A Brief History of Strahler Numbers

Javier Esparza Michael Luttenberger

Maximilian Schlund

Technical University of Munich

Robert E. Horton (1945)

BULLETIN OF THE GEOLOGICAL SOCIETY OF AMERICA

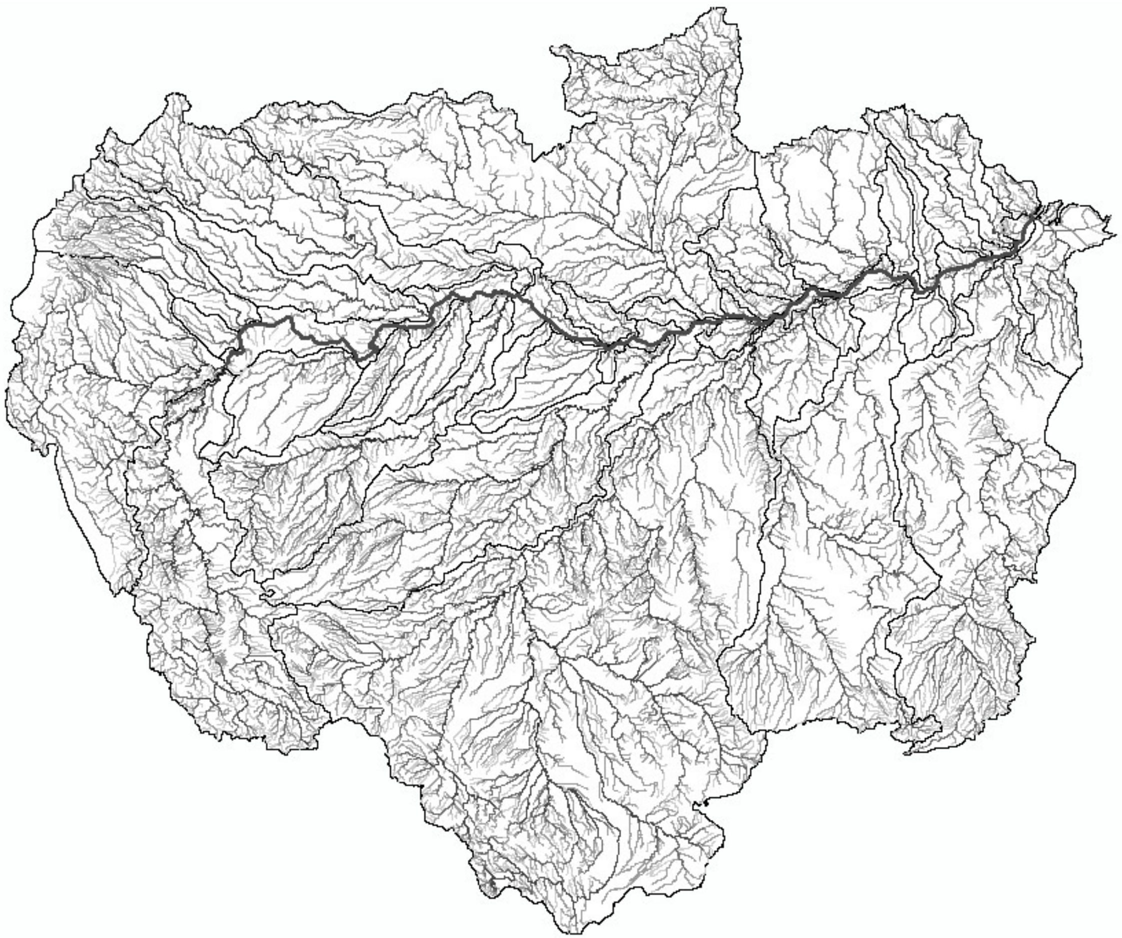
VOL. 56, PP. 275-370, 40 FIGS. MARCH 1945

EROSIONAL DEVELOPMENT OF STREAMS AND THEIR DRAINAGE BASINS; HYDROPHYSICAL APPROACH TO QUANTITATIVE MORPHOLOGY

BY ROBERT E. HORTON

Robert E. Horton (1945)

Which is the main stream of a stream system?



Robert E. Horton (1945)

Which is the main stream of a stream system?

Three step procedure.

First step: attach to each stream segment an order.

> *Unbranched fingertip tributaries are always designated as of order* **1***, tributaries or streams of the 2d order receive branches or tributaries of the 1st order, but these only; a 3d order stream must receive one or more tributaries of the 2d order but may also receive 1st order tributaries. A 4th order stream receives branches of the 3d and usually also of lower orders, and so on.*

Arthur N. Strahler (1952)

BULLETIN OF THE GEOLOGICAL SOCIETY OF AMERICA

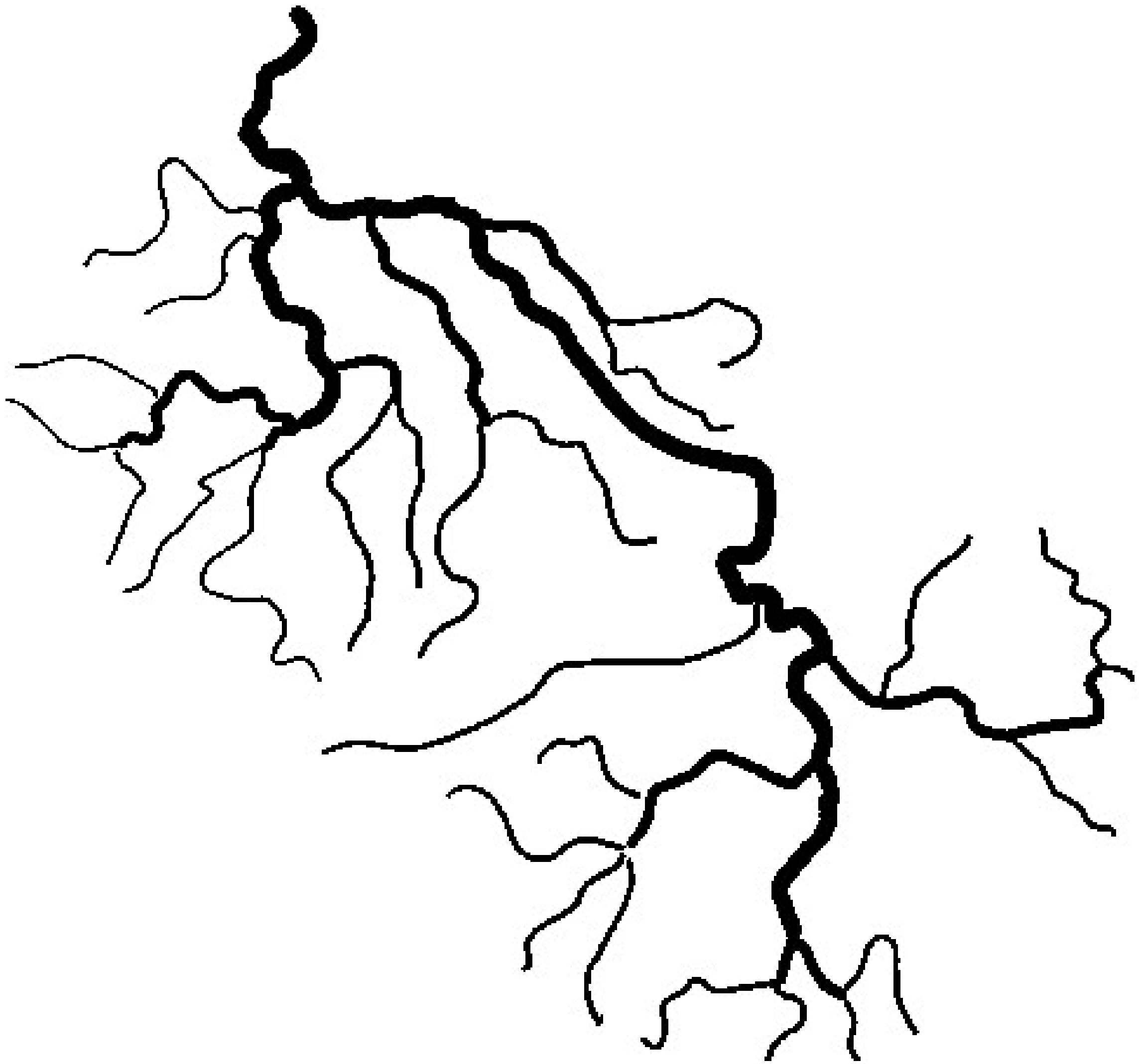
VOL. 63, PP. 1117-1142, 23 FIGS., 1 PL. NOVEMBER 1952

HYPSOMETRIC (AREA-ALTITUDE) ANALYSIS OF EROSIONAL TOPOGRAPHY

BY ARTHUR N. STRAHLER

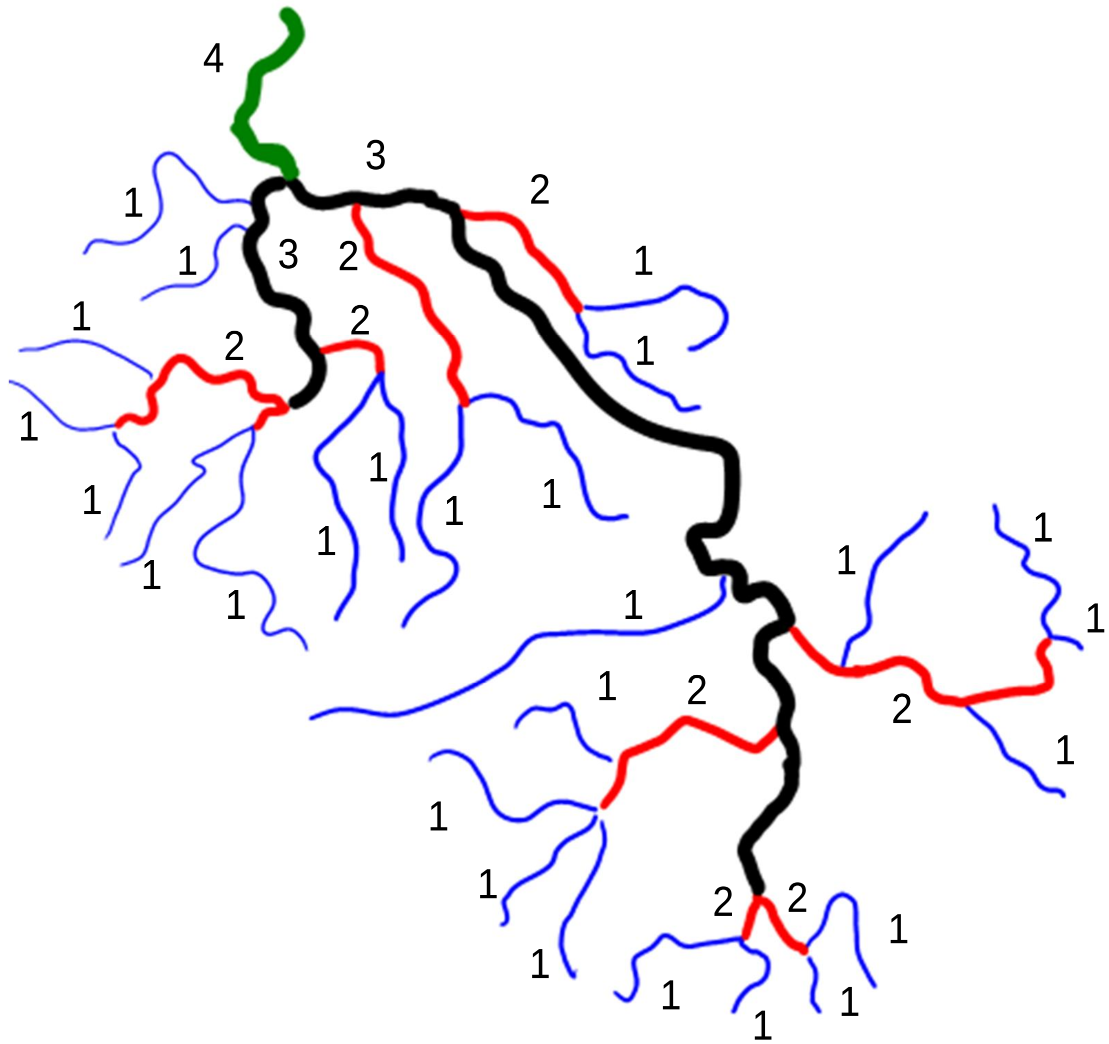
Arthur N.Strahler (1952)

The smallest, or ”finger-tip”, channels constitute the first-order segments. [. . .].
A second-order segment is formed by the junction of any two first-order streams; a third-order segment is formed by the joining of any two second order streams, etc.
Streams of lower order joining a higher order stream do not change the order of the higher stream



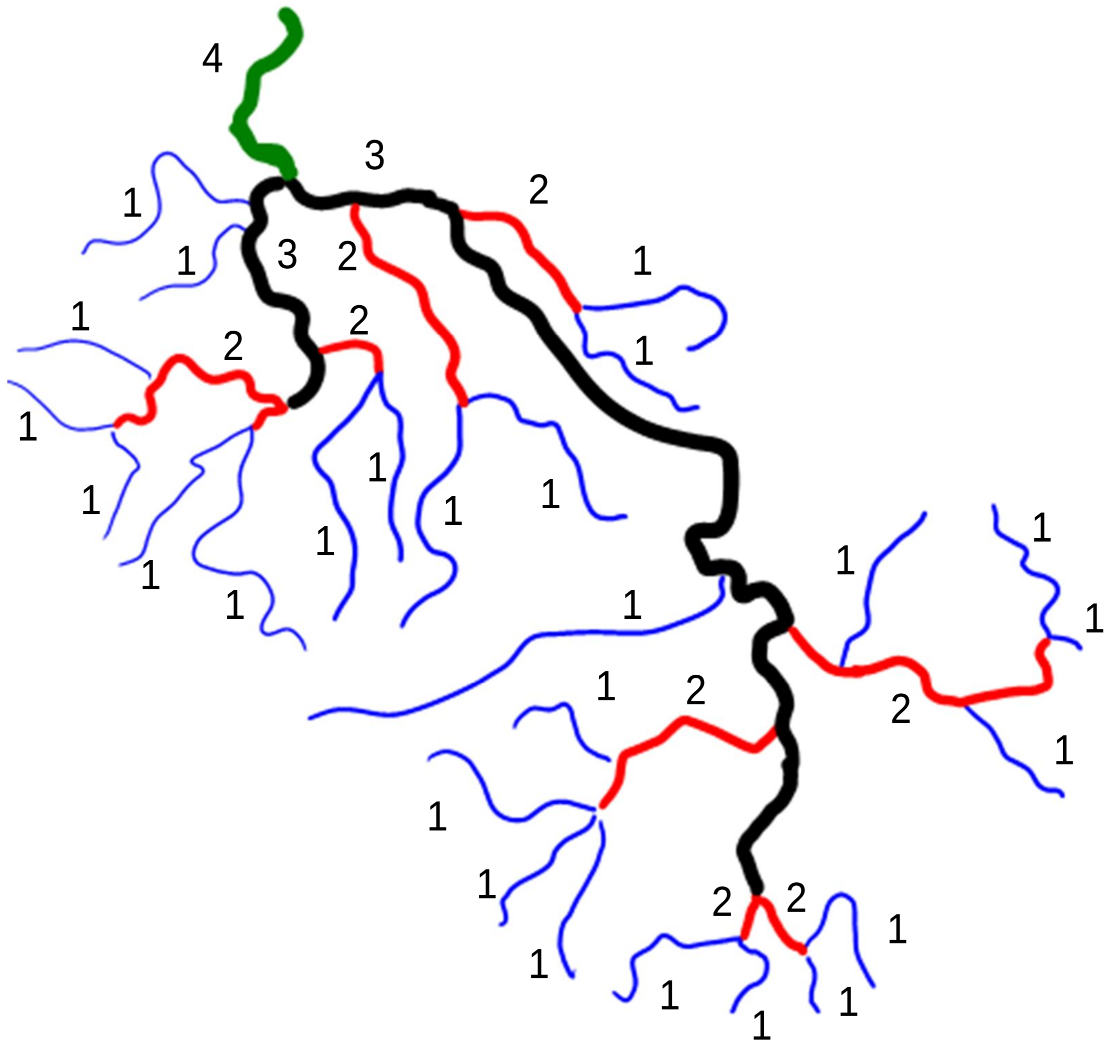
Arthur N.Strahler (1952)

The smallest, or ”finger-tip”, channels constitute the first-order segments. [. . .].
A second-order segment is formed by the junction of any two first-order streams; a third-order segment is formed by the joining of any two second order streams, etc.
Streams of lower order joining a higher order stream do not change the order of the higher stream



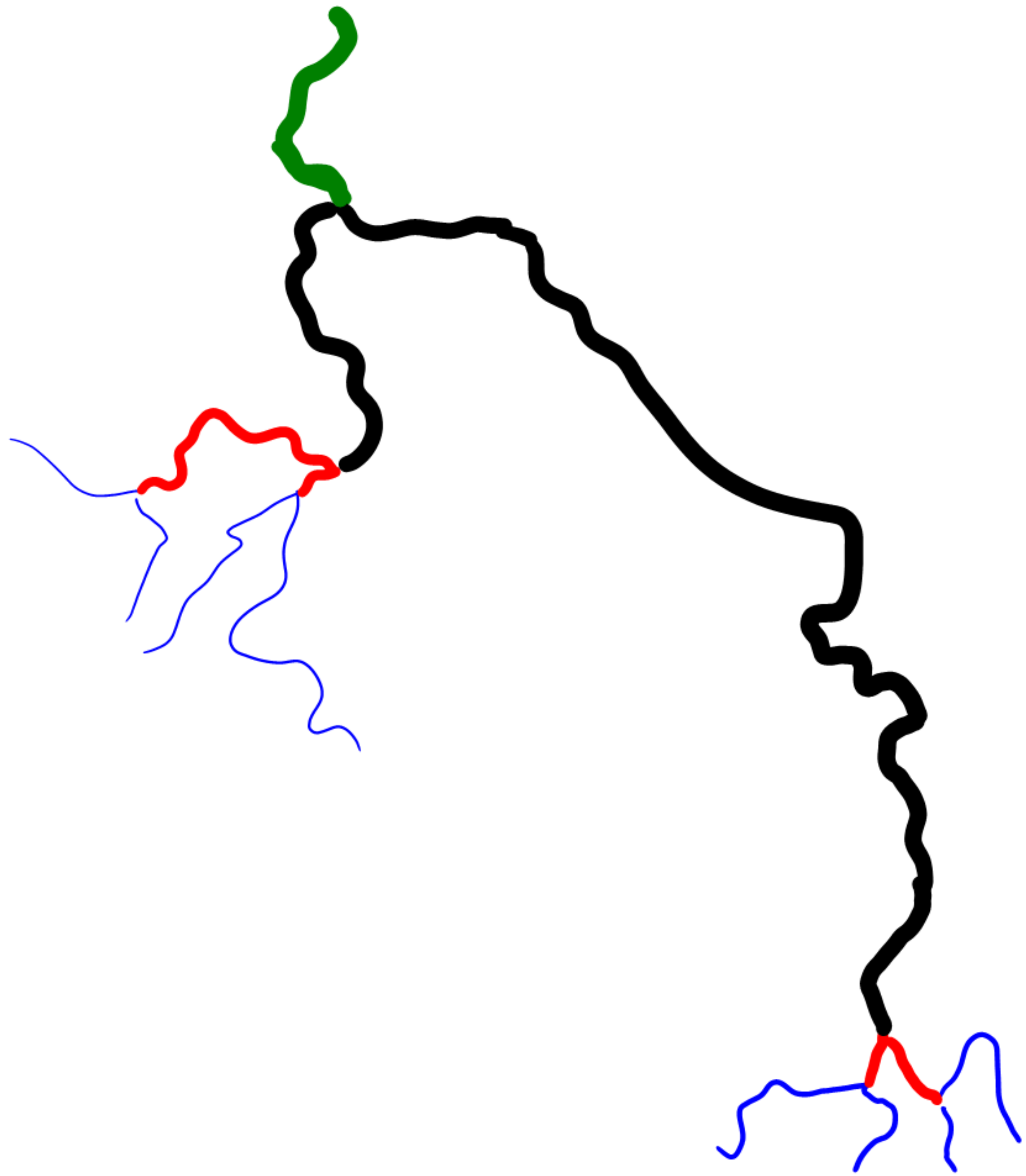
Arthur N.Strahler (1952)

Second step:
Remove all segments of lower order joining a higher order stream.



Arthur N.Strahler (1952)

Second step:
Remove all segments of lower order joining a higher order stream.

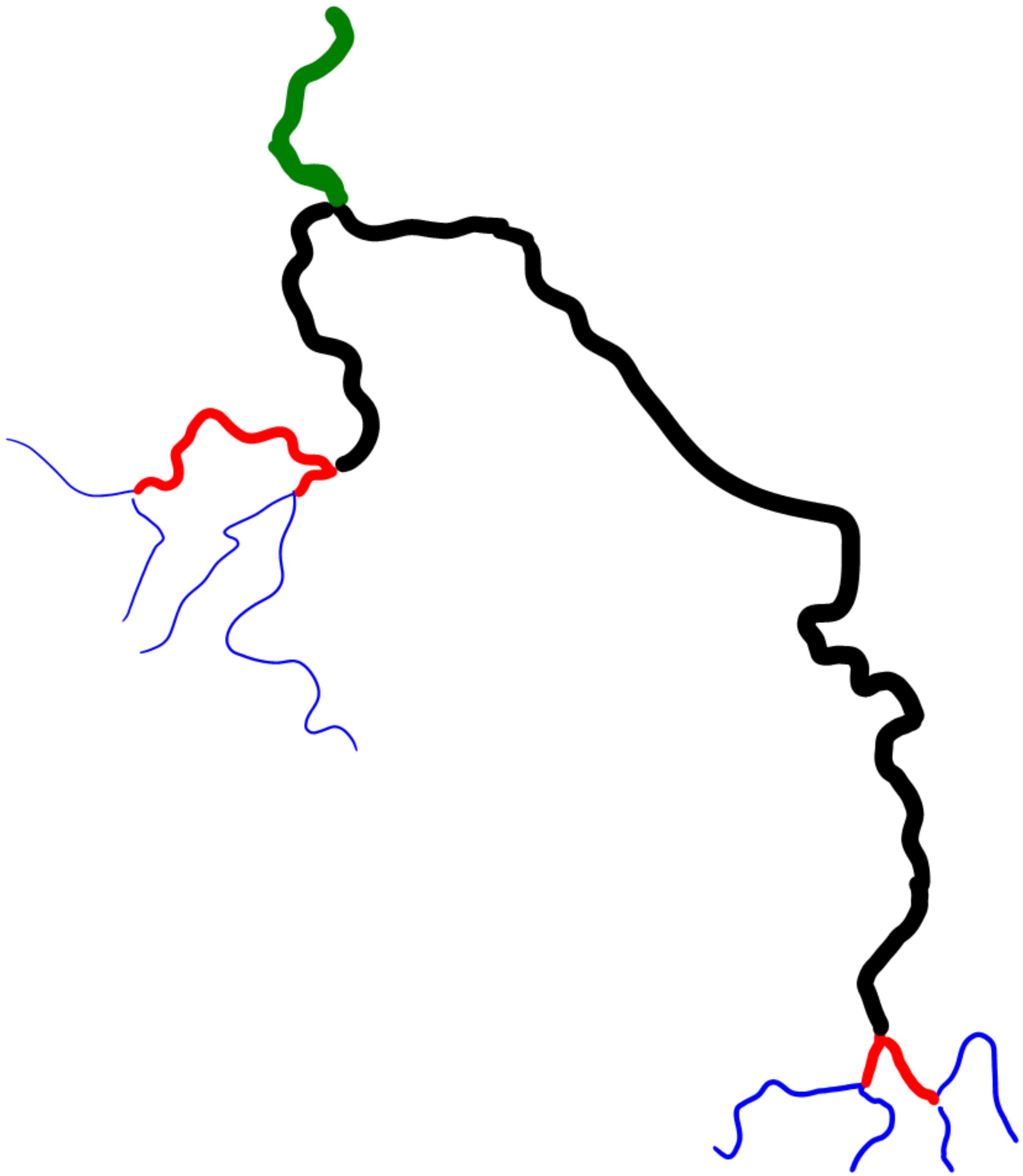


Robert E. Horton (1945)

Third step:

(1) *Starting below the junction, extend the parent stream upstream from the bifurcation in the same direction. The stream joining the parent stream at the greatest angle is of the lower order [. . .]*

(2) *If both streams are at about the same angle to the parent stream at the junction, the shorter is usually taken as of the lower order.*

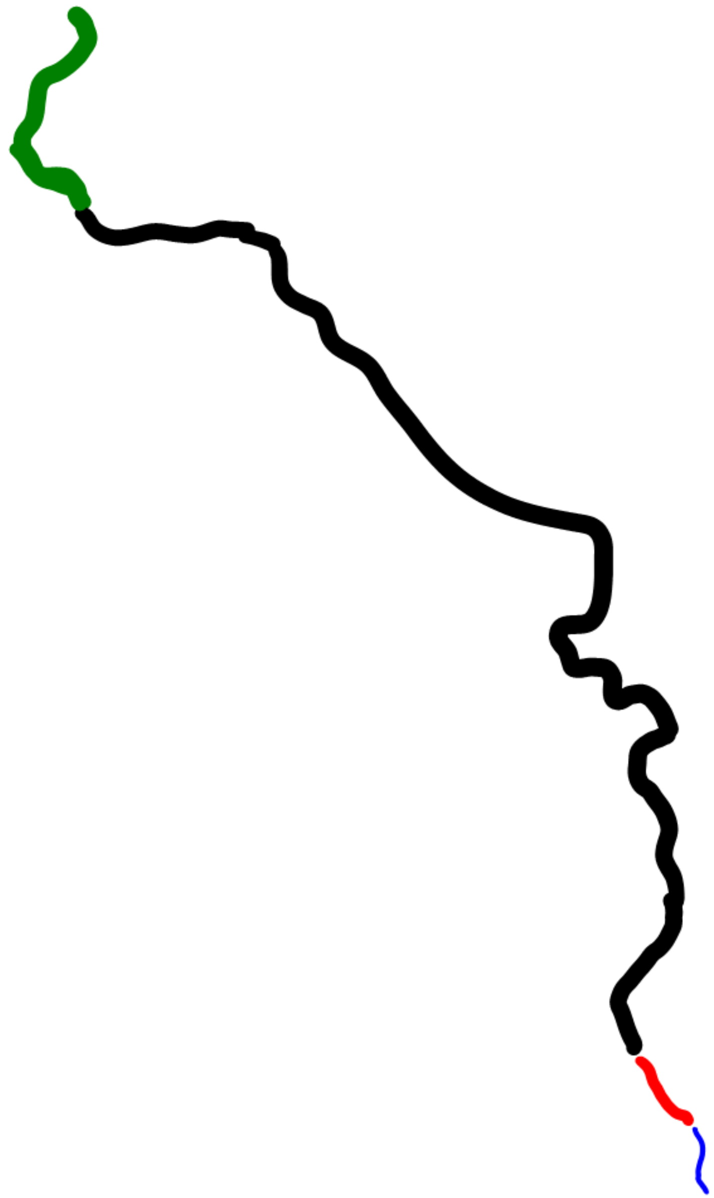


Robert E. Horton (1945)

Third step:

(1) *Starting below the junction, extend the parent stream upstream from the bifurcation in the same direction. The stream joining the parent stream at the greatest angle is of the lower order [. . .]*

(2) *If both streams are at about the same angle to the parent stream at the junction, the shorter is usually taken as of the lower order.*

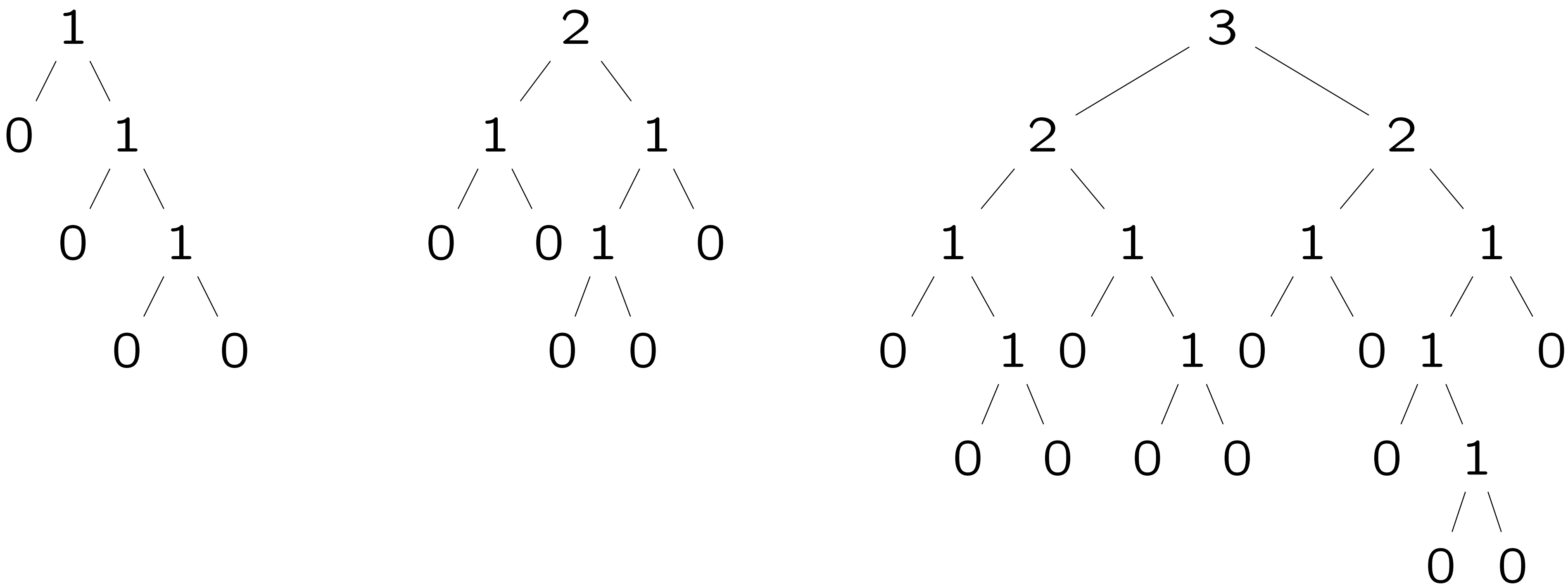


Strahler number of a tree

Definition: The Strahler number of a tree t, denoted by $S(t)$, is inductively defined as follows.

- If t has no subtrees (i.e., t has only one node), then $S(t) = 0$.
- If t has subtrees $t_1, \ldots, t_n$, then let $k = \max\{S(t_1), \ldots, S(t_n)\}$.
  If exactly one subtree of t has Strahler number k, then $S(t) = k$; otherwise, $S(t) = k + 1$.

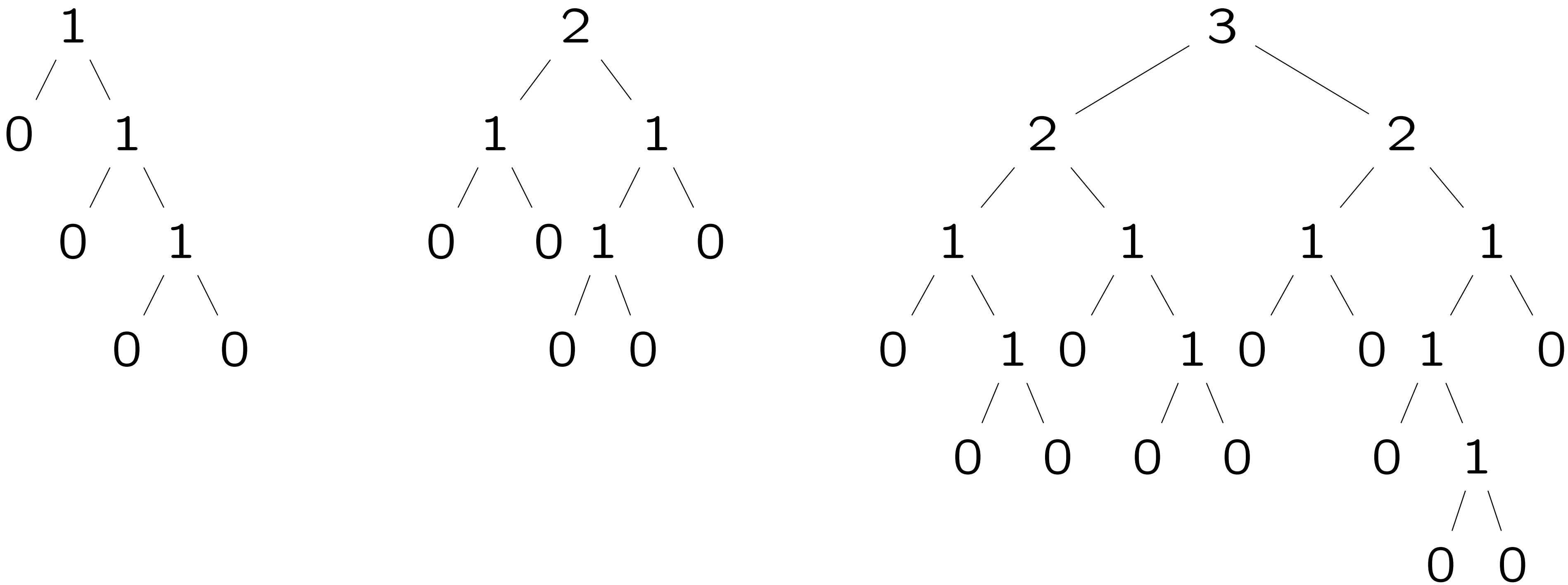
Strahler number of a tree



A characterization

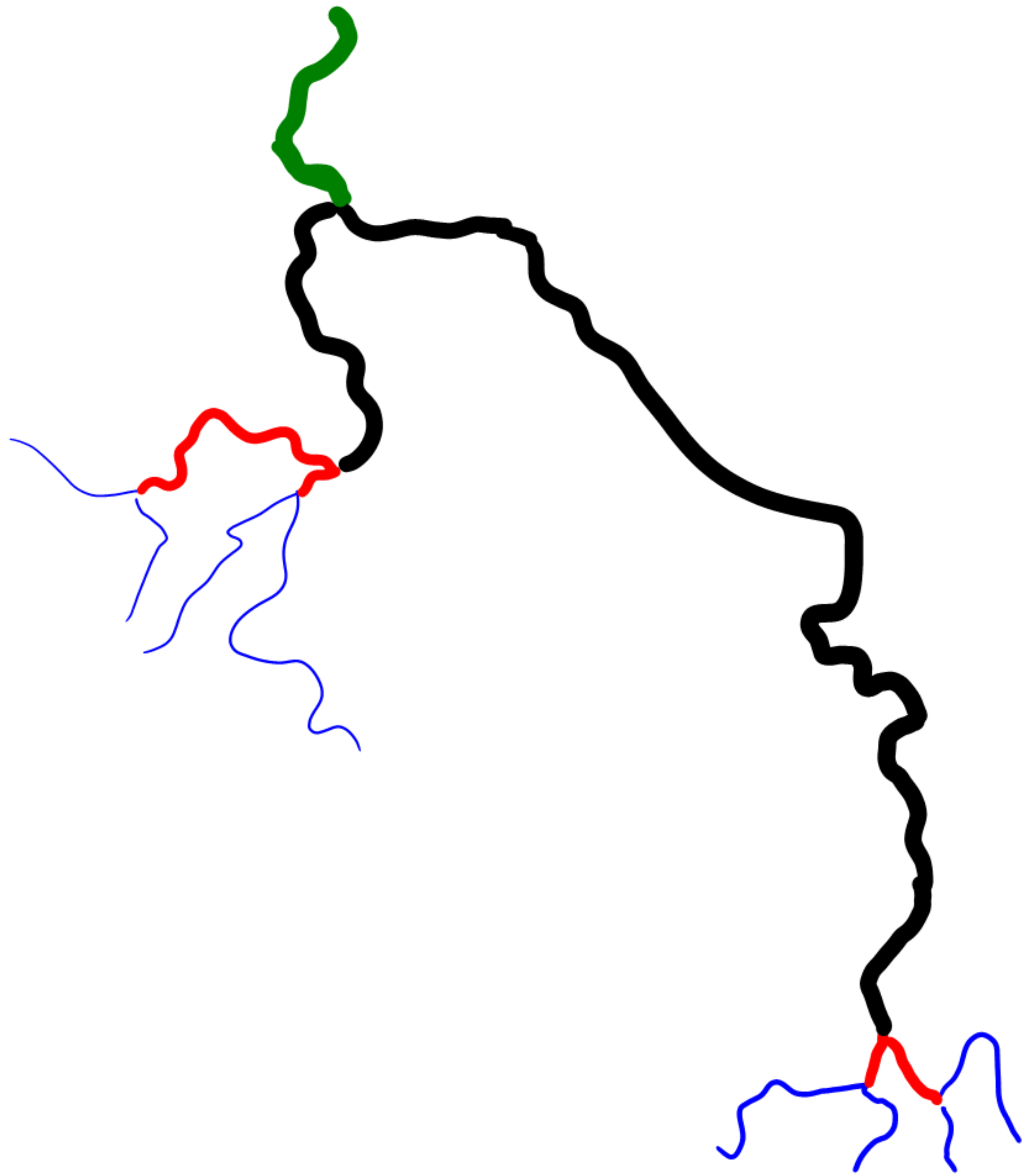
Fact: The Strahler number of a tree is the height of the largest minor of t that is a perfect binary tree.

Consequence: The height of a tree and the logarithm of its size are both upper bounds of its Strahler number.

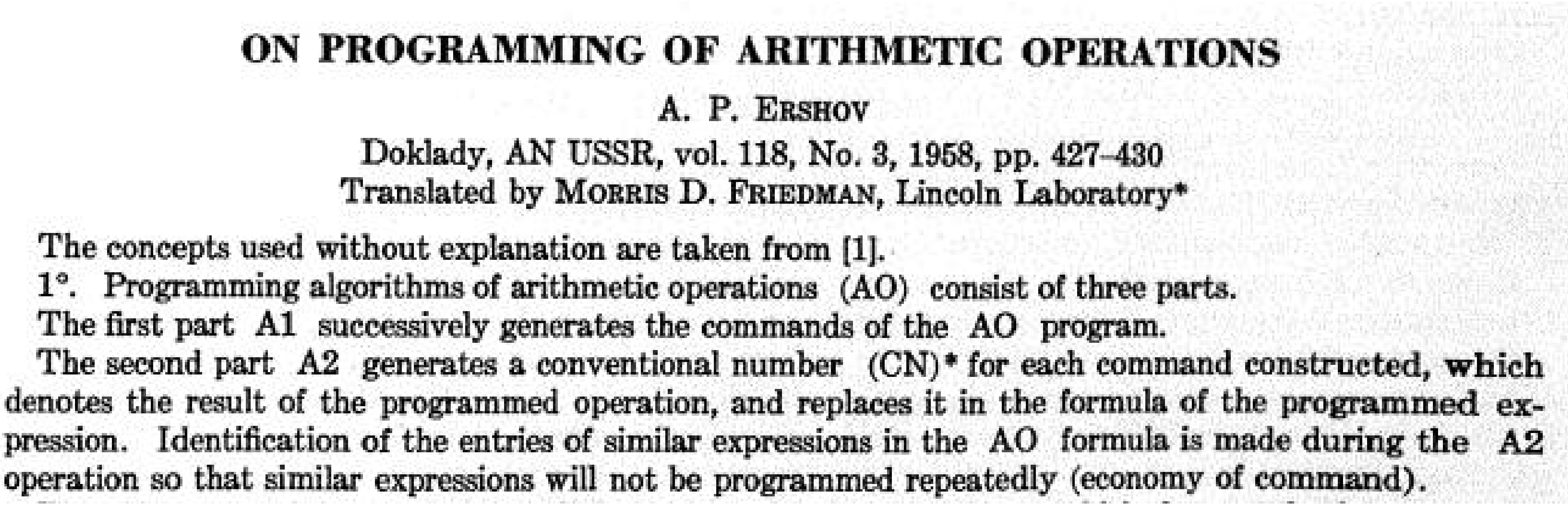


A characterization

Perfect binary tree for the Elbe river



Arithmetic circuits



ON PROGRAMMING OF ARITHMETIC OPERATIONS

A. P. Ershov

Doklady, AN USSR, vol. 118, No. 3, 1958, pp. 427–430

Translated by Morris D. Friedman, Lincoln Laboratory*

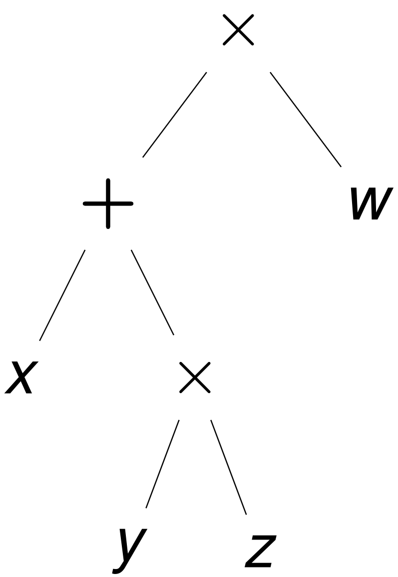
The concepts used without explanation are taken from [1].

1°. Programming algorithms of arithmetic operations (AO) consist of three parts.

The first part A1 successively generates the commands of the AO program.

The second part A2 generates a conventional number (CN)* for each command constructed, which denotes the result of the programmed operation, and replaces it in the formula of the programmed expression. Identification of the entries of similar expressions in the AO formula is made during the A2 operation so that similar expressions will not be programmed repeatedly (economy of command).

Arithmetic circuits



$$
\begin{aligned}
R_1 &\leftarrow x \\
R_2 &\leftarrow y \\
R_3 &\leftarrow z \\
R_4 &\leftarrow w \\
R_2 &\leftarrow R_2 \times R_3 \\
R_1 &\leftarrow R_1 + R_2 \\
R_4 &\leftarrow R_1 \times R_4
\end{aligned}
$$

$$
\begin{aligned}
R_1 &\leftarrow y \\
R_2 &\leftarrow z \\
R_2 &\leftarrow R_1 \times R_2 \\
R_2 &\leftarrow x \\
R_1 &\leftarrow R_1 + R_2 \\
R_2 &\leftarrow w \\
R_1 &\leftarrow R_1 \times R_2
\end{aligned}
$$

Arithmetic circuits

Theoretical Computer Science 9 (1979) 99–125

THE NUMBER OF REGISTERS REQUIRED FOR EVALUATING ARITHMETIC EXPRESSIONS*

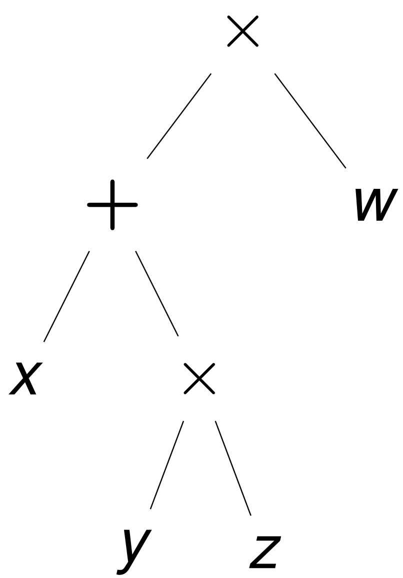
P. FLAJOLET, J.C. RAOULT and J. VUILLEMIN

Iria–Laboria, 78150 Rocquencourt, France
Université de Paris-Sud, Batiment 490, 91405 Orsay, France

Communicated by M. Nivat
Received January 1978
Revised May 1978

Arithmetic circuits

Fact: The minimal number of registers needed to evaluate an arithmetic expression is equal to the Strahler number of its syntax tree.



$$R_1 \leftarrow y$$
$$R_2 \leftarrow z$$
$$R_2 \leftarrow R_1 \times R_2$$
$$R_2 \leftarrow x$$
$$R_1 \leftarrow R_1 + R_2$$
$$R_2 \leftarrow w$$
$$R_1 \leftarrow R_1 \times R_2$$

Arithmetic circuits

Fact: The minimal number of registers needed to evaluate an arithmetic expression is equal to the Strahler number of its syntax tree plus 1.

Evaluation strategy: higher-number-first.

To evaluate $e_1 \; op \; e_2$:

evaluate the subexpression of higher Strahler number, say e_1,
storing the result in a register, say R_1;
reuse all other registers to evaluate e_2,
storing the result in one of them, say R_2;
store the result of $R_1 \; op \; R_2$ in R_1.

Arithmetic circuits

Which is the distribution of Strahler numbers in the binary trees with n leaves?

Consider binary trees with n internal nodes, chosen unformly at random. Let S_n be the random variable assigning to a tree its Strahler number.

Fact: $S_n \leq \lfloor \log_2(n+1) \rfloor$ (Strahler number $\leq$ height)

Theorem [Flajolet *et al.*'77, Kemp '79, Meir *et al.* '80]:

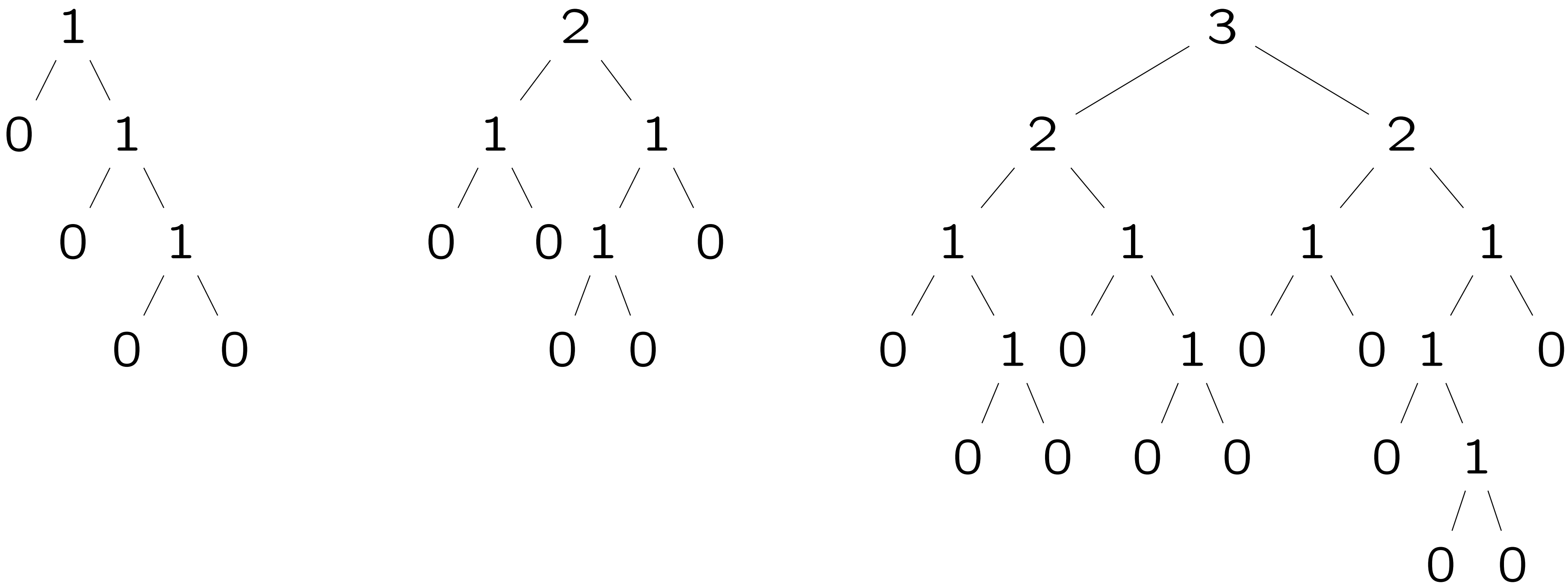
$$E[S_n] = \log_4 n + O(1) \qquad \text{and} \qquad Var[S_n] \in O(1)$$

Theorem [Devroye, Kruszewski, '95]: $Pr[S_n - \log_4 n \geq x] \leq \frac{c}{4^x}$.

A second characterization: Tree traversal

Fact [Flajolet, Raoult, Vuillemin '79]: The Strahler number of a binary tree is the minimal stack size needed to traverse it.

Follow a lower-number-first search strategy.



Strahler numbers and derivation trees

Definition [Ginsburg, Spanier '66]: The index of a derivation

$$S \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \cdots \Rightarrow \alpha_k \Rightarrow w$$

of a given CFG is the maximal number of variables occurring in any of $\alpha_1, \ldots, \alpha_k$

Example: $X \to aXX \mid b$

$X \Rightarrow aX\underline{X} \Rightarrow a\underline{X}aXX \Rightarrow aba\underline{X}X \Rightarrow abab\underline{X} \Rightarrow ababb$ Index 3

$X \Rightarrow a\underline{X}X \Rightarrow ab\underline{X} \Rightarrow aba\underline{X}X \Rightarrow abab\underline{X} \Rightarrow ababb$ Index 2

Fact: A derivation tree of a CFG in Chomsky normal form has index k iff its Strahler number is $(k-1)$.

Strahler numbers and derivation trees

From [Chytil and Monien, STACS '90]:

A caterpillar is an ordered tree in which all vertices of outdegree greater than one occur on a single path from the root to a leaf.
A 1-caterpillar is simply a caterpillar and for $k > 1$ a k-caterpillar is a tree obtained from a caterpillar by replacing each hair by a tree which is at most $(k-1)$-caterpillar.

Strahler numbers and derivation trees

Theorem [Chytil and Monien, STACS '90]: Let G be a CFG, and let $L_k(G)$ denote the words of $L(G)$ of index k. There is a nondeterministic Turing machine (pushdown automaton) with language $L(G)$ that recognizes $L_k(G)$ in $O(k \log |G|)$ space.

Proof idea: Let $a_1 \dots a_n$ be an input string.
At each moment the stack contains a sequence of triples of the form (A, i, j), where A non-terminal and $1 \leq i \leq j \leq n$.

(A, i, j) models the guess $A \Rightarrow^* a_i \dots a_j$.

Guess rule $A \rightarrow BC$, guess index $i \leq k < j$, guess which of B and C generates tree of smaller Strahler number, say C, and replace (A, i, j) by $(C, k+1, j)(B, i, j)$ (smaller-number-strategy).

Strahler numbers and derivation trees

Theorem: Emptiness of $L_k(G)$ can be checked in nondeterministic $O(k \log |G|)$ space.

Proof idea: Similar. At each moment stack contains a sequence of at most k variables.

Compare: emptiness of $L(G)$ is P-complete.

Strahler numbers and resolution proofs

Problem: estimate the complexity of resolution refutations

Space complexity: maximal number of "active clauses" during resolution.

The space complexity of a tree-like refutation is equal to its Strahler number.

Rich theory developed by several authors (recent survey by O. Kullmann)

Strahler numbers and Newton's method

We study systems of equations of the form

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

where the f_i's are polynomial expressions over ω-continuous semirings.

ω-continuous semirings

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

$(C, +, 0)$ is a commutative monoid $\qquad$ $\times$ distributes over $+$

$(C, \times, 1)$ is a monoid $\qquad$ $0 \times a = a \times 0 = 0$

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 tutorial: semiring $\equiv$ ω-continuous semiring.

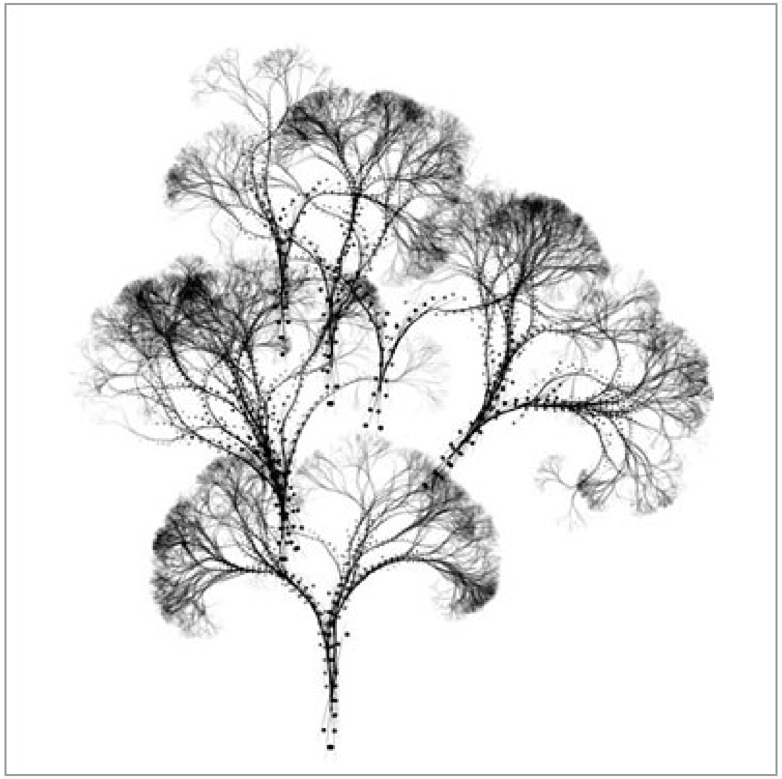
Context-free languages

Context-free grammar

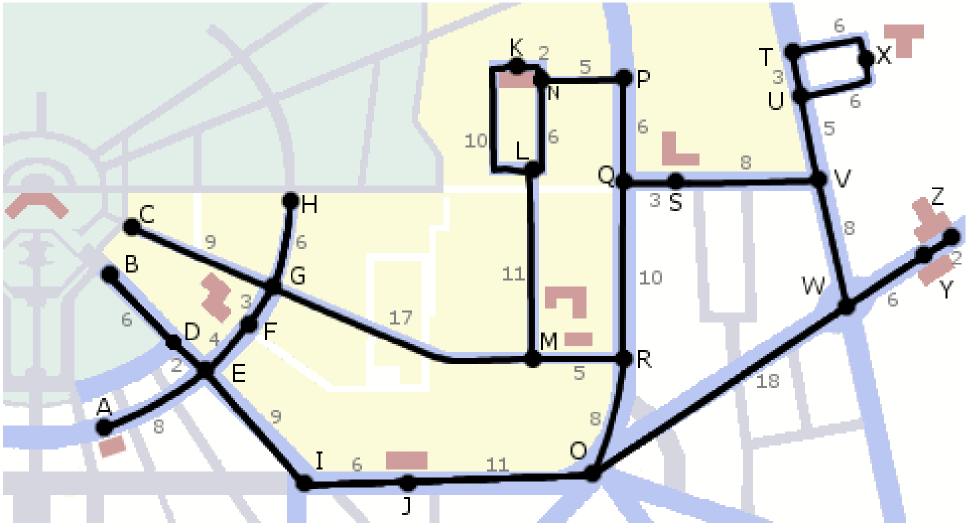
$$\begin{aligned} X &\rightarrow ZX \mid Z \\ Y &\rightarrow aYa \mid ZX \\ Z &\rightarrow b \mid aYa \end{aligned}$$

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

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



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.

Largest solution coincides with smallest solution over the tropical semiring.

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

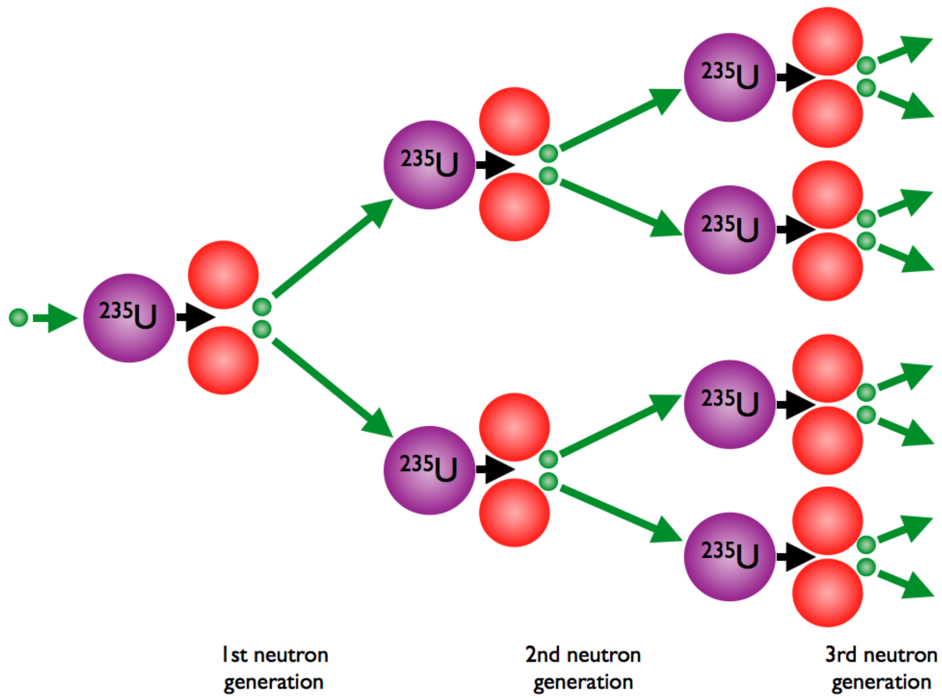
$$p_\alpha = k_\alpha + \int_0^D R_{\alpha,\beta}\, f(p_\beta)\, d\beta$$

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 $k_i, r_{i,j}$.



A generic solution method: Kleene iteration

Theorem [Kleene ’38, Tarsky ’55, Kuich ’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 $\mathcal{V}$.

Language-theoretic characterization of μf

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

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

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

Valuation: $\mathcal{V}(a) = 0.25, \mathcal{V}(b) = 0.25, \mathcal{V}(c) = 0.5$

Language-theoretic characterization of μf

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

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

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

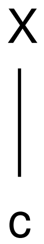
Valuation: $\mathcal{V}(a) = 0.25, \mathcal{V}(b) = 0.25, \mathcal{V}(c) = 0.5$

$\mathcal{V}$ extends to derivation trees and sets of derivation trees:

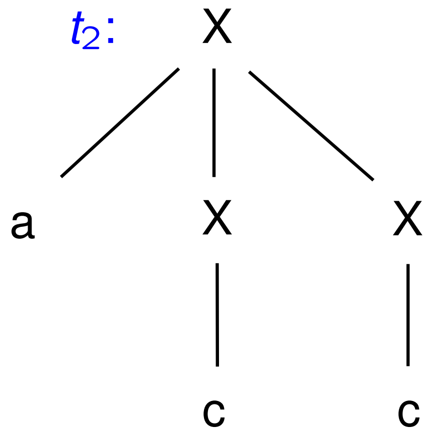
$$\begin{aligned} \mathcal{V}(t) &:= \text{ordered product of the leaves of } t \\ \mathcal{V}(T) &:= \sum_{t \in T} \mathcal{V}(t) \end{aligned}$$

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

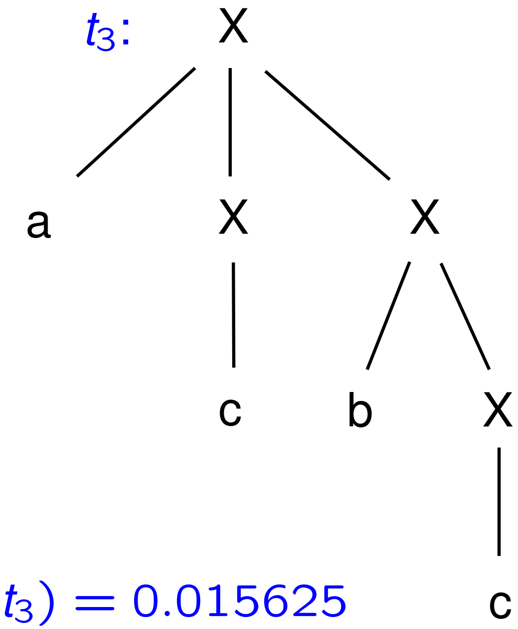
t_1:



t_2:



t_3:



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

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

Language-theoretic characterization of μf

Fundamental Theorem [Bozaalidis ’99, ..., E. Kiefer, Luttenberger’10]:
Let G be the grammar for $X = f(X)$, and let $T(G)$ be the set of derivation trees of G . Then $\mu f = \mathcal{V}(T(G))$

$$
\begin{tikzcd}
X = f(X) \arrow[r] \arrow[d] & G \arrow[d] \\
\mu f = \mathcal{V}(T(G)) & T(G) \arrow[l, "\mathcal{V}"']
\end{tikzcd}
$$

From now on: $\mathcal{V}(T(G)) \stackrel{def}{=} \mathcal{V}(G)$

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

- $T(U^i) \cap T(U^j) = \emptyset$ 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 $T(G^i) = T(U^1) \cup T(U^2) \cup \cdots \cup T(U^i)$

Approximating grammars

Define the operator Op as follows:

- $\mathcal{V}(G^1) = Op(0)$ and
- $\mathcal{V}(G^{i+1}) = Op(\mathcal{V}(G^i))$ 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).

(Height of a derivation tree measured after removing terminals)

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

.

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).

(Height of a derivation tree measured after removing terminals)

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

$$X_{\langle 0 \rangle} \;\rightarrow\; c$$

.

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).

(Height of a derivation tree measured after removing terminals)

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

$$X_{\langle 0 \rangle} \rightarrow c$$

$$X_{[0]} \rightarrow X_{\langle 0 \rangle}$$

.

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).

(Height of a derivation tree measured after removing terminals)

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

$$X_{\langle 0 \rangle} \rightarrow c$$

$$X_{[0]} \rightarrow X_{\langle 0 \rangle}$$

$$X_{\langle k \rangle} \rightarrow aX_{\langle k-1 \rangle}X_{\langle k-1 \rangle} \mid aX_{[k-2]}X_{\langle k-1 \rangle} \mid aX_{\langle k-1 \rangle}X_{[k-2]} \mid bX_{\langle k-1 \rangle}$$

.

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).

(Height of a derivation tree measured after removing terminals)

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

$$\begin{aligned}
X_{\langle 0 \rangle} &\rightarrow c \\
X_{[0]} &\rightarrow X_{\langle 0 \rangle} \\
X_{\langle k \rangle} &\rightarrow aX_{\langle k-1 \rangle}X_{\langle k-1 \rangle} \mid aX_{[k-2]}X_{\langle k-1 \rangle} \mid aX_{\langle k-1 \rangle}X_{[k-2]} \mid bX_{\langle k-1 \rangle} \\
X_{[k]} &\rightarrow X_{\langle k \rangle} \mid X_{[k-1]}
\end{aligned}$$

.

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).

(Height of a derivation tree measured after removing terminals)

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

$$\begin{aligned}
X_{\langle 0\rangle} &\rightarrow c \\
X_{[0]} &\rightarrow X_{\langle 0\rangle} \\
X_{\langle k\rangle} &\rightarrow aX_{\langle k-1\rangle}X_{\langle k-1\rangle} \mid aX_{[k-2]}X_{\langle k-1\rangle} \mid aX_{\langle k-1\rangle}X_{[k-2]} \mid bX_{\langle k-1\rangle} \\
X_{[k]} &\rightarrow X_{\langle k\rangle} \mid X_{[k-1]}
\end{aligned}$$

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

Computing approximants

$$X_{\langle k\rangle} \rightarrow aX_{\langle k-1\rangle}X_{\langle k-1\rangle} \mid aX_{[k-2]}X_{\langle k-1\rangle} \mid aX_{\langle k-1\rangle}X_{[k-2]} \mid bX_{\langle k-1\rangle}$$

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

Theorem: $\mathcal{V}(G^0) = f(0)$

$\mathcal{V}(G^{k+1}) = f(\mathcal{V}(G^k))$ for every $k \geq 1$

Example: $G : X \rightarrow aXX \mid bX \mid c \quad f(X) = aXX + bX + c$

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

Approximating grammars by Strahler number

Recall the approximation by height

$$\begin{aligned} X_{\langle k \rangle} &\rightarrow aX_{\langle k-1 \rangle}X_{\langle k-1 \rangle} \mid aX_{[k-2]}X_{\langle k-1 \rangle} \mid aX_{\langle k-1 \rangle}X_{[k-2]} \mid bX_{\langle k \rangle} \\ X_{[k]} &\rightarrow X_{\langle k \rangle} \mid X_{[k-1]} \end{aligned}$$

To capture more trees we now approximate by Strahler number.

$$\begin{aligned} X_{\langle k \rangle} &\rightarrow aX_{\langle k-1 \rangle}X_{\langle k-1 \rangle} \mid aX_{[k-1]}X_{\langle k \rangle} \mid aX_{\langle k \rangle}X_{[k-1]} \mid bX_{\langle k-1 \rangle} \\ X_{[k]} &\rightarrow X_{\langle k \rangle} \mid X_{[k-1]} \end{aligned}$$

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

Approximating grammars by Strahler number

Lemma: The derivation trees of U^k are the derivation trees of G of Strahler number k.

Lemma: The derivation trees of G^k are the derivation trees of G of Strahler number at most k.

Computing approximants

$$X_{\langle k \rangle} \;\rightarrow\; aX_{\langle k-1 \rangle}X_{\langle k-1 \rangle} \mid aX_{[k-1]}X_{\langle k \rangle} \mid aX_{\langle k \rangle}X_{[k-1]} \mid bX_{\langle k-1 \rangle}$$

$\mathcal{V}(U^k)$ is the least solution of the linear equation

$$X = a_{\mathcal{V}} \cdot \mathcal{V}(U^{k-1})^2 + a_{\mathcal{V}} \cdot \mathcal{V}(G^{k-1}) \cdot X + a_{\mathcal{V}} \cdot X \cdot \mathcal{V}(G^{k-1}) + b_{\mathcal{V}} \cdot X$$

and we get
$$\begin{aligned} \mathcal{V}(G^0) &= f(0) \\ \mathcal{V}(G^k) &= \mathcal{V}(G^{k-1}) + \mathcal{V}(U^k) \quad \text{for every } k \geq 1 \end{aligned}$$

Interpreting the new approximation

Recall that in our example $f(X) = aX^2 + bX + c$

Over the real semiring the equation

$$X = a_{\mathcal{V}} \cdot \mathcal{V}(U^{k-1})^2 + a_{\mathcal{V}} \cdot \mathcal{V}(G^{k-1}) \cdot X + a_{\mathcal{V}} \cdot X \cdot \mathcal{V}(G^{k-1}) + b_{\mathcal{V}} \cdot X$$

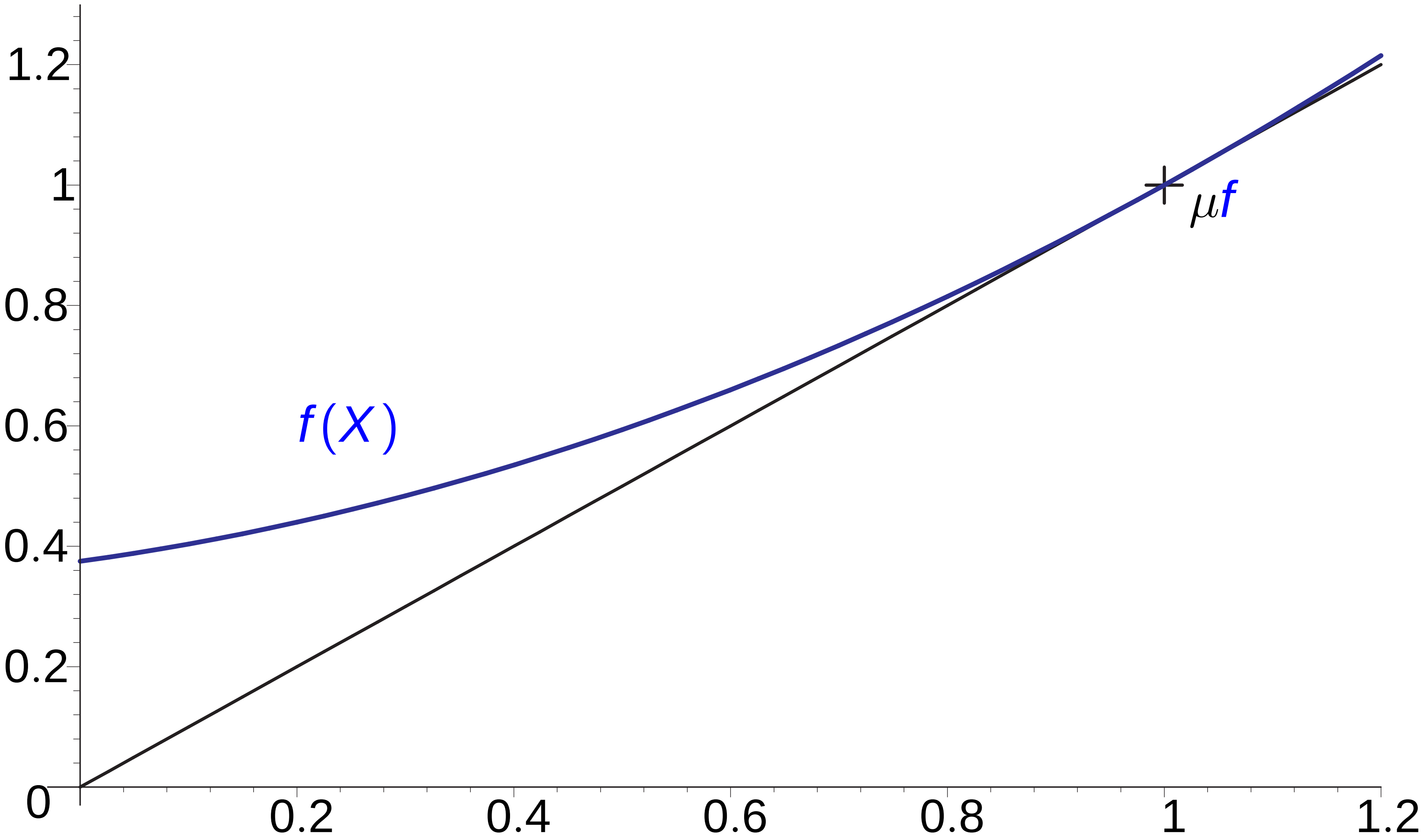
can be rewritten as

$$X = a_{\mathcal{V}} \cdot \mathcal{V}(U^{k-1})^2 + f'(\mathcal{V}(G^{k-1})) \cdot X$$

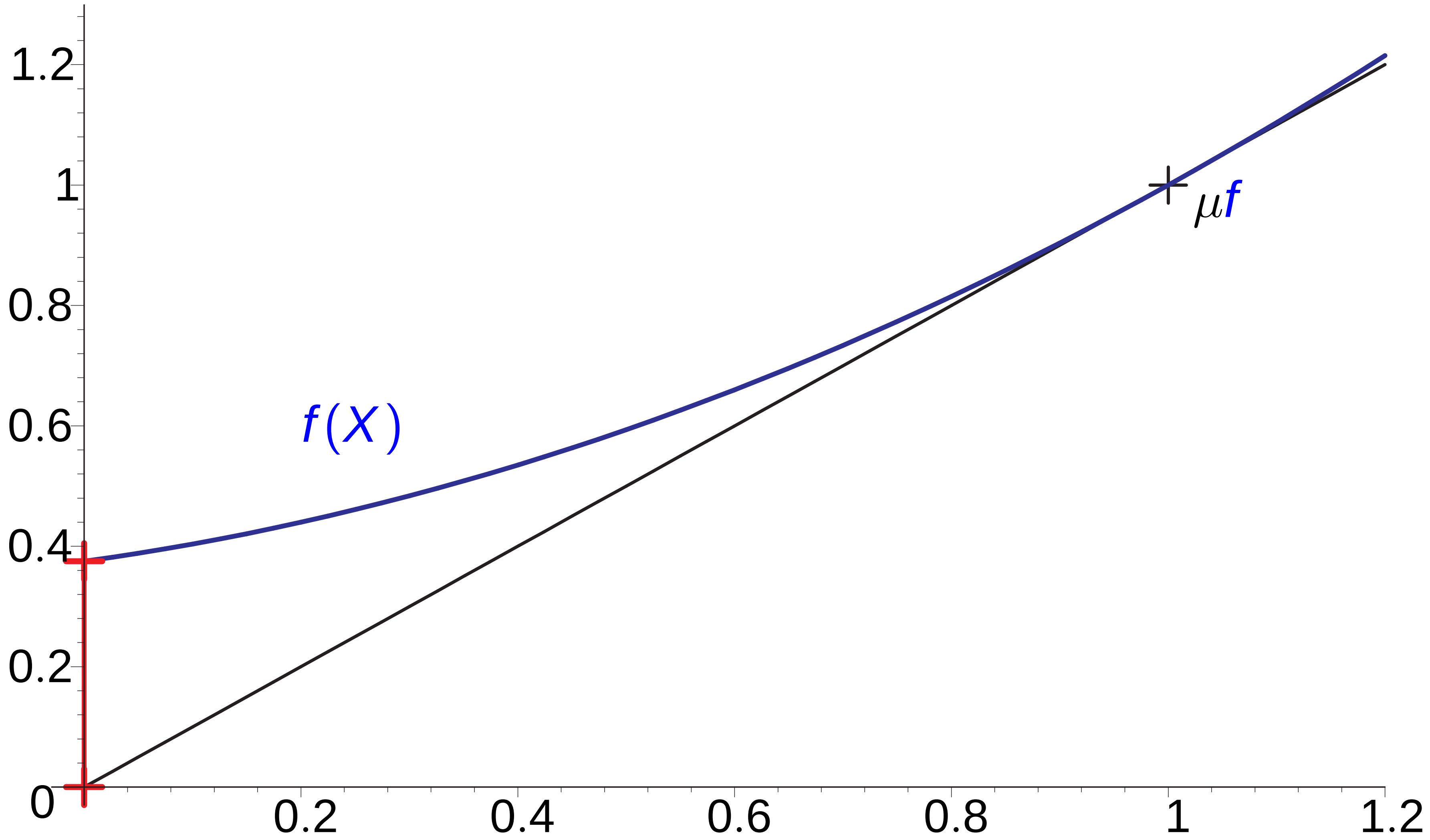
and therefore

$$\mathcal{V}(U^k) = \frac{a_{\mathcal{V}} \cdot \mathcal{V}(U^{k-1})^2}{1 - f'(\mathcal{V}(G^{k-1})} \qquad \mathcal{V}(G^k) = \mathcal{V}(G^{k-1}) - \frac{a_{\mathcal{V}} \cdot \mathcal{V}(U^{k-1})^2}{f'(\mathcal{V}(G^{k-1}) - 1}$$

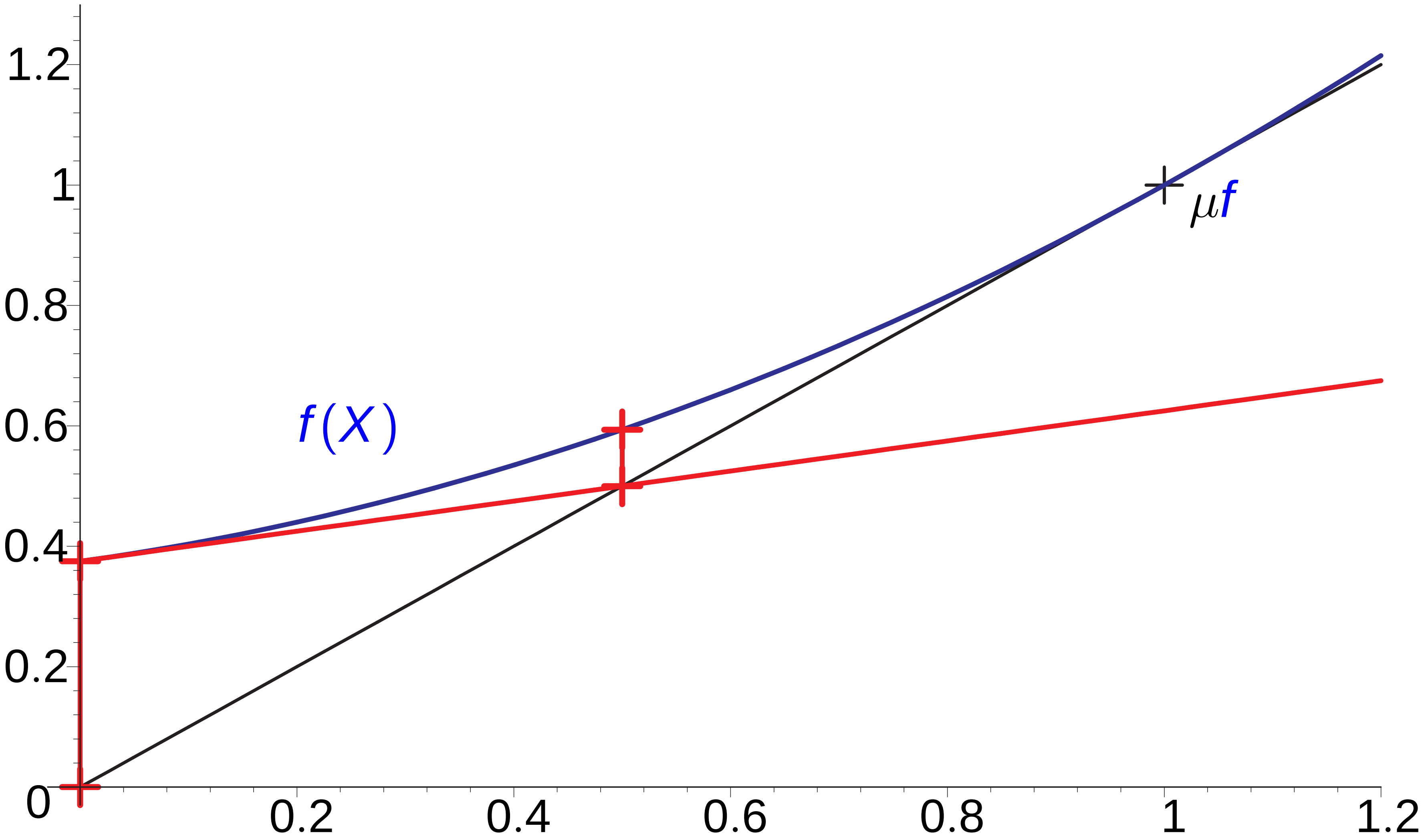
Newton's method for $X = f(X)$ (univariate case)



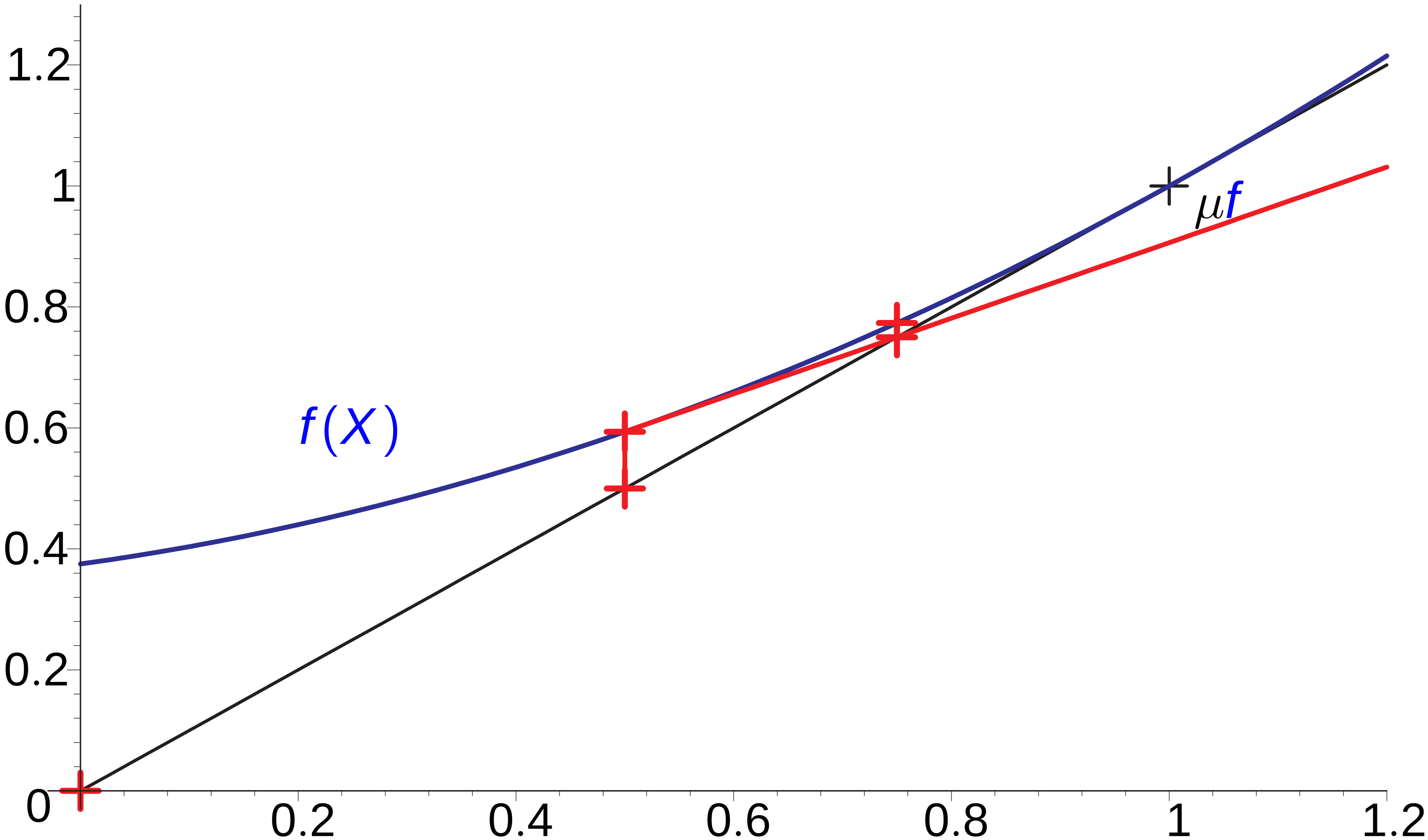
Newton's method for $X = f(X)$ (univariate case)



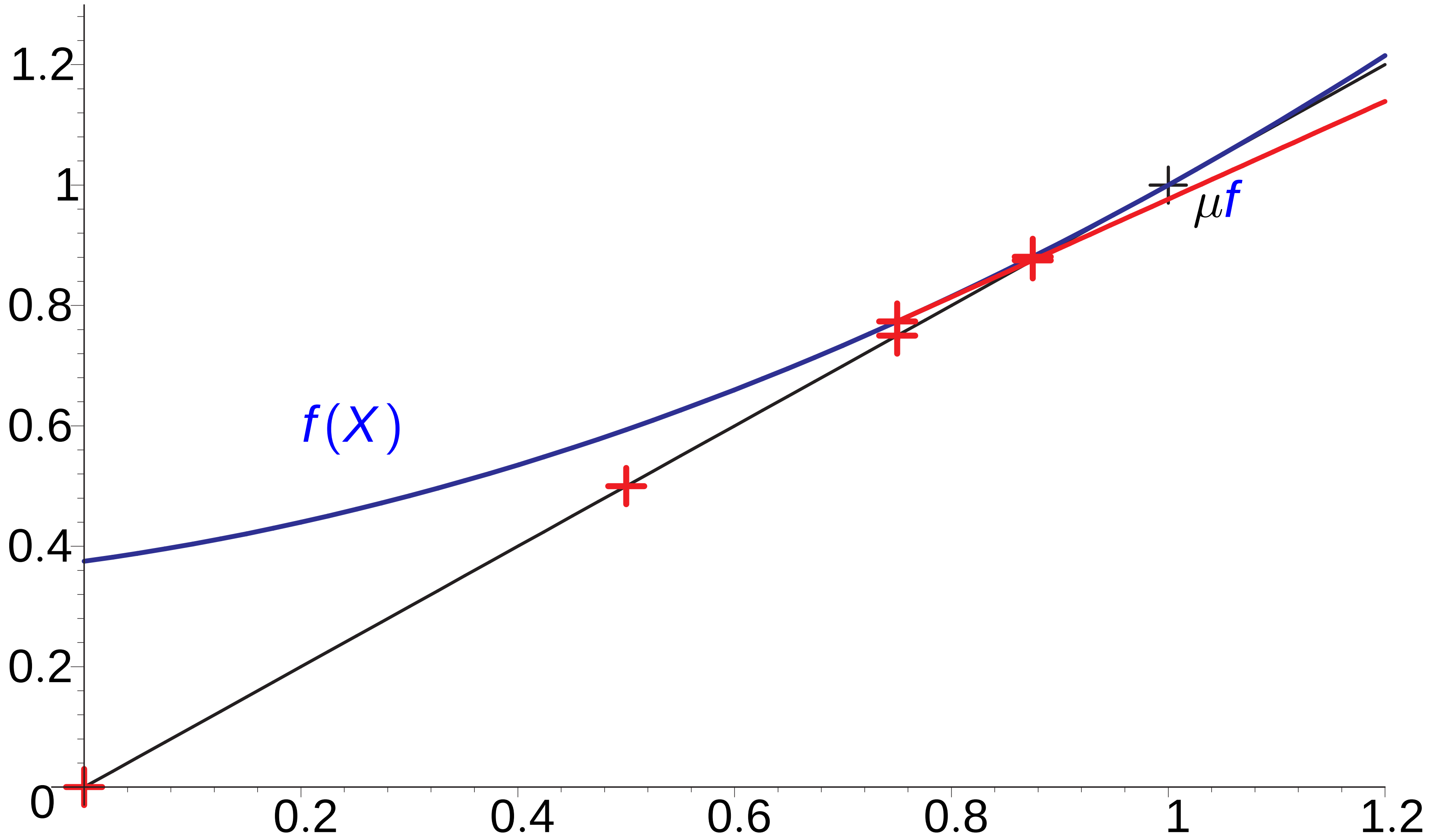
Newton's method for $X = f(X)$ (univariate case)



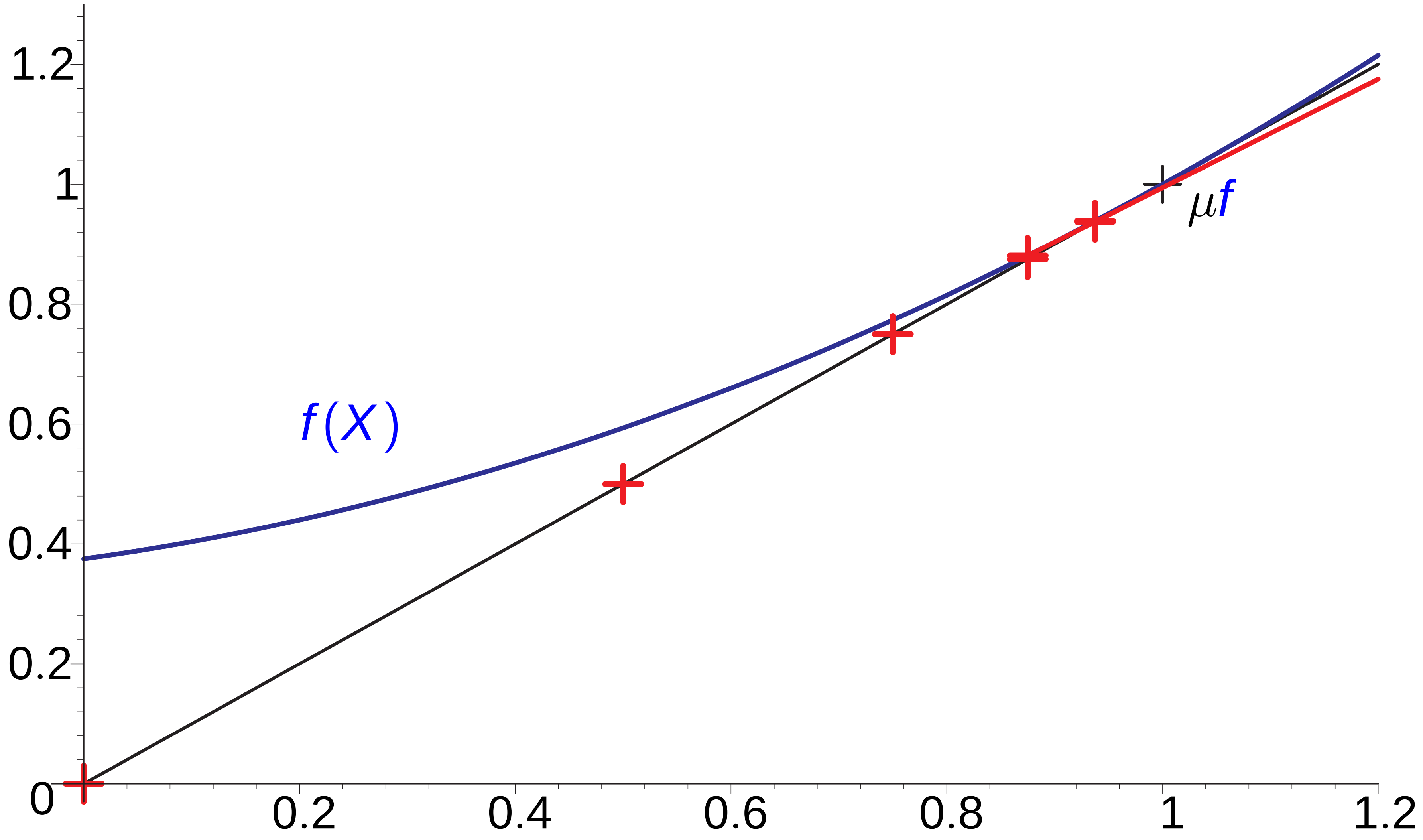
Newton's method for $X = f(X)$ (univariate case)



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



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



Mathematical formulation of Newton's Method

Let ν be some approximation of μf. (We start with $\nu = f(0)$.)

- Compute the linear function $T_\nu(X)$ for the tangent to $f(X)$ at ν
- Solve $X = T_\nu(X)$ (instead of $X = f(X)$), and take the solution as the new approximation

Elementary analysis: $T_\nu(X) = f'(\nu) \cdot (X - \nu) + f(\nu)$

Solving $X = T_\nu(X)$ yields

$$X = \frac{f(\nu) - f'(\nu) \cdot \nu}{1 - f'(\nu)} \qquad \nu' = \nu - \frac{f(\nu) - \nu}{f'(\nu) - 1}$$

Compare:

$$\mathcal{V}(G^k) = \mathcal{V}(G^{k-1}) - \frac{a_{\mathcal{V}} \cdot \mathcal{V}(U^{k-1})^2}{f'(\mathcal{V}(G^{k-1}) - 1}$$

$$\nu^{(k)} = \nu^{(k-1)} - \frac{f(\nu^{(k-1)}) - \nu^{(k-1)}}{f'(\nu^{(k-1)}) - 1}$$

Newton approximation corresponds to evaluating
the derivation trees of G by increasing Strahler number.

Convergence speed of Newton's method

For every semiring value v:

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

$$\mathcal{V}(G^i) = \sum_{v \in C} amb(i, v) \cdot v \qquad \mathcal{V}(G) = \sum_{v \in C} amb(v) \cdot v$$

Intuitively: $amb(i, v) \cdot v$ is the "contribution" of v to $\mathcal{V}(G^i)$.

$amb(v) \cdot v$ is the "contribution" of v to $\mathcal{V}(G)$.

We analyze how fast $amb(i, v)$ converges to $amb(v)$.

Convergence speed for commutative and idempotent semirings

In idempotent semirings $v + v = v$ holds, and so we only care whether $amb_i(v) = 0$ or $amb_i(v) > 0$

Theorem [E., Kiefer, Luttenberger '10]: Let $X = f(X)$ be a system with n equations over an idempotent and commutative semiring. Then for every value $v \in C$ we have $amb(v) > 0$ iff $amb_n(v) > 0$.

Corollary: $\mu f = \mathcal{V}(G^n)$.

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

Proof sketch

Show: For every value $v \in C$, if $\mathcal{V}(t) = v$ for some tree of G, then $\mathcal{V}(u) = v$ for some tree u of G^n.

Equivalently: For every derivation tree, some derivation tree of Strahler number at most n has the same value.

More generally: Let t be a derivation tree, and let k be the number of variables occurring in t. There is a tree u of Strahler number k such that $\mathcal{V}(t) = V(u)$.

Proof by induction on the number of nodes of t

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

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

Convergence speed for commutative semirings

Theorem [Luttenberger, Schlund '13]: Let $X = f(X)$ be a system with n equations over a commutative semiring. Then for every value $v \in C$ and for every $k \in \mathbb{N}$ we have $amb(n+k, v) \geq \min\{amb(v), 2^{2^k}\}$.

Solving equations over 1-bounded semirings

A semiring $(S, +, \cdot, 0, 1)$ is 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)

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

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 $i \geq 0$.

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

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

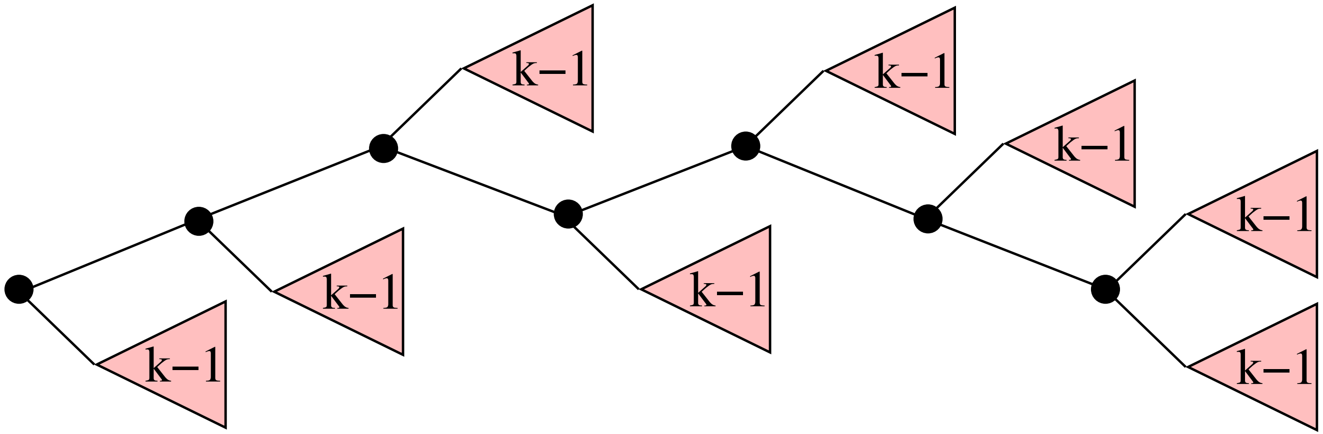
A semiring is star-distributive if it is idempotent, commutative, and $(a + b)^* = a^* + b^*$ for any semiring elements a, b.

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 $\mathcal{V}(t) = \mathcal{V}(t')$.

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

Parikh's theorem

Theorem [Parikh '66]: For every context-free language there is a regular language with the same commutative image.

Problem: Given a CFG G, construct an automaton A such that $L(G)$ and $L(A)$ have the same commutative image.

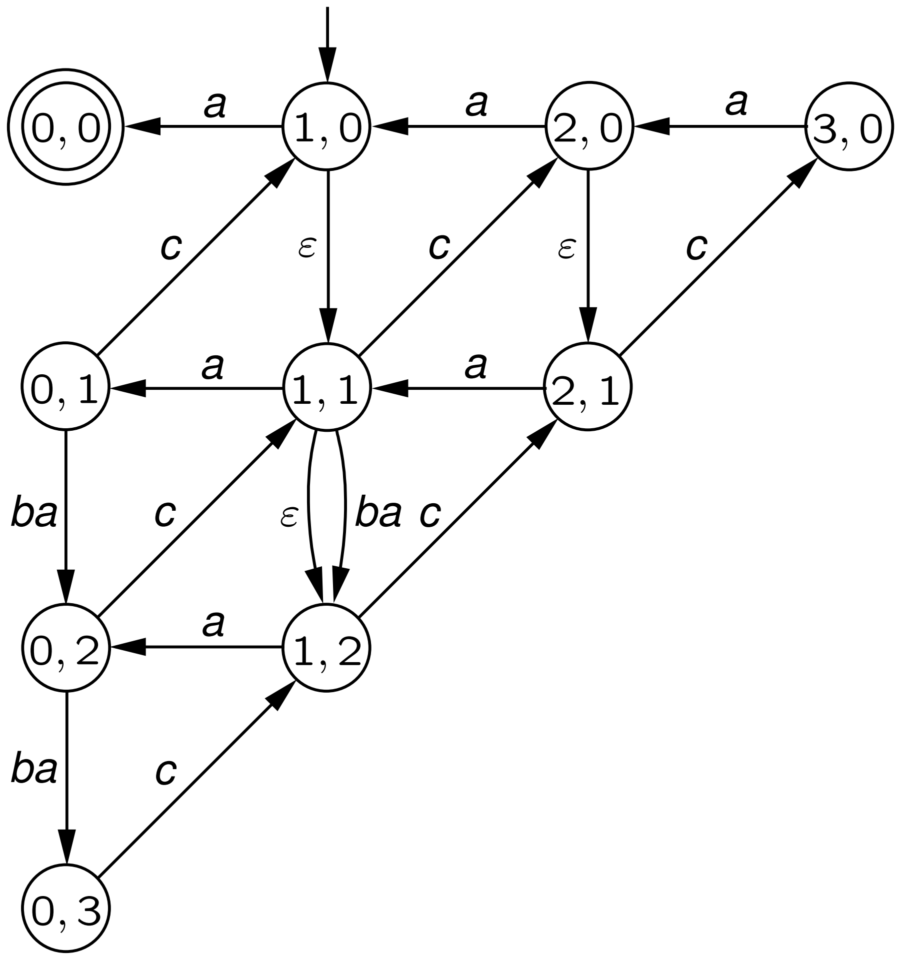
Solution: Use that $L(G)$ and $L(G^n)$ have the same commutative image.

Construct A whose runs "simulate" the derivations of G^n.

Parikh's theorem

Example: $A_1 \to A_1 A_2 \mid a \qquad A_2 \to bA_2 aA_2 | cA_1$

$$
\begin{array}{llcl}
 & A_1 & & (0,1) \\
\Rightarrow & A_1 A_2 & \xrightarrow{\epsilon} & (1,1) \\
\Rightarrow & A_1 b A_2 a A_2 & \xrightarrow{ba} & (1,2) \\
\Rightarrow & A_1 b c A_1 a A_2 & \xrightarrow{c} & (2,1) \\
\Rightarrow & abcA_1 a A_2 & \xrightarrow{a} & (1,1) \\
\Rightarrow & abcaaA_2 & \xrightarrow{a} & (0,1) \\
\Rightarrow & abcaacA_1 & \xrightarrow{c} & (1,0) \\
\Rightarrow & abcaaca & \xrightarrow{a} & (0,0)
\end{array}
$$



Lazy evaluation of And-Or trees

Nodes are only constructed and evaluated (to 0 or 1) if needed. (e.g., if left subtree of And-node evaluates to 1, right subtree is not constructed)

function And(node)
 if node.leaf() **then**
  return node.value()
 else
  v := Or(node.left)
  if $v = 0$ **then**
   return 0
  else
   return Or(node.right)

function Or(node)
 if node.leaf() **then**
  return node.value()
 else
  v := And(node.left)
  if $v = 1$ **then**
   return 1
  else
   return And(node.right)

Assume the probabilities that node.leaf() returns **true** and node.value() returns **1** are both $1/2$.

We perform an analysis to compute the probability that the evaluation terminates with a given value, and the average runtime.

Semiring elements: pairs (p, t).

Semiring operations:

$$
\begin{aligned}
(p_1, t_1) \cdot_e (p_2, t_2) &\stackrel{def}{=} (p_1 \cdot p_2, t_1 + t_2) \\
(p_1, t_1) +_e (p_2, t_2) &\stackrel{def}{=} \left(p_1 + p_2, \frac{p_1 \cdot t_1 + p_2 \cdot t_2}{p_1 + p_2}\right)
\end{aligned}
$$

The equations

$$
\begin{aligned}
\mathsf{And\_0} &= (0.25, 2) +_e (0.5, 1) \cdot_e (\mathsf{Or\_0} +_e \mathsf{Or\_1} \cdot_e \mathsf{Or\_0}) \\
\mathsf{And\_1} &= (0.25, 2) +_e (0.5, 1) \cdot_e \mathsf{Or\_1} \cdot_e \mathsf{Or\_1} \\
\mathsf{Or\_0} &= (0.25, 2) +_e (0.5, 1) \cdot_e \mathsf{And\_0} \cdot_e \mathsf{And\_0} \\
\mathsf{Or\_1} &= (0.25, 2) +_e (0.5, 1) \cdot_e (\mathsf{And\_1} +_e \mathsf{And\_0} \cdot_e \mathsf{And\_1})
\end{aligned}
$$

And_0: probability of (termination and) evaluation to 0, and average number of steps to termination

Kleene vs. Newton

Neither Kleene nor Newton terminate, but Newton converges faster:

i	$k^{(i)}$ And_0	$\nu^{(i)}$ And_0	$k^{(i)}$ And_1	$\nu^{(i)}$ And_1
0	$(0.250, 2.000)$	$(0.250, 2.000)$	$(0.250, 2.000)$	$(0.250, 2.000)$
1	$(0.406, 2.538)$	$(0.495, 3.588)$	$(0.281, 2.333)$	$(0.342, 3.383)$
2	$(0.448, 2.913)$	$(0.568, 5.784)$	$(0.333, 3.012)$	$(0.409, 5.906)$
3	$(0.491, 3.429)$	$(0.581, 6.975)$	$(0.350, 3.381)$	$(0.419, 7.194)$
4	$(0.511, 3.793)$	$(0.581, 7.067)$	$(0.370, 3.904)$	$(0.419, 7.295)$

Stochastic thread creation

Threads can spawn new threads with known probabilities.

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

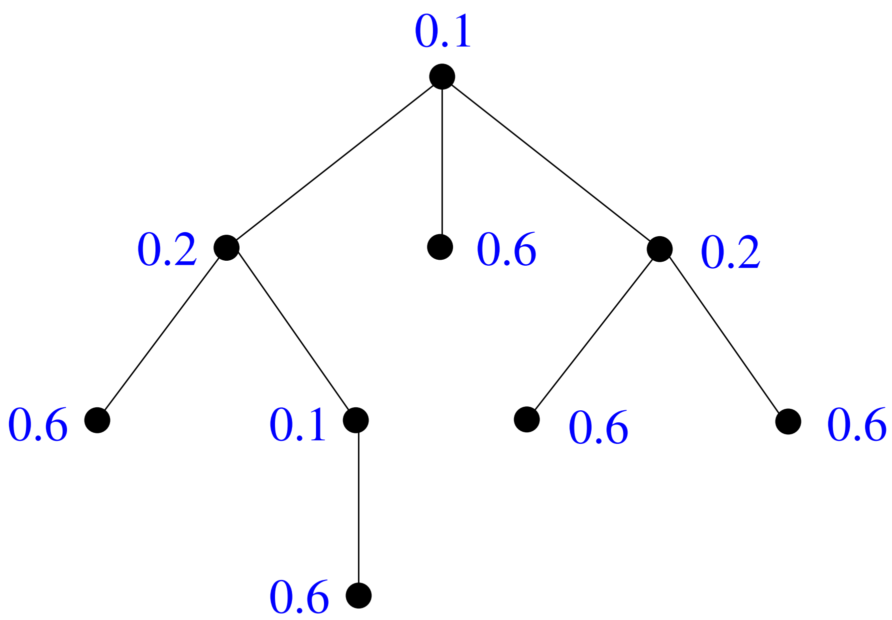
Example (only one type of thread):

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

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

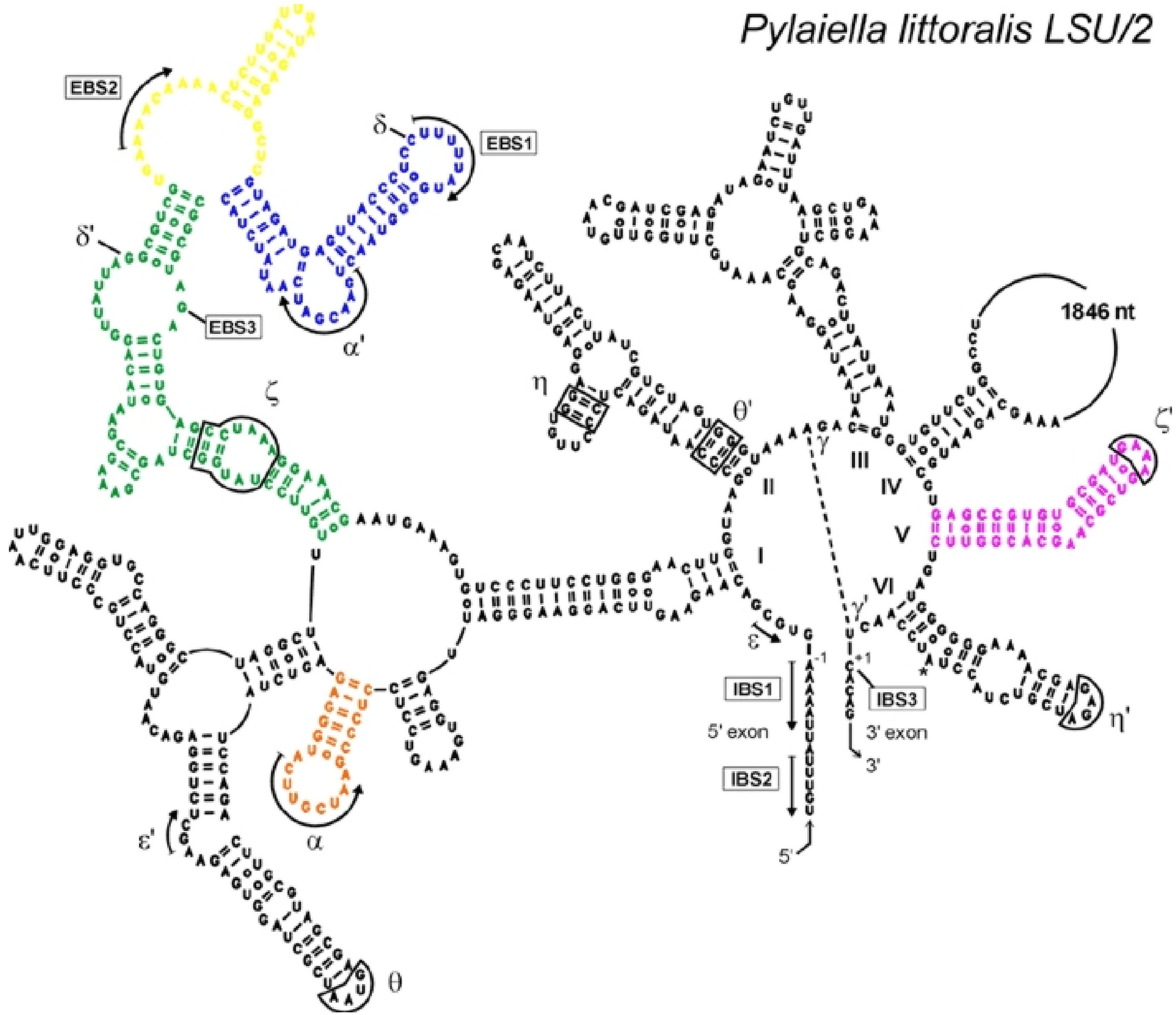
Lemma: The family trees with completion space $S^{op} = k$ “are” the derivation trees of Strahler number k.

Theorem [Bradzil *et al.* ’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

Secondary structure of RNA



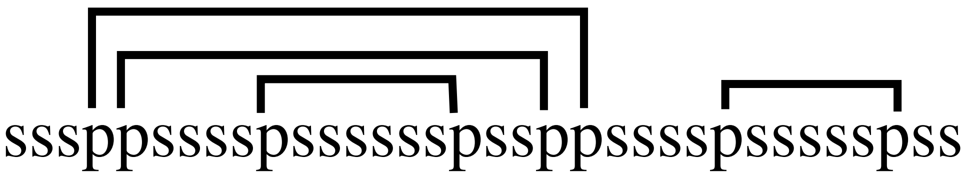
(image by Bassi, Costa, Michel; www.cgm.cnrs-gif.fr/michel/)

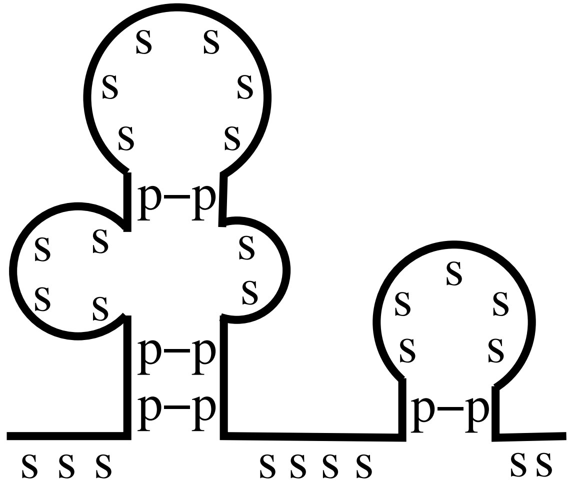
A stochastic context-free grammar

[Knudsen, Hein ’99]: Model the distribution of secondary structures as the derivation trees of the following stochastic unambiguous context-free grammar:

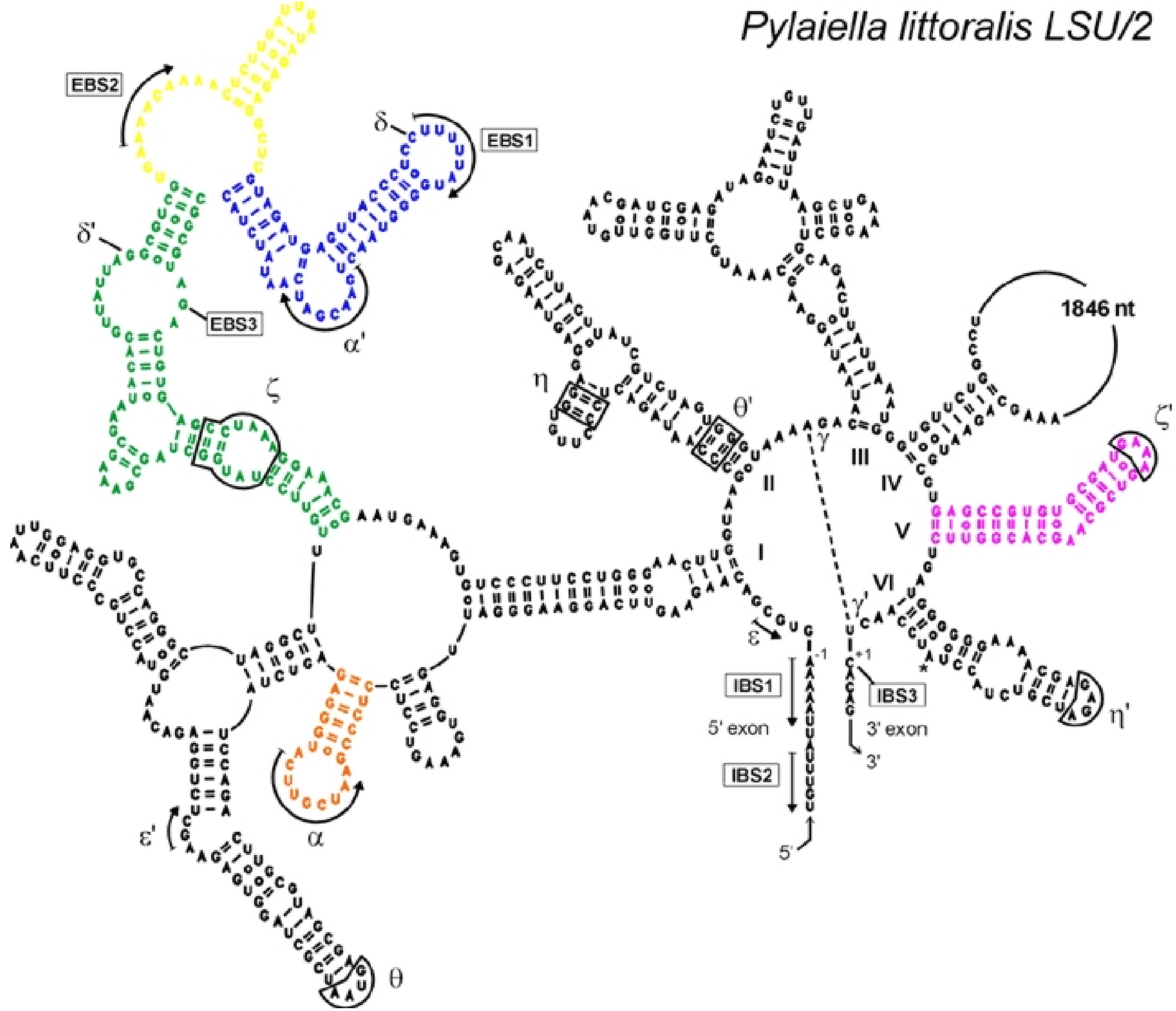
$$L \xrightarrow{0.869} CL \qquad L \xrightarrow{0.131} C$$
$$C \xrightarrow{0.895} s \qquad C \xrightarrow{0.105} pSp$$
$$S \xrightarrow{0.788} pSp \qquad S \xrightarrow{0.212} CL$$

Graphical interpretation:

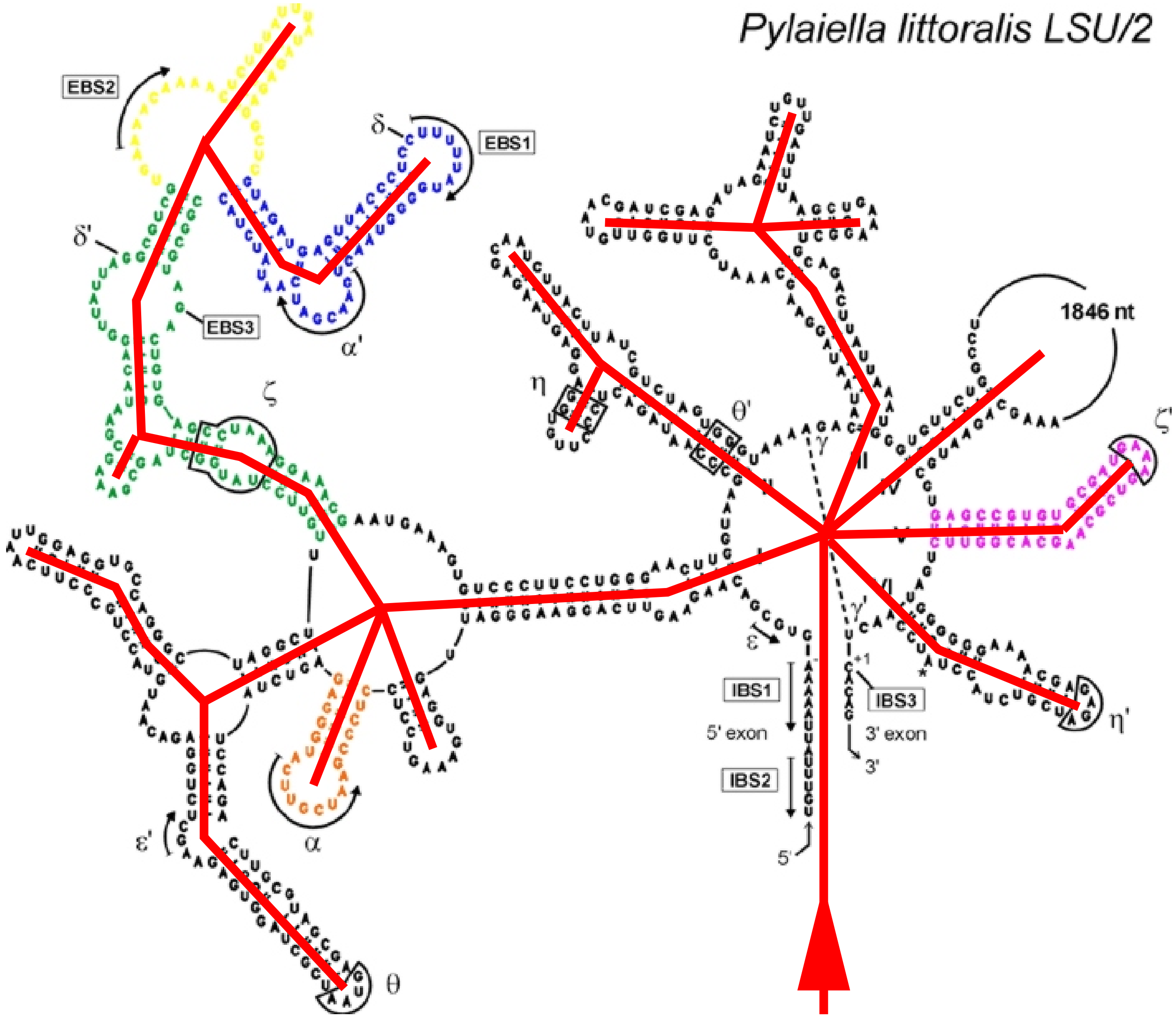




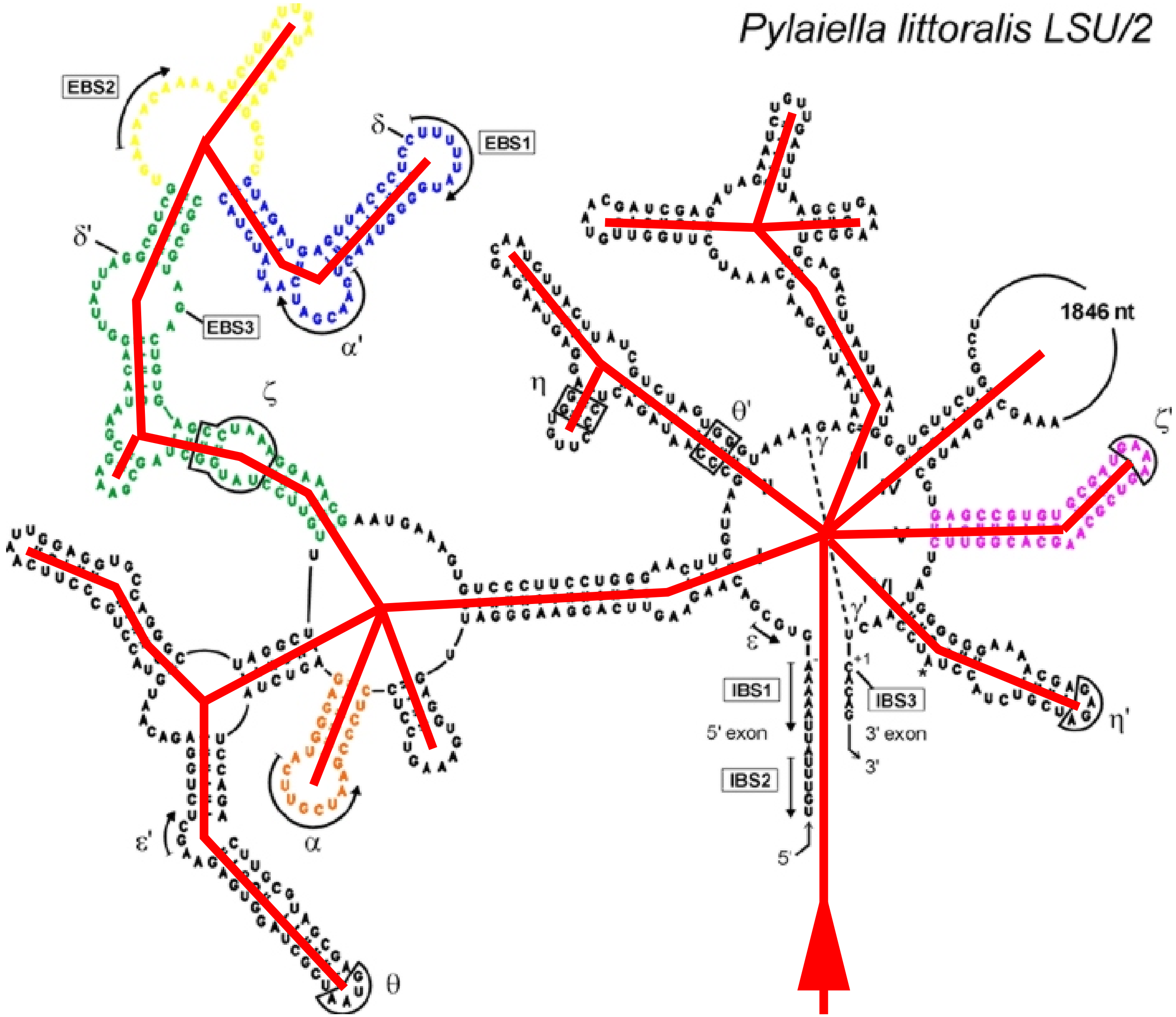
Visualizing the Strahler number of a word



Visualizing the Strahler number of a word



Visualizing the Strahler number of a word



Strahler number = maximal number of branching points from root to leaf

Grammar leads to two equation systems:

$$\begin{aligned} L &= C \cdot L + C \\ S &= p \cdot S \cdot p + C \cdot L \\ C &= s + p \cdot S \cdot p \end{aligned} \qquad \begin{aligned} \hat{L} &= 0.869 \cdot \hat{C} \cdot \hat{L} + 0.131 \cdot \hat{C} \\ \hat{S} &= 0.788 \cdot \hat{S} + 0.212 \cdot \hat{C} \cdot \hat{L} \\ \hat{C} &= 0.895 + 0.105 \cdot \hat{S} \end{aligned}$$

$$\begin{aligned} \nu_1(L) &= \text{der. of dim.} \leq 1 \\ \nu_2(L) &= \text{der. of dim.} \leq 2 \\ \nu_3(L) &= \text{der. of dim.} \leq 3 \\ \nu_4(L) &= \text{der. of dim.} \leq 4 \\ \nu_5(L) &= \text{der. of dim.} \leq 5 \\ \nu_6(L) &= \text{der. of dim.} \leq 6 \end{aligned} \qquad \begin{aligned} \hat{\nu}_1(L) &= 0.5585 \\ \hat{\nu}_2(L) &= 0.8050 \\ \hat{\nu}_3(L) &= 0.9250 \\ \hat{\nu}_4(L) &= 0.9789 \\ \hat{\nu}_5(L) &= 0.9972 \\ \hat{\nu}_6(L) &= 0.9999 \end{aligned}$$