If $\mathbf{q}^* - \mathbf{x}^{(k)} \leq \lambda(\mathbf{1} - \mathbf{q}^*)$ *for some* $\lambda > 0$, *then* $\mathbf{q}^* - \mathbf{x}^{(k+1)} \leq \frac{\lambda}{2}(\mathbf{1} - \mathbf{q}^*)$.

Lemma

For any PPS with LFP $\mathbf{q}^*$, *such that* $\mathbf{0} < \mathbf{q}^* < \mathbf{1}$, *for any* i,

$$q_i^* \leq 1 - 2^{-4|P|}.$$

The complexity of quantitative decision problems for BPs

Proposition

Given a PPS, $x = P(x)$, and a probability p, deciding whether $q_i^* \leq p$ is in PSPACE.

Proof.

$$\exists \mathbf{x}(\mathbf{x} = P(\mathbf{x}) \wedge x_i \leq p)$$

is expressible in the existential theory of reals. There are PSPACE decision procedures for $\exists\mathbb{R}$ ([Canny'89,Renegar'92]). □

Now some bad news:

Theorem ([E.-Yannakakis,'05,'07])

Given a PPS, $x = P(x)$, deciding whether $q_i^* \leq 1/2$ (or $q_i^* \leq p$ for any $p \in (0,1)$), is both Sqrt-Sum-hard and PosSLP-hard.

two "hard" problems

Sqrt-Sum: the square-root sum problem is the following decision problem:

Given $(d_1, \ldots, d_n) \in \mathbb{N}^n$ and $k \in \mathbb{N}$, decide whether $\sum_{i=1}^{n} \sqrt{d_i} \leq k$.

Solvable in PSPACE.

Open problem ([GareyGrahamJohnson'76]) whether it is in NP (or even the polynomial time hierarchy).

PosSLP: Given an arithmetic circuit (Straight Line Program) with gates $\{+, *, -\}$ with integer inputs, decide whether the output is > 0.

PosSLP captures all of polynomial time in the unit-cost arithmetic RAM model of computation.

[Allender, Bürgisser, Kjeldal-Petersen, Miltersen,2006] Gave a (Turing) reduction from Sqrt-Sum to PosSLP and showed both can be decided in the Counting Hierarchy: $P^{PP^{PP^{PP}}}$. Nothing better is known.

The quantitative decision problem for PPSs is PosSLP-equivalent

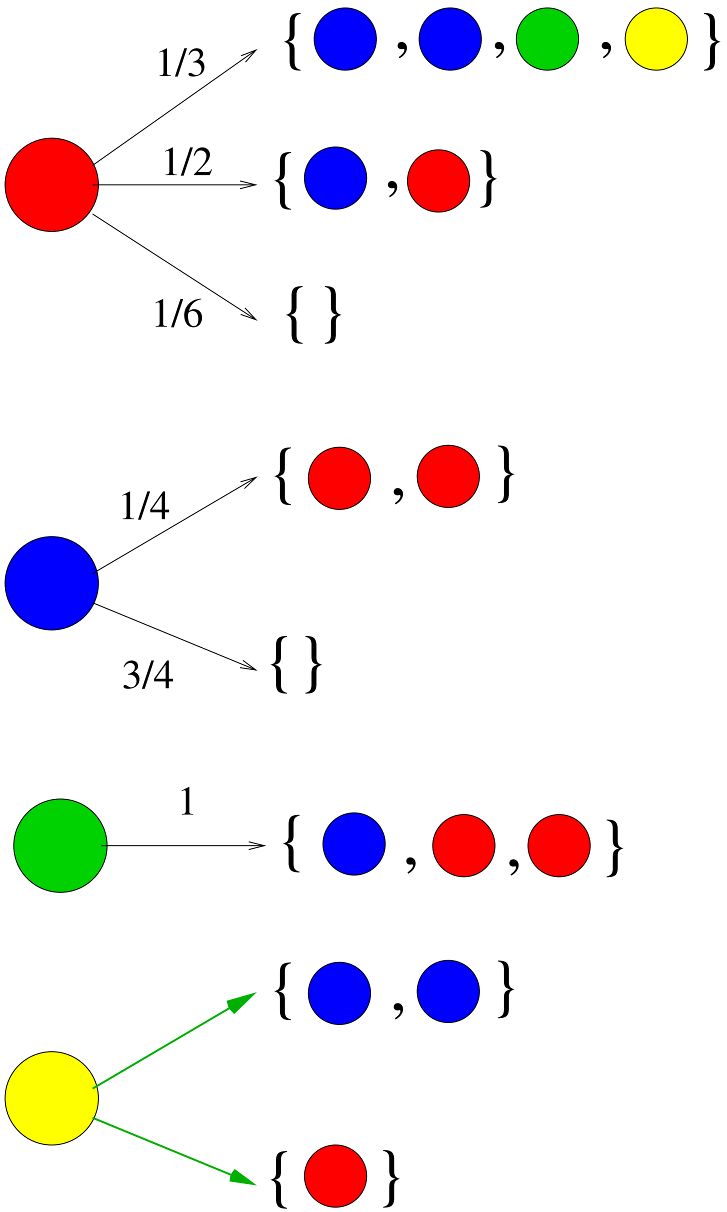
Theorem ([E.-Stewart-Yannakakis'12])

Given a PPS, $x = P(x)$, and a probability p, deciding whether $q_i^* < p$ is P-time (many-one) reducible to PosSLP. (And thus PosSLP-equivalent.)

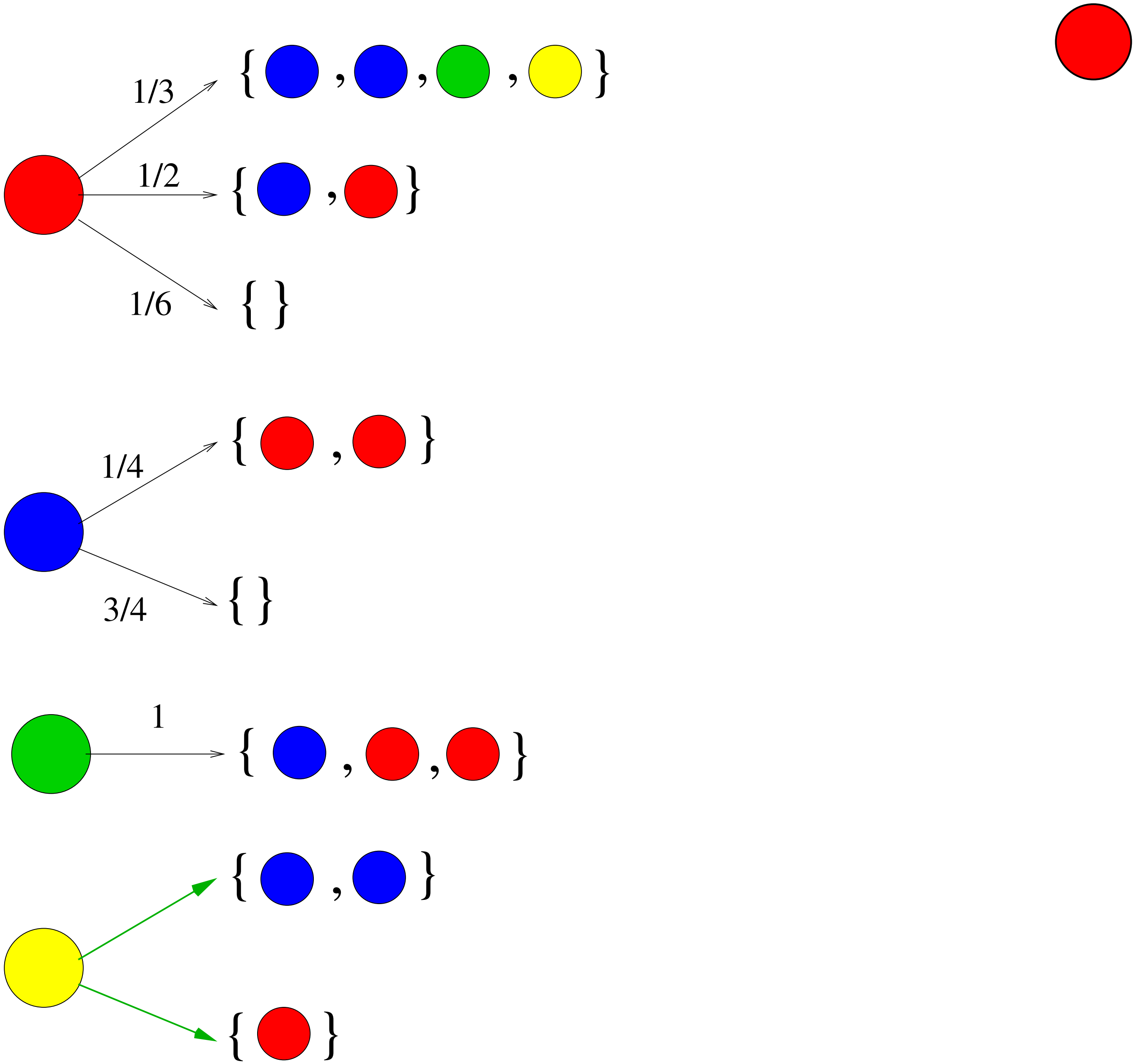
- Thus it captures the full power of polynomial time in the unit-cost arithmetic RAM model of computation.

  And by [Allender, et. al.'06], it is also in the Counting Hierarchy.

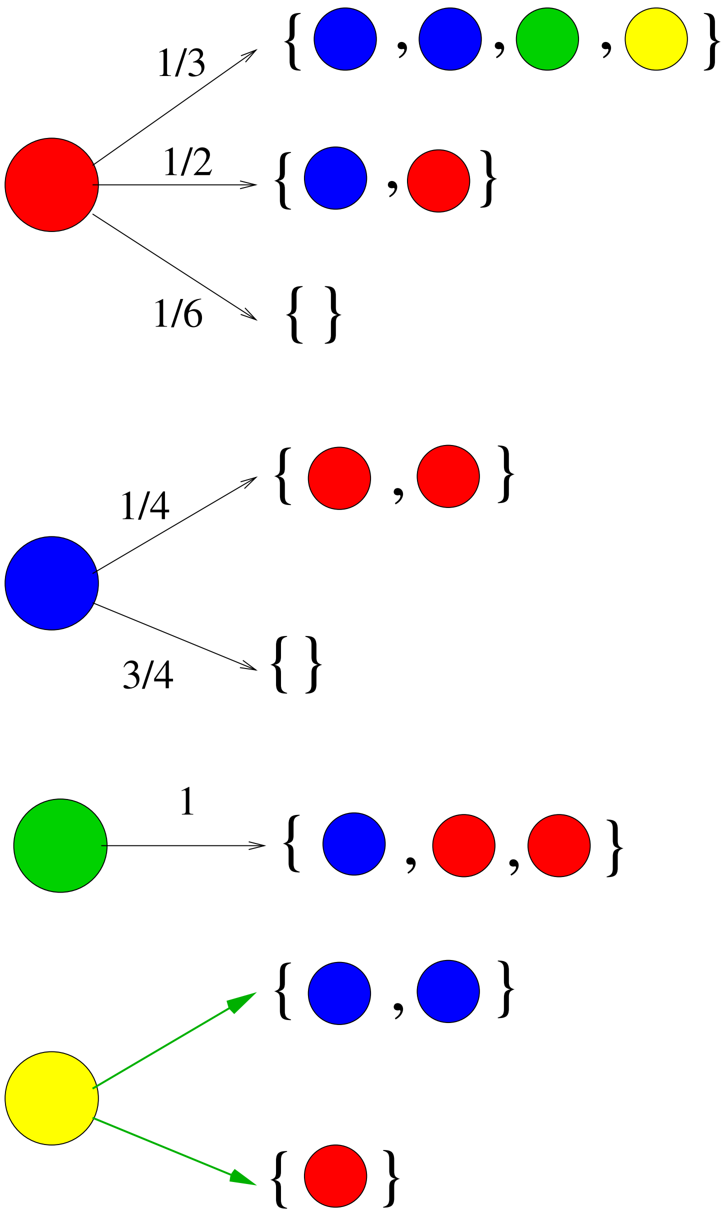
Branching Markov Decision Processes

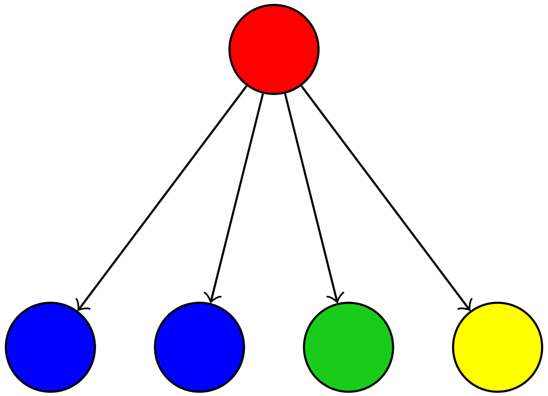


Branching Markov Decision Processes

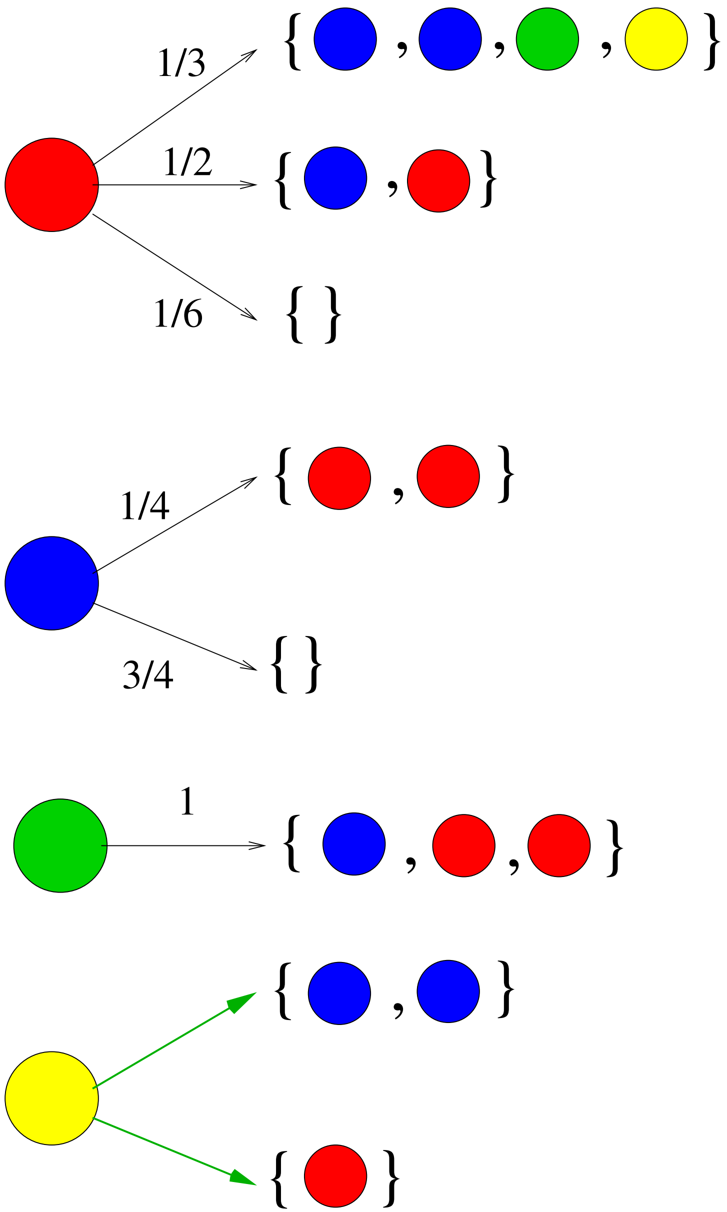


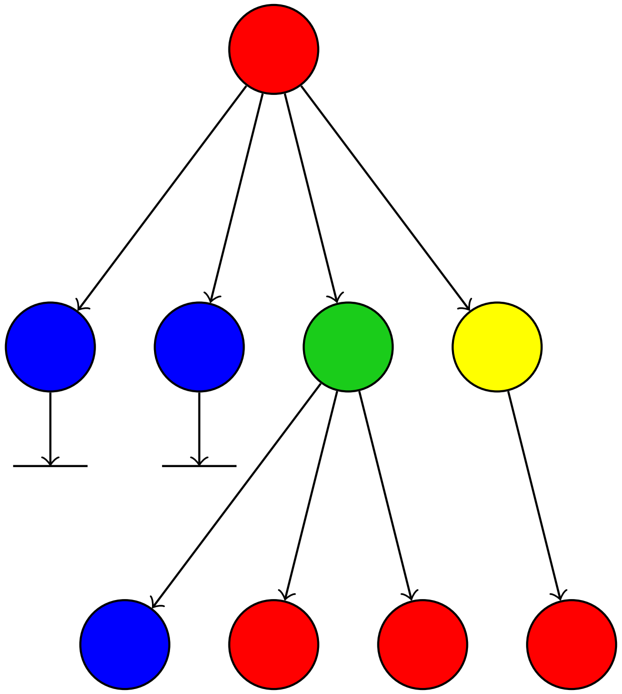
Branching Markov Decision Processes



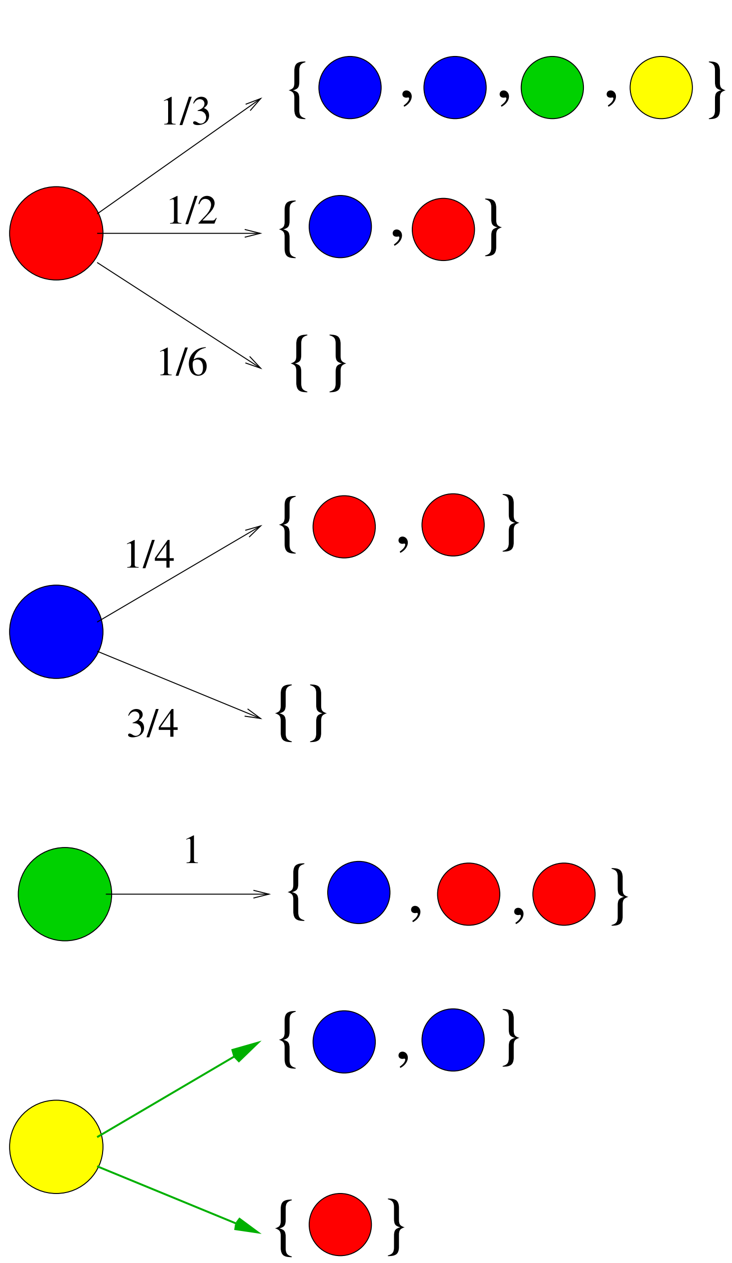


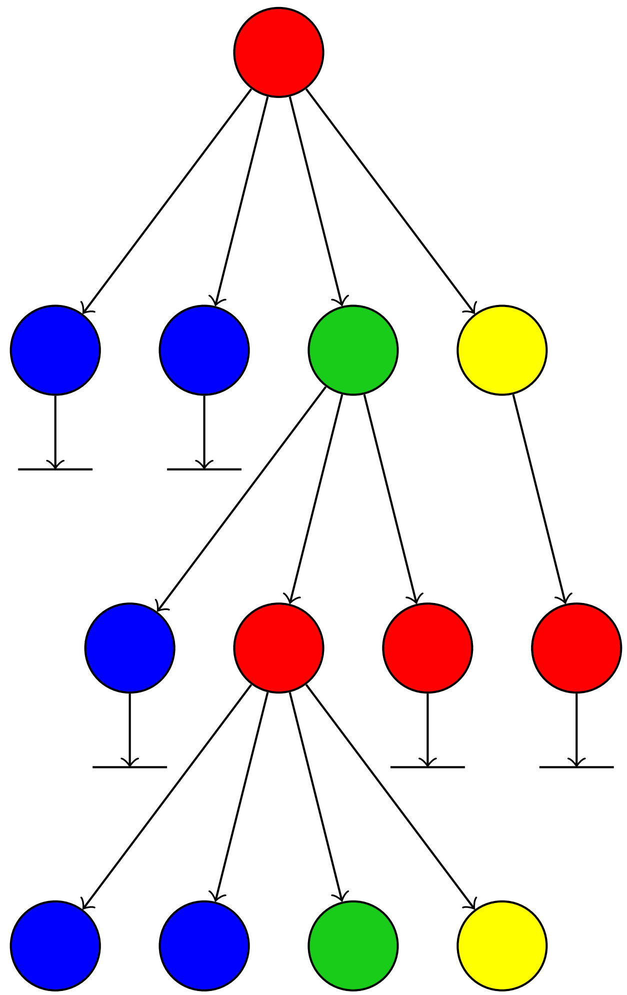
Branching Markov Decision Processes



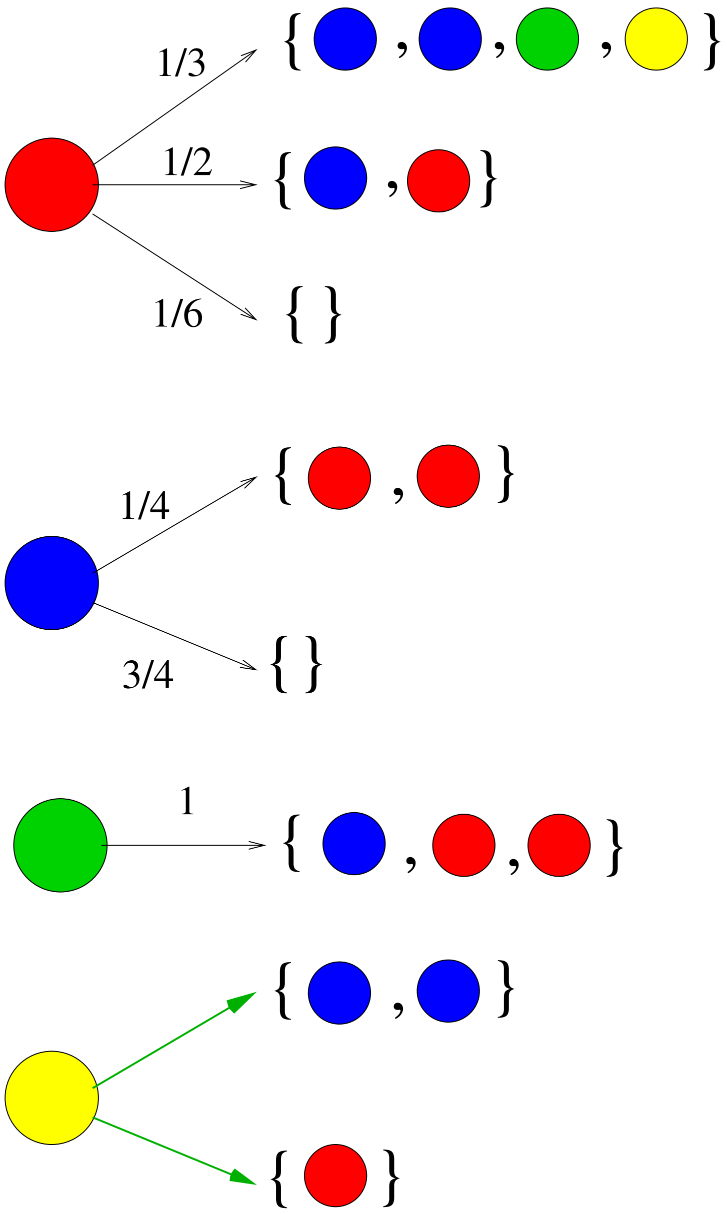


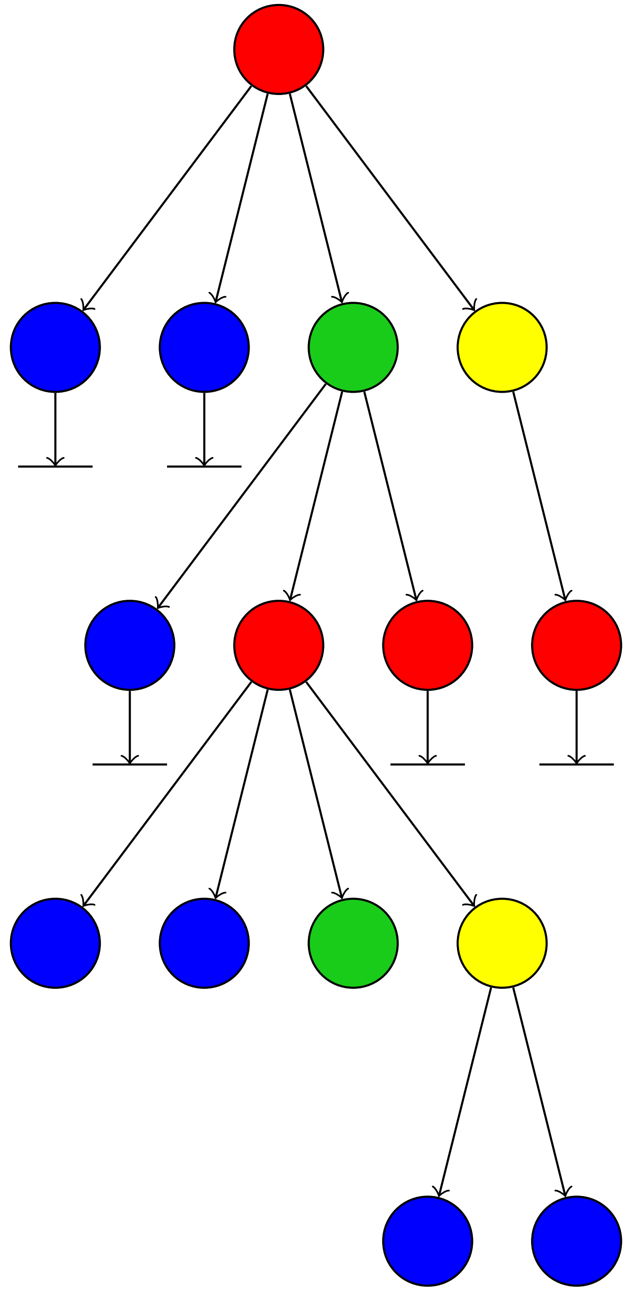
Branching Markov Decision Processes

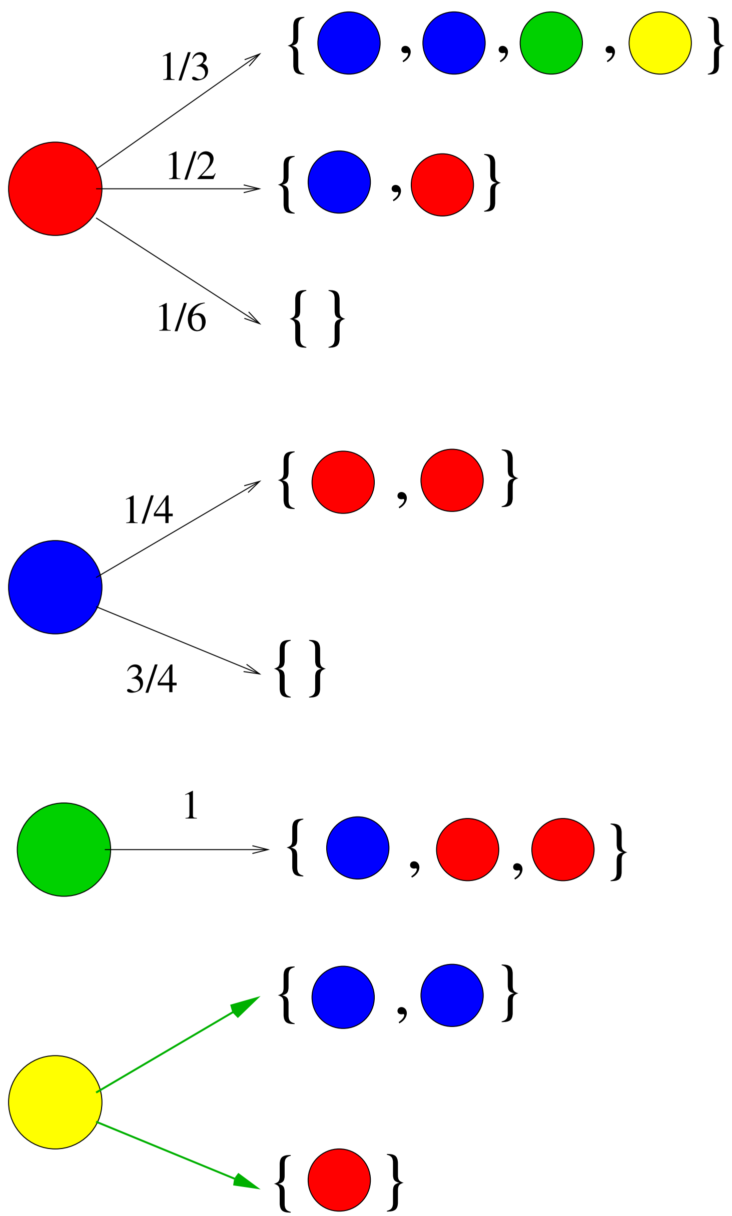




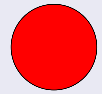
Branching Markov Decision Processes



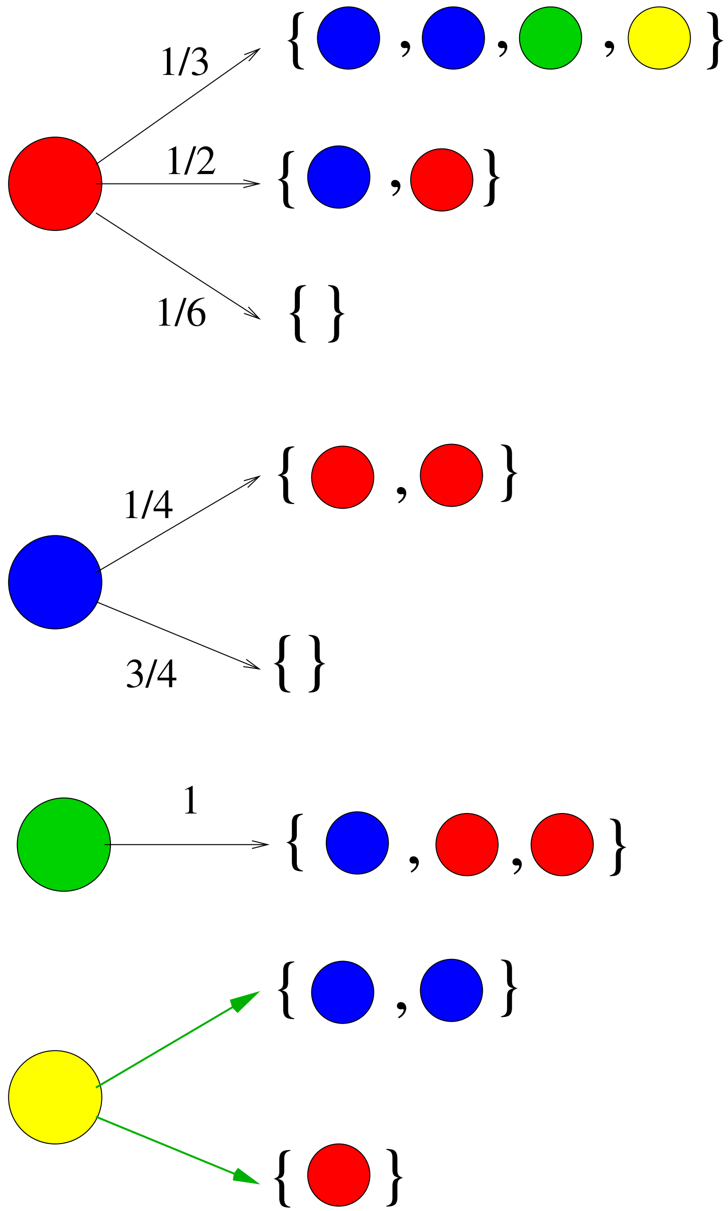




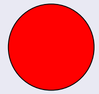
Question

What is the maximum probability of extinction, starting with one  ?

Branching Markov Decision Processes



Question

What is the maximum probability of extinction, starting with one  ?

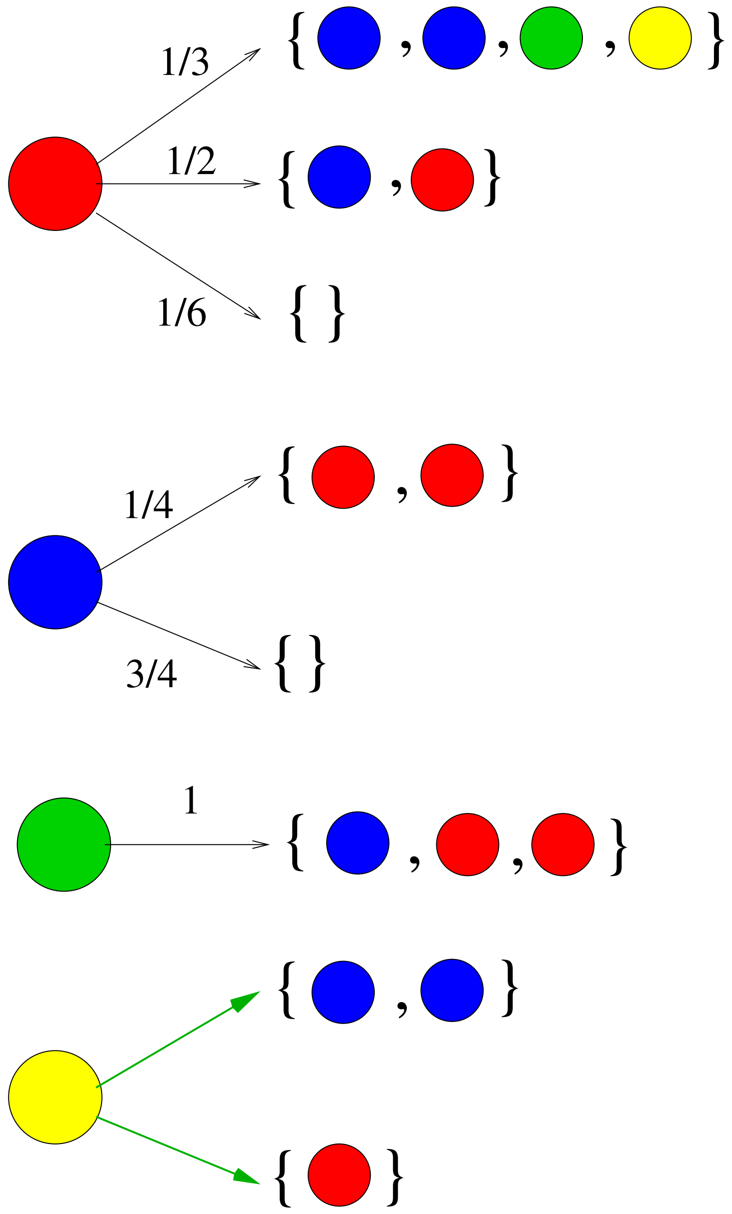
$$x_R = \frac{1}{3} x_B^2 x_G x_Y + \frac{1}{2} x_B x_R + \frac{1}{6}$$

$$x_B = \frac{1}{4} x_R^2 + \frac{3}{4}$$

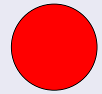
$$x_G = x_B x_R^2$$

$$x_Y =$$

Branching Markov Decision Processes



Question

What is the maximum probability of extinction, starting with one  ?

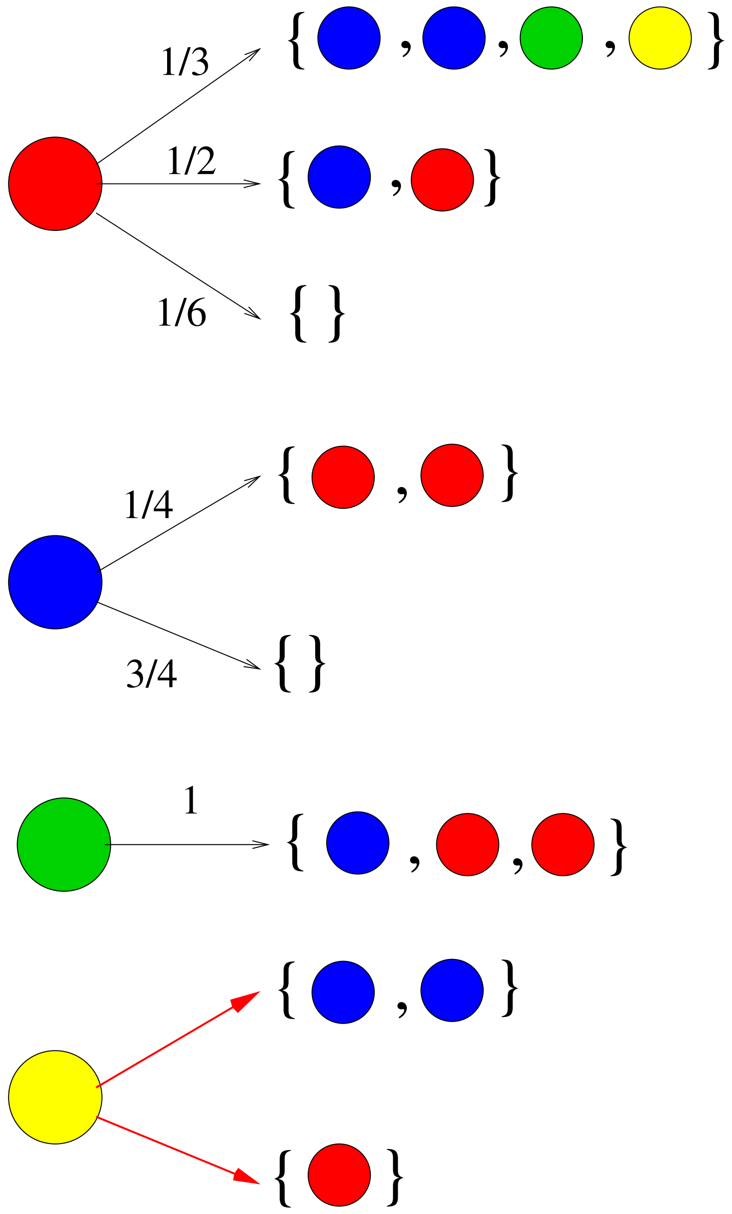
$$
\begin{aligned}
x_R &= \frac{1}{3}x_B^2 x_G x_Y + \frac{1}{2}x_B x_R + \frac{1}{6} \\
x_B &= \frac{1}{4}x_R^2 + \frac{3}{4} \\
x_G &= x_B x_R^2 \\
x_Y &= \max\{x_B^2, x_R\}
\end{aligned}
$$

We get fixed point equations, $\bar{\mathbf{x}} = P(\bar{\mathbf{x}})$.

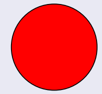
Fact [E.-Yannakakis'05]

The maximum extinction probabilities are the least fixed point, $\mathbf{q}^* \in [0,1]^3$, of $\bar{\mathbf{x}} = P(\bar{\mathbf{x}})$.

Branching Markov Decision Processes



Question

What is the minimum probability of extinction, starting with one  ?

$$
\begin{aligned}
x_R &= \frac{1}{3}x_B^2 x_G x_Y + \frac{1}{2}x_B x_R + \frac{1}{6} \\
x_B &= \frac{1}{4}x_R^2 + \frac{3}{4} \\
x_G &= x_B x_R^2 \\
x_Y &= \min\{x_B^2, x_R\}
\end{aligned}
$$

We get fixed point equations, $\bar{\mathbf{x}} = P(\bar{\mathbf{x}})$.

Fact [E.-Yannakakis'05]

The minimum extinction probabilities are the least fixed point, $\mathbf{q}^* \in [0,1]^3$, of $\bar{\mathbf{x}} = P(\bar{\mathbf{x}})$.

A Maximum Probabilistic Polynomial System (maxPPS) is a system

$$\mathbf{x}_i = \max\{p_{i,j}(\mathbf{x}) : j = 1, \ldots, m_i\} \qquad i = 1, \ldots, n$$

of n equations in n variables, where each $p_{i,j}(x)$ is a probabilistic polynomial. We denote the entire system by:

$$\mathbf{x} = P(\mathbf{x})$$

Minimum Probabilistic Polynomial Systems (minPPSs) are defined similarly.

These are Bellman optimality equations for maximizing (minimizing) extinction probabilities in a BMDP.

We use max/minPPS to refer to either a maxPPS or an minPPS.

Basic properties of max/minPPSs, $\mathbf{x} = P(\mathbf{x})$

$P : [0,1]^n \rightarrow [0,1]^n$ defines a monotone map on $[0,1]^n$.

Proposition. [E.-Yannakakis'05]

- *Every max/minPPS, $x = P(x)$ has a least fixed point, $q^* \in [0,1]^n$.*
- $q^* = \lim_{k\rightarrow\infty} P^k(\mathbf{0})$.
- *q^* is vector of optimal extinction probabilities for the BMDP.*

Question

Can we compute the probabilities q^* efficiently (in P-time)?

P-time approximation for BMDPs and max/minPPSs

Theorem ([E.-Stewart-Yannakakis,ICALP'12])

Given a max/minPPS, $\mathbf{x} = P(\mathbf{x})$, with LFP $\mathbf{q}^* \in [0,1]^n$, we can compute a rational vector $\mathbf{v} \in [0,1]^n$ such that

$$\|\mathbf{v} - \mathbf{q}^*\|_\infty \leq 2^{-j}$$

in time polynomial in the encoding size $|P|$ of the equations, and in j.

We establish this via a Generalized Newton's Method that uses linear programming in each iteration.

Newton iteration as a first-order (Taylor) approximation

An iteration of Newton's method on a PPS, applied on current vector $y \in \mathbb{R}^n$, solves the equation

$$P^{\mathbf{y}}(\mathbf{x}) = \mathbf{x}$$

where $P^{\mathbf{y}}(\mathbf{x}) \equiv P(\mathbf{y}) + P'(\mathbf{y})(\mathbf{x} - \mathbf{y})$ is a linear (first-order Taylor) approximation of $P(x)$.

Generalised Newton's method

Linearisation

Given a maxPPS

$$(P(\mathbf{x}))_i = \max\{p_{i,j}(\mathbf{x}) : j = 1, \ldots, m_i\} \qquad i = 1, \ldots, n$$

We define the linearisation, $P^y(x)$, by:

$$(P^{\mathbf{y}}(\mathbf{x}))_i = \max\{p_{i,j}(\mathbf{y}) + \nabla p_{i,j}(\mathbf{y}).(\mathbf{x} - \mathbf{y}) : j = 1, \ldots, m_i\} \qquad i = 1, \ldots, n$$

Generalised Newton's method

Linearisation

Given a maxPPS

$$(P(\mathbf{x}))_i = \max\{p_{i,j}(\mathbf{x}) : j = 1, \ldots, m_i\} \qquad i = 1, \ldots, n$$

We define the linearisation, $P^y(x)$, by:

$$(P^{\mathbf{y}}(\mathbf{x}))_i = \max\{p_{i,j}(\mathbf{y}) + \nabla p_{i,j}(\mathbf{y}).(\mathbf{x} - \mathbf{y}) : j = 1, \ldots, m_i\} \qquad i = 1, \ldots, n$$

Generalised Newton's method, applied at vector y

For a maxPPS, minimize $\sum_i x_i$ subject to $P^{\mathbf{y}}(\mathbf{x}) \leq \mathbf{x}$;

For a minPPS, maximize $\sum_i x_i$ subject to $P^{\mathbf{y}}(\mathbf{x}) \geq \mathbf{x}$;

These can both be phrased as linear programming problems. Their optimal solution solves $P^{\mathbf{y}}(\mathbf{x}) = \mathbf{x}$, and yields the GNM iteration we need.

Algorithm for max/minPPSs

1. Find and remove all variables x_i such that $q_i^* = 0$ or $q_i^* = 1$. ($q_i^* = 1$ decidable in P-time using LP [E.-Yannakakis'06]: reduces to a spectral radius optimization problem for non-negative square matrices.)

Algorithm for max/minPPSs

1. Find and remove all variables x_i such that $q_i^* = 0$ or $q_i^* = 1$.
   ($q_i^* = 1$ decidable in P-time using LP [E.-Yannakakis'06]: reduces to a spectral radius optimization problem for non-negative square matrices.)
2. On the resulting system of equations, run Generalized Newton's Method, starting from $\mathbf{0}$. After each iteration, round down to a multiple of 2^{-h}.
   Each iteration of GNM can be computed in P-time by solving an LP.

Algorithm for max/minPPSs

1. Find and remove all variables x_i such that $q_i^* = 0$ or $q_i^* = 1$.
   ($q_i^* = 1$ decidable in P-time using LP [E.-Yannakakis'06]: reduces to a spectral radius optimization problem for non-negative square matrices.)
2. On the resulting system of equations, run Generalized Newton's Method, starting from $\mathbf{0}$. After each iteration, round down to a multiple of 2^{-h}.
   Each iteration of GNM can be computed in P-time by solving an LP.

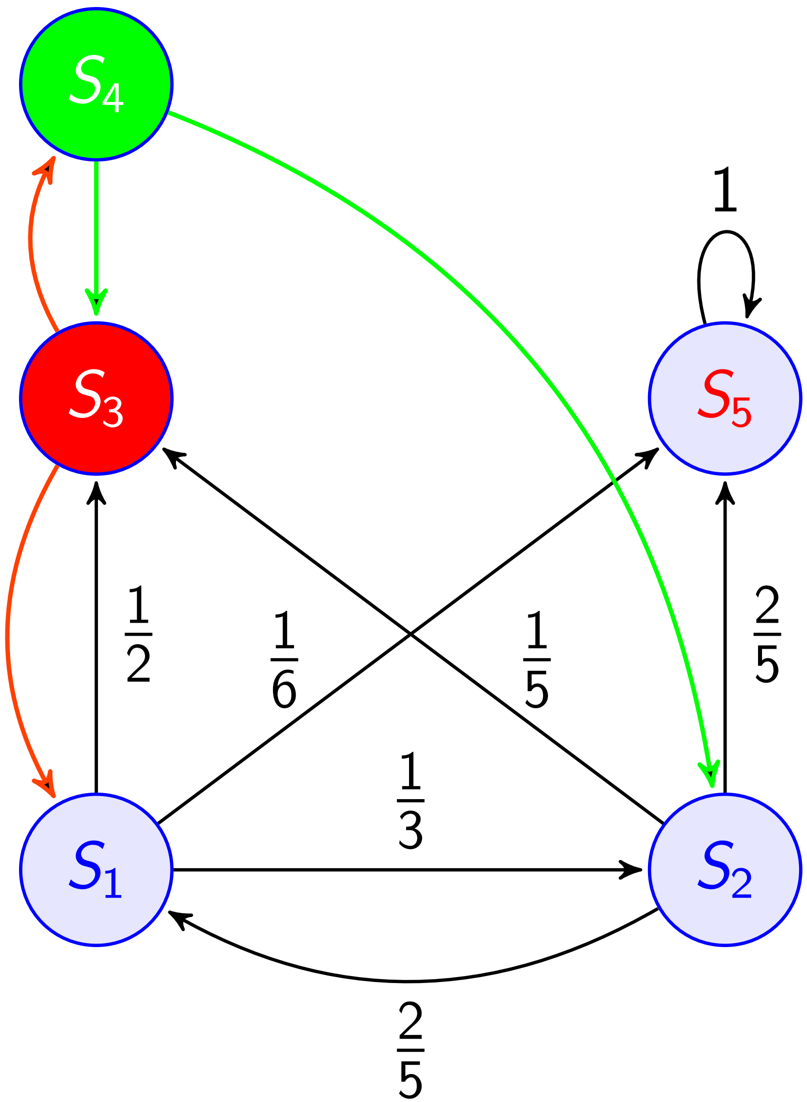
Theorem [ESY'12]

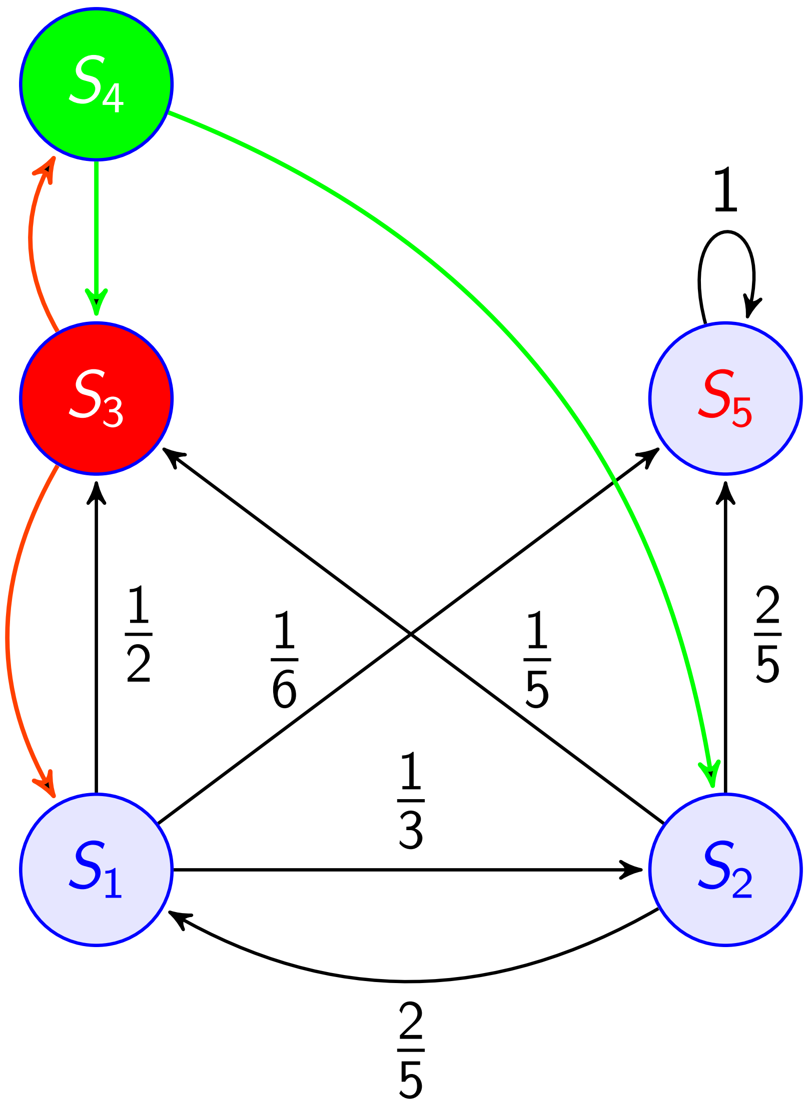
Given a max/minPPS $\mathbf{x} = P(\mathbf{x})$ with LFP $\mathbf{0} < \mathbf{q}^* < \mathbf{1}$, if we apply rounded GNM starting at $\mathbf{x}^{(0)} = \mathbf{0}$, using $h := 4|P| + j + 1$ bits of precision, then

$$\|\mathbf{q}^* - \mathbf{x}^{(4|P|+j+1)}\|_\infty \leq 2^{-j}.$$

We can do all this in time polynomial in $|P|$ and j.

finite-state Simple Stochastic Games

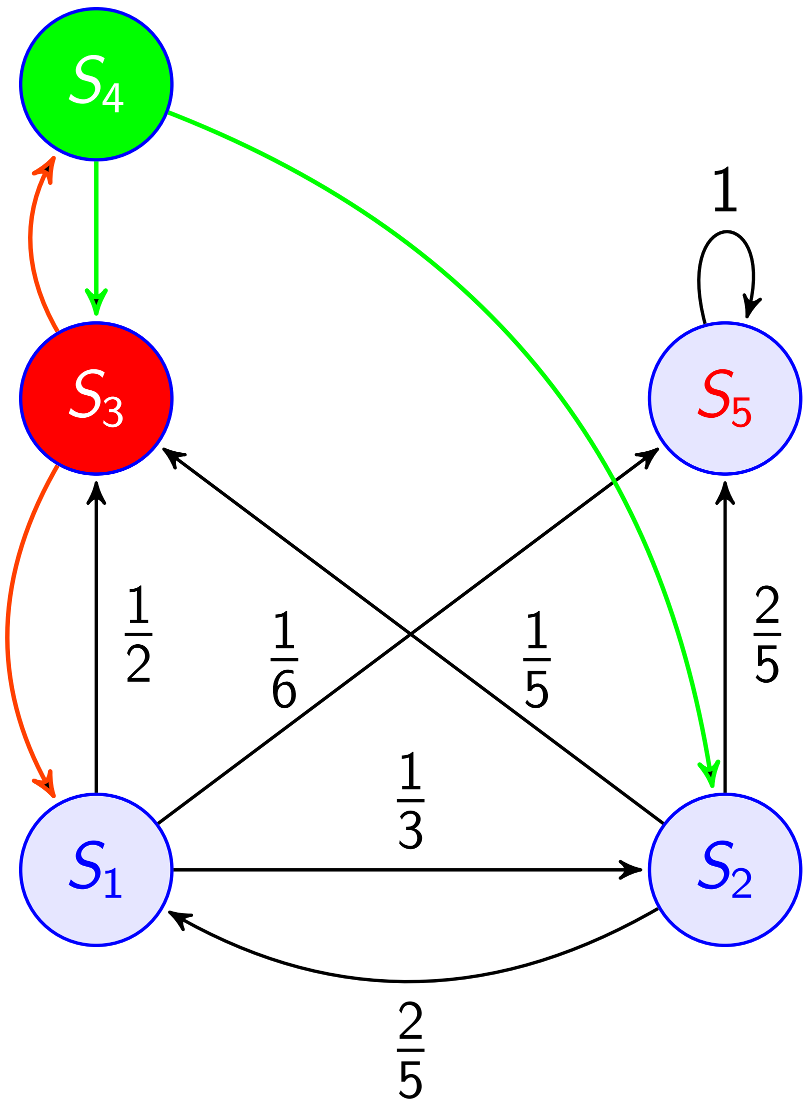




What is the value of the game for hitting S_5 starting at S_1?
(These games are determined.)

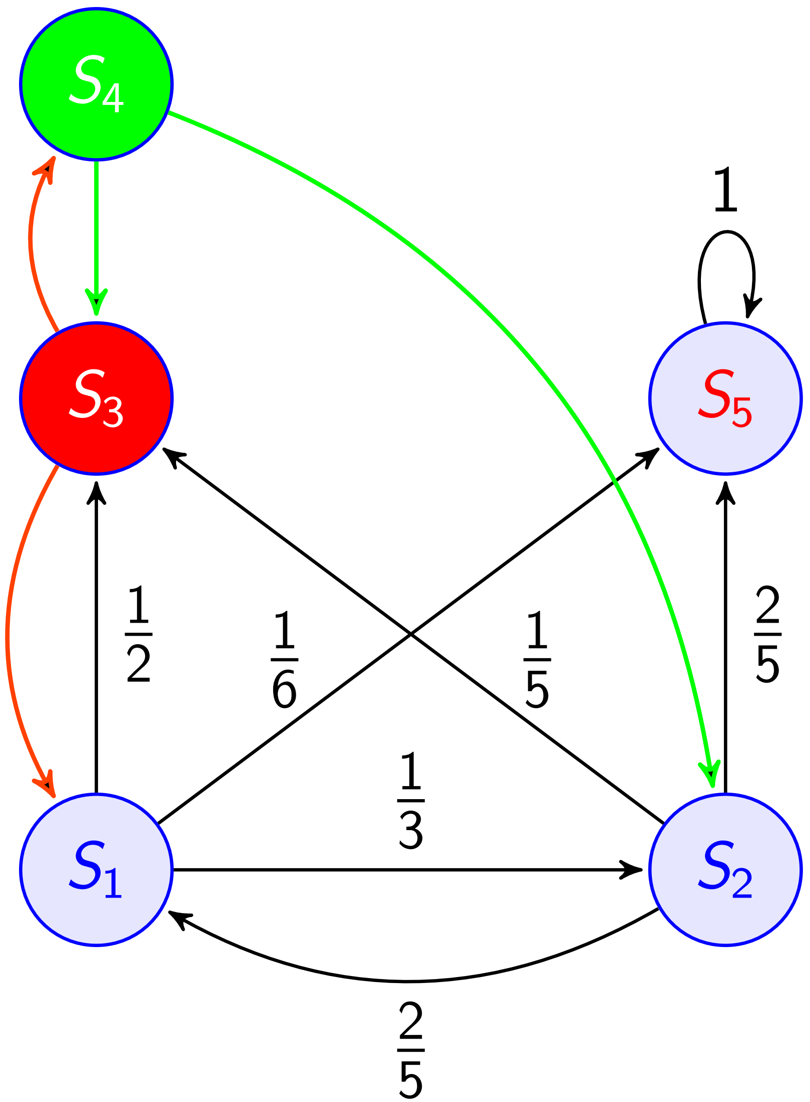
$$x_1 \quad =$$

finite-state Simple Stochas



What is the value of the game for hitting S_5 starting at S_1?
(These games are determined.)

$$
\begin{aligned}
x_1 &= \frac{1}{3}x_2 + \frac{1}{2}x_3 + \frac{1}{6} \\
x_2 &= \frac{2}{5}x_1 + \frac{1}{5}x_3 + \frac{2}{5} \\
x_3 &= \max\{x_1, x_4\} \\
x_4 &= \min\{x_2, x_3\}
\end{aligned}
$$

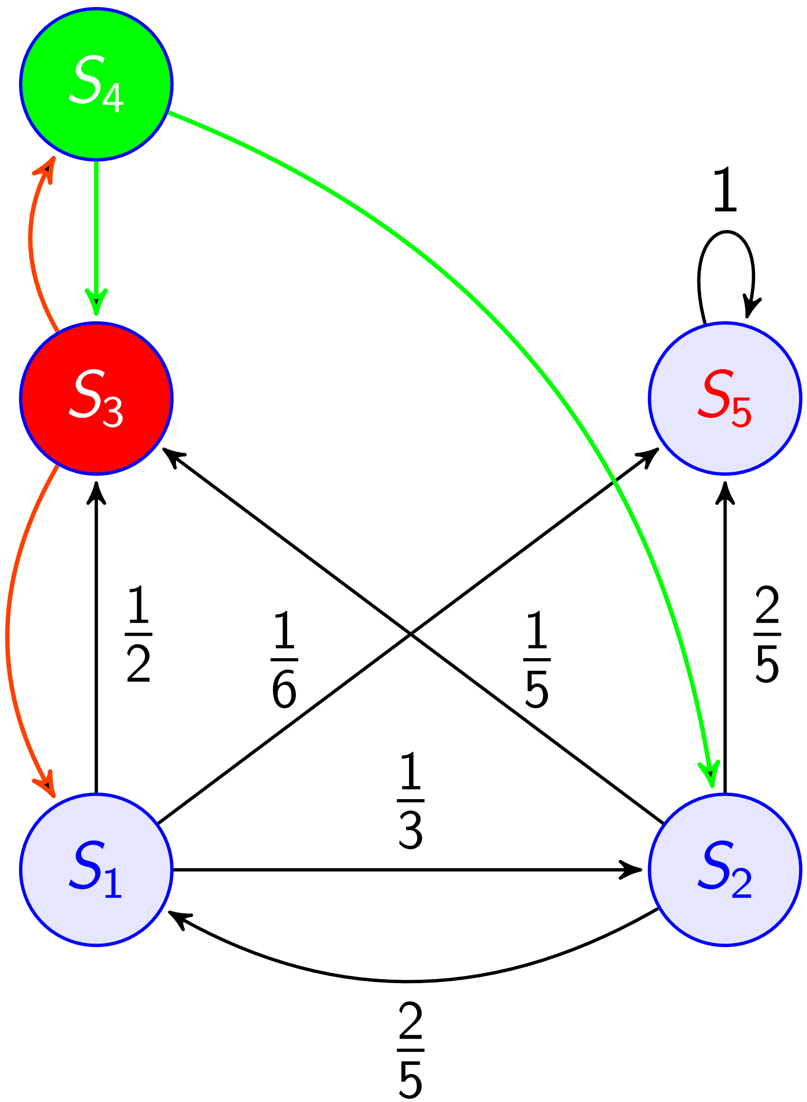


What is the value of the game for hitting S_5 starting at S_1?
(These games are determined.)

$$\begin{aligned} x_1 &= \frac{1}{3}x_2 + \frac{1}{2}x_3 + \frac{1}{6} \\ x_2 &= \frac{2}{5}x_1 + \frac{1}{5}x_3 + \frac{2}{5} \\ x_3 &= \max\{x_1, x_4\} \\ x_4 &= \min\{x_2, x_3\} \end{aligned}$$

We get linear-min-max equations,

$$\bar{\mathbf{x}} = P(\bar{\mathbf{x}}).$$



What is the value of the game for hitting S_5 starting at S_1?
(These games are determined.)

$$
\begin{aligned}
x_1 &= \frac{1}{3}x_2 + \frac{1}{2}x_3 + \frac{1}{6} \\
x_2 &= \frac{2}{5}x_1 + \frac{1}{5}x_3 + \frac{2}{5} \\
x_3 &= \max\{x_1, x_4\} \\
x_4 &= \min\{x_2, x_3\}
\end{aligned}
$$

We get linear-min-max equations,

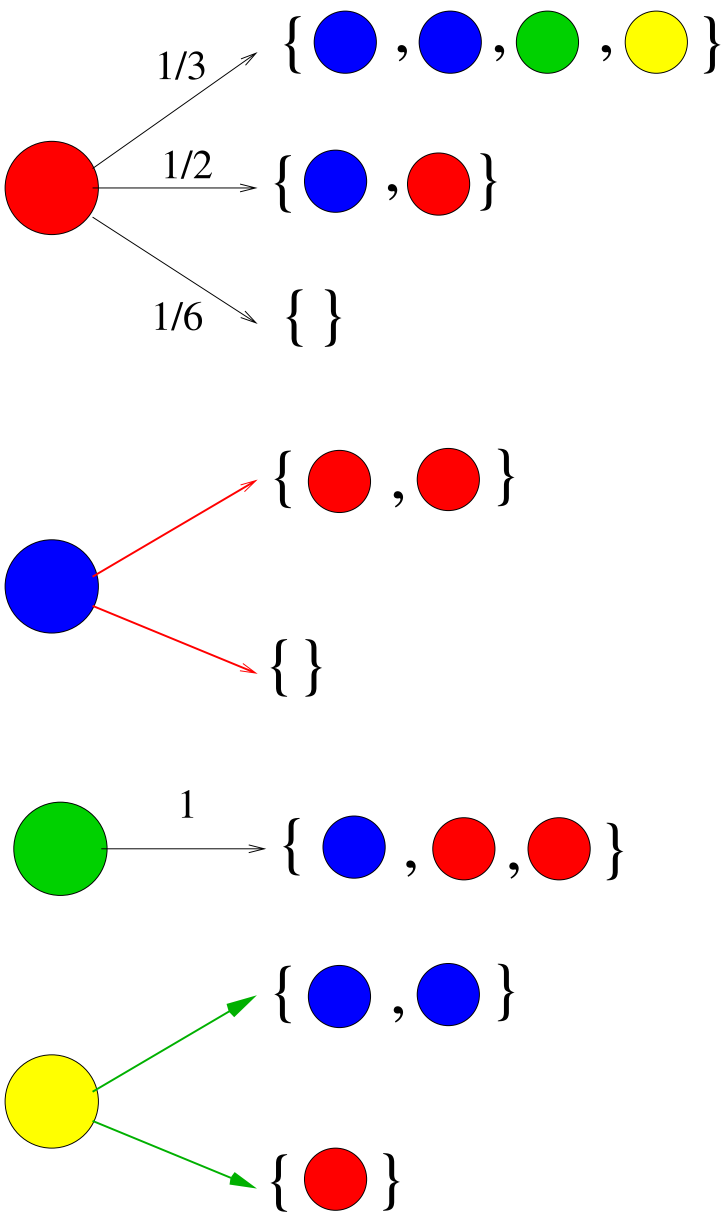
$$\bar{\mathbf{x}} = P(\bar{\mathbf{x}}).$$

Fact: [Shapley'53,Condon'92]
Hitting values are the least fixed point, $q^* \in [0,1]^4$, of $\mathbf{x} = P(\mathbf{x})$.

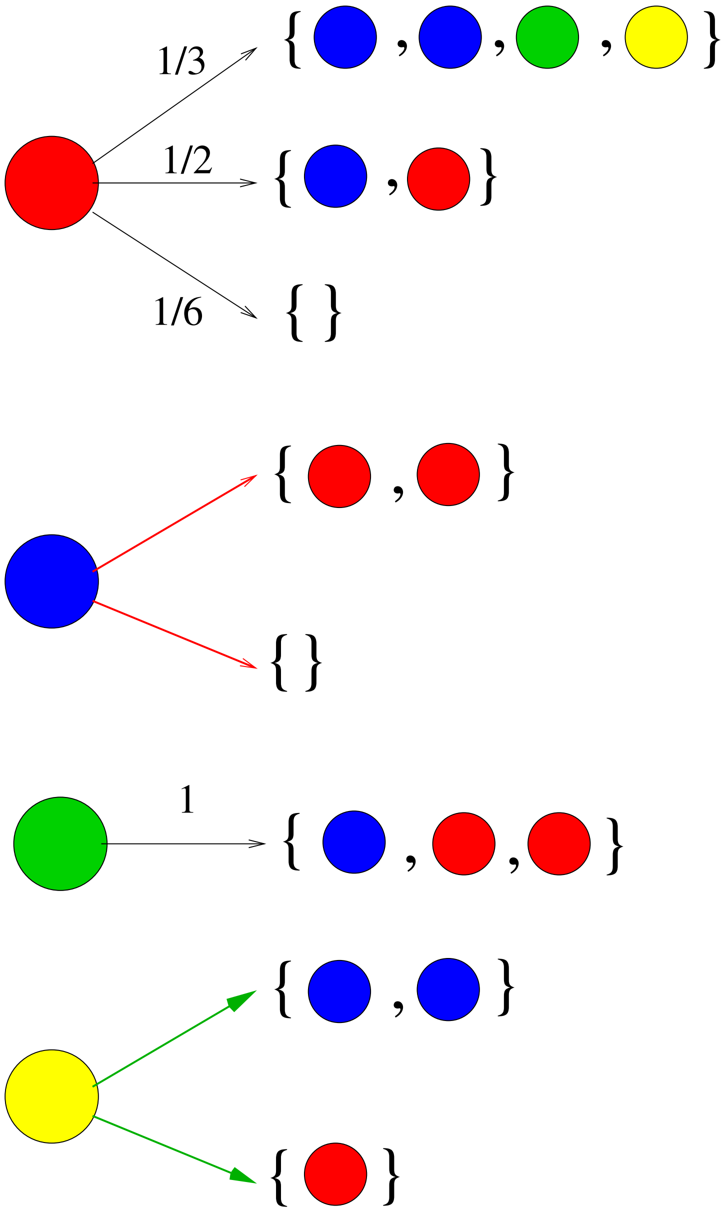
- In any finite-state SSG, both max and min, have optimal positional strategies (i.e., deterministic and memoryless optimal strategies).
- Thus [Condon'92]: deciding whether the game value $q^*_i \leq 1/2$, is in **NP** $\cap$ **coNP**.

  And computing the (exact, rational) values q^* is in **FNP**.
- Long standing open problem whether SSGs are solvable in **P**-time. (Subsumes parity games and mean payoff games.)

Branching Simple Stochastic Games



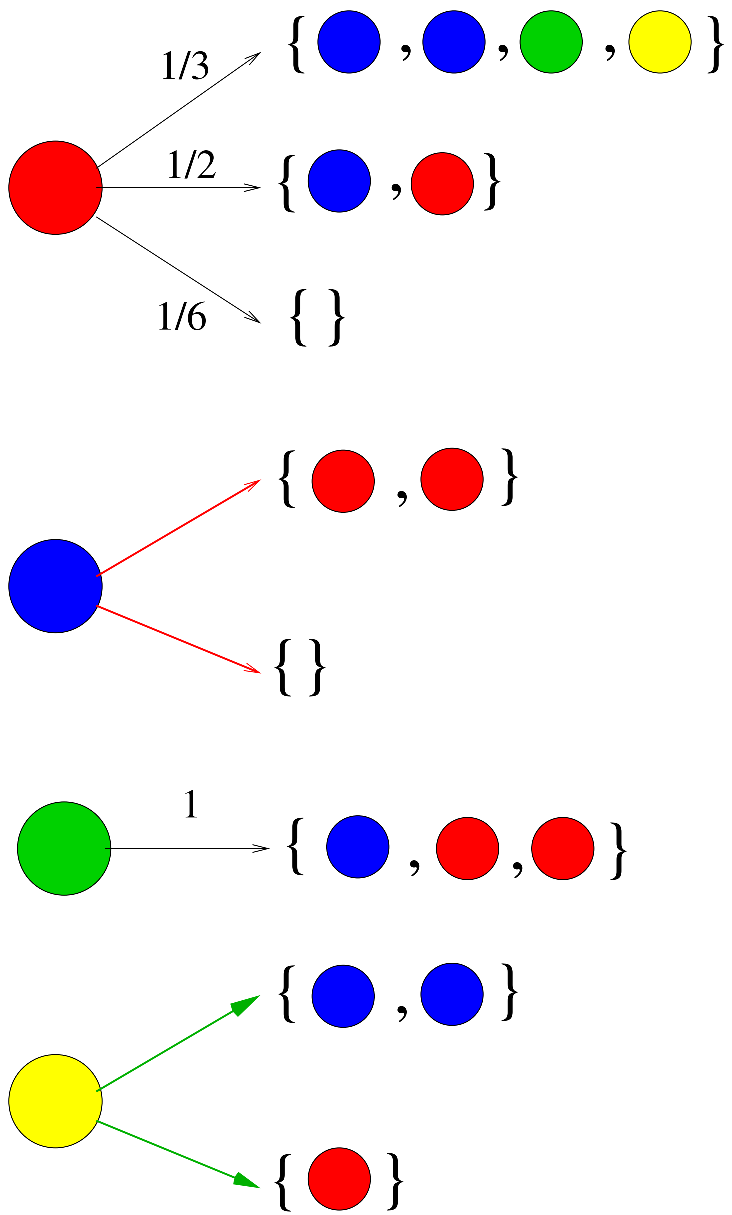
Branching Simple Stochastic Games



Types belonging to min:

Types belonging to max:

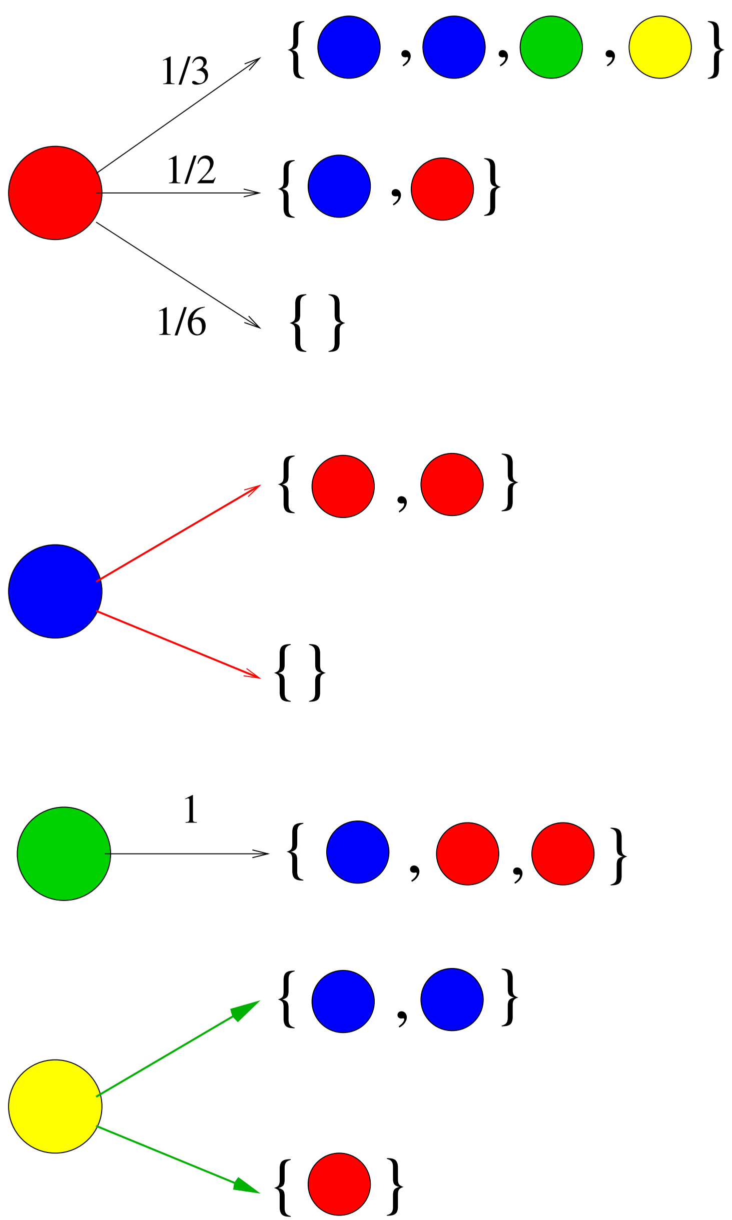
Branching Simple Stochastic Games



Question

What is the value of extinction, starting with one ?

Branching Simple Stochastic Games



Question

What is the value of extinction, starting with one ?

$$
\begin{aligned}
x_R &= \frac{1}{3}x_B^2 x_G x_Y + \frac{1}{2}x_B x_R + \frac{1}{6}\\
x_B &= \min\{x_R^2, 1\}\\
x_G &= x_B x_R^2\\
x_Y &= \max\{x_B^2, x_R\}
\end{aligned}
$$

We get fixed point equations, $\bar{\mathbf{x}} = P(\bar{\mathbf{x}})$.

Fact [E.-Yannakakis'05]

The extinction values are the LFP, $\mathbf{q}^* \in [0,1]^3$ of $\bar{\mathbf{x}} = P(\bar{\mathbf{x}})$.

Qualitative and Quantitative problems for BSSGs

Theorem ([E.-Yannakakis'05])

For any BSSG, both players have static positional optimal strategies for maximizing (minimizing) extinction probability.

A static positional strategy is one that, for every type belonging to the player, always deterministically chooses the same single rule.
(i.e., it is deterministic, memoryless, and "context-oblivious".)

Theorem ([E.-Yannakakis'06])

Given a BSSG, deciding if the extinction value is $q^*_i = 1$ is in **NP** $\cap$ **coNP**, *& is at least as hard as computing the exact value for a finite-state SSG.*

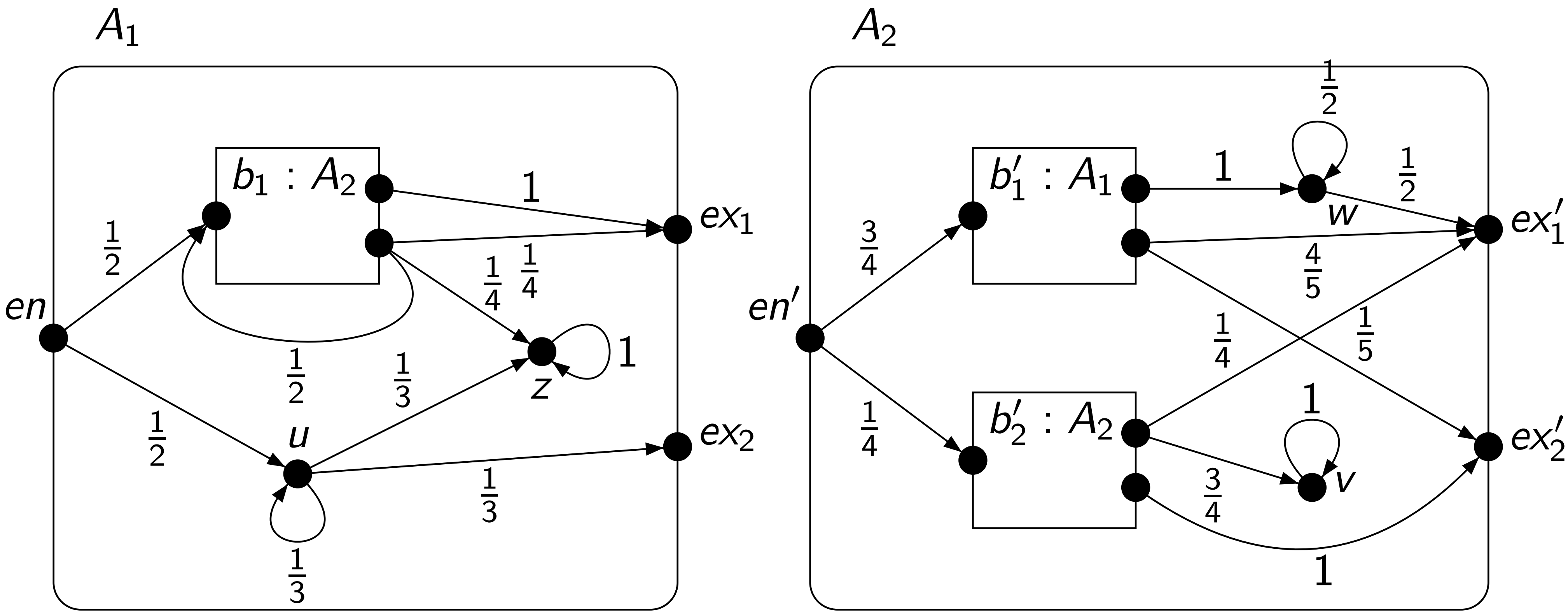
Theorem ([ESY'12])

Given a BSSG, and given $\epsilon > 0$, we can compute a vector $v \in [0,1]^n$, such that $\|v - q^*\|_\infty \leq \epsilon$, in **FNP**.

One piece of a larger story

- Many other analyses: expected total reward, discounted reward, expected limiting average reward, model checking.
- Many analyses require termination probabilities q^* as a prerequisite, but they also require non-trivial additional work.
- Recursive Markov Chains (RMCs) form a more general class of countable infinite-state discrete-time MCs. (BPs and SCFGs correspond to 1-exit RMCs.)

Recursive Markov Chain



- RMCs also have MPSs (not PPSs) whose LFP $q^* \in [0,1]^n$ gives their termination probabilities.
- However, any non-trivial approximation of q^* for RMCs is PosSLP-hard ([E.-Yannakakis'07]).
- For RMDPs and RSSGs any non-trivial approximation of their value vector is uncomputable! ([E.-Yannakakis'05]).

- But other subclasses of RMCs, corresponding to other important stochastic processes, are analyzable.
- 1-box RMCs correspond to (discrete-time) Quasi-Birth-Death processes (QBDs), and to probabilistic one-counter automata (OC-MCs).
- For QBDs we can approximate q^* in P-time ([E.-Wojtczak-Yannakakis'08], [Stewart-E.-Yannakakis'13]).
- Many problems for OC-MDPs and OC-SSGs are also decidable ([Brazdil-Brozek-E.-Kucera-Wojtczak'10,'10,'11]), but for many we don't know good complexity bounds.

Conclusion

- A very rich landscape, with still many open questions.
- Can we solve finite-state SSGs in **P**-time?
- Can we obtain any better upper bounds for PosSLP??
- Deciding $q^* \geq 1/2$ for Branching SSGs subsumes both of these problems.