Absorbing patterns in BST-like expression-trees

Pablo Rotondo

LIGM, Université Gustave Eiffel

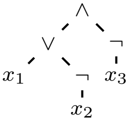
Work with

Florent Koechlin

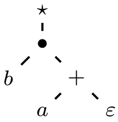
Séminaire Algo, LIGM, Univ. Gustave Eiffel,
Online, 12 January, 2021.

Introduction

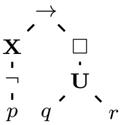
- Expression trees



$(x_1 \vee \neg x_2) \wedge \neg x_3$



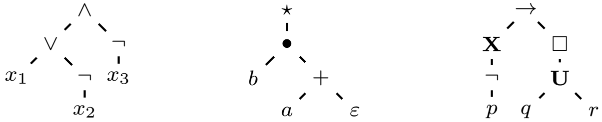
$(b \cdot (a + \varepsilon))^\star$



$\mathbf{X}(\neg p) \to \Box(q\mathbf{U}r)$

Introduction

- Expression trees

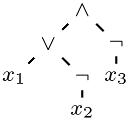


$(x_1 \vee \neg x_2) \wedge \neg x_3$ $\qquad$ $(b \cdot (a + \varepsilon))^\star$ $\qquad$ $\mathbf{X}(\neg p) \rightarrow \Box(q\mathbf{U}r)$

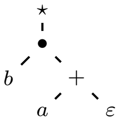
- Automated testing, benchmark testing
    - Correctness and performance of algorithms

Introduction

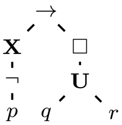
- Expression trees



$(x_1 \vee \neg x_2) \wedge \neg x_3$



$(b \cdot (a + \varepsilon))^\star$



$\mathbf{X}(\neg p) \rightarrow \Box(q\mathbf{U}r)$

- Automated testing, benchmark testing
  - Correctness and performance of algorithms
- Randomly generated input
  - Realistic distribution
  - Simple implementation, possibility of theoretical analysis.

BST-like trees: a natural construction algorithm

Idea to draw tree of size n:

– if drawn operator is binary, choose *size* of branches uniformly.

– build subtrees recursively and independently.

BST-like trees: a natural construction algorithm

Idea to draw tree of size n:

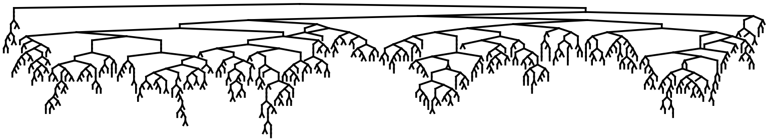
– if drawn operator is binary, choose *size* of branches uniformly.

– build subtrees recursively and independently.

Code used in tool `lbtt` (from `TCS`) to draw an LTL formula:

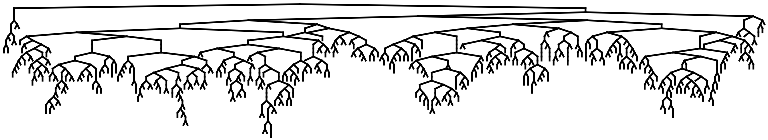
function RandomFormula(n):
if $n = 1$ **then**
  p := random symbol in $AP \cup \{\top, \bot\}$;
  return p;
else if $n = 2$ **then**
  op := random unary operator in $\{\neg, \mathbf{X}, \Box, \Diamond\}$;
  f := RandomFormula(1);
  return $op\ f$;
else
  op := random operator in $\{\neg, \mathbf{X}, \Box, \Diamond, \wedge, \vee, \rightarrow, \leftrightarrow, \mathbf{U}, \mathbf{R}\}$;
  if op *in* $\{\neg, \mathbf{X}, \Box, \Diamond\}$ **then**
    f := RandomFormula($n-1$);
    return $op\ f$;
  else
    x := uniform integer in $[1, n-2]$;
    f_1 := RandomFormula(x);
    f_2 := RandomFormula($n-x-1$);
    return $(f_1\ op\ f_2)$;

BST-like trees: distribution over unary-binary trees



BST-like tree distribution is not uniform.

BST-like trees: distribution over unary-binary trees



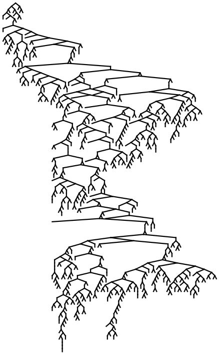
BST-like tree distribution is not uniform.

- Binary nodes ≈ balanced $\frac{n}{2}$–$\frac{n}{2}$.
  not for uniform trees

$$\mathbb{E}_n[\min(|T_L|, |T_R|)] \sim c_0\sqrt{n}\,.$$

- Expected height of different order

$$\Theta(\log n) \text{ vs } \Theta(\sqrt{n})\,.$$



Uniform and BST-like distributions

The uniform distribution:

- naturally maximizes entropy.
- can be sampled efficiently (Boltzmann,Recursive,Devroye's constrainted GW).
- is amenable to theoretical study (Analytic Combinatorics).

Uniform and BST-like distributions

The uniform distribution:

- naturally maximizes entropy.
- can be sampled efficiently
  (Boltzmann,Recursive,Devroye's constrainted GW).
- is amenable to theoretical study (Analytic Combinatorics).

The BST-like distribution:

- must be parametrized (prob. of operators).
- is easy to implement and very efficient.
- is often used in the automated checking of tools.

Uniform and BST-like distributions

The uniform distribution:

- naturally maximizes entropy.
- can be sampled efficiently
  (Boltzmann,Recursive,Devroye's constrainted GW).
- is amenable to theoretical study (Analytic Combinatorics).

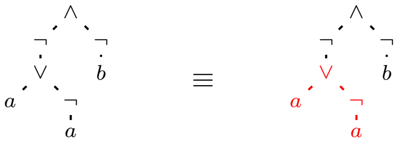
The BST-like distribution:

- must be parametrized (prob. of operators).
- is easy to implement and very efficient.
- is often used in the automated checking of tools.

Let us see what happens with uniform **expressions** first...

Semantic simplification

Number of nodes may not reflect true complexity of tree

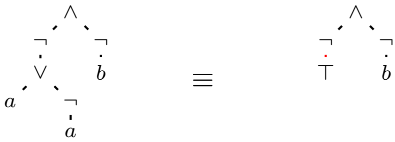


Semantic simplification

Number of nodes may not reflect true complexity of tree

Semantic simplification

Number of nodes may not reflect true complexity of tree



Semantic simplification

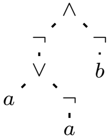
Number of nodes may not reflect true complexity of tree

Semantic simplification

Number of nodes may not reflect true complexity of tree

Semantic simplification

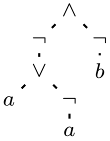
Number of nodes may not reflect true complexity of tree



$\equiv \quad \bot$

Semantic simplification

Number of nodes may not reflect true complexity of tree



$$\equiv \qquad \bot$$

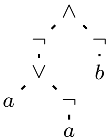
Universal result for uniform tree model:

Theorem (Koechlin,Nicaud,R,'20)

Expected size of reduction of uniform tree bounded, as size $\to \infty$.

Semantic simplification

Number of nodes may not reflect true complexity of tree



$\equiv$ $\perp$

Universal result for uniform tree model:

Theorem (Koechlin,Nicaud,R,'20)

Expected size of reduction of uniform tree bounded, as size $\to \infty$.

- Idea based on absorbing pattern $\mathcal{P}$, e.g., `false` $\wedge (\ldots) \equiv$ `false`,

$$\begin{array}{c}\circledast \\ / \ \backslash \\ \mathcal{P} \quad T\end{array} \rightsquigarrow \mathcal{P} \qquad \begin{array}{c}\circledast \\ / \ \backslash \\ T \quad \mathcal{P}\end{array} \rightsquigarrow \mathcal{P}$$

- For regular expressions on two letters, `constant bound` ≈ 77.8 .

In our work we

- draw an random BST-like tree expression of size n.
- study expected size of reduced expressions as $n \to \infty$.
- answer the question: do BST-like distributions present the same flaw as the uniform one?

[1] Left to right
$(p_\star, p_\bullet, p_+) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$,$(p_\star, p_\bullet, p_+) = (\frac{5}{29}, \frac{5}{29}, \frac{19}{29})$,$(p_\star, p_\bullet, p_+) = (\frac{1}{10}, \frac{1}{10}, \frac{8}{10})$

In our work we

- draw an random BST-like tree expression of size n.
- study expected size of reduced expressions as $n \to \infty$.
- answer the question: do BST-like distributions present the same flaw as the uniform one? $\Rightarrow$ we characterize 5 regimes

---

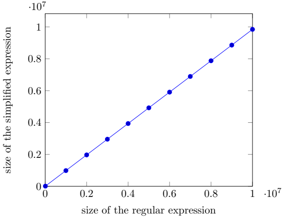
[1] Left to right $(p_\star, p_\bullet, p_+) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$,$(p_\star, p_\bullet, p_+) = (\frac{5}{29}, \frac{5}{29}, \frac{19}{29})$,$(p_\star, p_\bullet, p_+) = (\frac{1}{10}, \frac{1}{10}, \frac{8}{10})$

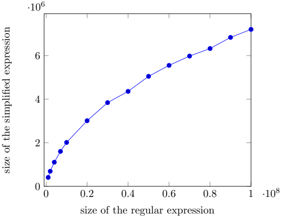
In our work we

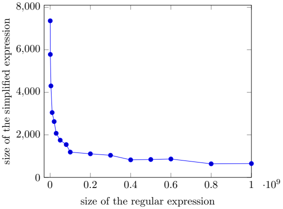
- draw an random BST-like tree expression of size n.
- study expected size of reduced expressions as $n \to \infty$.
- answer the question: do BST-like distributions present the same flaw as the uniform one? $\Rightarrow$ we characterize 5 regimes

Experimental expected size (10 000 samples)[1] on regular expressions $(+, \bullet, \star)$ on two letters a, b:

$\mathcal{P} = (a+b)^\star$ absorbing for union $\circledast = +$







[1] Left to right $(p_\star, p_\bullet, p_+) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$,$(p_\star, p_\bullet, p_+) = (\frac{5}{29}, \frac{5}{29}, \frac{19}{29})$,$(p_\star, p_\bullet, p_+) = (\frac{1}{10}, \frac{1}{10}, \frac{8}{10})$

Plan of the talk

1. Model: BST-like trees and absorbing patterns

2. Main Theorem and outline of the proof

3. Conclusions

Expression trees and BST-like model

Let $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2$ be non-empty finite sets of labels:
the *leaves*, *unary*, and *binary* operators respectively.

Definition

The family $\mathcal{E}(\mathcal{A})$ of expression trees over $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ is defined inductively from the leaves.

Expression trees and BST-like model

Let $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2$ be non-empty finite sets of labels:

the *leaves*, *unary*, and *binary* operators respectively.

Definition

The family $\mathcal{E}(\mathcal{A})$ of expression trees over $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ is defined inductively from the leaves.

BST-like procedure. Consider probabilities over leaves $\mathcal{A}_0$ and operators $\mathcal{A}_{\mathtt{ops}} = \mathcal{A}_1 \cup \mathcal{A}_2$, write $(p_a)_{\mathcal{A}_0}$ and $(p_{op})_{\mathcal{A}_{\mathtt{ops}}}$.

Expression trees and BST-like model

Let $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2$ be non-empty finite sets of labels:
the *leaves*, *unary*, and *binary* operators respectively.

Definition

The family $\mathcal{E}(\mathcal{A})$ of expression trees over $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ is defined inductively from the leaves.

BST-like procedure. Consider probabilities over leaves $\mathcal{A}_0$ and operators $\mathcal{A}_{\mathsf{ops}} = \mathcal{A}_1 \cup \mathcal{A}_2$, write $(p_a)_{\mathcal{A}_0}$ and $(p_{op})_{\mathcal{A}_{\mathsf{ops}}}$.

If $n \geq 3$: throw an operator according to $(p_{op})_{op \in \mathcal{A}_{\mathsf{ops}}}$.

Expression trees and BST-like model

Let $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2$ be non-empty finite sets of labels:
the *leaves*, *unary*, and *binary* operators respectively.

Definition

The family $\mathcal{E}(\mathcal{A})$ of expression trees over $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ is defined inductively from the leaves.

BST-like procedure. Consider probabilities over leaves $\mathcal{A}_0$ and operators $\mathcal{A}_{\mathsf{ops}} = \mathcal{A}_1 \cup \mathcal{A}_2$, write $(p_a)_{\mathcal{A}_0}$ and $(p_{op})_{\mathcal{A}_{\mathsf{ops}}}$.

If $n \geq 3$: throw an operator according to $(p_{op})_{op \in \mathcal{A}_{\mathsf{ops}}}$.

– If it is binary $op \in \mathcal{A}_2$, pick $k \in \{1, \dots, n-2\}$ uniformly, produce indep. $T_L = \texttt{BST}(k)$ and $T_R = \texttt{BST}(n-k-1)$.

return $\overset{op}{\underset{T_L \;\; T_R}{/\backslash}}$.

Expression trees and BST-like model

Let $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2$ be non-empty finite sets of labels:

the *leaves*, *unary*, and *binary* operators respectively.

Definition

The family $\mathcal{E}(\mathcal{A})$ of expression trees over $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ is defined inductively from the leaves.

BST-like procedure. Consider probabilities over leaves $\mathcal{A}_0$ and operators $\mathcal{A}_{\text{ops}} = \mathcal{A}_1 \cup \mathcal{A}_2$, write $(p_a)_{\mathcal{A}_0}$ and $(p_{op})_{\mathcal{A}_{\text{ops}}}$.

If $n \geq 3$: throw an operator according to $(p_{op})_{op \in \mathcal{A}_{\text{ops}}}$.

– If it is binary $op \in \mathcal{A}_2$, pick $k \in \{1, \ldots, n-2\}$ uniformly, produce indep. $T_L = \mathtt{BST}(k)$ and $T_R = \mathtt{BST}(n-k-1)$.

return $\overset{op}{\overset{/\backslash}{T_L \;\; T_R}}$.

– If it is unary $op \in \mathcal{A}_1$, produce indep. $T = \mathtt{BST}(n-1)$

return $\overset{op}{\overset{|}{T}}$.

Expression trees and BST-like model

Let $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2$ be non-empty finite sets of labels:
the *leaves*, *unary*, and *binary* operators respectively.

Definition

The family $\mathcal{E}(\mathcal{A})$ of expression trees over $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ is defined inductively from the leaves.

BST-like procedure. Consider probabilities over leaves $\mathcal{A}_0$ and operators $\mathcal{A}_{\text{ops}} = \mathcal{A}_1 \cup \mathcal{A}_2$, write $(p_a)_{\mathcal{A}_0}$ and $(p_{op})_{\mathcal{A}_{\text{ops}}}$.

If $n \geq 3$: throw an operator according to $(p_{op})_{op \in \mathcal{A}_{\text{ops}}}$.

– If it is binary $op \in \mathcal{A}_2$, pick $k \in \{1, \ldots, n-2\}$ uniformly, produce indep. $T_L = \texttt{BST}(k)$ and $T_R = \texttt{BST}(n-k-1)$.

return $\begin{array}{c} op \\ / \backslash \\ T_L \quad T_R \end{array}$.

– If it is unary $op \in \mathcal{A}_1$, produce indep. $T = \texttt{BST}(n-1)$

return $\begin{array}{c} op \\ | \\ T \end{array}$.

If $n = 1$: return a leaf a according to $(p_a)_{a \in \mathcal{A}_0}$.

Expression trees and BST-like model

Let $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2$ be non-empty finite sets of labels:
the *leaves*, *unary*, and *binary* operators respectively.

Definition

The family $\mathcal{E}(\mathcal{A})$ of expression trees over $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ is defined inductively from the leaves.

BST-like procedure. Consider probabilities over leaves $\mathcal{A}_0$ and operators $\mathcal{A}_{\mathsf{ops}} = \mathcal{A}_1 \cup \mathcal{A}_2$, write $(p_a)_{\mathcal{A}_0}$ and $(p_{op})_{\mathcal{A}_{\mathsf{ops}}}$.

If $n \geq 3$: throw an operator according to $(p_{op})_{op \in \mathcal{A}_{\mathsf{ops}}}$.

– If it is binary $op \in \mathcal{A}_2$, pick $k \in \{1, \ldots, n-2\}$ uniformly, produce indep. $T_L = \mathtt{BST}(k)$ and $T_R = \mathtt{BST}(n-k-1)$.

return $\begin{array}{c} op \\ / \backslash \\ T_L \;\; T_R \end{array}$.

– If it is unary $op \in \mathcal{A}_1$, produce indep. $T = \mathtt{BST}(n-1)$

return $\begin{array}{c} op \\ | \\ T \end{array}$.

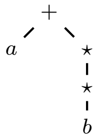
If $n = 1$: return a leaf a according to $(p_a)_{a \in \mathcal{A}_0}$.

If $n = 2$: pick op. of arity 1...

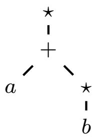
Example: BST-like distribution

Consider the regular expressions $(+, \bullet, \star)$ on two letters a, b

(i)



(ii)

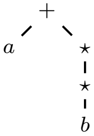
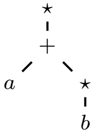


- The expression tree (i) is drawn with probability

$$p_+ \frac{1}{3} p_a p_\star p_b \, .$$

Example: BST-like distribution

Consider the regular expressions $(+, \bullet, \star)$ on two letters a, b

(i)  (ii) 

- The expression tree (i) is drawn with probability

$$p_+ \frac{1}{3} p_a p_\star p_b \, .$$

- The expression tree (ii) is drawn with probability

$$p_\star p_+ \frac{1}{2} p_a p_b \, .$$

Example: BST-like distribution

Consider the regular expressions $(+, \bullet, \star)$ on two letters a, b

- The expression tree (i) is drawn with probability

$$p_+ \frac{1}{3} p_a p_\star p_b \,.$$

- The expression tree (ii) is drawn with probability

$$p_\star p_+ \frac{1}{2} p_a p_b \,.$$

- Distribution not uniform for any choice of parameters.

Absorbing patters: simplifying the trees

Definition (Simplification, absorbing pattern)

Let $\mathcal{E}(\mathcal{A})$ be the family of tree expression over the family of labels (operators) $\mathcal{A} = (\mathcal{A}_i)$, consider

- an "operation" $\circledast \in \mathcal{A}_a$ with arity $a = 2$,
- an expression tree $\mathcal{P} \in \mathcal{E}(\mathcal{A})$.

Absorbing patters: simplifying the trees

Definition (Simplification, absorbing pattern)

Let $\mathcal{E}(\mathcal{A})$ be the family of tree expression over the family of labels (operators) $\mathcal{A} = (\mathcal{A}_i)$, consider

- an "operation" $\circledast \in \mathcal{A}_a$ with arity $a = 2$,
- an expression tree $\mathcal{P} \in \mathcal{E}(\mathcal{A})$.

We simplify by applying bottom-up the rule:

$$\begin{array}{c} \circledast \\ / \quad \backslash \\ C_1 \quad\quad C_2 \end{array} \rightsquigarrow \mathcal{P}\,, \text{ whenever } C_i = \mathcal{P} \text{ for some } i \in \{1, 2\}.$$

Absorbing patters: simplifying the trees

Definition (Simplification, absorbing pattern)

Let $\mathcal{E}(\mathcal{A})$ be the family of tree expression over the family of labels (operators) $\mathcal{A} = (\mathcal{A}_i)$, consider

- an "operation" $\circledast \in \mathcal{A}_a$ with arity $a = 2$,
- an expression tree $\mathcal{P} \in \mathcal{E}(\mathcal{A})$.

We simplify by applying bottom-up the rule:

$$\begin{array}{c} \circledast \\ / \quad \backslash \\ C_1 \quad\quad C_2 \end{array} \rightsquigarrow \mathcal{P}\,, \text{ whenever } C_i = \mathcal{P} \text{ for some } i \in \{1, 2\}.$$

⇒ We are interested in the *size* of the trees after simplification.

Absorbing patters: simplifying the trees

Definition (Simplification, absorbing pattern)

Let $\mathcal{E}(\mathcal{A})$ be the family of tree expression over the family of labels (operators) $\mathcal{A} = (\mathcal{A}_i)$, consider

- an "operation" $\circledast \in \mathcal{A}_a$ with arity $a = 2$,
- an expression tree $\mathcal{P} \in \mathcal{E}(\mathcal{A})$.

We simplify by applying bottom-up the rule:

$$\begin{array}{c} \circledast \\ / \quad \backslash \\ C_1 \quad\; C_2 \end{array} \;\rightsquigarrow \mathcal{P}\,,\ \text{whenever } C_i = \mathcal{P} \text{ for some } i \in \{1, 2\}.$$

⇒ We are interested in the *size* of the trees after simplification.

Denote by $\sigma(T) = \sigma(T, \mathcal{P}, \circledast)$ the simplification of $T \in \mathcal{E}(\mathcal{A})$.

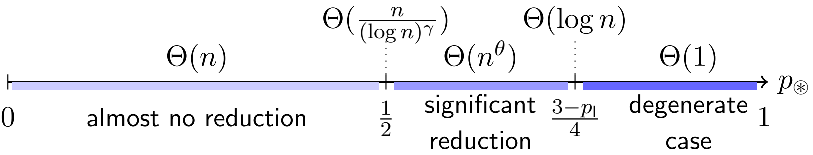
Theorem. *Consider a family of expression trees defined from unary and binary operators with an absorbing pattern* $\mathcal{P}$ *for an operator* $\circledast$ *of arity* 2.

Take the simplification consisting in inductively changing a $\circledast$*-node by* $\mathcal{P}$ *whenever one of its children simplifies to* $\mathcal{P}$.

Theorem. *Consider a family of expression trees defined from unary and binary operators with an absorbing pattern $\mathcal{P}$ for an operator $\circledast$ of arity 2.*

Take the simplification consisting in inductively changing a $\circledast$-node by $\mathcal{P}$ whenever one of its children simplifies to $\mathcal{P}$.

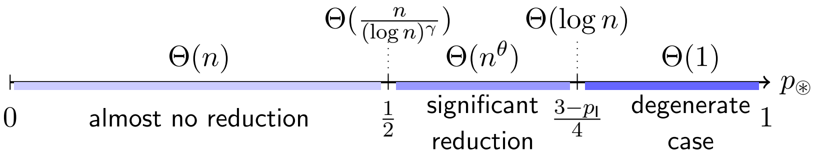
Then the expected size of the simplification of a random BST-like tree has an asymptotic behaviour given by the following cases, depending on the probability $p_\circledast$ of the absorbing operator:



Theorem. *Consider a family of expression trees defined from unary and binary operators with an* *absorbing pattern* $\mathcal{P}$ *for an operator* $\circledast$ *of arity* 2.

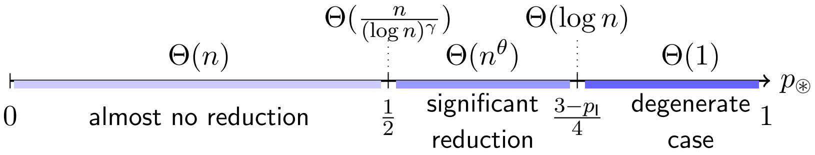
Take the *simplification* *consisting in inductively changing a* $\circledast$*-node by* $\mathcal{P}$ *whenever one of its children simplifies to* $\mathcal{P}$.

Then the *expected size* *of the* *simplification* *of a random BST-like tree has an asymptotic behaviour given by the following cases, depending on the probability* $p_\circledast$ *of the absorbing operator:*

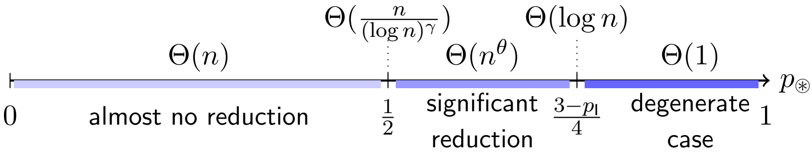


- Probability $p_\circledast$ of $\circledast$, and p_l of picking unary operator.
- Two critical points $p_\circledast = 1/2$ and $p_\circledast = (3 - p_\mathsf{l})/4$
- Regimes from no reduction $\Theta(n)$ to complete reduction $\Theta(1)$

The main regimes experimentally



The main regimes experimentally



Experimental plots (10 000 samples)for regular expressions on two letters a, b: $\mathcal{P} = (a+b)^\star$ absorbing for union $\circledast = +$

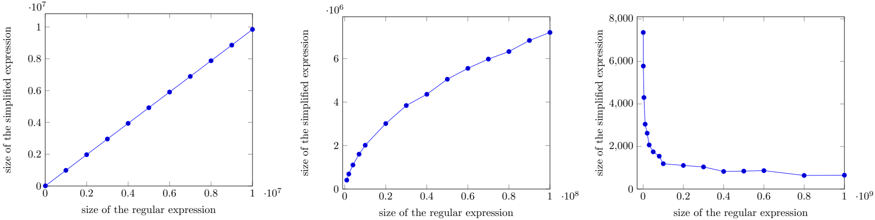


Figure: Left to right: linear ($p_+ = p_\star = p_. = \frac{1}{3}$), sublinear ($p_+ = \frac{19}{29}$, $p_\star = p_. = \frac{5}{29}$) and constant ($p_+ = \frac{8}{10}$, $p_\star = p_. = \frac{1}{10}$).

Scheme for the proof

We employ Analytic Combinatorics to study the expectation,

- Ordinary generating function

$$E(z) := \sum_{n=0}^{\infty} e_n z^n \,,$$

encodes sequence $e_n := \mathbb{E}_n[\sigma(T)]$.

Scheme for the proof

We employ Analytic Combinatorics to study the expectation,

- Ordinary generating function

$$E(z) := \sum_{n=0}^{\infty} e_n z^n \,,$$

  encodes sequence $e_n := \mathbb{E}_n[\sigma(T)]$.

- **Symbolic Step.** We find a formal equation describing $E(z)$.

  Here this will be an *ordinary differential equation*

$$E'(z) = B(z) + C(z) \cdot E(z) \,.$$

Scheme for the proof

We employ Analytic Combinatorics to study the expectation,

- Ordinary generating function

$$E(z) := \sum_{n=0}^{\infty} e_n z^n \,,$$

encodes sequence $e_n := \mathbb{E}_n[\sigma(T)]$.

- **Symbolic Step.** We find a formal equation describing $E(z)$.

  Here this will be an *ordinary differential equation*

$$E'(z) = B(z) + C(z) \cdot E(z)\,.$$

- **Analytic Step.** We look at $E(z)$ over the complex $z \in \mathbb{C}$.

  A *Transfer Theorem* links the behaviour of $E(z)$ at its dominant singularity to asymptotics of $e_n \Rightarrow$ Study singularities

$$E(z) \sim_{z\to 1} \lambda(1-z)^{-\alpha} \Longrightarrow e_n \sim \lambda n^{\alpha-1}/\Gamma(\alpha)$$

Symbolic step: recurrence

We consider a fundamental sequence

$$\gamma_n := \Pr_n \{\sigma(T) = \mathcal{P}\} \ ,$$

of the probabilities of *full* reduction.

Symbolic step: recurrence

We consider a fundamental sequence

$$\gamma_n := \Pr_n \{\sigma(T) = \mathcal{P}\} \ ,$$

of the probabilities of *full* reduction.

Recurrence for expected values

The recurrence for e_n involves γ_n,

$$e_{n+1} = 1+(s-1)\gamma_{n+1}\mathbf{1}_{n+1\neq s} + p_{\mathsf{I}} e_n + \frac{2p_{\mathsf{II}}}{n-1}\sum_{j=1}^{n-1} e_j + \frac{2p_{\circledast}}{n-1}\sum_{j=1}^{n-1}(e_j - s\gamma_j)(1-\gamma_{n-j})\,,$$

here $p_{\mathsf{II}} := 1 - p_{\mathsf{I}} - p_{\circledast}$ and $s = |\mathcal{P}|$.

Symbolic step: recurrence

We consider a fundamental sequence

$$\gamma_n := \Pr_n \{\sigma(T) = \mathcal{P}\} ,$$

of the probabilities of *full* reduction.

Recurrence for expected values

The recurrence for e_n involves γ_n,

$$e_{n+1} = 1 + (s-1)\gamma_{n+1}\mathbf{1}_{n+1\neq s} + p_{\mathsf{I}} e_n + \frac{2p_{\mathsf{II}}}{n-1}\sum_{j=1}^{n-1} e_j + \frac{2p_{\circledast}}{n-1}\sum_{j=1}^{n-1}(e_j - s\gamma_j)(1-\gamma_{n-j}) ,$$

here $p_{\mathsf{II}} := 1 - p_{\mathsf{I}} - p_{\circledast}$ and $s = |\mathcal{P}|$.

Proof.

Consider $\texttt{root} = \circledast$. Then $\sigma(T) = \mathcal{P}$ with proba γ_{n+1}. Else subtract the cases of reduced branches $\sigma(T_L) = \mathcal{P}$ or $\sigma(T_R) = \mathcal{P}$. □

Symbolic step: differential equation

Recurrence yields first order differential equation

$$E'(z) = F(z, A(z)) + \frac{1}{1-p_{\mathsf{I}}z}\left(\frac{2}{z} - p_{\mathsf{I}} + 2\,(1-p_{\mathsf{I}})\,\frac{z}{1-z} - 2p_{\circledast}A(z)\right) \cdot E(z)\,,$$

in terms of $A(z) = \sum_n \gamma_n z^n$.

Symbolic step: differential equation

Recurrence yields first order differential equation

$$E'(z) = F(z, A(z)) + \frac{1}{1-p_{\mathsf{I}} z}\left(\frac{2}{z} - p_{\mathsf{I}} + 2\,(1-p_{\mathsf{I}})\,\frac{z}{1-z} - 2p_{\circledast} A(z)\right) \cdot E(z)\,,$$

in terms of $A(z) = \sum_n \gamma_n z^n$.

Proof.

Differentiating we have $E'(z) = \sum (n+1) e_n z^n$. □

Symbolic step: differential equation

Recurrence yields first order differential equation

$$E'(z) = F(z, A(z)) + \tfrac{1}{1-p_{\mathsf{I}} z}\left(\tfrac{2}{z} - p_{\mathsf{I}} + 2\,(1-p_{\mathsf{I}})\,\frac{z}{1-z} - 2p_{\circledast} A(z)\right)\cdot E(z)\,,$$

in terms of $A(z) = \sum_n \gamma_n z^n$.

Proof.

Differentiating we have $E'(z) = \sum (n+1) e_n z^n$. □

First order differential equations can be solved explicitly

Proposition

The equation $U'(z) = f(z) + g(z)U(z)$ where f, g are analytic functions on Ω has a unique solution analytic on Ω, satisfying $U(0) = u_0$,

$$U(z) = \exp\left(\int_0^z g(\zeta)d\zeta\right)\left(u_0 + \int_0^z f(\zeta)\exp\left(-\int_0^\zeta g(w)dw\right)d\zeta\right).$$

Symbolic step: differential equation

Recurrence yields first order differential equation

$$E'(z) = F(z, A(z)) + \frac{1}{1-p_{\mathsf{l}} z}\left(\frac{2}{z} - p_{\mathsf{l}} + 2\,(1-p_{\mathsf{l}})\,\frac{z}{1-z} - 2p_{\circledast} A(z)\right) \cdot E(z)\,,$$

in terms of $A(z) = \sum_n \gamma_n z^n$.

Proof.

Differentiating we have $E'(z) = \sum (n+1) e_n z^n$. $\square$

First order differential equations can be solved explicitly

Proposition

The equation $U'(z) = f(z) + g(z)U(z)$ where f, g are analytic functions on Ω has a unique solution analytic on Ω, satisfying $U(0) = u_0$,

$$U(z) = \exp\left(\int_0^z g(\zeta) d\zeta\right)\left(u_0 + \int_0^z f(\zeta) \exp\left(-\int_0^\zeta g(w) dw\right) d\zeta\right).$$

Our coefficients depend on z and the unkown generating function $A(z)$.

Analytic step: the singularity $z = 1$

The generating functions $A(z)$ and $E(z)$

- are analytic for $|z| < 1$, as the series converge absolutely.

Analytic step: the singularity $z = 1$

The generating functions $A(z)$ and $E(z)$

- are analytic for $|z| < 1$, as the series converge absolutely.
- have $z = 1$ as a dominant singularity: $\rho = 1$ radius of convergence.

Analytic step: the singularity $z = 1$

The generating functions $A(z)$ and $E(z)$

- are analytic for $|z| < 1$, as the series converge absolutely.
- have $z = 1$ as a dominant singularity: $\rho = 1$ radius of convergence.

Solution of ODE gives asymptotics

$$E(z) \sim \frac{c}{(1-z)^2} \left(2 + \int_0^z F(w, A(w)) I(w) dw \right) (I(z))^{-1}, \quad z \to 1,$$

where $I(z) := \exp\left(2p_{\circledast} \int_0^z \frac{A(w)}{1-p_{\mathsf{I}} w} dw \right)$.

Analytic step: the singularity $z = 1$

The generating functions $A(z)$ and $E(z)$

- are analytic for $|z| < 1$, as the series converge absolutely.
- have $z = 1$ as a dominant singularity: $\rho = 1$ radius of convergence.

Solution of ODE gives asymptotics

$$E(z) \sim \frac{c}{(1-z)^2} \left(2 + \int_0^z F(w, A(w)) I(w) dw \right) (I(z))^{-1}, \quad z \to 1,$$

where $I(z) := \exp\left(2p_{\circledast} \int_0^z \frac{A(w)}{1-p_{\shortmid} w} dw \right)$.

To apply the Transfer Theorem and complete the proof:

- we require precise asymptotics for $A(z)$ at $z = 1$,
- we show that $A(z)$ and $E(z)$ are analytic over $\Omega = \mathbb{C} \setminus [1, \infty)$.

Fully reducible trees: probabilities

We study the generating function $A(z) = \sum \gamma_n z^n$

Proposition

The probabilities $\gamma_n = \Pr_n \{\sigma(T) = \mathcal{P}\}$ satisfy, for $n \geq |\mathcal{P}|$,

$$\gamma_{n+1} = \frac{p_{\circledast}}{n-1} \sum_{k=1}^{n-1} (\gamma_k + \gamma_{n-k} - \gamma_k \gamma_{n-k}) \, .$$

Fully reducible trees: probabilities

We study the generating function $A(z) = \sum \gamma_n z^n$

Proposition

The probabilities $\gamma_n = \Pr_n \{\sigma(T) = \mathcal{P}\}$ satisfy, for $n \geq |\mathcal{P}|$,

$$\gamma_{n+1} = \frac{p_{\circledast}}{n-1} \sum_{k=1}^{n-1} (\gamma_k + \gamma_{n-k} - \gamma_k \gamma_{n-k}) \,.$$

Recurrence translates into Riccati differential equation

$$A'(z) = (s-2)\gamma_s z^{s-1} + \left(\frac{2}{z} + 2p_{\circledast} \frac{z}{1-z} \right) A(z) - p_{\circledast} \cdot (A(z))^2 \,,$$

where $s = |\mathcal{P}|$.

Analytic step: linearization of Riccati

Considering $v(z)$ such that $p_{\circledast} A(z) = v'(z)/v(z)$,

Riccati equation becomes linear

$$v''(z) = p_{\circledast} \cdot (s-2)\gamma_s z^{s-1} v(z) + \left(\frac{2}{z} + 2p_{\circledast} \frac{z}{1-z} \right) v'(z) \,.$$

Analytic step: linearization of Riccati

Considering $v(z)$ such that $p_{\circledast} A(z) = v'(z)/v(z)$,

Riccati equation becomes linear

$$v''(z) = p_{\circledast} \cdot (s-2)\gamma_s z^{s-1} v(z) + \left(\frac{2}{z} + 2p_{\circledast}\frac{z}{1-z}\right) v'(z)\,.$$

For linear ODEs:

- domain of analyticity well-understood.
- Frobenius method characterizes behaviour at singularities.

Analytic step: linearization of Riccati

Considering $v(z)$ such that $p_{\circledast}A(z) = v'(z)/v(z)$,

Riccati equation becomes linear

$$v''(z) = p_{\circledast} \cdot (s-2)\gamma_s z^{s-1} v(z) + \left(\frac{2}{z} + 2p_{\circledast}\frac{z}{1-z}\right) v'(z)\,.$$

For linear ODEs:

- domain of analyticity well-understood.
- Frobenius method characterizes behaviour at singularities.

Proposition

The generating function $A(z)$ satisfies, $z \to 1$

- For $p_{\circledast} > \frac{1}{2}$, $A(z) = \frac{\gamma_\infty}{1-z} + O((1-z)^{2p_{\circledast}-2})$,
- For $p_{\circledast} = \frac{1}{2}$, $A(z) = \frac{2}{1-z}\left(\log\left(\frac{1}{1-z}\right)\right)^{-1}\left(1 + O\left(\log\left(\frac{1}{1-z}\right)^{-1}\right)\right)$
- For $p_{\circledast} < \frac{1}{2}$, $A(z) \sim \frac{D}{(1-z)^{2p_{\circledast}}}$,

where $\gamma_\infty := (2p_{\circledast}-1)/p_{\circledast}$ and $D > 0$ is a constant.

Probability of full reduction

Theorem

The probability γ_n of being fully reducible tends to the constant $\gamma_\infty := (2p_\circledast - 1)/p_\circledast$ for $p_\circledast > \frac{1}{2}$ and to zero otherwise.

Moreover,

- *for $p_\circledast = \frac{1}{2}$ we have $\gamma_n \sim \frac{2}{\log n}$,*
- *for $p_\circledast < \frac{1}{2}$, $\gamma_n \sim D \cdot n^{2p_\circledast - 1}$.*

Probability of full reduction

Theorem

The probability γ_n of being fully reducible tends to the constant $\gamma_\infty := (2p_\circledast - 1)/p_\circledast$ for $p_\circledast > \frac{1}{2}$ and to zero otherwise.

Moreover,

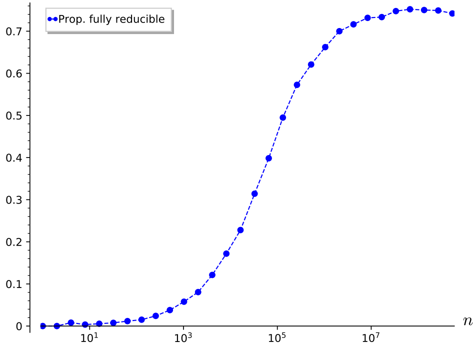
- *for $p_\circledast = \frac{1}{2}$ we have $\gamma_n \sim \frac{2}{\log n}$,*
- *for $p_\circledast < \frac{1}{2}$, $\gamma_n \sim D \cdot n^{2p_\circledast - 1}$.*

Experimental plot: regular expressions on two letters with

$$(p_+, p_\bullet, p_\star) = \left(\tfrac{8}{10}, \tfrac{1}{10}, \tfrac{1}{10}\right).$$

Then

$$\lim_{n\to\infty} \gamma_n = 3/4\,.$$



Proof principles: analytic step – Frobenius method

Recall. From the Riccati equation, we have a 2nd order linear ODE

$$v''(z) = p_{\circledast} \cdot (s-2)\gamma_s z^{s-1} v(z) + \left(\frac{2}{z} + 2p_{\circledast}\frac{z}{1-z}\right) v'(z)\,,$$

and we want the behavior around $z = 1$ (false pole at $z = 0$).

Proof principles: analytic step – Frobenius method

Recall. From the Riccati equation, we have a 2nd order linear ODE

$$v''(z) = p_{\circledast} \cdot (s-2)\gamma_s z^{s-1} v(z) + \left(\frac{2}{z} + 2p_{\circledast}\frac{z}{1-z}\right) v'(z)\,,$$

and we want the behavior around $z = 1$ (false pole at $z = 0$).

Frobenius Method [Fuchs' Theorem]

Let $v''(z) = \frac{q(z)}{(1-z)^2} v(z) + \frac{p(z)}{1-z} v'(z)$ with $q(z)$ and $p(z)$ analytic at $z = 1$.

Proof principles: analytic step – Frobenius method

Recall. From the Riccati equation, we have a 2nd order linear ODE

$$v''(z) = p_{\circledast} \cdot (s-2)\gamma_s z^{s-1} v(z) + \left(\frac{2}{z} + 2p_{\circledast} \frac{z}{1-z} \right) v'(z) \,,$$

and we want the behavior around $z = 1$ (false pole at $z = 0$).

Frobenius Method [Fuchs' Theorem]

Let $v''(z) = \frac{q(z)}{(1-z)^2} v(z) + \frac{p(z)}{1-z} v'(z)$ with $q(z)$ and $p(z)$ analytic at $z = 1$.

The ansatz $(1-z)^{\alpha} h_{\alpha}(z)$, with $h_{\alpha}(z)$ analytic at 1, $h_{\alpha}(1) = 1$,

Proof principles: analytic step – Frobenius method

Recall. From the Riccati equation, we have a 2nd order linear ODE

$$v''(z) = p_{\circledast} \cdot (s-2)\gamma_s z^{s-1} v(z) + \left(\frac{2}{z} + 2p_{\circledast}\frac{z}{1-z}\right) v'(z)\,,$$

and we want the behavior around $z = 1$ (false pole at $z = 0$).

Frobenius Method [Fuchs' Theorem]

Let $v''(z) = \frac{q(z)}{(1-z)^2} v(z) + \frac{p(z)}{1-z} v'(z)$ with $q(z)$ and $p(z)$ analytic at $z = 1$.

The ansatz $(1-z)^{\alpha} h_{\alpha}(z)$, with $h_{\alpha}(z)$ analytic at 1, $h_{\alpha}(1) = 1$, yields a solution when α is a root of the *indicial equation*

$$X^2 = (p(1)+1)X + q(1).$$

Proof principles: analytic step – Frobenius method

Recall. From the Riccati equation, we have a 2nd order linear ODE

$$v''(z) = p_{\circledast} \cdot (s-2)\gamma_s z^{s-1} v(z) + \left(\frac{2}{z} + 2p_{\circledast}\frac{z}{1-z}\right) v'(z)\,,$$

and we want the behavior around $z = 1$ (false pole at $z = 0$).

Frobenius Method [Fuchs' Theorem]

Let $v''(z) = \frac{q(z)}{(1-z)^2} v(z) + \frac{p(z)}{1-z} v'(z)$ with $q(z)$ and $p(z)$ analytic at $z = 1$.

The ansatz $(1-z)^{\alpha} h_{\alpha}(z)$, with $h_{\alpha}(z)$ analytic at 1, $h_{\alpha}(1) = 1$, yields a solution when α is a root of the *indicial equation*

$$X^2 = (p(1)+1)X + q(1).$$

Let α_1 and α_2 be the roots of the indicial equation.
– If $\alpha_1 - \alpha_2 \notin \mathbb{Z}$, then the two solutions are linearly independent.

Proof principles: analytic step – Frobenius method

Recall. From the Riccati equation, we have a 2nd order linear ODE

$$v''(z) = p_\circledast \cdot (s-2)\gamma_s z^{s-1} v(z) + \left(\frac{2}{z} + 2p_\circledast \frac{z}{1-z}\right) v'(z)\,,$$

and we want the behavior around $z=1$ (false pole at $z=0$).

Frobenius Method [Fuchs' Theorem]

Let $v''(z) = \frac{q(z)}{(1-z)^2} v(z) + \frac{p(z)}{1-z} v'(z)$ with $q(z)$ and $p(z)$ analytic at $z=1$.

The ansatz $(1-z)^\alpha h_\alpha(z)$, with $h_\alpha(z)$ analytic at 1, $h_\alpha(1)=1$, yields a solution when α is a root of the *indicial equation*

$$X^2 = (p(1)+1)X + q(1).$$

Let α_1 and α_2 be the roots of the indicial equation.
– If $\alpha_1 - \alpha_2 \notin \mathbb{Z}$, then the two solutions are linearly independent.

When $\alpha_1 - \alpha_2 \in \mathbb{Z}$, the factor $(1-z)^{|\alpha_1-\alpha_2|}$ is polynomial.
$\Rightarrow$ we obtain an independent solution by $\times \log(1-z)$.

Conclusions and further work

Conclusions

- BST-like expression trees present a richer range of behaviours than the uniform ones.

Conclusions and further work

Conclusions

- BST-like expression trees present a richer range of behaviours than the uniform ones.
- Not exempt of degenerate cases however
  $\Rightarrow$ tuning probabilities might be important.

Conclusions and further work

Conclusions

- BST-like expression trees present a richer range of behaviours than the uniform ones.
- Not exempt of degenerate cases however
  $\Rightarrow$ tuning probabilities might be important.

Conclusions and further work

Conclusions

- $\circledast$ BST-like expression trees present a richer range of behaviours than the uniform ones.
- $\circledast$ Not exempt of degenerate cases however
  $\Rightarrow$ tuning probabilities might be important.

Questions and further work

1. Absorbing operator $\circledast$ of arity $a \geq 3$?

Conclusions and further work

Conclusions

- $\circledast$ BST-like expression trees present a richer range of behaviours than the uniform ones.
- $\circledast$ Not exempt of degenerate cases however
  $\Rightarrow$ tuning probabilities might be important.

Questions and further work

1. Absorbing operator $\circledast$ of arity $a \geq 3$?
   $\Rightarrow$ expect similar results, threshold $\frac{1}{a}$ instead of $\frac{1}{2}$.

Conclusions and further work

Conclusions

- ⊛ BST-like expression trees present a richer range of behaviours than the uniform ones.
- ⊛ Not exempt of degenerate cases however
  $\Rightarrow$ tuning probabilities might be important.

Questions and further work

1. Absorbing operator $\circledast$ of arity $a \geq 3$?
   $\Rightarrow$ expect similar results, threshold $\frac{1}{a}$ instead of $\frac{1}{2}$.
2. Absorbing pattern model is general
   $\Rightarrow$ consider interactions between operators?

Conclusions and further work

Conclusions

- $\circledast$ BST-like expression trees present a richer range of behaviours than the uniform ones.
- $\circledast$ Not exempt of degenerate cases however
  $\Rightarrow$ tuning probabilities might be important.

Questions and further work

1. Absorbing operator $\circledast$ of arity $a \geq 3$?
   $\Rightarrow$ expect similar results, threshold $\frac{1}{a}$ instead of $\frac{1}{2}$.
2. Absorbing pattern model is general
   $\Rightarrow$ consider interactions between operators?
3. Take a concrete case: LTL formulas.

Thank you!