A Taxonomy of Fixed point Computation Problems for Algebraically-Defined Functions and their Computational Complexity

Kousha Etessami

LFCS, School of Informatics
University of Edinburgh

Based on joint works with:

Mihalis Yannakakis	&	Alistair Stewart
Columbia Uni.		U. of Edinburgh

Workshop on "Solving Polynomial Equations"
Simons Institute, Berkeley
October 2014

Algorithms for Branching Markov Decision Processes and probabilistic min/max polynomial Bellman equations

Kousha Etessami

University of Edinburgh

Based on joint works with:

Alistair Stewart & Mihalis Yannakakis

U. of Edinburgh Columbia Uni.

Workshop on "Solving Polynomial Equations"
Simons Institute, Berkeley
October 2014

Fixed Point Computation

Brouwer's fixed point theorem

Every continuous function $F : D \to D$ from a compact convex set $D \subseteq \mathbb{R}^m$ to itself has a fixed point, i.e., $\exists\, x^* \in D$ such that $F(x^*) = x^*$.

Fixed Point Computation

Brouwer's fixed point theorem

Every continuous function $F : D \to D$ from a compact convex set $D \subseteq \mathbb{R}^m$ to itself has a fixed point, i.e., $\exists\, x^* \in D$ such that $F(x^*) = x^*$.

Computation Task: "Given" $F(x)$, compute/approximate a fixed point.

Fixed Point Computation

Brouwer's fixed point theorem

Every continuous function $F : D \to D$ from a compact convex set $D \subseteq \mathbb{R}^m$ to itself has a fixed point, i.e., $\exists\, x^* \in D$ such that $F(x^*) = x^*$.

Computation Task: "Given" $F(x)$, compute/approximate a fixed point.

Two different notions of ϵ-approximation of a fixed point:

- (**Almost**) Given $F : D \to D$, compute $x' \in D \cap \mathbb{Q}^m$ such that:
$$\|F(x') - x'\|_\infty < \epsilon$$
- (**Near**) Given $F : D \to D$, compute $x' \in D \cap \mathbb{Q}^m$ s.t. there exists $x^* \in D$ where $F(x^*) = x^*$ and:
$$\|x^* - x'\|_\infty < \epsilon$$

These two notions can have rather different complexity characteristics. In this talk, we are interested in Near.

The complexity class FIXP and FIXP_a

FIXP (FIXP_a) is a class of real-valued (respectively, discrete) total search problems:

FIXP (FIXP_a)

- **Input:** algebraic circuit, a.k.a., straight-line program, using gates $\{ + , * , \max \}$ and rational constants, having n input variables $x = (x_1, \ldots, x_n)$, and n output gates, such that the circuit represents a continuous function $F : [0, 1]^n \mapsto [0, 1]^n$.

  (We are also given an error parameter $\epsilon > 0$ as input for FIXP_a.)

- **Output:** Compute a (ϵ-near approximate) fixed point of F.

Close these search problems under suitable (P-time) reductions.

The resulting class is called FIXP (respectively, FIXP_a).

(near approximation of) Nash Equilibrium is FIXP$_{(a)}$-complete

Theorem ([E.-Yannakakis'07])

Computing a (ϵ-near approximation of) a Nash Equilibrium for a game Γ with 3 or more players, given Γ (and given $\epsilon > 0$), is FIXP*-complete (respectively,* FIXP$_a$*-complete).*

PPAD (**Papadimitriou (1992)**): given a succinctly represented directed graph with in-degree ≤ 1 & out-degree ≤ 1, and given a source node (indegree $= 0$), find some other source or sink node. (Closing this search problem under P-time reductions yields PPAD.)

Let linear-FIXP denote the subclass of FIXP where the algebraic circuits are restricted to gates $\{+, \max\}$ and multiplication by rational constants.

Theorem ([E.-Yannakakis'07])

The following are all P-time equivalent:

1. PPAD
2. linear-FIXP
3. *exact fixed point problem for "polynomial piecewise-linear functions".*
4. *(cf. [Scarf'67]) ϵ-almost-fixed point computation for "polynomially computable" and "polynomially continuous" functions, $F_I(x)$, given instance I, and $\epsilon > 0$.*
5. *[Mehta, 2014]: 2-variable*-linear-FIXP

By Scarf's algorithm, computing a ϵ-NE is in PPAD.

By the Lemke-Howson algorithm, computing a exact NE for 2-player games is in PPAD.

Theorem

1. *[Daskalakis-Goldberg-Papadimitriou'06], [Chen-Deng'06]:*
   Computing a ϵ-NE for a 3 player game is PPAD*-complete.*
2. *[Chen-Deng'06]:*
   Computing an exact (rational) NE for a 2 player game is PPAD*-complete.*

Note: Scarf's algorithm does *not* in general yield a point ϵ-near a fixed point.

A “hard” problem

[Allender, Bürgisser, Kjeldgaard-Pedersen, Miltersen,2006]

PosSLP: Given an arithmetic circuit (Straight Line Program) with gates $\{+, *, -\}$, and with input 1, decide whether the output value is positive.

PosSLP captures the power of P-time in the unit-cost arithmetic RAM model of computation.

Theorem [ABKM’06]

PosSLP is decidable in the Counting Hierarchy: $P^{PP^{PP^{PP}}}$.

(Nothing better is known.)

PosSLP $\leq_p$ any near approximation of a 3-player NE

Theorem ([E.-Yannakakis'07])

Any non-trivial near approximation of an NE is **PosSLP***-hard.*

More precisely: for every fixed $\epsilon > 0$,

PosSLP *is P-time reducible to the following problem:*

Given a 3-player normal form game, Γ, with the promise that:

1. *Γ has a unique NE, x^*, which is fully mixed, and*
2. *In x^*, the probability that player 1 plays pure strategy α is either:*

$$\text{(a.)} \quad < \epsilon \quad , \qquad \text{or} \qquad \text{(b.)} \quad \geq (1 - \epsilon)$$

Decide which of (a.) or (b.) is the case.

What makes a fixed point problem “hard” or “easy”??

Note: These problems are in general not NP-hard, because existence of a solution (fixed point) is guaranteed.

What makes a fixed point problem “hard” or “easy”??

Note: These problems are in general not NP-hard, because existence of a solution (fixed point) is guaranteed.

- PPAD-hardness captures a combinatorial difficulty for computing, or even almost-approximating, a fixed point.

What makes a fixed point problem "hard" or "easy"??

Note: These problems are in general not NP-hard, because existence of a solution (fixed point) is guaranteed.

- PPAD-hardness captures a combinatorial difficulty for computing, or even almost-approximating, a fixed point.
- But there can also be an additional numerical, difficulty for near-approximating a fixed point, which is not captured by PPAD-hardness.

  It is captured by PosSLP-hardness.

These two kinds of difficulties are somewhat "orthogonal".

FIXP_a-complete problems have both of these difficulties.

Rich landscape within FIXP:

Numerical Difficulty

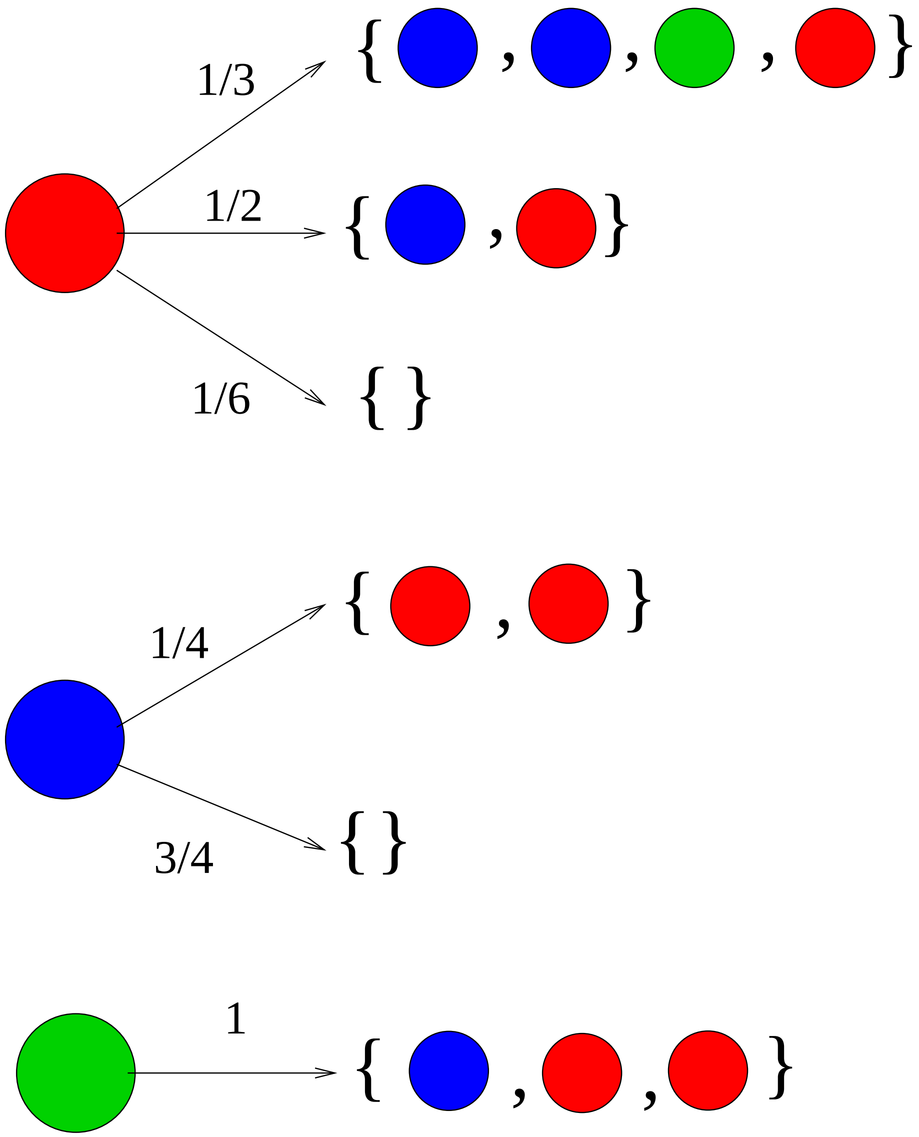
	No	P.G.–hard	PPAD–hard
PosSLP–hard *	approx–Recursive Markov chains exact–Branching processes exact–Branching–MDPs	exact–Branch–simple–stoc–game approx–Unique–nonlinear Brouwer fixed point ??approx–Unique–3–player–Nash?? exact–concurrent–stocastic–game exact–Shapley–stocastic–game	approx–3–player–Nash approx–nonlin.–Arrow–Debreu market equilibrium approx–nonlinear Brouwer fixed point FIXP$_a$–complete
* No	PIT / ACIT exact–linear–Arrow–Debreu market equilibrium approx–Branching–MDPs approx–Branching–process exact–MDPs	approx–Branch–simple–stoc–game approx–Shapley–stochastic–game exact–Unique–piecewise–linear Brouwer fixed point ??exact–Unique–2–player–Nash?? exact–Condon–simple–stoc–game exact–mean–payoff–game parity–game	exact–2–player–Nash exact–Arrow–Debreu market equilibrium with SPLC utilities exact–piecewise–linear Brouwer fixed point "Almost"– nonlinear–Brouwer fixed point "Almost"–(epsilon)–Nash for >= 3 players

Combinatorial Difficulty

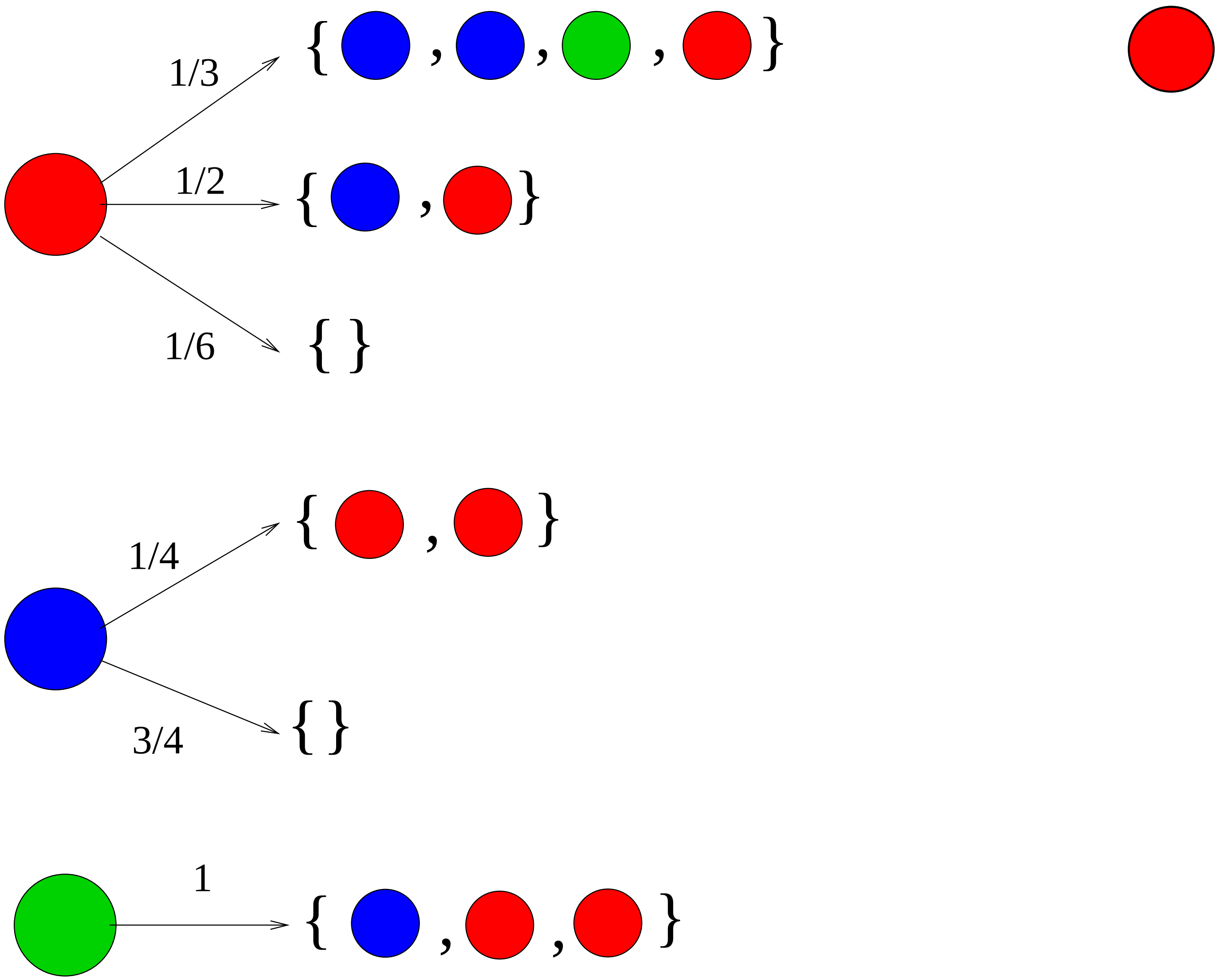
For the rest of this talk, our focus will be on fixed point problems for monotone algebraically-defined functions.

(These arise in many applications, as we shall see.)

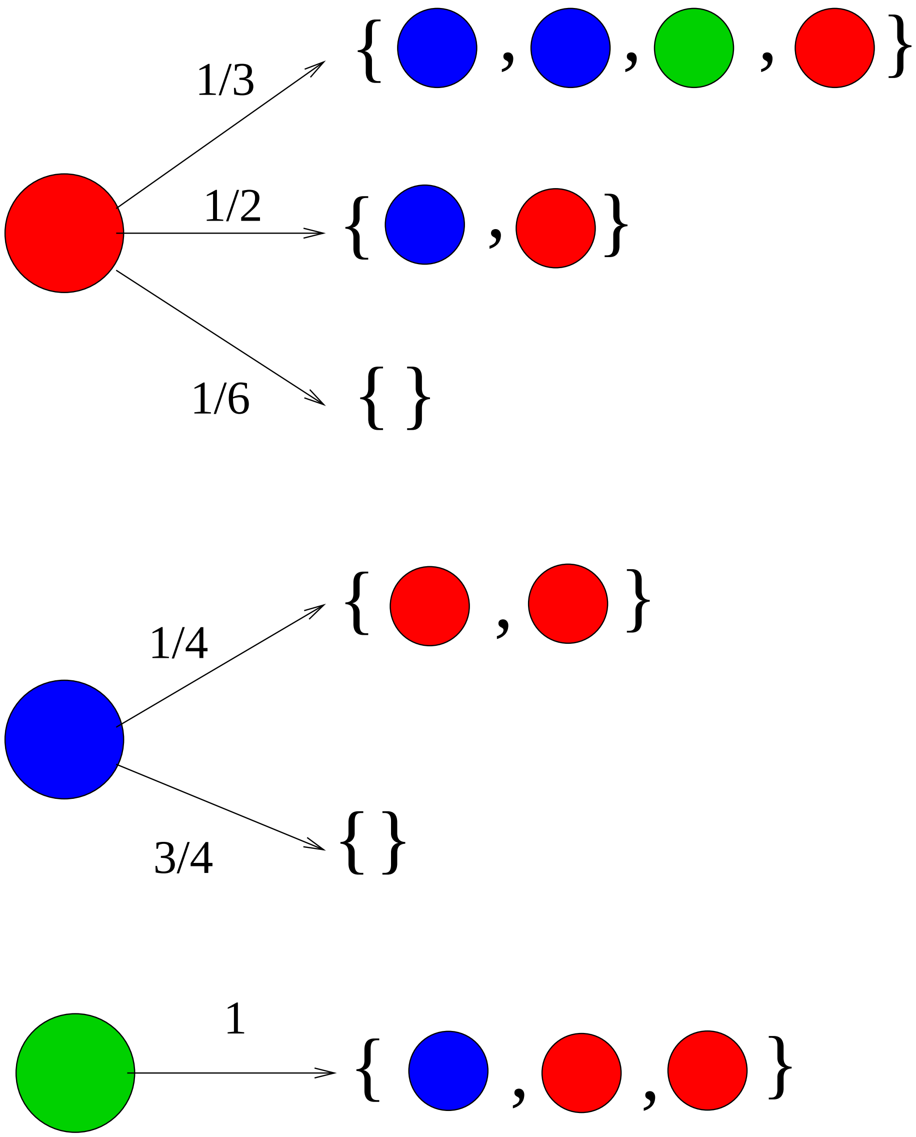
Multi-type Branching Processes (Kolmogorov,1940s)

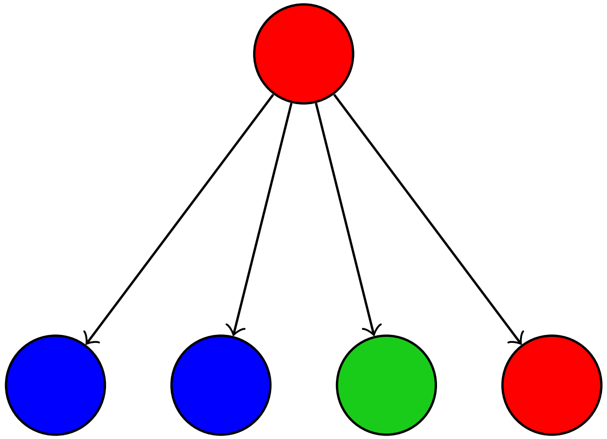


Multi-type Branching Processes (Kolmogorov,1940s)

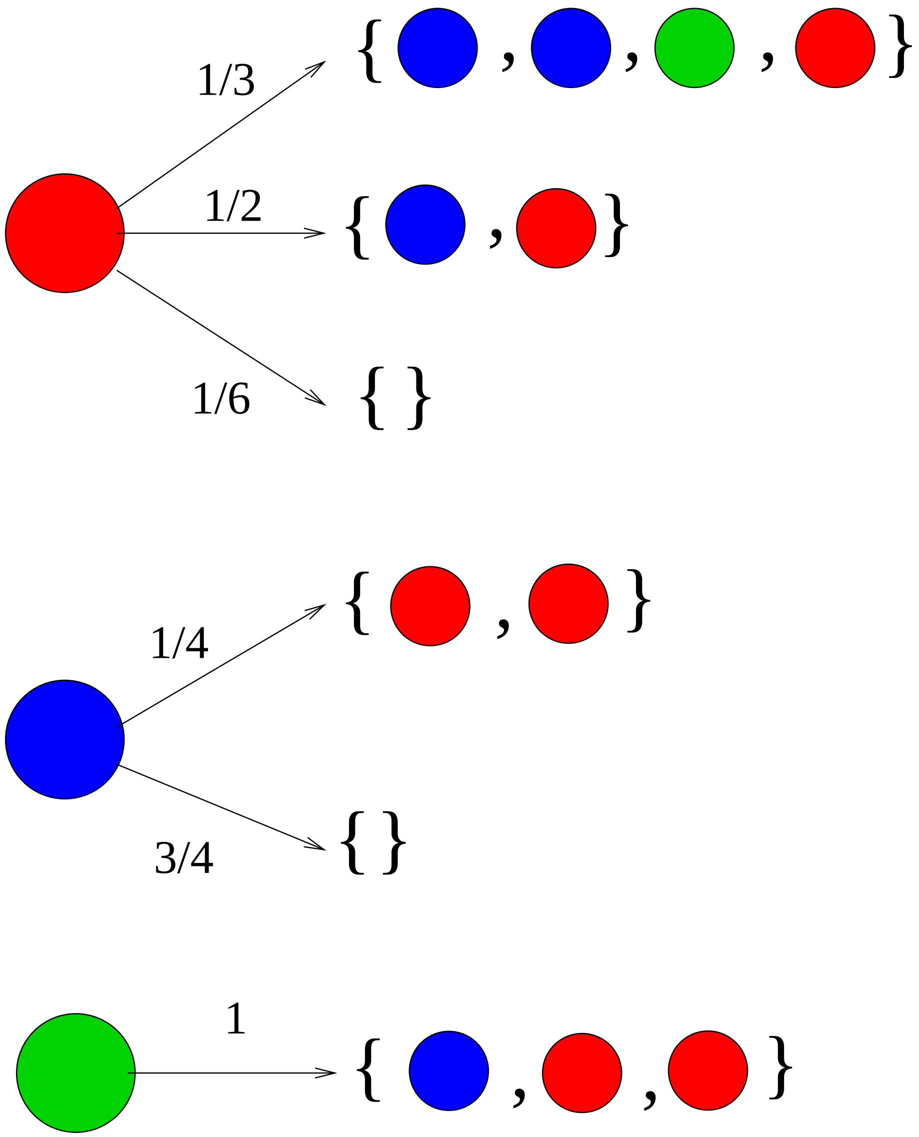


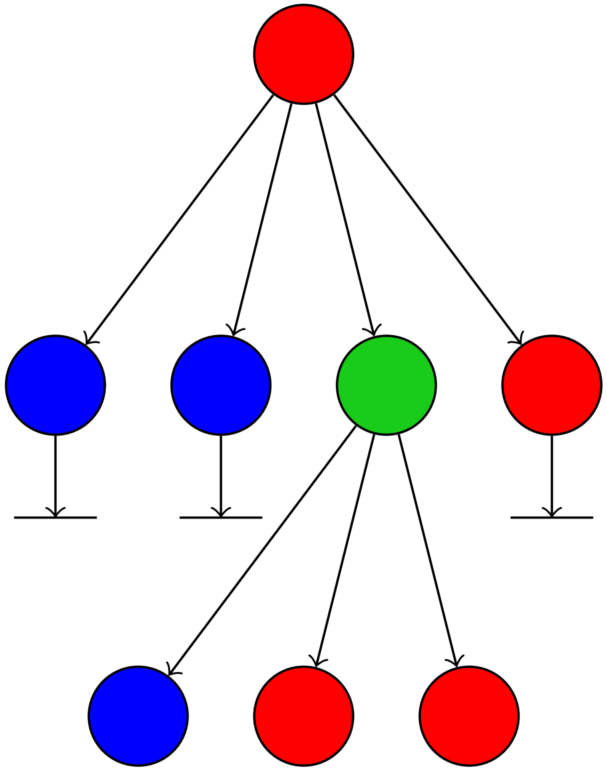
Multi-type Branching Processes (Kolmogorov,1940s)



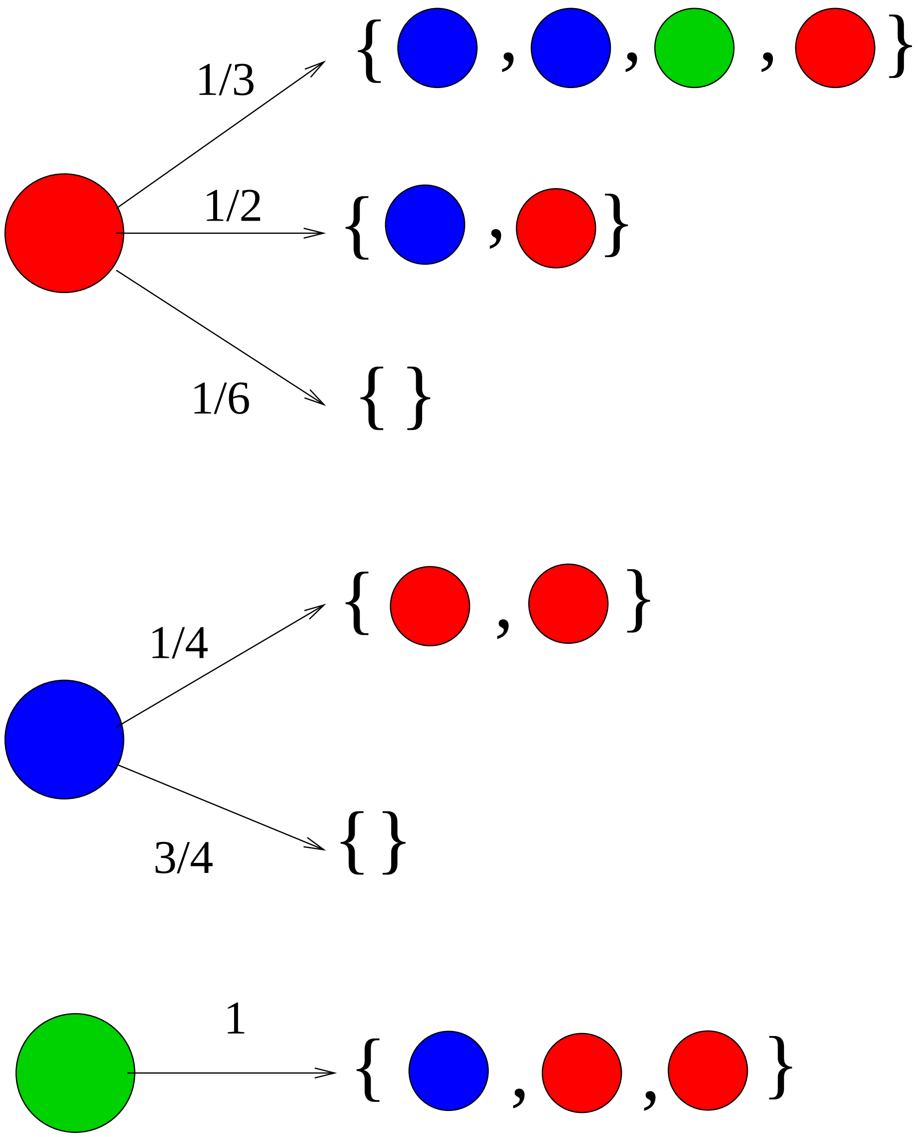


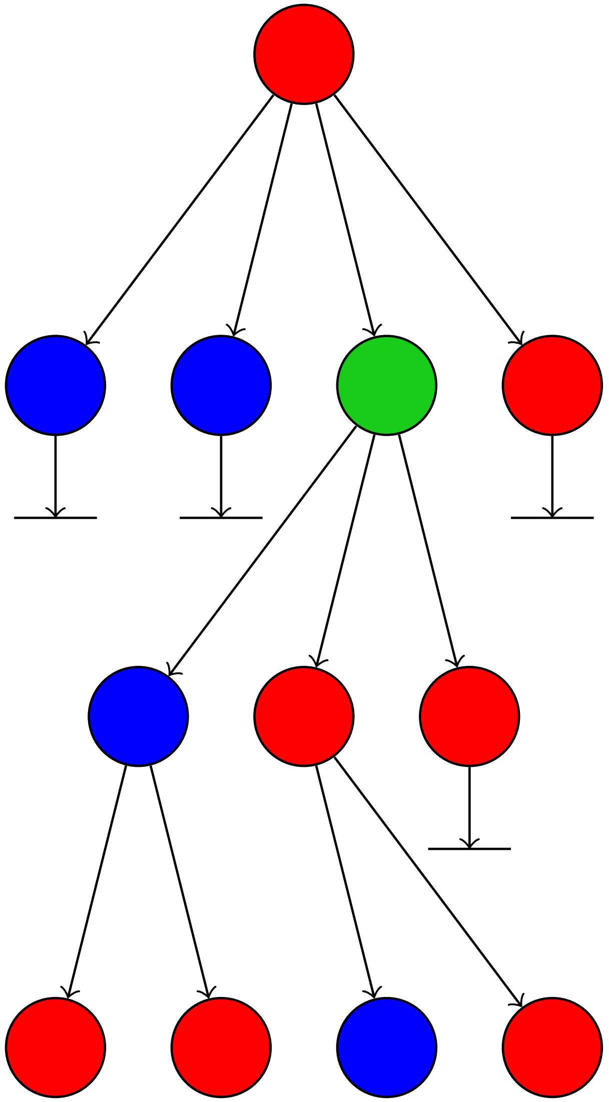
Multi-type Branching Processes (Kolmogorov,1940s)



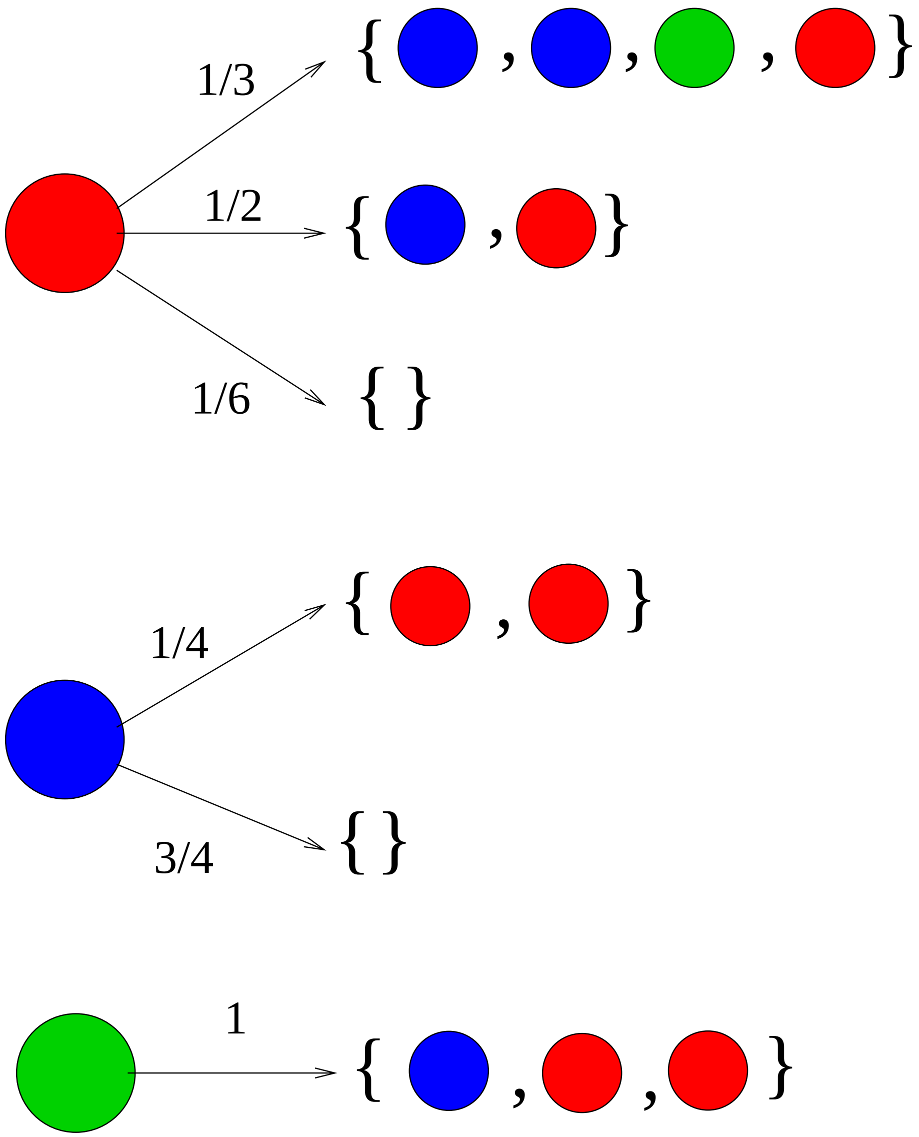


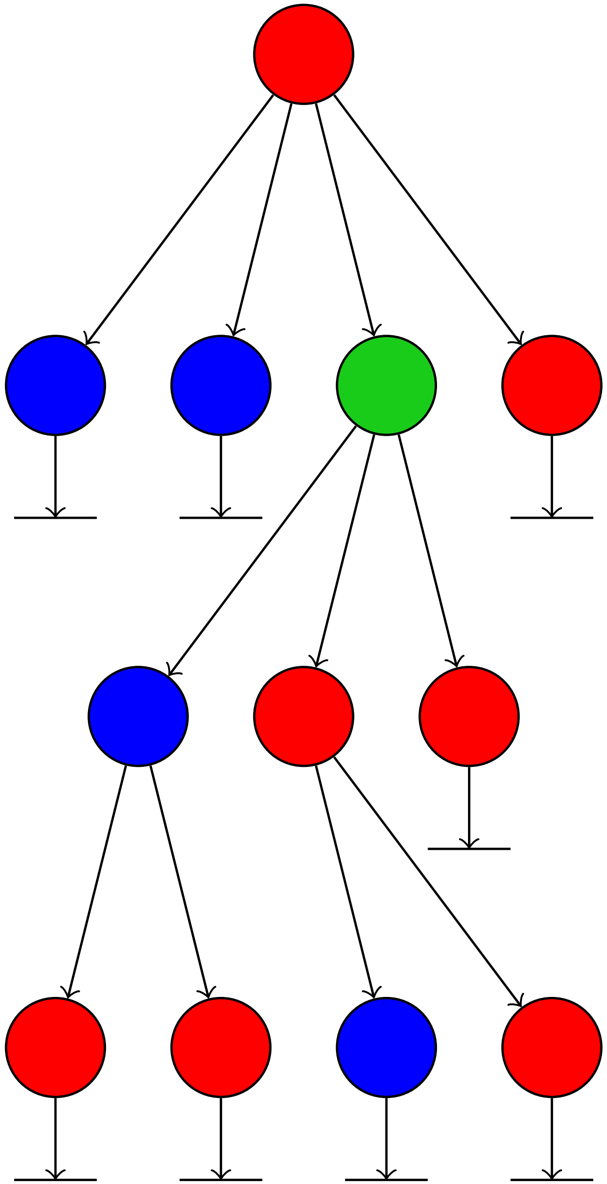
Multi-type Branching Processes (Kolmogorov,1940s)



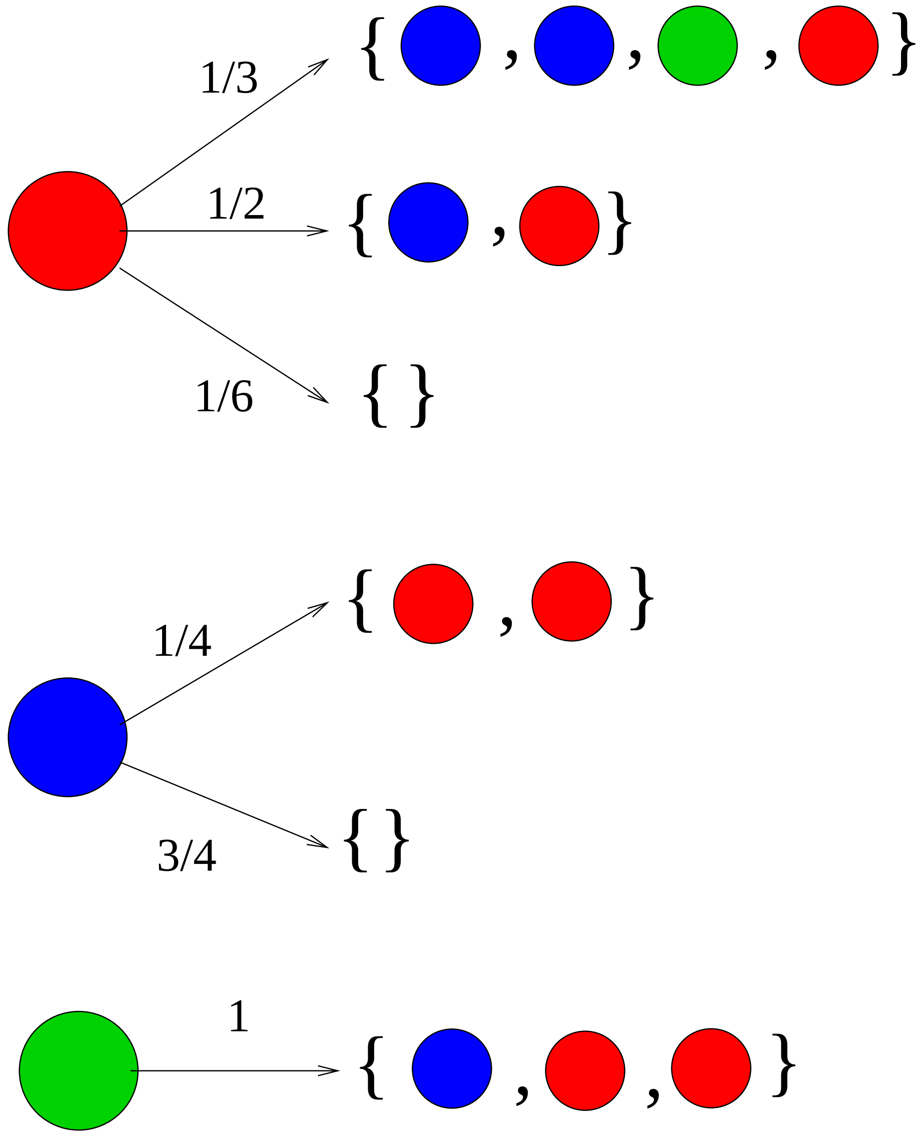


Multi-type Branching Processes (Kolmogorov,1940s)



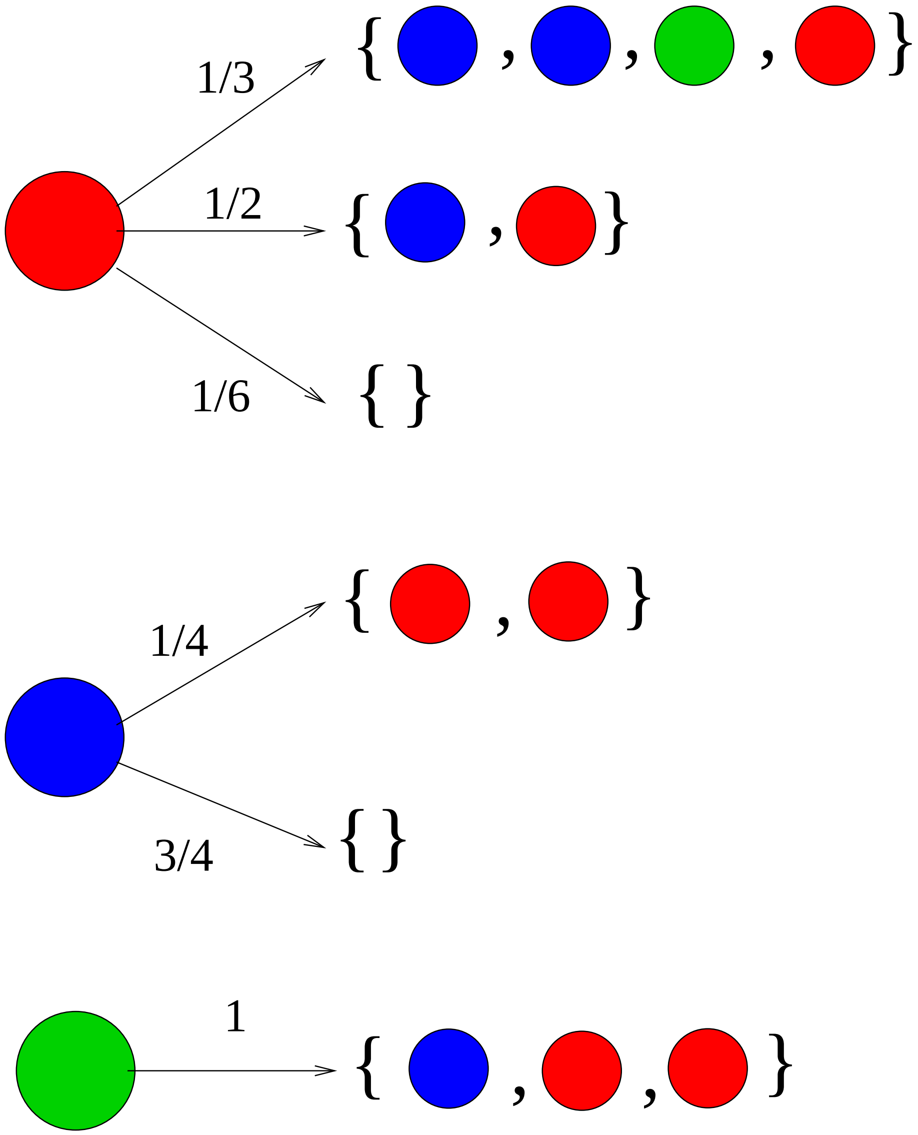


Multi-type Branching Processes (Kolmogorov 1940s)



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

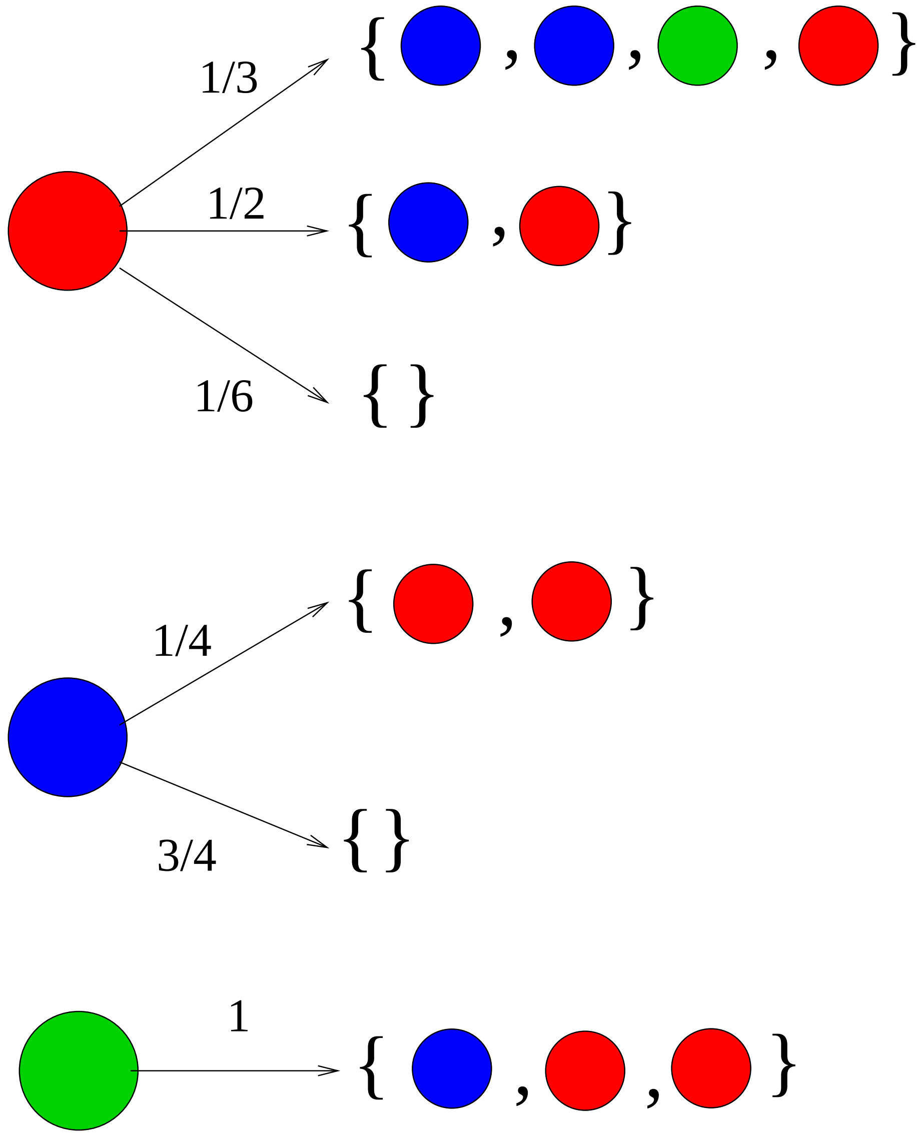
Multi-type Branching Processes (Kolmogorov 1940s)



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

$$x_R \;=$$

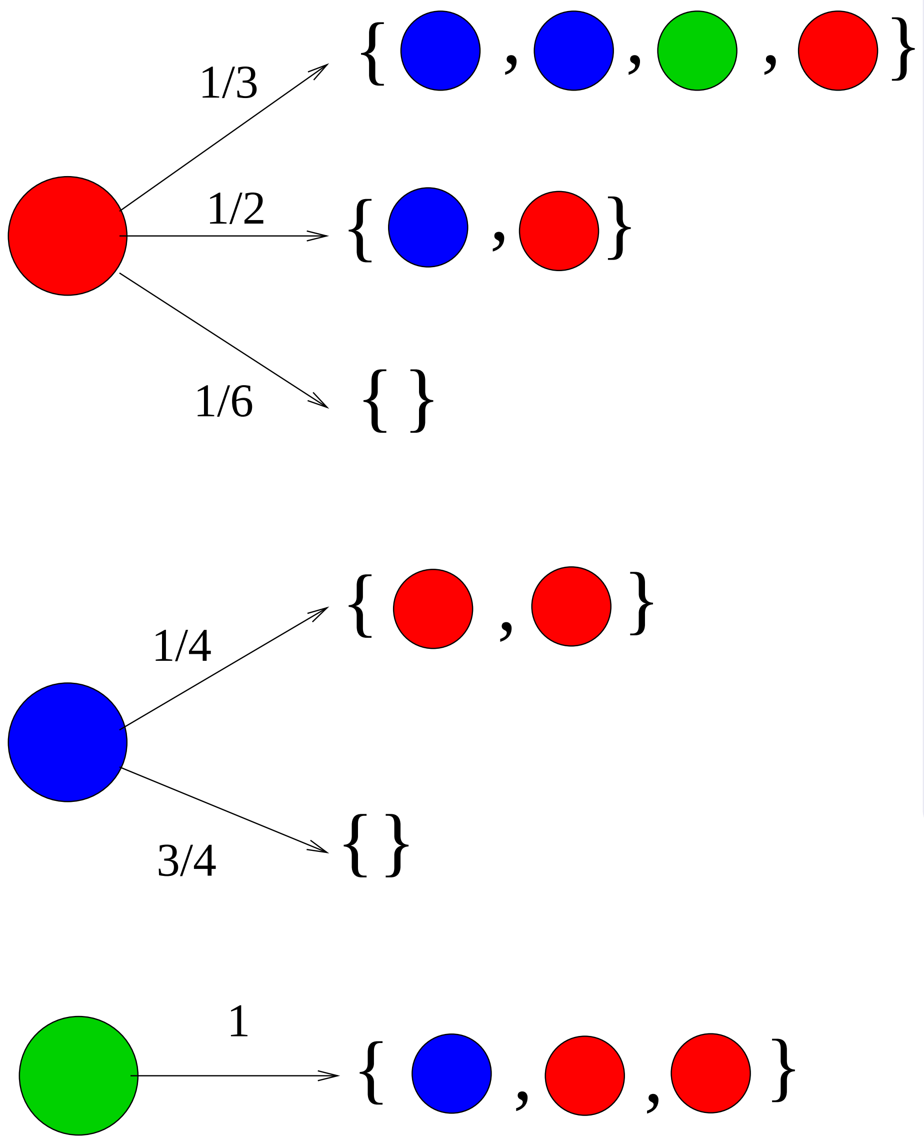
Multi-type Branching Processes (Kolmogorov 1940s)



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

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

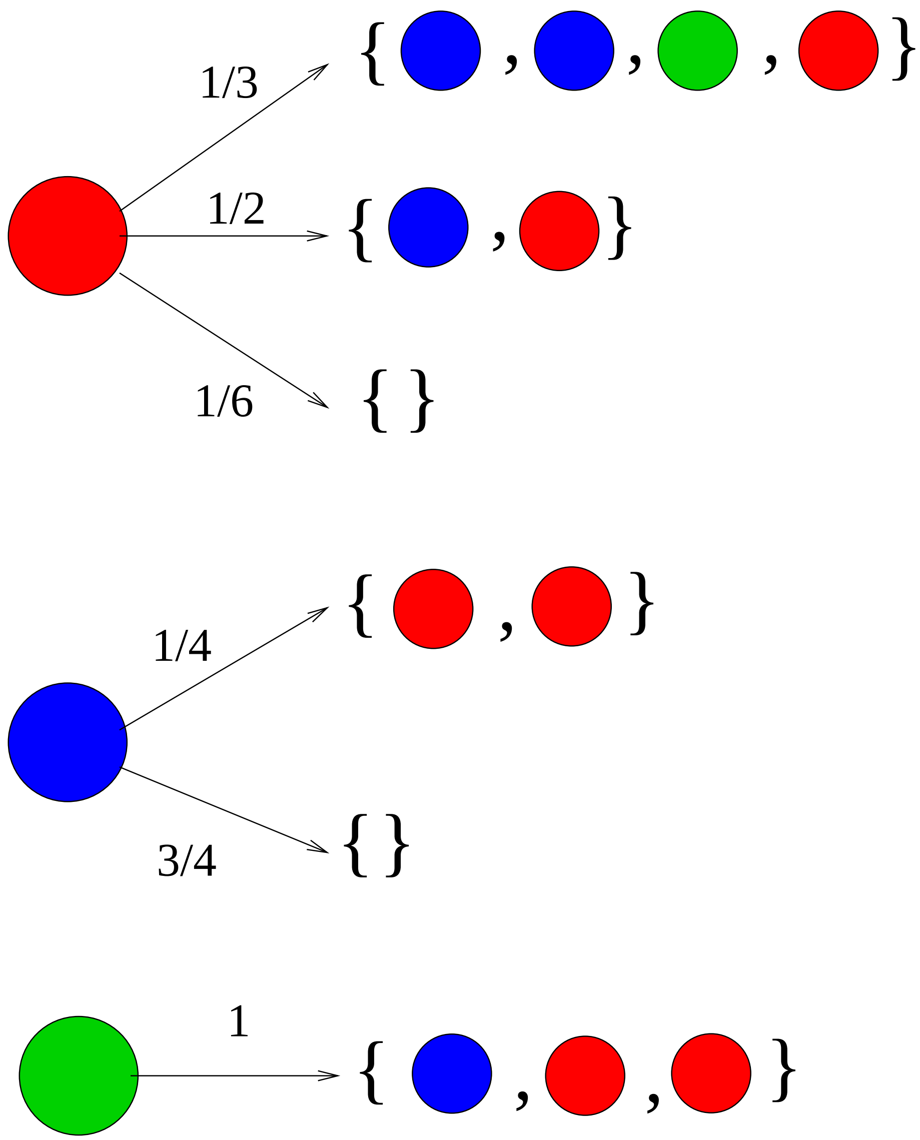
Multi-type Branching Processes (Kolmogorov 1940s)



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

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

Multi-type Branching Processes (Kolmogorov 1940s)



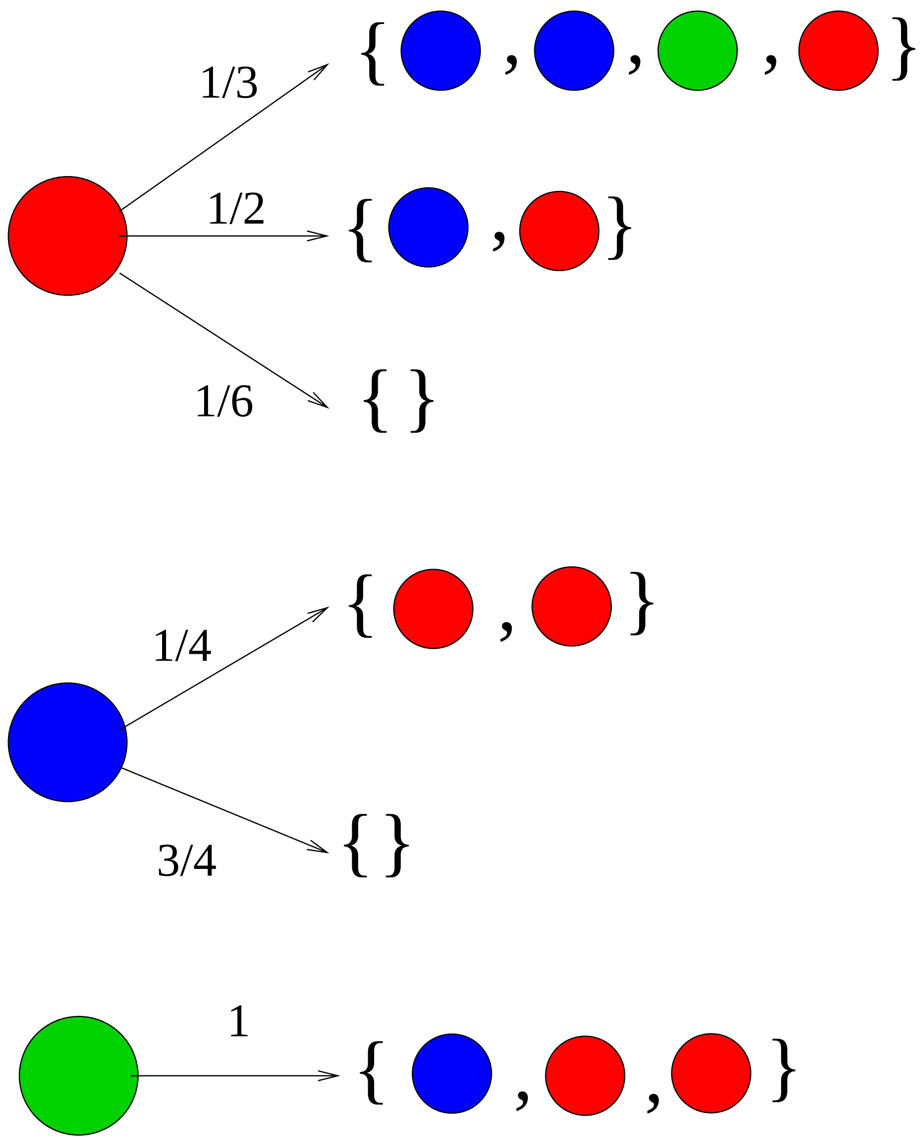
Question: What is the probability of eventual extinction, starting with one ● ?

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

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 ● ?

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

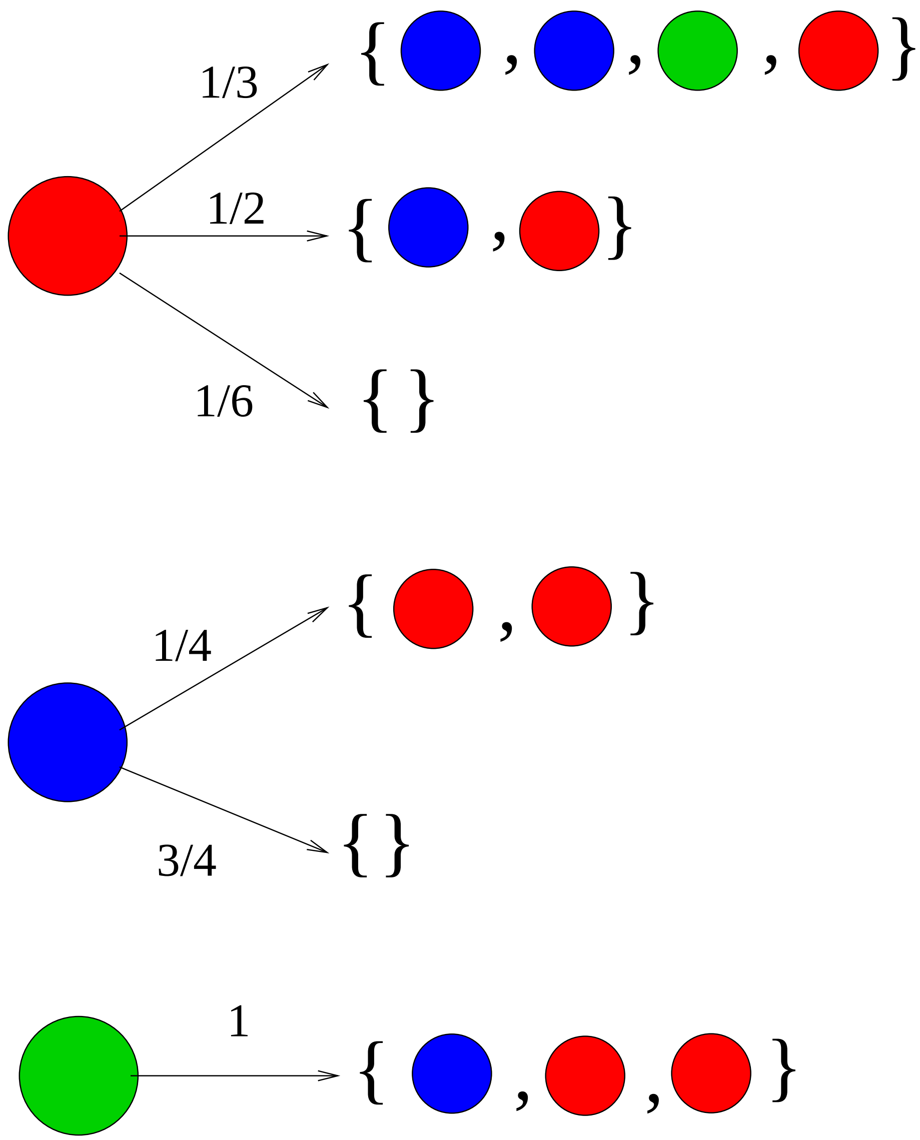
We get nonlinear fixed point equations:

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

Fact

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

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 $\bar{\mathbf{x}} = P(\bar{\mathbf{x}})$. $q_R^* = 0.276$; $q_B^* = 0.769$; $q_G^* = 0.059$.

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 corresponds to a PPS with n variables, and vice-versa.

Monotone Polynomial Systems of Equations

$$5x_B^2x_Gx_R + 2x_Bx_R + \frac{1}{6}$$

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.

Basic properties of PPSs and MPSs

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

For a MPS, $P : [0,+\infty]^n \to [0,+\infty]^n$ defines monotone map on $[0,+\infty]^n$.

Proposition

- *A PPS, $x = P(x)$ has a least fixed point (LFP), $q^* \in [0,1]^n$.*
  (q^* can be irrational.)
- *A MPS $x = P(x)$ has a LFP, $q^* \in [0,+\infty]^n$.*
  (The MPS is called feasible if $q^* \in \mathbb{R}^n_{\geq 0}$.)
- $q^* = \lim_{k\to\infty} P^k(\mathbf{0})$*, for both PPSs and MPSs.*
- *For a PPS, q^* is the vector of extinction probabilities for the corresponding BP. (For a MPS, q^* is the partition function of the corresponding WCFG.)*

Question: Can we compute q^* efficiently (in P-time)?

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

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 $\mathbf{q}^*$, starting from $\mathbf{x}^{(0)} := \mathbf{0}$, for PPSs, and more generally for all feasible MPSs.

But...

- In [E.-Yannakakis'05,'09], we gave no upper bounds on # of iterations needed for PPSs or MPSs.
- We proved PosSLP-hardness for any nontrivial approximation of the LFP $\mathbf{q}^* \in [0,1]^n$ of MPSs corresponding to Recursive Markov Chains.

A “hard” problem

[Allender, Bürgisser, Kjeldgaard-Pedersen, Miltersen,2006]

PosSLP: Given an arithmetic circuit (Straight Line Program) with gates $\{+, *, -\}$, and with input 1, decide whether the output value is positive.

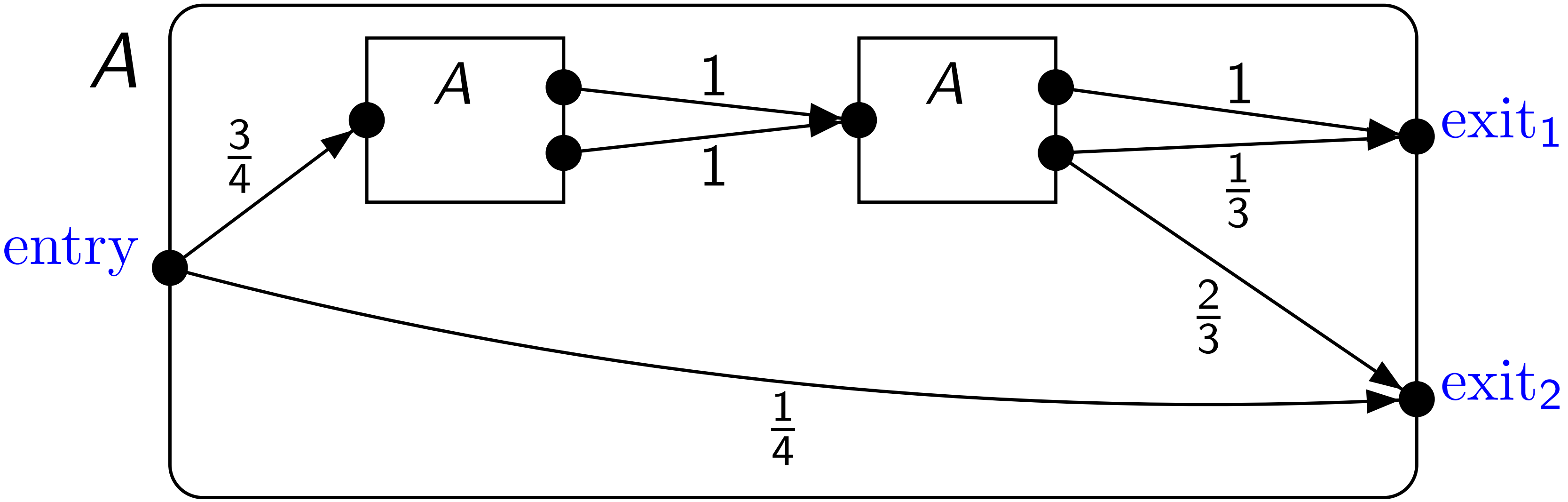
PosSLP captures the power of P-time in the unit-cost arithmetic RAM model of computation.

Theorem [ABKM’06]

PosSLP is decidable in the Counting Hierarchy: $P^{PP^{PP^{PP}}}$.

(Nothing better is known.)

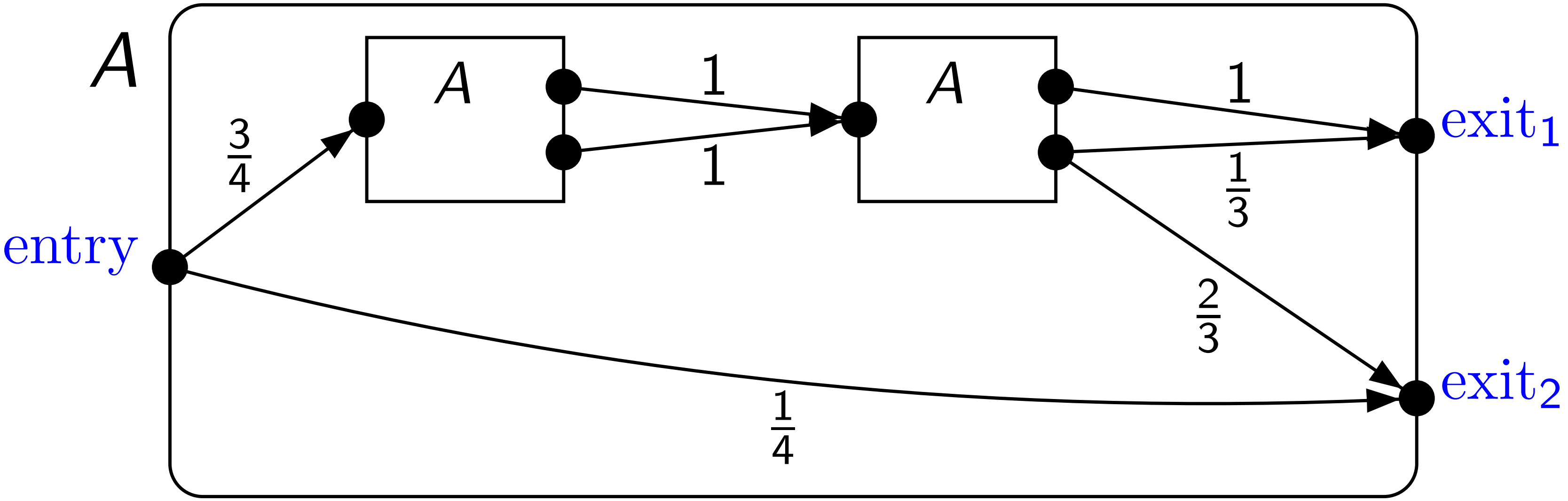
Recursive Markov Chains



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

$$x_2 \quad =$$

Recursive Markov Chains



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

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

(Note: coefficients sum to > 1)

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

approximation for Recursive Markov chains

Theorem

1. *[EY'07]: Any non-trivial approximation of the termination probabilities q^* of an RMC is PosSLP-hard:*

   Deciding whether (a.) $q_1^* = 1$ or (b.) $q_1^* < \epsilon$, is PosSLP-hard.

2. *[ESY'12]: ϵ-approximation of q^* is in* FIXP_a.

   (It can be reduced to approximating a unique Brouwer fixed point, and to approximating an (actual) Nash equilibrium of a game.)

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

[Esparza,Kiefer,Luttenberger,'07,'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 $|P|$.

More recently, in [Stewart-E.-Yannakakis'13], we have established a matching exponential upper bound in $|P|$ for arbitrary PPSs and feasible MPSs.

P-time approximation for PPSs

Theorem ([E.-Stewart-Yannakakis,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 a 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 a 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 a 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 ([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 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...

The quantitative decision problem for PPSs is PosSLP-equivalent

Theorem ([E.-Yannakakis'07, E.-Stewart-Yannakakis'12])

Given a PPS, $x = P(x)$, and a probability p, deciding whether $q_i^* < p$ is PosSLP-equivalent.

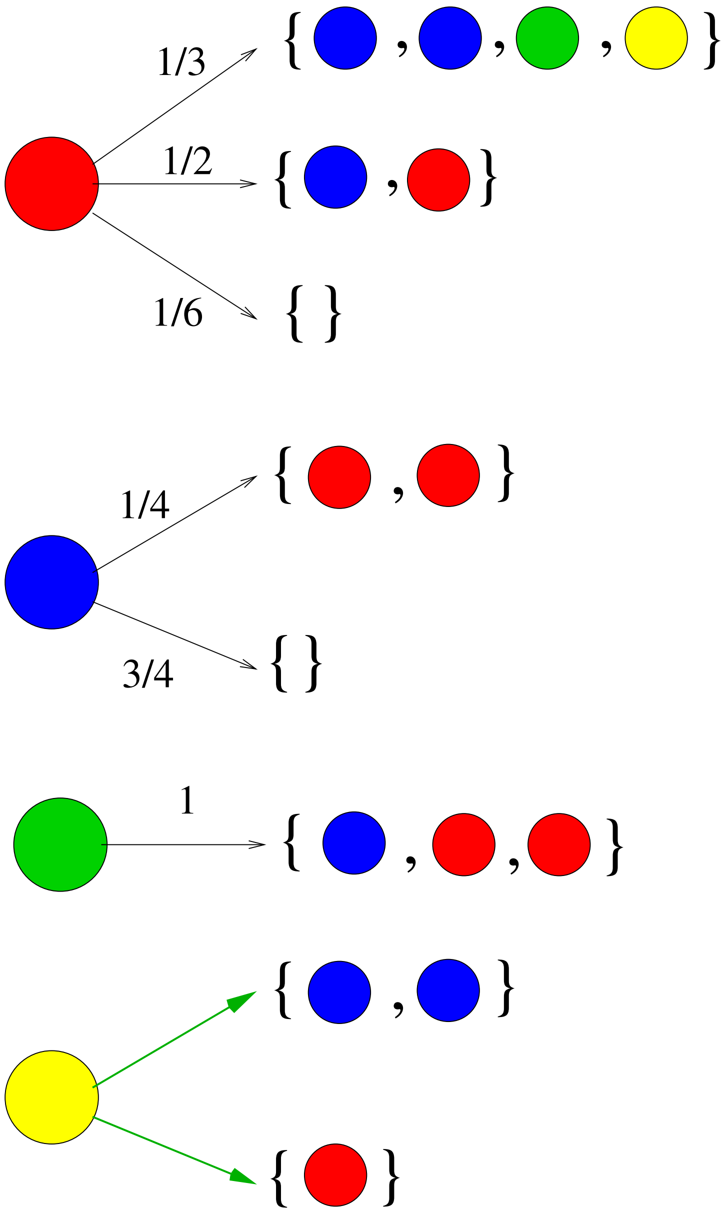
Reduction to PosSLP exploits quadratic convergence with explicit & "good" constants:

Theorem ([ESY'12])

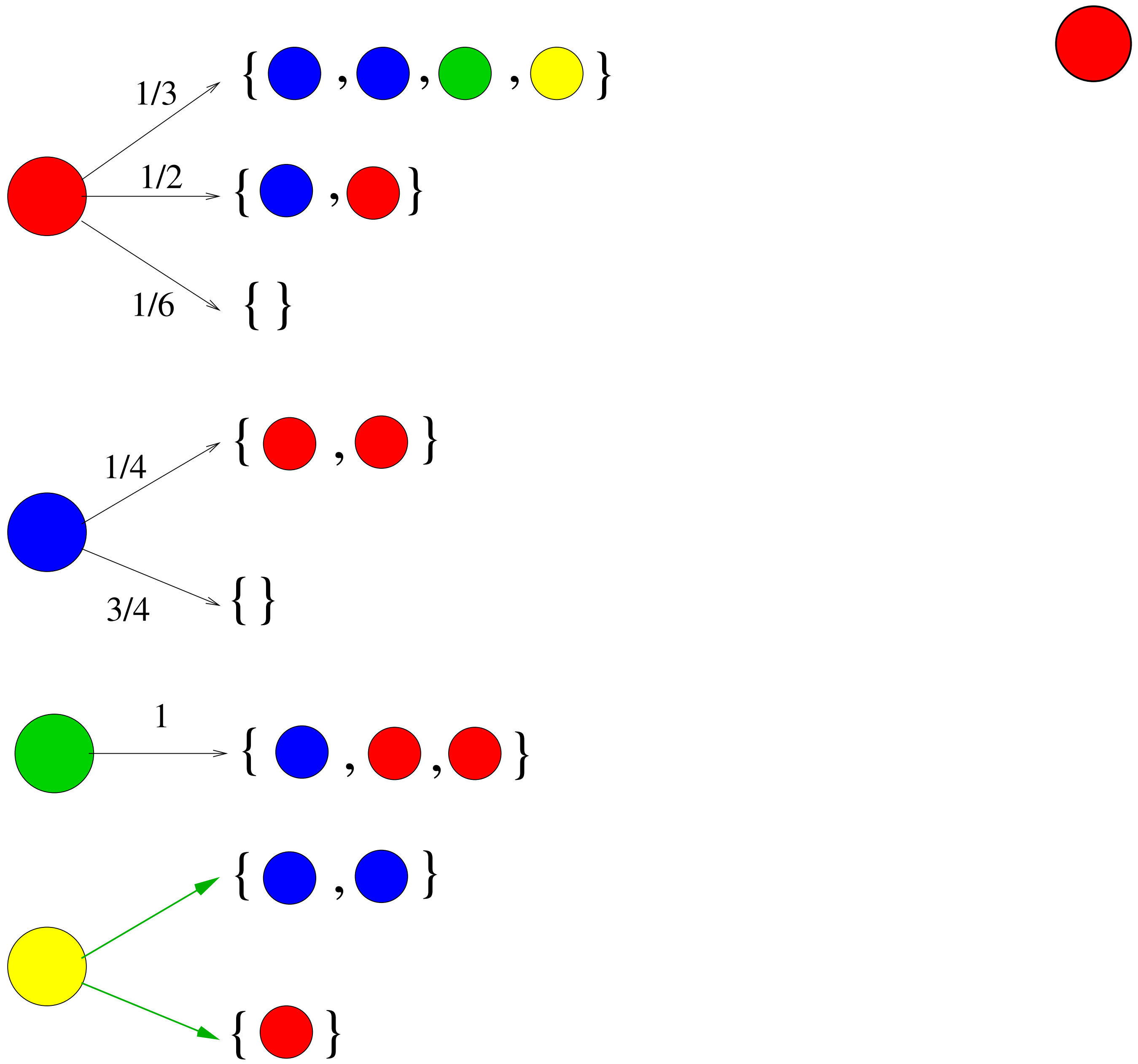
Given a PPS $\mathbf{x} = P(\mathbf{x})$ with LFP $\mathbf{0} < \mathbf{q}^* < \mathbf{1}$, if we apply Newton starting at $\mathbf{x}^{(0)} = \mathbf{0}$, then

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

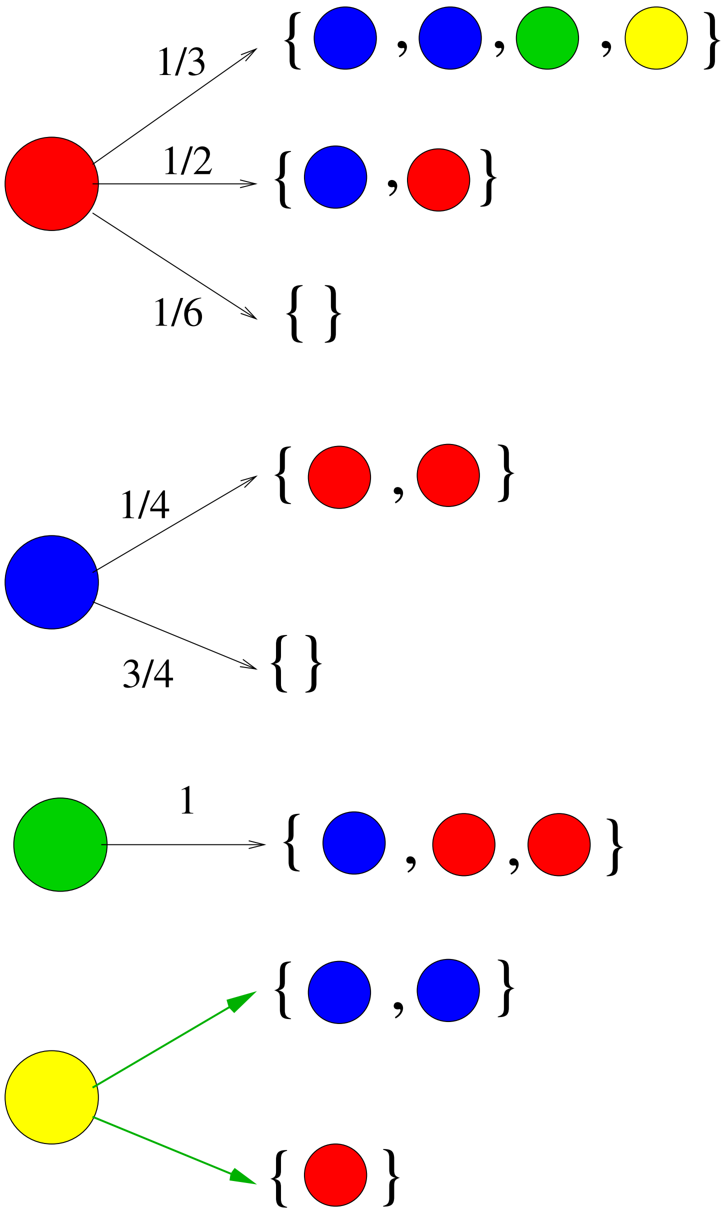
Branching Markov Decision Processes

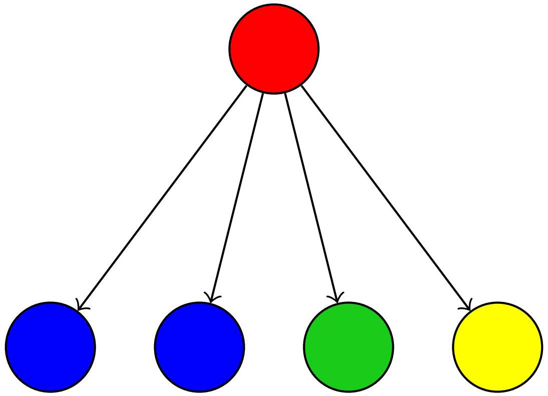


Branching Markov Decision Processes

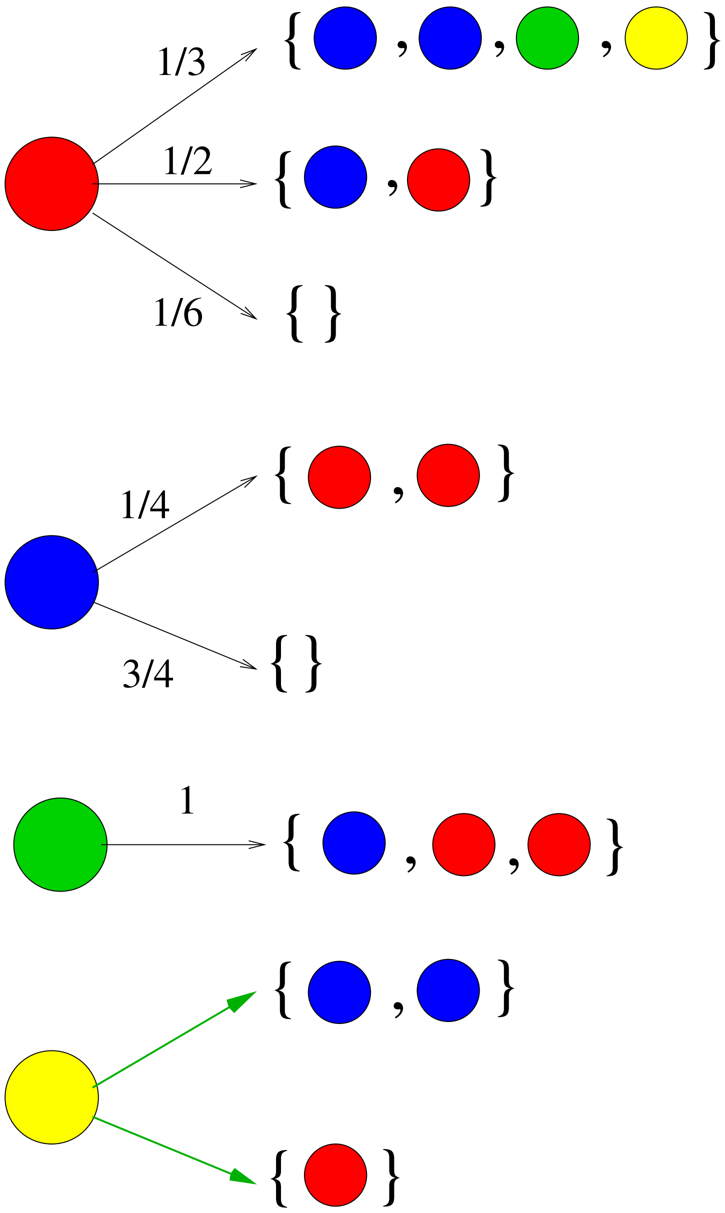


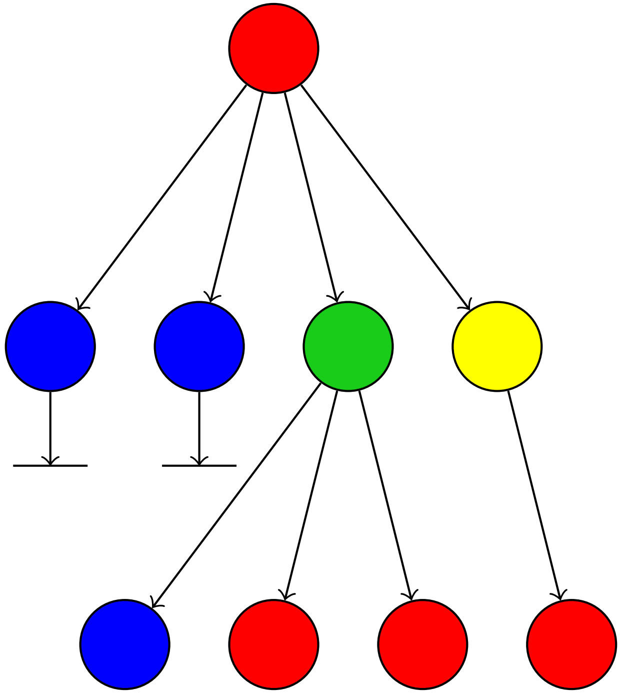
Branching Markov Decision Processes



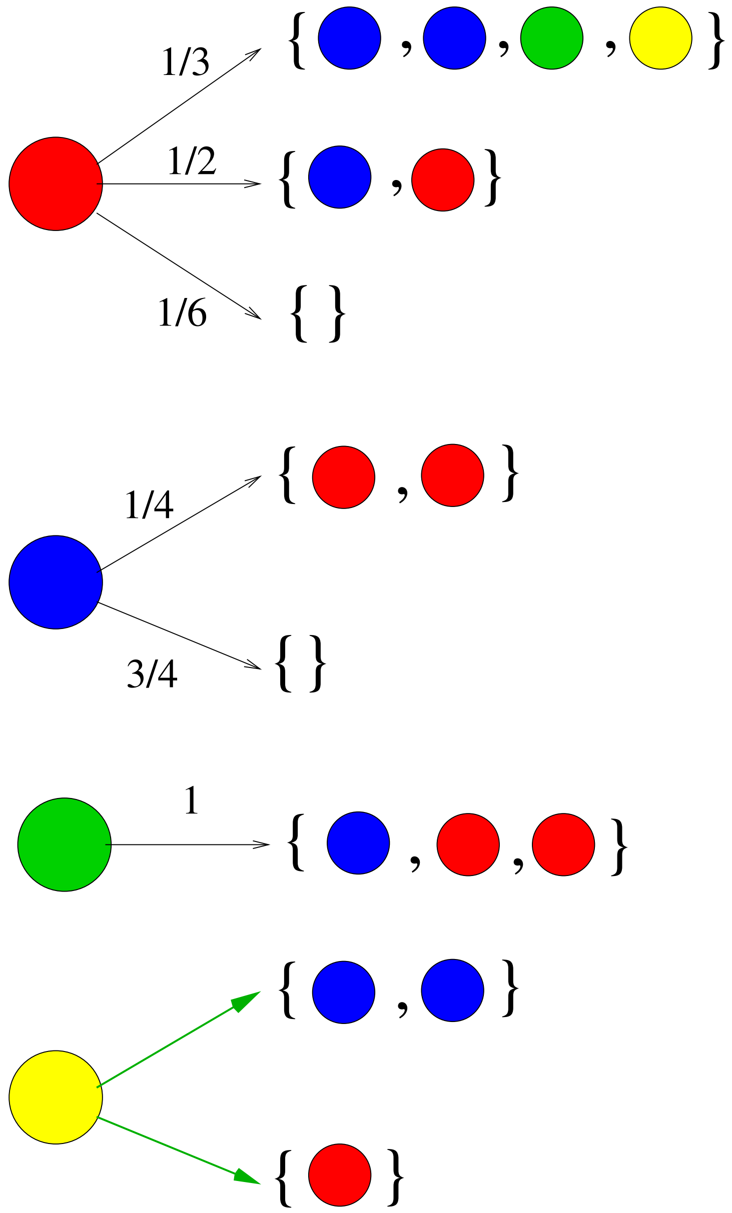


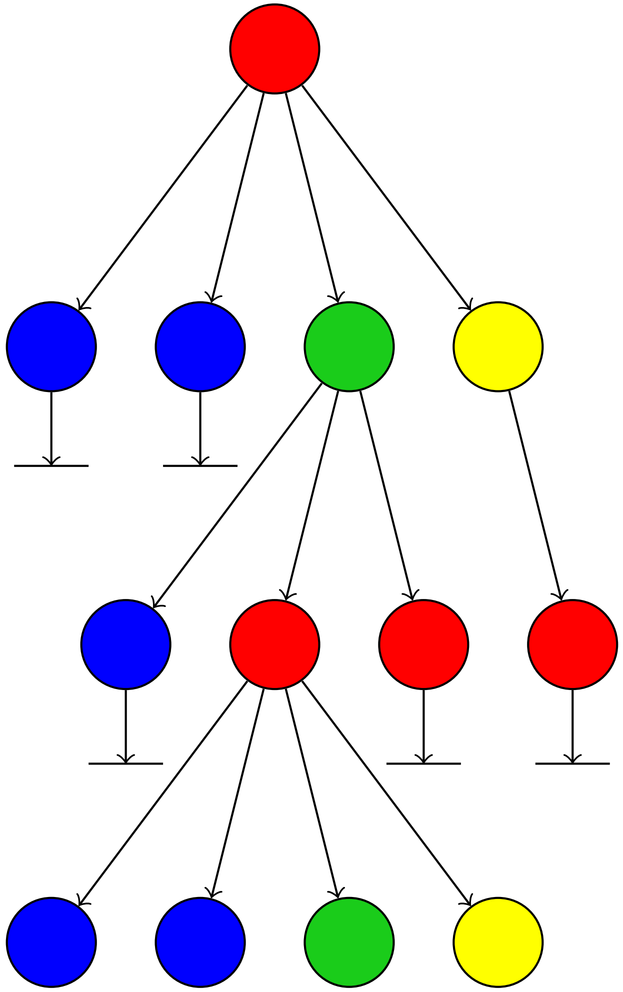
Branching Markov Decision Processes



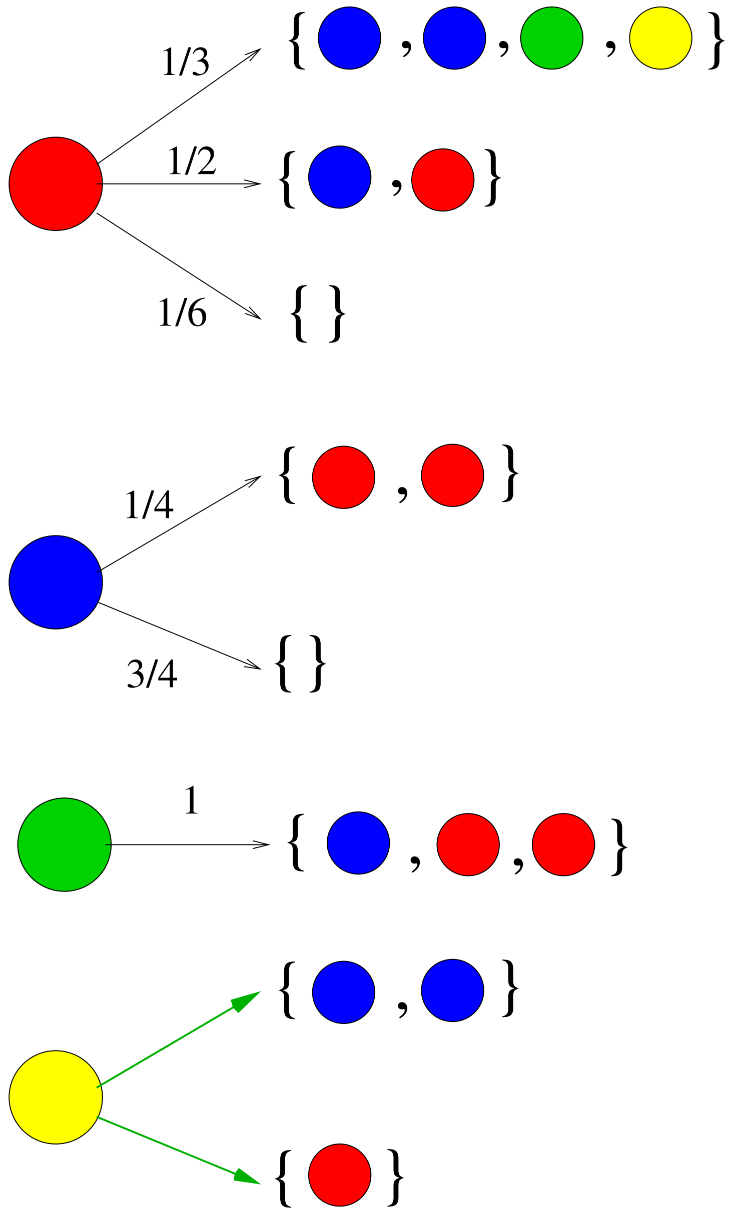


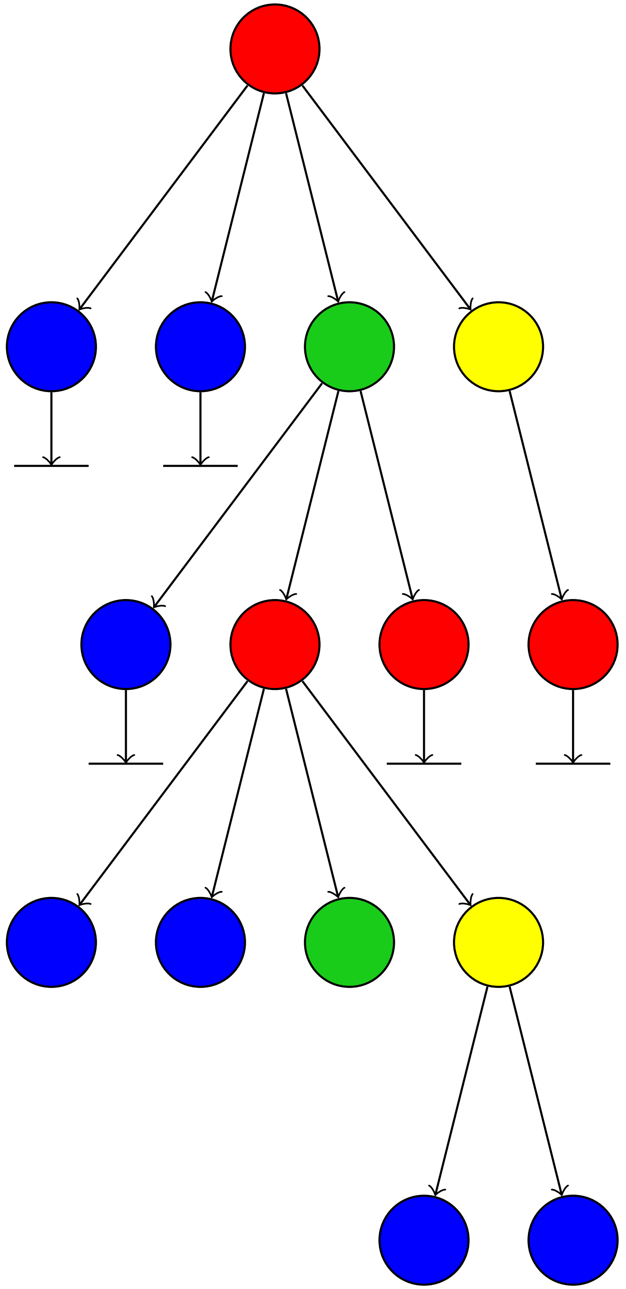
Branching Markov Decision Processes



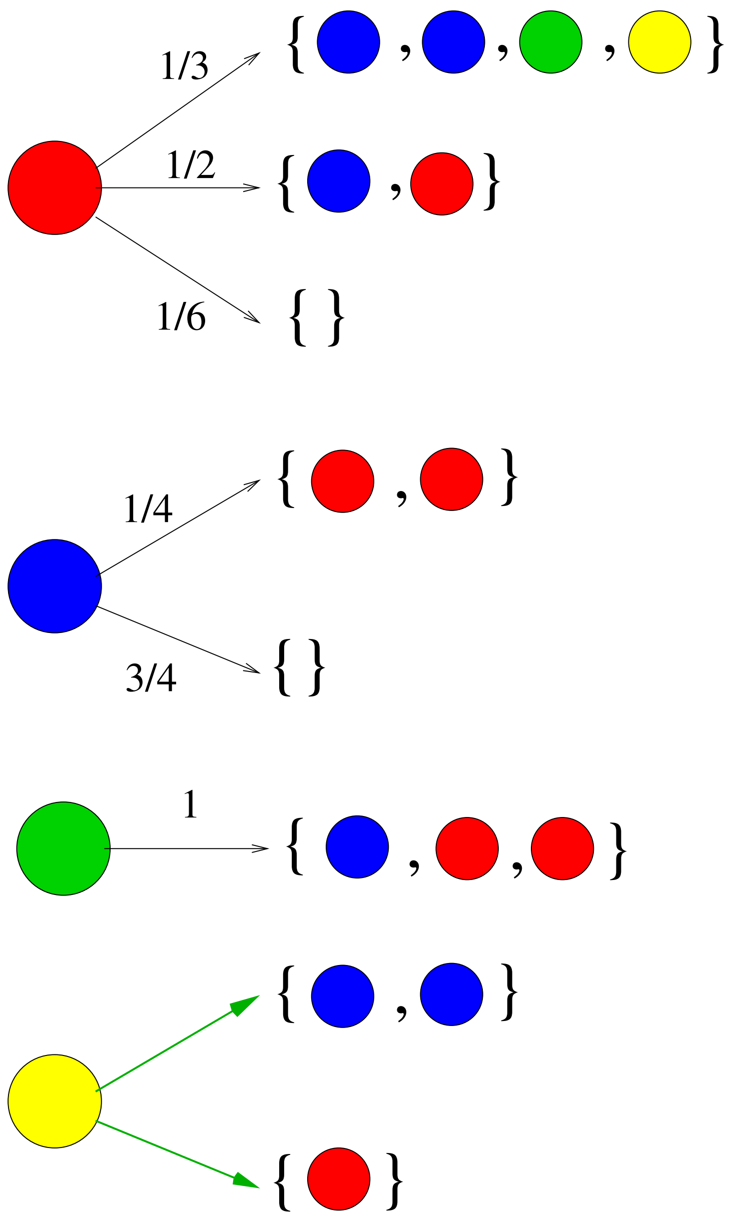


Branching Markov Decision Processes





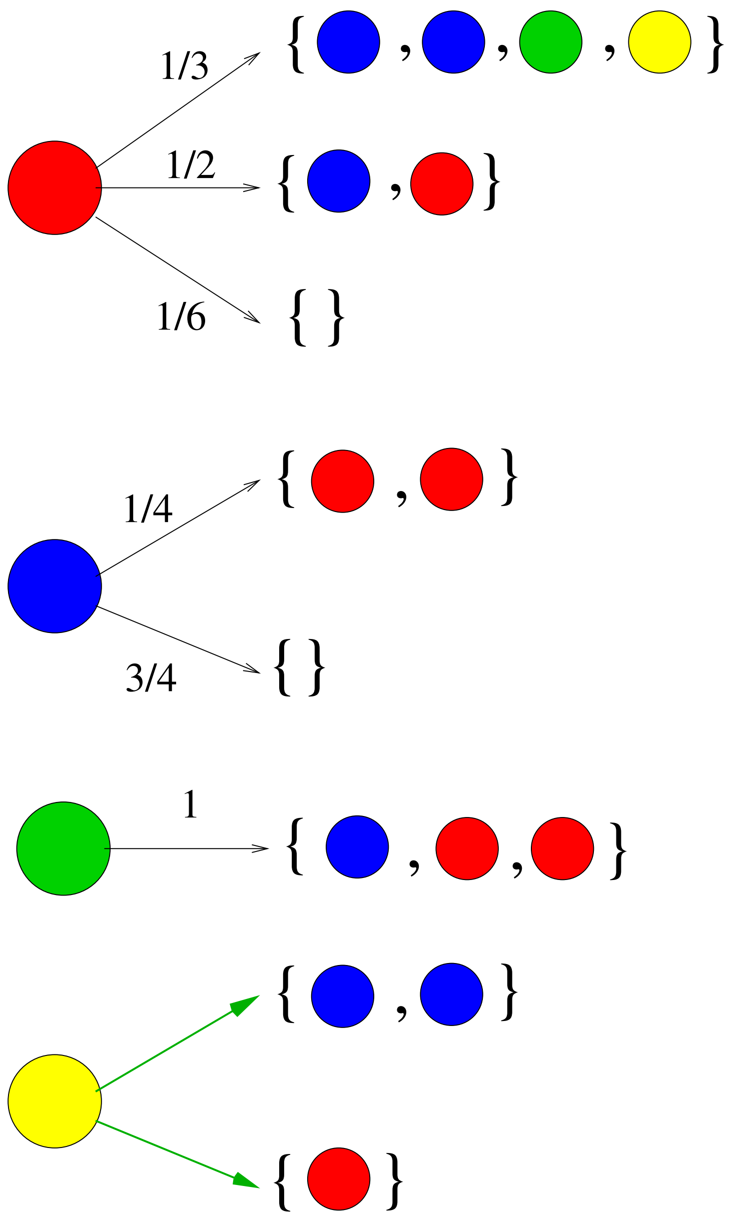
Branching Markov Decision Processes



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 🔴 ?

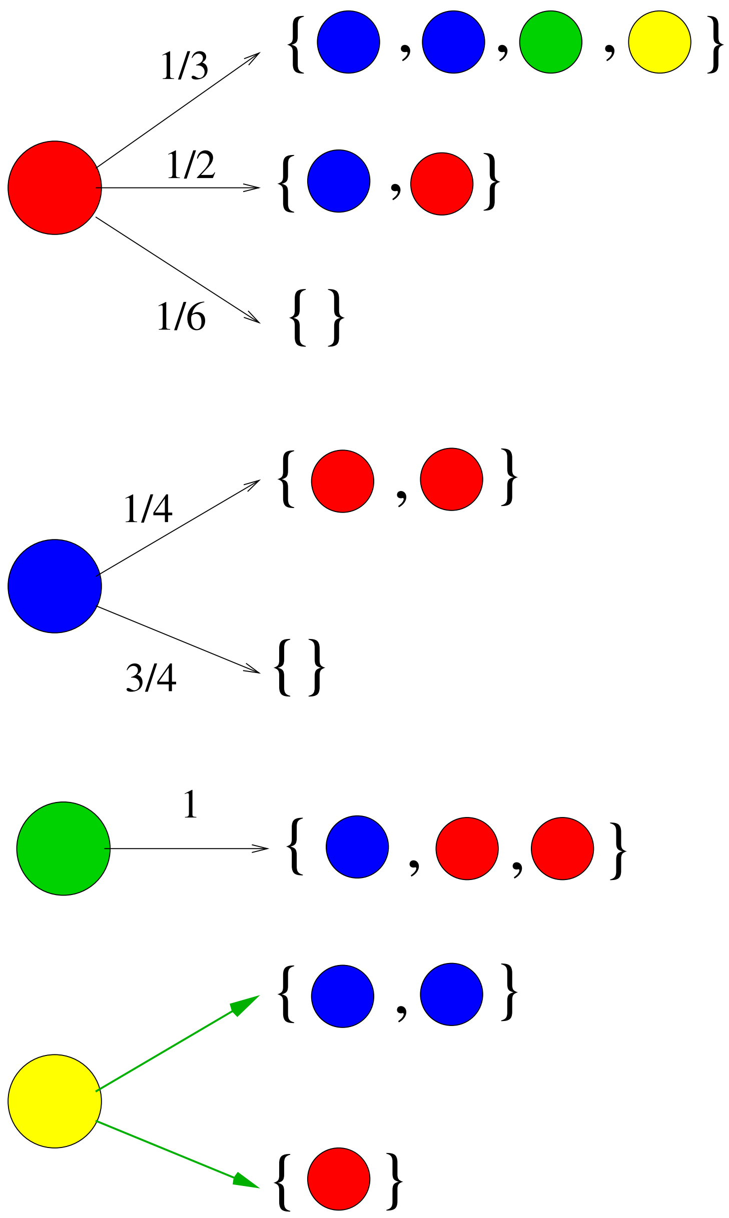
$$x_R = \frac{1}{3}x_B^2 x_G x_Y + \frac{1}{2}x_B x_R + \frac{1}{6}$$

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

$$x_G = x_B x_R^2$$

$$x_Y =$$

Branching Markov Decision Processes



Question

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

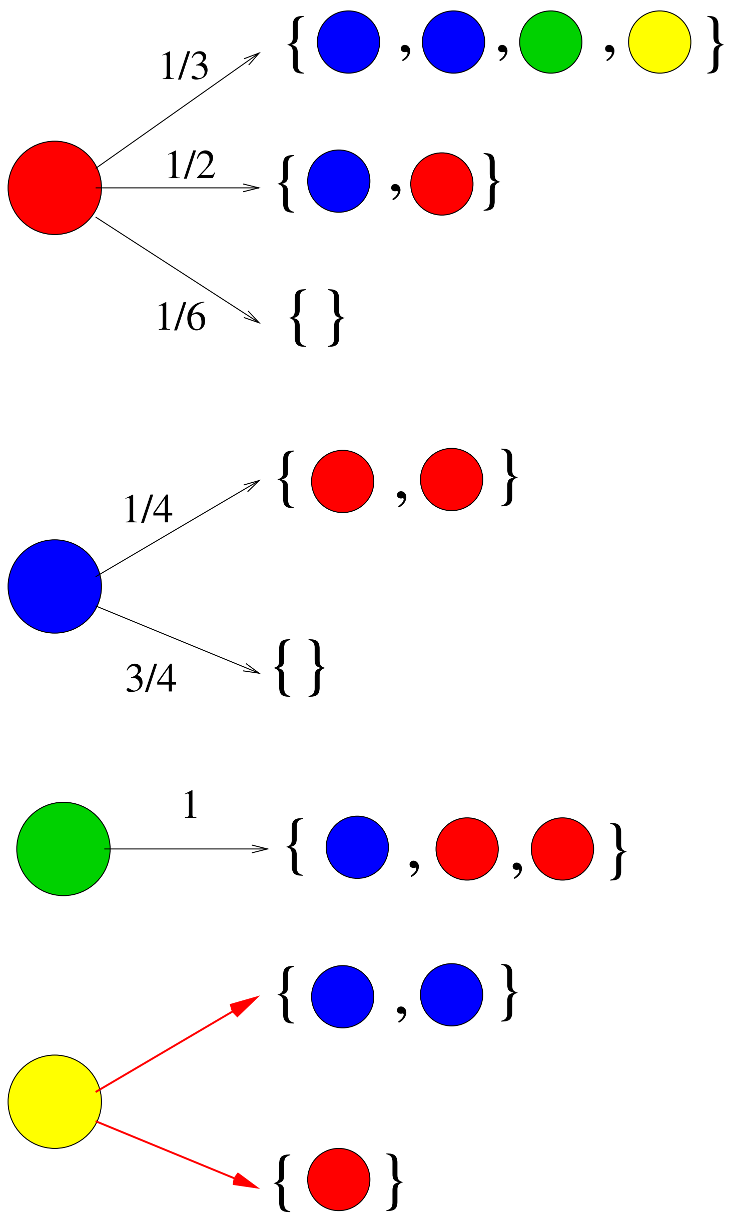
$$
\begin{aligned}
x_R &= \frac{1}{3}x_B^2 x_G x_Y + \frac{1}{2}x_B x_R + \frac{1}{6} \\
x_B &= \frac{1}{4}x_R^2 + \frac{3}{4} \\
x_G &= x_B x_R^2 \\
x_Y &= \max\{x_B^2, x_R\}
\end{aligned}
$$

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

Maximum Probabilistic Polynomial Systems of Equations

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

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

Proposition. [E.-Yannakakis'05]

- *Every max/minPPS, $x = P(x)$ has a least fixed point, $q^* \in [0,1]^n$.*
- $q^* = \lim_{k\to\infty} P^k(\mathbf{0})$.
- 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,2012])

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 new 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: iteration 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 one GNM iteration.

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

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

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

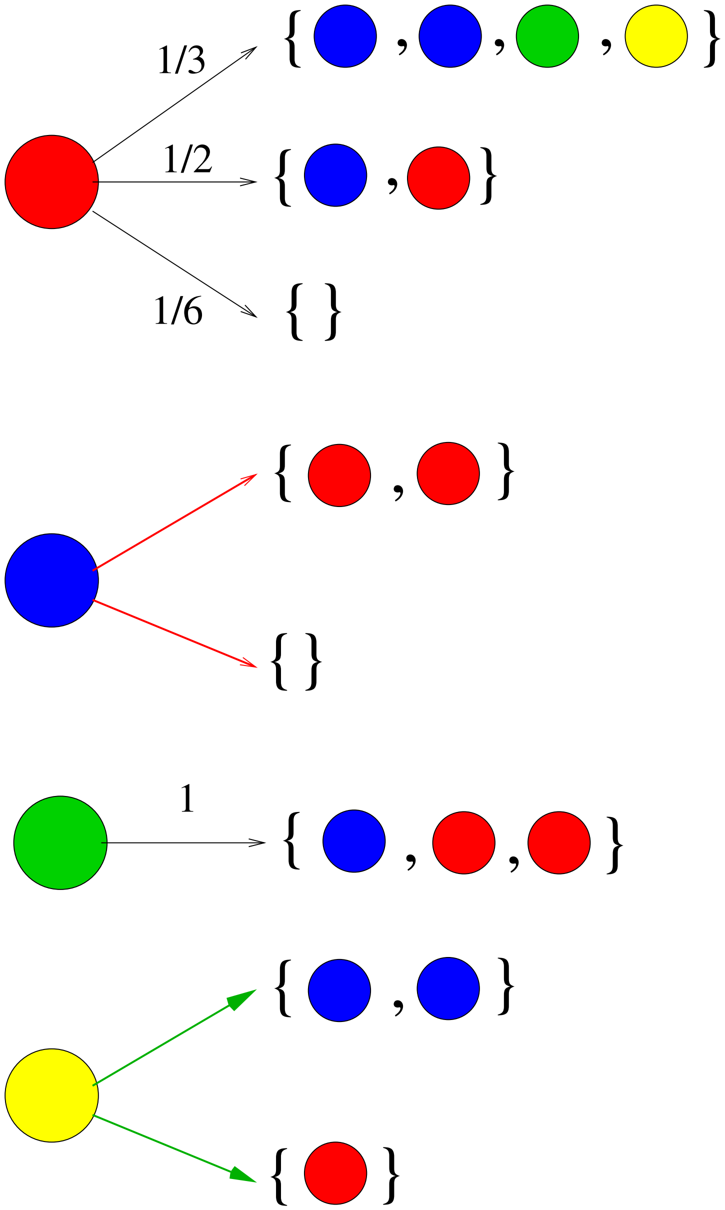
If $\mathbf{q}^* - \mathbf{x}^{(k)} \leq \lambda(\mathbf{1} - \mathbf{q}^*)$ *for some* $\lambda > 0$, *then* $\mathbf{q}^* - \mathbf{x}^{(k+1)} \leq \frac{\lambda}{2}(\mathbf{1} - \mathbf{q}^*)$.

Lemma

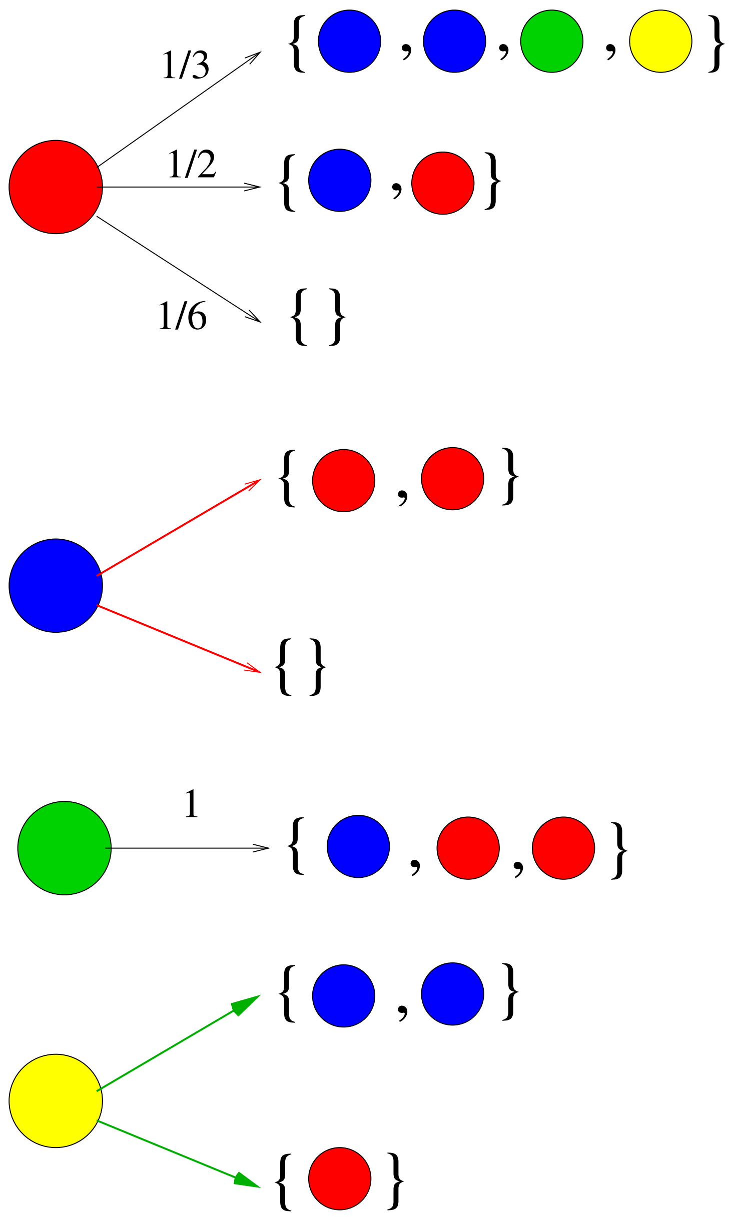
For any Max(Min) PPS with LFP $\mathbf{q}^*$, *such that* $\mathbf{0} < \mathbf{q}^* < \mathbf{1}$, *for any* i,

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

Branching Simple Stochastic Games



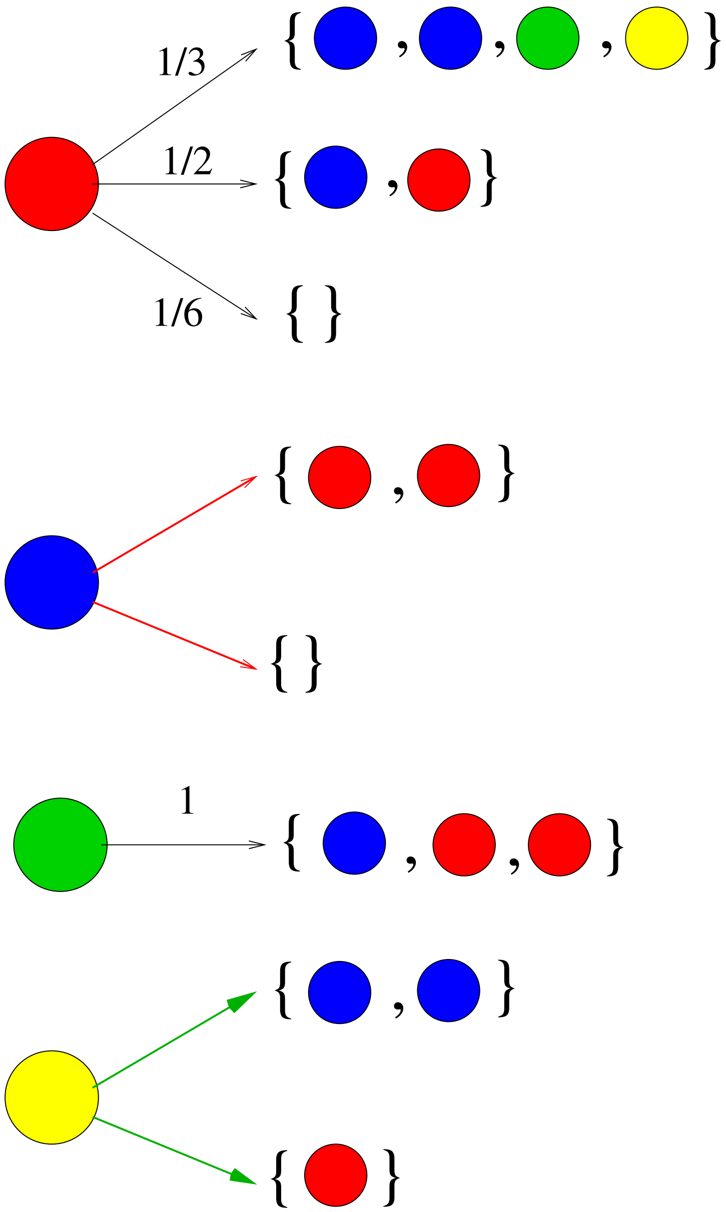
Branching Simple Stochastic Games



Types belonging to min: 🔵

Types belonging to max: 🟡

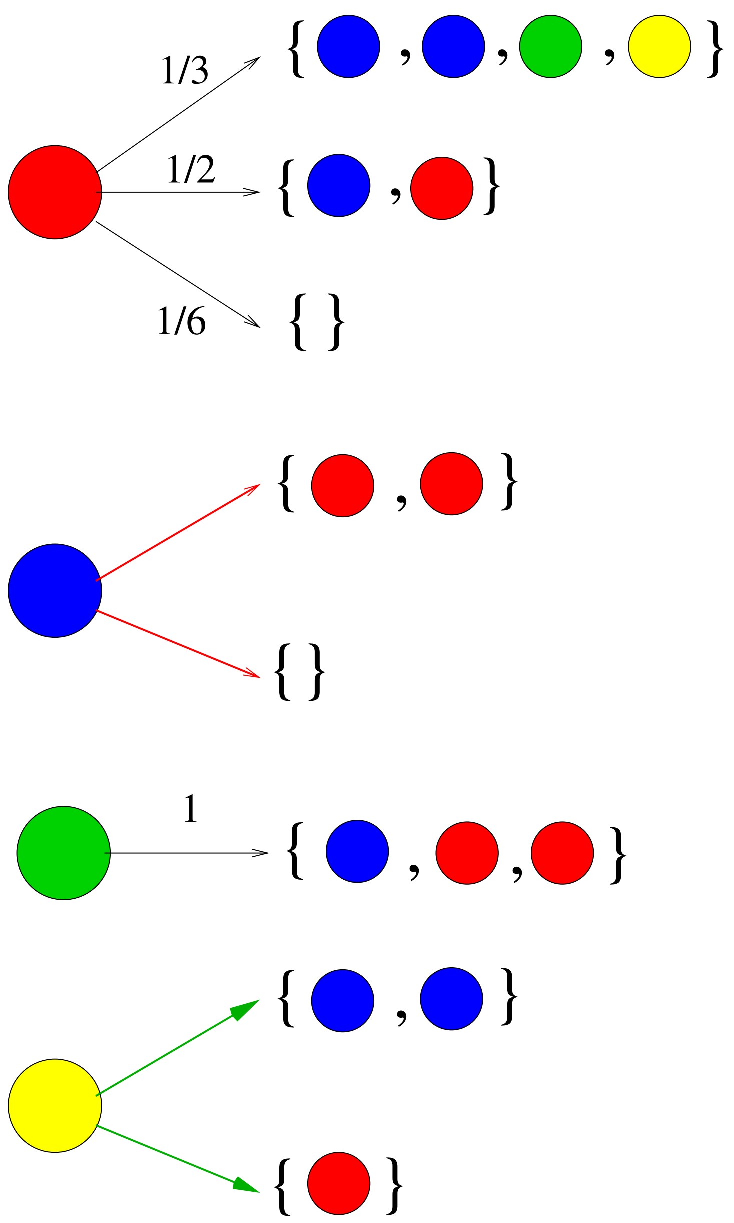
Branching Simple Stochastic Games



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** *(and in fact in* **PLS***).*

Conclusion

- We have established P-time algorithms for a number of fundamental analysis problems for Multi-type Branching Processes and Branching MDPs.
- These algorithms also yield **FNP** (and in fact **PLS**) complexity upper bounds for approximating the value of Branching Simple Stochastic Games with the same objectives.
- Can we use GNM to solve other classes of $\{+, *, \max\}$-equations??

Open problems

Question: Can we obtain better complexity bounds for **PosSLP**?

Open problems

Question: Can we obtain better complexity bounds for **PosSLP**?

Here is a very basic approach:

Given a $\{+,-,*\}$-circuit, C, guess a monotone $\{+,*\}$-circuit, C', as a "witness of positivity", and verify $C - C' = 0$ in **co-RP**.

(Checking equality to 0 is ACIT-equivalent ([ABKM'06]).)

For $a \in \mathbb{N}$, let $\tau(a)$ denote size of smallest $\{+,*,-\}$-circuit expressing a.

Let $\tau_+(a)$ denote size of smallest monotone $\{+,*\}$-circuit expressing a.

Conjecture. " τ vs. τ_+-conjecture " ("this does not work")

There exists a family of positive integers, $\langle a_n \rangle_{n \in \mathbb{N}}$, with $\tau(a_n) \in O(n)$, but such that for some fixed $c > 0$: $\tau_+(a_n) \in 2^{\Omega(n^c)}$

Remark: [Valiant'79] proved an exponential lower bound for monotone polynomials. (This does not imply lower bounds in the integer setting.)

Current state of knowledge for integers is abismal. ([Jindal-Saranurak'12]).

A better approach

Definition: call a circuit, C, quasi-monotone if it consists of some squared, $\{+,*,-\}$-subcircuits, $(C_i)^2$, $i = 1,\ldots,k$, which are inputs to a monotone $\{+,*\}$-circuit, C', whose output is the output of C. (**Note:** these circuits generalize both monotone circuits and S.O.S..)

Better approach: Given a $\{+,-,*\}$-circuit, C, guess a pair of quasi-monotone circuits C', C'' as a "witness of positivity" for C, & verify the equality $((C''+1)*C - C') = 0$ in **co-RP**.

Here is a VERY optimistic conjecture:

Conjecture: " very effective *Positivestellensatz for integers* "

This works: there is a polynomial, $p(\cdot)$, such that for any $a \in \mathbb{N}$ with $\tau(a) = n$, there exist quasi-montone circuits C'_a & C''_a, with $\texttt{size}(C'_a) \leq p(n)$ & $\texttt{size}(C''_a) \leq p(n)$, such that:

$$a = \frac{C'_a}{C''_a + 1}.$$

This would of course imply PosSLP $\in$ **MA**.