The computational complexity of analyzing infinite-state structured Markov Chains and structured MDPs

Kousha Etessami

University of Edinburgh

Based mainly on joint works with:

Alistair Stewart & Mihalis Yannakakis
U. of Edinburgh (now USC) Columbia Uni.

MAM-9 Conference
Budapest, July 2016

Why am I bringing coals to Newcastle??

- Infinite-state but finitely-presented ("structured") Markov chains, and numerical methods for them, have been studied for a long time in the MAM community.

Why am I bringing coals to Newcastle??

- Infinite-state but finitely-presented ("structured") Markov chains, and numerical methods for them, have been studied for a long time in the MAM community.

- In the last decade there has also been a substantial body of (independent) research in the theoretical computer science and probabilistic verification community, focused on the **computational complexity** of analyzing such stochastic models, as well as generalizations of them to Markov decision processes (MDPs) and stochastic games.

Why am I bringing coals to Newcastle??

- Infinite-state but finitely-presented ("structured") Markov chains, and numerical methods for them, have been studied for a long time in the MAM community.

- In the last decade there has also been a substantial body of (independent) research in the theoretical computer science and probabilistic verification community, focused on the **computational complexity** of analyzing such stochastic models, as well as generalizations of them to Markov decision processes (MDPs) and stochastic games.

- In this talk I hope to give you a flavor of this research in TCS.

  (I can not be comprehensive: it is by now a rich body of work.)

Why am I bringing coals to Newcastle??

- Infinite-state but finitely-presented ("structured") Markov chains, and numerical methods for them, have been studied for a long time in the MAM community.
- In the last decade there has also been a substantial body of (independent) research in the theoretical computer science and probabilistic verification community, focused on the **computational complexity** of analyzing such stochastic models, as well as generalizations of them to Markov decision processes (MDPs) and stochastic games.
- In this talk I hope to give you a flavor of this research in TCS.

  (I can not be comprehensive: it is by now a rich body of work.)
- I hope my talk will help foster more interactions between the MAM community and those doing related research in TCS and verification.

Overview of the talk

- I will focus mainly on a series of results we have obtained on the complexity of analyzing the following models (in discrete time):
  - **Multi-type Branching Processes** (a.k.a., Markovian Trees), and their generalization: **Branching MDPs**.
  - **One-counter Markov Chains** (a.k.a., QBDs), and **one-counter MDPs**.
  - **Recursive Markov Chains** (a.k.a., tree-structured/tree-like-QBDs), and **Recursive MDPs**.
- A key aspect of our results: new algorithmic bounds for computing the **least fixed point** (the least non-negative solution) for **monotone systems of (min/max)-polynomial equations**.

  Such equations arise for various stochastic models and MDPs (e.g., as their Bellman optimality equations).

A word about traditional numerical analysis vs. computational complexity analysis

- In **numerical analysis** it is often typical to establish "linear/quadratic convergence" for an iterative algorithm.

  This provides upper bounds on the number of iterations required to achieve desired accuracy $\epsilon > 0$, as a function of ϵ, but in general it does not provide any bounds as a function of the encoding size of the input equations.

A word about traditional numerical analysis vs. computational complexity analysis

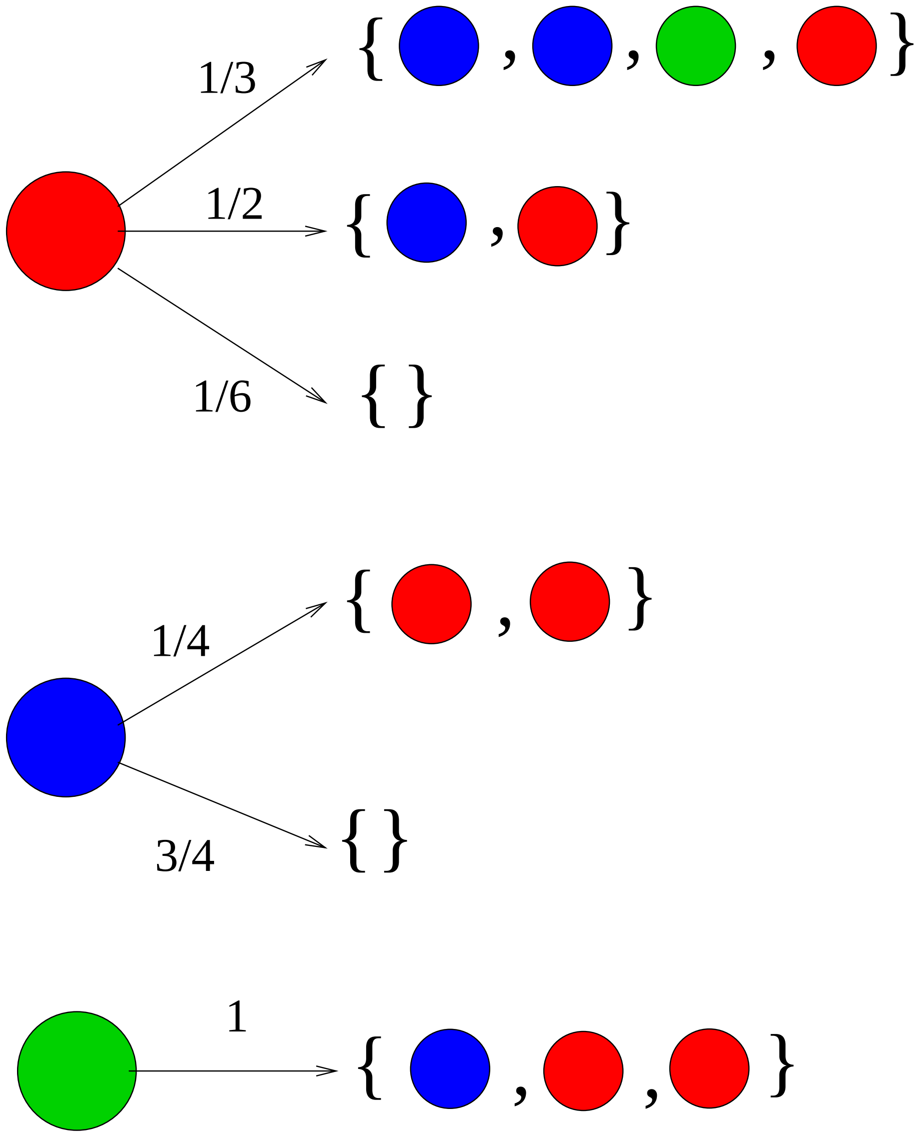
- In **numerical analysis** it is often typical to establish "linear/quadratic convergence" for an iterative algorithm.

  This provides upper bounds on the number of iterations required to achieve desired accuracy $\epsilon > 0$, as a function of ϵ, but in general it does not provide any bounds as a function of the encoding size of the input equations.

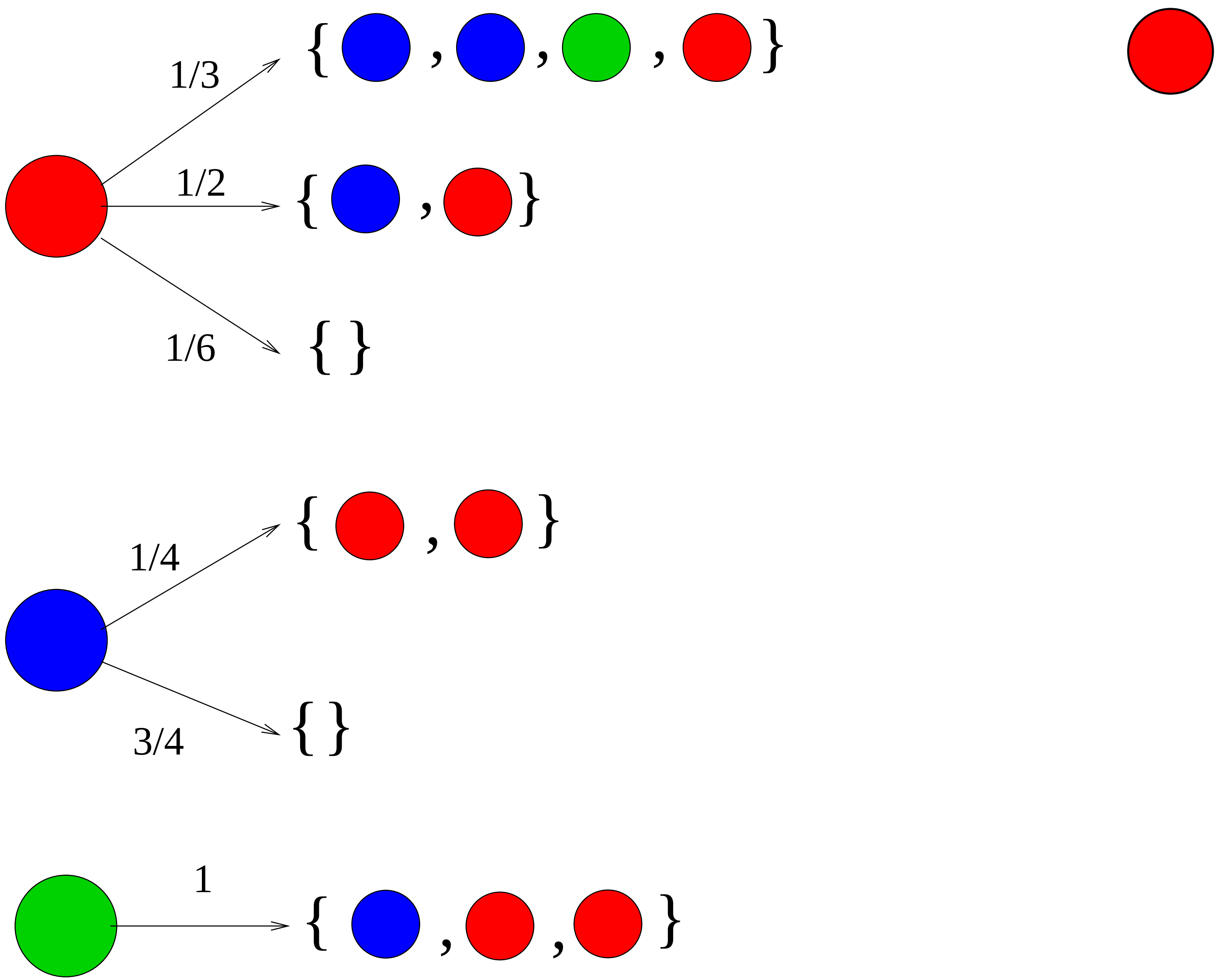
- By contrast, **computational complexity analysis** aims to bound the running time (hopefully polynomially or better) as a function of both the encoding size of the input system of equations and $\log(1/\epsilon)$.

  We aim for **worst case** complexity analysis, in the **standard Turing model** of computation, not in the unit-cost arithmetic model (a.k.a. BSS model), so no hiding of consequences of roundoff errors.

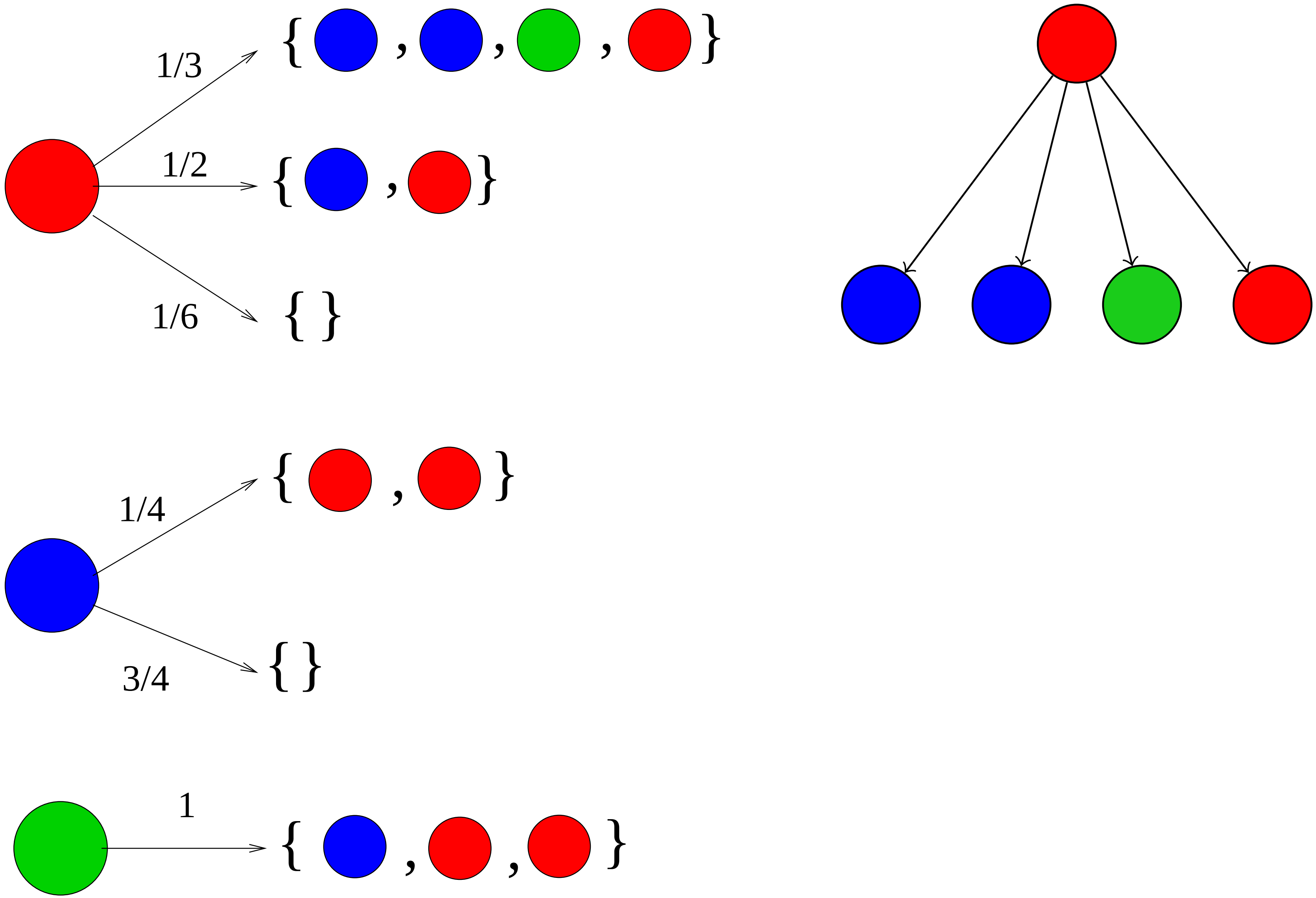
Multi-type Branching Processes (BPs) (Kolmogorov,1940s)



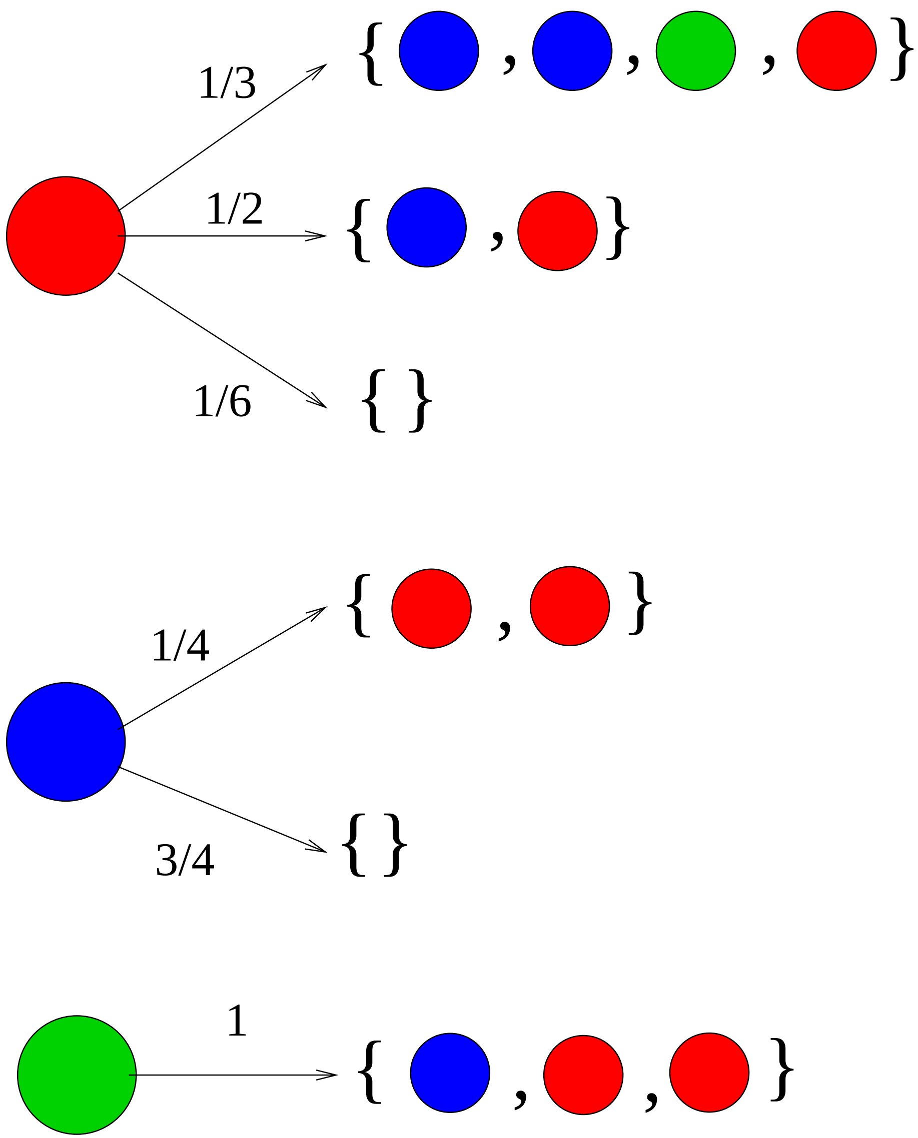
Multi-type Branching Processes (BPs) (Kolmogorov,1940s)

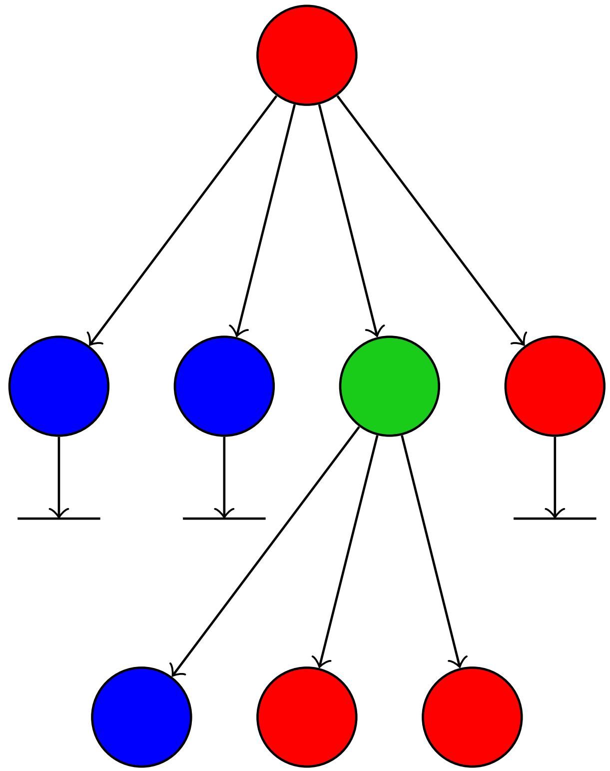


Multi-type Branching Processes (BPs) (Kolmogorov,1940s)

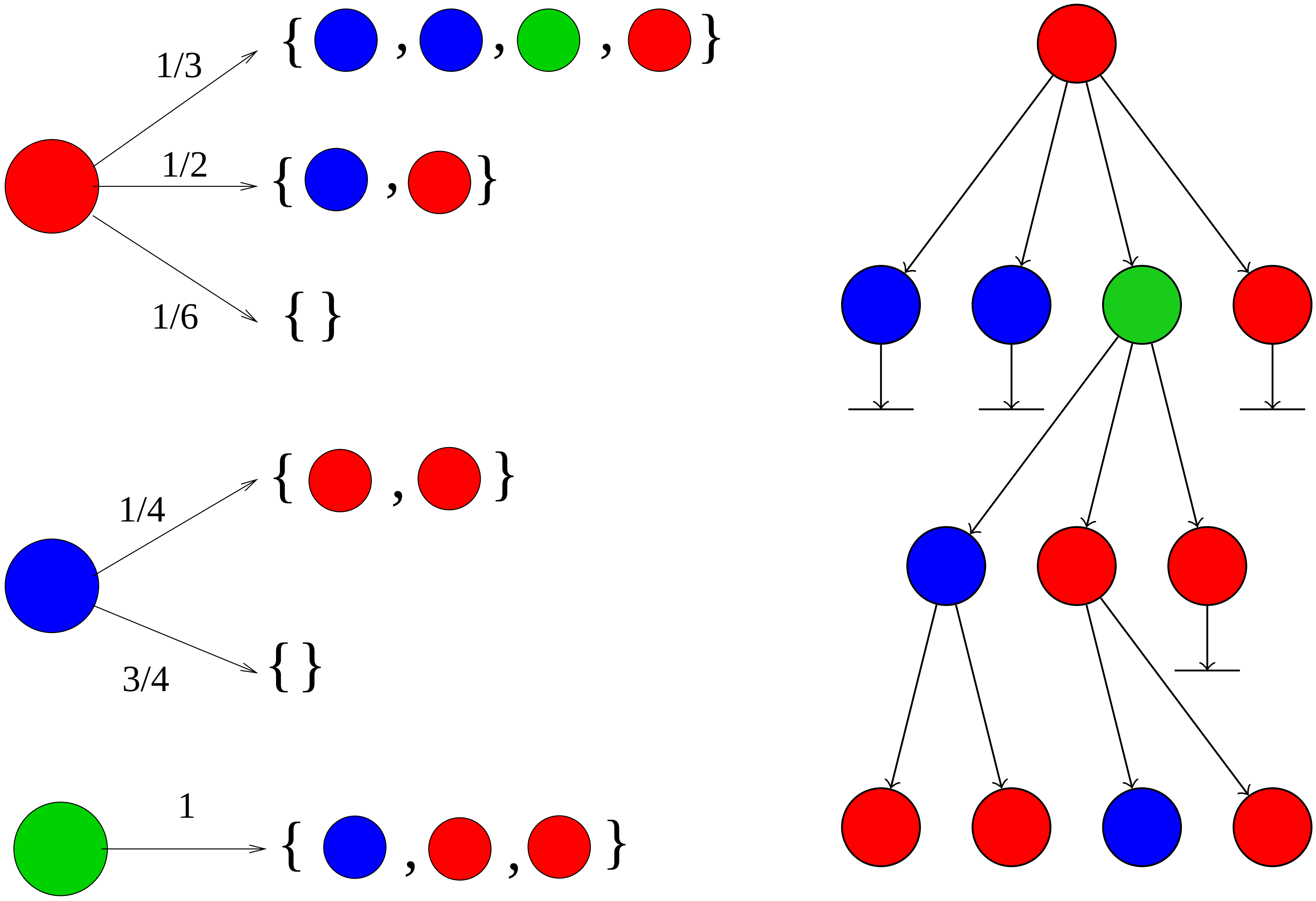


Multi-type Branching Processes (BPs) (Kolmogorov,1940s)

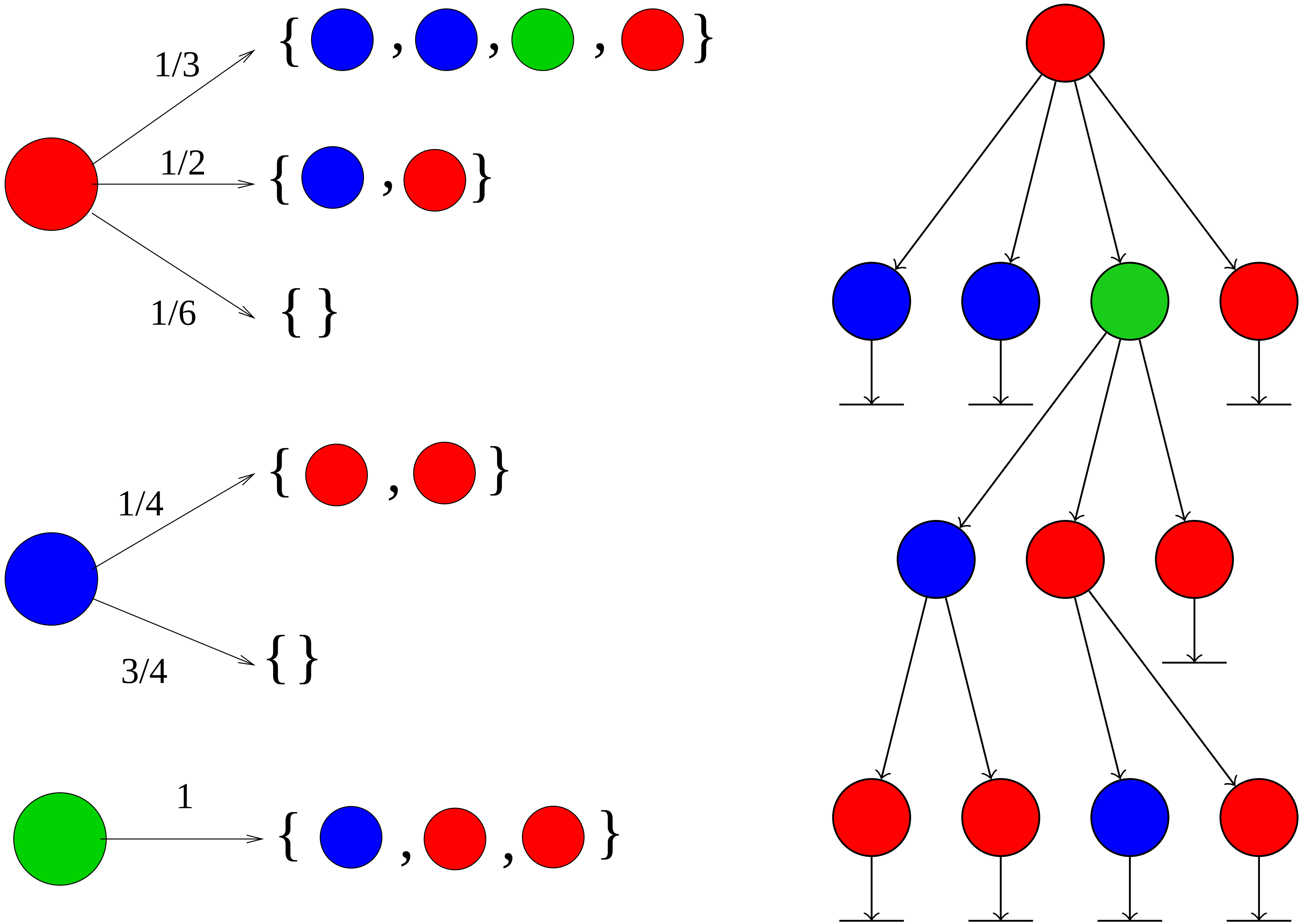




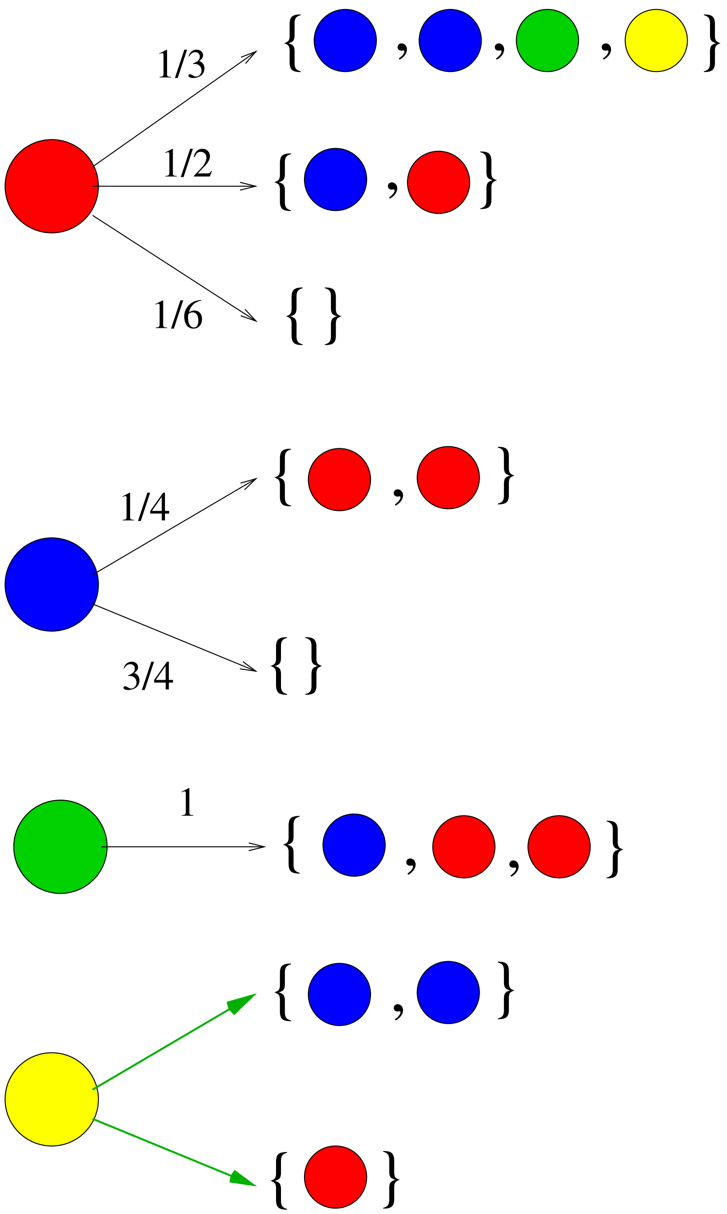
Multi-type Branching Processes (BPs) (Kolmogorov,1940s)



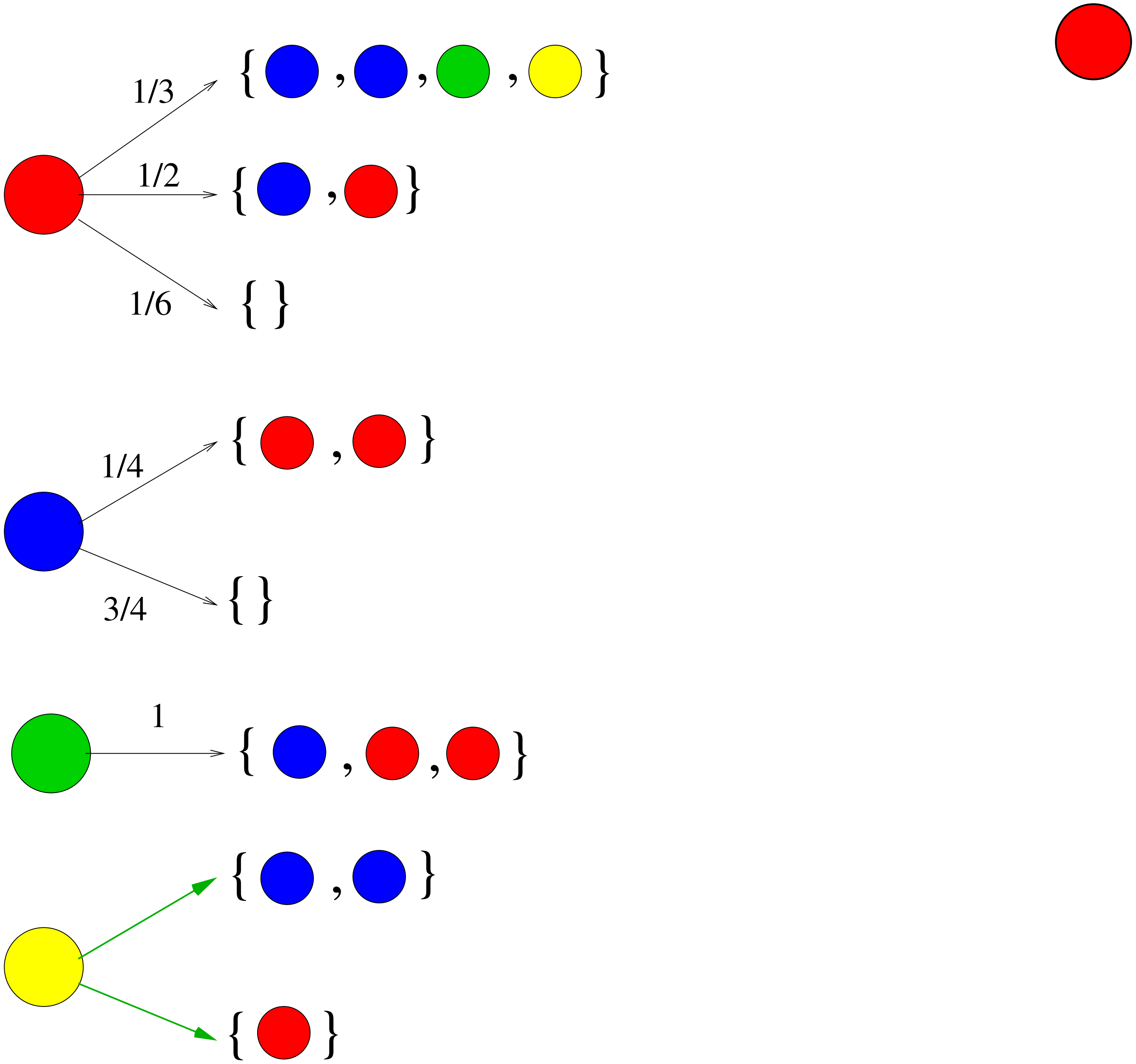
Multi-type Branching Processes (BPs) (Kolmogorov,1940s)



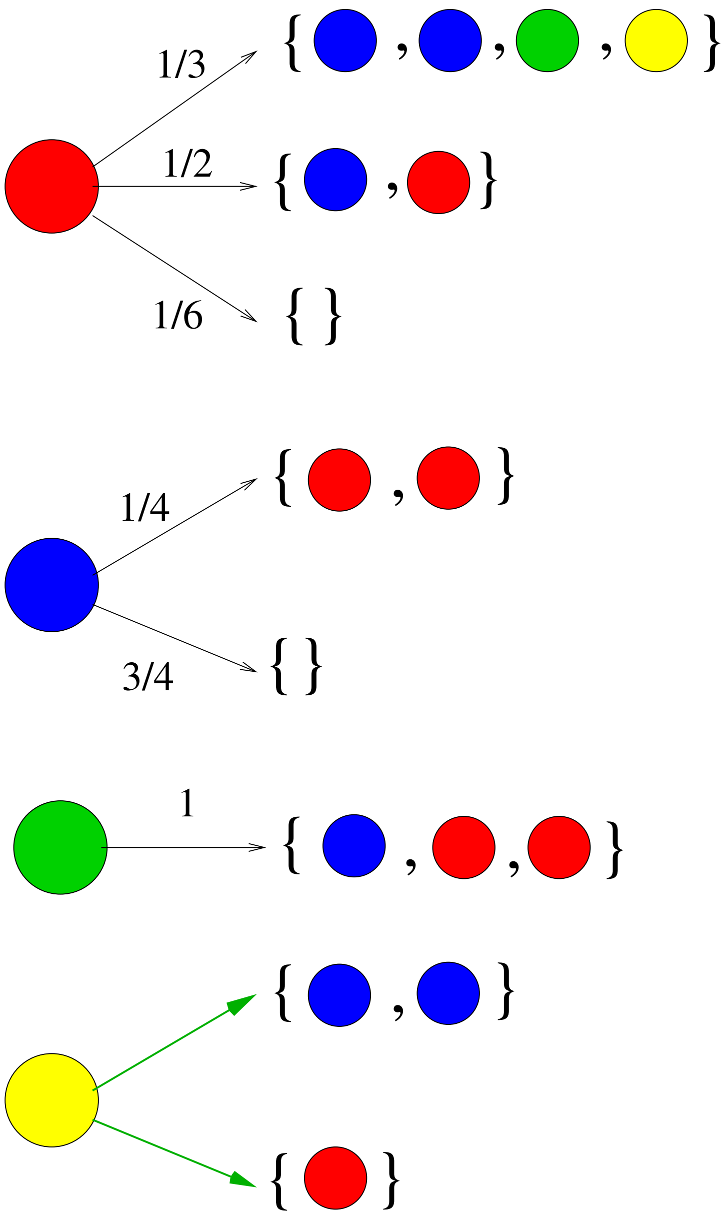
Branching Markov Decision Processes

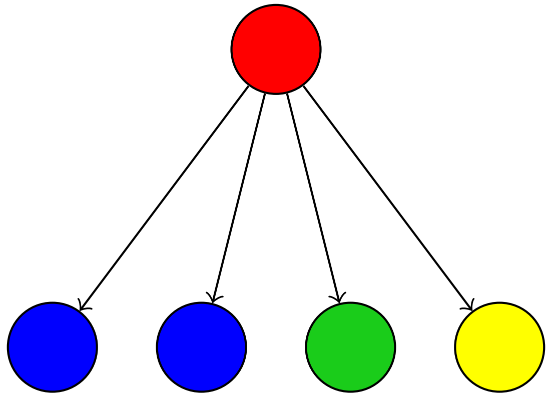


Branching Markov Decision Processes

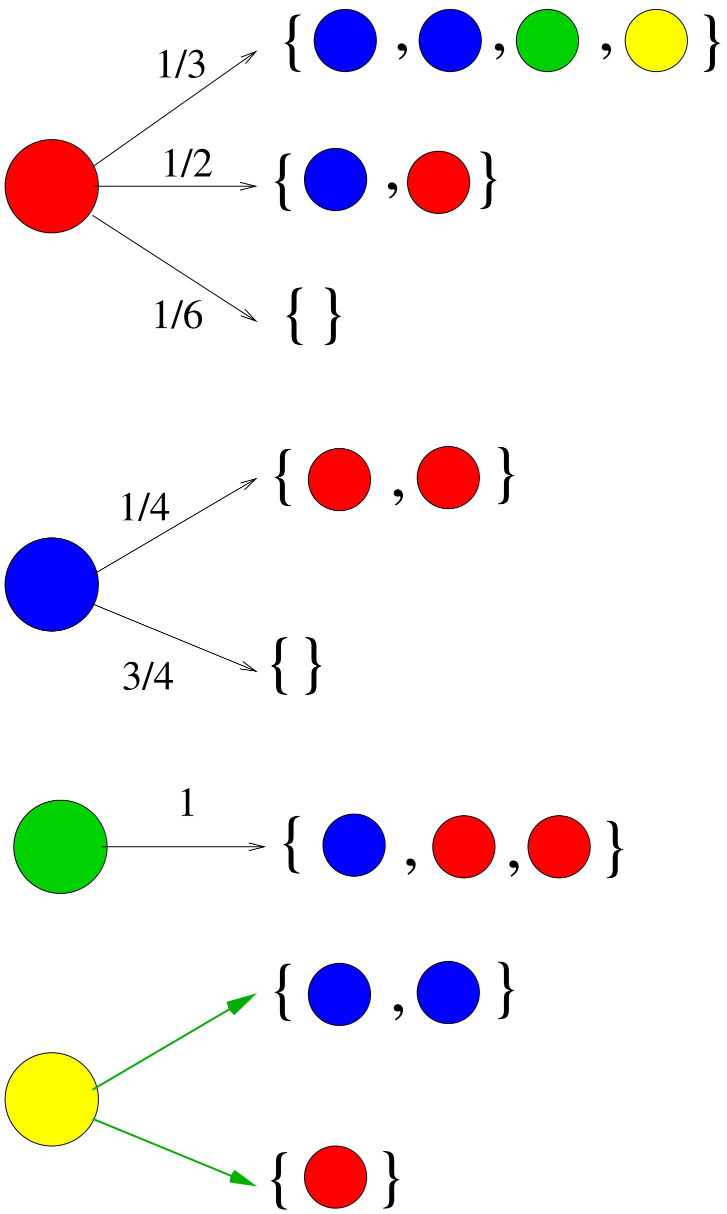


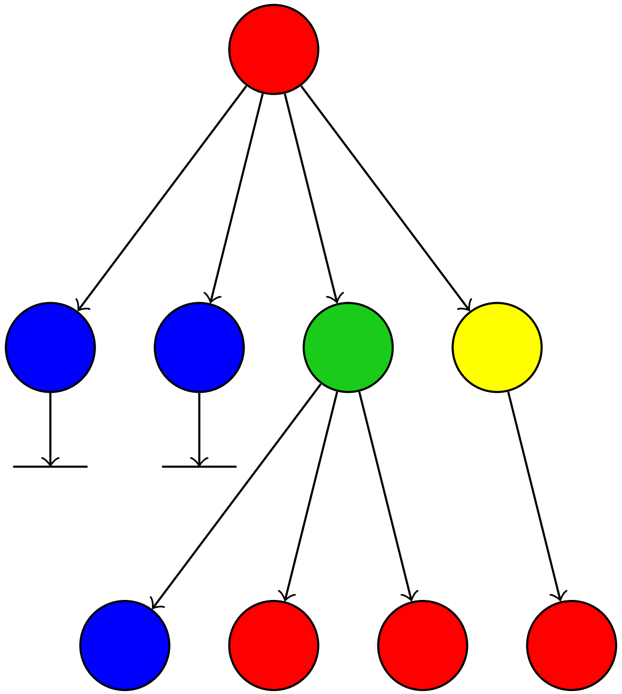
Branching Markov Decision Processes



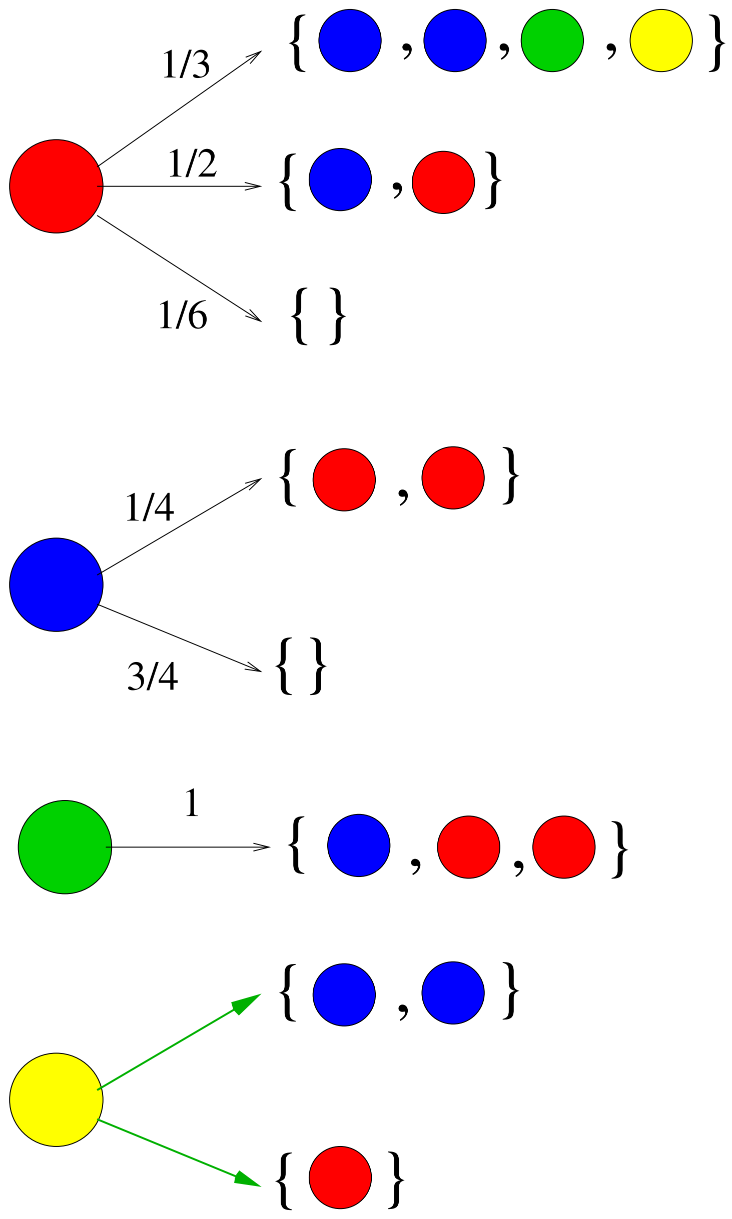


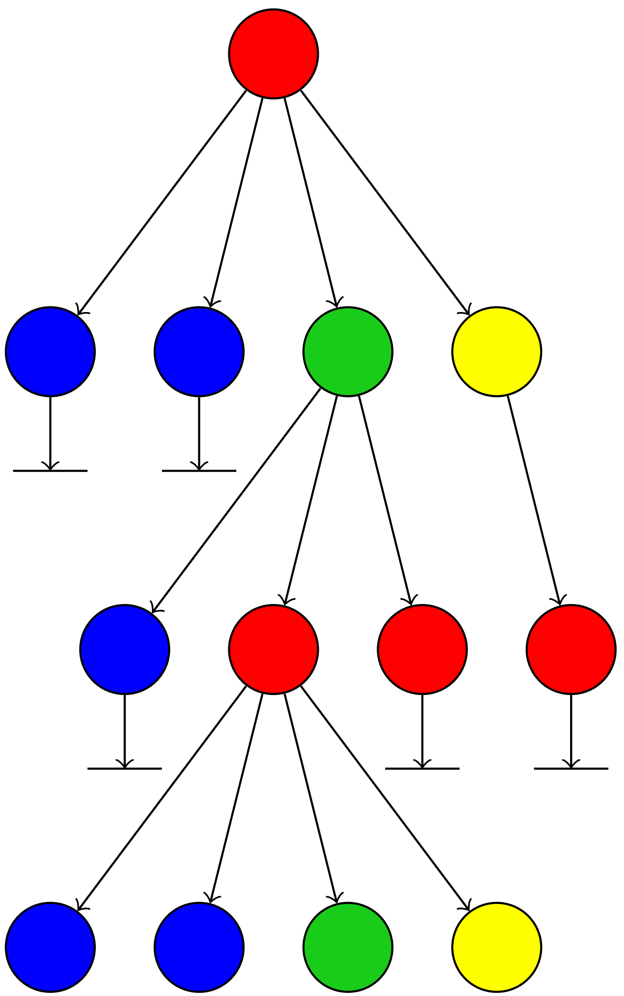
Branching Markov Decision Processes



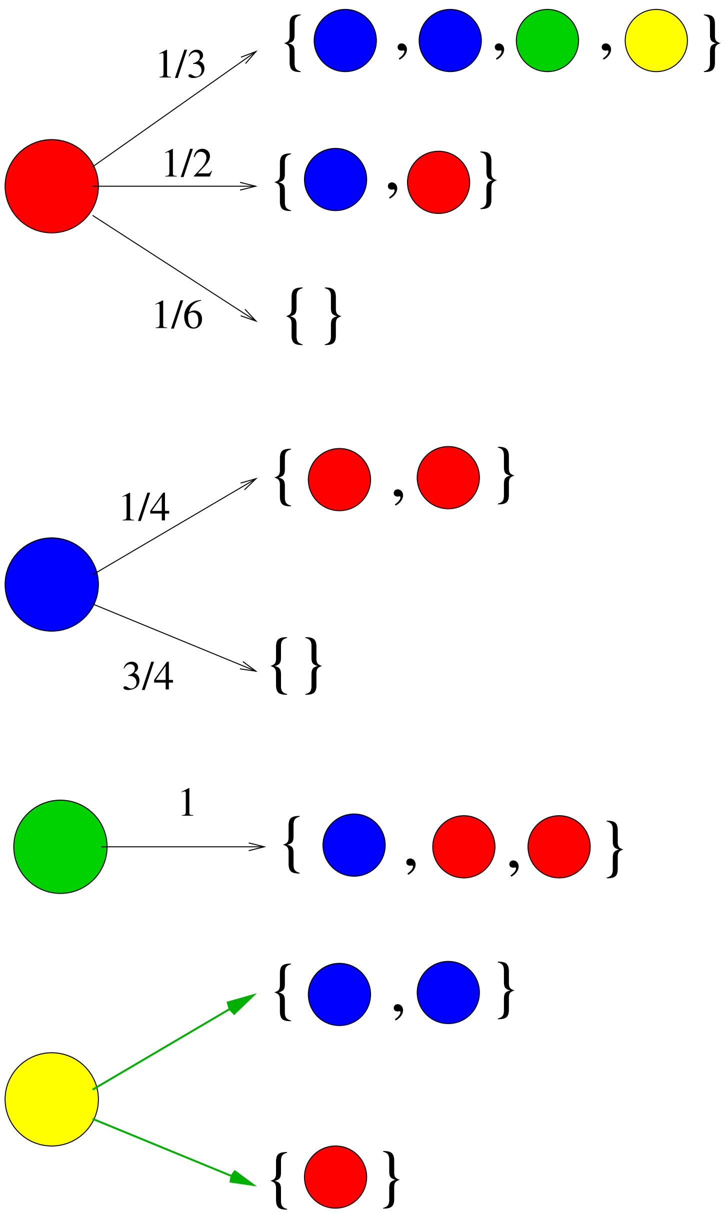


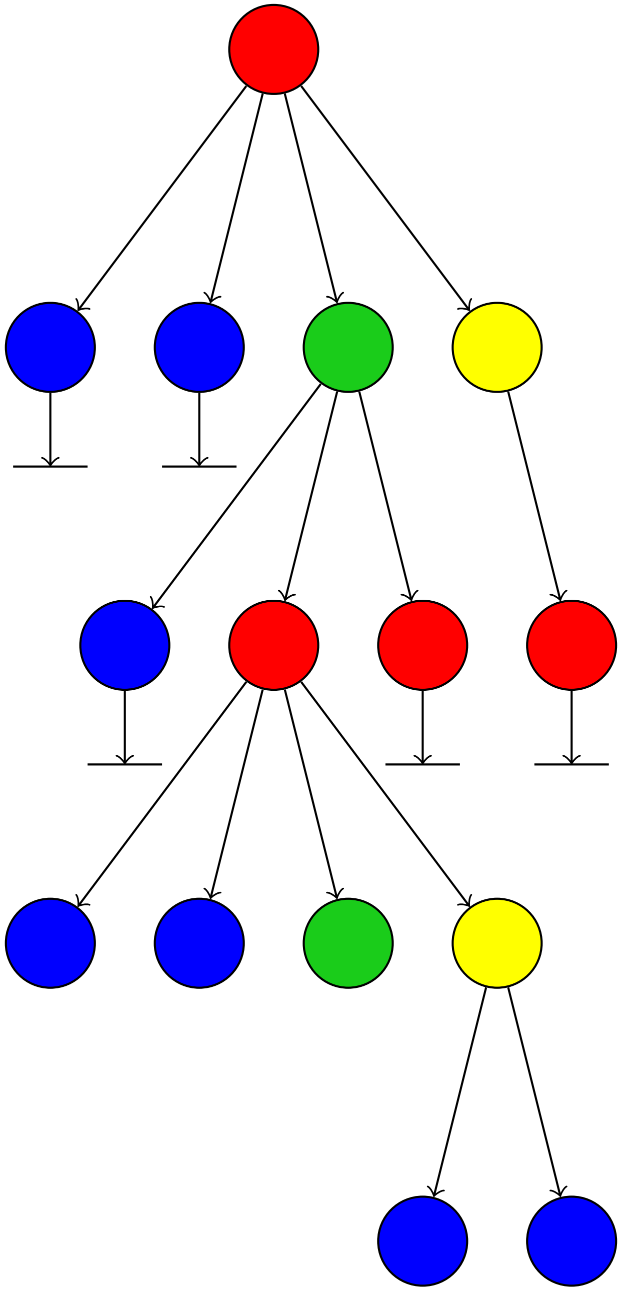
Branching Markov Decision Processes



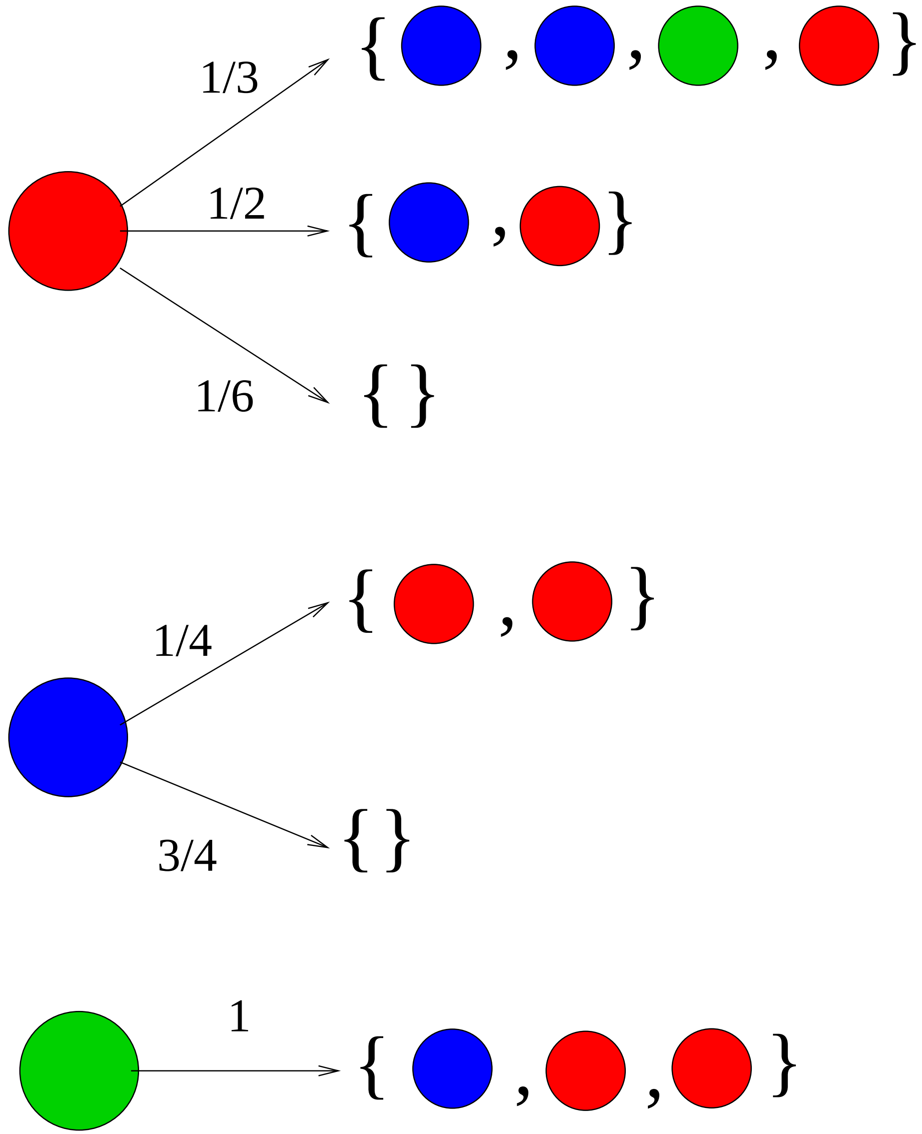


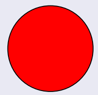
Branching Markov Decision Processes



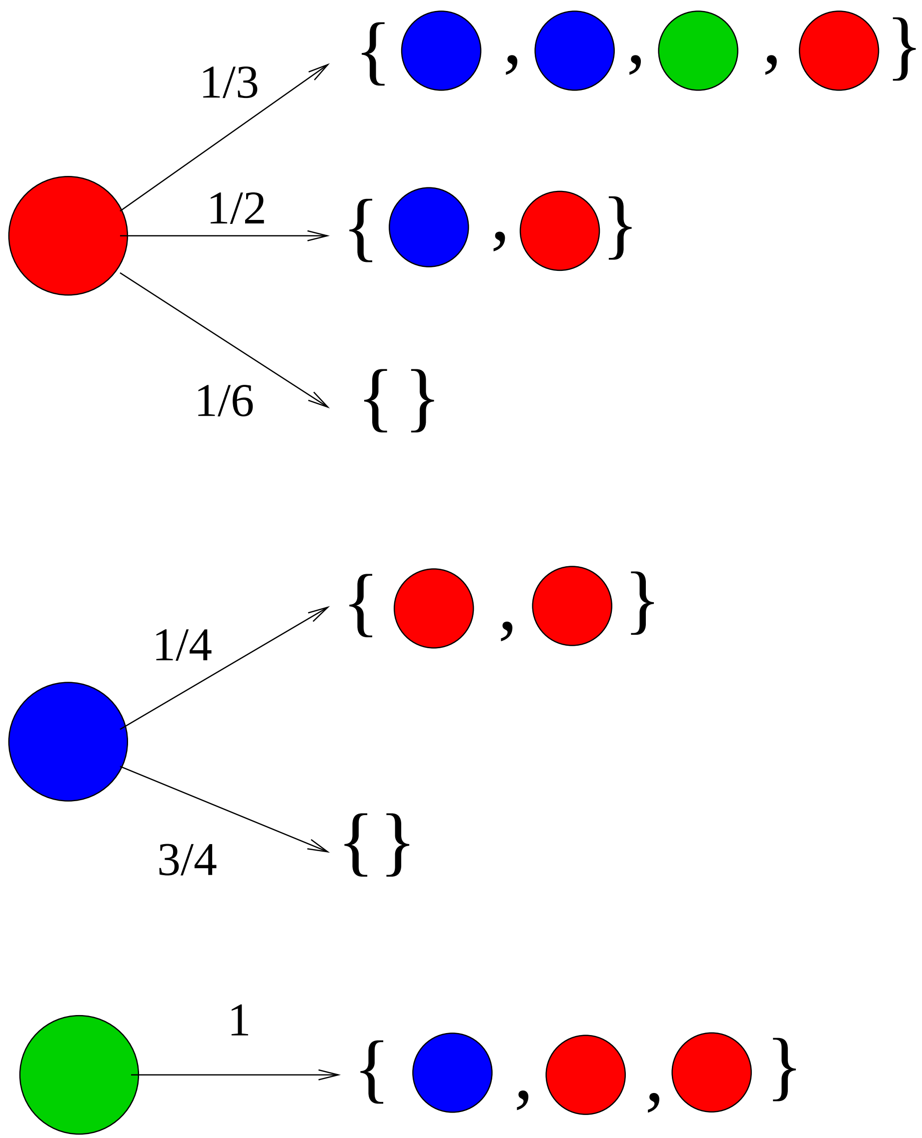


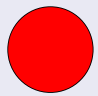
Multi-type Branching Processes (Kolmogorov,1940s)



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

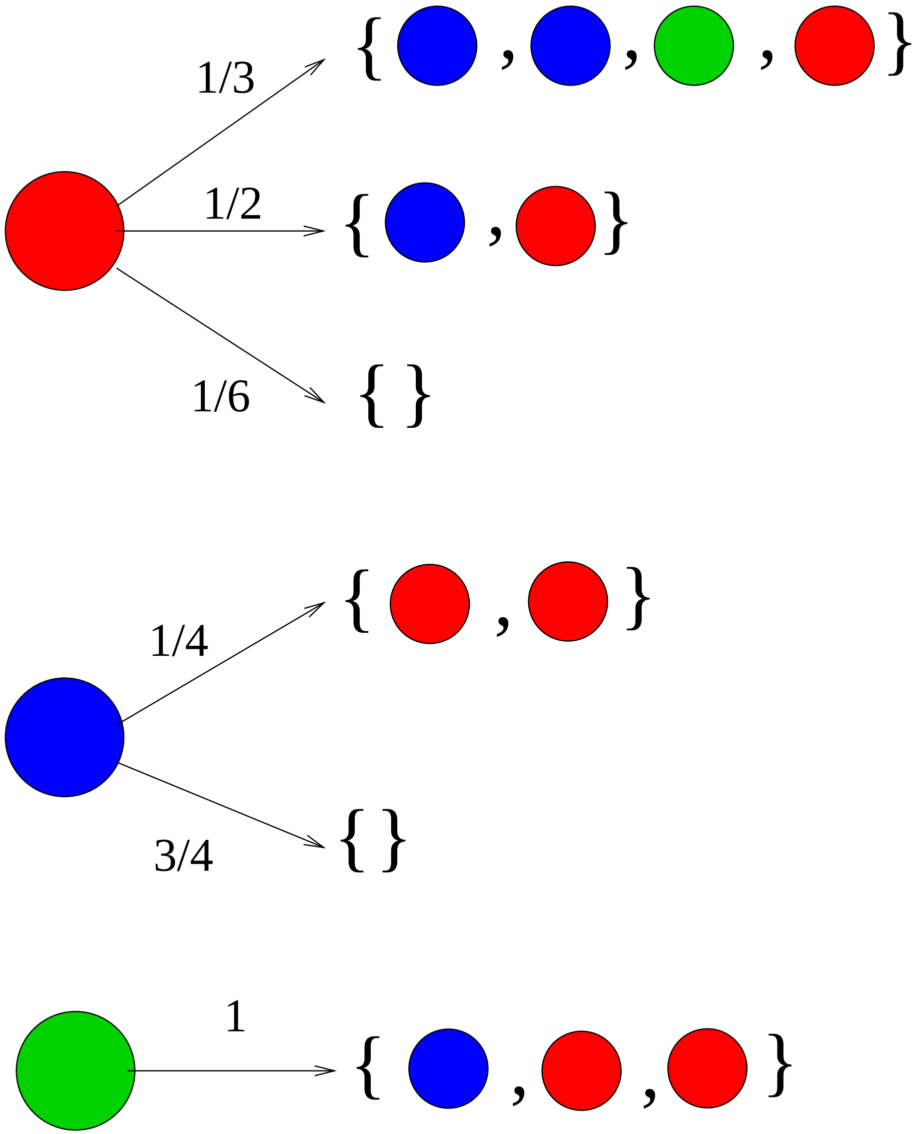
Multi-type Branching Processes (Kolmogorov,1940s)

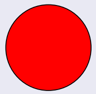


Question: What is the probability of eventual extinction, starting with one  ?

$$x_R =$$

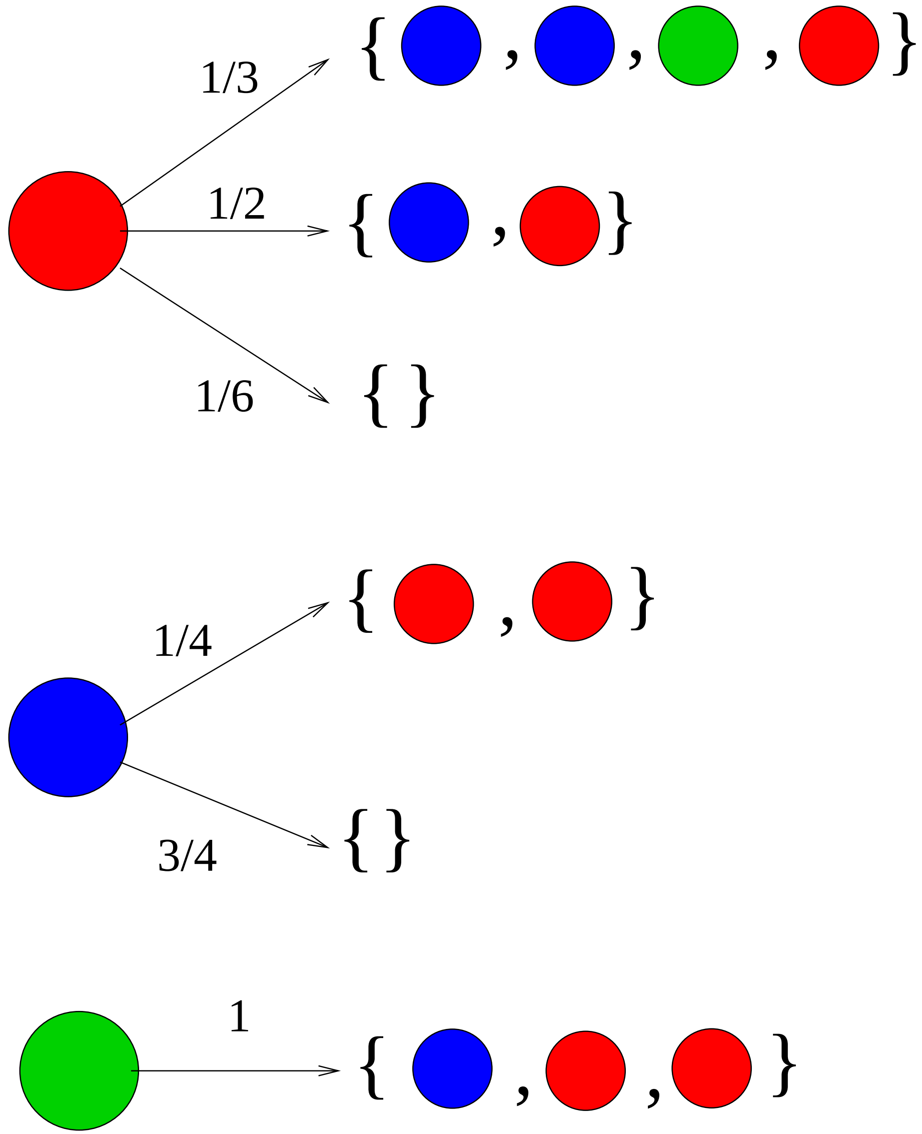
Multi-type Branching Processes (Kolmogorov,1940s)

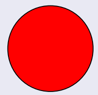


Question: What is the probability of eventual extinction, starting with one  ?

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

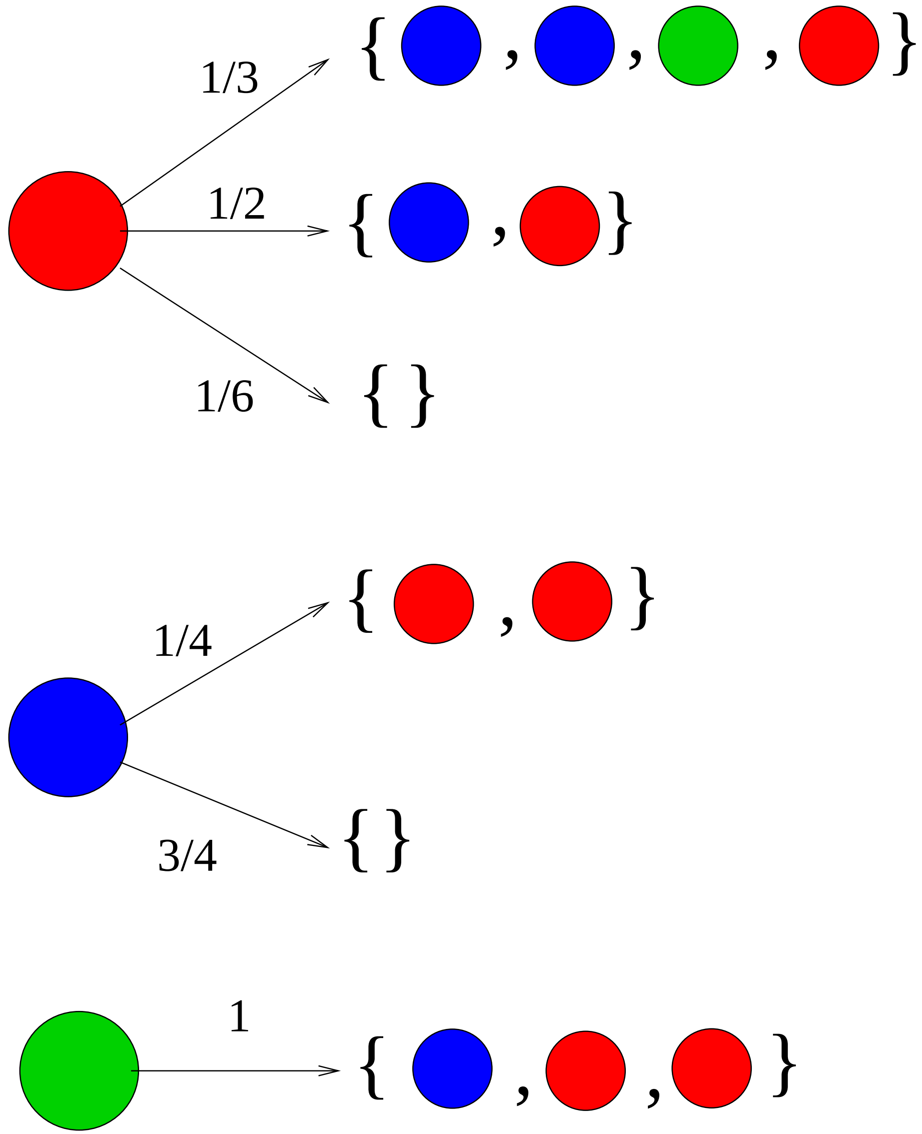
Multi-type Branching Processes (Kolmogorov,1940s)

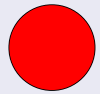


Question: What is the probability of eventual extinction, starting with one  ?

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

Multi-type Branching Processes (Kolmogorov,1940s)



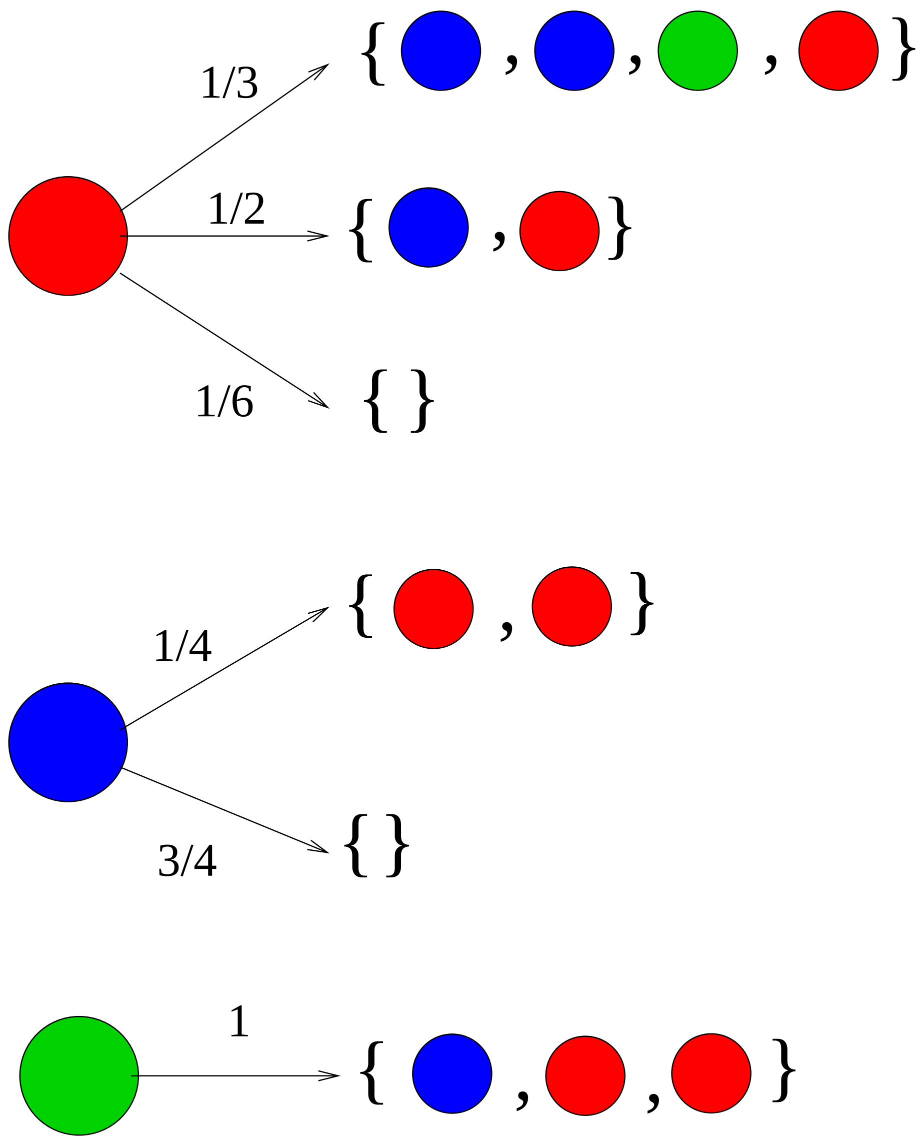
Question: What is the probability of eventual extinction, starting with one  ?

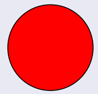
$$
\begin{aligned}
x_R &= \frac{1}{3} x_B^2 x_G x_R + \frac{1}{2} x_B x_R + \frac{1}{6} \\
x_B &= \frac{1}{4} x_R^2 + \frac{3}{4} \\
x_G &= x_B x_R^2
\end{aligned}
$$

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  ?

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

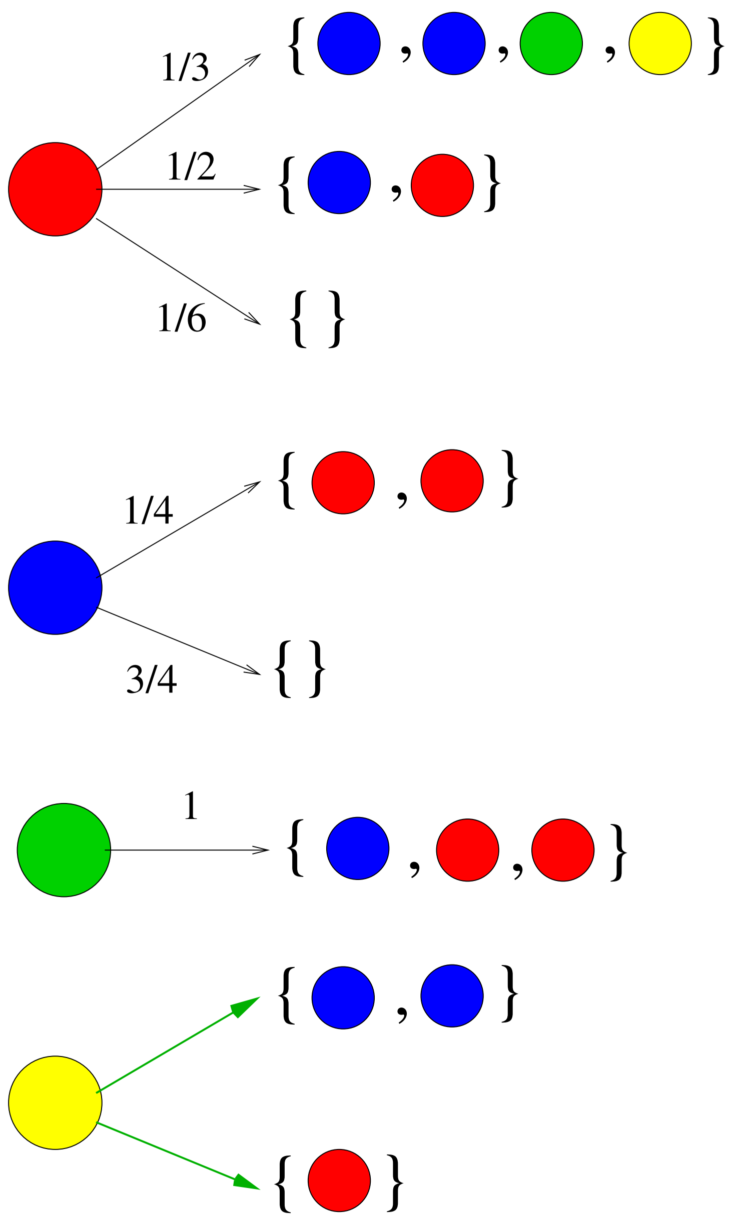
We get nonlinear fixed point equations:

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

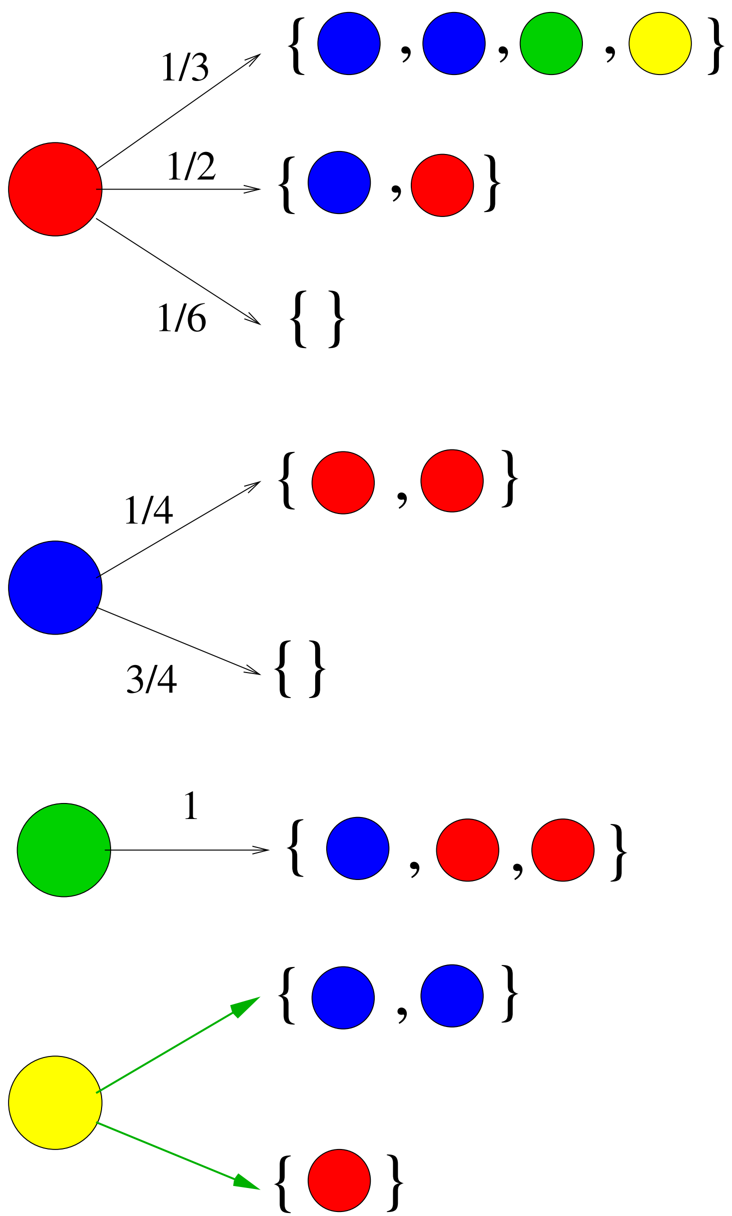
Fact

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

Branching Markov Decision Processes



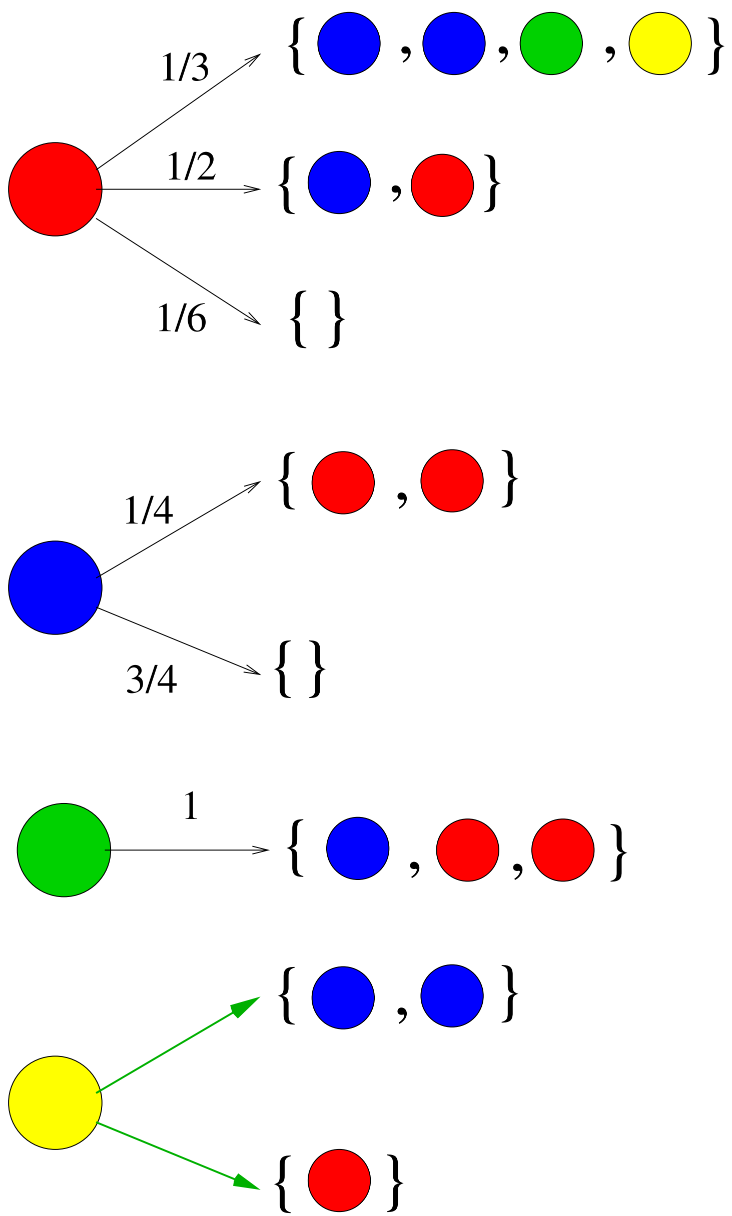
Branching Markov Decision Processes



Question

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

Branching Markov Decision Processes

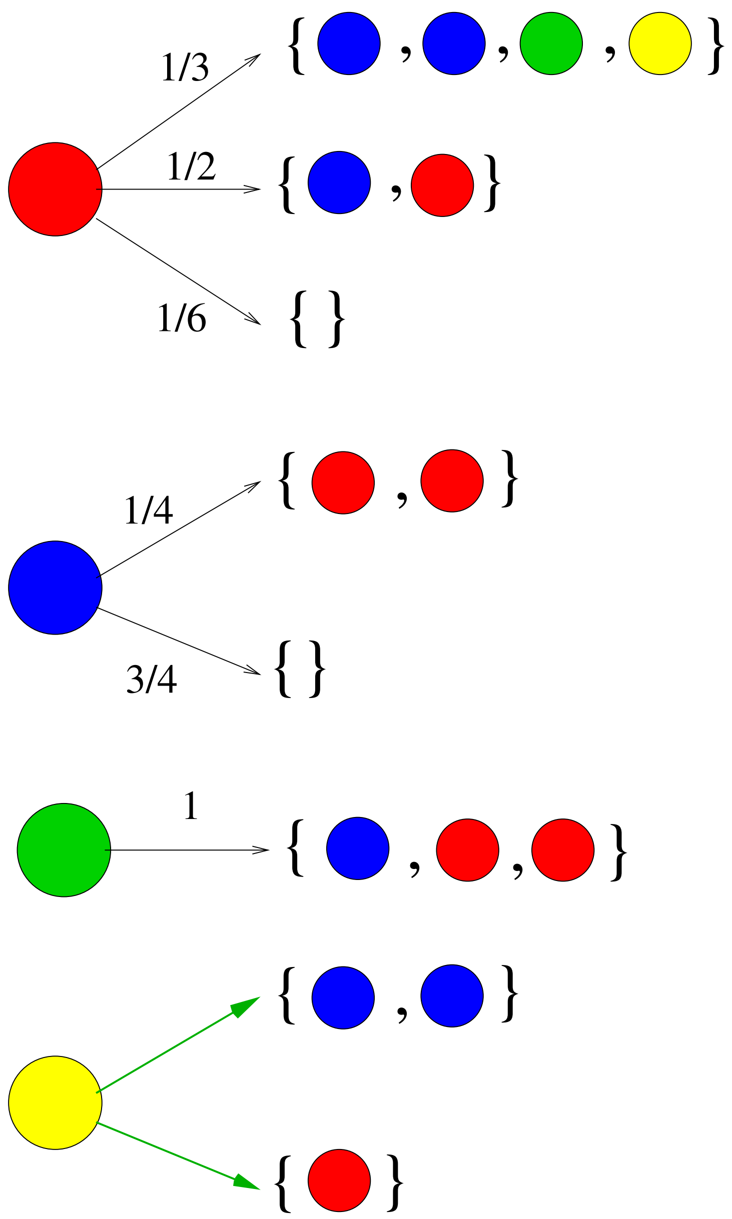


Question

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

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

Branching Markov Decision Processes



Question

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

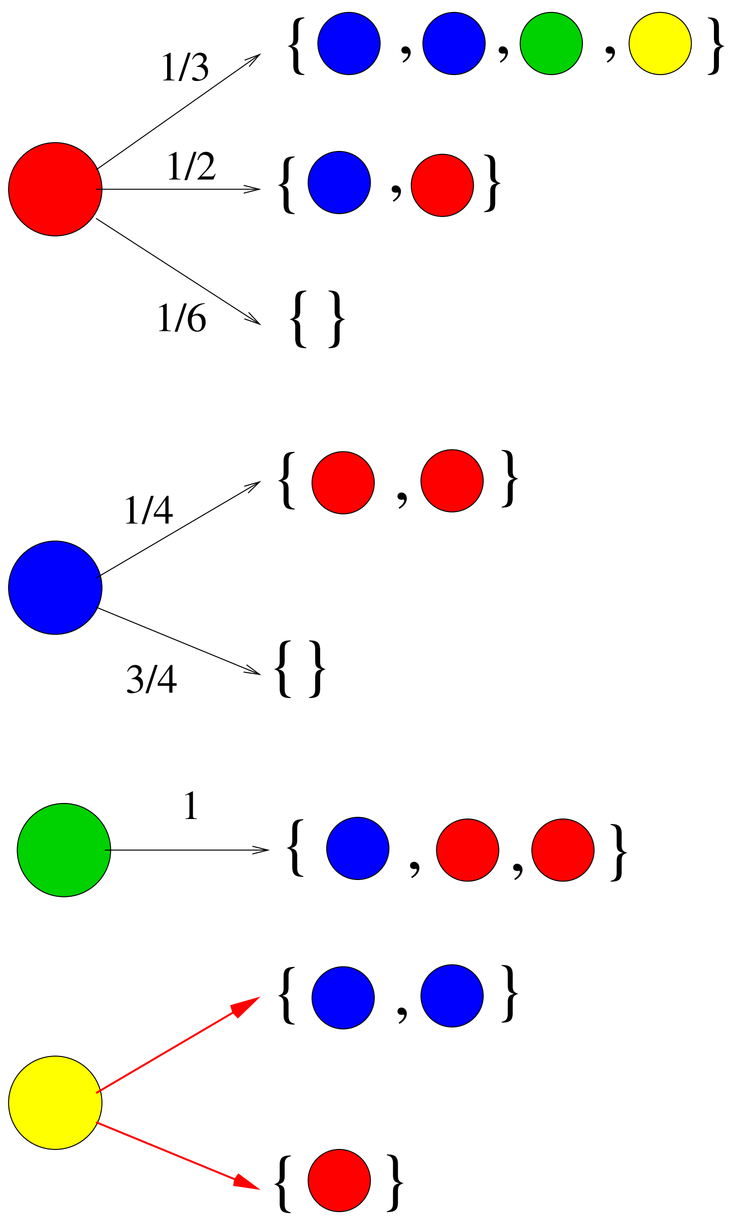
$$
\begin{aligned}
x_R &= \frac{1}{3} x_B^2 x_G x_Y + \frac{1}{2} x_B x_R + \frac{1}{6} \\
x_B &= \frac{1}{4} x_R^2 + \frac{3}{4} \\
x_G &= x_B x_R^2 \\
x_Y &= \max\{x_B^2, x_R\}
\end{aligned}
$$

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

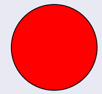
Theorem [E.-Yannakakis'05]

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

Theorem [E.-Yannakakis'05]

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

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

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

A 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.

(max/min) Monotone Polynomial Systems of Equations

$$5x_B^2 x_G x_R + 2x_B x_R + \frac{1}{6}$$

is a Monotone Polynomial: the coefficients are positive.

A Monotone Polynomial System (MPS), is a system of n equations

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

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

We similiarly define max/minMPSs.

Basic properties of (max/min) PPSs & MPSs

A (max/min)PPS, $P : [0,1]^n \rightarrow [0,1]^n$ defines a monotone map on $[0,1]^n$.
A (max/min)MPS, $P : [0,\infty]^n \rightarrow [0,\infty]^n$ gives monotone map on $[0,\infty]^n$.

Proposition

- *[Tarski'55] A (max/min)PPS, $x = P(x)$ has a least fixed point (LFP) solution, $q^* \in [0,1]^n$. (q^* can be irrational.)*
- *[Tarski'55] A (max/min)MPS $x = P(x)$ has a LFP, $q^* \in [0,\infty]^n$. (The (max/min)MPS is called feasible if $q^* \in \mathbb{R}^n_{\geq 0} \doteq [0,\infty)^n$.)*
- $q^* = \lim_{k \rightarrow \infty} P^k(\mathbf{0})$*, monotonically, for all (max/min)PPSs/MPSs.*
- *For a (max/min)PPS, q^* is the vector of (optimal) extinction probabilities for the corresponding BP (BMDP).*

  (For a (max/min) MPS, q^* is, e.g., the partition function of the corresponding (max/min) Weighted Context-Free Grammar ((max/min)WCFG).)

Key Question

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

- For BPs and their corresponding PPSs, this question was considered already by Kolmogorov & Sevastyanov (1940s).
- Analogous questions have been considered for many other stochastic models and their corrrsponding monotone equations (in particular, in the MAM community).
- Nevertheless, the computational complexity of these basic questions (are they solvable in P-time?) remained open until recently.

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**:

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

For MPSs, $F(x) \equiv (P(x) - x)$; 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's method on PPSs and feasible MPSs

To enable monotone Newton methods ([Ortega-Rheinboldt,1970]) to apply to all PPSs and all feasible MPSs, we must first do some simple (P-time) preprocessing of the equations:

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

Proposition [E.-Yannakakis'05]

Decomposed Newton's method converges monotonically to the LFP $\mathbf{q}^*$, starting from $\mathbf{x}^{(0)} := \mathbf{0}$, for all feasible MPSs.

But this does not imply P-time for feasible MPSs

Theorem ([E.-Yannakakis'05,JACM'09]): any nontrivial approximation of the LFP $\mathbf{q}^* \in [0,1]^n$ of a family of feasible MPSs corresponding to Recursive Markov Chains is PosSLP-hard (thus even doing it in **NP** would be a breakthrough).

What is Newton's worst case behavior for PPSs and MPSs?

- There are bad examples of PPSs. Here's a simple example ([Stewart-E.-Yannakakis,'13, JACM'15]):

$$x_0 = \frac{1}{2}x_0^2 + \frac{1}{2}; \quad x_i = \frac{1}{2}x_i^2 + \frac{1}{2}x_{i-1}^2; i = 1, \ldots, n$$

  Fact: $q^* = \mathbf{1}$, but $\|q^* - x^{(2^n-1)}\|_\infty > \frac{1}{2}$, starting from $x^{(0)} := \mathbf{0}$.

- This slightly simplifies an earlier exponential example by [Esparza,Kiefer,Luttenberger'10], who also gave exponential upper bounds on the restricted class of strongly-connected MPSs.

  But they gave no upper bounds for general feasible PPSs or MPSs.

- In [Stewart-E.-Yannakakis,'13, JACM'15] we established (essentially optimal) exponential upper bounds for $\sharp$ of Newton iterations required (in worst case) starting from $x^{(0)} = \mathbf{0}$, in terms of both $|P|$ and $\log(1/\epsilon)$ to compute the LFP q^* with error $< \epsilon$, for all feasible MPSs.

P-time approximation for PPSs

Theorem ([E.-Stewart-Yannakakis, STOC'2012])

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

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

(It is even in strongly-P-time ([Esparza-Gaiser-Kiefer'10]).)

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

(It is even in strongly-P-time ([Esparza-Gaiser-Kiefer'10]).)

Deciding whether $q_i^* = 0$ is also easily in (strongly) P-time.

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 ([E.-Stewart-Yannakakis,STOC'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*

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

and

$$\|\mathbf{q}^* - \mathbf{x}^{(18|P|+j+2)}\|_\infty \leq 2^{-2^j}$$

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)

$$\varrho(P'(q^*)) < 1$$

- 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, among other things...

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

Towards Generalized Newton's Method: 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 the linear (first-order Taylor) approximation of $P(x)$ at the point $\mathbf{y}$.

Generalized Newton's method

Linearization of max/minPPSs

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 linearization, $P^y(x)$, by:

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

Generalized Newton's method

Linearization of max/minPPSs

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 linearization, $P^y(x)$, 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: iteration applied at vector y

Solve $P^{\mathbf{y}}(\mathbf{x}) = \mathbf{x}$. Specifically:

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 one GNM iteration.

Algorithm for max/minPPSs

1. Find and remove all variables x_i such that $q_i^* = 0$ or $q_i^* = 1$.
   Deciding $q_i^* \stackrel{?}{=} 0$ is again easily in P-time.

   Theorem ([E.-Yannakakis'06]): $q_i^* \stackrel{?}{=} 1$ is decidable in P-time.

   (Reduces to a spectral radius optimization problem for non-negative square matrices, which we can solve using linear programming.)

Algorithm for max/minPPSs

1. Find and remove all variables x_i such that $q_i^* = 0$ or $q_i^* = 1$.

   Deciding $q_i^* \stackrel{?}{=} 0$ is again easily in P-time.

   Theorem ([E.-Yannakakis'06]): $q_i^* \stackrel{?}{=} 1$ is decidable in P-time.

   (Reduces to a spectral radius optimization problem for non-negative square matrices, which we can solve using linear programming.)

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 $q_i^* = 1$.

   Deciding $q_i^* \stackrel{?}{=} 0$ is again easily in P-time.

   Theorem ([E.-Yannakakis'06]): $q_i^* \stackrel{?}{=} 1$ is decidable in P-time.

   (Reduces to a spectral radius optimization problem for non-negative square matrices, which we can solve using linear programming.)
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 [E.-Stewart-Yannakakis'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

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

Thus, algorithm runs in time polynomial in $|P|$ and j.

Proof outline: some key lemmas

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

Lemma

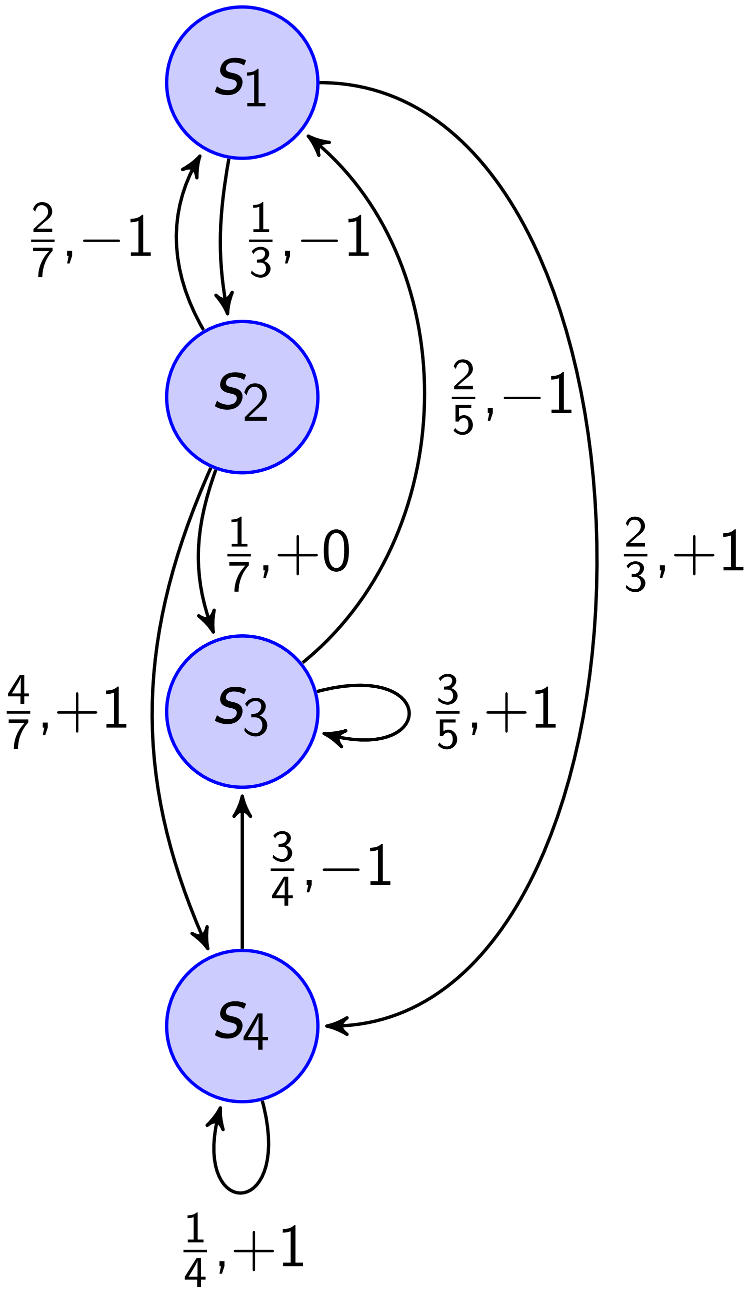
If $\mathbf{q}^* - \mathbf{x}^{(k)} \leq \lambda(\mathbf{1} - \mathbf{q}^*)$ *for some* $\lambda > 0$, *then* $\mathbf{q}^* - \mathbf{x}^{(k+1)} \leq \frac{\lambda}{2}(\mathbf{1} - \mathbf{q}^*)$.

Lemma

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

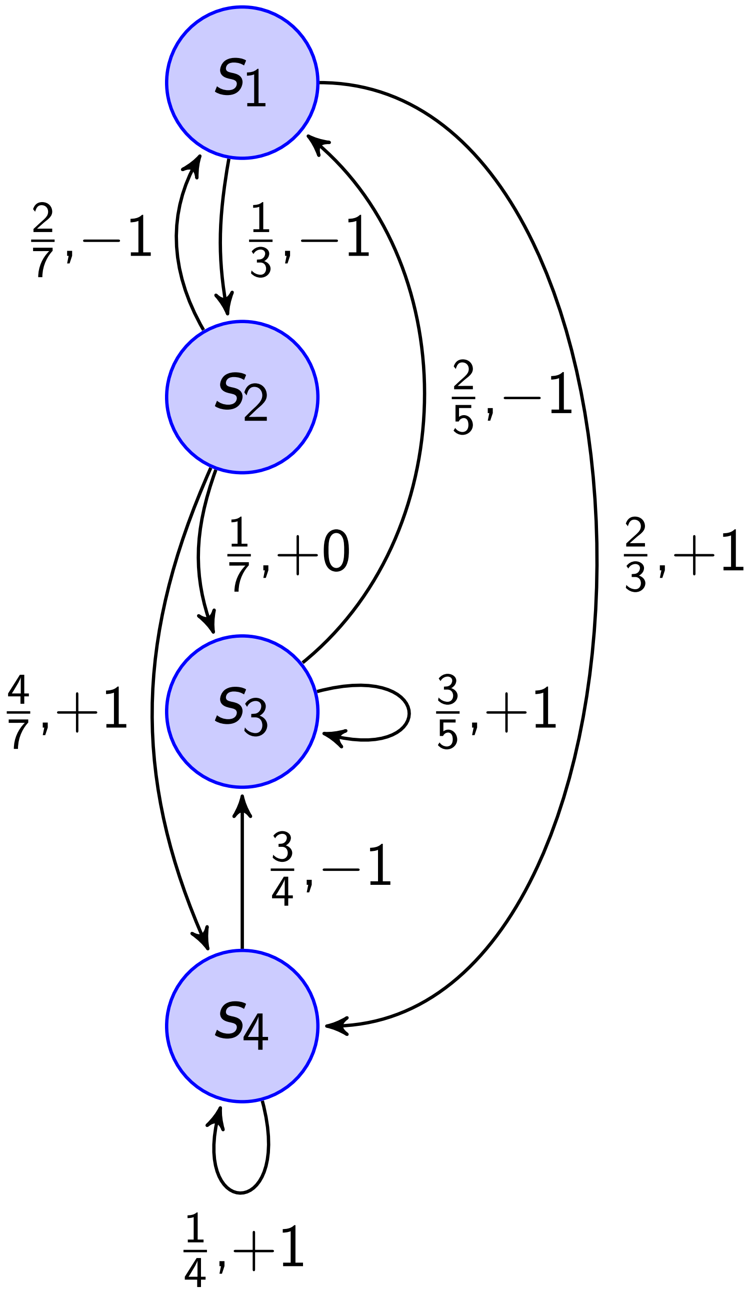
$$q_i^* \leq 1 - 2^{-4|P|}.$$

one-counter Markov chain (discrete-time QBD)



Question: What is the probability of termination (reaching counter value $= 0$ for the first time) in state $\mathbf{s}_2$, starting with counter value $= 1$ in state $\mathbf{s}_1$?

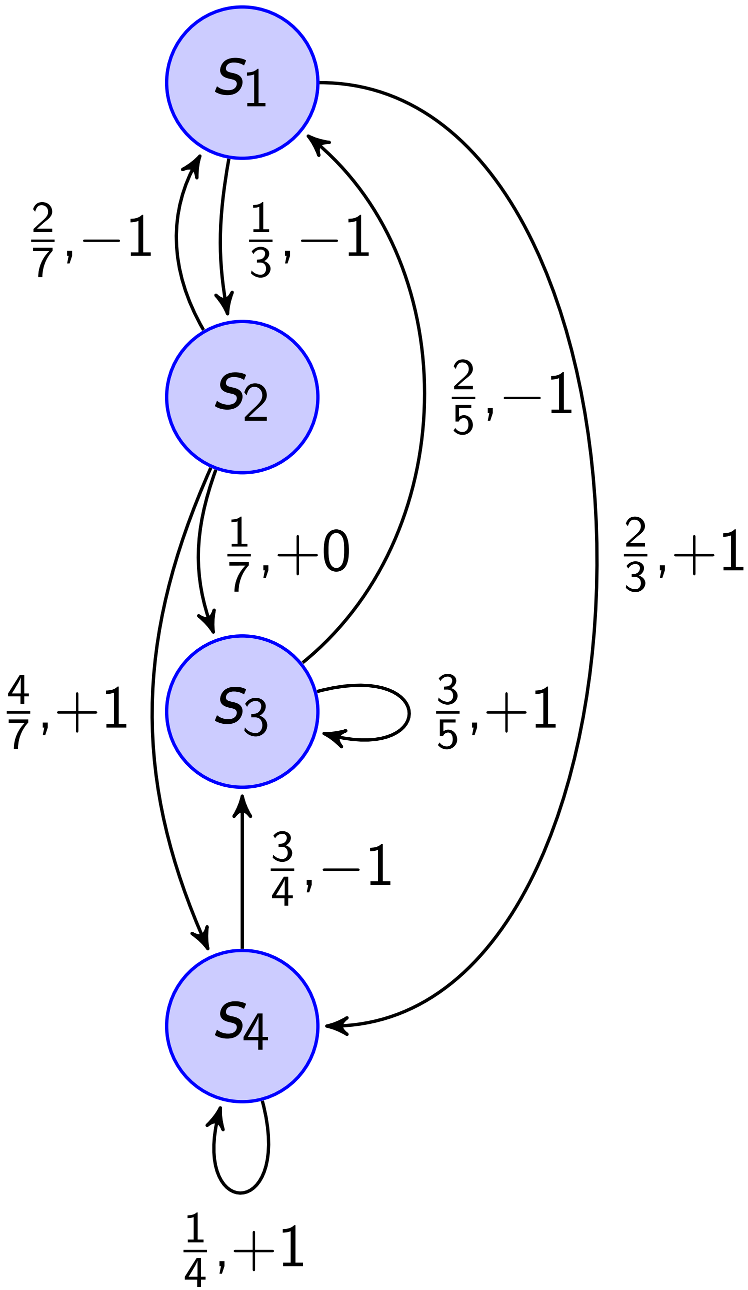
one-counter Markov chain (discrete-time QBD)



Question: What is the probability of termination (reaching counter value $= 0$ for the first time) in state $\mathbf{s}_2$, starting with counter value $= 1$ in state $\mathbf{s}_1$?

$$x_{1,2} \quad =$$

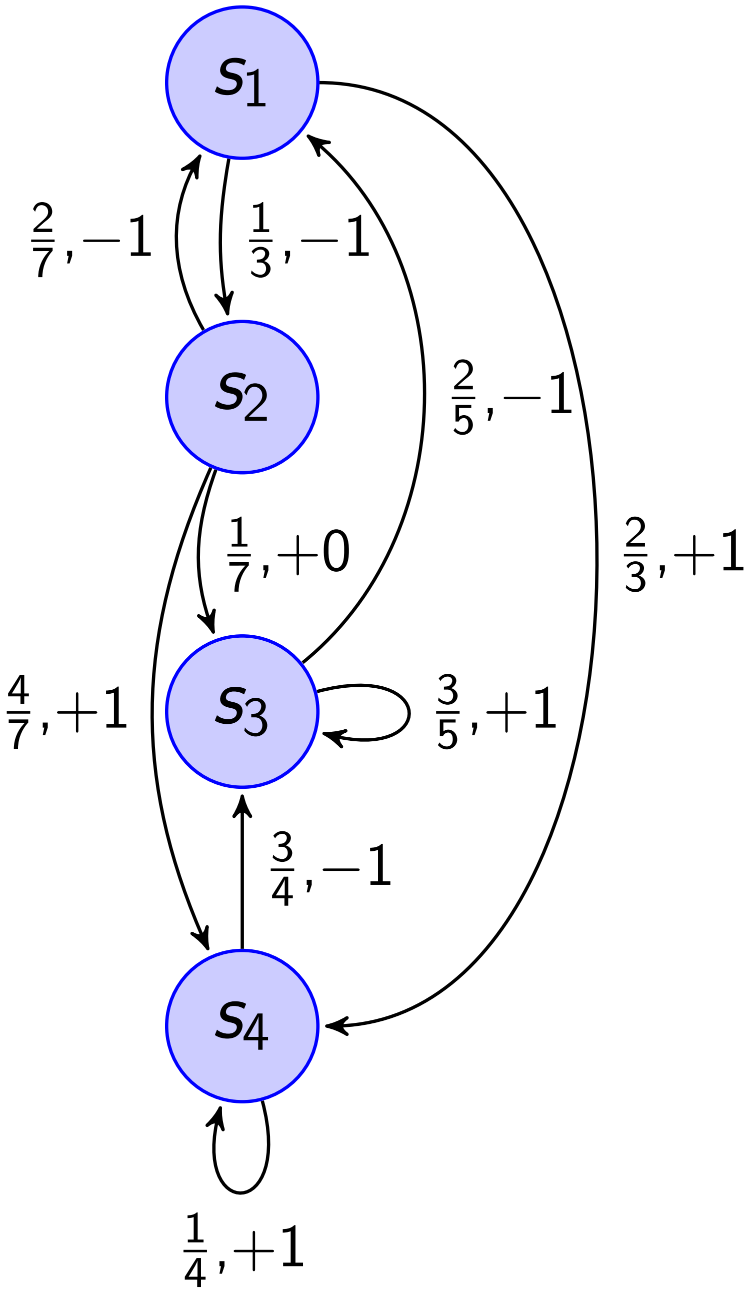
one-counter Markov chain (discrete-time QBD)



Question: What is the probability of termination (reaching counter value $= 0$ for the first time) in state $\mathbf{s}_2$, starting with counter value $= 1$ in state $\mathbf{s}_1$?

$$
\begin{aligned}
x_{1,2} &= \frac{1}{3} + \frac{2}{3} \sum_j x_{4,j} x_{j,2} \\
x_{4,3} &= \frac{3}{4} + \frac{1}{4} \sum_j x_{4,k} x_{k,2} \\
\cdots &= \cdots
\end{aligned}
$$

one-counter Markov chain (discrete-time QBD)

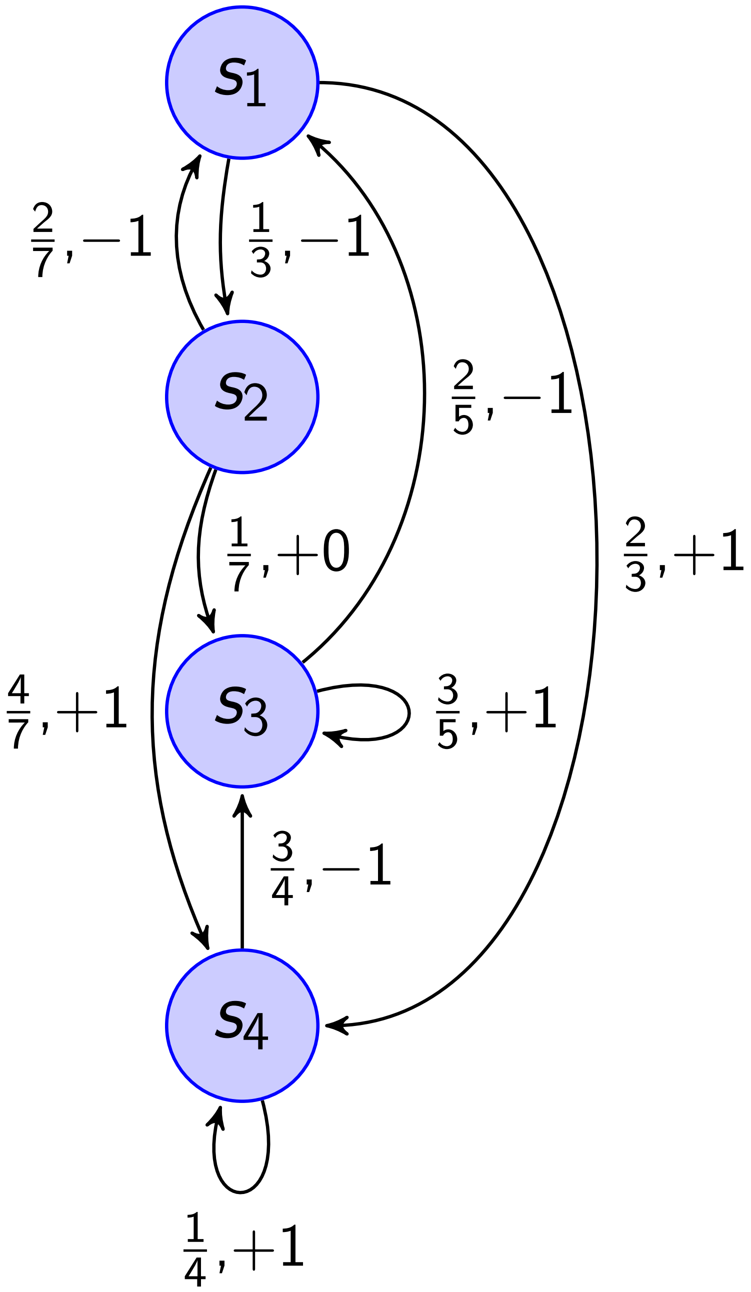


Question: What is the probability of termination (reaching counter value $= 0$ for the first time) in state $\mathbf{s}_2$, starting with counter value $= 1$ in state $\mathbf{s}_1$?

$$
\begin{aligned}
x_{1,2} &= \frac{1}{3} + \frac{2}{3} \sum_j x_{4,j} x_{j,2} \\
x_{4,3} &= \frac{3}{4} + \frac{1}{4} \sum_j x_{4,k} x_{k,2} \\
\cdots &= \cdots
\end{aligned}
$$

In matrix notation, the familiar G-matrix monotone fixed point equations for a QBD: $X = A_{-1} + A_0 X + A_1 X^2$.

one-counter Markov chain (discrete-time QBD)



Question: What is the probability of termination (reaching counter value $= 0$ for the first time) in state $\mathbf{s}_2$, starting with counter value $= 1$ in state $\mathbf{s}_1$?

$$
\begin{aligned}
x_{1,2} &= \frac{1}{3} + \frac{2}{3} \sum_j x_{4,j} x_{j,2} \\
x_{4,3} &= \frac{3}{4} + \frac{1}{4} \sum_j x_{4,k} x_{k,2} \\
\dots &= \dots
\end{aligned}
$$

In matrix notation, the familiar G-matrix monotone fixed point equations for a QBD: $X = A_{-1} + A_0 X + A_1 X^2$.

Fact (cf., [Neuts,1970s])

The G-matrix of termination probabilities is the LFP, $\mathbf{q}^* \in [0,1]^{4 \times 4}$.

Lemma [E.-Wojtczak-Yannakakis'08]

The minimum positive G-matrix entry for an n-state QBD is $\geq (p_{\min})^{n^3}$, where $p_{\min} > 0$ is the minimum positive transition probability (minimum positive entry of A_{-1}, A_0, or A_1) for the QBD.

(Proof uses a basic pumping argument for one-counter automata.)

Lemma [E.-Wojtczak-Yannakakis'08]

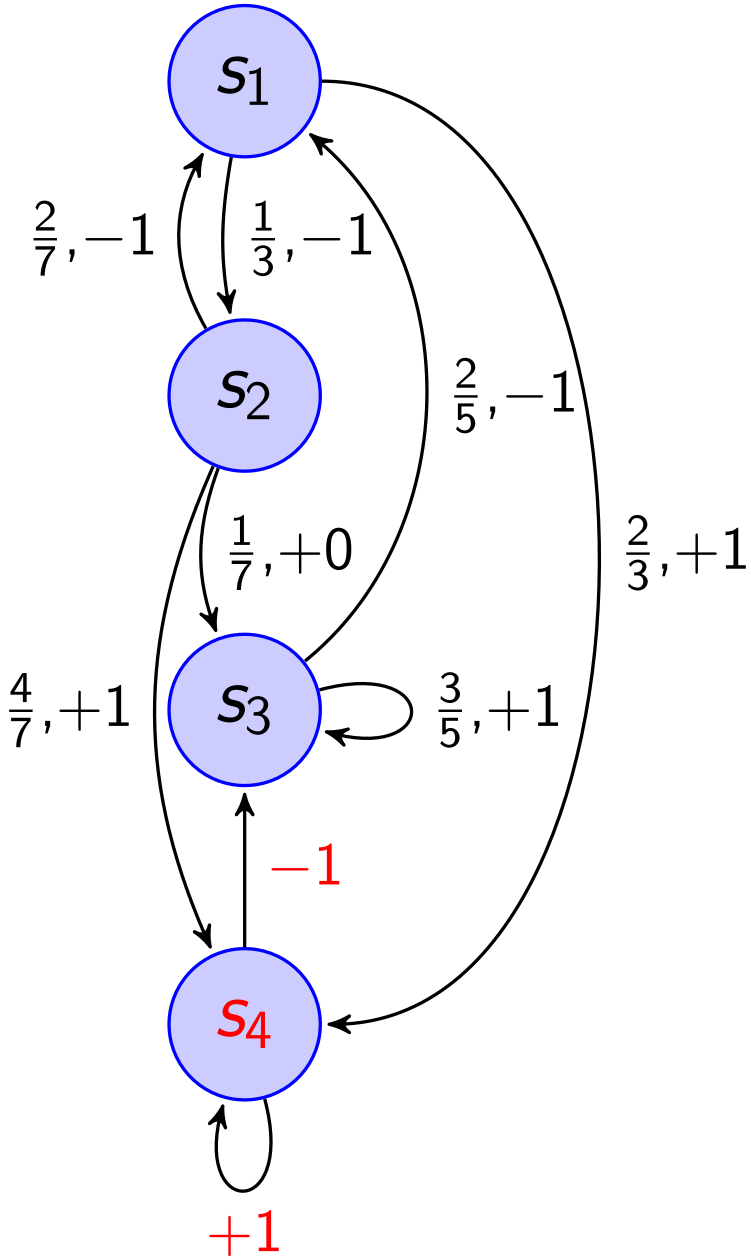
The DAG of strongly connected components (SCCs) of the equations for a QBD can only contain a single non-linear SCC on each directed path.

Using these Lemmas, and the bounds for Newton's method on monotone feasible MPSs, we obtain:

Theorem [E.-Wojtczak-Yannakakis'08], [Stewart-E.-Yannakakis,'13]

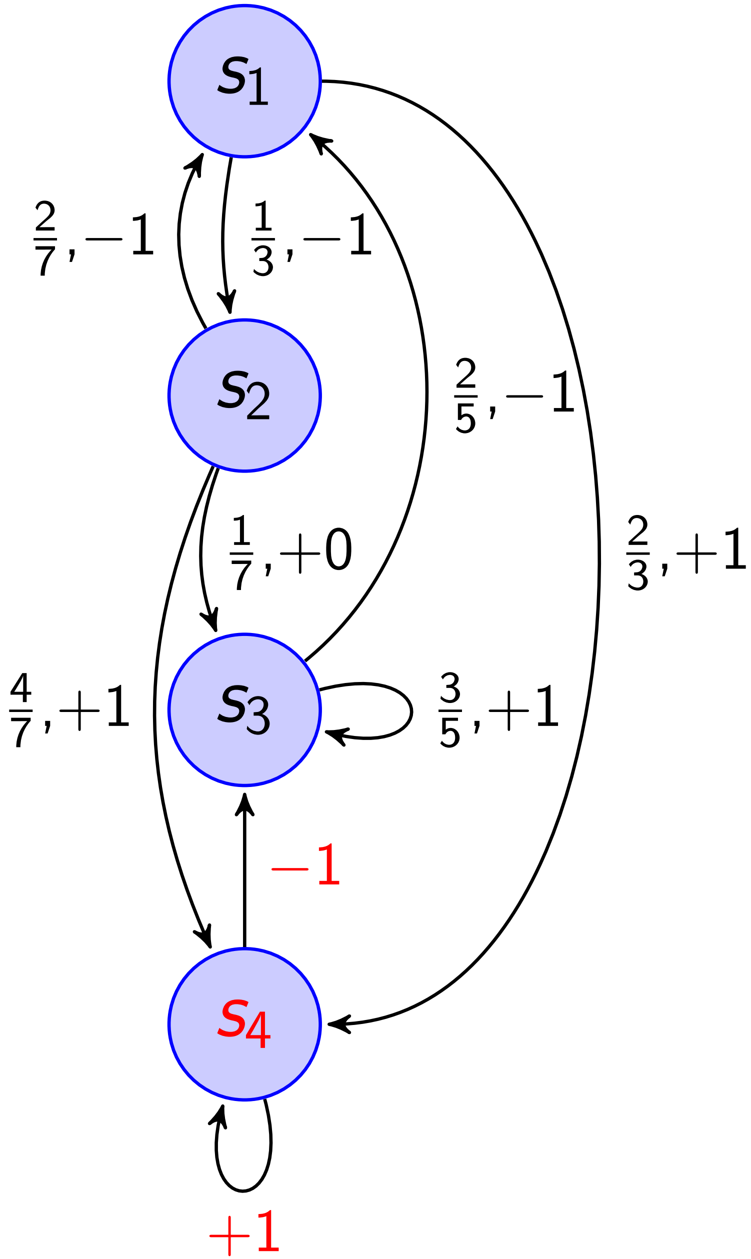
The G-matrix of a QBD, Q, can be approximated to desired accuracy $\epsilon > 0$ in time polynomial in both the encoding size $|Q|$ and $\log(1/\epsilon)$ (in the standard Turing model of computation).

one-counter Markov Decision Processes



Question: What is the optimal (supremum or infimum) probability of termination (reaching counter value $= 0$) in any state, starting with counter value $= 1$ in state $\mathbf{s}_1$?

one-counter Markov Decision Processes



Question: What is the optimal (supremum or infimum) probability of termination (reaching counter value $= 0$) in any state, starting with counter value $= 1$ in state $\mathbf{s}_1$?

We do not know any min/max-monotone polynomial equations that capture these optimal probabilities.

But we do have algorithms to compute them.....

Theorem [Brazdil-Brózek-E.-Kucera,2011]

Given a OC-MDP, M, we can compute the optimal (supremum/infimum) termination probabilities to accuracy $\epsilon > 0$ in time polynomial in $\log(1/\epsilon)$, but unfortunately exponential in $|M|$.

Algorithm involves (exponentially large) finite-state (mean-payoff) MDPs. Proof uses an intriguing martingale derived from LPs associated with optimizing mean-payoff MDPs, and the Azuma inequality.

Theorem [Brazdil-Brózek-E.-Kucera-Wojtzak,SODA'2010]

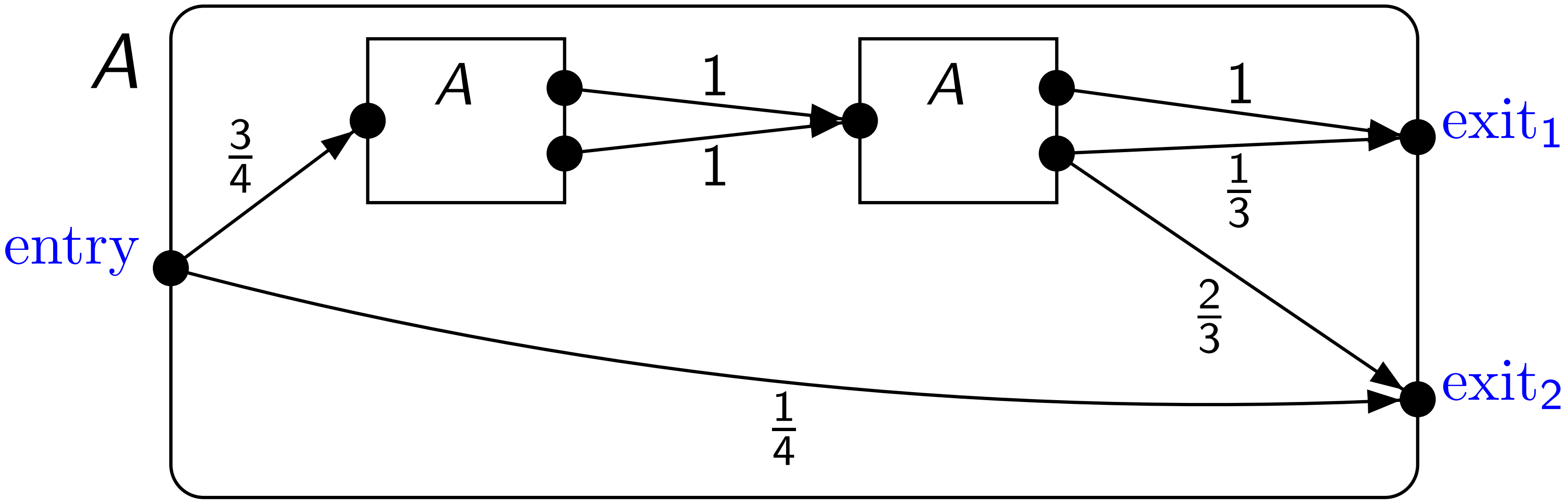
We can decide whether the optimal termination probabilities for a given OC-MDP are $= 1$ in P-time.

Proof uses LPs, and limit theorems for sums of i.i.d. random variables.

Theorem [Brazdil-Brózek-E.-Kucera-Wojtzak,SODA'2010]

Given a OC-MDP, deciding whether the maximum achievable probability of terminating in a specific state, $\mathbf{s}_i$, is $= 1$, is **NP-hard** (and even **PSPACE-hard**), and is decidable in **EXPTIME**.

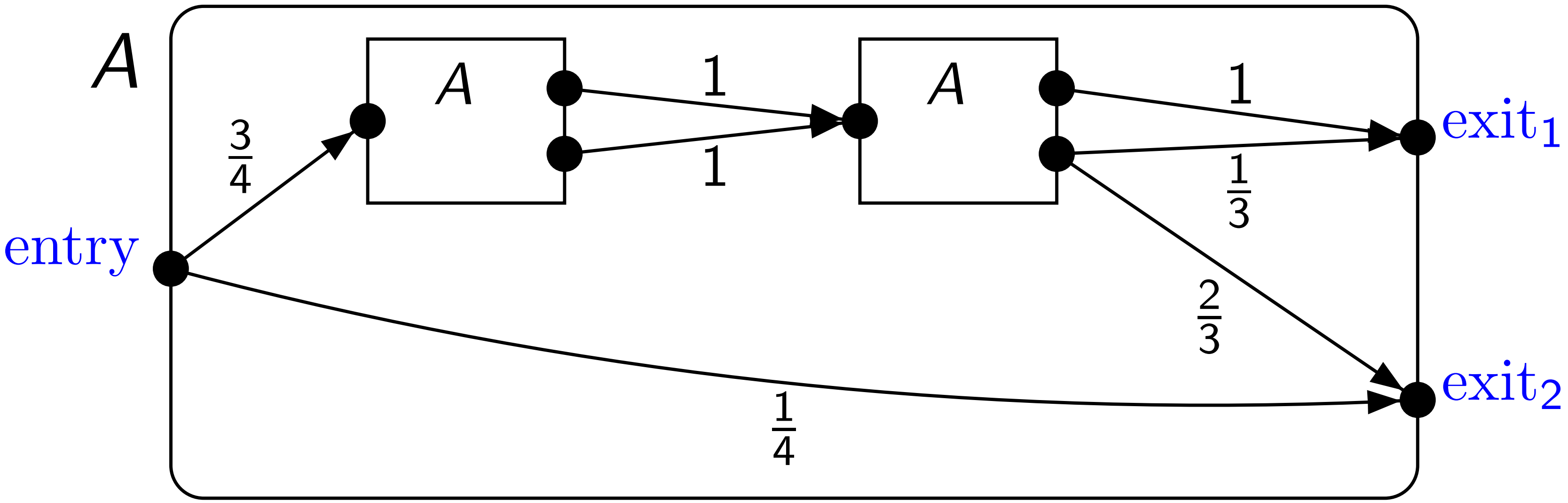
Recursive Markov Chains ($\approx$ tree-like-QBDs)



What is the probability of terminating at exit_2, starting at entry?

$$x_2 \quad =$$

Recursive Markov Chains ($\approx$ tree-like-QBDs)



What is the probability of terminating at exit$_2$, starting at entry?

$$x_2 = \frac{1}{4} + \frac{1}{2}x_2^2 + \frac{1}{2}x_1x_2 \qquad \text{(Note: coefficients sum to} > 1\text{)}$$

$$x_1 = \frac{3}{4}x_1^2 + \frac{3}{4}x_2x_1 + \frac{1}{4}x_1x_2 + \frac{1}{4}x_2^2$$

Fact: ([E.-Yannakakis'05]) The Least Fixed Point, $q^* \in [0,1]^n$, gives the termination probabilities.

approximation for Recursive Markov chains is "hard"

Theorem [E.-Yannakakis'05,JACM'09]

Any non-trivial approximation of the termination probabilities q^* of an RMC is PosSLP-hard.

In fact, deciding whether (a.) $q_1^* = 1$ or (b.) $q_1^* < \epsilon$, given the promise that one of the two is the case, is PosSLP-hard.

(Thus, even approximation in **NP** would yield a major breakthrough on the complexity of the BSS model and exact numerical computation; and P-time approximation is very unlikely.)

Note: this is despite the fact that Newton's method converges monotonically, starting from $\mathbf{0}$, to the LFP q^*, for all feasible MPSs.

Theorem [E.-Yannakakis'05b,JACM'15a]

For Recursive Markov Decision Processes, any non-trivial apporoximation of the optimal termination probabilities is not computable at all.

Conclusion

- We have established P-time algorithms for a number of fundamental analysis problems for various important classes of infinite-state ("structured") Markov chains and MDPs.
  (All of which are effectively subclasses of RMCs and RMDPs.)
- These are also key building blocks for efficient probabilistic model checking algorithms for these stochastic models.
- On the other hand, we have shown some complexity-theoretic "hardness" results relative to long-standing open problems (and even undecidability results) for approximating fundamental quantities for general RMCs (and RMDPs, respectively).
- Many, many, open questions remain.

Some papers

- K. Etessami and M. Yannakakis. Recursive Markov chains, stochastic grammars, and monotone systems of nonlinear equations. Journal of the ACM, 56(1), 2009.
- K. Etessami and M. Yannakakis. Recursive Markov decision processes and recursive stochastic games. Journal of the ACM, 62(2), 2015.
- A. Stewart, K. Etessami, and M. Yannakakis. Upper bounds for Newton's method on monotone polynomial systems, and P-time model checking of probabilistic one-counter automata. Journal of the ACM, 64(4), 2015.
- K. Etessami, A. Stewart, and M. Yannakakis. Polynomial time algorithms for multi-type branching processes and stochastic context-free grammars. Proceedings of *STOC*, 2012. Full version: arXiv:1201.2374
- K. Etessami, A. Stewart, and M. Yannakakis. Polynomial time algorithms for Branching Markov Decision Processes and Probabilistic Min/Max Polynomial Bellman Equations. Proceedings of ICALP, 2012. Full version: arXiv:1202.4798
- K. Etessami, D. Wojtczak, and M. Yannakakis. Quasi-Birth-Death Processes, Tree-like QBDs, Probabilistic 1-Counter Automata, and Pushdown Systems. QEST'08, and Performance Evaluation, 67(9):837-857, 2010.
- T. Brazdil, V. Brozek, K. Etessami, & A. Kucera. Approximating the termination value of one-counter MDPs and stochastic games, ICALP'11 and Information and Computation, 222(2):121-138, 2013.

Other related papers accessible from my web page.