Absorbing patterns in BST-like expression-trees

Pablo Rotondo

LIGM, Université Gustave Eiffel

Joint work with

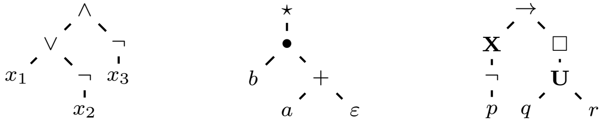
Florent Koechlin

STACS 2021,

Online, March, 2021.

Introduction

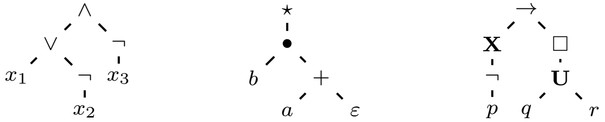
- Expression trees



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

Introduction

- Expression trees

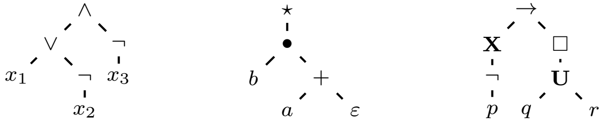


$(x_1 \vee \neg x_2) \wedge \neg x_3$ $\qquad$ $(b \cdot (a + \varepsilon))^\star$ $\qquad$ $\mathbf{X}(\neg p) \rightarrow \Box(q \mathbf{U} r)$

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

Introduction

- Expression trees



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

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

BST-like trees

Target: produce random tree with given *number of nodes* n.

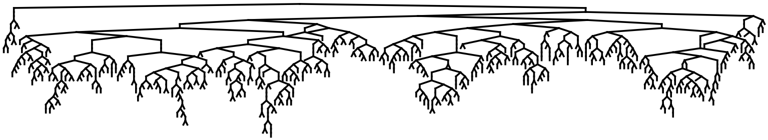
BST-like trees

Target: produce random tree with given *number of nodes* n.

```
function RandomFormula(n):
if n = 1 then
    p := random symbol in AP ∪ {⊤, ⊥};
    return p;
else if n = 2 then
    op := random unary operator in {¬, X, □, ◊};
    f := RandomFormula(1);
    return op f;
else
    op := random operator in {¬, X, □, ◊, ∧, ∨, →, ↔, U, R};
    if op in {¬, X, □, ◊} then
        f := RandomFormula(n − 1);
        return op f;
    else
        x := uniform integer in [1, n − 2];
        f1 := RandomFormula(x);
        f2 := RandomFormula(n − x − 1);
        return (f1 op f2);
```

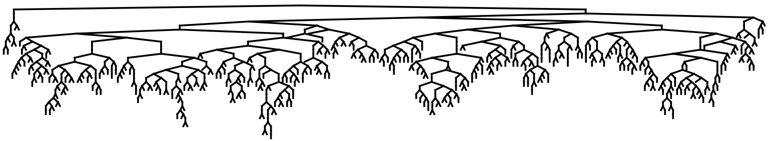
Figure: Code used in tool `lbtt` (from `TCS`) to draw an LTL formula.

BST-like trees: distribution over unary-binary trees



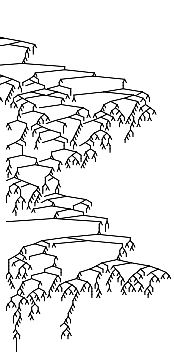
BST-like tree distribution is not uniform.

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

Universal result for uniform tree model:

Theorem (Koechlin,Nicaud,R,'20)

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

Semantic simplification

Universal result for uniform tree model:

Theorem (Koechlin,Nicaud,R,'20)

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

- Reduction based on absorbing pattern $\mathcal{P}$,

$$\overset{\circledast}{\mathcal{P} \quad T} \rightsquigarrow \mathcal{P} \qquad \overset{\circledast}{T \quad \mathcal{P}} \rightsquigarrow \mathcal{P}$$

Semantic simplification

Universal result for uniform tree model:

Theorem (Koechlin,Nicaud,R,'20)

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

- Reduction based on absorbing pattern $\mathcal{P}$,

$$\begin{array}{c}\circledast\\ \diagup\ \ \diagdown\\ \mathcal{P}\quad\ T\end{array} \rightsquigarrow \mathcal{P} \qquad\qquad \begin{array}{c}\circledast\\ \diagup\ \ \diagdown\\ T\quad\ \mathcal{P}\end{array} \rightsquigarrow \mathcal{P}$$

- Wide variety of examples:

$$\begin{array}{c}\vee\\ \diagup\ \ \diagdown\\ x_i\quad\ \neg x_i\end{array} \qquad\qquad \begin{array}{c}\star\\ \vdots\\ +\\ \diagup\ \ \diagdown\\ a\quad\ b\end{array} \qquad\qquad x \mapsto 0$$

operator $\vee$ | operator $+$ | operator $\times$

Semantic simplification

Universal result for uniform tree model:

Theorem (Koechlin,Nicaud,R,'20)

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

- Reduction based on absorbing pattern $\mathcal{P}$,

$$\begin{array}{c}\circledast\\ \diagup\ \diagdown\\ \mathcal{P}\quad T\end{array} \rightsquigarrow \mathcal{P} \qquad \begin{array}{c}\circledast\\ \diagup\ \diagdown\\ T\quad \mathcal{P}\end{array} \rightsquigarrow \mathcal{P}$$

- Wide variety of examples:

$$\begin{array}{c}\vee\\ x_i \quad \neg x_i\end{array} \qquad \begin{array}{c}\star\\ \vdots\\ +\\ a \quad b\end{array} \qquad x \mapsto 0$$

operator $\vee$ — operator $+$ — operator $\times$

- 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?

In our work we

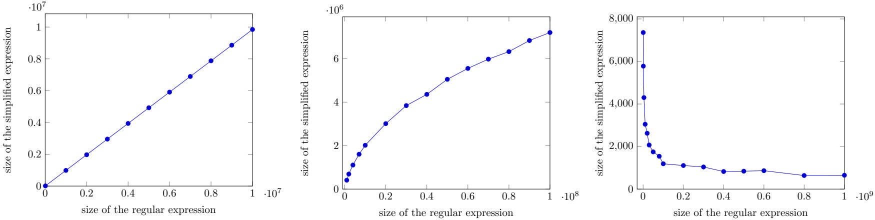
- draw an random BST-like tree expression of size n.
- study expected size of reduced expressions as $n \to \infty$.
- answer the question: do BST-like distributions present the same flaw as the uniform one? $\Rightarrow$ we characterize 5 regimes

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 $(p_\star, p_\bullet, p_+)$: $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, $(\frac{5}{29}, \frac{5}{29}, \frac{19}{29})$, and $(\frac{1}{10}, \frac{1}{10}, \frac{8}{10})$

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

Unary-binary trees:

leaves (constants), *unary*, and *binary* operators.

Expression trees and BST-like model

Unary-binary trees:

leaves (constants), *unary*, and *binary* operators.

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

Expression trees and BST-like model

Unary-binary trees:

leaves (constants), *unary*, and *binary* operators.

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

- **If** $n \geq 3$: draw an operator according to $(p_{op})_{op \in \mathcal{A}_{\texttt{ops}}}$.

Expression trees and BST-like model

Unary-binary trees:

leaves (constants), *unary*, and *binary* operators.

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

- **If** $n \geq 3$: draw an operator according to $(p_{op})_{op \in \mathcal{A}_{\mathtt{ops}}}$.
  - Case binary operator: pick $k \in \{1, \ldots, n-2\}$ uniformly.

    return $\begin{array}{ccc} & op & \\ \mathtt{BST}(k) & & \mathtt{BST}(n-k-1) \end{array}$

Expression trees and BST-like model

Unary-binary trees:

leaves (constants), *unary*, and *binary* operators.

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

- **If** $n \geq 3$: draw an operator according to $(p_{op})_{op \in \mathcal{A}_{\text{ops}}}$.
  - Case binary operator: pick $k \in \{1, \ldots, n-2\}$ uniformly.

    return $\begin{array}{ccc} & op & \\ \swarrow & & \searrow \\ \mathtt{BST}(k) & & \mathtt{BST}(n-k-1) \end{array}$
  - Case unary operator :

    return $\begin{array}{c} op \\ | \\ \mathtt{BST}(n-1) \end{array}$.

Expression trees and BST-like model

Unary-binary trees:

leaves (constants), *unary*, and *binary* operators.

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

- **If** $n \geq 3$: draw an operator according to $(p_{op})_{op \in \mathcal{A}_{\text{ops}}}$.
  - Case binary operator: pick $k \in \{1, \ldots, n-2\}$ uniformly.

    return $\begin{array}{ccc} & op & \\ \swarrow & & \searrow \\ \mathtt{BST}(k) & & \mathtt{BST}(n-k-1) \end{array}$
  - Case unary operator :

    return $\begin{array}{c} op \\ | \\ \mathtt{BST}(n-1) \end{array}$.
- **If** $n = 1$: return a leaf a according to $(p_a)_{a \in \mathcal{A}_0}$.

Expression trees and BST-like model

Unary-binary trees:

leaves (constants), *unary*, and *binary* operators.

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

- **If** $n \geq 3$: draw an operator according to $(p_{op})_{op \in \mathcal{A}_{\text{ops}}}$.
  - Case binary operator: pick $k \in \{1, \ldots, n-2\}$ uniformly.

    return $\overset{op}{\texttt{BST}(k) \quad \texttt{BST}(n-k-1)}$
  - Case unary operator :

    return $\begin{array}{c} op \\ | \\ \texttt{BST}(n-1) \end{array}$.
- **If** $n = 1$: return a leaf a according to $(p_a)_{a \in \mathcal{A}_0}$.
- **If** $n = 2$: pick op. of arity 1...

Absorbing patters: simplifying the trees

Definition (Simplification, absorbing pattern)

Consider a family of unary-binary tree expressions, consider

- an "operation" $\circledast$ with arity $a = 2$,
- a fixed expression tree $\mathcal{P}$.

Absorbing patters: simplifying the trees

Definition (Simplification, absorbing pattern)

Consider a family of unary-binary tree expressions, consider

- an "operation" $\circledast$ with arity $a = 2$,
- a fixed expression tree $\mathcal{P}$.

We simplify by applying bottom-up the rule:

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

Absorbing patters: simplifying the trees

Definition (Simplification, absorbing pattern)

Consider a family of unary-binary tree expressions, consider

- an "operation" $\circledast$ with arity $a = 2$,
- a fixed expression tree $\mathcal{P}$.

We simplify by applying bottom-up the rule:

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

$\Rightarrow$ We are interested in the *size* (*number of nodes*)
of the trees after simplification.

Absorbing patters: simplifying the trees

Definition (Simplification, absorbing pattern)

Consider a family of unary-binary tree expressions, consider

- an "operation" $\circledast$ with arity $a = 2$,
- a fixed expression tree $\mathcal{P}$.

We simplify by applying bottom-up the rule:

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

$\Rightarrow$ We are interested in the *size* (*number of nodes*) of the trees after simplification.

Denote by $\sigma(T)$ the simplification of T.

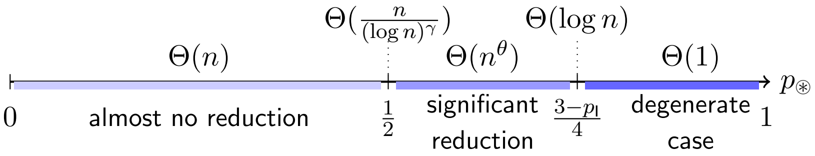
Theorem. *Consider a family of expression trees defined from unary and binary operators with an absorbing pattern $\mathcal{P}$ for an operator $\circledast$ of arity 2.*

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 $\mathcal{P}$.

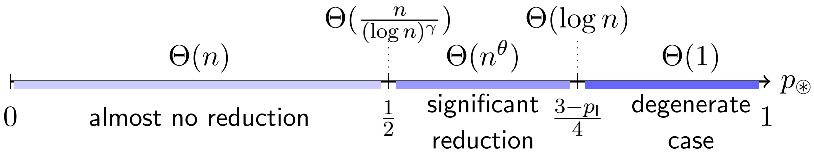
Then the expected size of the simplification of a random BST-like tree has an asymptotic behaviour given by the following cases, depending on the probability $p_{\circledast}$ of the absorbing operator:



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.

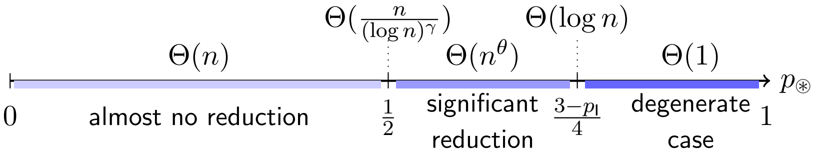
Take the *simplification* *consisting in inductively changing a* $\circledast$*-node by* $\mathcal{P}$ *whenever one of its children simplifies to* $\mathcal{P}$.

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:*

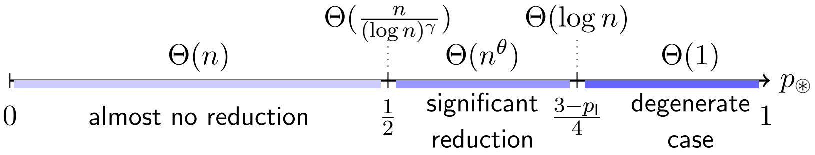


- Probability $p_\circledast$ of $\circledast$, and p_l of picking unary operator.
- Regimes from no reduction $\Theta(n)$ to complete reduction $\Theta(1)$

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 $\circledast = +$

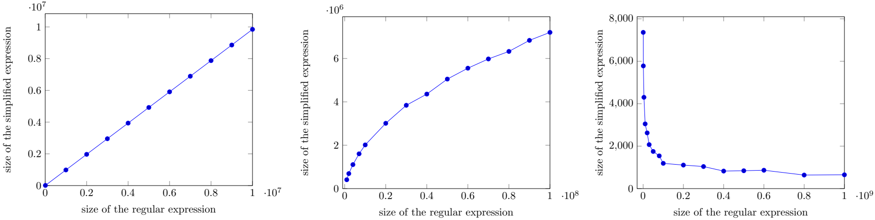


Figure: Linear ($p_+ = p_\star = p_. = \frac{1}{3}$), sublinear ($p_+ = \frac{19}{29}$, $p_\star = p_. = \frac{5}{29}$) and constant ($p_+ = \frac{8}{10}$, $p_\star = p_. = \frac{1}{10}$).

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

We consider a fundamental sequence

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

of the probabilities of *full* reduction.

Symbolic step

We consider a fundamental sequence

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

of the probabilities of *full* reduction.

First order **differential equation** for the generating function

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

where $A(z) = \sum_n \gamma_n z^n$.

Symbolic step

We consider a fundamental sequence

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

of the probabilities of *full* reduction.

First order **differential equation** for the generating function

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

where $A(z) = \sum_n \gamma_n z^n$.

Proof. Recurrence for e_n involving γ_n,

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

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

Analytic step: the singularity $z = 1$

Symbolic step gives differential equation:

- First order differential equations can be solved explicitly,
- But coefficients depend on unknown generating function $A(z)$.

Analytic step: the singularity $z = 1$

Symbolic step gives differential equation:

- First order differential equations can be solved explicitly,
- But coefficients depend on unknown generating function $A(z)$.

To apply the Transfer Theorem and complete the proof:

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

Analytic step: the singularity $z = 1$

Symbolic step gives differential equation:

- First order differential equations can be solved explicitly,
- But coefficients depend on unknown generating function $A(z)$.

To apply the Transfer Theorem and complete the proof:

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

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

Fully reducible trees: probabilities

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

Proposition

Generating function satisfies 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 $s = |\mathcal{P}|$.

Fully reducible trees: probabilities

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

Proposition

Generating function satisfies 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 $s = |\mathcal{P}|$.

Proof.

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

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

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 for $p_\circledast \leq \frac{1}{2}$.

Probability of full reduction

Theorem

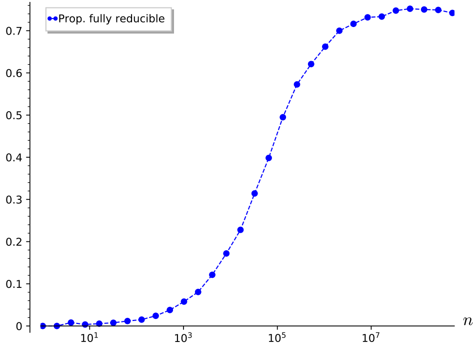
The probability γ_n of being fully reducible tends to the constant $\gamma_\infty := (2p_\circledast - 1)/p_\circledast$ for $p_\circledast > \frac{1}{2}$, and to zero for $p_\circledast \leq \frac{1}{2}$.

Experimental plot: regular expressions on two letters with

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

Then

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



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

- ⊛ 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 $a \geq 3$?

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

- $\circledast$ 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?

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!