Solving fixed-point equations by derivation tree analysis

Javier Esparza

Technische Universität München

Joint work with

Stefan Kiefer and Michael Luttenberger

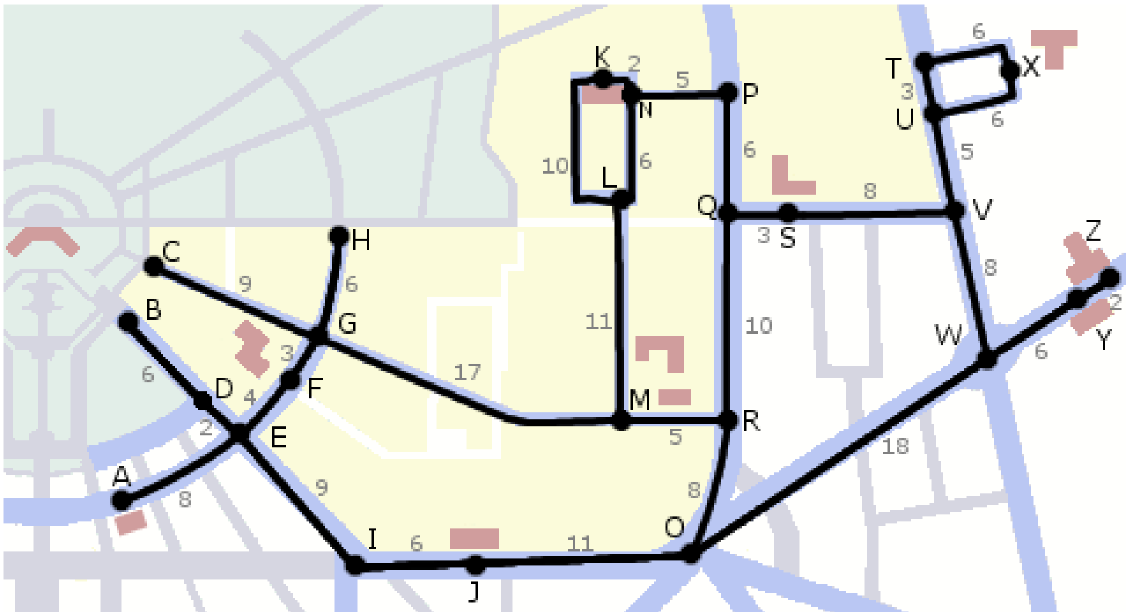
Fixed-point equations

We study systems of equations of the form

$$\begin{aligned}
X_1 &= f_1(X_1, \ldots, X_n) \\
X_2 &= f_2(X_1, \ldots, X_n) \\
&\cdots \\
X_n &= f_n(X_1, \ldots, X_n)
\end{aligned}$$

where the f_i's are "polynomial expressions".

Shortest paths



Lengths d_i of shortest paths from vertex 0 to vertex i in graph $G = (V, E)$ are the largest solution of

$$d_i = \min_{(i,j) \in E} (d_i, d_j + w_{ji})$$

where w_{ij} is the distance from i to j.

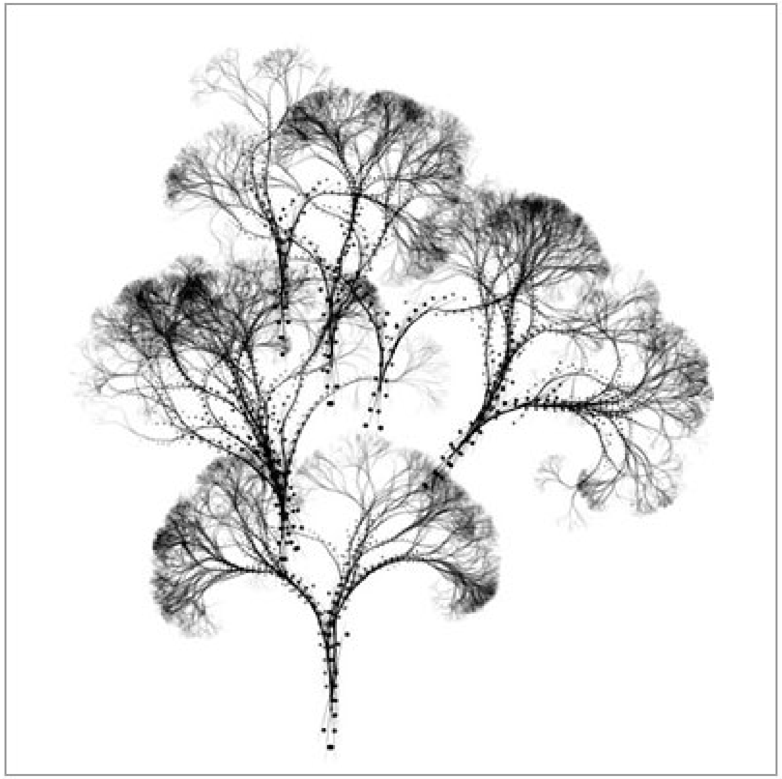
Context-free languages

Context-free grammar

$$
\begin{aligned}
X &\rightarrow ZX \mid Z \\
Y &\rightarrow aYa \mid ZX \\
Z &\rightarrow b \mid aYa
\end{aligned}
$$

Languages generated from X, Y, Z are the least solution of

$$
\begin{aligned}
L_X &= (L_Z \cdot L_X) \cup L_Z \\
L_Y &= (\{a\} \cdot L_Y \cdot \{a\}) \cup (L_Z \cdot L_X) \\
L_Z &= \{b\} \cup (\{a\} \cdot L_Y \cdot \{a\})
\end{aligned}
$$



Nuclear chain reaction

^{235}U ball of radius D, spontaneous fission.

Probability of a chain reaction is $(1 - p_0)$,

where p_α for $0 \leq \alpha \leq D$ is least solution of

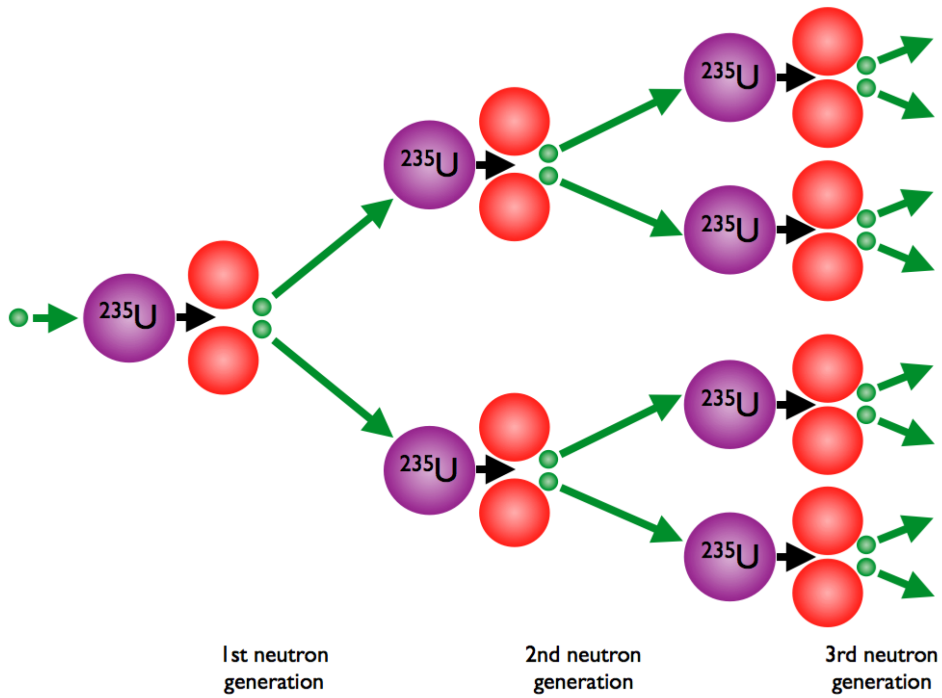
$$p_\alpha = k_\alpha + \int_0^D R_{\alpha,\beta} \, f(p_\beta) \, d\beta$$

for constants $k_\alpha, R_{\alpha,\beta}$ and polynomial $f(x)$.

Discretizing the interval $[0, D]$ we get

$$p_i = k_i + \sum_{j=1}^{n} r_{i,j} \, f(p_j)$$

for constants $k_i, r_{i,j}$.



And many others . . .

Stochastic theory:	Stationary distribution of Markov chains
	Extinction probability of branching processes
Physics:	Heat equation
	Electrostatic equilibrium
Biology:	RNA structure prediction
	Population dynamics
Computer science:	Dataflow equations (abstract interpretation)
	Reputation systems
	Provenance in databases

Underlying structure: ω-continuous semirings

Semiring $(C, +, \times, 0, 1)$:

$(C, +, 0)$ is a commutative monoid

$(C, \times, 1)$ is a monoid

$\times$ distributes over $+$

$0 \times a = a \times 0 = 0$

ω-continuity:

the relation $a \sqsubseteq b \;\Leftrightarrow\; \exists c : \; a + c = b$ is a partial order

$\sqsubseteq$-chains have limits

Examples: nonnegative integers and reals plus ∞, min-plus (tropical), languages, complete lattices, multisets, Viterbi ...

In the rest of the talk: semiring $\equiv$ ω-continuous semiring.

Research program

Develop generic solution methods valid for all semirings, or at least for large classes.

- Generic implementations.
- Exchange of algorithms and proof techniques between numerical mathematics, algebraic computation and language theory.

Research program

Develop generic solution methods valid for all semirings, or at least for large classes.

- Generic implementations.
- Exchange of algorithms and proof techniques between numerical mathematics, algebraic computation and language theory.

In this talk: brief survey of our work on derivation tree analysis.

THE generic solution method: Kleene iteration

Theorem [Klee 38, Tars 55, Kui 97]: A system f of fixed-point equations over a semiring has a least solution μf w.r.t. the natural order $\sqsubseteq$.

This least solution is the supremum of the Kleene approximants, denoted by $\{k_i\}_{i\geq 0}$, and given by

$$\begin{aligned} k_0 &= f(0) \\ k_{i+1} &= f(k_i)\ . \end{aligned}$$

Basic algorithm for calculation of μf : compute $k_0, k_1, k_2, \ldots$ until either $k_i = k_{i+1}$ or the approximation is considered adequate.

Kleene iteration may be slow

Set interpretations: Kleene iteration never terminates if μf is an infinite set.

- $X = \{a\} \cdot X \cup \{b\} \qquad \mu f = a^* b$

  Kleene approximants are finite sets: $k_i = (\epsilon + a + \ldots + a^i)b$

Real semiring: convergence can be very slow.

- $X = 0.5\, X^2 + 0.5 \qquad \mu f = 1 = 0.99999\ldots$

  "Logarithmic convergence": k iterations give $O(\log k)$ correct digits.

$$k_n \leq 1 - \frac{1}{n+1} \qquad k_{2000} = 0.9990$$

Language-theoretic characterization of μf

An equation $X = f(X)$ over a semiring induces a context-free grammar G and a valuation V

Language-theoretic characterization of μf

An equation $X = f(X)$ over a semiring induces a context-free grammar G and a valuation V

Example: $X = 0.25X^2 + 0.25X + 0.5$

Grammar: $X \rightarrow aXX \mid bX \mid c$

Valuation: $V(a) = 0.25, V(b) = 0.25, V(c) = 0.5$

Language-theoretic characterization of μf

An equation $X = f(X)$ over a semiring induces a context-free grammar G and a valuation V

Example: $X = 0.25X^2 + 0.25X + 0.5$

Grammar: $X \rightarrow a\,X\,X \mid b\,X \mid c$

Valuation: $V(a) = 0.25, V(b) = 0.25, V(c) = 0.5$

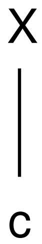
V extends to derivation trees and sets of derivation trees:

$$V(t) \quad := \quad \text{ordered product of the leaves of } t$$

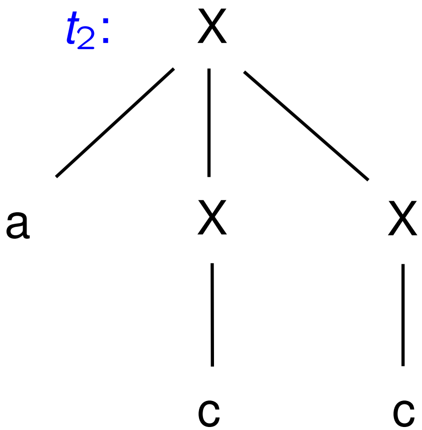
$$V(T) \quad := \quad \sum_{t \in T} V(t)$$

$X \rightarrow aXX \mid bX \mid c$ $\qquad V(a) = V(b) = 0.25, V(c) = 0.5$

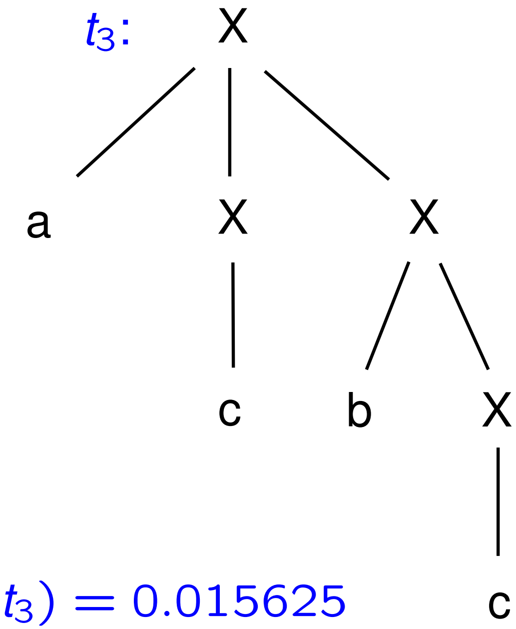
t_1:



t_2:



t_3:



$V(t_1) = 0.5$ $\qquad V(t_2) = 0.25 \cdot 0.5 \cdot 0.5 = 0.0625$ $\qquad V(t_3) = 0.015625$

$$V(\{t_1, t_2, t_3\}) = 0.5 + 0.0625 + 0.015625 = 0.578125$$

Language-theoretic characterization of μf

Fundamental Theorem [Boz99,EKL10]: Let G be the grammar for $X = f(X)$, and let $T(G)$ be the set of derivation trees of G . Then $\mu f = V(T(G)) \stackrel{def}{=} V(G)$

$$
\begin{tikzcd}
X = f(X) \arrow[rr] \arrow[d] & & G \arrow[d] \\
\begin{array}{c} \mu f \\ = \\ V(T(G)) \end{array} & & T(G) \arrow[ll, "V"']
\end{tikzcd}
$$

Derivation tree analysis

Use language-theoretic results about the

set of derivation trees of the associated context-free grammar

to derive approximation or solution algorithms for the

system of equations.

Approximating grammars

Let G be the grammar for $X = f(X)$.

An unfolding of G is a sequence $U^1, U^2, U^3, \ldots$ of grammars such that

- $T(U^i) \cap T(U^j)$ for every $i \neq j$, and
- there is a bijection between $\bigcup_{i=1}^{\infty} T(U^i)$ and $T(G)$ that preserves the yield.

From $U^1, U^2, U^3, \ldots$ we get another sequence $G^1, G^2, G^3, \ldots$ such that $T(G^j) = \bigcup_{i=1}^{j} T(U^i)$

Approximating grammars

Let Op be the operator on the semiring such that

- $V(G^1) = Op(0)$ and
- $V(G^{i+1}) = Op(V(G^i))$ for every $i \geq 1$

By the fundamental theorem we get $\mu f = \sup_{i=1}^{\infty} Op^i(0)$

Op yields a procedure to approximate μf.

Approximating grammars by height

Goal: Yield-preserving bijection between $T(U^i)$ $(T(G^i))$ and the derivation trees of G of height i (at most i).

$$G: \; X \; \rightarrow \; a X X \; | \; b X \; | \; c \; .$$

.

Approximating grammars by height

Goal: Yield-preserving bijection between $T(U^i)$ $(T(G^i))$ and the derivation trees of G of height i (at most i).

$$G: \; X \;\longrightarrow\; a\,X\,X \;\mid\; b\,X \;\mid\; c \;.$$

$$X^{\langle 1 \rangle} \;\longrightarrow\; c$$

.

Approximating grammars by height

Goal: Yield-preserving bijection between $T(U^i)$ ($T(G^i)$) and the derivation trees of G of height i (at most i).

$$G: \; X \;\longrightarrow\; a\,X\,X \;\mid\; b\,X \;\mid\; c \;.$$

$$X^{\langle 1 \rangle} \;\longrightarrow\; c$$

$$X^{[1]} \;\longrightarrow\; X^{\langle 1 \rangle}$$

.

Approximating grammars by height

Goal: Yield-preserving bijection between $T(U^i)$ $(T(G^i))$ and the derivation trees of G of height i (at most i).

$$G: \; X \rightarrow a\,X\,X \mid b\,X \mid c \; .$$

$$X^{\langle 1 \rangle} \longrightarrow c$$

$$X^{[1]} \longrightarrow X^{\langle 1 \rangle}$$

$$X^{\langle k \rangle} \longrightarrow aX^{\langle k-1 \rangle}X^{\langle k-1 \rangle} \mid aX^{[k-2]}X^{\langle k-1 \rangle} \mid aX^{\langle k-1 \rangle}X^{[k-2]} \mid bX^{\langle k-1 \rangle}$$

.

Approximating grammars by height

Goal: Yield-preserving bijection between $T(U^i)$ $(T(G^i))$ and the derivation trees of G of height i (at most i).

$$G: \; X \rightarrow a X X \mid b X \mid c \; .$$

$$X^{\langle 1 \rangle} \rightarrow c$$

$$X^{[1]} \rightarrow X^{\langle 1 \rangle}$$

$$X^{\langle k \rangle} \rightarrow aX^{\langle k-1 \rangle}X^{\langle k-1 \rangle} \mid aX^{[k-2]}X^{\langle k-1 \rangle} \mid aX^{\langle k-1 \rangle}X^{[k-2]} \mid bX^{\langle k-1 \rangle}$$

$$X^{[k]} \rightarrow X^{\langle k \rangle} \mid X^{[k-1]}$$

.

Approximating grammars by height

Goal: Yield-preserving bijection between $T(U^i)$ $(T(G^i))$ and the derivation trees of G of height i (at most i).

$$G: \ X \ \rightarrow \ a\,X\,X \ | \ b\,X \ | \ c \ .$$

$$X^{\langle 1 \rangle} \ \longrightarrow \ c$$

$$X^{[1]} \ \longrightarrow \ X^{\langle 1 \rangle}$$

$$X^{\langle k \rangle} \ \longrightarrow \ aX^{\langle k-1 \rangle}X^{\langle k-1 \rangle} \ | \ aX^{[k-2]}X^{\langle k-1 \rangle} \ | \ aX^{\langle k-1 \rangle}X^{[k-2]} | \ bX^{\langle k-1 \rangle}$$

$$X^{[k]} \ \longrightarrow \ X^{\langle k \rangle} \ | \ X^{[k-1]}$$

U^i (G^i) is the grammar with $X^{\langle i \rangle}$ $(X^{[i]})$ as axiom.

Approximating grammars by height

$$
\begin{aligned}
X^{\langle k\rangle} &\rightarrow aX^{\langle k-1\rangle}X^{\langle k-1\rangle} \mid aX^{[k-2]}X^{\langle k-1\rangle} \mid aX^{\langle k-1\rangle}X^{[k-2]} \mid bX^{\langle k-1\rangle} \\
X^{[k]} &\rightarrow X^{\langle k\rangle} \mid X^{[k-1]}
\end{aligned}
$$

"Taking values" we get:

$$
\begin{aligned}
V(U^k) &= V(a)\cdot V(U^{k-1})^2 + V(a)\cdot V(G^{k-2})\cdot V(U^{k-1}) \\
&+ V(a)\cdot V(U^{k-1})\cdot V(G^{k-2}) + V(b)\cdot V(U^{k-1}) \\
V(G^k) &= V(G^{k-1}) + V(U^k)
\end{aligned}
$$

and since $f(X) = V(a)\cdot X^2 + V(b)\cdot X + V(c)$

$$
\begin{aligned}
V(G^1) &= f(0) \\
V(G^{i+1}) &= f(V(G^i)) \quad \text{for every } i \geq 1
\end{aligned}
$$

$\boxed{\text{Kleene approximation corresponds to evaluating the derivation trees of } G \text{ by increasing height.}}$

A "faster" approximation

$$G: \; X \; \rightarrow \; a\,X\,X \;\mid\; b\,X \;\mid\; c \;\; .$$

Recall the approximation by height

$$X^{\langle k\rangle} \;\rightarrow\; aX^{\langle k-1\rangle}X^{\langle k-1\rangle} \mid aX^{[k-2]}X^{\langle k-1\rangle} \mid aX^{\langle k-1\rangle}X^{[k-2]} \mid bX^{\langle k-1\rangle}$$

To capture more trees we allow linear recursion.

$$X^{\langle k\rangle} \;\rightarrow\; aX^{\langle k-1\rangle}X^{\langle k-1\rangle} \mid aX^{[k-1]}X^{\langle k\rangle} \mid aX^{\langle k\rangle}X^{[k-1]} \mid bX^{\langle k-1\rangle}$$

U^i (G^i) defined as before.

Taking values

$$X^{\langle k \rangle} \;\to\; aX^{\langle k-1 \rangle}X^{\langle k-1 \rangle} \mid aX^{[k-1]}X^{\langle k \rangle} \mid aX^{\langle k \rangle}X^{[k-1]} \mid bX^{\langle k-1 \rangle}$$

$V(U^i)$ is the least solution of the linear equation

$$\begin{aligned} X \;=\; & \quad V(a) \cdot V(U^{i-1})^2 + V(a) \cdot V(G^{i-1}) \cdot X \\ & + \; V(a) \cdot X \cdot V(G^{i-1}) + V(b) \cdot X \end{aligned}$$

Iterative approximation of $V(G)$:

- $V(G^1) =$ least solution of $X = V(b) \cdot X + V(c)$
- $V(G^{i+1}) = V(G^i) + V(U^{i+1})$ for every $i \geq 1$

Recipe to approximate μf by solving linear equations.

Interpreting the new approximation

Consider equations $X = f(X)$ on the real semiring

Let $g(X) = f(X) - X$. Then μf is a zero of $g(X)$.

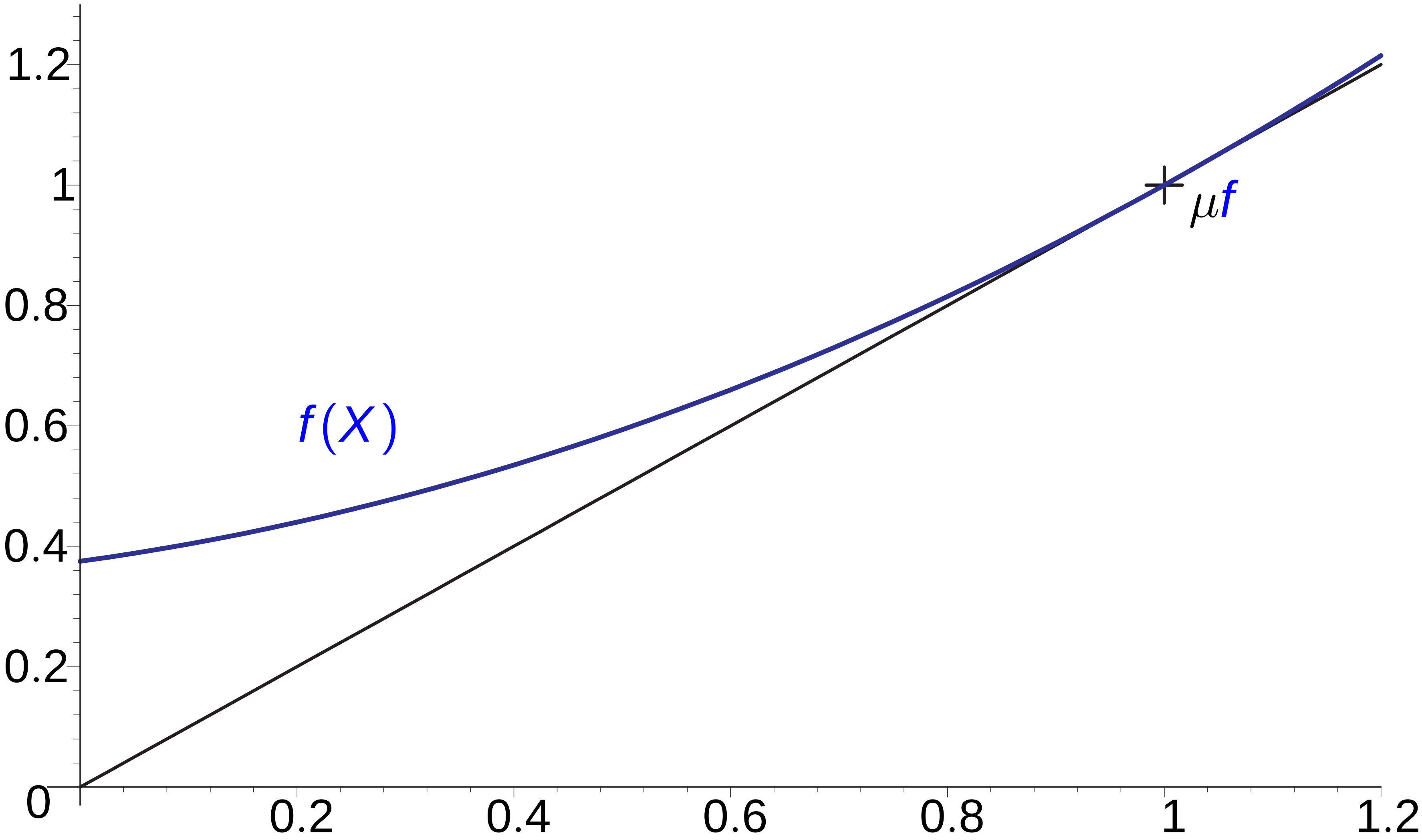
Simple arithmetic yields

$$V(G^{i+1}) = V(G^i) - \frac{g(V(G^i))}{g'(V(G^i))}$$

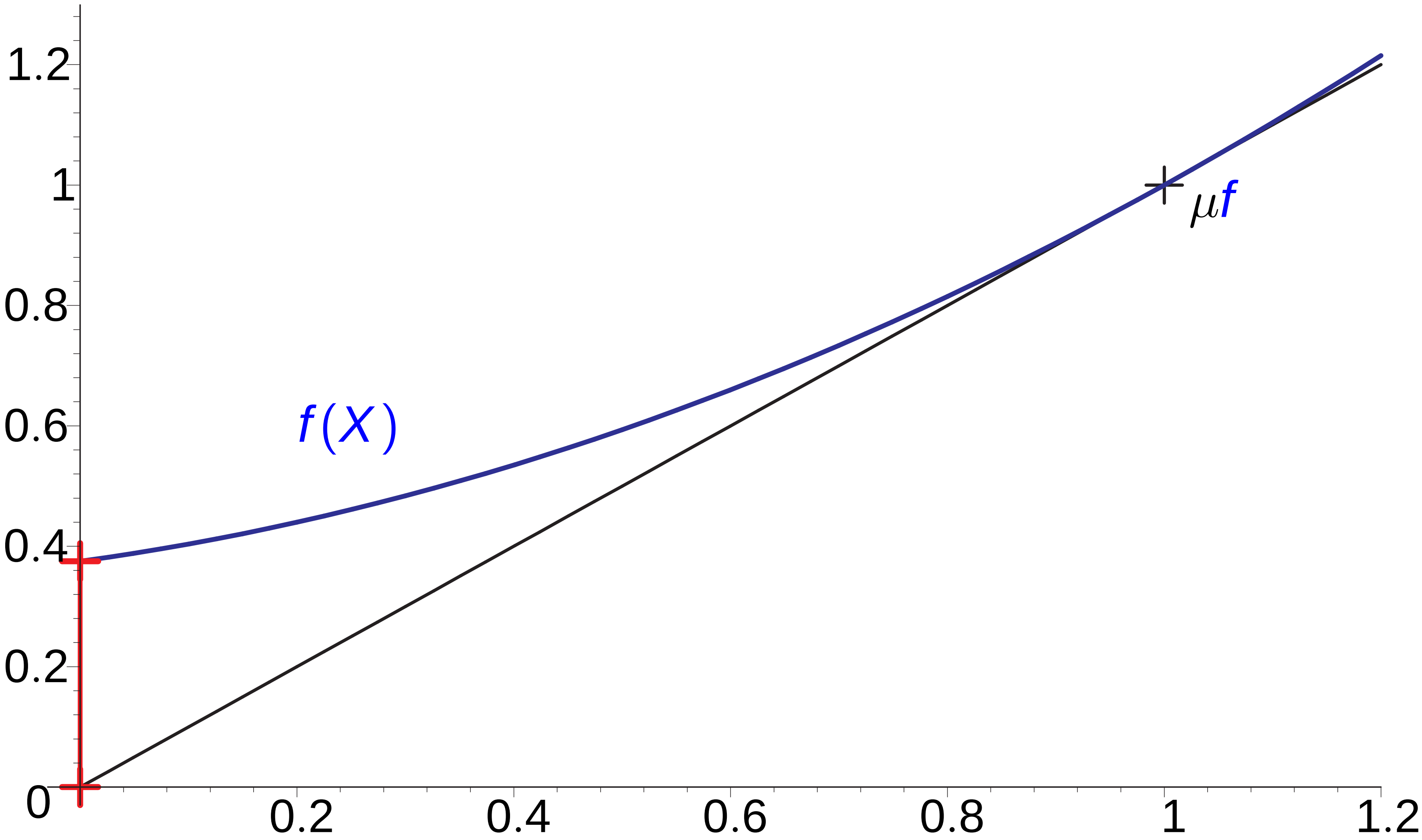
where $g'(X)$ is the derivative of g.

This is Newton's method for approximating a zero of a differentiable function.

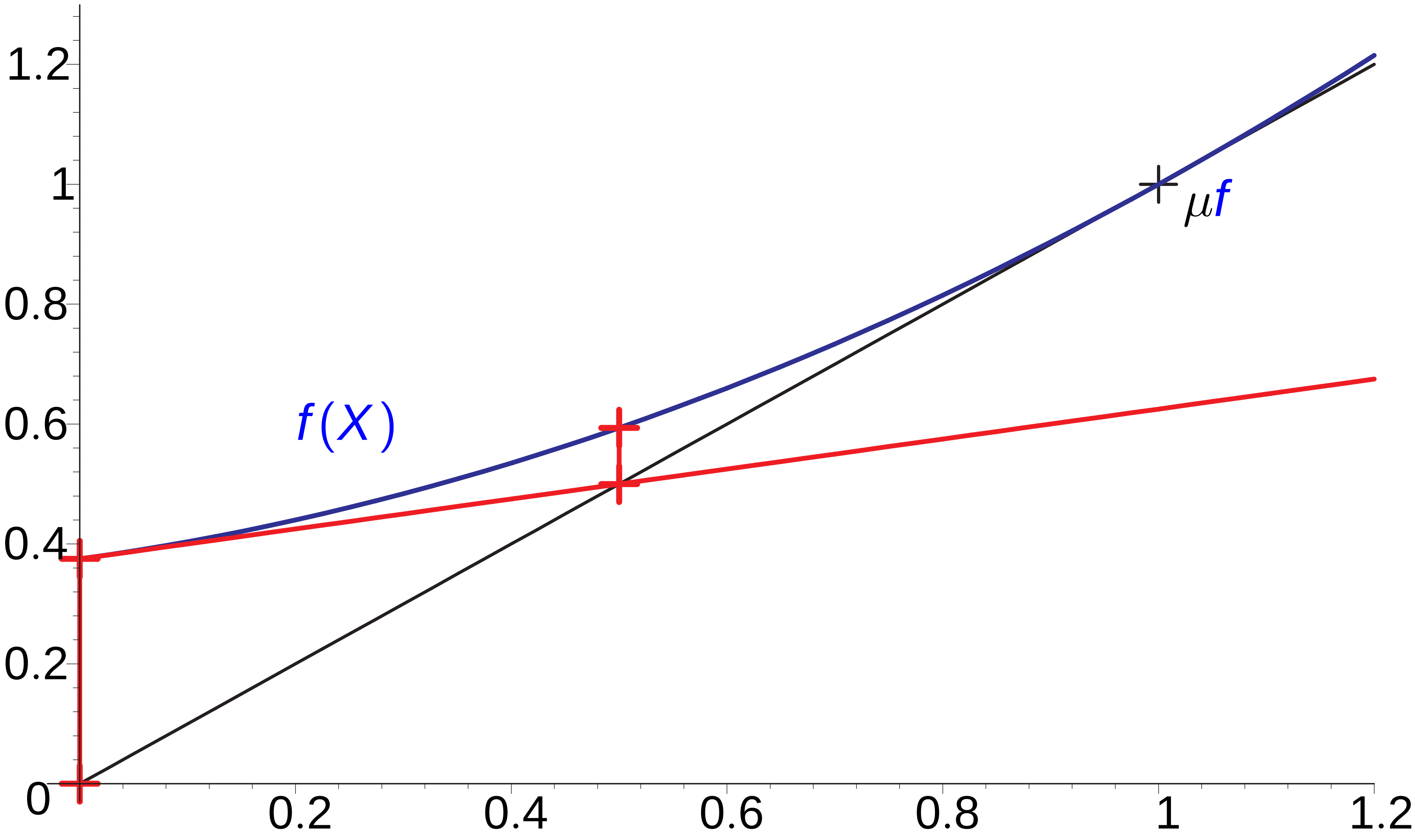
Newton's method for $X = f(X)$ (univariate case)



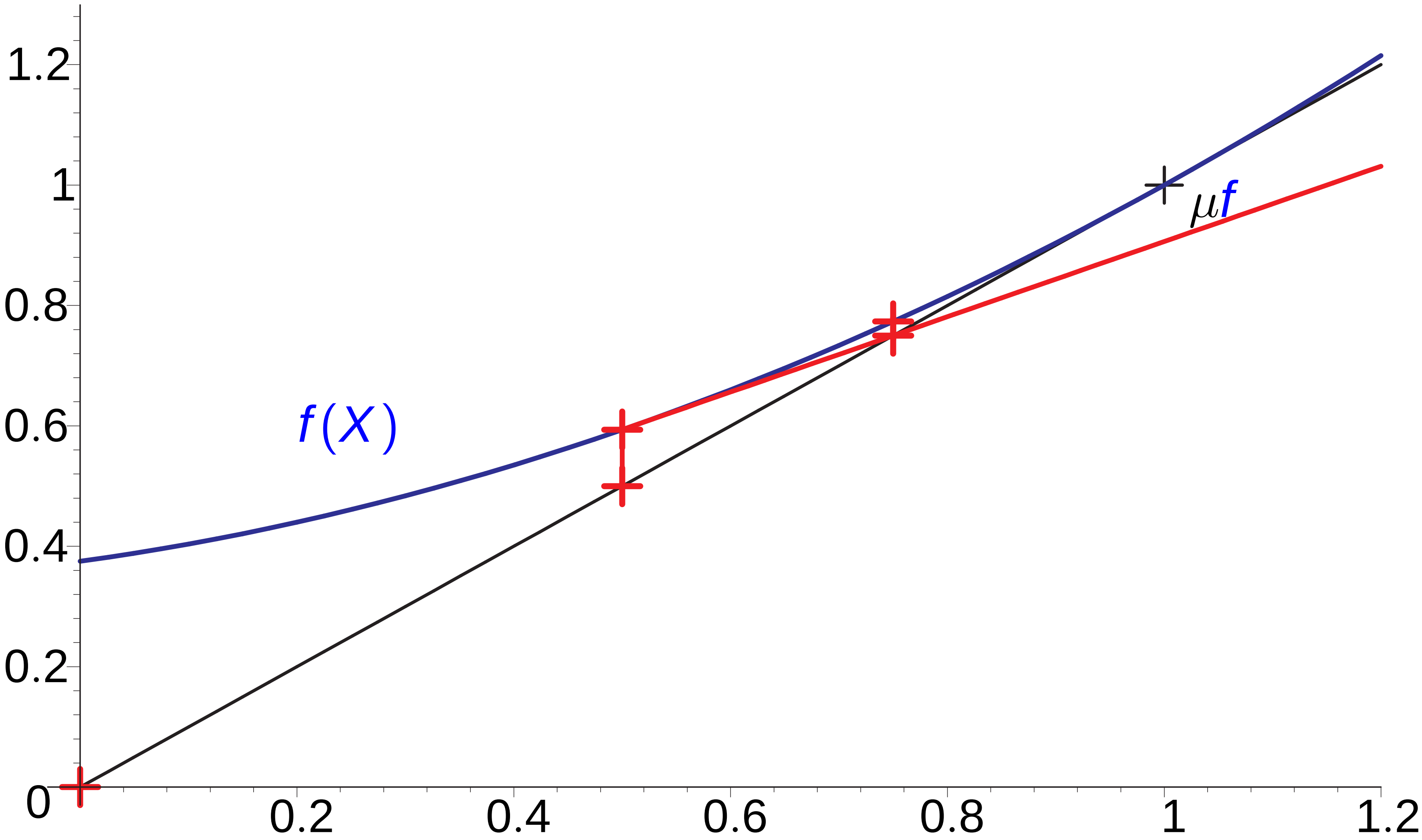
Newton's method for $X = f(X)$ (univariate case)



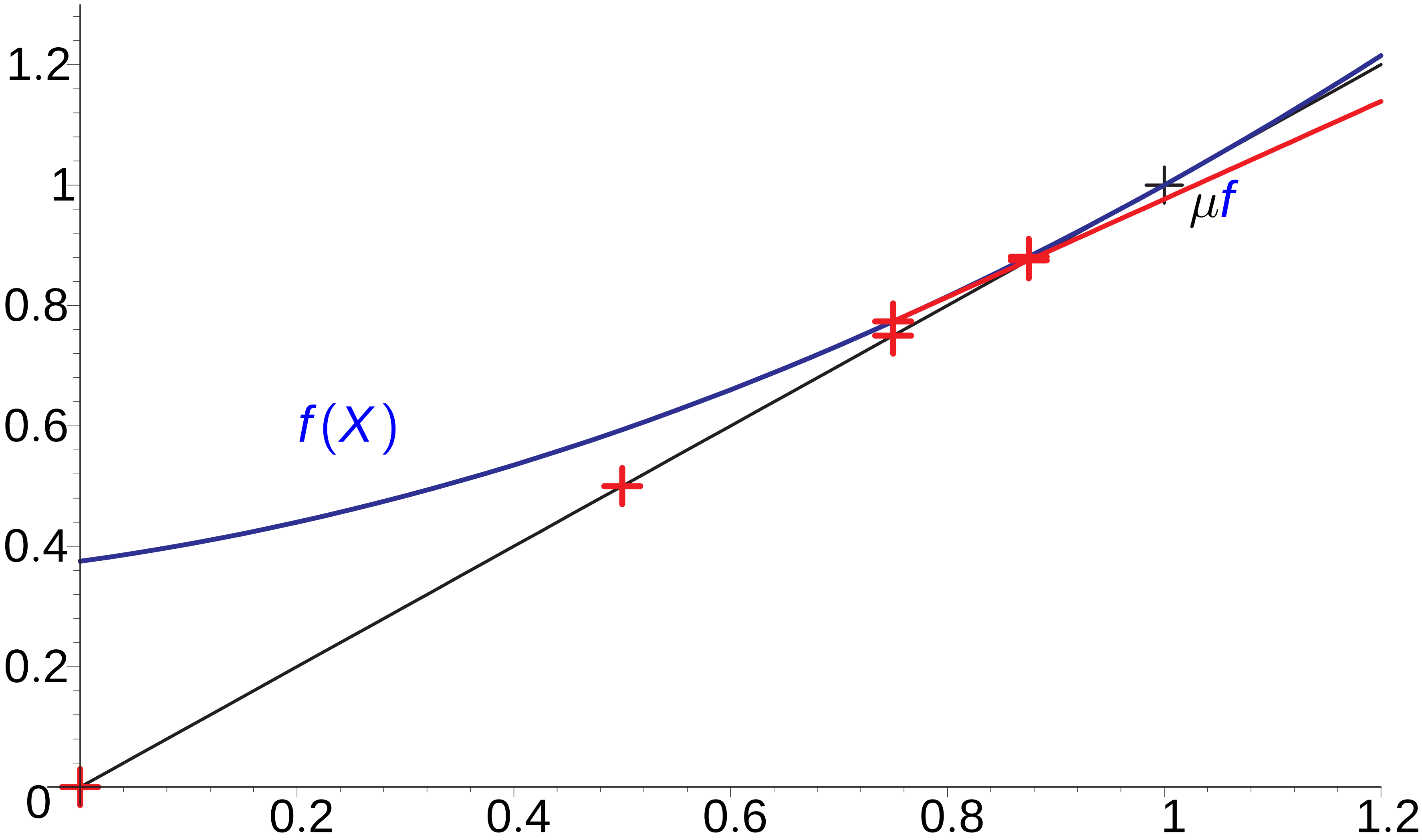
Newton's method for $X = f(X)$ (univariate case)



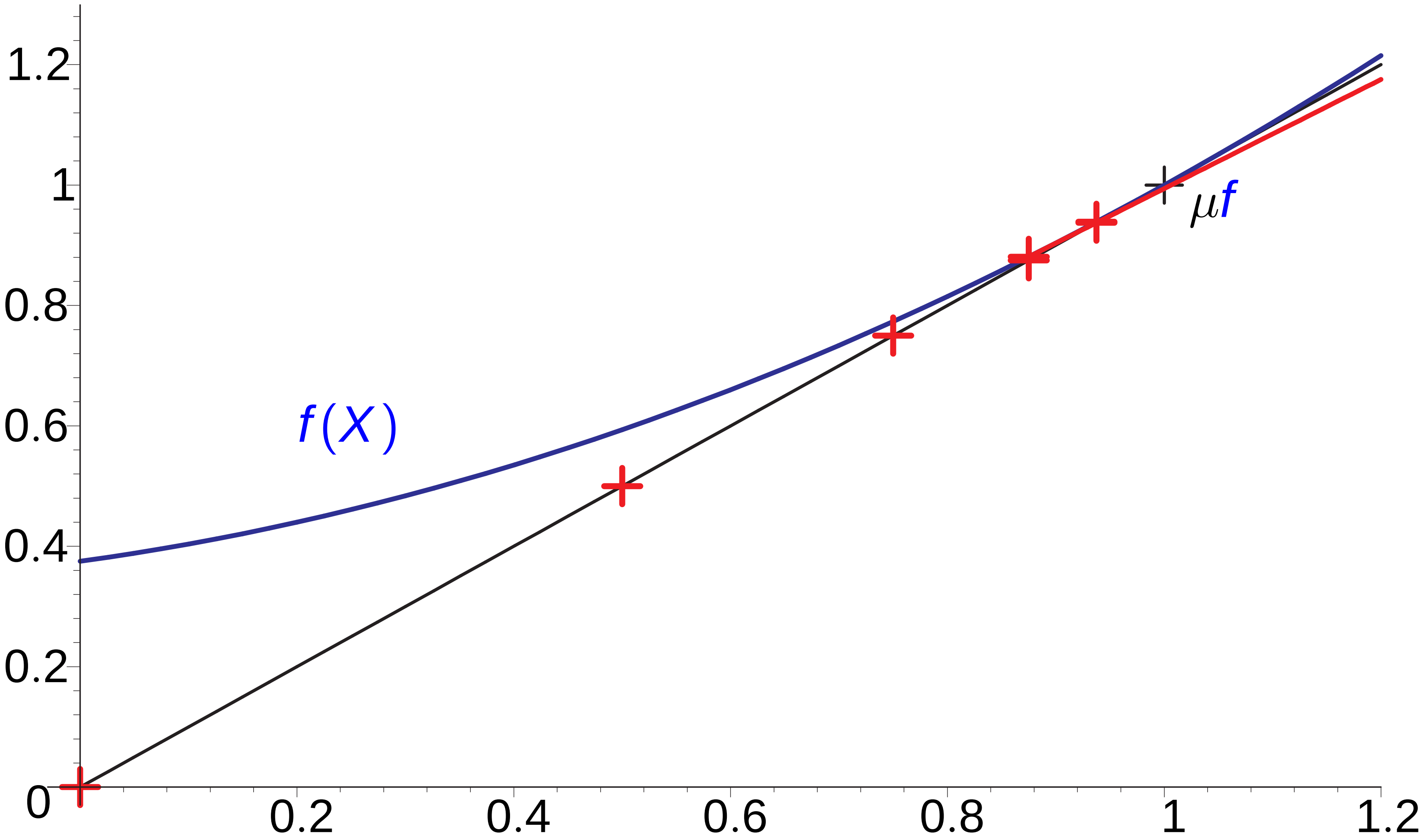
Newton's method for $X = f(X)$ (univariate case)



Newton's method for $X = f(X)$ (univariate case)



Newton's method for $X = f(X)$ (univariate case)



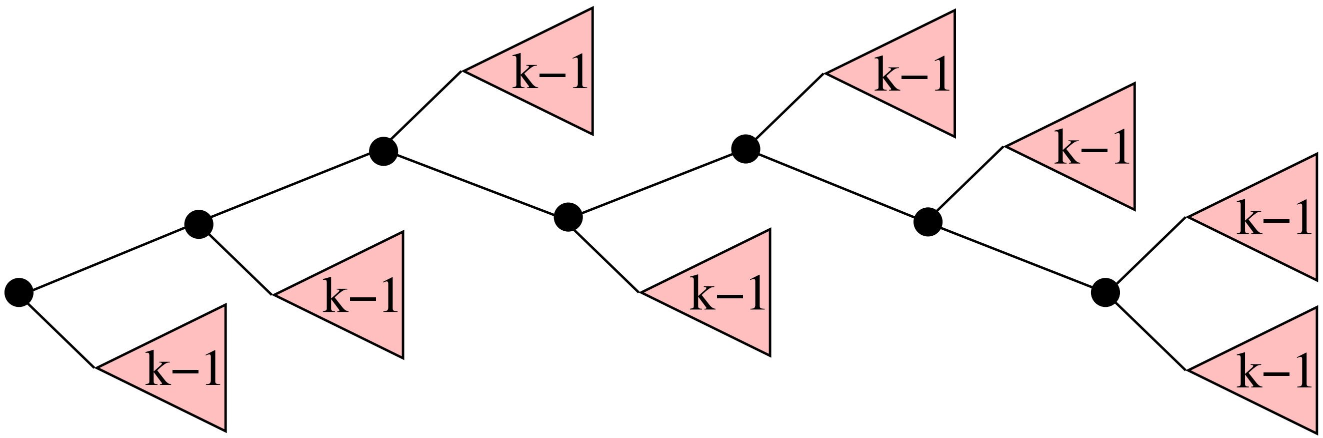
Language theoretic view of Newton's method

$$X^{\langle k \rangle} \;\rightarrow\; aX^{\langle k-1 \rangle}X^{\langle k-1 \rangle} \mid aX^{[k-1]}X^{\langle k \rangle} \mid aX^{\langle k \rangle}X^{[k-1]} \mid bX^{\langle k-1 \rangle}$$

Say a tree of G has dimension k if it is derived from U^k

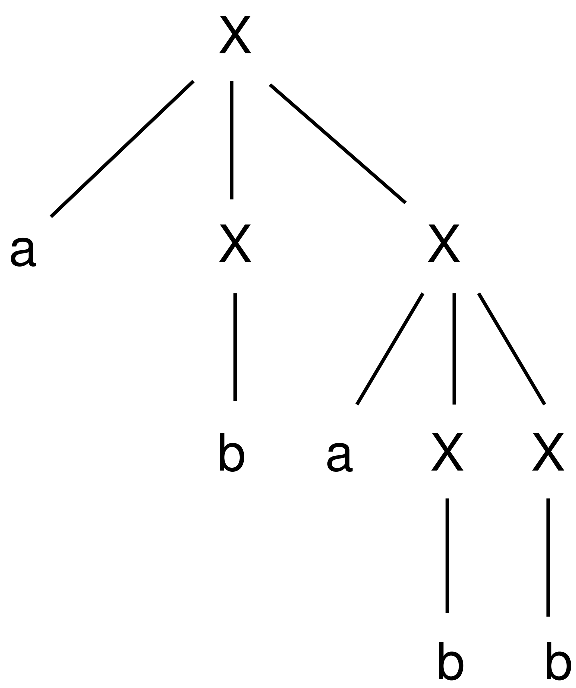
A derivation tree has dimension 0 if it has one node.

A derivation tree has dimension $k > 0$ if it consists of a spine with subtrees of dimension at most $k-1$ (and at least one subtree of dimension $k-1$).



Understanding dimension

The dimension of a derivation tree is the height of the largest full binary tree embeddable in it (ignoring terminals).



Newton approximation corresponds to evaluating the derivation trees of *G* by increasing dimension.

Convergence speed of Newton's method

At least as good as Kleene's approximation

For every value v let $\alpha^i(v)$ be the number of trees of G^i with that value, if the number is finite, and $\alpha^i(v) = \infty$ otherwise.

$$V(G^i) = \sum_v \sum_{i=1}^{\alpha^i(v)} v \qquad V(G) = \sum_v \sum_{i=1}^{\alpha(v)} v$$

Intuitively: $\sum_{i=1}^{\alpha(v)} v$ is the "contribution" of v to $V(G)$.
$\sum_{i=1}^{\alpha^i(v)} v$ is the "contribution" of v to $V(G^i$.

We analyze how fast $\alpha^i(v)$ converges to $\alpha(v)$.

Convergence speed for commutative semirings

Theorem (Luttenberger, unpublished): Given a system of n equations over a commutative semiring,

$$\alpha^{k \cdot n+1}(v) \geq \min\{\alpha(v), k\}$$

for every semiring value v and every $k \geq 1$.

In words: $k \cdot n + 1$ Newton steps ”capture” at least k trees of each value v (if there are that many).

Convergence speed for commutative and idempotent semirings

In idempotent semirings $v + v = v$ holds, and so

> capturing one single tree of value v amounts to
> capturing the whole contribution of v to $V(G)$.

Theorem [EKL 10]: Let $X = f(X)$ be a system with n equations over an idempotent and commutative semiring. Then $\mu f = V(G^{n+1})$.

Stronger version of a theorem by Hopkins and Kozen in LICS'99.

Solving the linear equations

Recall: $V(U^i)$ is the least solution of

$$\begin{aligned} X &= V(a) \cdot V(U^{i-1})^2 + V(a) \cdot V(G^{i-1}) \cdot X \\ &+ V(a) \cdot X \cdot V(G^{i-1}) + V(b) \cdot X \end{aligned}$$

Neither left- nor right linear!

In a commutative and idempotent semiring the equation is equivalent to

$$X = V(a) \cdot V(U^{i-1})^2 + (V(a) \cdot V(G^{i-1}) + V(b)) \cdot X$$

which gives

$$V(U^i) = (V(a) \cdot V(G^{i-1}) + V(b))^* \cdot V(a) \cdot V(U^{i-1})^2$$

Solving equations over 1-bounded semirings

A semiring $S, +, \cdot, 0, 1$ 1-bounded if it is idempotent and $a \sqsubseteq 1$ for every semiring element a.

(Note: commutativity not required)

Example: Viterbi's semiring for computing maximal probabilities.

We use derivation tree analysis to show that for a system on n equations (and so n variables)

$$\mu f = V(G^n) = f^n(0)$$

Solving equations over 1-bounded semirings

Every tree t of height greater than n is pumpable: if t has yield w then there is $uvxyz = w$ and trees t^i with yield $uv^ixy^iz = w$ for every $i \geq 0$.

$$
\begin{array}{rcll}
V(t) + V(t^0) & = & V(uvxyz) + V(uxz) & \\
 & \sqsubseteq & V(u) \cdot 1 \cdot V(x) \cdot 1 \cdot V(z) & \\
 & & + \; V(u) \cdot V(x) \cdot V(z) & \text{(1-boundedness)} \\
 & = & V(uxz) & \text{(idempotence)} \\
 & = & V(t^0) &
\end{array}
$$

So t^0 captures the total contribution of value v.

Use now that t^0 has height at most n.

Solving equations over star-distributive semirings

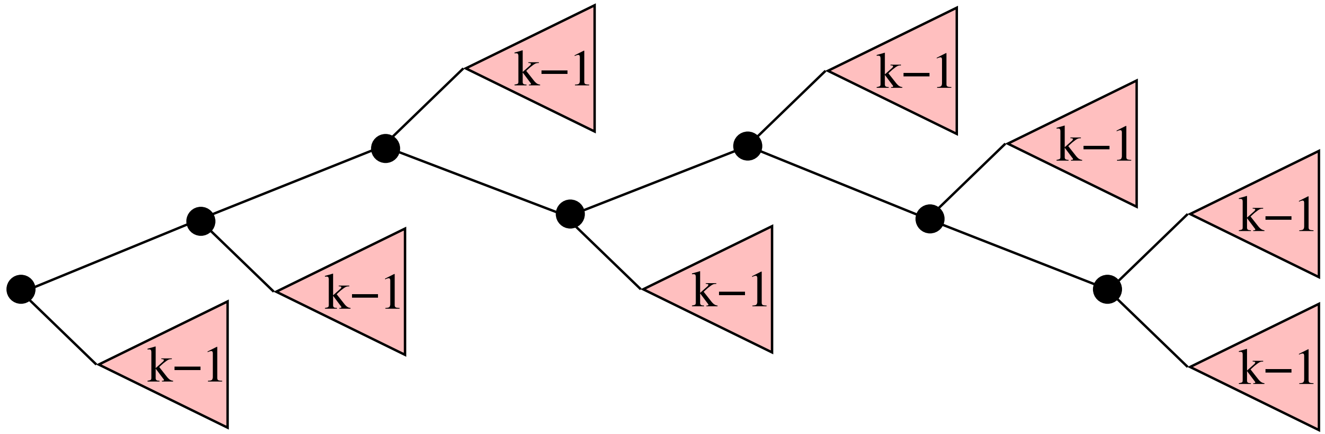
A semiring is star-distributive if it is idempotent, commutative, and $(a + b)^* = a^* + b^*$ for any semiring elements a, b.

Example: tropical semiring.

We use derivation tree analysis to show that for a system on n equations μf can be computed by n Kleene steps followed by one Newton step.

Solving equations over star-distributive semirings

A derivation tree is a bamboo if it has a path, the stem, such that the height of every subtree not containing a node of the stem is at most n.



Proposition: For every tree t there is a bamboo t' such that $V(t) = V(t')$.

Corollary: Bamboos already capture the contribution of all trees.

To compute: n Kleene steps for the trees of height at most n followed by one Newton step for the bamboos.

Some applications

Three new algorithms

$O(n^3)$ algorithm for computing the throughput of context-free grammars (improving $O(n^4)$ algorithm by Caucal et al.) [EKL TCS ’11].

New algorithm for pattern-based verification of multithreaded procedural programs with fixed number of threads [GMM CAV ’10, EG POPL ’11].

Very simple algorithm for transforming a context-free grammar into a Parikh-equivalent NFA [EGKL IPL ’11].

Stochastic thread creation

Threads can spawn new threads with known probabilities.

Execution by one processor. We assume termination with probability 1.

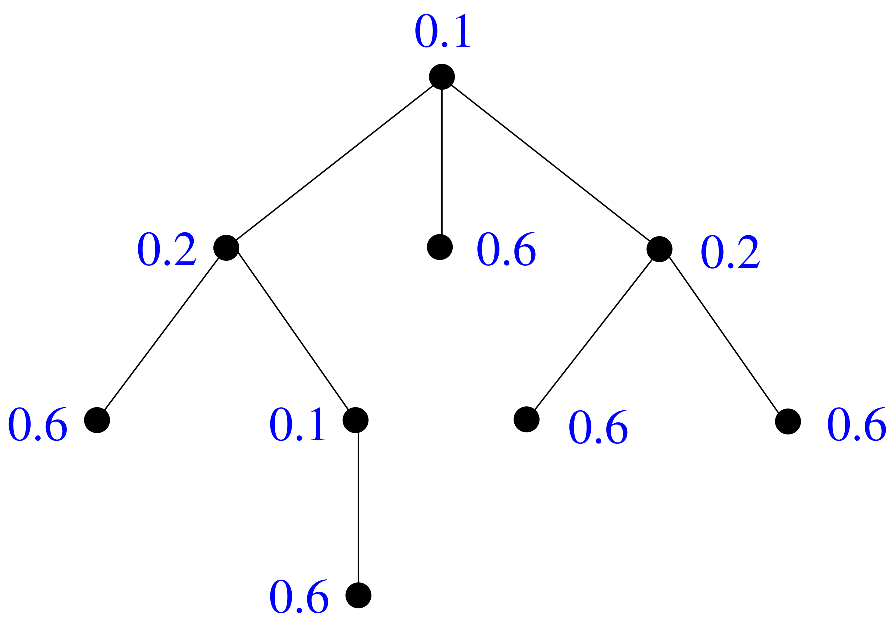
Example (only one type of thread):

$$X \xrightarrow{0.1} \langle X, X, X \rangle \quad X \xrightarrow{0.2} \langle X, X \rangle \quad X \xrightarrow{0.1} X \quad X \xrightarrow{0.6} \epsilon$$

Probability generating function

$$f(X) = 0.1X^3 + 0.2X^2 + 0.1X + 0.6$$

Describing executions: family trees



Probability of a family tree: product of the probabilities of its nodes.

Execution order depends on a scheduler that chooses a thread from the pool of inactive threads and executes it for one time unit.

Completion space S^{σ} for a scheduler σ: maximal size reached by the pool during execution.

Completion space of the optimal scheduler

Lemma: The family trees with completion space $S^{op} = k$ "are" the derivation trees of dimension k.

Theorem [BEKL I&C '11]: The probability $\Pr[S^{op} \leq k]$ of completing execution within space at most k is equal to the k-th Newton approximant of $X = f(X)$.

In our example:

$\Pr[S^{op} = 1]$	$= 2$	$= 3$	$= 4$	$= 5$
0.667	0.237	0.081	0.014	0.001

Conclusions and future work

New connections between analysis and numerical mathematics and TCS, leading to several new algorithms.

Open questions:

- Use language theory to derive convergence bounds of Newton's method over the reals
- Algebraic proof of the convergence speed theorem
- Applications to linear programming ?

Thermal equilibrium (2d)

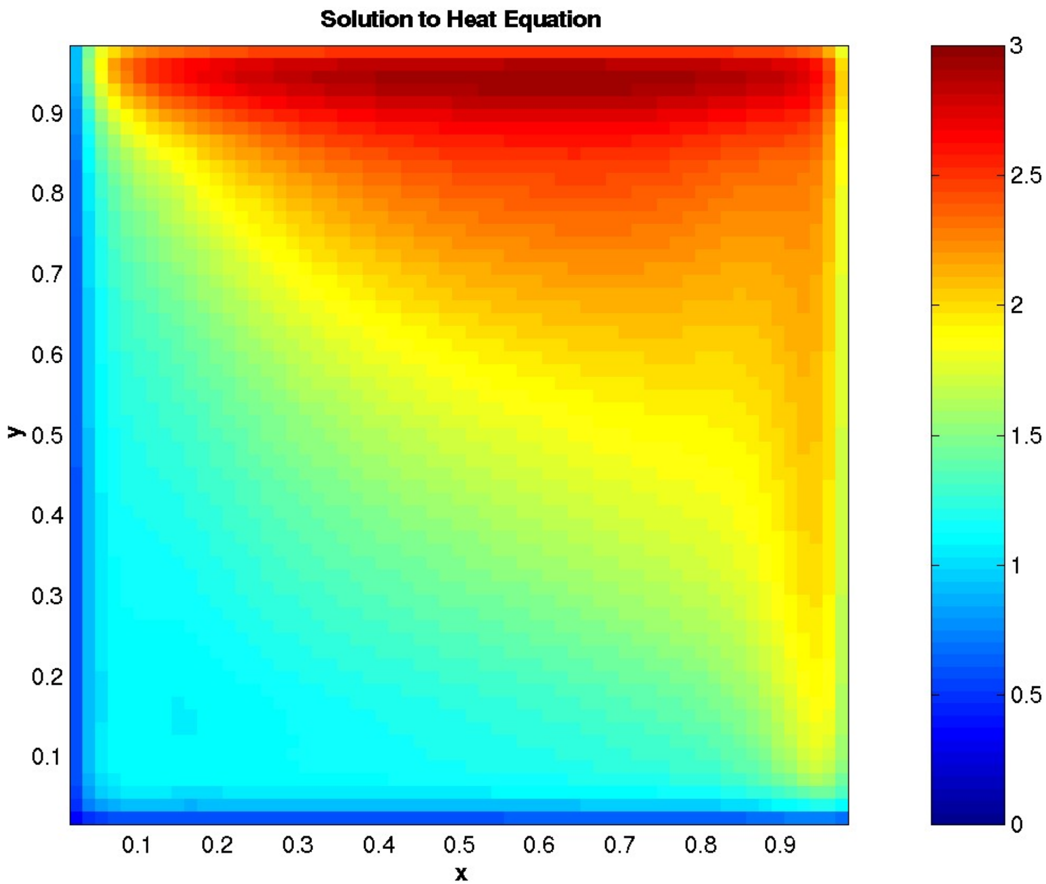
Heat equation in 2d

$$\frac{\partial u}{\partial t} = h^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$$

After discretization, temperature at thermal equilibrium is a solution of

$$u_{i,j} = k_{i,j}(u_{i-1,j} + u_{i+1,j} + u_{i,j+1} + u_{i,j-1})$$

for constants $k_{i,j}$ plus boundary conditions.



Abstract Interpretation: Collecting semantics

Collecting semantics of a program: assigns to each program point p the possible values of the memory when the program reaches p.

Solution of the equations

$$p_i \; Store = \bigsqcup_{p_j \in pred(p_i)} f_{ij}(p_j \; Store)$$

Basis of abstract interpretation

Idempotent semirings: derivation tree analysis

Idempotent semiring: $a + a = a$

Technique for computing μf algebraically:

(1) Identify a set $T \subseteq D$ of trees such that $Y(T)$ can be computed algebraically.

(2) Show that for every $t \in D$ there is $t' \in T$ such that $Y(t) \sqsubseteq Y(t')$.

Then by idempotence we have $\mu f = Y(D) = Y(T)$