Independent Sets of Families of Graphs by Finite State Automata

Giovanna Janet Lavado

Joint work with:
P. Codara, O.M. D'Antona and M. Galasi

Università degli Studi di Milano

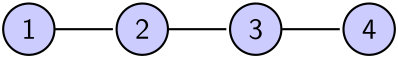
13 January, 2017

Independent sets

$G = (V, E)$

Independent Sets [Codara&D'Antona '13]

An *independent subset* of a graph G is a subset of V not containing adjacent vertices.



$\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 3\}, \{1, 4\}, \{2, 4\}$

Independent sets of powers of paths $P_n^{(h)}$

For $n, h, k \geq 0$,

the number of independent k-subsets of $P_n^{(h)}$: [Hoggatt '70]

$$p_{n,k}^{(h)} = \binom{n - hk + h}{k}$$

$h = 1$

$p_{n,k}^{(1)}$	$k = 0$	1	2	3
$n = 0$	1			
1	1	1		
2	1	2		
3	1	3	1	
4	1	4	3	
5	1	5	6	1

n	0	1	2	3	4	5	6	7
IND_n	1	2	3	5	8	13	21	34

Motivations and research

NP-hard problems:

- The independent set decision problem
- The maximum independent set problem

Combinatorial problems:

- Enumerating independent sets of power of paths and cycles [Hoggatt '70, Codara&D'Antona '13]
- Number of independent sets of a grid graph [Calkin&Wilf '98]

Our goal

We are looking for a bijection between the *independent sets* of families of graphs and the *words accepted by a finite state automaton*

Preliminaries: regular languages

Regular grammar

$$\mathcal{G} = (V, \Sigma, S, P)$$

- V is a finite set of *variables*
- Σ is a finite set of *terminals* ($V \cap \Sigma = \varnothing$)
- $S \in V$ is the *start symbol* or *axiom*
- P is a finite set of rules or *productions* of the form
  - $A \to a$
  - $A \to aB$

  where $A, B \in V$, $a \in \Sigma$

$$L(\mathcal{G}) = \left\{ w \in \Sigma^* \mid S \overset{*}{\Rightarrow} w \right\}$$

Preliminaries: regular languages

Finite state automaton

$$\mathcal{M} = (Q, \Sigma, \delta, q_0, F)$$

- Q is a finite set of *states*
- Σ is a finite *input alphabet*
- $\delta : Q \times \Sigma \to Q$ is a *transition function*
- $q_0 \in Q$ is the *initial state*
- $F \subseteq Q$ is a set of *final states*

$$L(\mathcal{M}) = \{w \in \Sigma^* \mid \delta(q_0, w) \in F\}$$

Regular languages

$\Sigma = \{e, a\}$

$$L = \{\lambda, e, a, ee, ea, ae, eee, aea, eae, eea, \ldots\}$$

Forbidden factor: aa

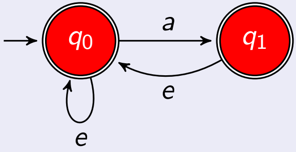
The regular grammar $\mathcal{G}$

$$S \to eS \mid aA \mid \lambda$$

$$A \to eS \mid \lambda$$

$$S \Longrightarrow aA \Longrightarrow aeS \Longrightarrow ae$$

The automaton $\mathcal{M}$



Regular languages

$\Sigma = \{e, a\}$

$$L = \{\lambda, e, a, ee, ea, ae, eee, aea, eae, eea, \ldots\}$$

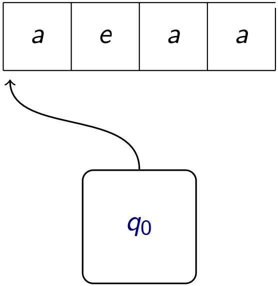
Forbidden factor: aa

Finite state automaton $\mathcal{M}$

Input: $w \in \Sigma^*$

Question: $w \in L(\mathcal{M})$?

Answer: Yes / No



Regular languages

$\Sigma = \{e, a\}$

$$L = \{\lambda, e, a, ee, ea, ae, eee, aea, eae, eea, \dots\}$$

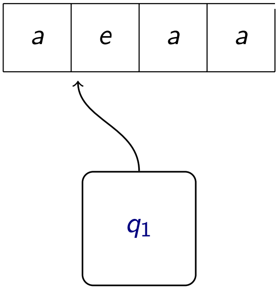
Forbidden factor: aa

Finite state automaton $\mathcal{M}$

Input: $w \in \Sigma^*$

Question: $w \in L(\mathcal{M})$?

Answer: Yes / No

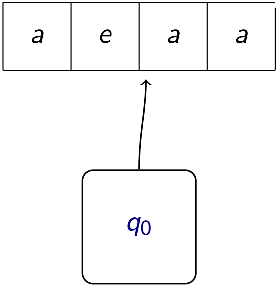


Regular languages

$\Sigma = \{e, a\}$

$$L = \{\lambda, e, a, ee, ea, ae, eee, aea, eae, eea, \ldots\}$$

Forbidden factor: aa



Finite state automaton $\mathcal{M}$

Input: $w \in \Sigma^*$

Question: $w \in L(\mathcal{M})$?

Answer: Yes / No

Regular languages

$\Sigma = \{e, a\}$

$$L = \{\lambda, e, a, ee, ea, ae, eee, aea, eae, eea, \ldots\}$$

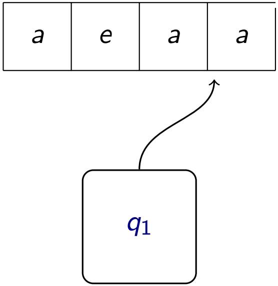
Forbidden factor: aa

Finite state automaton $\mathcal{M}$

Input: $w \in \Sigma^*$

Question: $w \in L(\mathcal{M})$?

Answer: Yes / No



Regular languages

$\Sigma = \{e, a\}$

$$L = \{\lambda, e, a, ee, ea, ae, eee, aea, eae, eea, \ldots\}$$

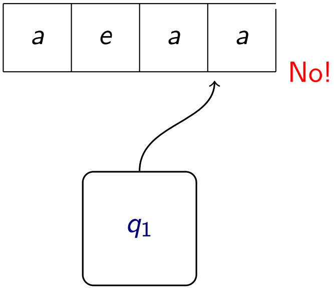
Forbidden factor: aa

Finite state automaton $\mathcal{M}$

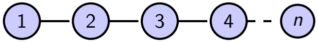
Input: $w \in \Sigma^*$

Question: $w \in L(\mathcal{M})$?

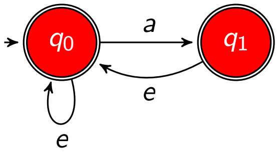
Answer: Yes / No



Bijection



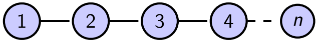
$\Longleftrightarrow$



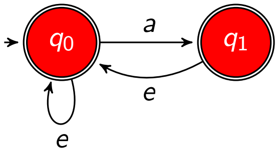
?

They have the same generating function!!!

Bijection



$\Longleftrightarrow$

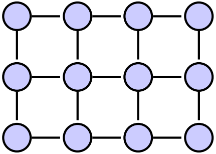


?

They have the same generating function!!!

Method of Calkin and Wilf

$G_{m,n}$



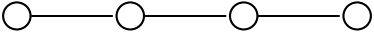
Transfer matrix [Calkin&Wilf '98]

The transfer matrix of a grid graph $G_{m,n}$ is a matrix $Fib_{m+1} \times Fib_{m+1}$ (as many rows and as many columns as the independent sets of a vertical path in $G_{m,n}$ are) such that:

$$T[i,j] = \begin{cases} 1 & \text{if } (i,j) \text{ form an independent set} \\ 0 & \text{otherwise} \end{cases}$$

$$Fib_0 = 1, Fib_1 = 1, \forall m > 1, Fib_m = Fib_{m-1} + Fib_{m-2}$$

Method of Calkin and Wilf: generating function



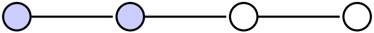
Taking a vertical path we have two independent sets:

Transfer matrix:

$T =$

	e	a
e	1	1
a	1	0

Method of Calkin and Wilf: generating function

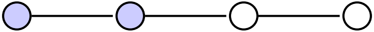


Taking a vertical path we have two independent sets:

Transfer matrix:

$$T = \begin{array}{c|cc} & e & a \\ \hline e & 1 & 1 \\ a & 1 & \color{red}{0} \end{array}$$

Method of Calkin and Wilf: generating function



Taking a vertical path we have two independent sets:

Transfer matrix:

$T =$

	e	a
e	1	1
a	1	0

Method of Wilf:

$$W(x) = (I - xT)^{-1}$$

where I is the identity matrix.

Method of Calkin and Wilf: generating function

$$(I - xT)^{-1} = \begin{bmatrix} 1-x & -x \\ -x & 1 \end{bmatrix}^{-1} = \frac{1}{1-x-x^2}\begin{bmatrix} 1 & x \\ x & 1-x \end{bmatrix}$$

Summing the elements of the matrix:

$$W(x) = \frac{2+x}{1-x-x^2}$$

Method of Calkin and Wilf: generating function

Considering the empty graph:

$$1 + xW(x) = 1 + \frac{2x + x^2}{1 - x - x^2} = \frac{1 + x}{1 - x - x^2}$$

Generating function:

$$S(x) = \frac{1 + x}{1 - x - x^2}$$

$$S(x) = 1 + 2x + 3x^2 + 5x^3 + 8x^4 + 13x^5 + \mathcal{O}(x^6)$$

Method of Calkin and Wilf: generating function

Considering the empty graph:

$$1 + xW(x) = 1 + \frac{2x + x^2}{1 - x - x^2} = \frac{1 + x}{1 - x - x^2}$$

Generating function:

$$S(x) = \frac{1 + x}{1 - x - x^2}$$

$$S(x) = 1 + 2x + 3x^2 + 5x^3 + 8x^4 + 13x^5 + \mathcal{O}(x^6)$$

$\varnothing$

Method of Calkin and Wilf: generating function

Considering the empty graph:

$$1 + xW(x) = 1 + \frac{2x + x^2}{1 - x - x^2} = \frac{1 + x}{1 - x - x^2}$$

Generating function:

$$S(x) = \frac{1 + x}{1 - x - x^2}$$

$$S(x) = 1 + 2x + 3x^2 + 5x^3 + 8x^4 + 13x^5 + \mathcal{O}(x^6)$$

$$\varnothing, \{1\}$$

Method of Calkin and Wilf: generating function

Considering the empty graph:

$$1 + xW(x) = 1 + \frac{2x + x^2}{1 - x - x^2} = \frac{1 + x}{1 - x - x^2}$$

Generating function:

$$S(x) = \frac{1 + x}{1 - x - x^2}$$

$$S(x) = 1 + 2x + 3x^2 + 5x^3 + 8x^4 + 13x^5 + \mathcal{O}(x^6)$$

$$\varnothing, \{1\}, \{2\}$$

Method of Calkin and Wilf: generating function

Considering the empty graph:

$$1 + xW(x) = 1 + \frac{2x + x^2}{1 - x - x^2} = \frac{1 + x}{1 - x - x^2}$$

Generating function:

$$S(x) = \frac{1 + x}{1 - x - x^2}$$

$$S(x) = 1 + 2x + 3x^2 + 5x^3 + 8x^4 + 13x^5 + \mathcal{O}(x^6)$$

$$\varnothing, \{1\}, \{2\}, \{3\}, \{1, 3\}$$

Method with finite state automata

[Chomsky&Schützenberger '63, Gruger&Lee&Shallit '05]

$$S \to eS \mid aA \mid \lambda \qquad A \to eS \mid \lambda$$

Set of equations from the grammar:

$$\begin{cases} S(t) = & tS(t) + tA(t) + 1 \\ A(t) = & tS(t) + 1 \end{cases}$$

Generating function:

$$S(t) = \frac{1+t}{1-t-t^2}$$

$$S(t) = 1 + 2t + 3t^2 + 5t^3 + 8t^4 + 13t^5 + \mathcal{O}(t^6)$$

Method with finite state automata

[Chomsky&Schützenberger '63, Gruger&Lee&Shallit '05]

$$S \to eS \mid aA \mid \lambda \qquad A \to eS \mid \lambda$$

Set of equations from the grammar:

$$\begin{cases} S(t) = & tS(t) + tA(t) + 1 \\ A(t) = & tS(t) + 1 \end{cases}$$

Generating function:

$$S(t) = \frac{1+t}{1-t-t^2}$$

$$S(t) = 1 + 2t + 3t^2 + 5t^3 + 8t^4 + 13t^5 + \mathcal{O}(t^6)$$

$$\lambda$$

Method with finite state automata

[Chomsky&Schützenberger '63, Gruger&Lee&Shallit '05]

$$S \to eS \mid aA \mid \lambda \qquad A \to eS \mid \lambda$$

Set of equations from the grammar:

$$\begin{cases} S(t) = & tS(t) + tA(t) + 1 \\ A(t) = & tS(t) + 1 \end{cases}$$

Generating function:

$$S(t) = \frac{1+t}{1-t-t^2}$$

$$S(t) = 1 + 2t + 3t^2 + 5t^3 + 8t^4 + 13t^5 + \mathcal{O}(t^6)$$

e, a

Method with finite state automata

[Chomsky&Schützenberger '63, Gruger&Lee&Shallit '05]

$$S \to eS \mid aA \mid \lambda \qquad A \to eS \mid \lambda$$

Set of equations from the grammar:

$$\begin{cases} S(t) = & tS(t) + tA(t) + 1 \\ A(t) = & tS(t) + 1 \end{cases}$$

Generating function:

$$S(t) = \frac{1+t}{1-t-t^2}$$

$$S(t) = 1 + 2t + 3t^2 + 5t^3 + 8t^4 + 13t^5 + \mathcal{O}(t^6)$$

ee, ae, ea

Method with finite state automata

[Chomsky&Schützenberger '63, Gruger&Lee&Shallit '05]

$$S \to eS \mid aA \mid \lambda \qquad A \to eS \mid \lambda$$

Set of equations from the grammar:

$$\begin{cases} S(t) = & tS(t) + tA(t) + 1 \\ A(t) = & tS(t) + 1 \end{cases}$$

Generating function:

$$S(t) = \frac{1+t}{1-t-t^2}$$

$$S(t) = 1 + 2t + 3t^2 + 5t^3 + 8t^4 + 13t^5 + \mathcal{O}(t^6)$$

eee, *eea*, *aee*, *aea*, *eae*

Method with finite state automata

[Chomsky&Schützenberger '63, Gruger&Lee&Shallit '05]

$$S \to eS \mid aA \mid \lambda \qquad A \to eS \mid \lambda$$

Set of equations from the grammar:

$$\begin{cases} S(t) = & tS(t) + tA(t) + 1 \\ A(t) = & tS(t) + 1 \end{cases}$$

Generating function:

$$S(t) = \frac{1+t}{1-t-t^2}$$

$$S(t) = 1 + 2t + 3t^2 + 5t^3 + 8t^4 + 13t^5 + \mathcal{O}(t^6)$$

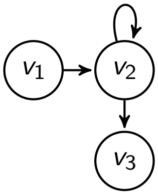
eee, *eea*, *aee*, *aea*, *eae*

Our idea to use automata is a new original approach!

Why?

Interesting property of the adjacency matrix

Let A be the adjacency matrix of a graph G, then $A^n[i,j]$ is the number of paths of length n to get from vertex i to vertex j.



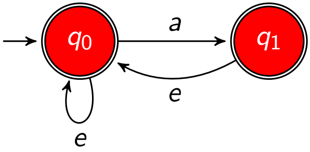
$A =$

	v_1	v_2	v_3
v_1	0	1	0
v_2	0	1	1
v_3	0	0	0

$A^2 =$

	v_1	v_2	v_3
v_1	0	1	1
v_2	0	1	1
v_3	0	0	0

Matrix representation of automata



$M =$

	q_0	q_1
q_0	1	1
q_1	1	0

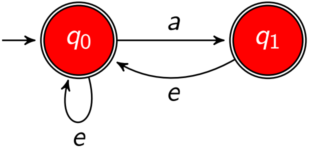
$M^2 =$

	q_0	q_1
q_0	2	1
q_1	1	1

$$\delta(q_0, ee) = q_0 \qquad \delta(q_0, ae) = q_0$$

$$(I - xM)^{-1} = \frac{I}{I - xM} = \sum_{n \geq 0} (xM)^n = I + xM + x^2M^2 + \dots$$

Matrix representation of automata



$M =$

	q_0	q_1
q_0	1	1
q_1	1	0

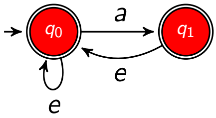
$M^2 =$

	q_0	q_1
q_0	2	1
q_1	1	1

$\delta(q_0, ee) = q_0 \qquad \delta(q_0, ae) = q_0$

$$(I - xM)^{-1} = \frac{I}{I - xM} = \sum_{n \geq 0} (xM)^n = I + xM + x^2 M^2 + \dots$$

Matrix representation of automata



$$\frac{I}{I - xM} = I + xM + x^2M^2 + \cdots =$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} x & x \\ x & 0 \end{bmatrix} + \begin{bmatrix} 2x^2 & x^2 \\ x^2 & x^2 \end{bmatrix} + \cdots$$

$$= \begin{bmatrix} 1 + x + 2x^2 + \mathcal{O}(x^3) & x + x^2 + \mathcal{O}(x^3) \\ x + x^2 + \mathcal{O}(x^3) & 1 + x^2 + \mathcal{O}(x^3) \end{bmatrix}$$

$$F(x) = 2 + 3x + 5x^2 + \mathcal{O}(x^3)$$

Matrix representation of automata

Given the characteristic vector of initial state and the vector of final states,

$$\mathrm{A}_i = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad ; \qquad \Omega_f = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

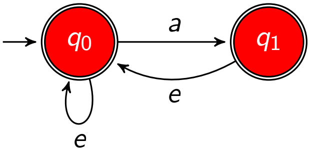
the number of paths from the initial state to final states is:

$$\mathrm{A}_i \frac{I}{I - xM} \Omega_f$$

Generating function:

$$S(x) = 1 + 2x + 3x^2 + \mathcal{O}(x^3)$$

The regular expression



System of equations in non-commutative variables:

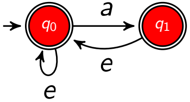
$$\begin{cases} L_0 = & eL_0 + aL_1 + \lambda \\ L_1 = & eL_0 + \lambda \end{cases}$$

$$L_0 = \frac{\lambda}{\lambda - (e + ae)}(a + \lambda)$$

$$\frac{\lambda}{\lambda - (e + ae)} = \sum_{n \geq 0} (e+ae)^n = \lambda + (e+ae) + (e+ae)^2 + \cdots = (e+ae)^*$$

$$L(\mathcal{M}) = (e + ae)^*(a + \lambda)$$

The bijection



n	$\mathcal{M}$	$*$	$P_n^{(1)}$
1	q_0	e	$\varnothing$
	q_1	a	$\{1\}$
2	$q_0 q_0$	ee	$\varnothing$
	$q_0 q_1$	ea	$\{2\}$
	$q_1 q_0$	ae	$\{1\}$
3	$q_0 q_0 q_0$	eee	$\varnothing$
	$q_0 q_0 q_1$	eea	$\{3\}$
	$q_0 q_1 q_0$	eae	$\{2\}$
	$q_1 q_0 q_0$	aee	$\{1\}$
	$q_1 q_0 q_1$	aea	$\{1, 3\}$

Summarizing

$$S(t) = \frac{1+t}{1-t-t^2} = 1 + 2t + 3t^2 + 5t^3 + 8t^4 + \mathcal{O}(t^5)$$

- The generating functions obtained using the method of Calkin/Wilf and the method with the automata are the same
- We have shown a bijection between the independent sets of paths of power $h = 1$ and the words accepted by an automaton
- The transfer matrix has been designed to work with grid graphs and does not work directly when $h \geq 2$

How to proceed if one of the above conditions is not fulfilled?

A possible solution is using a new concept called *telescopic family of graphs*

Summarizing

$$S(t) = \frac{1+t}{1-t-t^2} = 1 + 2t + 3t^2 + 5t^3 + 8t^4 + \mathcal{O}(t^5)$$

- The generating functions obtained using the method of Calkin/Wilf and the method with the automata are the same
- We have shown a bijection between the independent sets of paths of power $h = 1$ and the words accepted by an automaton
- The transfer matrix has been designed to work with grid graphs and does not work directly when $h \geq 2$

How to proceed if one of the above conditions is not fulfilled?

A possible solution is using a new concept called *telescopic family of graphs*

Summarizing

$$S(t) = \frac{1+t}{1-t-t^2} = 1 + 2t + 3t^2 + 5t^3 + 8t^4 + \mathcal{O}(t^5)$$

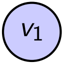
- The generating functions obtained using the method of Calkin/Wilf and the method with the automata are the same
- We have shown a bijection between the independent sets of paths of power $h = 1$ and the words accepted by an automaton
- The transfer matrix has been designed to work with grid graphs and does not work directly when $h \geq 2$

How to proceed if one of the above conditions is not fulfilled?

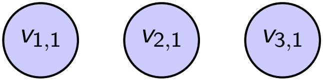
A possible solution is using a new concept called *telescopic family of graphs*

Telescopic family of graphs

- M, the *module*, is any (non-empty) graph, and $v_1, v_2, \ldots, v_m$ its vertices, e.g., $M = K_1$:



- h, the *power*, is a non-negative integer and $F_{M,h}$ is a *layered graph* of the $h+1$ replicas of M, e.g., $h = 2$:

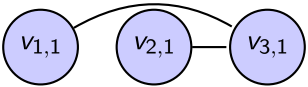


- X, the *cross-connection*, is any subset of *inter-layer edges* of $F_{M,h}$, e.g.:

$$X = \{(v_{1,1}, v_{3,1}), (v_{2,1}, v_{3,1})\}$$

Telescopic family of graphs

A graph C, the *connection*, is obtained by adding the inter-layer edges of X to the h-frame of M:



Telescopic family of graphs

A *telescopic family of graphs*, TFG, is a sequence of graphs $\{G_n\}_{n\geq 0}$ identified by a triplet (M, h, X)

Telescopic family of graphs

The graphs of the TFG $\{G_n\}_{n\geq 0}$ identified by (M, h, X) with $M = K_1$, $h = 2$ and $X = \{(v_{1,1}, v_{3,1}), (v_{2,1}, v_{3,1})\}$ are:

Telescopic family of graphs

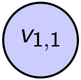
The graphs of the TFG $\{G_n\}_{n\geq 0}$ identified by (M, h, X) with $M = K_1$, $h = 2$ and $X = \{(v_{1,1}, v_{3,1}), (v_{2,1}, v_{3,1})\}$ are:

$n = 0 \quad \emptyset$

Telescopic family of graphs

The graphs of the TFG $\{G_n\}_{n \geq 0}$ identified by (M, h, X) with $M = K_1$, $h = 2$ and $X = \{(v_{1,1}, v_{3,1}), (v_{2,1}, v_{3,1})\}$ are:

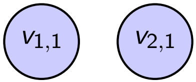
$n = 1$



Telescopic family of graphs

The graphs of the TFG $\{G_n\}_{n\geq 0}$ identified by (M, h, X) with $M = K_1$, $h = 2$ and $X = \{(v_{1,1}, v_{3,1}), (v_{2,1}, v_{3,1})\}$ are:

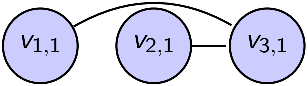
$n = 2$



Telescopic family of graphs

The graphs of the TFG $\{G_n\}_{n\geq 0}$ identified by (M, h, X) with $M = K_1$, $h = 2$ and $X = \{(v_{1,1}, v_{3,1}), (v_{2,1}, v_{3,1})\}$ are:

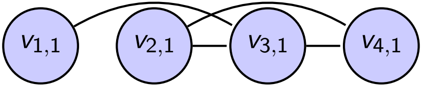
$n = 3$



Telescopic family of graphs

The graphs of the TFG $\{G_n\}_{n\geq 0}$ identified by (M, h, X) with $M = K_1$, $h = 2$ and $X = \{(v_{1,1}, v_{3,1}), (v_{2,1}, v_{3,1})\}$ are:

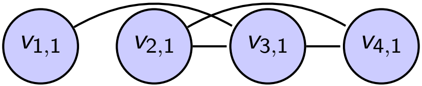
$n = 4$



Telescopic family of graphs

The graphs of the TFG $\{G_n\}_{n\geq 0}$ identified by (M, h, X) with $M = K_1$, $h = 2$ and $X = \{(v_{1,1}, v_{3,1}), (v_{2,1}, v_{3,1})\}$ are:

$n = 4$



- $G_1 \simeq M$
- $G_{h+1} = C$
- For $n \geq 0$, G_n is the subgraph of G_{n+1} induced by $V_1 \cup V_2 \cup \cdots \cup V_n$, where V_i is the set of vertices of the i-th replica of M

Telescopic family of graphs

The number $T_{n,k}$ of independent k-subsets of the TFG $\{G_n\}_{n\geq 0}$:

$T_{n,k}$	$k=0$	1	2	3
$n=0$	1			
1	1	1		
2	1	2	1	
3	1	3	1	
4	1	4	2	
5	1	5	4	1
6	1	6	7	2
7	1	7	11	4

and the number IND_n of all independent sets of G_n:

n	0	1	2	3	4	5	6	7
IND_n	1	2	4	5	7	11	16	23

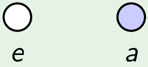
Building a finite state automaton from a TFG

The alphabet Σ

The alphabet is obtained by assigning a symbol to each independent set of the module M

Example

- We have two independent sets from the module $M = K_1$:



- $\Sigma = \{e, a\}$

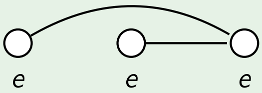
Building a finite state automaton from a TFG

The states Q

The *legal words* are those strings $w \in \Sigma^*$ of length at most $h+1$ which are associated with the independent sets of the graphs $G_0, G_1, \ldots, G_{h+1}$

Example

- eee, eea are legal words, while eaa is not a legal word:



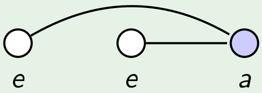
Building a finite state automaton from a TFG

The states Q

The *legal words* are those strings $w \in \Sigma^*$ of length at most $h+1$ which are associated with the independent sets of the graphs $G_0, G_1, \ldots, G_{h+1}$

Example

- *eee*, *eea* are legal words, while *eaa* is not a legal word:



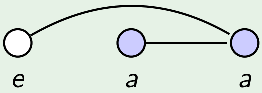
Building a finite state automaton from a TFG

The states Q

The *legal words* are those strings $w \in \Sigma^*$ of length at most $h+1$ which are associated with the independent sets of the graphs $G_0, G_1, \ldots, G_{h+1}$

Example

- *eee*, *eea* are legal words, while *eaa* is not a legal word:



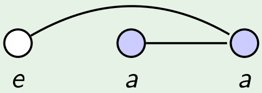
Building a finite state automaton from a TFG

The states Q

The *legal words* are those strings $w \in \Sigma^*$ of length at most $h + 1$ which are associated with the independent sets of the graphs $G_0, G_1, \ldots, G_{h+1}$

Example

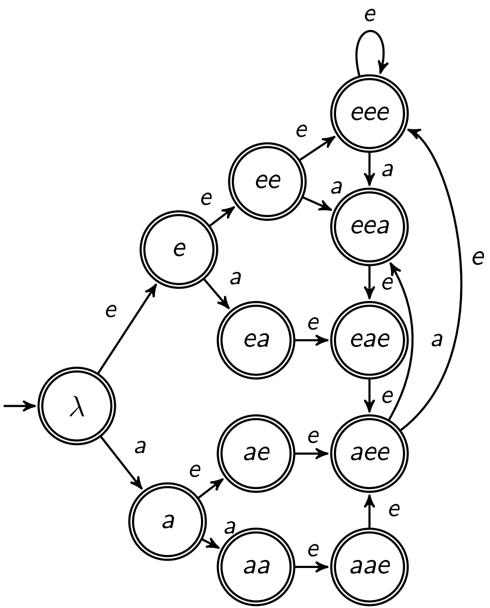
- *eee*, *eea* are legal words, while *eaa* is not a legal word:



- $Q = \{q_w \mid w \text{ is a legal word}\}$
- q_λ is the initial state
- All states are accepting states

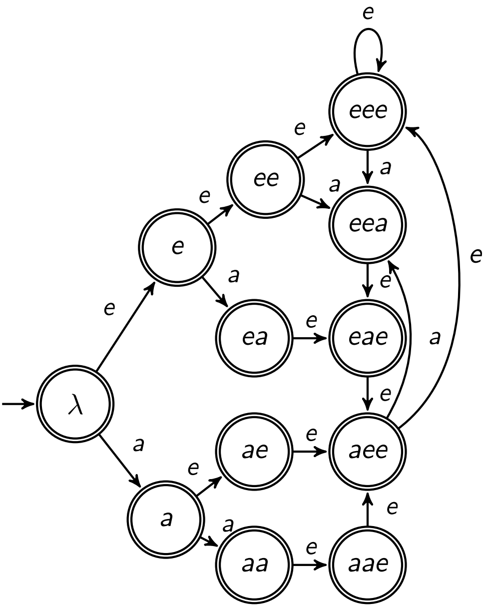
Building a finite state automaton from a TFG

Transition graph



Building a finite state automaton from a TFG

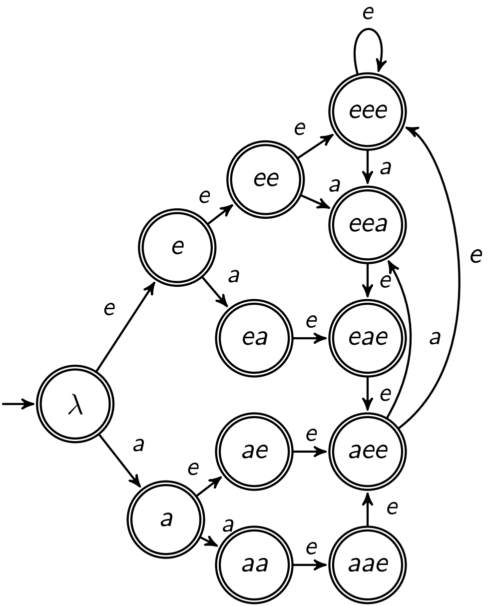
Transition graph



- $\delta(q_{ea}, e) = q_{eae}$

Building a finite state automaton from a TFG

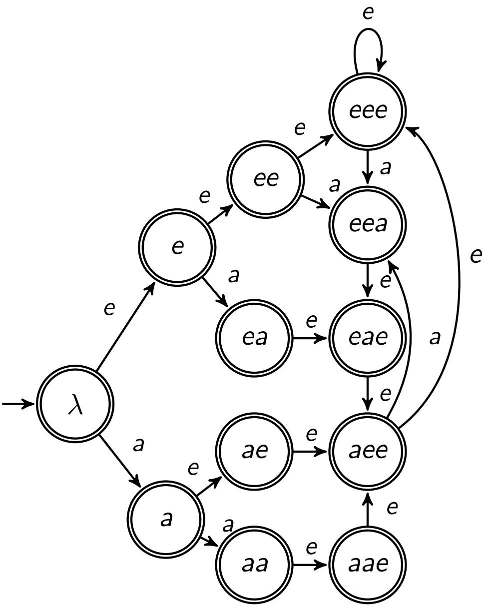
Transition graph



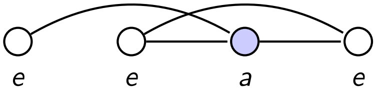
- $\delta(q_{eea}, e) = q_{eae}$

Building a finite state automaton from a TFG

Transition graph



- $\delta(q_{eea}, e) = q_{eae}$



Building a finite state automaton from a TFG

Conclusion

Generating function:

$$S(t) = 1 + 2t + 4t^2 + 5t^3 + 7t^4 + 11t^5 + 16t^6 + 23t^7 + 34t^8 + \mathcal{O}(t^9)$$

- The coefficient of a generic t^n represents the number of words of length n
- These coefficients coincide with the number of all independent sets of the graph G_n

n	0	1	2	3	4	5	6	7
IND_n	1	2	4	5	7	11	16	23

- We are also developing a technique in order to treat cyclic families of graphs

Bibliography I

N. Chomsky and M.P. Schützenberger.
The Algebraic Theory of Context-Free Languages.
In *Studies in Logic and the Foundations of Mathematics*, volume 35, pages 118–161. Elsevier, 1963.

Jonathan Lee and Jeffrey Shallit.
Enumerating Regular Expressions and Their Languages.
In *Implementation and Application of Automata*, volume 3317, pages 2–22. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

Neil J. Calkin and Herbert S. Wilf.
The Number of Independent Sets in a Grid Graph.
SIAM Journal on Discrete Mathematics, 11(1):54–60, 1998.

Bibliography II

Pietro Codara and Ottavio M. D'Antona.
Investigating Independent Subsets of Graphs, with Mathematica.
CoRR, abs/1307.1335, 2013.

Verner E. Hoggatt.
Combinatorial Problems for Generalized Fibonacci Numbers.
Fibonacci Quarterly, 8(5), 456–462, 1970.