Equation discovery for integer sequences with probabilistic grammars

Boštjan Gec

mentor: prof. dr. Ljupčo Todorovski

26. september 2023

Introduction

Equation discovery for integer sequences with probabilistic grammars

- Equation discovery
- Probabilistic grammars
- Integer sequences

Dataset is a matrix $X \in \mathbb{R}^{m \times n}$, where:

- columns correspond to **observed (numeric) variables (features)**, that we usually $x_1, x_2, ..., x_n$ ter
- rows correspond to **instances** and each instance is made up of measured values of observed variables.

Therefore:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n \end{bmatrix}$$

For example:

Temperature (x_1) [K]	Volume (x_2) [l]	Pressure (x_3) [kPa]
291	6.97	348.26
285	9.12	260.24

Algorithm for equation discovery

Dataset

For example:

Temperature (x_1) [K]	Volume (x_2) [l]	Pressure (x_3) [kPa]
291	6.97	348.26
285	9.12	260.24
$\vdots$	$\vdots$	$\vdots$
307	6.61	386.84

Tabela: Table of measurements of physical experiment.

This way we get dataset X of dimension 50×3.

- The task is to find out equations of the form:

$$x_n = f(\vec{c}, \vec{x}) = f((c_1, ..., c_k), (x_1, x_2, ..., x_{n-1})) \quad ,$$

where $\vec{c}$ is a vector of real konstants, $\vec{x} := (x_1, x_2, ..., x_{n-1})$ vector of observed variables and $f(\vec{c}, \vec{x})$ an arithmetič expression, which includes any number of real constants and any choice of observed variables $x_1, ..., x_{n-1}$.
We search for equations, that connect columns $x_1, x_2, ..., x_n$ with each other and that fit to the dataset as much as possible.

- Model error

$$Err(f, \vec{c}, X) := \frac{1}{m} \sum_{i=1}^{m} (x_{in} - f(\vec{c}, (x_{i1}, \cdots, x_{i(n-1)})))^2 \quad .$$

- The task is a optimization problem of finding the minimum:

$$\min \quad Err(f, \vec{c}, X)$$

Algorithm for equation discovery

Task of equation discovery

- The task is to find out equations of the form:

$$x_n = f(\vec{c}, \vec{x}) = f((c_1, ..., c_k), (x_1, x_2, ..., x_{n-1})) \ ,$$

  where $\vec{c}$ is a vector of real konstants, $\vec{x} := (x_1, x_2, ..., x_{n-1})$ vector of observed variables and $f(\vec{c}, \vec{x})$ an arithmetič expression, which includes any number of real constants and any choice of observed variables $x_1, ..., x_{n-1}$.
  We search for equations, that connect columns $x_1, x_2, ..., x_n$ with each other and that fit to the dataset as much as possible.

- Model error

$$Err(f, \vec{c}, X) := \frac{1}{m} \sum_{i=1}^{m} (x_{in} - f(\vec{c}, (x_{i1}, \cdots, x_{i(n-1)})))^2 \ ,$$

- The task is a optimization problem of finding the minimum:

$$\min \quad Err(f, \vec{c}, X)$$

Algorithm for equation discovery

Task of equation discovery

- The task is to find out equations of the form:

$$x_n = f(\vec{c}, \vec{x}) = f((c_1, ..., c_k), (x_1, x_2, ..., x_{n-1})) \quad ,$$

- Model error

$$Err(f, \vec{c}, X) := \frac{1}{m} \sum_{i=1}^{m} (x_{in} - f(\vec{c}, (x_{i1}, \cdots, x_{i(n-1)})))^2 \quad ,$$

  where f is the expression structure, $\vec{c}$ the vector of constants, m and n number of instances and number of observed variables in dataset $X \in \mathbb{R}^{m \times n}$ and x_{ij} represents ij-th element of matrix X.

- The task is a optimization problem of finding the minimum:

$$\min_{\vec{c} \in I_1, f \in L_X} Err(f, \vec{c}, X) \quad ,$$

  where I_1 is the set of all real vectors of arbitrary dimension and

- The task is to find out equations of the form:

$$x_n = f(\vec{c}, \vec{x}) = f((c_1, ..., c_k), (x_1, x_2, ..., x_{n-1})) \ ,$$

- Model error

$$Err(f, \vec{c}, X) := \frac{1}{m} \sum_{i=1}^{m} (x_{in} - f(\vec{c}, (x_{i1}, \cdots, x_{i(n-1)})))^2 \ ,$$

- The task is a optimization problem of finding the minimum:

$$\min_{\vec{c} \in I_1, f \in L_X} Err(f, \vec{c}, X) \ ,$$

where I_1 is the set of all real vectors of arbitrary dimension and L_X is the chosen set of all expression structures.

Algorithm for equation discovery

Algorithm 1 Equation discovery with approach **generate and test**

Input: Dataset X, number of generated equation structures N, functions *generate* and *test*

Output: List *equations*, that contains all pairs of equations and corresponding errors.

function EquationDiscovery(X, N, *generate*, *test*)
    $equations \leftarrow [\ ]$
    for $i = 1, ..., N$ **do**
        $f \leftarrow generate(X)$
        $(equation, error) \leftarrow test(f, X)$
        $equations.\mathrm{append}((equation, error))$
    end for
    return *equations*.sorted(key=*error*)
end function

Context-free grammar

Definition

Context-free grammar G is quadruple (N, T, R, S), such that:

1. T is finite set of **terminal symbols**,
2. N is finite set of **nonterminal symbols**, for witch the intersection of N and T is empty. Elements from $N \cup T$ are called *symbols*.
3. R is finite set of **rewriting rules**, i.e. pairs (A, α), where $A \in N$ and $\alpha \in (N \cup T)^*$, where * denotes the set of all finite sequences with terms from the given set.
4. S is nonterminal symbol from N and is named **start** (or *sentence*) **symbol**.

Context-free grammar

Definition (Rewriting relation)

Rewriting relation $\Rightarrow_G$ over $(N \cup T)^*$:

$$\beta A \gamma \Rightarrow_G \beta \alpha \gamma,$$

if

$$(A, \alpha) \in R,$$

where $\beta, \gamma \in (N \cup T)^*$.

Usually

We write $A \to \alpha$ instead of $(A, \alpha) \in R$.

Context-free grammar

Definition (Rewriting relation)

Rewriting relation $\Rightarrow_G$ over $(N \cup T)^*$:

$$\beta A \gamma \Rightarrow_G \beta \alpha \gamma,$$

if

$$(A, \alpha) \in R,$$

where $\beta, \gamma \in (N \cup T)^*$.

Usually

We write $A \rightarrow \alpha$ instead of $(A, \alpha) \in R$.

Context-free grammar

Closure of relation

The reflexive, transitive closure $\Rightarrow_G^*$ of rewriting relation $\Rightarrow_G$.

Sentential form

Sequence of symbols γ is **sentential form**, if $S \Rightarrow_G^* \gamma$.

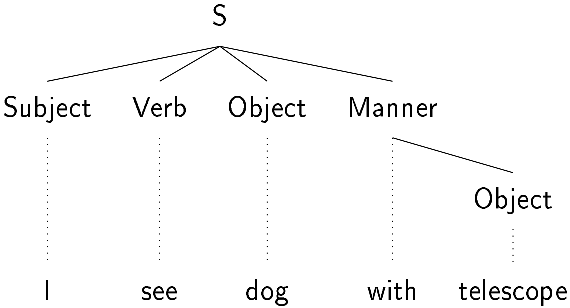
Sentences of the grammar

$L_G := \{\alpha \in T^* | S \Rightarrow_G^* \alpha\}$

T := {I, see, invent, dog, with, telescope}
N := {S, Subject, Verb, Object, Manner}
R := {
S $\rightarrow$ Subject Verb Object
S $\rightarrow$ Subject Verb Object Manner
Subject $\rightarrow$ I
Verb $\rightarrow$ see
Verb $\rightarrow$ invent
Object $\rightarrow$ Object with Object
Object $\rightarrow$ dog
Object $\rightarrow$ telescope
Manner $\rightarrow$ with Object
}

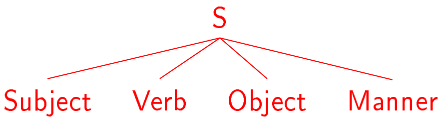
Sentence in language of this grammar

`I see dog with telescope`



R :

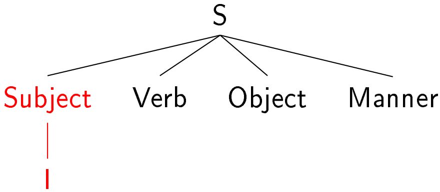
S $\rightarrow$ Subject Verb Object
S $\rightarrow$ Subject Verb Object Manner
Subject $\rightarrow$ I
Verb $\rightarrow$ see
Verb $\rightarrow$ invent
Object $\rightarrow$ Object with Object
Object $\rightarrow$ dog
Object $\rightarrow$ telescope
Manner $\rightarrow$ with Object



R :

S $\rightarrow$ Subject Verb Object
S $\rightarrow$ Subject Verb Object Manner
Subject $\rightarrow$ I
Verb $\rightarrow$ see
Verb $\rightarrow$ invent

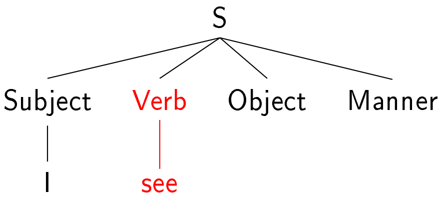
Object $\rightarrow$ Object with Object
Object $\rightarrow$ dog
Object $\rightarrow$ telescope
Manner $\rightarrow$ with Object



R :

S → Subject Verb Object
S → Subject Verb Object Manner
Subject → I
Verb → see
Verb → invent

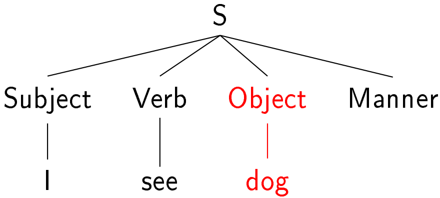
Object → Object with Object
Object → dog
Object → telescope
Manner → with Object



R :

S $\rightarrow$ Subject Verb Object
S $\rightarrow$ Subject Verb Object Manner
Subject $\rightarrow$ I
Verb $\rightarrow$ see
Verb $\rightarrow$ invent

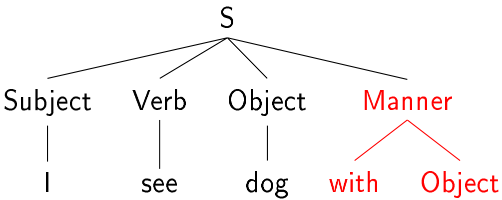
Object $\rightarrow$ Object with Object
Object $\rightarrow$ dog
Object $\rightarrow$ telescope
Manner $\rightarrow$ with Object



R :

S $\rightarrow$ Subject Verb Object
S $\rightarrow$ Subject Verb Object Manner
Subject $\rightarrow$ I
Verb $\rightarrow$ see
Verb $\rightarrow$ invent

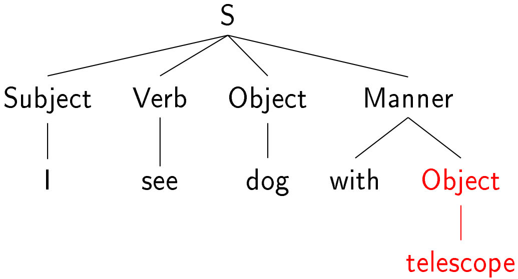
Object $\rightarrow$ Object with Object
Object $\rightarrow$ dog
Object $\rightarrow$ telescope
Manner $\rightarrow$ with Object



R :

S $\rightarrow$ Subject Verb Object
S $\rightarrow$ Subject Verb Object Manner
Subject $\rightarrow$ I
Verb $\rightarrow$ see
Verb $\rightarrow$ invent

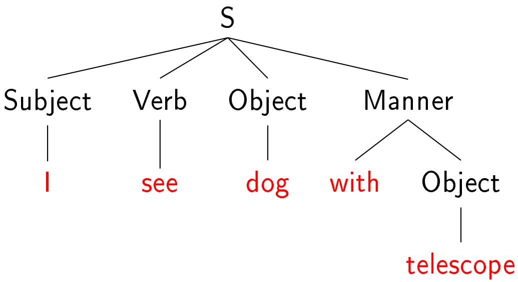
Object $\rightarrow$ Object with Object
Object $\rightarrow$ dog
Object $\rightarrow$ telescope
Manner $\rightarrow$ with Object



R :

S → Subject Verb Object
S → Subject Verb Object Manner
Subject → I
Verb → see
Verb → invent

Object → Object with Object
Object → dog
Object → telescope
Manner → with Object



T := {I, see, invent, dog, with, telescope}

R :

S → Subject Verb Object
S → Subject Verb Object Manner
Subject → I
Verb → see
Verb → invent

Object → Object with Object
Object → dog
Object → telescope
Manner → with Object

Example: Universal grammar for arithmetic expression structures

T := {'+', '-', '*', '/', 'C', '(', ')', 'sin(', 'cos(', 'sqrt(', 'exp(', 'x1', 'x2', 'x3'}

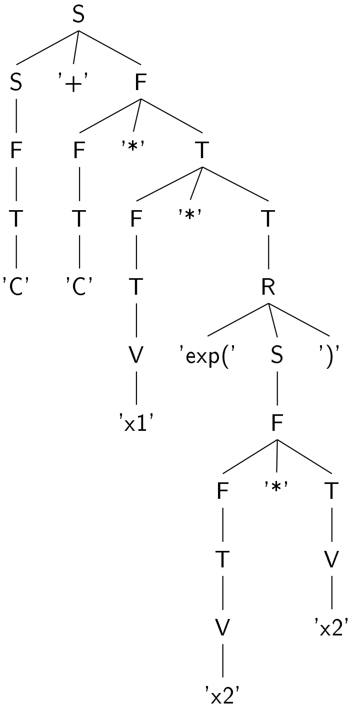
N:= {S, F, T, R, V}

R:

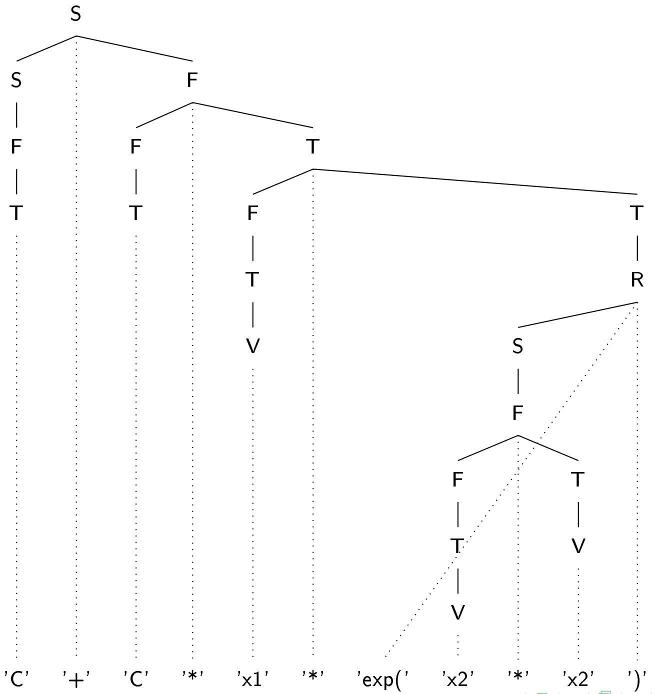
S → S '+' F	S → S '-' F
S → F	F → F '*' T
F → F '/' T	F → T
T → R	T → 'C'
T → V	R → '(' S ')'
R → 'sin(' S ')'	R → 'cos(' S ')'
R → 'sqrt(' S ')'	R → 'exp(' S ')'
V → 'x1'	V → 'x2'
V → 'x3'	

Example of a sentence in this grammar:

'C' '+' 'C' '*' 'x1' '*' 'exp(' 'x2' '*' 'x2' ')'



S → S '+' F
S → S '-' F
S → F
F → F '*' T
F → F '/' T
F → T
T → R
T → 'C'
T → V
R → '(' S ')'
R → 'sin(' S ')'
R → 'cos(' S ')'
R → 'sqrt(' S ')'
R → 'exp(' S ')'
V → 'x1'
V → 'x2'
V → 'x3'


S
S F
F F T
T T F T
T R
V S
F
F T
T V
V
'C' '+' 'C' '*' 'x1' '*' 'exp(' 'x2' '*' 'x2' ')'

Probabilistic context-free grammars

Example from before:

T := {I, see, invent, dog, with, telescope}
N := {S, Subject, Verb, Object, Manner}
R := {
S $\rightarrow$ Subject Verb Object
S $\rightarrow$ Subject Verb Object Manner
Subject $\rightarrow$ I
Verb $\rightarrow$ see
Verb $\rightarrow$ invent
Object $\rightarrow$ Object with Object
Object $\rightarrow$ dog
Object $\rightarrow$ telescope
Manner $\rightarrow$ with Object
}

Probabilistic context-free grammars

Example from before:

T := {I, see, invent, dog, with, telescope}
N := {S, Subject, Verb, Object, Manner}
R := {
S $\rightarrow$ Subject Verb Object
S $\rightarrow$ Subject Verb Object Manner
Subject $\rightarrow$ I
Verb $\rightarrow$ see
Verb $\rightarrow$ invent
Object $\rightarrow$ Object with Object
Object $\rightarrow$ dog
Object $\rightarrow$ telescope
Manner $\rightarrow$ with Object
}

Probabilistic context-free grammars

Example from before:

T := {I, see, invent, dog, with, telescope}
N := {S, Subject, Verb, Object, Manner}
R := {
S $\rightarrow$ Subject Verb Object [0.3]
S $\rightarrow$ Subject Verb Object Manner [0.7]
Subject $\rightarrow$ I [1]
Verb $\rightarrow$ see [0.9]
Verb $\rightarrow$ invent [0.1]
Object $\rightarrow$ Object with Object [0.2]
Object $\rightarrow$ dog [0.3]
Object $\rightarrow$ telescope [0.5]
Manner $\rightarrow$ with Object [1]
}

Condition

$\forall A \in N$:

$$\sum_{(A \to \alpha) \in R} p(A \to \alpha) = 1,$$

where $p : R \to [0, 1]$ is a function, that prescribes value to each rewriting rule.

Probability of rewriting rule

$$p(A \to \alpha)$$

is called **probability of rewriting rule**.

Algorithm 2 Function *generate*, that uses probabilistic grammars

Input: Probabilistic grammar $G = (N, T, R, S)$, symbol $A \in N$
Output: Sentence s in grammar G

function Generate(G, A)
    $s \leftarrow [\,]$
    Randomly choose rewriting rule $(A \rightarrow A_1\ A_2 \cdots A_k) \in R$
    for $i = 1, ..., k$ **do**
        if $A_i \in T$ **then**
            $s = s$.append(A_i)
        else
            s_i = Generate(G, A_i)
            $s = s$.append(s_i)
        end if
    end for
    return s
end function

Randomized algorithm

1. Start in start symbol S
2. Repeat the step:
   Take first nonterminal symbol in current sentential form.
3. Choose a rewriting rule randomly according to the distribution given by probabilities of rules.
4. Finish when we get sentence (in sentential form only terminal symbols remain).

Probability of derivation tree

Probability that we get the given derivation tree

$\forall \tau \in \Psi$:

$$P(\tau) := \prod_{(A\to\alpha)\in R} p(A \to \alpha)^{f(A\to\alpha;\tau)} =: p(\tau)$$

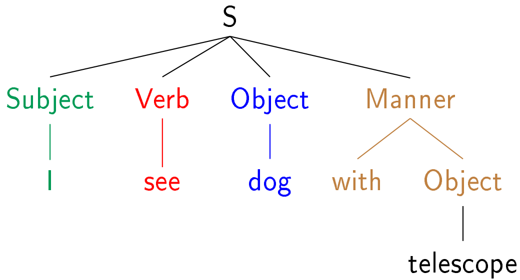
where $p(A \to \alpha)$ is probability of rewriting rule, $f(A \to \alpha; \tau)$ frequency or number of occurrences of the rule $A \to \alpha$ in the tree τ and Ψ is the set of all finite derivation trees.

Probability of derivation tree

$$p(\tau)$$

is called **probability of derivation tree**.

$\tau :=$



$$
\begin{aligned}
p(\tau) &= \\
&= p(S \to \ Subject\ Verb\ Object\ Manner) \cdot \\
&\cdot p(Subject\ \to I) \cdot p(Verb\ \to see) \cdot p(Object\ \to dog) \cdot \\
&\cdot p(Manner \to with\ \ Object) \cdot \ \cdot p(Object \to telescope)
\end{aligned}
$$

New sample space

- Randomized algorithm has its own probability distribution denoted by P, that prescribes probabilities to events of the underlying random process. We already saw that: $\forall \tau \in \Psi : P(\tau) = p(\tau)$.
- New sample space: $\tilde{\Omega} := \Psi = \{\text{finite derivation trees}\}$
- We want probability distribution P_Ψ on Ψ, that is a restriction of the distribution P, i.e.:

$$\forall \tau \in \Psi : P_\Psi(\tau) = p(\tau)$$

- Possible only if coverage equals 1, i.e.:

$$P_\Psi(\Psi) = \sum_{\tau \in \Psi} P_\Psi(\tau) = \sum_{\tau \in \Psi} p(\tau) =: Coverage_G = 1$$

New sample space

- Randomized algorithm has its own probability distribution denoted by P, that prescribes probabilities to events of the underlying random process. We already saw that: $\forall \tau \in \Psi : P(\tau) = p(\tau)$.
- New sample space: $\tilde{\Omega} := \Psi = \{\text{finite derivation trees}\}$
- We want probability distribution P_Ψ on Ψ, that is a restriction of the distribution P, i.e.:

$$\forall \tau \in \Psi : P_\Psi(\tau) = p(\tau)$$

- Possible only if coverage equals 1, i.e.:

$$P_\Psi(\Psi) = \sum_{\tau \in \Psi} P_\Psi(\tau) = \sum_{\tau \in \Psi} p(\tau) =: Coverage_G = 1$$

New sample space

- Randomized algorithm has its own probability distribution denoted by P, that prescribes probabilities to events of the underlying random process. We already saw that: $\forall \tau \in \Psi : P(\tau) = p(\tau)$.
- New sample space: $\tilde{\Omega} := \Psi = \{\text{finite derivation trees}\}$
- We want probability distribution P_Ψ on Ψ, that is a restriction of the distribution P, i.e.:

$$\forall \tau \in \Psi : P_\Psi(\tau) = p(\tau)$$

- Possible only if coverage equals 1, i.e.:

$$P_\Psi(\Psi) = \sum_{\tau \in \Psi} P_\Psi(\tau) = \sum_{\tau \in \Psi} p(\tau) =: Coverage_G = 1$$

New sample space

- Randomized algorithm has its own probability distribution denoted by P, that prescribes probabilities to events of the underlying random process. We already saw that: $\forall \tau \in \Psi : P(\tau) = p(\tau)$.
- New sample space: $\tilde{\Omega} := \Psi = \{\text{finite derivation trees}\}$
- We want probability distribution P_Ψ on Ψ, that is a restriction of the distribution P, i.e.:

$$\forall \tau \in \Psi : P_\Psi(\tau) = p(\tau)$$

- Possible only if coverage equals 1, i.e.:

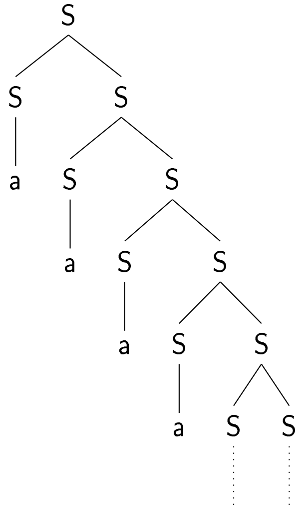
$$P_\Psi(\Psi) = \sum_{\tau \in \Psi} P_\Psi(\tau) = \sum_{\tau \in \Psi} p(\tau) =: Coverage_G = 1$$

Counterexample

T:= {a}
N:= {S}
R:= {
S $\rightarrow$ S S [0.6]
S $\rightarrow$ a [0.4]
}

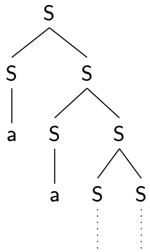
Counterexample

T:= {a}
N:= {S}
R:= {
S → S S [0.6]
S → a [0.4]
}



Counterexample

T:= {a}
N:= {S}
R:= {
S $\rightarrow$ S S [0.6]
S $\rightarrow$ a [0.4]
}

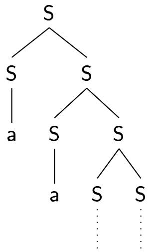


Coverage is less than 1

$$Coverage_G = \sum_{\tau \in \Psi} p(\tau) = \frac{2}{3} < 1$$

Counterexample

T:= {a}
N:= {S}
R:= {
S $\rightarrow$ S S [0.5]
S $\rightarrow$ a [0.5]
}



Coverage equals 1

$$Coverage_G = \sum_{\tau \in \Psi} p(\tau) = 1$$

Multitype branching processes

Proving theorem in the field of multi-type branching processes, that tries to answer the question: When can we define such probability distribution.

Multitype branching processes

- Random vector of generation

$$\mathrm{Z}_n := (Z_n^1, ..., Z_n^m) : \Omega \to \mathbb{N}_0^m \quad ,$$

which represents the number of individuals of each type in n-th generation.

- m ... number of all types
- n ... generation
- Z_n^i ... number of individuals of type i in the generation n
- Random vector of the individual I

$$N(I) : \Omega \to \mathbb{N}_0^m \quad ,$$

Multitype branching processes

- Random vector of generation

$$\mathrm{Z}_n := (Z_n^1, ..., Z_n^m) : \Omega \to \mathbb{N}_0^m \quad ,$$

which represents the number of individuals of each type in n-th generation.

Probabilistic grammar (N, T, S, R, p) as multitype branching process

- $m = |N|$
- generation: level in the derivation tree (generation zero corresponds to the root node)
- individuals: nonterminal symbols in the derivation tree
- $\mathrm{Z}_0 = \mathrm{e}_1$

Generating function of type i

This is the function $f^i : [-1,1]^m \to \mathbb{R}$

$$f^i(\mathrm{s}) := f^i(s_1, \dots, s_m) := \sum_{\mathrm{r} \in \mathbb{N}_0^m} P(\mathrm{Z}_{n+1} = \mathrm{r} | \mathrm{Z}_n = \mathrm{e}_i) s_1^{r_1} \cdots s_m^{r_m}$$

where $\mathrm{r} := (r_1, \dots, r_m), \mathrm{e}_i \in \mathbb{N}_0^m$ and e_i is the standard unit vector.

Generating function of a nonterminal symbol

Function $f^i : [-1,1]^m \to \mathbb{R}$

$$f^{A_i}(s_1, \dots, s_m) := \sum_{\mathrm{r} \in \mathbb{N}_0^m} \left(\sum_{A_i \to \alpha; \mathrm{v}(\alpha) = \mathrm{r}} p(A_i \to \alpha) \right) s_1^{r_1} \cdots s_m^{r_m}$$

where r as before and $\mathrm{v}(\alpha) := (f(A_1; \alpha), \dots, f(A_m; \alpha))$, where $f(A_j; \alpha)$ counts the number of symbols A_j in α.

Generating function of type i

This is the function $f^i : [-1,1]^m \to \mathbb{R}$

$$f^i(\mathrm{s}) := f^i(s_1, \ldots, s_m) := \sum_{\mathrm{r} \in \mathbb{N}_0^m} P(\mathrm{Z}_{n+1} = \mathrm{r} | \mathrm{Z}_n = \mathrm{e}_i) s_1^{r_1} \cdots s_m^{r_m}$$

where $\mathrm{r} := (r_1, \ldots, r_m), \mathrm{e}_i \in \mathbb{N}_0^m$ and e_i is the standard unit vector.

Generating function of a nonterminal symbol

Function $f^i : [-1,1]^m \to \mathbb{R}$

$$f^{A_i}(s_1, \ldots, s_m) := \sum_{\mathrm{r} \in \mathbb{N}_0^m} \left(\sum_{A_i \to \alpha; \mathrm{v}(\alpha) = \mathrm{r}} p(A_i \to \alpha) \right) s_1^{r_1} \cdots s_m^{r_m}$$

where r as before and $\mathrm{v}(\alpha) := (f(A_1; \alpha), \ldots, f(A_m; \alpha))$, where $f(A_j; \alpha)$ counts the number of symbols A_j in α.

Example from before:

$T := \{$I, see, invent, dog, with, telescope$\}$
$N := \{$S, Subject, Verb, Object, Manner$\}$
$R := \{$
S $\rightarrow$ Subject Verb Object [0.3]
S $\rightarrow$ Subject Verb Object Manner [0.7]
Subject $\rightarrow$ I [1]
Verb $\rightarrow$ see [0.9]
Verb $\rightarrow$ invent [0.1]
Object $\rightarrow$ Object with Object [0.2]
Object $\rightarrow$ dog [0.3]
Object $\rightarrow$ telescope [0.5]
Manner $\rightarrow$ with Object [1]
$\}$

1. Friendlier notation

T := {I, see, inv, d, w, t}

N := {S, Sub, V, O, M}

R :

S → Sub V O [0.3]

S → Sub V O M [0.7]

Sub → I [1]

V → see [0.9]

V → inv [0.1]

O → O w O [0.2]

O → d [0.3]

O → t [0.5]

M → with O [1]

1. Friendlier notation

T := {I, see, inv, d, w, t}

N := {S, Sub, V, O, M}

R :

- S $\rightarrow$ Sub V O [0.3]
- S $\rightarrow$ Sub V O M [0.7]
- Sub $\rightarrow$ I [1]
- V $\rightarrow$ see [0.9]
- V $\rightarrow$ inv [0.1]
- O $\rightarrow$ O w O [0.2]
- O $\rightarrow$ d [0.3]
- O $\rightarrow$ t [0.5]
- M $\rightarrow$ with O [1]

2. Remove terminal symbols

N := {S, Sub, V, O, M}

R :

- S $\rightarrow$ Sub V O [0.3]
- S $\rightarrow$ Sub V O M [0.7]
- Sub $\rightarrow$ ϵ [1]
- V $\rightarrow$ ϵ [0.9]
- V $\rightarrow$ ϵ [0.1]
- O $\rightarrow$ O O [0.2]
- O $\rightarrow$ ϵ [0.3]
- O $\rightarrow$ ϵ [0.5]
- M $\rightarrow$ O [1]

2. Remove terminal symbols

N := {S, Sub, V, O, M}

R :

$S \rightarrow$ Sub V O [0.3]

$S \rightarrow$ Sub V O M [0.7]

Sub $\rightarrow \epsilon$ [1]

$V \rightarrow \epsilon$ [0.9]

$V \rightarrow \epsilon$ [0.1]

$O \rightarrow$ O O [0.2]

$O \rightarrow \epsilon$ [0.3]

$O \rightarrow \epsilon$ [0.5]

$M \rightarrow$ O [1]

3. Merge identical rules

R :

$S \rightarrow$ Sub V O [0.3]

$S \rightarrow$ Sub V O M [0.7]

Sub $\rightarrow \epsilon$ [1]

$V \rightarrow \epsilon$ [1]

$O \rightarrow$ O O [0.2]

$O \rightarrow \epsilon$ [0.8]

$M \rightarrow$ O [1]

3. Merge identical rules

R :

$S \to$ Sub V O [0.3]
$S \to$ Sub V O M [0.7]
Sub $\to \epsilon$ [1]

$V \to \epsilon$ [1]

$O \to$ O O [0.2]

$O \to \epsilon$ [0.8]

$M \to$ O [1]

4. Convert strings to vectors

R :

$(1,0,0,0,0) \to (0,1,1,1,0)$ [0.3]
$(1,0,0,0,0) \to (0,1,1,1,1)$ [0.7]
$(0,1,0,0,0) \to (0,0,0,0,0)$ [1]

$(0,0,1,0,0) \to (0,0,0,0,0)$ [1]

$(0,0,0,1,0) \to (0,0,0,2,0)$ [0.2]

$(0,0,0,1,0) \to (0,0,0,0,0)$ [0.8]

$(0,0,0,0,1) \to (0,0,0,1,0)$ [1]

4. Convert strings to vectors

R :

$$(1,0,0,0,0) \to (0,1,1,1,0)\ [0.3]$$
$$(1,0,0,0,0) \to (0,1,1,1,1)\ [0.7]$$
$$(0,1,0,0,0) \to (0,0,0,0,0)\ [1]$$
$$(0,0,1,0,0) \to (0,0,0,0,0)\ [1]$$
$$(0,0,0,1,0) \to (0,0,0,2,0)\ [0.2]$$
$$(0,0,0,1,0) \to (0,0,0,0,0)\ [0.8]$$
$$(0,0,0,0,1) \to (0,0,0,1,0)\ [1]$$

5. What we get

R :

$$P(Z_{n+1} = (0,1,1,1,0) | Z_n = e_1) = 0.3$$
$$P(Z_{n+1} = (0,1,1,1,1) | Z_n = e_1) = 0.7$$
$$P(Z_{n+1} = (0,0,0,0,0) | Z_n = e_2) = 1$$
$$P(Z_{n+1} = (0,0,0,0,0) | Z_n = e_3) = 1$$
$$P(Z_{n+1} = (0,0,0,2,0) | Z_n = e_4) = 0.2$$
$$P(Z_{n+1} = (0,0,0,0,0) | Z_n = e_4) = 0.8$$
$$P(Z_{n+1} = (0,0,0,1,0) | Z_n = e_5) = 1$$

5. What we get

R :

$$P(Z_{n+1} = (0,1,1,1,0) | Z_n = e_1) = 0.3$$
$$P(Z_{n+1} = (0,1,1,1,1) | Z_n = e_1) = 0.7$$
$$P(Z_{n+1} = (0,0,0,0,0) | Z_n = e_2) = 1$$
$$P(Z_{n+1} = (0,0,0,0,0) | Z_n = e_3) = 1$$
$$P(Z_{n+1} = (0,0,0,2,0) | Z_n = e_4) = 0.2$$
$$P(Z_{n+1} = (0,0,0,0,0) | Z_n = e_4) = 0.8$$
$$P(Z_{n+1} = (0,0,0,1,0) | Z_n = e_5) = 1$$

6. Corresponding generating functions

$$f^1(s) := 0.3 \cdot s_2 s_3 s_4 + 0.7 \cdot s_2 s_3 s_4 s_5$$
$$f^2(s) := 1$$
$$f^3(s) := 1$$
$$f^4(s) := 0.8 + 0.2 \cdot s_4^2$$
$$f^5(s) := s_4$$

Extinction probability vector

Extinction probability vector of multitype branching process is vector $\mathrm{q} := (q_1, ..., q_m)$ of probabilities, where its i-th component equals the probability that descendants of a individual of type i becomes extinct, i.e.

$$q_i := P(\mathrm{Z}_n = 0 \text{ for some } n \in \mathbb{N} \mid \mathrm{Z}_0 = \mathrm{e}_i) \ .$$

The i-th component q_i is called **extinction probability**.

Singularity

Multitype branching process is **singular**, if all of its generating functions are linear combinations of terms $s_1, s_2, ..., s_m$:

$$\forall i = 1, ..., m : f^i(\mathrm{s}) = \pi_{i1} s_1 + \pi_{i2} s_2 + ... + \pi_{im} s_m \ ,$$

where $\sum_{j=1}^{m} \pi_{ij} = 1$.

Extinction probability vector

Extinction probability vector of multitype branching process is vector $\mathrm{q} := (q_1, \dots, q_m)$ of probabilities:

$$q_i := P(\mathrm{Z}_n = 0 \text{ for some } n \in \mathbb{N} \mid \mathrm{Z}_0 = \mathrm{e}_i) \ .$$

The i-th component q_i is called **extinction probability**.

Singularity

Multitype branching process is **singular**, if all of its generating functions are linear combinations of terms $s_1, s_2, \dots, s_m$:

$$\forall i = 1, \dots, m : f^i(\mathrm{s}) = \pi_{i1} s_1 + \pi_{i2} s_2 + \dots + \pi_{im} s_m \ ,$$

where $\sum_{j=1}^{m} \pi_{ij} = 1$.

Matrix of first moments M

$$M = (m_{ij}) \in \mathbb{R}^{m \times m}$$

$$m_{ij} := E[Z_1^j | \mathrm{Z}_0 = \mathrm{e}_i]$$

$$= \frac{"\partial" f^i(\mathrm{s})}{"\partial" \mathrm{s}_j} := \sum_{\mathrm{r} \in \mathbb{N}_0^m} \frac{\partial}{\partial \mathrm{s}_j} (P(\mathrm{Z}_{n+1} = \mathrm{r} | \mathrm{Z}_n = \mathrm{e}_i) s_1^{r_1} \cdots s_m^{r_m})$$

Matrix of first moments M

$$M = (m_{ij}) \in \mathbb{R}^{m \times m}$$

$$m_{ij} := E[Z_1^j | \mathrm{Z}_0 = \mathrm{e}_i]$$

$$= \frac{"\partial" f^i(\mathrm{s})}{"\partial" \mathrm{s}_j} := \sum_{\mathrm{r} \in \mathbb{N}_0^m} \frac{\partial}{\partial \mathrm{s}_j} (P(\mathrm{Z}_{n+1} = \mathrm{r} | \mathrm{Z}_n = \mathrm{e}_i) s_1^{r_1} \cdots s_m^{r_m})$$

Positively regular process

Multitype branching process is **positively regular**, if there exists $n \in \mathbb{N}$, such that the n-th power of the matrix of first moments M^n is positive, i.e. $\forall i, j : [M^n]_{ij} > 0$.

Matrix of first moments M

$$M = (m_{ij}) \in \mathbb{R}^{m\times m}$$
$$m_{ij} := E[Z_1^j | \mathrm{Z}_0 = \mathrm{e}_i]$$
$$= \frac{"\partial" f^i(\mathrm{s})}{"\partial" \mathrm{s}_j} := \sum_{\mathrm{r}\in\mathbb{N}_0^m} \frac{\partial}{\partial \mathrm{s}_j}(P(\mathrm{Z}_{n+1} = \mathrm{r}|\mathrm{Z}_n = \mathrm{e}_i)s_1^{r_1}\cdots s_m^{r_m})$$

$$\begin{aligned} f^1(\mathrm{s}) &:= 0.3\cdot s_2 s_3 s_4 + 0.7\cdot s_2 s_3 s_4 s_5 \\ f^2(\mathrm{s}) &:= 1 \\ f^3(\mathrm{s}) &:= 1 \\ f^4(\mathrm{s}) &:= 0.8 + 0.2\cdot s_4^2 \\ f^5(\mathrm{s}) &:= s_4 \end{aligned}$$

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0.7 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.4 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Positively regular process

Multitype branching process is **positively regular**, if there exists $n \in \mathbb{N}$, such that the n-th power of the matrix of first moments M^n

Theorem: extinction probability

Let multitype branching process be non-singular, positively regular and let ρ be spectral radius of corresponding matrix M of first moments. Then for extinction probability vector q it holds:
(i) If $\rho \leq 1$, then $\mathrm{q} = 1$.
(ii) If $\rho > 1$, then $\mathrm{q} < 1$.

In terms of probabilistic grammars:
$\rho \leq 1 \iff Coverage_G = 1$ ($\iff$ restriction to new sample space of finite trees is possible).

Theorem: extinction probability

Let multitype branching process be non-singular, positively regular and let ρ be spectral radius of corresponding matrix M of first moments. Then for extinction probability vector q it holds:
(i) If $\rho \leq 1$, then $\mathrm{q} = 1$.
(ii) If $\rho > 1$, then $\mathrm{q} < 1$.

In terms of probabilistic grammars:
$\rho \leq 1 \iff Coverage_G = 1$ ($\iff$ restriction to new sample space of finite trees is possible).

Theorem: extinction probability

Let multitype branching process be non-singular, positively regular and let ρ be spectral radius of corresponding matrix M of first moments. Then for extinction probability vector q it holds:
(i) If $\rho \leq 1$, then $\mathrm{q} = 1$.
(ii) If $\rho > 1$, then $\mathrm{q} < 1$.

In terms of probabilistic grammars:
$\rho \leq 1 \iff Coverage_G = 1$ ($\iff$ restriction to new sample space of finite trees is possible).

Theorem: Sevast'yanov

Let multitype branching process be without **final classes** and let ρ be spectral radius of corresponding matrix M of first moments. Then for extinction probability vector q it holds:
(i) If $\rho \leq 1$, then $\mathrm{q} = 1$.
(ii) If $\rho > 1$, then $\mathrm{q} < 1$.

Theorem: extinction probability

Let multitype branching process be non-singular, positively regular and let ρ be spectral radius of corresponding matrix M of first moments. Then for extinction probability vector q it holds:
(i) If $\rho \leq 1$, then $\mathrm{q} = 1$.
(ii) If $\rho > 1$, then $\mathrm{q} < 1$.

In terms of probabilistic grammars:
$\rho \leq 1 \iff Coverage_G = 1$ ($\iff$ restriction to new sample space of finite trees is possible).

Theorem: Sevast'yanov

Let multitype branching process be without **final classes** and let ρ be spectral radius of corresponding matrix M of first moments. Then for extinction probability vector q it holds:
(i) If $\rho \leq 1$, then $\mathrm{q} = 1$.
(ii) If $\rho > 1$, then $\mathrm{q} < 1$.

Integer sequences

ndation is supported by donations from users of the OEIS and by a grant from the Simons Found

THE ON–LINE ENCYCLOPEDIA OF INTEGER SEQUENCES ®

founded in 1964 by N. J. A. Sloane

Search Hints

(Greetings from The On-Line Encyclopedia of Integer Sequences!)

A000045 Fibonacci numbers: F(n) = F(n-1) + F(n-2) with F(0) = 0 and F(1) = 1. 5093
(Formerly M0692 N0256)

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418, 317811, 514229, 832040, 1346269, 2178309, 3524578, 5702887, 9227465, 14930352, 24157817, 39088169, 63245986, 102334155 (list; graph; refs; listen; history; text; internal format)

OFFSET 0,4

COMMENTS Also sometimes called Lamé's sequence.
F(n+2) = number of binary sequences of length n that have no consecutive 0's.
F(n+2) = number of subsets of {1,2,...,n} that contain no consecutive integers.
F(n+1) = number of tilings of a 2 X n rectangle by 2 X 1 dominoes.
F(n+1) = number of matchings (i.e., Hosoya index) in a path graph on n vertices:
F(5)=5 because the matchings of the path graph on the vertices A, B, C, D are

ID	Name
A000009	Expansion of Product $\prod_{m=1}^{\infty}(1+x^m)$
A000040	Prime numbers
A000045	Fibonacci numbers
A000124	Central polygonal numbers
A000219	Number of planar partitions of n
A000292	Tetrahedral numbers
A000720	Number of primes $\leq n$
A001045	Jacobsthal sequence
A001097	Twin primes
A001481	Numbers that are the sum of 2 squares
A001615	Dedekind psi function
A002572	Number of partitions of 1 into n powers of 1/2
A005230	Stern's sequence
A027642	Denominator of Bernoulli number B_n

Tabela: List of 14 sequences

Dataset

- Dataset is adjusted to the discovery of recursive equations of the form:

$$a_n = f(\vec{c}, n, a_{n-1}, \ldots, a_{n-49}) \; .$$

-

$$X := \begin{bmatrix} 1 & a_0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & a_1 \\ 2 & a_1 & a_0 & 0 & 0 & \cdots & 0 & 0 & 0 & a_2 \\ 3 & a_2 & a_1 & a_0 & 0 & \cdots & 0 & 0 & 0 & a_3 \\ 4 & a_3 & a_2 & a_1 & a_0 & \cdots & 0 & 0 & 0 & a_4 \\ 5 & a_4 & a_3 & a_2 & a_1 & \cdots & 0 & 0 & 0 & a_5 \\ \vdots & & & & & \ddots & & & & \vdots \\ 47 & a_{46} & a_{45} & a_{44} & a_{43} & \cdots & a_0 & 0 & 0 & a_{47} \\ 48 & a_{47} & a_{46} & a_{45} & a_{44} & \cdots & a_1 & a_0 & 0 & a_{48} \\ 49 & a_{48} & a_{47} & a_{46} & a_{45} & \cdots & a_2 & a_1 & a_0 & a_{49} \end{bmatrix} .$$

Probabilistic grammar in use:

$$
\begin{array}{lcl}
E & \rightarrow & P/R\ [0.2] \mid P\ [0.8] \\
P & \rightarrow & P + c * R\ [0.4] \mid c * R\ [0.3] \mid c\ [0.3] \\
R & \rightarrow & M\ [0.60024] \mid sqrt\ (\ c * M\ )\ [0.133253] \mid \\
 & & exp\ (\ c * M\ )\ [0.133253] \mid log\ (\ c * M\ )\ [0.133253] \\
M & \rightarrow & M * V\ [0.4] \mid V\ [0.6] \\
V & \rightarrow & n\ [0.5] \mid a_{n-1}\ [0.319672] \mid a_{n-2}\ [0.0965392] \mid a_{n-3}\ [0.0295993] \mid \\
 & & a_{n-4}\ [0.0095173] \mid a_{n-5}\ [0.00349271] \mid a_{n-6}\ [0.00168534] \mid \\
 & & a_{n-7}\ [0.00114312] \mid a_{n-8}\ [0.00098046] \mid a_{n-9}\ [0.000931661] \mid \\
 & & a_{n-10}\ [0.000917021] \mid a_{n-11}\ [0.000912629] \mid a_{n-12}\ [0.000911311] \mid \\
 & & a_{n-13}\ [0.000910916] \mid a_{n-14}\ [0.000910798] \mid a_{n-15}\ [0.000910762] \mid \\
 & & a_{n-16}\ [0.000910751] \mid a_{n-17}\ [0.000910748] \mid a_{n-18}\ [0.000910747] \mid \\
 & & a_{n-19}\ [0.000910747] \mid a_{n-20}\ [0.000910747] \mid a_{n-21}\ [0.000910747] \mid \\
 & & \vdots \\
 & & a_{n-49}\ [0.000910747] \quad .
\end{array}
$$

Probabilistic grammar

- Expressions are of the form P/R, where P is polynomial and R is monomial, where in place of monomials we allow functions sqrt, exp and log.
- Probabilities of rewriting rules of the form $(V \to a_{n-k})$ are descending in a way that their ratio is $r : r^2 : r^3 : \cdots : r^{49}$, where $r = \frac{3}{10}$.

Settings of the function 'test'

- Equation error is defined in the same way as on first slides.
- Optimization algorithm of differential evolution.
- Interval for optimization was set to $[-4, 4]$.

Number of generated equations

- $N = 100$

Results

Sequence ID	Equations discovered
$A000009$	none
$A000040$	none
A000045	$a_n = 0.4472 * exp(0.4812 * n)$ $a_n = 0.9886 * a_{n-1} + 1.018 * a_{n-2}$ $a_n = 2.618 * a_{n-2} + 0.03653$ $a_n = 1.591 * a_{n-1} + 0.07085 * a_{n-3} + 0.01706$
A000124	$a_n = 0.992 * a_{n-1} + 0.007 * a_{n-2} + 1.007 * n - 0.006$ $a_n = 2.081 * a_{n-1} - 1.083 * a_{n-2}$
$A000219$	none
$A000292$	$a_n = 1.999 * a_{n-1} - 0.999 * a_{n-2} + 1.000 * n - 0.000$
$A000720$	none
A001045	$a_n = 1.000 * a_{n-1} + 1.998 * a_{n-2}$ $a_n = 2.956 * a_{n-1} - 3.824 * a_{n-3} - 0.029$
$A001097$	none
$A001481$	none
$A001615$	none
$A002572$	none
$A005230$	none
$A027642$	none

Sources and literature

- J. Blazej, *Procesi razvejanja*, diplomsko delo, Fakulteta za matematiko in fiziko, Univerza v Ljubljani, 2016.
- J. Brence, *From Deterministic to Probabilistic Approaches in Automated Modeling*, Ljubljana, 2020.
- J. Brence, L. Todorovski in S. Džeroski, *Probabilistic grammars for equation discovery*, Knowledge-Based Systems **224** (2021), doi: 10.1016/j.knosys.2021. 107077.
- Z. Chi, *Statistical properties of probabilistic context-free grammars*, Computational Linguistics **25**(1) (1999) 131–160, doi: 10.5555/973215.973219.
- M. Collins, *Probabilistic Context-Free Grammars (PCFGs)*, [ogled 9. 6. 2021], dostopno na `www.cs.columbia.edu/~mcollins/courses/nlp2011/notes/pcfgs.pdf`.

- T. Csernica, *Extinction in single and multi-type branching processes*, [ogled 9. 6. 2021], dostopno na `http://math.uchicago.edu/~may/REU2015/REUPapers/Csernica.pdf`.
- T. Harris, *The Theory of Branching Processes*, Grundlehren der mathematischen Wissenschaften **119**, Springer-Verlag Berlin Heidelberg, Berlin, 1963.
- S. G. in M. Johnson, *Probabilistic grammars and their applications*, International Encyclopedia of the Social & Behavioral Sciences **158**, Pergamon, Oxford, 2001.
- M. Jiřina, *A simplified proof of the sevastyanov theorem on branching processes*, Annales de l'I.H.P. Probabilités et statistiques **6**(1) (1970) 1–7.
- M. Miller, *Entropies and combinatorics of random branching processes and context-free languages*, IEEE Transactions on Information Theory **38**(4) (1992) 1292–1310, doi: 10.1109/18.144710.

- C. J. Mode, *Multitype Branching Processes: Theory and Applications*, Modern analytic and computational methods in science and mathematics, American Elsevier Publishing Company, New York, 1971.
- M. Schmidt in H. Lipson, *Distilling free-form natural laws from experimental data*, Science (New York, N.Y.) **324** (2009) 81–85, doi: 10.1126/science. 1165893.
- N. J. A. Sloane, *The On-Line Encyclopedia of Integer Sequences*, [ogled 1. 6. 2021], dostopno na `https://oeis.org`.
- R. Storn in K. Price, *Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces*, Journal of Global Optimization **11** (1997) 341–359, doi: 10.1023/A:1008202821328.
- L. Todorovski, *Declarative Bias of Equation Discovery*, magistsko delo, Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, 1998.

- L. Todorovski, *Odkrivanje enačb in predznanje*, Ljubljana, 2021, prosojnice pri predmetu Napredno strojno učenje, [ogled 14. 7. 2021], dostopno na `http://kt.ijs.si/~ljupco/lectures/nsu/11-ed-knowledge.pdf`.
- L. Todorovski, *Uvod v odkrivanje enačb in simbolno regresijo*, Ljubljana, 2021, prosojnice pri predmetu Napredno strojno učenje, [ogled 14. 7. 2021], dostopno na `http://kt.ijs.si/~ljupco/lectures/nsu/10-ed-intro.pdf`.
- L. Todorovski in S. Džeroski, *Declarative bias in equation discovery*, v: ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning, Morgan Kaufmann Publishers, San Francisco, ZDA, 1997, str. 376–384, doi: 10.5555/645526.657279.

Theorem: Sevast'yanov

Let multitype branching process be without **final classes** and let ρ be spectral radius of corresponding matrix M of first moments. Then for extinction probability vector q it holds:
(i) If $\rho \leq 1$, then $\mathrm{q} = 1$.
(ii) If $\rho > 1$, then $\mathrm{q} < 1$.

Theorem: Sevast'yanov

Let ρ be spectral radius of matrix M of first moments of multitype branching process. Then extinction probability vector equals to $\mathrm{q} = 1$ if and only if:
(a) $\rho \leq 1$ and
(b) it has no **final classes**.

▸ -