CIMPA Summer School 2014

University An Najah, Nablus

Automata and Motif Statistics

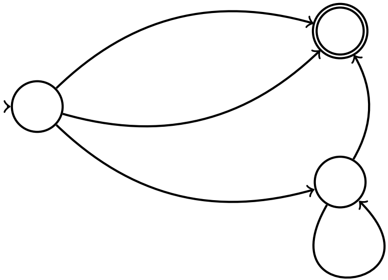
Pierre Nicodème

LIPN, University Paris 13

Motif Statistics - Course I - Counting with Automata

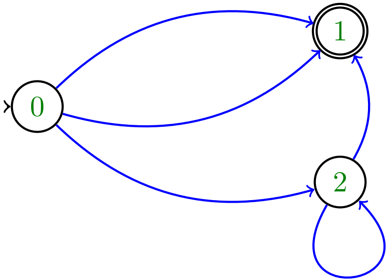
- Basics of Automata theory
- Pattern Matching
- Counting with automata in random texts
- Applications

What is an automaton?



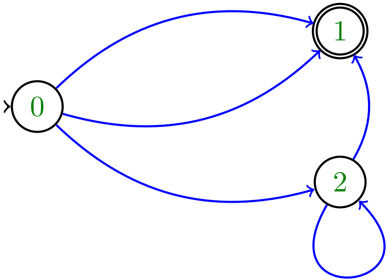
A directed graph

What is an automaton?



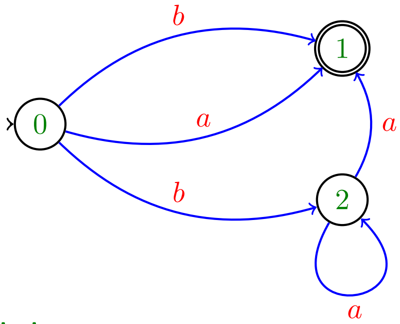
A directed graph
where vertices are called states,

What is an automaton?



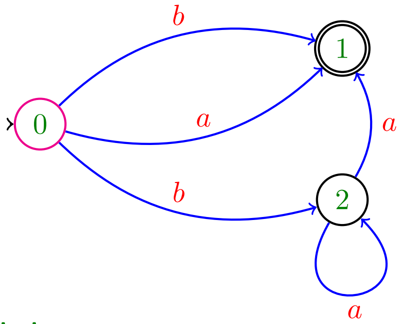
A directed graph
where vertices are called states,
edges are called transitions,

What is an automaton?



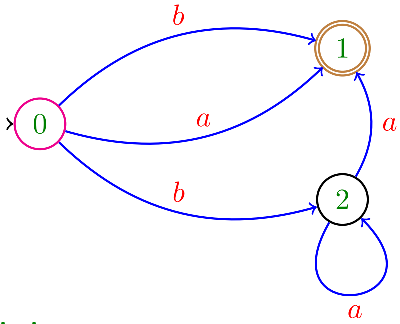
A directed graph
where vertices are called states,
edges are called transitions,
and labelled by letters of a finite alphabet;

What is an automaton?



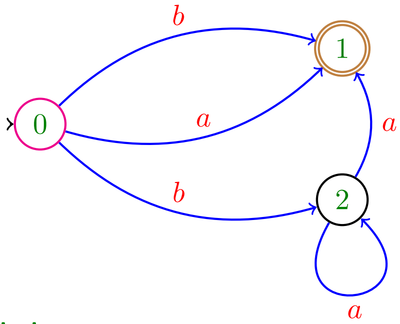
A directed graph
where vertices are called states,
edges are called transitions,
and labelled by letters of a finite alphabet;
there is a specific state called start,

What is an automaton?



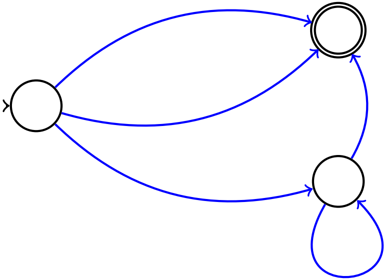
A directed graph
where vertices are called states,
edges are called transitions,
and labelled by letters of a finite alphabet;
there is a specific state called start,
and there are accepting states;

What is an automaton?



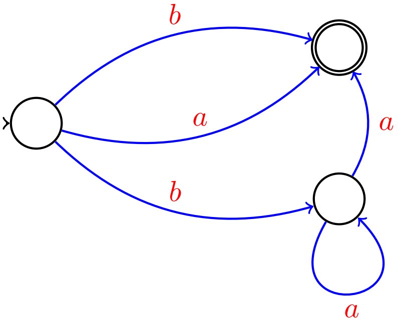
A directed graph
where vertices are called states,
edges are called transitions,
and labelled by letters of a finite alphabet;
there is a specific state called start,
and there are accepting states;
The function mapping the **nodes to their successors**
is called **"transition function"**

What is an automaton?



$$\mathrm{AUTO} = (\mathcal{A}, Q, \mathrm{start}, \delta, F)$$

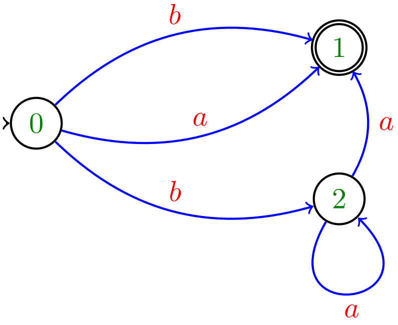
What is an automaton?



$$\mathrm{AUTO} = (\mathcal{A}, Q, \mathrm{start}, \delta, F)$$

1. Alphabet - $\mathcal{A} = \{a, b\}$

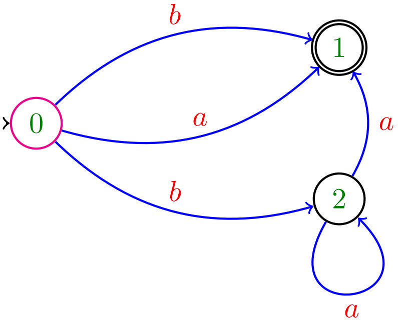
What is an automaton?



$$\mathrm{AUTO} = (\mathcal{A}, Q, \mathrm{start}, \delta, F)$$

1. Alphabet - $\mathcal{A} = \{a, b\}$
2. Set of States - $Q = \{1, 2, 3\}$

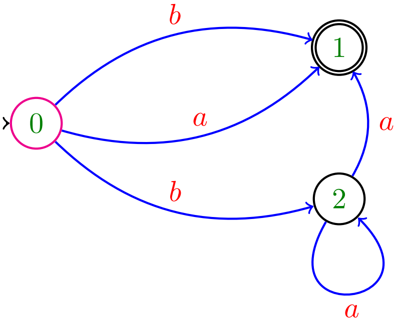
What is an automaton?



$$\mathrm{AUTO} = (\mathcal{A}, Q, \mathrm{start}, \delta, F)$$

1. Alphabet - $\mathcal{A} = \{a, b\}$
2. Set of States - $Q = \{1, 2, 3\}$
3. $\mathrm{start} = \{0\}$

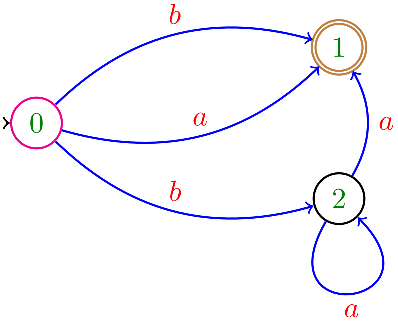
What is an automaton?



$$\text{AUTO} = (\mathcal{A}, Q, \text{start}, \delta, F)$$

1. Alphabet - $\mathcal{A} = \{a, b\}$
2. Set of States - $Q = \{1, 2, 3\}$
3. $\text{start} = \{0\}$
4. Transition function δ: $\begin{cases} \delta(0,a) = \{1\} & \delta(0,b) = \{1,2\} \\ \delta(1,a) = \{\} & \delta(1,b) = \{\} \\ \delta(2,a) = \{2,1\} & \delta(2,b) = \{\} \end{cases}$

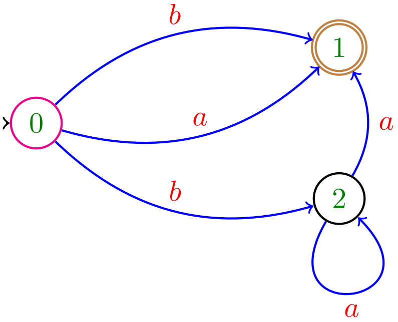
What is an automaton?



$$\text{AUTO} = (\mathcal{A}, Q, \text{start}, \delta, F)$$

1. Alphabet - $\mathcal{A} = \{a, b\}$
2. Set of States - $Q = \{1, 2, 3\}$
3. $\text{start} = \{0\}$
4. Transition function δ: $\begin{cases} \delta(0, a) = \{1\} & \delta(0, b) = \{1, 2\} \\ \delta(1, a) = \{\} & \delta(1, b) = \{\} \\ \delta(2, a) = \{2, 1\} & \delta(2, b) = \{\} \end{cases}$
5. Accepting states: $F = \{1\}$

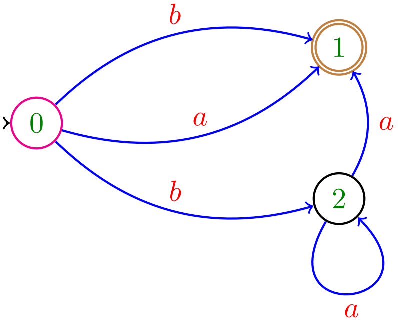
What is an automaton?



$$\mathrm{AUTO} = (\mathcal{A}, Q, \mathrm{start}, \delta, F)$$

- A **run** of length n is a sequence $(q_0, q_1, \ldots, q_n)$ such that
  1. $q_0 = \mathrm{start}$
  2. there exists $a_1 a_2 \ldots a_n \in \mathcal{A}^n$ and $q_{i+1} \in \delta(q_i, a_{i+1})$

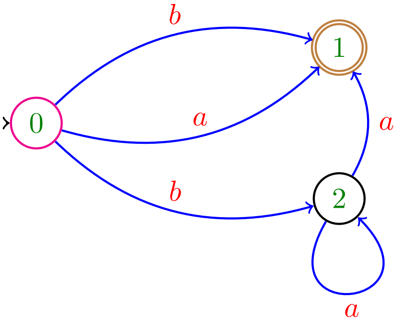
What is an automaton?



$$\mathrm{AUTO} = (\mathcal{A}, Q, \mathrm{start}, \delta, F)$$

- A **run** of length n is a sequence $(q_0, q_1, \ldots, q_n)$ such that
  1. $q_0 = \mathrm{start}$
  2. there exists $a_1 a_2 \ldots a_n \in \mathcal{A}^n$ and $q_{i+1} \in \delta(q_i, a_{i+1})$
- A word $w = a_1 a_2 \ldots a_n$ is **accepted** if there is **at least a run** of length n **spelling its letters** and **ending** in an **accepting state**.

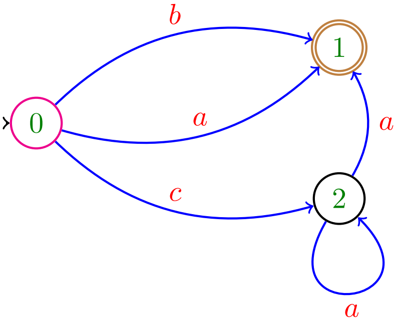
What is an automaton?



$\mathrm{AUTO} = (\mathcal{A}, Q, \mathrm{start}, \delta, F)$

- A **run** of length n is a sequence $(q_0, q_1, \ldots, q_n)$ such that
  1. $q_0 = \mathrm{start}$
  2. there exists $a_1 a_2 \ldots a_n \in \mathcal{A}^n$ and $q_{i+1} \in \delta(q_i, a_{i+1})$
- A word $w = a_1 a_2 \ldots a_n$ is **accepted** if there is **at least a run** of length n **spelling its letters** and **ending** in an **accepting state**.
- The **set of words accepted** by the automaton is the **language recognized** by the automaton.
  (A language is a possibly infinite set of words)

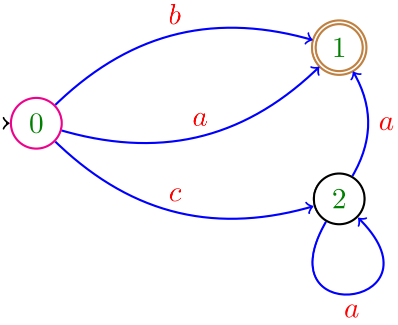
What is an automaton?



$\mathrm{AUTO} = (\mathcal{A}, Q, \mathrm{start}, \delta, F)$

- **Some not accepted words:**
  $c, a^m, ab, b^n \qquad (m \geq 2, n \geq 2)$
- **Accepted words:**
  $a, b, ca^n \qquad (n \geq 1)$

What is an automaton?



$$\text{AUTO} = (\mathcal{A}, Q, \text{start}, \delta, F)$$

- **Some not accepted words:**
  $c, a^m, ab, b^n \qquad (m \geq 2, n \geq 2)$
- **Accepted words:**
  $a, b, ca^n \qquad (n \geq 1)$
- **Recognized language**
  $a + b + ca^+ \qquad (a^+ = \sum_{n \geq 1} a^n)$

What are Automata and Motif Statistics useful for?

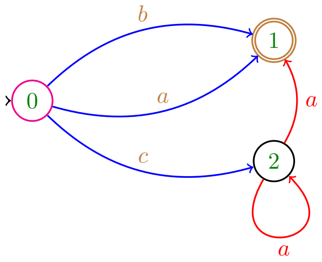
Automata are used

- in **hardware technology** (circuits)
- in **compilers** and **lexical analyzers**
- for **pattern matching**
- to build **groups** with **specific cogrowth**

Motif Statistics is used in

- **linguistics**
- **bioinformatics**
- **Web analysis**

What is an automaton? Deterministic or Non-Deterministic

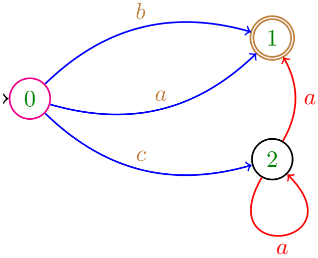


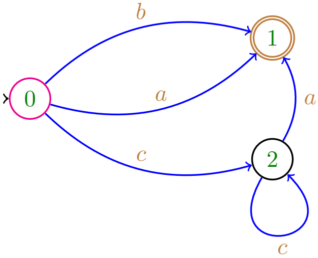
A **NFA**
(Non-deterministic Finite Automaton)

$$|\delta(2,a)| = \big|\{2,1\}\big| > 1$$

Several successors
with the **same letter**

What is an automaton? Deterministic or Non-Deterministic





A **NFA**
(Non-deterministic Finite Automaton)

$|\delta(2, a)| = \big|\{2, 1\}\big| > 1$

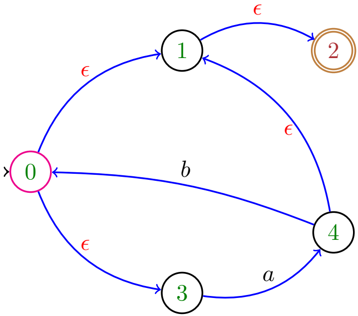
Several successors
with the **same letter**

A **DFA**
(Deterministic Finite Automaton)

$\forall q \in Q, \forall \ell \in \mathcal{A}, |\delta(q, \ell)| = 1$

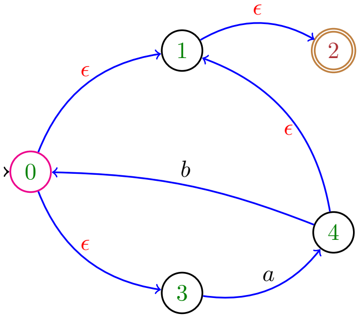
Only one successor
with **one letter** at **each state**

Finite Automata and ϵ−transitions



ϵ-auto $= \big(\mathcal{A} = \{a, b, \epsilon\}, Q = \{0, 1, 2, 3, 4\}, s = 0, \delta, F = \{2\}\big)$

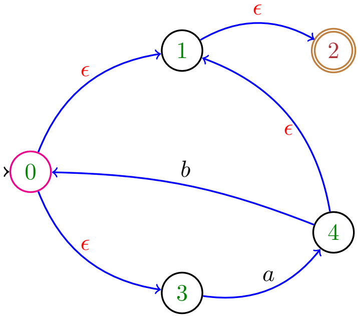
Finite Automata and ϵ−transitions



ϵ-auto $= \big(\mathcal{A} = \{a, b, \epsilon\}, Q = \{0, 1, 2, 3, 4\}, s = 0, \delta, F = \{2\}\big)$

An ϵ-transition consumes **no input**

Finite Automata and ϵ−transitions

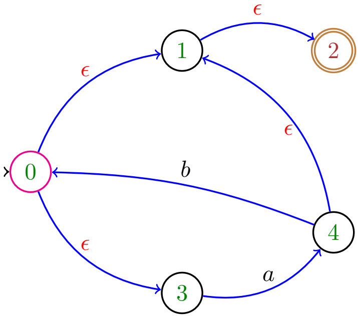


ϵ-auto $= \big(\mathcal{A} = \{a, b, \epsilon\}, Q = \{0, 1, 2, 3, 4\}, s = 0, \delta, F = \{2\}\big)$

An ϵ-transition consumes **no input**

ϵ-closure: $\forall q \in Q,\ \epsilon\text{-}cl(q) := \{p \mid p$ is accessible from q without consuming input$\}$

Finite Automata and $\epsilon-$transitions



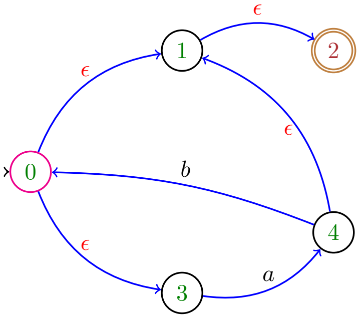
$\epsilon\text{-}cl(4) = \{4, 1, 2\}$

ϵ-auto $= \big(\mathcal{A} = \{a, b, \epsilon\}, Q = \{0, 1, 2, 3, 4\}, s = 0, \delta, F = \{2\}\big)$

An ϵ-transition consumes **no input**

ϵ-closure: $\forall q \in Q,\ \epsilon\text{-}cl(q) := \{p \mid p$ is accessible from q without consuming input$\}$

Finite Automata and ϵ−transitions



$\epsilon\text{-}cl(4) = \{4, 1, 2\}$

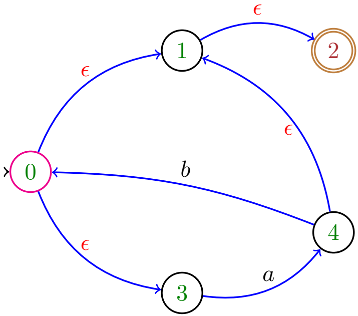
ϵ-auto $= \big(\mathcal{A} = \{a, b, \epsilon\}, Q = \{0, 1, 2, 3, 4\}, s = 0, \delta, F = \{2\}\big)$

An ϵ-transition consumes **no input**

ϵ-closure: $\quad \forall q \in Q,\ \epsilon\text{-}cl(q) := \{p \mid p$ is accessible from q without consuming input$\}$

auto-without-ϵ $= \big(\mathcal{A} = \{a, b\}, Q, s, \Delta, F'\big)$

Finite Automata and ϵ−transitions



$\epsilon\text{-}cl(4) = \{4, 1, 2\}$

$F' = \{0, 1, 4, 2\}$

ϵ-auto $= \big(\mathcal{A} = \{a, b, \epsilon\}, Q = \{0, 1, 2, 3, 4\}, s = 0, \delta, F = \{2\}\big)$

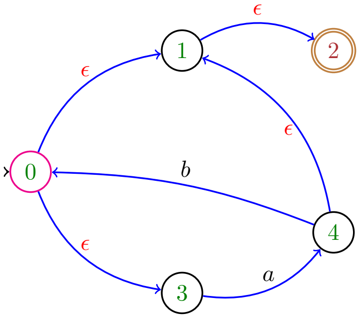
An ϵ-transition consumes **no input**

ϵ-closure: $\forall q \in Q,\ \epsilon\text{-}cl(q) := \{p \mid p$ is accessible from q without consuming input$\}$

auto-without-ϵ $= \big(\mathcal{A} = \{a, b\}, Q, s, \Delta, F'\big)$

$F' = F \bigcup \{q \mid \epsilon\text{-}cl(q) \bigcap F \neq \emptyset\} = \{0, 4, 1, 2\}$

Finite Automata and ϵ−transitions



$\epsilon\text{-}cl(4) = \{4, 1, 2\}$

$F' = \{0, 1, 4, 2\}$

ϵ-auto $= (\mathcal{A} = \{a, b, \epsilon\}, Q = \{0, 1, 2, 3, 4\}, s = 0, \delta, F = \{2\})$

An ϵ-transition consumes **no input**

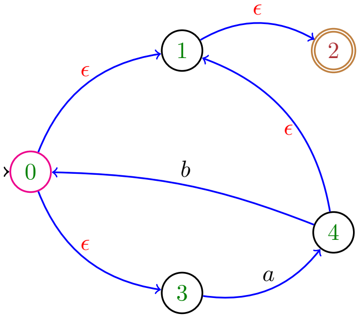
ϵ-closure: $\forall q \in Q,\ \epsilon\text{-}cl(q) := \{p \mid p$ is accessible from q without consuming input$\}$

auto-without-ϵ $= (\mathcal{A} = \{a, b\}, Q, s, \Delta, F')$

$F' = F \bigcup \{q \mid \epsilon\text{-}cl(q) \bigcap F \neq \emptyset\} = \{0, 4, 1, 2\}$

$\Delta(q, \ell) = \epsilon\text{-}cl(\bigcup_{p \in \epsilon\text{-}cl(q)} \delta(p, \ell))$

Finite Automata and ϵ−transitions



$\epsilon\text{-}cl(4) = \{4, 1, 2\}$

$F' = \{0, 1, 4, 2\}$

$\epsilon\text{-}cl(0) = \{0, 1, 2, 3\}$

ϵ-auto $= \big(\mathcal{A} = \{a, b, \epsilon\}, Q = \{0, 1, 2, 3, 4\}, s = 0, \delta, F = \{2\}\big)$

An ϵ-transition consumes **no input**

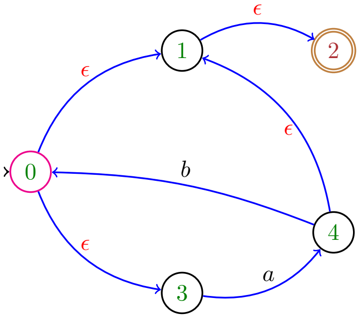
ϵ-closure: $\forall q \in Q,\ \epsilon\text{-}cl(q) := \{p \mid p$ is accessible from q without consuming input$\}$

auto-without-ϵ $= \big(\mathcal{A} = \{a, b\}, Q, s, \Delta, F'\big)$

$F' = F \bigcup \{q \mid \epsilon\text{-}cl(q) \bigcap F \neq \emptyset\} = \{0, 4, 1, 2\}$

$\Delta(q, \ell) = \epsilon\text{-}cl(\bigcup_{p \in \epsilon\text{-}cl(q)} \delta(p, \ell))$

Finite Automata and ϵ−transitions



$$\epsilon\text{-}cl(4) = \{4, 1, 2\}$$

$$F' = \{0, 1, 4, 2\}$$

$$\epsilon\text{-}cl(0) = \{0, 1, 2, 3\}$$

$$\Delta(0, a) = \epsilon\text{-}cl\left(\bigcup_{p \in \{0,1,2,3\}} \delta(p, a)\right)$$
$$= \epsilon\text{-}cl\left(\{4\}\right) = \{4, 1, 2\}$$

ϵ-auto $= \left(\mathcal{A} = \{a, b, \epsilon\}, Q = \{0, 1, 2, 3, 4\}, s = 0, \delta, F = \{2\}\right)$

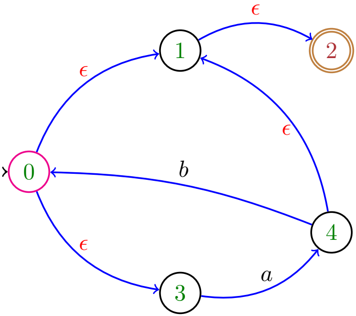
An ϵ-transition consumes **no input**
ϵ-closure: $\forall q \in Q,\ \epsilon\text{-}cl(q) := \{p \mid p$ is accessible from q without consuming input$\}$

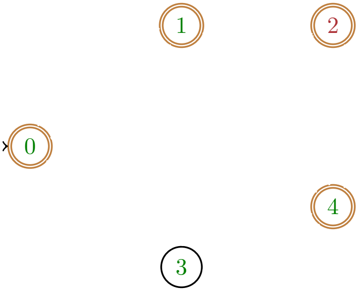
auto-without-ϵ $= \left(\mathcal{A} = \{a, b\}, Q, s, \Delta, F'\right)$

$F' = F \bigcup \{q \mid \epsilon\text{-}cl(q) \bigcap F \neq \emptyset\} = \{0, 4, 1, 2\}$

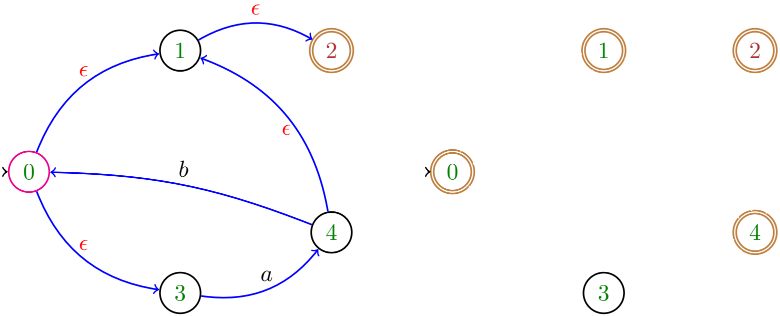
$\Delta(q, \ell) = \epsilon\text{-}cl(\bigcup_{p \in \epsilon\text{-}cl(q)} \delta(p, \ell))$

Finite Automata and ϵ−transitions





Finite Automata and $\epsilon-$transitions



$\epsilon\text{-}cl(0) = \{0, 1, 2, 3\}$

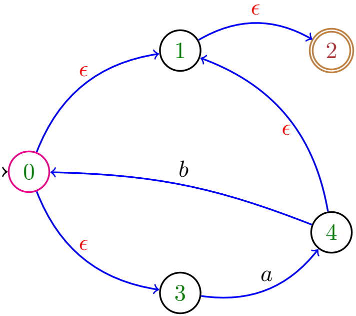
$\epsilon\text{-}cl(1) = \{1, 2\}$

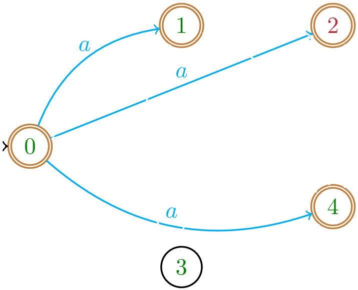
$\epsilon\text{-}cl(2) = \{2\}$

$\epsilon\text{-}cl(3) = \{3\}$

$\epsilon\text{-}cl(4) = \{4, 1, 2\}$

Finite Automata and ϵ−transitions





$\epsilon\text{-}cl(0) = \{0, 1, 2, 3\}$

$\epsilon\text{-}cl(1) = \{1, 2\}$

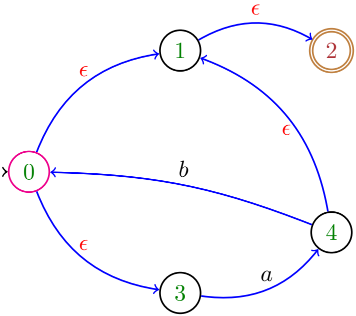
$\epsilon\text{-}cl(2) = \{2\}$

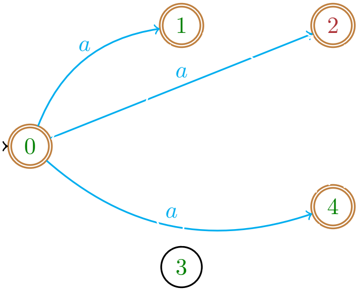
$\epsilon\text{-}cl(3) = \{3\}$

$\epsilon\text{-}cl(4) = \{4, 1, 2\}$

$\Delta(0, a) = \{4, 1, 2\}$

Finite Automata and ϵ–transitions





$$\epsilon\text{-}cl(0) = \{0, 1, 2, 3\}$$

$$\epsilon\text{-}cl(1) = \{1, 2\}$$

$$\epsilon\text{-}cl(2) = \{2\}$$

$$\epsilon\text{-}cl(3) = \{3\}$$

$$\epsilon\text{-}cl(4) = \{4, 1, 2\}$$

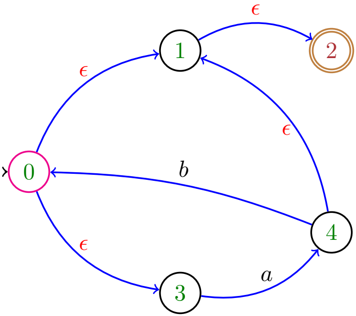
$$\Delta(0, a) = \{4, 1, 2\}$$

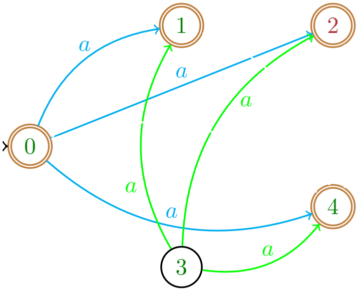
$$\Delta(0, b) = \{\}$$

$$\Delta(1, a) = \Delta(1, b) = \{\}$$

$$\Delta(2, a) = \Delta(2, b) = \{\}$$

Finite Automata and ϵ−transitions





$\epsilon\text{-}cl(0) = \{0, 1, 2, 3\}$

$\epsilon\text{-}cl(1) = \{1, 2\}$

$\epsilon\text{-}cl(2) = \{2\}$

$\epsilon\text{-}cl(3) = \{3\}$

$\epsilon\text{-}cl(4) = \{4, 1, 2\}$

$\Delta(0, a) = \{4, 1, 2\}$

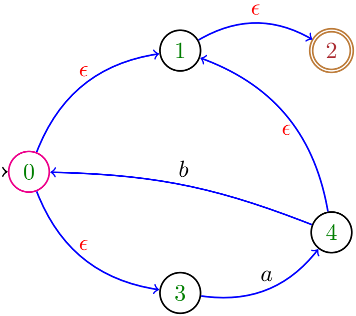
$\Delta(0, b) = \{\}$

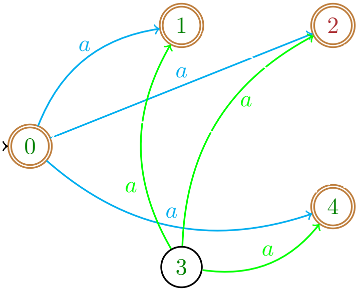
$\Delta(1, a) = \Delta(1, b) = \{\}$

$\Delta(2, a) = \Delta(2, b) = \{\}$

$\Delta(3, a) = \{4, 1, 2\}$

Finite Automata and ϵ−transitions





$\epsilon\text{-}cl(0) = \{0, 1, 2, 3\}$

$\epsilon\text{-}cl(1) = \{1, 2\}$

$\epsilon\text{-}cl(2) = \{2\}$

$\epsilon\text{-}cl(3) = \{3\}$

$\epsilon\text{-}cl(4) = \{4, 1, 2\}$

$\Delta(0, a) = \{4, 1, 2\}$

$\Delta(0, b) = \{\}$

$\Delta(1, a) = \Delta(1, b) = \{\}$

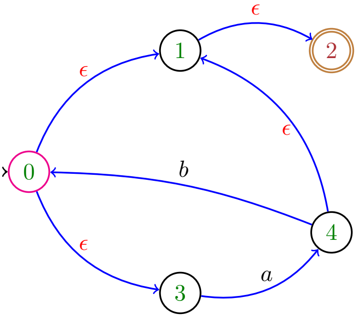
$\Delta(2, a) = \Delta(2, b) = \{\}$

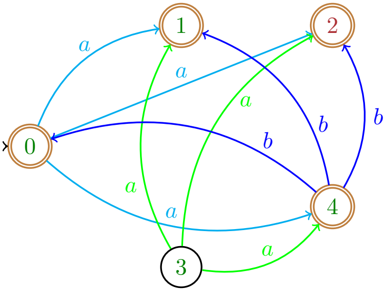
$\Delta(3, a) = \{4, 1, 2\}$

$\Delta(3, b) = \{\}$

$\Delta(4, a) = \{\}$

Finite Automata and ϵ−transitions





$\epsilon\text{-}cl(0) = \{0, 1, 2, 3\}$

$\epsilon\text{-}cl(1) = \{1, 2\}$

$\epsilon\text{-}cl(2) = \{2\}$

$\epsilon\text{-}cl(3) = \{3\}$

$\epsilon\text{-}cl(4) = \{4, 1, 2\}$

$\Delta(0, a) = \{4, 1, 2\}$

$\Delta(0, b) = \{\}$

$\Delta(1, a) = \Delta(1, b) = \{\}$

$\Delta(2, a) = \Delta(2, b) = \{\}$

$\Delta(3, a) = \{4, 1, 2\}$

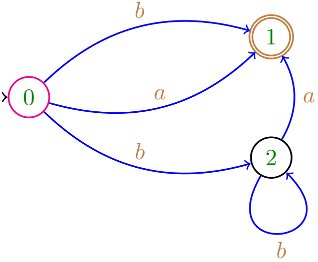
$\Delta(3, b) = \{\}$

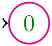
$\Delta(4, a) = \{\}$

$\Delta(4, b) = \{0, 1, 2\}$

Determinisation of an automaton

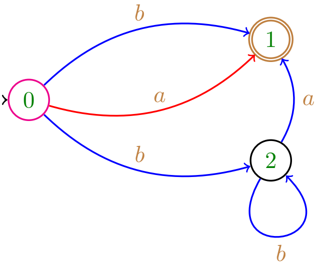
$M_{\text{NFA}} = (\mathcal{A}, Q, 0, \delta, F)$ $\qquad$ $M'_{\text{DFA}} = (\mathcal{A}, Q', 0, \Delta, F')$

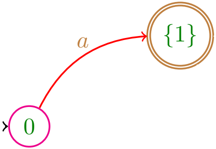




Determinisation of an automaton

$M_{\text{NFA}} = (\mathcal{A}, Q, 0, \delta, F)$ $M'_{\text{DFA}} = (\mathcal{A}, Q', 0, \Delta, F')$

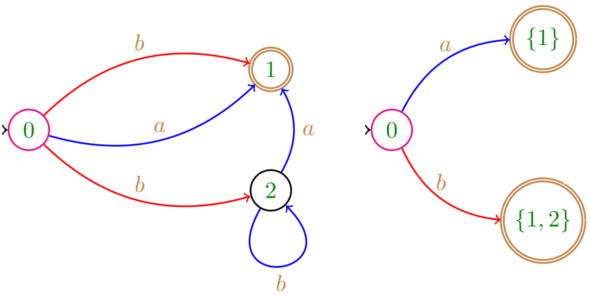




$$\Delta(0, a) = \{1\}$$

Determinisation of an automaton

$M_{\text{NFA}} = (\mathcal{A}, Q, 0, \delta, F)$ $\qquad$ $M'_{\text{DFA}} = (\mathcal{A}, Q', 0, \Delta, F')$

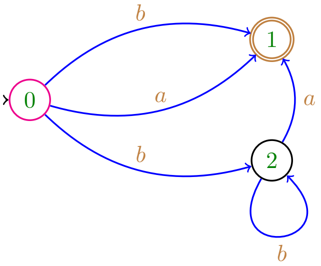


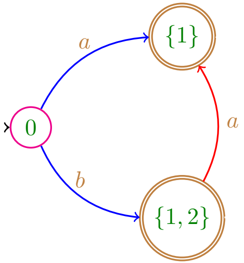
$$\Delta(0, a) = \{1\}$$

$$\Delta(0, b) = \{1, 2\}$$

Determinisation of an automaton

$M_{\mathrm{NFA}} = (\mathcal{A}, Q, 0, \delta, F)$ $\qquad$ $M'_{\mathrm{DFA}} = (\mathcal{A}, Q', 0, \Delta, F')$





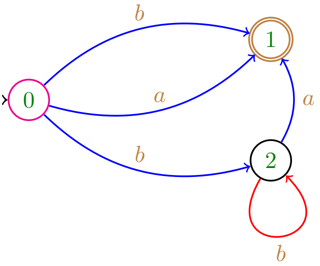
$$\Delta(0, a) = \{1\}$$

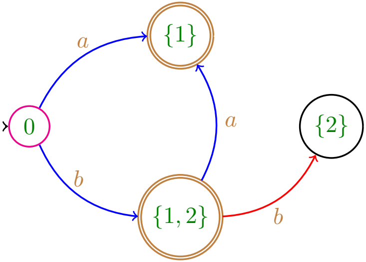
$$\Delta(0, b) = \{1, 2\}$$

$$\Delta(\{1, 2\}, a) = \{1\}$$

Determinisation of an automaton

$M_{\mathrm{NFA}} = (\mathcal{A}, Q, 0, \delta, F)$ $\qquad$ $M'_{\mathrm{DFA}} = (\mathcal{A}, Q', 0, \Delta, F')$





$$\Delta(0, a) = \{1\}$$

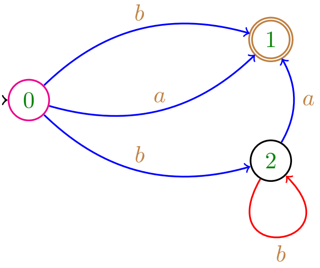
$$\Delta(0, b) = \{1, 2\}$$

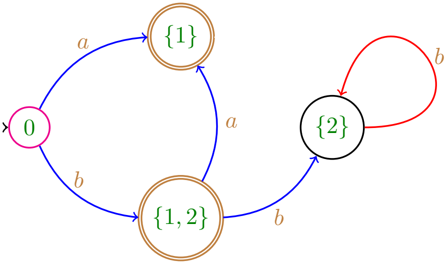
$$\Delta(\{1, 2\}, a) = \{1\}$$

$$\Delta(\{1, 2\}, b) = \{2\}$$

Determinisation of an automaton

$M_{\text{NFA}} = (\mathcal{A}, Q, 0, \delta, F)$ $\qquad$ $M'_{\text{DFA}} = (\mathcal{A}, Q', 0, \Delta, F')$





$\Delta(0, a) = \{1\}$

$\Delta(0, b) = \{1, 2\}$

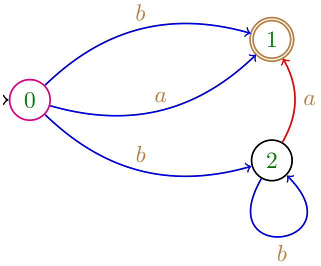
$\Delta(\{1, 2\}, a) = \{1\}$

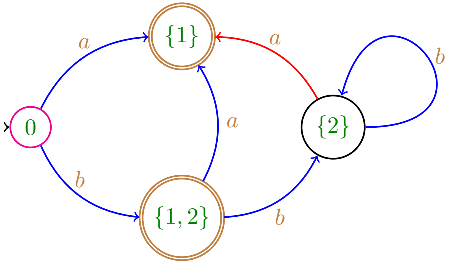
$\Delta(\{1, 2\}, b) = \{2\}$

$\Delta(\{2\}, b) = \{2\}$

Determinisation of an automaton

$M_{\mathrm{NFA}} = (\mathcal{A}, Q, 0, \delta, F)$ $\qquad$ $M'_{\mathrm{DFA}} = (\mathcal{A}, Q', 0, \Delta, F')$





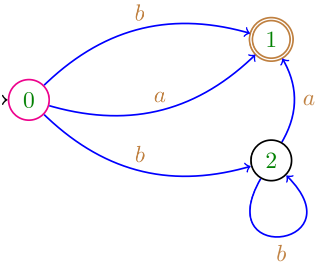
$\Delta(0, a) = \{1\}$ $\qquad$ $\Delta(\{1, 2\}, b) = \{2\}$

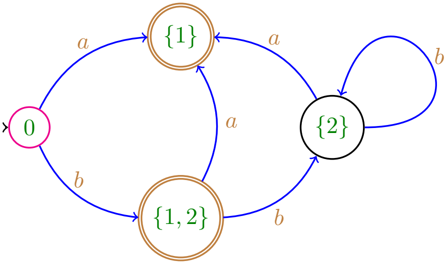
$\Delta(0, b) = \{1, 2\}$ $\qquad$ $\Delta(\{2\}, b) = \{2\}$

$\Delta(\{1, 2\}, a) = \{1\}$ $\qquad$ $\Delta(\{2\}, a) = \{1\}$

Determinisation of an automaton

$M_{\text{NFA}} = (\mathcal{A}, Q, 0, \delta, F)$ $\quad$ $M'_{\text{DFA}} = (\mathcal{A}, Q', 0, \Delta, F')$





$\Delta(0, a) = \{1\}$

$\Delta(0, b) = \{1, 2\}$

$\Delta(\{1, 2\}, a) = \{1\}$

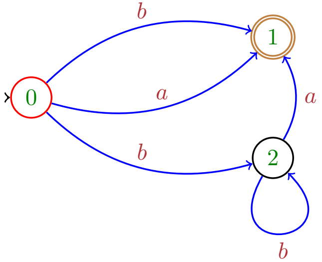
$\Delta(\{1, 2\}, b) = \{2\}$

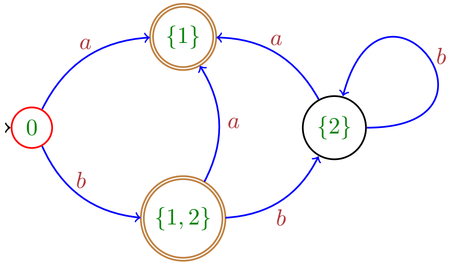
$\Delta(\{2\}, b) = \{2\}$

$\Delta(\{2\}, a) = \{1\}$

Determinisation of an automaton

$M_{\text{NFA}} = (\mathcal{A}, Q, s, \delta, F)$ $\quad$ $M'_{\text{DFA}} = (\mathcal{A}, Q', s', \Delta, F')$

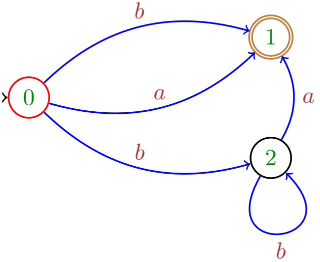


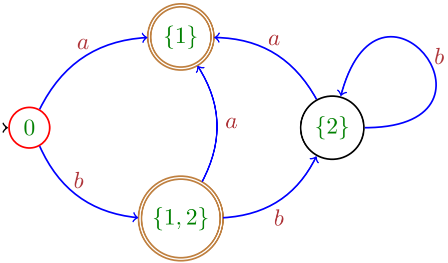


$Q' \subset 2^Q$ (the **subsets** of Q)

Determinisation of an automaton

$M_{\mathrm{NFA}} = (\mathcal{A}, Q, s, \delta, F)$ $\qquad$ $M'_{\mathrm{DFA}} = (\mathcal{A}, Q', s', \Delta, F')$



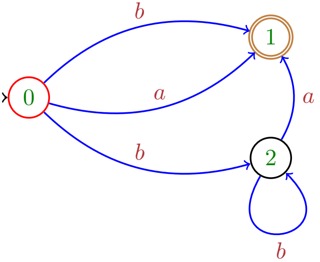


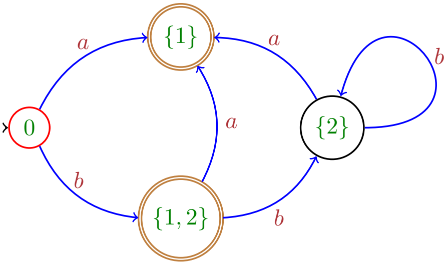
$Q' \subset 2^{Q}$ (the **subsets** of Q)

$s' = s$

Determinisation of an automaton

$M_{\mathrm{NFA}} = (\mathcal{A}, Q, s, \delta, F)$ $\qquad$ $M'_{\mathrm{DFA}} = (\mathcal{A}, Q', s', \Delta, F')$





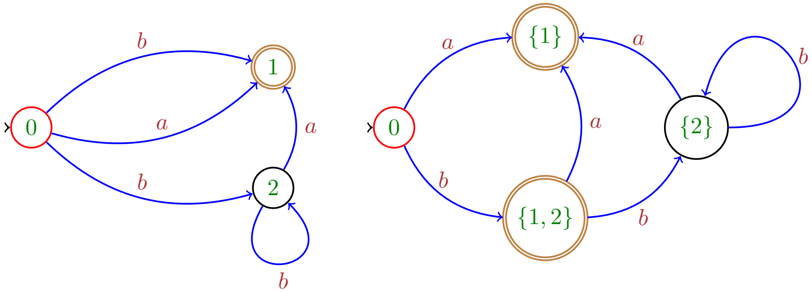
$Q' \subset 2^Q$ (the **subsets** of Q)

$s' = s$

$F' = \{f \in Q'; \quad f \cap F \neq \emptyset\}$ $\left\{\begin{array}{l}\text{the subsets that contain} \\ \text{at least one accepting state of } M\end{array}\right.$

Determinisation of an automaton

$M_{\text{NFA}} = (\mathcal{A}, Q, s, \delta, F)$ $\qquad$ $M'_{\text{DFA}} = (\mathcal{A}, Q', s', \Delta, F')$



$Q' \subset 2^Q$ (the **subsets** of Q)

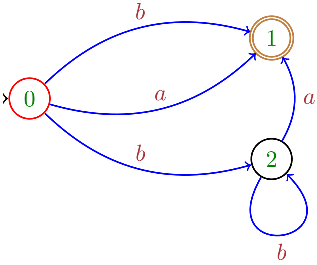
$s' = s$

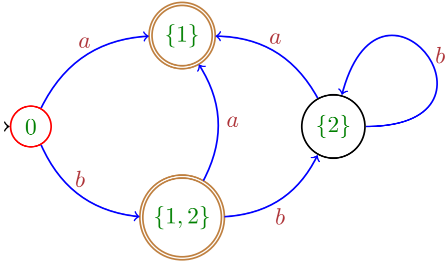
$F' = \{f \in Q'; \quad f \cap F \neq \emptyset\}$ $\Big\{$ the **subsets** that **contain** **at least one accepting state of** M

$$\forall S \in Q', \forall \ell \in \mathcal{A}, \quad \Delta(S, \ell) = \bigcup_{q \in S} \delta(q, \ell)$$

The automata M_{NFA} and M_{DFA} are equivalent

$M_{\mathrm{NFA}} = (\mathcal{A}, Q, s, \delta, F)$ $\quad$ $M'_{\mathrm{DFA}} = (\mathcal{A}, Q', s', \Delta, F')$

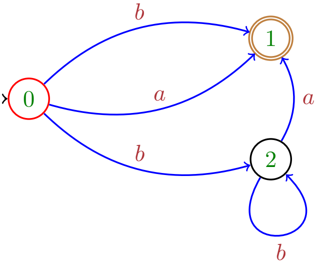


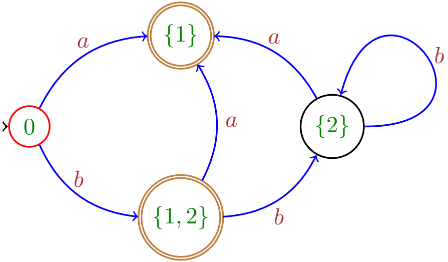


each **accepted run** of M_{NFA} **translates** to **an accepted run** of M_{NFA}

The automata M_{NFA} and M_{DFA} are equivalent

$M_{\text{NFA}} = (\mathcal{A}, Q, s, \delta, F)$ $\qquad$ $M'_{\text{DFA}} = (\mathcal{A}, Q', s', \Delta, F')$





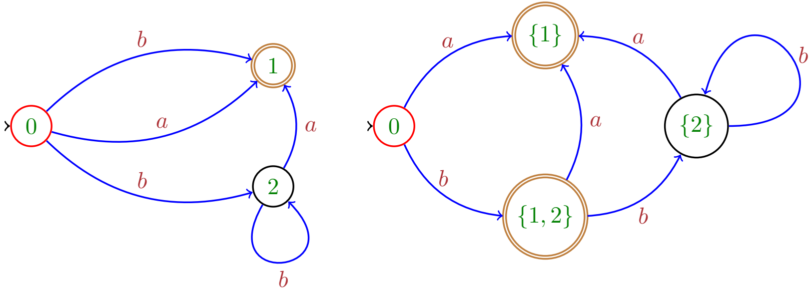
each **accepted run** of M_{NFA} **translates** to **an accepted run** of M_{NFA}

each **non accepted run** of M_{DFA} is the **translation**

of a **non accepted run** of M_{DFA}

The automata M_{NFA} and M_{DFA} are equivalent

$M_{\mathrm{NFA}} = (\mathcal{A}, Q, s, \delta, F)$ $\qquad$ $M'_{\mathrm{DFA}} = (\mathcal{A}, Q', s', \Delta, F')$



each **accepted run** of M_{NFA} **translates** to **an accepted run** of M_{NFA}

each **non accepted run** of M_{DFA} is the **translation**

of a **non accepted run** of M_{DFA}

Proof by induction

Equivalence of Non-Deterministic and Deterministic automata

Two automata $M = (Q, \mathcal{A}, s, \delta, F)$ and $M' = (Q', \mathcal{A}', s', \delta', F')$ are **equivalent** if they recognize the **same language** $(\mathcal{L}(M) = \mathcal{L}(M'))$

Theorem (Rabin-Scott 1959)

Let $M = (Q, \mathcal{A}, s, \Delta, F)$ be a NFA. Then there exists a DFA $M' = (Q', \mathcal{A}', s', \delta', F')$ that is equivalent to M.

Remark: each **DFA** is a **NFA**

Corollary

(i) The NFA's are no more powerful than the DFAs in terms of the languages they accept.
(ii) The NFA's and DFA's recognize the same set of languages.

Another characterization of the languages recognized by Finite Automata (NFA and DFA)?

Another characterization of the languages recognized by Finite Automata (NFA and DFA)?

YES!!!

Another characterization of the languages recognized by Finite Automata (NFA and DFA)?

YES!!!

The Regular Languages and Regular Expressions

What is a Regular Language?

Definition

Let $\mathcal{A}$ be a **finite alphabet**.

The collection of **regular languages** over $\mathcal{A}$ is defined **recursively** by

What is a Regular Language?

Definition

Let $\mathcal{A}$ be a **finite alphabet**.
The collection of **regular languages** over $\mathcal{A}$ is defined **recursively** by

1. $\emptyset$ is a regular language

What is a Regular Language?

Definition

Let $\mathcal{A}$ be a **finite alphabet**.
The collection of **regular languages** over $\mathcal{A}$ is defined **recursively** by

1. $\emptyset$ is a regular language
2. $\{\epsilon\}$ is a regular language

What is a Regular Language?

Definition

Let $\mathcal{A}$ be a **finite alphabet**.
The collection of **regular languages** over $\mathcal{A}$ is defined **recursively** by

1. $\emptyset$ is a regular language
2. $\{\epsilon\}$ is a regular language
3. $\{\ell\}$ is a regular language for each $\ell \in \mathcal{A}$

What is a Regular Language?

Definition

Let $\mathcal{A}$ be a **finite alphabet**.

The collection of **regular languages** over $\mathcal{A}$ is defined **recursively** by

1. $\emptyset$ is a regular language
2. $\{\epsilon\}$ is a regular language
3. $\{\ell\}$ is a regular language for each $\ell \in \mathcal{A}$
4. if A and B are regular languages, so are

What is a Regular Language?

Definition

Let $\mathcal{A}$ be a **finite alphabet**.

The collection of **regular languages** over $\mathcal{A}$ is defined **recursively** by

1. $\emptyset$ is a regular language
2. $\{\epsilon\}$ is a regular language
3. $\{\ell\}$ is a regular language for each $\ell \in \mathcal{A}$
4. if A and B are regular languages, so are
    - $A \bigcup B$ (Ex: $\{ab\} \bigcup \{c\} = \{ab, c\}$)

What is a Regular Language?

Definition

Let $\mathcal{A}$ be a **finite alphabet**.
The collection of regular languages over $\mathcal{A}$ is defined recursively by

1. $\emptyset$ is a regular language
2. $\{\epsilon\}$ is a regular language
3. $\{\ell\}$ is a regular language for each $\ell \in \mathcal{A}$
4. if A and B are regular languages, so are
   - $A \bigcup B$ (Ex: $\{ab\} \bigcup \{c\} = \{ab, c\}$)
   - $A \bullet B$ (Ex: $\{ab, c\} \bullet \{d, e\} = \{abd, cd, abe, ce\}$)

What is a Regular Language?

Definition

Let $\mathcal{A}$ be a **finite alphabet**.
The collection of regular languages over $\mathcal{A}$ is defined **recursively** by

1. $\emptyset$ is a regular language
2. $\{\epsilon\}$ is a regular language
3. $\{\ell\}$ is a regular language for each $\ell \in \mathcal{A}$
4. if A and B are regular languages, so are
   - $A \bigcup B$ (Ex: $\{ab\} \bigcup \{c\} = \{ab, c\}$)
   - $A \bullet B$ (Ex: $\{ab, c\} \bullet \{d, e\} = \{abd, cd, abe, ce\}$)
   - $A^{\star}$ (Ex: $\{ab\}^{\star} = \{\epsilon, ab, abab, \ldots, (ab)^n, \ldots\}$)

What is a Regular Language?

Definition

Let $\mathcal{A}$ be a **finite alphabet**.

The collection of regular languages over $\mathcal{A}$ is defined recursively by

1. $\emptyset$ is a regular language
2. $\{\epsilon\}$ is a regular language
3. $\{\ell\}$ is a regular language for each $\ell \in \mathcal{A}$
4. if A and B are regular languages, so are
   - $A \bigcup B$ (Ex: $\{ab\} \bigcup \{c\} = \{ab, c\}$)
   - $A \bullet B$ (Ex: $\{ab, c\} \bullet \{d, e\} = \{abd, cd, abe, ce\}$)
   - $A^\star$ (Ex: $\{ab\}^\star = \{\epsilon, ab, abab, \ldots, (ab)^n, \ldots\}$)
5. No other languages over $\mathcal{A}$ are regular

Regular Expressions

Regular expressions are **shorthands** for **regular languages**

$a + b$	denotes	$\{a, b\} = \{a\} \bigcup \{b\}$
ab	denotes	$\{ab\} = \{a \bullet b\}$
$a^\star$	denotes	$\{a\}^\star$
a^+	denotes	$a.a^\star = a \bullet a^\star$

Formal definition of Regular Expressions

Regular expressions are defined recursively by

1. $\emptyset$ and ϵ are regular expressions
2. ℓ is a regular expressions for each $\ell \in \mathcal{A}$
3. if r and s are regular expressions, so are
   - $r + s$
   - $r.s$
   - $r^\star$
4. No other sequence of symbols is a regular expression.

Kleene Theorem

Lemma (i)

Every regular language can be accepted by a finite automaton

Lemma (ii)

Every language accepted by a finite automaton is regular

Theorem (Kleene 1956)

A language is regular if and only if it is accepted by a Finite Automaton

Lemma(i) - From Regular Expressions to Finite Automata

1. **Atomic Languages**

$$\emptyset \quad \text{is accepted by} \quad (\mathcal{A}, \{0\}, 0, \delta = \emptyset, \emptyset)$$

$$\epsilon \quad \text{is accepted by} \quad (\mathcal{A}, \{0\}, 0, \delta = \emptyset, \{0\})$$

$$\ell \in \mathcal{A} \quad \text{is accepted by} \quad (\mathcal{A}, \{0, 1\}, 0, \delta(0, \ell) = \{1\}, \{1\})$$

2. let $\mathcal{L}_1$ and $\mathcal{L}_2$ **regular languages** respectively **accepted** by automata A_1 and A_2.

$$\mathcal{L}_1 . \mathcal{L}_2 \quad \text{is accepted by} \quad A_1 . A_2$$

$$\mathcal{L}_1 + \mathcal{L}_2 \quad \text{is accepted by} \quad A_1 \cup A_2$$

$$\mathcal{L}_1{}^\star \quad \text{is accepted by} \quad A_1{}^\star$$

Lemma(i) - From Regular Expressions to Finite Automata

1. **Atomic Languages**

$\emptyset$	is accepted by	$(\mathcal{A}, \{0\}, 0, \delta = \emptyset, \emptyset)$
ϵ	is accepted by	$(\mathcal{A}, \{0\}, 0, \delta = \emptyset, \{0\})$
$\ell \in \mathcal{A}$	is accepted by	$(\mathcal{A}, \{0, 1\}, 0, \delta(0, \ell) = \{1\}, \{1\})$

2. let $\mathcal{L}_1$ and $\mathcal{L}_2$ **regular languages** respectively **accepted** by automata A_1 and A_2.

$\mathcal{L}_1.\mathcal{L}_2$	is accepted by	$A_1.A_2$
$\mathcal{L}_1+\mathcal{L}_2$	is accepted by	$A_1 \cup A_2$
$\mathcal{L}_1{}^{\star}$	is accepted by	$A_1{}^{\star}$

Starting from the atomic languages, one **builds recursively** a **ϵ-NFA recognizing a given regular expression**

Lemma(ii) - From Finite Automata to Regular Expressions

$A = (\mathcal{A} + \epsilon, \{q_1, q_2, \ldots, q_m\}, S \subseteq Q, \delta, F \subseteq Q)$ a finite automaton

1. let $L(i, j, k) = \left\{ w \;\middle|\; \begin{array}{l} w \text{ is the label of a path from } q_i \text{ to } q_j \\ \text{where intermediate nodes have labels } \leq k \end{array} \right\}$

Lemma(ii) - From Finite Automata to Regular Expressions

$A = (\mathcal{A} + \epsilon, \{q_1, q_2, \ldots, q_m\}, S \subseteq Q, \delta, F \subseteq Q)$ a finite automaton

1. let $L(i, j, k) = \left\{ w \;\middle|\; \begin{array}{l} w \text{ is the label of a path from } q_i \text{ to } q_j \\ \text{where intermediate nodes have labels } \leq k \end{array} \right\}$
2. $L(i, j, 0)$ has no intermediate labels $\Longrightarrow L(i, j, 0) \subseteq \mathcal{A} \cup \epsilon$ is regular

Lemma(ii) - From Finite Automata to Regular Expressions

$A = (\mathcal{A} + \epsilon, \{q_1, q_2, \ldots, q_m\}, S \subseteq Q, \delta, F \subseteq Q)$ a finite automaton

1. let $L(i,j,k) = \left\{ w \;\middle|\; \begin{array}{l} w \text{ is the label of a path from } q_i \text{ to } q_j \\ \text{where intermediate nodes have labels} \leq k \end{array} \right\}$
2. $L(i,j,0)$ has no intermediate labels $\Longrightarrow L(i,j,0) \subseteq \mathcal{A} \cup \epsilon$ is regular
3. Assume $L(i,j,k)$ regular and consider $L(i,j,k+1)$
   Let p be a path form q_i to q_j where **intermediate nodes** have **labels** $\leq k+1$.
   - (a) $p \in L(i,j,k)$ (the path p does not reach q_{k+1})
   - (b) p begins at q_i,reaches q_{k+1} a **first time**, possibly **other times**, until a **last time**, and ends at q_j

   Cases (a) and (b) give
   $L(i,j,k+1) = L(i,j,k) \cup L(i,k+1,k)L(k+1,k+1,k)^{\star}L(k+1,j,k)$
   Therefore $L(i,j,k+1)$ is **regular**

Lemma(ii) - From Finite Automata to Regular Expressions

$A = (\mathcal{A} + \epsilon, \{q_1, q_2, \ldots, q_m\}, S \subseteq Q, \delta, F \subseteq Q)$ a finite automaton

1. let $L(i, j, k) = \left\{ w \;\middle|\; \begin{array}{l} w \text{ is the label of a path from } q_i \text{ to } q_j \\ \text{where intermediate nodes have labels } \leq k \end{array} \right\}$
2. $L(i, j, 0)$ has no intermediate labels $\Longrightarrow L(i, j, 0) \subseteq \mathcal{A} \cup \epsilon$ is regular
3. Assume $L(i, j, k)$ regular and consider $L(i, j, k+1)$
   Let p be a path form q_i to q_j where **intermediate nodes** have **labels** $\leq k+1$.
   - (a) $p \in L(i, j, k)$ (the path p does not reach q_{k+1})
   - (b) p begins at q_i,reaches q_{k+1} a **first time**, possibly **other times**, until a **last time**, and ends at q_j

   Cases (a) and (b) give
   $L(i, j, k+1) = L(i, j, k) \cup L(i, k+1, k) L(k+1, k+1, k)^{\star} L(k+1, j, k)$
   Therefore $L(i, j, k+1)$ is **regular**
4. In particular $L(i, j, m)$ is regular

Lemma(ii) - From Finite Automata to Regular Expressions

$A = (\mathcal{A} + \epsilon, \{q_1, q_2, \ldots, q_m\}, S \subseteq Q, \delta, F \subseteq Q)$ a finite automaton

1. let $L(i,j,k) = \left\{ w \;\middle|\; \begin{array}{l} w \text{ is the label of a path from } q_i \text{ to } q_j \\ \text{where intermediate nodes have labels } \leq k \end{array} \right\}$
2. $L(i,j,0)$ has no intermediate labels $\Longrightarrow L(i,j,0) \subseteq \mathcal{A} \cup \epsilon$ is regular
3. Assume $L(i,j,k)$ regular and consider $L(i,j,k+1)$
   Let p be a path form q_i to q_j where **intermediate nodes** have **labels** $\leq k+1$.
   - (a) $p \in L(i,j,k)$ (the path p does not reach q_{k+1})
   - (b) p begins at q_i,reaches q_{k+1} a **first time**, possibly **other times**, until a **last time**, and ends at q_j

   Cases (a) and (b) give
   $L(i,j,k+1) = L(i,j,k) \cup L(i,k+1,k)L(k+1,k+1,k)^{\star}L(k+1,j,k)$
   Therefore $L(i,j,k+1)$ is **regular**
4. In particular $L(i,j,m)$ is regular

Conclusion: $L(A) = \bigcup \{L(i,j,m) | q_i \in S, q_j \in F\}$ is **regular**, since it is a **finite union of regular languages**

Counting - Generating Function of a Language

$\mathcal{L}$ a language (a possibly infinite set of words)

- **Enumeration**

$$L(z) = \sum_{w \in \mathcal{L}} z^{|w|} = \sum_{n \geq 0} l_n z^n$$

where l_n is the **number of words** of **length** n of $\mathcal{L}$

- **Weighted generating Function**

$$W(z) = \sum_{w \in \mathcal{L}} \mathbf{P}(w) z^{|w|} = \sum_{n \geq 0} p_n z^n$$

where p_n is the **probability** that a **random word of length** n belongs to $\mathcal{L}$

Counting - Generating Function of a Language

- Enumeration

$$L(a,b) = \sum_{w \in \mathcal{L}} a^{|w|_a} b^{|w|_b} = \sum_{i,j} l_{i,j} a^i b^j$$

$l_{i,j}$ = **number of words in the language with** $\begin{cases} i \text{ **letters** } a \\ j \text{ **letters** } b \end{cases}$

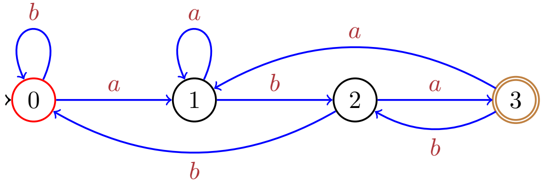
$$F(z) = L(z,z) = \sum_n f_n z^n, \quad f_n =$$ **number of words of length** n **in the language**

- Weighted counting $F(z) = L(\mathbf{P}(a)z, \mathbf{P}(b)z) = \sum_n p_n z^n$

$p_n =$ **probability that a word of length** n **is in the language**

Generating Function of a Regular Expression
Chomsky-Schützenberger (1963)

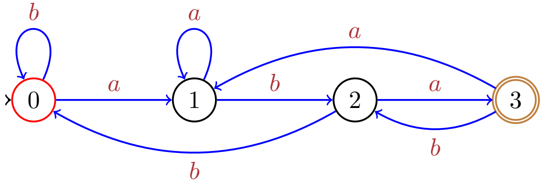
$$P = \mathcal{A}^\star aba = (a+b)^\star aba$$



Generating Function of a Regular Expression Chomsky-Schützenberger (1963)

$$P = \mathcal{A}^\star aba = (a+b)^\star aba$$

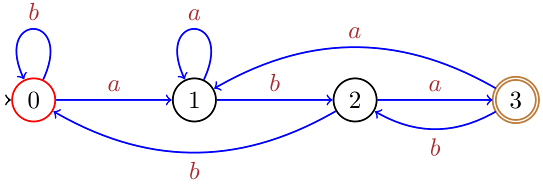
The automaton **accepts** the words **terminating** with aba



Generating Function of a Regular Expression
Chomsky-Schützenberger (1963)

$$P = \mathcal{A}^\star aba = (a+b)^\star aba$$

The automaton **accepts** the words **terminating** with aba

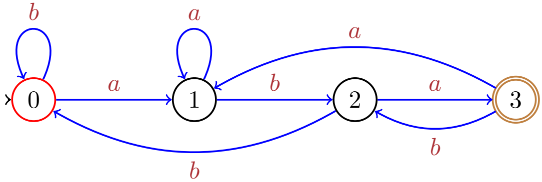


$\mathcal{L}_i$ **language of runs** $\begin{cases} \text{that } \textbf{start at state } i \\ \text{and } \textbf{terminate in an accepting state} \end{cases}$

Generating Function of a Regular Expression
Chomsky-Schützenberger (1963)

$$P = \mathcal{A}^\star aba = (a+b)^\star aba$$

The automaton **accepts** the words **terminating** with aba



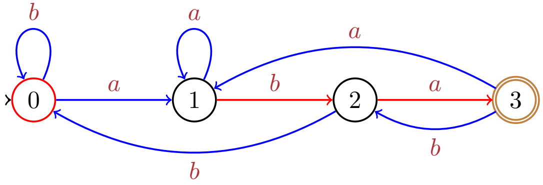
$\mathcal{L}_i$ **language of runs** $\begin{cases} \text{that start at state } i \\ \text{and terminate in an accepting state} \end{cases}$

$$\mathcal{L}_1 =$$

Generating Function of a Regular Expression
Chomsky-Schützenberger (1963)

$$P = \mathcal{A}^\star aba = (a+b)^\star aba$$

The automaton **accepts** the words **terminating** with aba



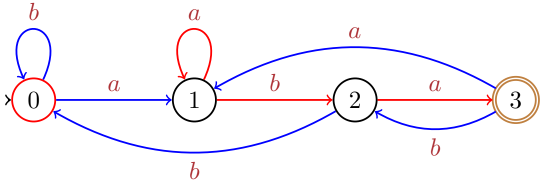
$\mathcal{L}_i$ **language of runs** $\left\{\begin{array}{l}\text{that } \textbf{start at state } i \\ \text{and } \textbf{terminate in an accepting state}\end{array}\right.$

$$\mathcal{L}_1 = ba$$

Generating Function of a Regular Expression
Chomsky-Schützenberger (1963)

$$P = \mathcal{A}^\star aba = (a+b)^\star aba$$

The automaton **accepts** the words **terminating** with aba



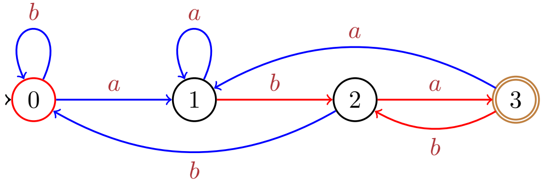
$\mathcal{L}_i$ **language of runs** $\left\{\begin{array}{l}\text{that start at state } i\\ \text{and terminate in an accepting state}\end{array}\right.$

$$\mathcal{L}_1 = ba + a^\star ba$$

Generating Function of a Regular Expression
Chomsky-Schützenberger (1963)

$$P = \mathcal{A}^\star aba = (a+b)^\star aba$$

The automaton **accepts** the words **terminating** with aba



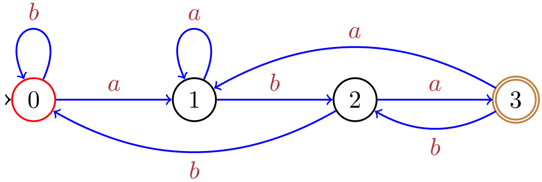
$\mathcal{L}_i$ **language of runs** $\left\{ \begin{array}{l} \text{that } \textbf{start at state } i \\ \text{and } \textbf{terminate in an accepting state} \end{array} \right.$

$$\mathcal{L}_1 = ba + a^\star ba + ba(ba)^\star$$

Generating Function of a Regular Expression Chomsky-Schützenberger (1963)

$$P = \mathcal{A}^\star aba = (a+b)^\star aba$$

The automaton **accepts** the words **terminating** with aba



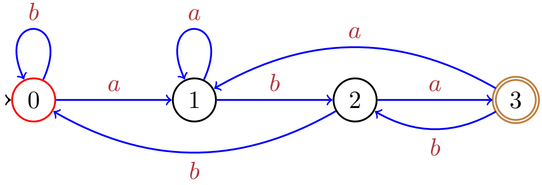
$\mathcal{L}_i$ **language of runs** $\left\{ \begin{array}{l} \text{that } \textbf{start at state } i \\ \text{and } \textbf{terminate in an accepting state} \end{array} \right.$

$$\mathcal{L}_1 = ba + a^\star ba + ba(ba)^\star + \ldots$$

Generating Function of a Regular Expression Chomsky-Schützenberger (1963)

$$P = \mathcal{A}^\star aba = (a+b)^\star aba$$

The automaton **accepts** the words **terminating** with aba



$\mathcal{L}_i$ **language of runs** $\begin{cases} \text{that start at state } i \\ \text{and terminate in an accepting state} \end{cases}$

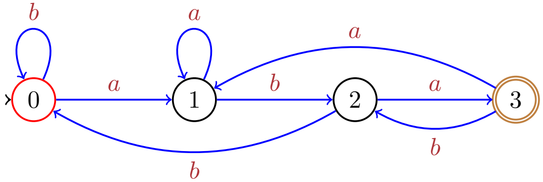
$$\mathcal{L}_1 = ba + a^\star ba + ba(ba)^\star + \ldots$$

$$\mathcal{L}_0 = a.\mathcal{L}_1 + b.\mathcal{L}_0$$

Generating Function of a Regular Expression
Chomsky-Schützenberger (1963)

$$P = \mathcal{A}^\star aba = (a+b)^\star aba$$

The automaton **accepts** the words **terminating** with aba



$\mathcal{L}_i$ **language of runs** $\left\{ \begin{array}{l} \text{that } \textbf{start at state } i \\ \text{and } \textbf{terminate in an accepting state} \end{array} \right.$

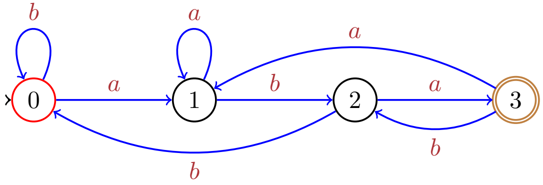
$$\mathcal{L}_1 = ba + a^\star ba + ba(ba)^\star + \ldots$$

$$\mathcal{L}_0 = a.\mathcal{L}_1 + b.\mathcal{L}_0 \qquad\qquad L_0(a,b) = a \times L_1(a,b) + b \times L_0(a,b)$$

Generating Function of a Regular Expression
Chomsky-Schützenberger (1963)

$$P = \mathcal{A}^\star aba = (a+b)^\star aba$$

The automaton **accepts** the words **terminating** with aba



$\mathcal{L}_i$ **language of runs** that **start at state** i and **terminate in an accepting state**

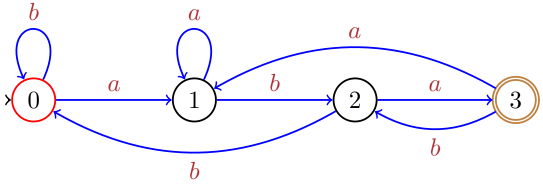
$$\mathcal{L}_1 = ba + a^\star ba + ba(ba)^\star + \dots$$

$$\mathcal{L}_0 = a.\mathcal{L}_1 + b.\mathcal{L}_0 \qquad L_0(a,b) = a \times L_1(a,b) + b \times L_0(a,b)$$
$$\mathcal{L}_1 = a.\mathcal{L}_1 + b.\mathcal{L}_2 \qquad L_1(a,b) = a \times L_1(a,b) + b \times L_2(a,b)$$

Generating Function of a Regular Expression Chomsky-Schützenberger (1963)

$$P = \mathcal{A}^\star aba = (a+b)^\star aba$$

The automaton **accepts** the words **terminating** with aba



$\mathcal{L}_i$ **language of runs** $\left\{\begin{array}{l}\text{that start at state } i \\ \text{and terminate in an accepting state}\end{array}\right.$

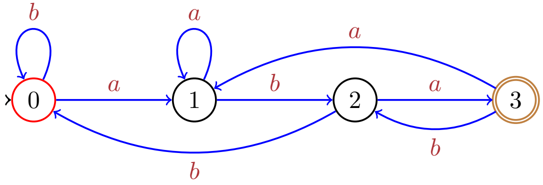
$$\mathcal{L}_1 = ba + a^\star ba + ba(ba)^\star + \ldots$$

$$\begin{aligned}\mathcal{L}_0 &= a.\mathcal{L}_1 + b.\mathcal{L}_0 \\ \mathcal{L}_1 &= a.\mathcal{L}_1 + b.\mathcal{L}_2 \\ \mathcal{L}_2 &= a.\mathcal{L}_3 + b.\mathcal{L}_0\end{aligned} \qquad \begin{aligned}L_0(a,b) &= a \times L_1(a,b) + b \times L_0(a,b) \\ L_1(a,b) &= a \times L_1(a,b) + b \times L_2(a,b) \\ L_2(a,b) &= a \times L_3(a,b) + b \times L_0(a,b)\end{aligned}$$

Generating Function of a Regular Expression
Chomsky-Schützenberger (1963)

$$P = \mathcal{A}^\star aba = (a+b)^\star aba$$

The automaton **accepts** the words **terminating** with aba



$\mathcal{L}_i$ **language of runs** $\left\{\begin{array}{l}\text{that start at state } i\\ \text{and terminate in an accepting state}\end{array}\right.$

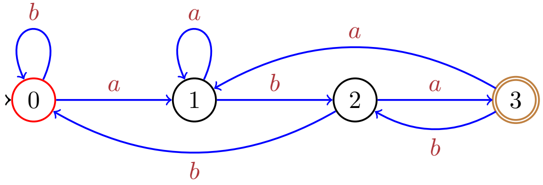
$$\mathcal{L}_1 = ba + a^\star ba + ba(ba)^\star + \dots$$

$$\begin{aligned}
\mathcal{L}_0 &= a.\mathcal{L}_1 + b.\mathcal{L}_0 & L_0(a,b) &= a \times L_1(a,b) + b \times L_0(a,b)\\
\mathcal{L}_1 &= a.\mathcal{L}_1 + b.\mathcal{L}_2 & L_1(a,b) &= a \times L_1(a,b) + b \times L_2(a,b)\\
\mathcal{L}_2 &= a.\mathcal{L}_3 + b.\mathcal{L}_0 & L_2(a,b) &= a \times L_3(a,b) + b \times L_0(a,b)\\
\mathcal{L}_3 &= a.\mathcal{L}_1 + b.\mathcal{L}_2 + \epsilon & &
\end{aligned}$$

Generating Function of a Regular Expression
Chomsky-Schützenberger (1963)

$$P = \mathcal{A}^\star aba = (a+b)^\star aba$$

The automaton **accepts** the words **terminating** with aba



$\mathcal{L}_i$ **language of runs** that **start at state** i and **terminate in an accepting state**

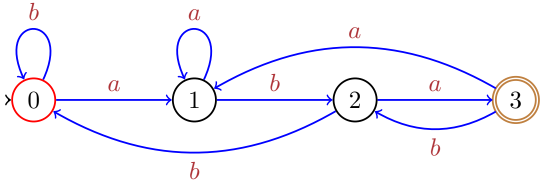
$$\mathcal{L}_1 = ba + a^\star ba + ba(ba)^\star + \dots$$

$$\begin{aligned}
\mathcal{L}_0 &= a.\mathcal{L}_1 + b.\mathcal{L}_0 & L_0(a,b) &= a \times L_1(a,b) + b \times L_0(a,b)\\
\mathcal{L}_1 &= a.\mathcal{L}_1 + b.\mathcal{L}_2 & L_1(a,b) &= a \times L_1(a,b) + b \times L_2(a,b)\\
\mathcal{L}_2 &= a.\mathcal{L}_3 + b.\mathcal{L}_0 & L_2(a,b) &= a \times L_3(a,b) + b \times L_0(a,b)\\
\mathcal{L}_3 &= a.\mathcal{L}_1 + b.\mathcal{L}_2 + \epsilon & L_3(a,b) &= a \times L_1(a,b) + b \times L_2(a,b) + 1
\end{aligned}$$

Generating Function of a Regular Expression Chomsky-Schützenberger (1963)

$$P = \mathcal{A}^\star aba = (a+b)^\star aba$$

The automaton **accepts** the words **terminating** with aba



$\mathcal{L}_i$ **language of runs** $\begin{cases} \text{that start at state } i \\ \text{and terminate in an accepting state} \end{cases}$

$$\mathcal{L}_1 = ba + a^\star ba + ba(ba)^\star + \dots$$

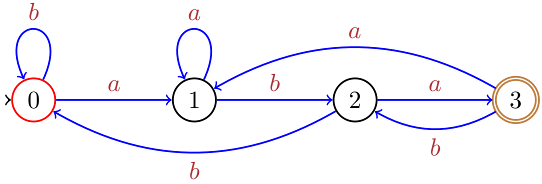
$$\begin{aligned} \mathcal{L}_0 &= a.\mathcal{L}_1 + b.\mathcal{L}_0 & L_0(a,b) &= a \times L_1(a,b) + b \times L_0(a,b) \\ \mathcal{L}_1 &= a.\mathcal{L}_1 + b.\mathcal{L}_2 & L_1(a,b) &= a \times L_1(a,b) + b \times L_2(a,b) \\ \mathcal{L}_2 &= a.\mathcal{L}_3 + b.\mathcal{L}_0 & L_2(a,b) &= a \times L_3(a,b) + b \times L_0(a,b) \\ \mathcal{L}_3 &= a.\mathcal{L}_1 + b.\mathcal{L}_2 + \epsilon & L_3(a,b) &= a \times L_1(a,b) + b \times L_2(a,b) + 1 \end{aligned}$$

solve:

Generating Function of a Regular Expression
Chomsky-Schützenberger (1963)

$$P = \mathcal{A}^\star aba = (a+b)^\star aba$$

The automaton **accepts** the words **terminating** with aba



$\mathcal{L}_i$ **language of runs** $\begin{cases} \text{that start at state } i \\ \text{and terminate in an accepting state} \end{cases}$

$$\mathcal{L}_1 = ba + a^\star ba + ba(ba)^\star + \dots$$

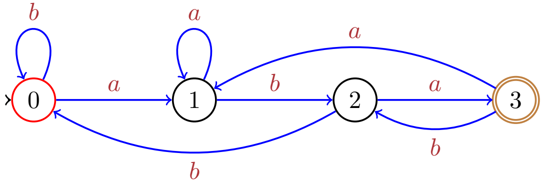
$$\begin{aligned}
\mathcal{L}_0 &= a.\mathcal{L}_1 + b.\mathcal{L}_0 \qquad & L_0(a,b) &= a \times L_1(a,b) + b \times L_0(a,b) \\
\mathcal{L}_1 &= a.\mathcal{L}_1 + b.\mathcal{L}_2 \qquad & L_1(a,b) &= a \times L_1(a,b) + b \times L_2(a,b) \\
\mathcal{L}_2 &= a.\mathcal{L}_3 + b.\mathcal{L}_0 \qquad & L_2(a,b) &= a \times L_3(a,b) + b \times L_0(a,b) \\
\mathcal{L}_3 &= a.\mathcal{L}_1 + b.\mathcal{L}_2 + \epsilon \qquad & L_3(a,b) &= a \times L_1(a,b) + b \times L_2(a,b) + 1
\end{aligned}$$

solve: $$L_0(a,b) = \frac{1}{1-(a+b)} \times aba$$

Generating Function of a Regular Expression Chomsky-Schützenberger (1963)

$$P = \mathcal{A}^\star aba = (a+b)^\star aba$$

The automaton **accepts** the words **terminating** with aba



$\mathcal{L}_i$ **language of runs** $\left\{ \begin{array}{l} \text{that start at state } i \\ \text{and terminate in an accepting state} \end{array} \right.$

$$\mathcal{L}_1 = ba + a^\star ba + ba(ba)^\star + \dots$$

$$\begin{aligned}
\mathcal{L}_0 &= a.\mathcal{L}_1 + b.\mathcal{L}_0 & L_0(a,b) &= a \times L_1(a,b) + b \times L_0(a,b) \\
\mathcal{L}_1 &= a.\mathcal{L}_1 + b.\mathcal{L}_2 & L_1(a,b) &= a \times L_1(a,b) + b \times L_2(a,b) \\
\mathcal{L}_2 &= a.\mathcal{L}_3 + b.\mathcal{L}_0 & L_2(a,b) &= a \times L_3(a,b) + b \times L_0(a,b) \\
\mathcal{L}_3 &= a.\mathcal{L}_1 + b.\mathcal{L}_2 + \epsilon & L_3(a,b) &= a \times L_1(a,b) + b \times L_2(a,b) + 1
\end{aligned}$$

solve: $L_0(a,b) = \dfrac{1}{1-(a+b)} \times aba$ $\quad F(z) = \sum p_n z^n = L_0(\mathbf{P}(a)z, \mathbf{P}(b)z)$

Asymptotics of a rational expression

- if $F(z) = \dfrac{P(z)}{Q(z)}$ with $P(\rho \neq 0),\ Q(\rho = 0)$
- and ρ **real**, **positive**, **dominant singularity** of **order** k

Then,

$$f_n = [z^n]F(z) = \frac{P(\rho)}{Q(\rho)} \times \rho^{-n} \times (n - k + 1) \times (1 + A^n) \qquad (A < 1)$$

Expand the **polynomial** $P(z)$ at ρ

$$P(z) = P(\rho) + (z - \rho)P'(\rho) + \frac{1}{2!}(z - \rho)^2 P''(\rho) + \ldots$$

to get a **full expansion**

Generating Functions of Regular Languages

1. Any regular expression is recognized by a Finite Automaton
2. The Chomsky-Schützenberger algorithm **applies** to **any regular expression**.

Generating Functions of Regular Languages

1. Any regular expression is recognized by a Finite Automaton
2. The Chomsky-Schützenberger algorithm **applies** to **any regular expression**.

Theorem (Chomsky-Schützenberger 1963)

The generating function of a regular language is rational.

Corollary

Let $\mathcal{R}$ a regular language and $\mathcal{R}_n = \mathcal{R} \cap \mathcal{A}^n$.
$\exists n_0, \ \forall n > n_0, \quad |\mathcal{R}_n| = p_1(n)\lambda_1^n + \cdots + p_k(n)\lambda_k^n$, *with* $p_i(n)$ *complex polynomials and* $\lambda_i \in \mathbb{C}$

An asymptotic test of non-regularity

For any regular language $\mathcal{R}$, there exists a real positive number λ and a polynomial $p(n)$ such that

$$\lim_{n\to\infty} r_n = \lambda^n \times p(n), \qquad r_n = \left|\mathcal{R} \bigcap \mathcal{A}^n\right|$$

- The number of words of length $2n$ in **Dyck Languages** $((()(())))$ is the Catalan number $\binom{2n}{n}/(2n+1)$ asymptotic to $\frac{4^n}{n^{3/2}\sqrt{\pi}}$.

  Dyck languages are **not regular** and **cannot be recognized by a DFA**; however they can be recognized by a push-down automaton, and they have an algebraic generating function.

- Let $\pi(x)$ be the **number of prime numbers less** than $x \in \mathbb{R}^+$.

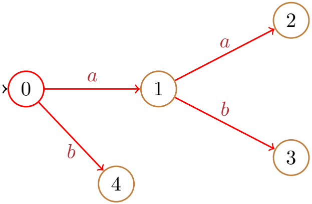
$$\lim_{x\to\infty} \frac{\pi(x)}{x/\log x} = 1$$

  There is **no known generating function enumerating the primes**. Would one find one it would **not be regular**. It is **not possible to enumerate** the **primes** by an **automaton**.

Some classical pattern matching algorithms

Aho-Corasick (1975) - Finite Motif - Multiple Counting

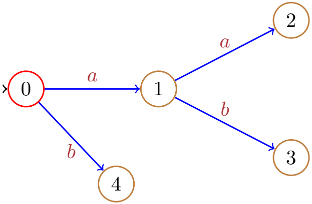
$$P = \{a, aa, ab, b\}$$



- build a **trie** over the words of P

Aho-Corasick (1975) - Finite Motif - Multiple Counting

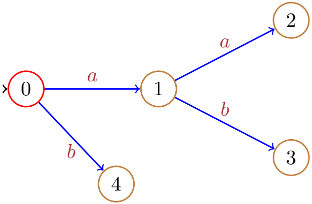
$$P = \{a, aa, ab, b\}$$



- build a **trie** over the words of P
  let Q be the set of nodes of the trie: $Q = \{0, 1, 2, 3, 4\}$
  $\forall q \in Q$, let w_q the word spelling the run from 0 to q – $(w_3 = ab)$

Aho-Corasick (1975) - Finite Motif - Multiple Counting

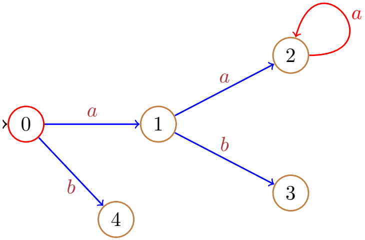
$$P = \{a, aa, ab, b\}$$



- build a **trie** over the words of P
  let Q be the set of nodes of the trie: $Q = \{0, 1, 2, 3, 4\}$
  $\forall q \in Q$, let w_q the word spelling the run from 0 to q – $(w_3 = ab)$
- for each **node** q with a **missing transition** ℓ
  add a transition $\delta(q, \ell)$ to state q'
  such that $w_{q'}$ is **the longest possible suffix** of $w_q.\ell$

Aho-Corasick (1975) - Finite Motif - Multiple Counting

$$P = \{a, aa, ab, b\}$$

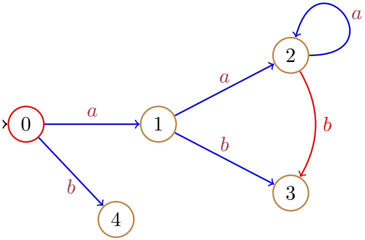


$\delta(2, a) = 2 \quad w_2.a = a.aa$

- build a **trie** over the words of P
  let Q be the set of nodes of the trie: $Q = \{0, 1, 2, 3, 4\}$
  $\forall q \in Q$, let w_q the word spelling the run from 0 to q – ($w_3 = ab$)
- for each **node** q with a **missing transition** ℓ
  add a transition $\delta(q, \ell)$ to state q'
  such that $w_{q'}$ is **the longuest possible suffix** of $w_q.\ell$

Aho-Corasick (1975) - Finite Motif - Multiple Counting

$$P = \{a, aa, ab, b\}$$



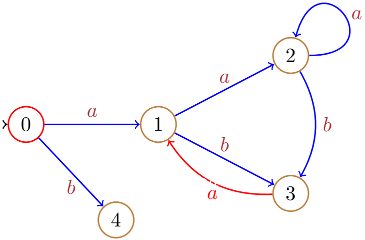
$\delta(2, a) = 2 \quad w_2.a = a.aa$

$\delta(2, b) = 3 \quad w_2.b = a.ab$

- build a **trie** over the words of P
  let Q be the set of nodes of the trie: $Q = \{0, 1, 2, 3, 4\}$
  $\forall q \in Q$, let w_q the word spelling the run from 0 to q – $(w_3 = ab)$
- for each **node** q with a **missing transition** ℓ
  add a transition $\delta(q, \ell)$ to state q'
  such that $w_{q'}$ is **the longest possible suffix** of $w_q.\ell$

Aho-Corasick (1975) - Finite Motif - Multiple Counting

$$P = \{a, aa, ab, b\}$$



$\delta(2, a) = 2 \quad w_2.a = a.aa$

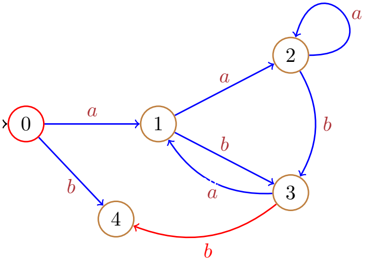
$\delta(2, b) = 3 \quad w_2.b = a.ab$

$\delta(3, a) = 1 \quad w_3.a = ab.a$

- build a **trie** over the words of P
  let Q be the set of nodes of the trie: $Q = \{0, 1, 2, 3, 4\}$
  $\forall q \in Q$, let w_q the word spelling the run from 0 to q – $(w_3 = ab)$
- for each **node** q with a **missing transition** ℓ
  add a transition $\delta(q, \ell)$ to state q'
  such that $w_{q'}$ is **the longest possible suffix** of $w_q.\ell$

Aho-Corasick (1975) - Finite Motif - Multiple Counting

$$P = \{a, aa, ab, b\}$$



$\delta(2, a) = 2 \quad w_2.a = a.aa$

$\delta(2, b) = 3 \quad w_2.b = a.ab$

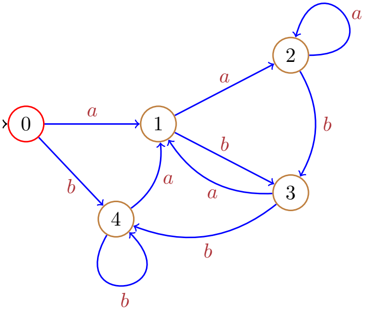
$\delta(3, a) = 1 \quad w_3.a = ab.a$

$\delta(3, b) = 4 \quad w_3.b = ab.b$

- build a **trie** over the words of P
  let Q be the set of nodes of the trie: $Q = \{0, 1, 2, 3, 4\}$
  $\forall q \in Q$, let w_q the word spelling the run from 0 to q – $(w_3 = ab)$
- for each **node** q with a **missing transition** ℓ
  add a transition $\delta(q, \ell)$ to state q'
  such that $w_{q'}$ is **the longest possible suffix** of $w_q.\ell$

Aho-Corasick (1975) - Finite Motif - Multiple Counting

$$P = \{a, aa, ab, b\}$$



$\delta(2, a) = 2 \quad w_2.a = a.aa$

$\delta(2, b) = 3 \quad w_2.b = a.ab$

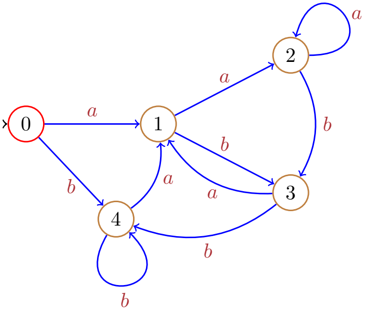
$\delta(3, a) = 1 \quad w_3.a = ab.a$

$\delta(3, b) = 4 \quad w_3.b = ab.b$

- build a **trie** over the words of P
  let Q be the set of nodes of the trie: $Q = \{0, 1, 2, 3, 4\}$
  $\forall q \in Q$, let w_q the word spelling the run from 0 to q – $(w_3 = ab)$
- for each **node** q with a **missing transition** ℓ
  add a transition $\delta(q, \ell)$ to state q'
  such that $w_{q'}$ is **the longest possible suffix** of $w_q.\ell$

Aho-Corasick (1975) - Finite Motif - Multiple Counting

$$P = \{a, aa, ab, b\}$$



$\delta(2, a) = 2 \quad w_2.a = a.aa$

$\delta(2, b) = 3 \quad w_2.b = a.ab$

$\delta(3, a) = 1 \quad w_3.a = ab.a$

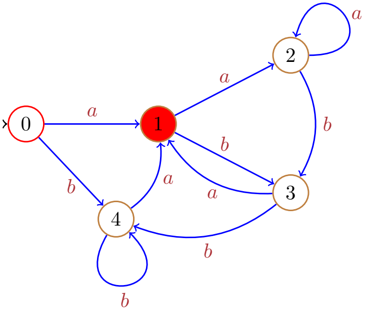
$\delta(3, b) = 4 \quad w_3.b = ab.b$

for **each specific match** ring a **bell**

- build a **trie** over the words of P
  let Q be the set of nodes of the trie: $Q = \{0, 1, 2, 3, 4\}$
  $\forall q \in Q$, let w_q the word spelling the run from 0 to q – $(w_3 = ab)$
- for each **node** q with a **missing transition** ℓ
  add a transition $\delta(q, \ell)$ to state q'
  such that $w_{q'}$ is **the longest possible suffix** of $w_q.\ell$

Aho-Corasick (1975) - Finite Motif - Multiple Counting

$P = \{a, aa, ab, b\}$



$\delta(2, a) = 2 \quad w_2.a = a.aa$

$\delta(2, b) = 3 \quad w_2.b = a.ab$

$\delta(3, a) = 1 \quad w_3.a = ab.a$

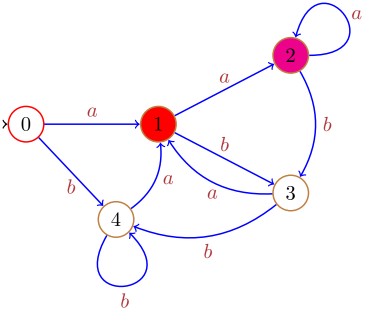
$\delta(3, b) = 4 \quad w_3.b = ab.b$

for each specific match ring a bell

- build a **trie** over the words of P
  let Q be the set of nodes of the trie: $Q = \{0, 1, 2, 3, 4\}$
  $\forall q \in Q$, let w_q the word spelling the run from 0 to q – $(w_3 = ab)$
- for each **node** q with a **missing transition** ℓ
  add a transition $\delta(q, \ell)$ to state q'
  such that $w_{q'}$ is **the longest possible suffix** of $w_q.\ell$

Aho-Corasick (1975) - Finite Motif - Multiple Counting

$$P = \{a, aa, ab, b\}$$



$\delta(2, a) = 2 \quad w_2.a = a.aa$

$\delta(2, b) = 3 \quad w_2.b = a.ab$

$\delta(3, a) = 1 \quad w_3.a = ab.a$

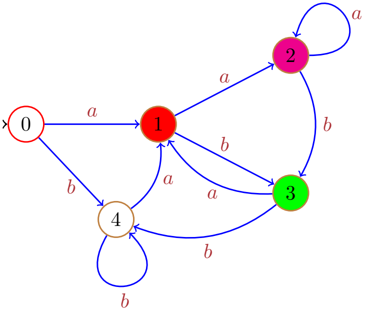
$\delta(3, b) = 4 \quad w_3.b = ab.b$

for **each specific match** ring a **bell**

- build a **trie** over the words of P
  let Q be the set of nodes of the trie: $Q = \{0, 1, 2, 3, 4\}$
  $\forall q \in Q$, let w_q the word spelling the run from 0 to q – $(w_3 = ab)$
- for each **node** q with a **missing transition** ℓ
  add a transition $\delta(q, \ell)$ to state q'
  such that $w_{q'}$ is **the longest possible suffix** of $w_q.\ell$

Aho-Corasick (1975) - Finite Motif - Multiple Counting

$$P = \{a, aa, ab, b\}$$



$\delta(2, a) = 2 \quad w_2.a = a.aa$

$\delta(2, b) = 3 \quad w_2.b = a.ab$

$\delta(3, a) = 1 \quad w_3.a = ab.a$

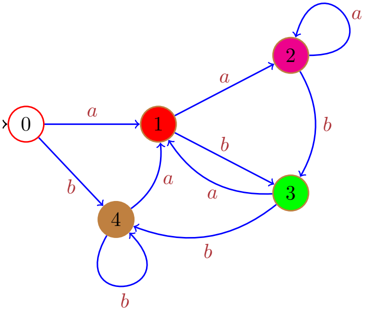
$\delta(3, b) = 4 \quad w_3.b = ab.b$

for **each specific match** ring a **bell**

- build a **trie** over the words of P
  let Q be the set of nodes of the trie: $Q = \{0, 1, 2, 3, 4\}$
  $\forall q \in Q$, let w_q the word spelling the run from 0 to q – ($w_3 = ab$)
- for each **node** q with a **missing transition** ℓ
  add a transition $\delta(q, \ell)$ to state q'
  such that $w_{q'}$ is **the longest possible suffix** of $w_q.\ell$

Aho-Corasick (1975) - Finite Motif - Multiple Counting

$P = \{a, aa, ab, b\}$



$\delta(2, a) = 2 \quad w_2.a = a.aa$

$\delta(2, b) = 3 \quad w_2.b = a.ab$

$\delta(3, a) = 1 \quad w_3.a = ab.a$

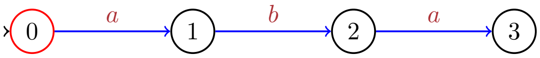
$\delta(3, b) = 4 \quad w_3.b = ab.b$

for each specific match ring a bell

- build a **trie** over the words of P
  let Q be the set of nodes of the trie: $Q = \{0, 1, 2, 3, 4\}$
  $\forall q \in Q$, let w_q the word spelling the run from 0 to q – $(w_3 = ab)$
- for each **node** q with a **missing transition** ℓ
  add a transition $\delta(q, \ell)$ to state q'
  such that $w_{q'}$ is the longest possible suffix of $w_q.\ell$

Knuth-Morris-Pratt automaton (1977) - Only one word

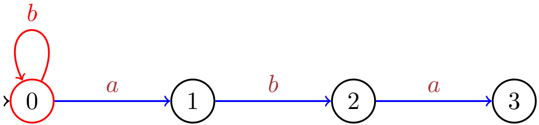
$$P = aba$$



same construction as Aho-Corasick

Knuth-Morris-Pratt automaton (1977) - Only one word

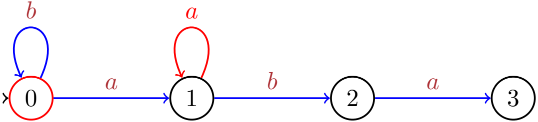
$$P = aba$$



same construction as Aho-Corasick

Knuth-Morris-Pratt automaton (1977) - Only one word

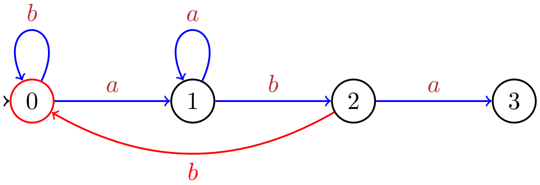
$$P = aba$$



same construction as Aho-Corasick

Knuth-Morris-Pratt automaton (1977) - Only one word

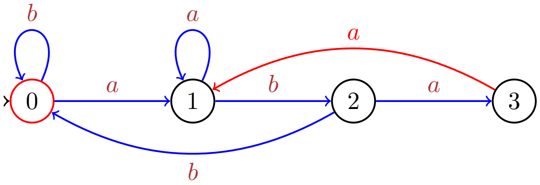
$$P = aba$$



same construction as Aho-Corasick

Knuth-Morris-Pratt automaton (1977) - Only one word

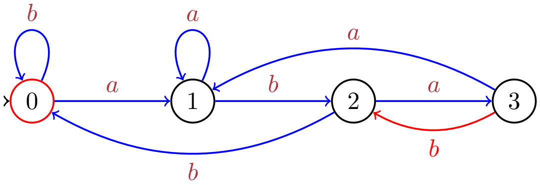
$$P = aba$$



same construction as Aho-Corasick

Knuth-Morris-Pratt automaton (1977) - Only one word

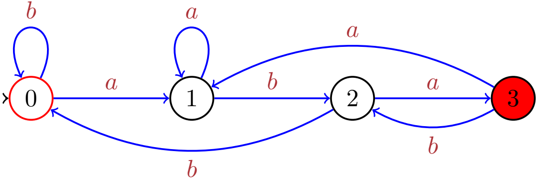
$P = aba$



same construction as Aho-Corasick

Knuth-Morris-Pratt automaton (1977) - Only one word

$P = aba$

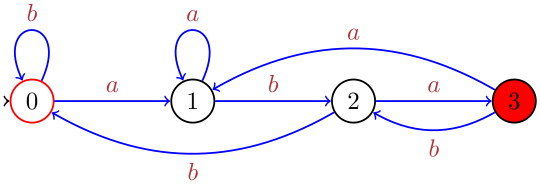


same construction as Aho-Corasick

for each match ring the bell

Knuth-Morris-Pratt automaton (1977) - Only one word

$P = aba$



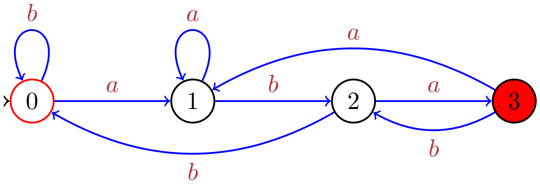
same construction as Aho-Corasick

for each match ring the **bell**

$aaaaaba$

Knuth-Morris-Pratt automaton (1977) - Only one word

$$P = aba$$



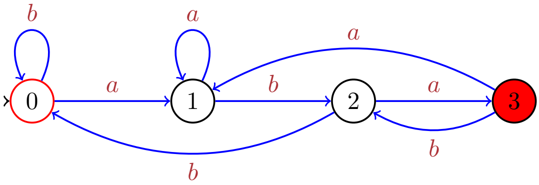
same construction as Aho-Corasick

for each match ring the **bell**

$aaaaaba$

Knuth-Morris-Pratt automaton (1977) - Only one word

$P = aba$



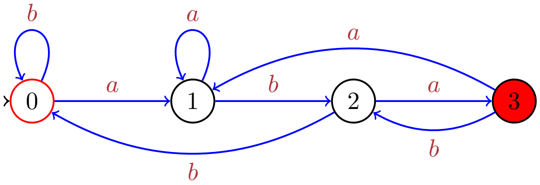
same construction as Aho-Corasick

for each match ring the bell

$aaaaaba$🔔$bbaba$

Knuth-Morris-Pratt automaton (1977) - Only one word

$P = aba$



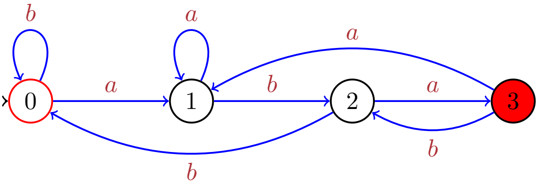
same construction as Aho-Corasick

for each match ring the **bell**

$aaaaaba$🔔$bbaba$🔔

Knuth-Morris-Pratt automaton (1977) - Only one word

$$P = aba$$



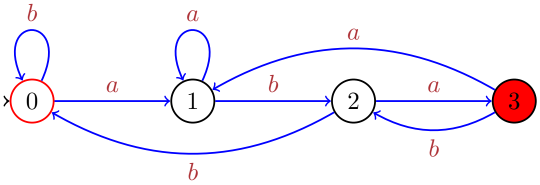
same construction as Aho-Corasick

for **each match** ring the **bell**

$aaaaababbababa$

Knuth-Morris-Pratt automaton (1977) - Only one word

$$P = aba$$



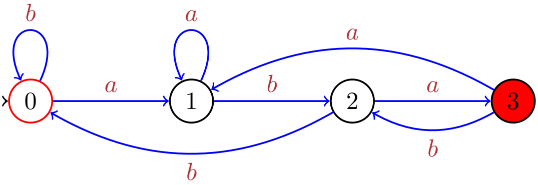
same construction as Aho-Corasick

for **each match** ring the **bell**

$aaaaaba$🔔$bbaba$🔔ba🔔

Knuth-Morris-Pratt automaton (1977) - Only one word

$$P = aba$$



same construction as Aho-Corasick

for **each match** ring the **bell**

$aaaaaba\ bbaba\ ba\ bb$

Pattern matching and Statistics - Regular patterns

1. We learned how to compute the
   - **number of matches** of a **finite pattern**
   - **in a particular text**

Pattern matching and Statistics - Regular patterns

1. We learned how to compute the
   - **number of matches** of a **finite pattern**
   - **in a particular text**
2. **In a random text**, **what** about
   - **finding** the **occurrences** of a **Regular Expression** (*i.e*, the **number of positions at which a match is found**)
   - and **counting them**

Tools and Aim - Generating Functions

For a **given pattern** P, we want to compute

$$F(z,u) = \sum_{n\geq 0, k\geq 0} f_{n,k} u^k z^n$$

$$\text{where} \quad f_{n,k} = \mathbf{P}\left(\begin{array}{l} P \text{ occurs } k \text{ times} \\ \text{in a } \textbf{random text of length } n \end{array} \right)$$

Tools and Aim - Generating Functions

For a **given pattern** P, we want to compute

$$F(z,u) = \sum_{n\geq 0, k\geq 0} f_{n,k} u^k z^n$$

where $\quad f_{n,k} = \mathbf{P}\left(\begin{array}{l} P \text{ occurs } k \text{ times} \\ \text{in a } \textbf{random text of length } n \end{array} \right)$

If X_n is the random variable

- counting the number of occurrences of P
- in a random text of size n

$$F(z,u) = \sum_{n\geq 0, k\geq 0} f_{n,k} u^k z^n = \sum_{n\geq 0} z^n \sum_{k\geq 0} \mathbf{P}(X_n = k) u^k$$

The variables z and u are formal variables

- z is related to the length of the texts
- u is related to the number of occurrences of P

Counting with Regular Expressions - The right language

1. **Input**:
   - a finite alphabet $\mathcal{A}$
   - a regular expression $\mathcal{R}$
2. **Output**:

$$F(z, u) = \sum_{n \geq 0, k \geq 0} f_{n,k} u^k z^n$$

Counting with Regular Expressions - The right language

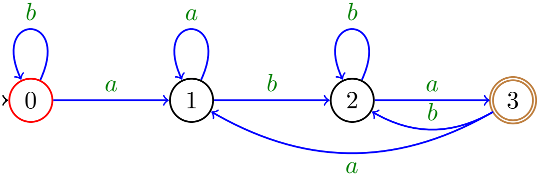
1. **Input**:
   - a finite alphabet $\mathcal{A}$
   - a regular expression $\mathcal{R}$
2. **Output**:

$$F(z, u) = \sum_{n \geq 0, k \geq 0} f_{n,k} u^k z^n$$

- **Method**
  1. Build the DFA recognizing $\mathcal{A}^\star.\mathcal{R}$
  2. Use a variant of Chomsky-Schützenberger to ring the bell and produce the variable u

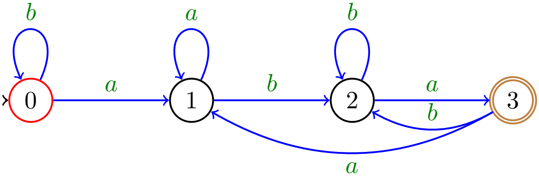
Counting the number of occurrences of ab^+a

$$P = \mathcal{A}^\star ab^+a = (a+b)^\star ab^+a$$



Counting the number of occurrences of ab^+a

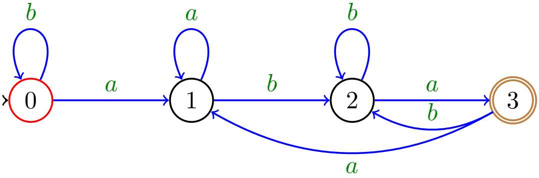
$$P = \mathcal{A}^\star ab^+a = (a+b)^\star ab^+a$$



$$\mathcal{L}_0 = a.\mathcal{L}_1 + b.\mathcal{L}_0 \qquad L_0(a,b\quad) = a \times L_1(a,b\quad) + b \times L_0(a,b\quad)$$
$$\mathcal{L}_1 = a.\mathcal{L}_1 + b.\mathcal{L}_2 \qquad L_1(a,b\quad) = a \times L_1(a,b\quad) + b \times L_2(a,b\quad)$$

Counting the number of occurrences of ab^+a

$$P = \mathcal{A}^\star ab^+a = (a+b)^\star ab^+a$$

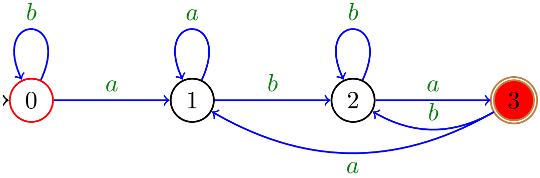


$$\begin{aligned}
\mathcal{L}_0 &= a.\mathcal{L}_1 + b.\mathcal{L}_0 \\
\mathcal{L}_1 &= a.\mathcal{L}_1 + b.\mathcal{L}_2 \\
\mathcal{L}_2 &= a.\mathcal{L}_3 + b.\mathcal{L}_2
\end{aligned}$$

$$\begin{aligned}
L_0(a,b\quad) &= a\times L_1(a,b\quad) + b\times L_0(a,b\quad) \\
L_1(a,b\quad) &= a\times L_1(a,b\quad) + b\times L_2(a,b\quad)
\end{aligned}$$

Counting the number of occurrences of ab^+a

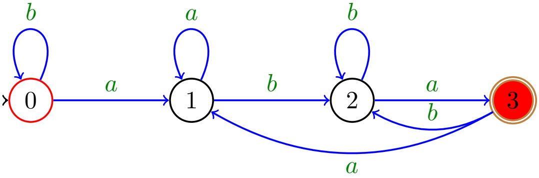
$$P = \mathcal{A}^\star ab^+a = (a+b)^\star ab^+a$$



$$\mathcal{L}_0 = a.\mathcal{L}_1 + b.\mathcal{L}_0 \qquad L_0(a,b\quad) = a \times L_1(a,b\quad) + b \times L_0(a,b\quad)$$
$$\mathcal{L}_1 = a.\mathcal{L}_1 + b.\mathcal{L}_2 \qquad L_1(a,b\quad) = a \times L_1(a,b\quad) + b \times L_2(a,b\quad)$$
$$\mathcal{L}_2 = a.\mathcal{L}_3 \; + b.\mathcal{L}_2$$

Counting the number of occurrences of ab^+a

$$P = \mathcal{A}^\star ab^+a = (a+b)^\star ab^+a$$

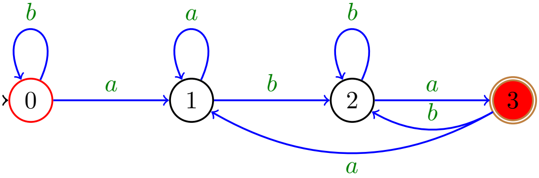


$$\begin{aligned}
\mathcal{L}_0 &= a.\mathcal{L}_1 + b.\mathcal{L}_0 \\
\mathcal{L}_1 &= a.\mathcal{L}_1 + b.\mathcal{L}_2 \\
\mathcal{L}_2 &= a.\mathcal{L}_3 + b.\mathcal{L}_2
\end{aligned}$$

$$\begin{aligned}
L_0(a,b\quad) &= a \times L_1(a,b\quad) + b \times L_0(a,b\quad) \\
L_1(a,b\quad) &= a \times L_1(a,b\quad) + b \times L_2(a,b\quad)
\end{aligned}$$

Counting the number of occurrences of ab^+a

$$P = \mathcal{A}^\star ab^+a = (a+b)^\star ab^+a$$

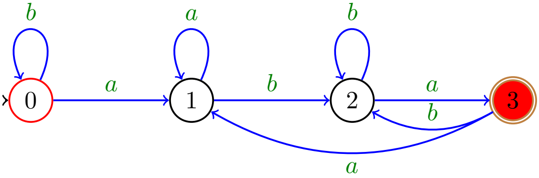


$$\begin{aligned}
\mathcal{L}_0 &= a.\mathcal{L}_1 + b.\mathcal{L}_0 \\
\mathcal{L}_1 &= a.\mathcal{L}_1 + b.\mathcal{L}_2 \\
\mathcal{L}_2 &= a.\mathcal{L}_3 + b.\mathcal{L}_2
\end{aligned}$$

$$\begin{aligned}
L_0(a,b\quad) &= a \times L_1(a,b\quad) + b \times L_0(a,b\quad) \\
L_1(a,b\quad) &= a \times L_1(a,b\quad) + b \times L_2(a,b\quad) \\
L_2(a,b\quad) &= a \times u \times L_3(a,b\quad) + b \times L_2(a,b\quad)
\end{aligned}$$

Counting the number of occurrences of ab^+a

$$P = \mathcal{A}^\star ab^+a = (a+b)^\star ab^+a$$

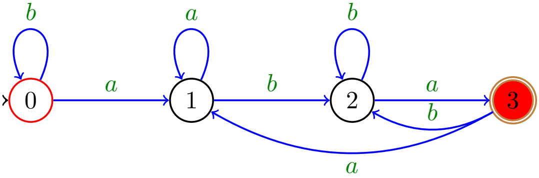


$$\begin{aligned}
\mathcal{L}_0 &= a.\mathcal{L}_1 + b.\mathcal{L}_0 \\
\mathcal{L}_1 &= a.\mathcal{L}_1 + b.\mathcal{L}_2 \\
\mathcal{L}_2 &= a.\mathcal{L}_3 + b.\mathcal{L}_2
\end{aligned}$$

$$\begin{aligned}
L_0(a,b,u) &= a \times L_1(a,b,u) + b \times L_0(a,b,u) \\
L_1(a,b,u) &= a \times L_1(a,b,u) + b \times L_2(a,b,u) \\
L_2(a,b,u) &= a \times u \times L_3(a,b,u) + b \times L_2(a,b,u)
\end{aligned}$$

Counting the number of occurrences of ab^+a

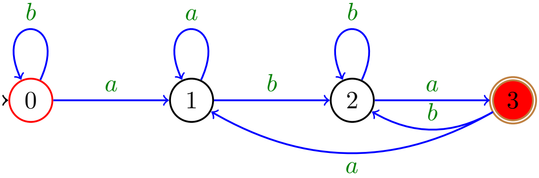
$$P = \mathcal{A}^\star ab^+a = (a+b)^\star ab^+a$$



$$\mathcal{L}_0 = a.\mathcal{L}_1 + b.\mathcal{L}_0 \qquad L_0(a,b,u) = a \times L_1(a,b,u) + b \times L_0(a,b,u)$$
$$\mathcal{L}_1 = a.\mathcal{L}_1 + b.\mathcal{L}_2 \qquad L_1(a,b,u) = a \times L_1(a,b,u) + b \times L_2(a,b,u)$$
$$\mathcal{L}_2 = a.\mathcal{L}_3 + b.\mathcal{L}_2 \qquad L_2(a,b,u) = a \times u \times L_3(a,b,u) + b \times L_2(a,b,u)$$
$$\mathcal{L}_3 = a.\mathcal{L}_1 + b.\mathcal{L}_2 + \epsilon \qquad L_3(a,b,u) = a \times L_1(a,b,u) + b \times L_2(a,b,u) + 1$$

Counting the number of occurrences of ab^+a

$$P = \mathcal{A}^\star ab^+a = (a+b)^\star ab^+a$$



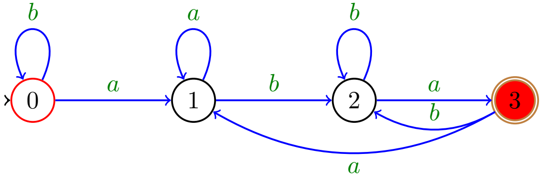
$$\begin{aligned}
\mathcal{L}_0 &= a.\mathcal{L}_1 + b.\mathcal{L}_0 \\
\mathcal{L}_1 &= a.\mathcal{L}_1 + b.\mathcal{L}_2 \\
\mathcal{L}_2 &= a.\mathcal{L}_3 + b.\mathcal{L}_2 \\
\mathcal{L}_3 &= a.\mathcal{L}_1 + b.\mathcal{L}_2 + \epsilon
\end{aligned}$$

$$\begin{aligned}
L_0(a,b,u) &= a \times L_1(a,b,u) + b \times L_0(a,b,u) \\
L_1(a,b,u) &= a \times L_1(a,b,u) + b \times L_2(a,b,u) \\
L_2(a,b,u) &= a \times u \times L_3(a,b,u) + b \times L_2(a,b,u) \\
L_3(a,b,u) &= a \times L_1(a,b,u) + b \times L_2(a,b,u) + 1
\end{aligned}$$

solve:

Counting the number of occurrences of ab^+a

$$P = \mathcal{A}^\star ab^+a = (a+b)^\star ab^+a$$

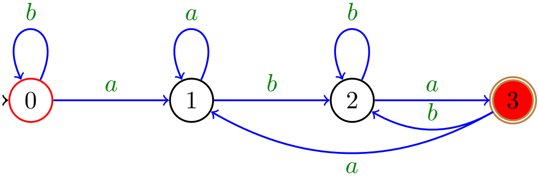


$$\begin{aligned}
\mathcal{L}_0 &= a.\mathcal{L}_1 + b.\mathcal{L}_0 & L_0(a,b,u) &= a \times L_1(a,b,u) + b \times L_0(a,b,u)\\
\mathcal{L}_1 &= a.\mathcal{L}_1 + b.\mathcal{L}_2 & L_1(a,b,u) &= a \times L_1(a,b,u) + b \times L_2(a,b,u)\\
\mathcal{L}_2 &= a.\mathcal{L}_3 + b.\mathcal{L}_2 & L_2(a,b,u) &= a \times u \times L_3(a,b,u) + b \times L_2(a,b,u)\\
\mathcal{L}_3 &= a.\mathcal{L}_1 + b.\mathcal{L}_2 + \epsilon & L_3(a,b,u) &= a \times L_1(a,b,u) + b \times L_2(a,b,u) + 1
\end{aligned}$$

solve: $$L_0(a,b,u) = \frac{1 - b + ab - uab}{1 - a - 2b + 2ab + b^2 - ab^2 - u(ab - ab^2)}$$

Counting the number of occurrences of ab^+a

$$P = \mathcal{A}^\star ab^+a = (a+b)^\star ab^+a$$



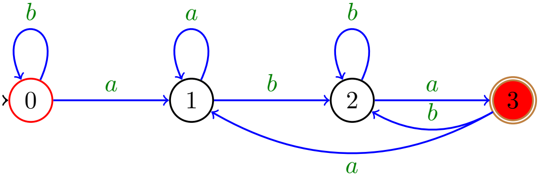
$$\begin{aligned}
\mathcal{L}_0 &= a.\mathcal{L}_1 + b.\mathcal{L}_0 & L_0(a,b,u) &= a\times L_1(a,b,u) + b\times L_0(a,b,u)\\
\mathcal{L}_1 &= a.\mathcal{L}_1 + b.\mathcal{L}_2 & L_1(a,b,u) &= a\times L_1(a,b,u) + b\times L_2(a,b,u)\\
\mathcal{L}_2 &= a.\mathcal{L}_3 + b.\mathcal{L}_2 & L_2(a,b,u) &= a\times u\times L_3(a,b,u) + b\times L_2(a,b,u)\\
\mathcal{L}_3 &= a.\mathcal{L}_1 + b.\mathcal{L}_2 + \epsilon & L_3(a,b,u) &= a\times L_1(a,b,u) + b\times L_2(a,b,u) + 1
\end{aligned}$$

solve:
$$L_0(a,b,u) = \frac{1-b+ab-uab}{1-a-2b+2ab+b^2-ab^2-u(ab-ab^2)}$$

$$F(z,u) = \sum f_{n,k}u^k z^n = L_0(\mathbf{P}(a)z, \mathbf{P}(b)z, u)$$

Counting the number of occurrences of ab^+a

$$P = \mathcal{A}^\star ab^+a = (a+b)^\star ab^+a$$



$$\begin{array}{ll}
\mathcal{L}_0 = a.\mathcal{L}_1 + b.\mathcal{L}_0 & L_0(a,b,u) = a \times L_1(a,b,u) + b \times L_0(a,b,u) \\
\mathcal{L}_1 = a.\mathcal{L}_1 + b.\mathcal{L}_2 & L_1(a,b,u) = a \times L_1(a,b,u) + b \times L_2(a,b,u) \\
\mathcal{L}_2 = a.\mathcal{L}_3 + b.\mathcal{L}_2 & L_2(a,b,u) = a \times u \times L_3(a,b,u) + b \times L_2(a,b,u) \\
\mathcal{L}_3 = a.\mathcal{L}_1 + b.\mathcal{L}_2 + \epsilon & L_3(a,b,u) = a \times L_1(a,b,u) + b \times L_2(a,b,u) + 1
\end{array}$$

solve: $$L_0(a,b,u) = \frac{1 - b + ab - uab}{1 - a - 2b + 2ab + b^2 - ab^2 - u(ab - ab^2)}$$

$$F(z,u) = \sum f_{n,k} u^k z^n = L_0(\mathbf{P}(a)z, \mathbf{P}(b)z, u)$$

$$\mathbf{P}(a) = \mathbf{P}(b) = \frac{1}{2} \quad \rightsquigarrow \quad F(z,u) = \frac{8 - 4z + 2z^2 - 2uz^2}{8 - 12z + 6z^2 - z^3 - u(2z^2 - z^3)}$$

Exploiting the generating Function

$$R = ab^+a, \quad F(z,u) = \frac{8 - 4z + 2z^2 - 2uz^2}{8 - 12z + 6z^2 - z^3 - u(2z^2 - z^3)}$$

Exploiting the generating Function

$$R = ab^+a, \quad F(z,u) = \frac{8-4z+2z^2-2uz^2}{8-12z+6z^2-z^3-u(2z^2-z^3)}$$

- Expand in series with respect to z in the neighborhood of 0

Exploiting the generating Function

$$R = ab^{+}a, \quad F(z,u) = \frac{8 - 4z + 2z^2 - 2uz^2}{8 - 12z + 6z^2 - z^3 - u(2z^2 - z^3)}$$

- Expand in series with respect to z in the neighborhood of 0

$$F(z,u) = 1 + z + z^2 + \left(\frac{1}{8}u + \frac{7}{8}\right)z^3 + \left(\frac{5}{16}u + \frac{11}{16}\right)z^4 + \left(\frac{1}{2} + \frac{15}{32}u + \frac{1}{32}u^2\right)z^5 + \mathcal{O}(z^6)$$

Exploiting the generating Function

$$R = ab^+a, \quad F(z,u) = \frac{8 - 4z + 2z^2 - 2uz^2}{8 - 12z + 6z^2 - z^3 - u(2z^2 - z^3)}$$

- Expand in series with respect to z in the neighborhood of 0

$$F(z,u) = 1 + z + z^2 + \left(\frac{1}{8}u + \frac{7}{8}\right)z^3 + \left(\frac{5}{16}u + \frac{11}{16}\right)z^4 + \left(\frac{1}{2} + \frac{15}{32}u + \frac{1}{32}u^2\right)z^5 + \mathcal{O}(z^6)$$

- Compute the generating function of the expectations of the number of occurrences of the pattern

Exploiting the generating Function

$$R = ab^{+}a, \quad F(z,u) = \frac{8 - 4z + 2z^2 - 2uz^2}{8 - 12z + 6z^2 - z^3 - u(2z^2 - z^3)}$$

- Expand in series with respect to z in the neighborhood of 0

$$F(z,u) = 1 + z + z^2 + \left(\frac{1}{8}u + \frac{7}{8}\right)z^3 + \left(\frac{5}{16}u + \frac{11}{16}\right)z^4 + \left(\frac{1}{2} + \frac{15}{32}u + \frac{1}{32}u^2\right)z^5 + \mathcal{O}(z^6)$$

- Compute the generating function of the expectations of the number of occurrences of the pattern

$$E(z) = \sum_n \mathbf{E}(X_n)z^n = \left.\frac{\partial F(z,u)}{\partial u}\right|_{u=1} = -\frac{1}{2}\frac{z^2}{1-z} + \frac{1}{4}\frac{z^2}{1-\frac{1}{2}z} + \frac{1}{4}\frac{z^2}{(1-z)^2}$$

Exploiting the generating Function

$$R = ab^+a, \quad F(z,u) = \frac{8 - 4z + 2z^2 - 2uz^2}{8 - 12z + 6z^2 - z^3 - u(2z^2 - z^3)}$$

- Expand in series with respect to z in the neighborhood of 0

$$F(z,u) = 1+z+z^2+\left(\frac{1}{8}u+\frac{7}{8}\right)z^3+\left(\frac{5}{16}u+\frac{11}{16}\right)z^4+\left(\frac{1}{2}+\frac{15}{32}u+\frac{1}{32}u^2\right)z^5+\mathcal{O}(z^6)$$

- Compute the generating function of the expectations of the number of occurrences of the pattern

$$E(z) = \sum_n \mathbf{E}(X_n)z^n = \left.\frac{\partial F(z,u)}{\partial u}\right|_{u=1} = -\frac{1}{2}\frac{z^2}{1-z}+\frac{1}{4}\frac{z^2}{1-\frac{1}{2}z}+\frac{1}{4}\frac{z^2}{(1-z)^2}$$

- Get $\mathbf{E}(X_n)$

Exploiting the generating Function

$$R = ab^+a, \quad F(z,u) = \frac{8 - 4z + 2z^2 - 2uz^2}{8 - 12z + 6z^2 - z^3 - u(2z^2 - z^3)}$$

- Expand in series with respect to z in the neighborhood of 0

$$F(z,u) = 1+z+z^2+\left(\frac{1}{8}u+\frac{7}{8}\right)z^3+\left(\frac{5}{16}u+\frac{11}{16}\right)z^4+\left(\frac{1}{2}+\frac{15}{32}u+\frac{1}{32}u^2\right)z^5+\mathcal{O}(z^6)$$

- Compute the generating function of the expectations of the number of occurrences of the pattern

$$E(z) = \sum_n \mathbf{E}(X_n)z^n = \left.\frac{\partial F(z,u)}{\partial u}\right|_{u=1} = -\frac{1}{2}\frac{z^2}{1-z}+\frac{1}{4}\frac{z^2}{1-\frac{1}{2}z}+\frac{1}{4}\frac{z^2}{(1-z)^2}$$

- Get $\mathbf{E}(X_n)$

$$\mathbf{E}(X_n) = -\frac{1}{2}+2^{-n}+\frac{1}{4}(n-1) = \frac{1}{4}(n-3)+2^{-n}$$

Exploiting the generating Function

$$R = ab^{+}a, \quad F(z, u) = \frac{8 - 4z + 2z^2 - 2uz^2}{8 - 12z + 6z^2 - z^3 - u(2z^2 - z^3)}$$

Exploiting the generating Function

$$R = ab^+a, \quad F(z,u) = \frac{8 - 4z + 2z^2 - 2uz^2}{8 - 12z + 6z^2 - z^3 - u(2z^2 - z^3)}$$

- **Generating function of the Second Moment**

Exploiting the generating Function

$$R = ab^{+}a, \quad F(z,u) = \frac{8 - 4z + 2z^2 - 2uz^2}{8 - 12z + 6z^2 - z^3 - u(2z^2 - z^3)}$$

- **Generating function of the Second Moment**

$$M_2(z) = \sum_{n \geq 0} \mathbf{E}(X_n^2) z^n = \left. \frac{\partial}{\partial u} u \frac{\partial F(z,u)}{\partial u} \right|_{u=1}$$

Exploiting the generating Function

$$R = ab^{+}a, \quad F(z,u) = \frac{8 - 4z + 2z^2 - 2uz^2}{8 - 12z + 6z^2 - z^3 - u(2z^2 - z^3)}$$

- **Generating function of the Second Moment**

$$M_2(z) = \sum_{n \geq 0} \mathbf{E}(X_n^2) z^n = \frac{\partial}{\partial u} u \frac{\partial F(z,u)}{\partial u}\bigg|_{u=1}$$

$$M_2(z) = \frac{1}{4}\frac{z^2(z^2-2)}{1-z} - \frac{1}{4}\frac{z^2(z^2-1)}{(1-z)^2} - \frac{1}{8}\frac{z^2(z^2-2)}{1-\dfrac{z}{2}} + \frac{1}{8}\frac{z^4}{(1-z)^3}$$

Exploiting the generating Function

$$R = ab^{+}a, \quad F(z,u) = \frac{8-4z+2z^2-2uz^2}{8-12z+6z^2-z^3-u(2z^2-z^3)}$$

- **Generating function of the Second Moment**

$$M_2(z) = \sum_{n\geq 0} \mathbf{E}(X_n^2) z^n = \frac{\partial}{\partial u} u \frac{\partial F(z,u)}{\partial u}\bigg|_{u=1}$$

$$M_2(z) = \frac{1}{4}\frac{z^2(z^2-2)}{1-z} - \frac{1}{4}\frac{z^2(z^2-1)}{(1-z)^2} - \frac{1}{8}\frac{z^2(z^2-2)}{1-\frac{z}{2}} + \frac{1}{8}\frac{z^4}{(1-z)^3}$$

- **Extract** the nth. **Taylor coefficient**

$$\mathbf{E}(X_n^2) = [z^n]M_2(z) = \frac{1}{16}n^2 - \frac{5}{16}n + \frac{5}{8} - 2^{-n}$$

Exploiting the generating Function

$$R = ab^+a, \quad F(z,u) = \frac{8-4z+2z^2-2uz^2}{8-12z+6z^2-z^3-u(2z^2-z^3)}$$

- **Generating function of the Second Moment**

$$M_2(z) = \sum_{n\geq 0} \mathbf{E}(X_n^2) z^n = \frac{\partial}{\partial u} u \frac{\partial F(z,u)}{\partial u}\Bigg|_{u=1}$$

$$M_2(z) = \frac{1}{4}\frac{z^2(z^2-2)}{1-z} - \frac{1}{4}\frac{z^2(z^2-1)}{(1-z)^2} - \frac{1}{8}\frac{z^2(z^2-2)}{1-\frac{z}{2}} + \frac{1}{8}\frac{z^4}{(1-z)^3}$$

- **Extract** the nth. **Taylor coefficient**

$$\mathbf{E}(X_n^2) = [z^n]M_2(z) = \frac{1}{16}n^2 - \frac{5}{16}n + \frac{5}{8} - 2^{-n}$$

- **Standard Deviation** σ_n

$$\sigma_n = \sqrt{\mathbf{E}(X_n^2) - \mathbf{E}^2(X_n)} = \frac{1}{4}\sqrt{n+1-2^{-n+3}n+2^{-n+3}-4^{-n+2}}$$

Limit law

- **Laplace transform $\mathbf{L}$** of a **random variable X**

$$\mathbf{L}(X, t) = \mathbf{E}(e^{tX})$$

- **Laplace transform** of a **standard Gaussian variable $\mathcal{N}$**

$$\mathbf{L}(\mathcal{N}, t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{tx} e^{-x^2/2} dx = e^{t^2/2}$$

Limit law

- **Laplace transform** $\mathbf{L}$ of a **random variable** X

$$\mathbf{L}(X, t) = \mathbf{E}(e^{tX})$$

- **Laplace transform** of a **standard Gaussian variable** $\mathcal{N}$

$$\mathbf{L}(\mathcal{N}, t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{tx} e^{-x^2/2} dx = e^{t^2/2}$$

Theorem (Paul Lévy Continuity Theorem - 1925)

If for $t \in [-\alpha, +\alpha]$ $\quad \lim_{n\to\infty} \mathbf{E}(e^{tX_n}) = \mathbf{L}(\mathcal{N}) = e^{t^2/2}$

then $X_n \xrightarrow{\mathcal{D}} \mathcal{N}$ *(convergence in distribution or law)*

$$\lim_{n\to\infty} \mathbf{P}(X_n < x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-w^2/2} dw$$

Limit law of the occurrences of ab^+a

$$F(z,u) = \frac{8-4z+2z^2-2uz^2}{8-12z+6z^2-z^3-u(2z^2-z^3)}$$
$$= -\frac{1-u}{u\left(1-\frac{z}{2}\right)} + \frac{1+\sqrt{u}}{2u\left(1-z\frac{1+\sqrt{u}}{2}\right)} + \frac{1-\sqrt{u}}{2u\left(1-z\frac{1-\sqrt{u}}{2}\right)}$$

Limit law of the occurrences of ab^+a

$$F(z,u) = \frac{8-4z+2z^2-2uz^2}{8-12z+6z^2-z^3-u(2z^2-z^3)}$$

$$= -\frac{1-u}{u\left(1-\frac{z}{2}\right)} + \frac{1+\sqrt{u}}{2u\left(1-z\frac{1+\sqrt{u}}{2}\right)} + \frac{1-\sqrt{u}}{2u\left(1-z\frac{1-\sqrt{u}}{2}\right)}$$

$$\Psi_n(u) = [z^n]F(z,u) = \frac{1}{u}\left(\frac{1+\sqrt{u}}{2}\right)^{n+1} + O\left(\frac{1}{2^n}\right) \qquad \text{for } u \text{ close of } 1$$

Limit law of the occurrences of ab^+a

$$F(z,u) = \frac{8-4z+2z^2-2uz^2}{8-12z+6z^2-z^3-u(2z^2-z^3)}$$
$$= -\frac{1-u}{u\left(1-\frac{z}{2}\right)} + \frac{1+\sqrt{u}}{2u\left(1-z\frac{1+\sqrt{u}}{2}\right)} + \frac{1-\sqrt{u}}{2u\left(1-z\frac{1-\sqrt{u}}{2}\right)}$$

$$\Psi_n(u) = [z^n]F(z,u) = \frac{1}{u}\left(\frac{1+\sqrt{u}}{2}\right)^{n+1} + O\left(\frac{1}{2^n}\right) \qquad \text{for } u \text{ close of } 1$$

We consider $\Psi_n(e^t) = \mathbf{E}(e^{tX_n})$ and the **normalised law** $\dfrac{X_n-\mu_n}{\sigma_n}$

$$\Phi_n(t) = \Psi_n\left(t\frac{X_n-\mu_n}{\sigma_n}\right) = \mathbf{E}\left[\exp\left(\frac{t(X_n-\mu_n)}{\sigma_n}\right)\right] = \exp\left(-\frac{\mu_n t}{\sigma_n}\right)\mathbf{E}\left[\exp\left(\frac{tX_n}{\sigma_n}\right)\right]$$

Limit law of the occurrences of ab^+a

$$F(z,u) = \frac{8-4z+2z^2-2uz^2}{8-12z+6z^2-z^3-u(2z^2-z^3)}$$
$$= -\frac{1-u}{u\left(1-\frac{z}{2}\right)} + \frac{1+\sqrt{u}}{2u\left(1-z\frac{1+\sqrt{u}}{2}\right)} + \frac{1-\sqrt{u}}{2u\left(1-z\frac{1-\sqrt{u}}{2}\right)}$$

$$\Psi_n(u) = [z^n]F(z,u) = \frac{1}{u}\left(\frac{1+\sqrt{u}}{2}\right)^{n+1} + O\left(\frac{1}{2^n}\right) \qquad \text{for } u \text{ close of } 1$$

We consider $\Psi_n(e^t) = \mathbf{E}(e^{tX_n})$ and the **normalised law** $\dfrac{X_n-\mu_n}{\sigma_n}$

$$\Phi_n(t) = \Psi_n\left(t\frac{X_n-\mu_n}{\sigma_n}\right) = \mathbf{E}\left[\exp\left(\frac{t(X_n-\mu_n)}{\sigma_n}\right)\right] = \exp\left(-\frac{\mu_n t}{\sigma_n}\right)\mathbf{E}\left[\exp\left(\frac{tX_n}{\sigma_n}\right)\right]$$

We substitute: $\quad \mu_n = \dfrac{n-3}{4} + \mathcal{O}(2^{-n}), \quad \sigma_n = \dfrac{\sqrt{n+1}}{4} + \mathcal{O}(2^{-n})$

Limit law of the occurrences of ab^+a

$$F(z,u) = \frac{8-4z+2z^2-2uz^2}{8-12z+6z^2-z^3-u(2z^2-z^3)}$$
$$= -\frac{1-u}{u\left(1-\frac{z}{2}\right)} + \frac{1+\sqrt{u}}{2u\left(1-z\frac{1+\sqrt{u}}{2}\right)} + \frac{1-\sqrt{u}}{2u\left(1-z\frac{1-\sqrt{u}}{2}\right)}$$

$$\Psi_n(u) = [z^n]F(z,u) = \frac{1}{u}\left(\frac{1+\sqrt{u}}{2}\right)^{n+1} + O\left(\frac{1}{2^n}\right) \qquad \text{for } u \text{ close of } 1$$

We consider $\Psi_n(e^t) = \mathbf{E}(e^{tX_n})$ and the **normalised law** $\frac{X_n-\mu_n}{\sigma_n}$

$$\Phi_n(t) = \Psi_n\left(t\frac{X_n-\mu_n}{\sigma_n}\right) = \mathbf{E}\left[\exp\left(\frac{t(X_n-\mu_n)}{\sigma_n}\right)\right] = \exp\left(-\frac{\mu_n t}{\sigma_n}\right)\mathbf{E}\left[\exp\left(\frac{tX_n}{\sigma_n}\right)\right]$$

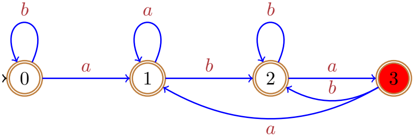
We substitute: $\mu_n = \frac{n-3}{4} + \mathcal{O}(2^{-n}), \quad \sigma_n = \frac{\sqrt{n+1}}{4} + \mathcal{O}(2^{-n})$

In a **neighborhood** of $t=0$, we **expand** $\log(\Phi_n(t))$

$$\log(\Phi_n(t)) = \frac{t^2}{2} - \frac{t^4}{12(n+1)} + \mathcal{O}\left(\frac{t^6}{n^2}\right) \xrightarrow{n\to\infty} \frac{t^2}{2}$$

The Gaussian law is general

$$R = ab^+a \qquad P = \mathcal{A}^\star ab^+a$$

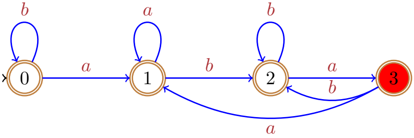


$$\begin{aligned} L_0(z,u) = L_0 &= zp_aL_1 + zp_bL_0 + 1, \\ L_1 &= zp_bL_2 + zp_aL_1 + 1, \\ L_2 &= zp_a u L_3 + zp_bL_2 + 1 \\ L_3 &= zp_aL_1 + zp_bL_2 + 1 \end{aligned}$$

$$\mathbf{L} = \begin{pmatrix} L_0 \\ \vdots \\ L_n \end{pmatrix} = z\mathbb{T}(u)\mathbf{L} + \mathbf{1}$$

The Gaussian law is general

$$R = ab^+a \qquad P = \mathcal{A}^\star ab^+a$$



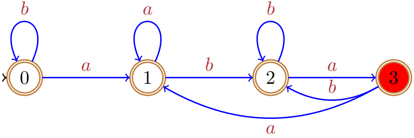
$$\begin{aligned} L_0(z,u) = L_0 &= zp_aL_1 + zp_bL_0 + 1, \\ L_1 &= zp_bL_2 + zp_aL_1 + 1, \\ L_2 &= zp_a u L_3 + zp_bL_2 + 1 \\ L_3 &= zp_aL_1 + zp_bL_2 + 1 \end{aligned}$$

$$\mathbf{L} = \begin{pmatrix} L_0 \\ \vdots \\ L_n \end{pmatrix} = z\mathbb{T}(u)\mathbf{L} + \mathbf{1}$$

general case: $\mathbb{T}(u)$ **positive** $n \times n$ matrix for $u \geq 0$

The Gaussian law is general

$$R = ab^+a \qquad P = \mathcal{A}^\star ab^+a$$



$$\begin{aligned} L_0(z,u) = L_0 &= zp_aL_1 + zp_bL_0 + 1, \\ L_1 &= zp_bL_2 + zp_aL_1 + 1, \\ L_2 &= zp_a u L_3 + zp_bL_2 + 1 \\ L_3 &= zp_aL_1 + zp_bL_2 + 1 \end{aligned} \qquad \mathbf{L} = \begin{pmatrix} L_0 \\ \vdots \\ L_n \end{pmatrix} = z\mathbb{T}(u)\mathbf{L} + \mathbf{1}$$

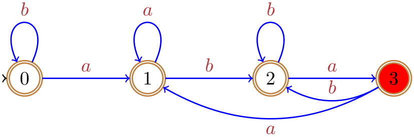
general case: $\mathbb{T}(u)$ **positive** $n \times n$ matrix for $u \geq 0$

Theorem (Perron-Frobenius, 1907-1912)

If $\mathbb{T}(u)$ is positive, irreducible and aperiodic, the dominant eigenvalue is unique, real and positive.

The Gaussian law is general

$$R = ab^+a \qquad P = \mathcal{A}^\star ab^+a$$



$$\begin{aligned} L_0(z,u) = L_0 &= zp_aL_1 + zp_bL_0 + 1, \\ L_1 &= zp_bL_2 + zp_aL_1 + 1, \\ L_2 &= zp_a u L_3 + zp_bL_2 + 1 \\ L_3 &= zp_aL_1 + zp_bL_2 + 1 \end{aligned} \qquad \mathbf{L} = \begin{pmatrix} L_0 \\ \vdots \\ L_n \end{pmatrix} = z\mathbb{T}(u)\mathbf{L} + \mathbf{1}$$

general case: $\mathbb{T}(u)$ **positive** $n \times n$ matrix for $u \geq 0$

Theorem (Perron-Frobenius, 1907-1912)

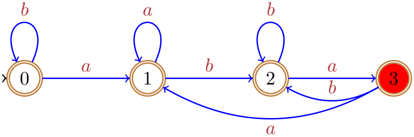
If $\mathbb{T}(u)$ is positive, irreducible and aperiodic, the dominant eigenvalue is unique, real and positive.

$$L_0(z,u) = \frac{P(z,u)}{Q(z,u)} = \frac{P(z,u)}{(1 - z\lambda_1(u)) \cdots (1 - z\lambda_n(u))}$$

$$\lambda_1(u) \text{ dominant } \Longrightarrow \frac{1}{|\lambda_1(u)|} < \frac{1}{|\lambda_2(u)|} \leq \dots$$

Perron-Frobenius conditions

$R = ab^+a \qquad P = \mathcal{A}^\star ab^+a$



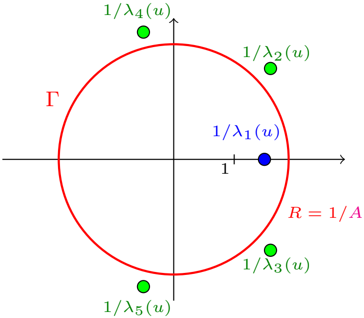
In the context of automata,

- **irreducibility**: from any state, any other state can be reached (The above automaton is not irreducible)
- **primitivity**: there exists a large enough e such that any state can be reached by any other state in exactly e steps

Remarks

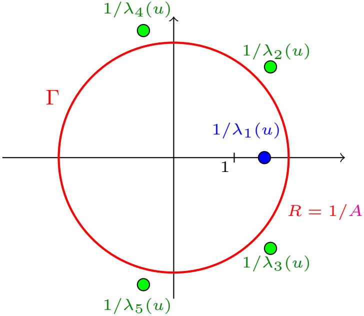
- The above automaton with initial state 1 and states $1, 2, 3$, is irreducible and primitive
- The automaton with states $0, 1, 2, 3$ is such that $L_0 = \dfrac{L_1}{1 - zp_b} + \dfrac{1}{1 - zp_b}$
- For $u = 1$, we have $L_0 = L_1 = L_2 = L_3 = 1/(1 - z)$
- by continuity, $\lambda_1(u)$ is close of 1 for $u \in [1 - \epsilon, 1 + \epsilon]$
- for L_0, we have $\dfrac{1}{\lambda_1(u)} < \dfrac{1}{p_b}$

Uniform Separation Property with respect to n



$$
\begin{aligned}
p_n(u) &= [z^n]F(z,u) = \frac{1}{2i\pi}\oint_{\Gamma} \frac{dz}{z^{n+1}} F(z,u), \\
&= \frac{1}{2i\pi}\oint_{\Gamma} \frac{c(u)}{z^{n+1}(1-\lambda_1(u)z)} + \frac{1}{z^{n+1}} g(z,u)\,dz, \\
&= c(u)\lambda_1(u)^n(1+O(A^n)) \qquad (A<1)
\end{aligned}
$$

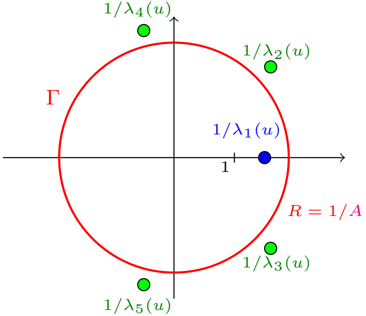
Uniform Separation Property with respect to n



$$
\begin{aligned}
p_n(u) &= [z^n]F(z,u) = \frac{1}{2i\pi}\oint_\Gamma \frac{dz}{z^{n+1}}F(z,u),\\
&= \frac{1}{2i\pi}\oint_\Gamma \frac{c(u)}{z^{n+1}(1-\lambda_1(u)z)} + \frac{1}{z^{n+1}}g(z,u)\,dz,\\
&= c(u)\lambda_1(u)^n(1+O(A^n)) \qquad (A<1)
\end{aligned}
$$

Hwang's **quasi-power** theorem $\rightarrow$ limiting **Gaussian distribution**

Uniform Separation Property with respect to n



$$
\begin{aligned}
p_n(u) &= [z^n]F(z,u) = \frac{1}{2i\pi}\oint_\Gamma \frac{dz}{z^{n+1}}F(z,u), \\
&= \frac{1}{2i\pi}\oint_\Gamma \frac{c(u)}{z^{n+1}(1-\lambda_1(u)z)} + \frac{1}{z^{n+1}}g(z,u)\,dz, \\
&= c(u)\lambda_1(u)^n(1+O(A^n)) \qquad (A<1)
\end{aligned}
$$

Hwang's **quasi-power** theorem $\to$ limiting **Gaussian distribution**

Variability condition: $\lambda''(1)+\lambda'(1)-\lambda'(1)^2 \neq 0 \qquad (\lambda(u) = \lambda_1(u))$

Statistics of one regular motif

Let X_n count the number of occurrences of a regular motif R in a random text of length n.

$$F(z,u) = \sum_{n,k} \mathbf{P}(X_n = k) u^k z^n = \frac{c(u)}{1-\lambda(u)z} + g(z,u)$$

Theorem (N, Salvy, Flajolet - 1999)

Both in the **Bernoulli** *and* **Markov** *model, with $\mathbb{T}(u)$ the fundamental matrix, and $\lambda(u)$ its dominant eigenvalue,*

1. $F(z,u)$ *is* **rational** *and* **can be computed** *explicitly*
2. **Moments** $\begin{cases} \mathrm{E}(X_n) & = \lambda'(1)n + c_1 + O(A^n), \quad (c_1 = c'(1)) \\ \mathrm{Var}(X_n) & = (\lambda''(1) + \lambda'(1) - \lambda'(1)^2)n + c_2 + O(A^n) \\ & \qquad (c_2 = c''(1) + c'(1) - c'(1)^2) \end{cases}$
3. **Limit Gaussian law**: $\quad \Pr\left(\frac{X_n - \mu n}{\sigma\sqrt{n}}\right) \to \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2}\,dt$

[Bourdon, Vallée - 2006] Extension to **dynamical sources**

Counts of $R = ab^+a$

- $\mathbf{P}(a) = \mathbf{P}(b) = \frac{1}{2}$
- X_n number of occurrences of R in a random text of size n
- $\sigma_n = \sqrt{\mathbf{Var}(X_n)} = \frac{\sqrt{n+1}}{4} + \mathcal{O}(2^{-n})$

Variability condition:

$$\mathbf{Var}(X_n) = (\lambda''(1) + \lambda'(1) - \lambda'(1)^2)n + c_2 + O(A^n) = \Theta(n)$$

We have $\mathbf{Var}(X_n) = \Theta(n) \implies$ **normal limit law**

Counts of $R = ab^\star$

$$\mathbf{P}(a) = \mathbf{P}(b) = \frac{1}{2}$$

$$F(z,u) = \sum_{n\geq 0}\sum_{k\geq 0}\mathbf{P}(X_n = k)u^k z^n = \frac{uz/2 - 1}{1 - z/2 - uz + uz^2}$$

$$\begin{cases} \mathbf{E}(X_n) = n - 1 + 2^{-n} \\ \mathbf{E}(X_n^2) = n^2 - 2n + 3 - 3\times 2^{-n} \\ \mathbf{Var}(X_n) = 2 - (2n+1)2^{-n} - 4^{-n} \\ \lim_{n\to\infty} \mathbf{Var}(X_n) = 2 \end{cases}$$

- The **variation condition** is **not verified**
- The **limiting law** is **not normal**

Hwang's Quasi-Power theorem - Gaussian form

Notation: $m(f) = \frac{f'(1)}{f(1)}, \qquad v(f) = \frac{f"(1)}{f(1)} + \frac{f'(1)}{f(1)} - \left(\frac{f'(1)}{f(1)}\right)^2$

Theorem (Hwang 1994)

Let the X_n be non-negative discrete random variables (supported by $\mathbb{Z}_{\geq 0}$) with probability generating function $p_n(u)$. Assume that, uniformly in a complex neighborhood of $u = 1$, for sequences $\beta_n, \kappa_n \to \infty$, there holds

$$p_n(u) = A(u).B(u)^{\beta_n}\left(1 + \mathcal{O}\left(\frac{1}{\kappa_n}\right)\right),$$

where $A(u), B(u)$ are analytic at $u = 1$ and $A(1) = B(1) = 1$. Assume finally that $B(u)$ satisfies the so-called "variability condition",

$$v(B(u)) \equiv B''(1) + B'(1) - B'(1)^2 \neq 0.$$

Under these conditions, the mean and variance of X_n satisfy

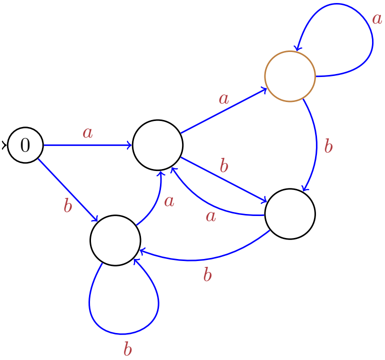
$$\begin{aligned} \mu_n \equiv \quad & \mathbf{E}(X_n) = \quad & \beta_n m(B(1)) + m(A(1)) + \mathcal{O}\left(\kappa_n^{-1}\right) \\ \sigma_n^2 \equiv \quad & \mathbf{Var}(X_n) = \quad & \beta_n v(B(1)) + v(A(1)) + \mathcal{O}\left(\kappa_n^{-1}\right). \end{aligned}$$

The distribution of X_n is, after standardization, asymptotically Gaussian,

$$Pr\left\{\frac{X_n - \mathbf{E}(X_n)}{\sqrt{\mathbf{Var}(X_n)}} \leq x\right\} = \mathcal{N}(x) + \mathcal{O}\left(\frac{1}{\kappa_n} + \frac{1}{\sqrt{\beta_n}}\right),$$

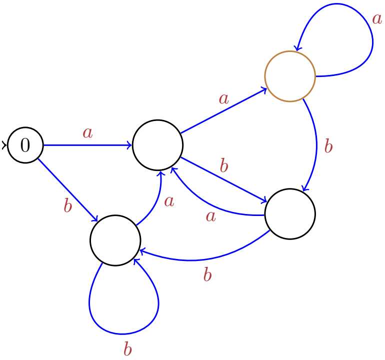
What about counting with several motifs simultaneously?

$P = \{a, aa, ab, b\}$ **Several Finite Motifs**



What about counting with several motifs simultaneously?

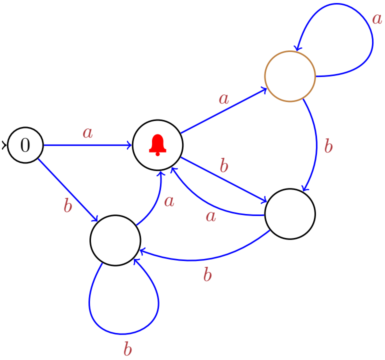
$P = \{a, aa, ab, b\}$ **Several Finite Motifs**



Where are the **bells**?

What about counting with several motifs simultaneously?

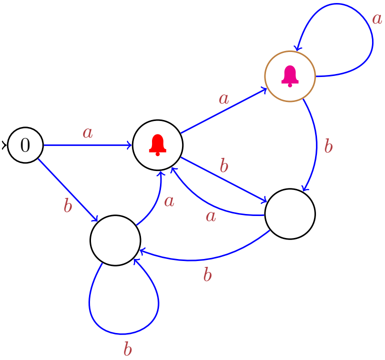
$P = \{a, aa, ab, b\}$ **Several Finite Motifs**



Where are the **bells**?

What about counting with several motifs simultaneously?

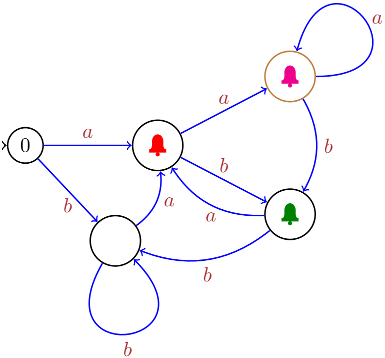
$P = \{a, aa, ab, b\}$ **Several Finite Motifs**



Where are the **bells**?

What about counting with several motifs simultaneously?

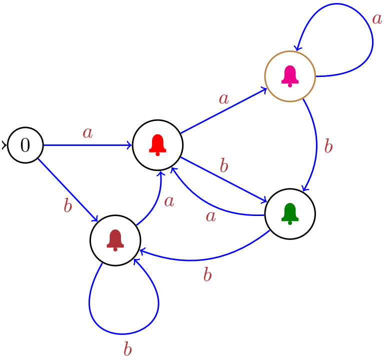
$P = \{a, aa, ab, b\}$ **Several Finite Motifs**



Where are the **bells**?

What about counting with several motifs simultaneously?

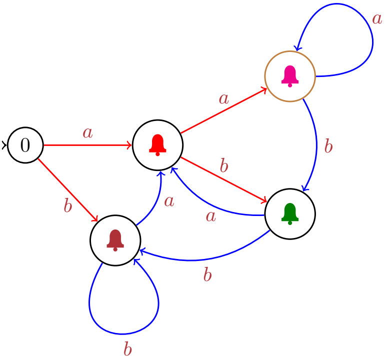
$P = \{a, aa, ab, b\}$ **Several Finite Motifs**



Where are the **bells**?

What about counting with several motifs simultaneously?

$P = \{a, aa, ab, b\}$ **Several Finite Motifs**

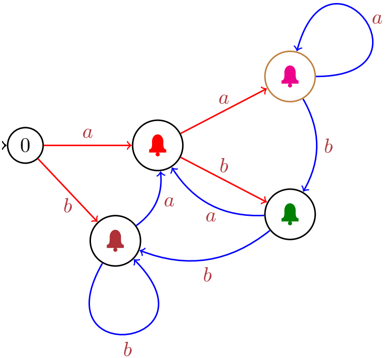


Where are the **bells**?

Easy: upon some **nodes** of the **trie**

What about counting with several motifs simultaneously?

$P = \{a, aa, ab, b\}$ **Several Finite Motifs**



Where are the **bells**?

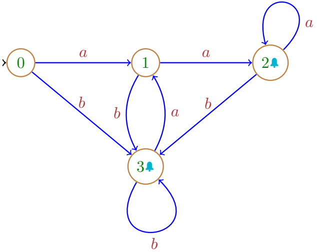
Easy: upon some **nodes** of the **trie**

Not so easy for a **general regular motif**

Product of Marked Automata

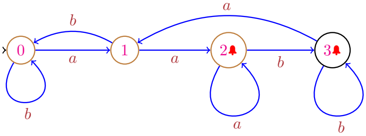
$U = aa + b$

$\mathrm{AutoU} = (\mathcal{A}, 0, Q, \delta, F = Q, \mathrm{Mark} = \{2, 3\})$

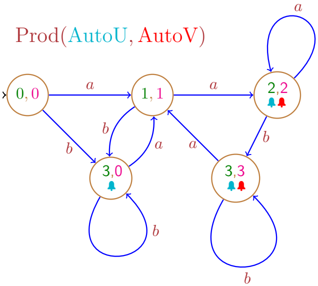


$V = b^{\star}aab^{\star}$;

$\mathrm{AutoV} = (\mathcal{A}, 0, Q, \delta, F = Q, \mathrm{Mark} = \{2, 3\})$



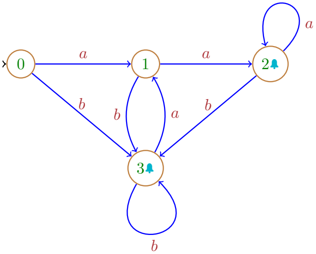
Prod(AutoU, AutoV)



Product of Marked Automata

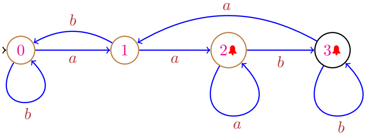
$U = aa + b$

$\mathrm{AutoU} = (\mathcal{A}, 0, Q, \delta, F = Q, \mathrm{Mark} = \{2, 3\})$

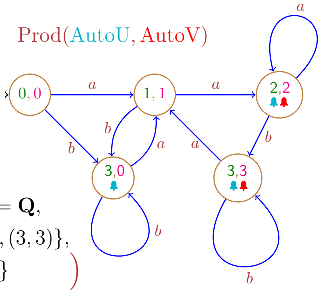


$V = b^\star aab^\star;$

$\mathrm{AutoV} = (\mathcal{A}, 0, Q, \delta, F = Q, \mathrm{Mark} = \{2, 3\})$



Prod(AutoU, AutoV)



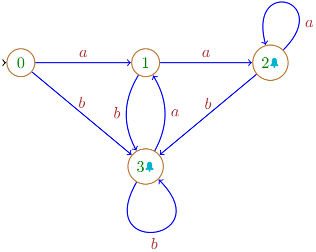
$$\mathrm{Prod}(\mathrm{AutoU}, \mathrm{AutoV}) = \Big(\mathcal{A}, (0, 0), \mathbf{Q} \subseteq Q \times Q, \Delta, \mathbf{F} = \mathbf{Q}, \\ \mathrm{Mark}_1 = \{(2, 2), (3, 0), (3, 3)\}, \\ \mathrm{Mark}_2 = \{(2, 2), (3, 3)\} \Big)$$

$$\Delta\big((q_i, q_j), (\ell_1, \ell_2)\big) = \big(\delta(q_i, \ell_1), \delta(q_j, \ell_2)\big)$$

Product of Marked Automata

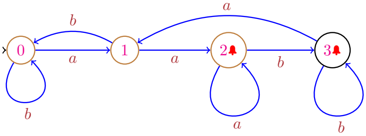
$U = aa + b$

$\text{AutoU} = (\mathcal{A}, 0, Q, \delta, F = Q, \text{Mark} = \{2, 3\})$

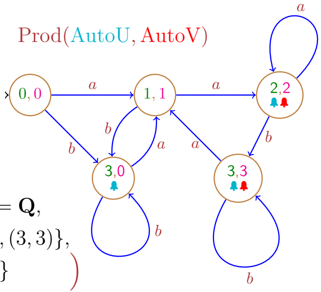


$V = b^{\star}aab^{\star};$

$\text{AutoV} = (\mathcal{A}, 0, Q, \delta, F = Q, \text{Mark} = \{2, 3\})$



$\text{Prod}(\text{AutoU}, \text{AutoV})$

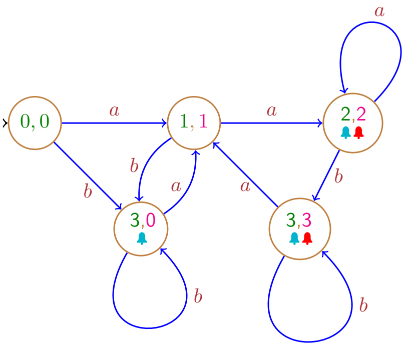


$$\text{Prod}(\text{AutoU}, \text{AutoV}) = \Big(\mathcal{A}, (0, 0), \mathbf{Q} \subseteq Q \times Q, \Delta, \mathbf{F} = \mathbf{Q}, \\ \text{Mark}_1 = \{(2, 2), (3, 0), (3, 3)\}, \\ \text{Mark}_2 = \{(2, 2), (3, 3)\} \Big)$$

$$\Delta\big((q_i, q_j), (\ell_1, \ell_2)\big) = \big(\delta(q_i, \ell_1), \delta(q_j, \ell_2)\big)$$

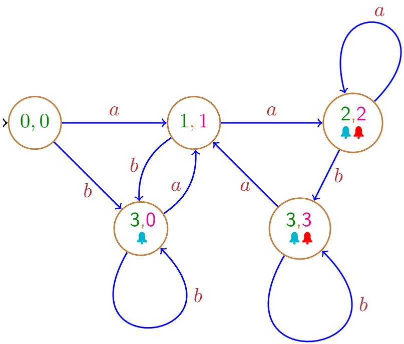
$$\text{Mark}_1 = \mathbf{Q} \bigcap \Big(\bigcup_{q \in \text{Mark}} q \times Q\Big) \qquad \text{Mark}_2 = \mathbf{Q} \bigcap \Big(\bigcup_{q \in \text{Mark}} Q \times q\Big)$$

Getting the Multivariate generating Function



$$U = aa + b, \qquad V = b^{\star}aab^{\star}$$

Getting the Multivariate generating Function



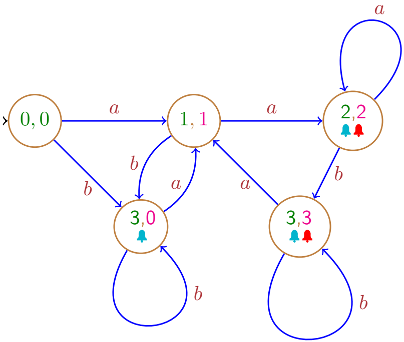
$$U = aa + b, \qquad V = b^\star aab^\star$$

Chomsky-Schützenberger again

$$\begin{aligned}
L_{00} &= \pi_a z L_{11} &&+ \pi_b z u L_{30} &&+ 1\\
L_{11} &= \pi_a z u v L_{22} &&+ \pi_b z u L_{30} &&+ 1\\
L_{30} &= \pi_a z L_{11} &&+ \pi_b z u L_{30} &&+ 1\\
L_{22} &= \pi_a z u v L_{22} &&+ \pi_b z u v L_{33} &&+ 1\\
L_{33} &= \pi_a z L_{11} &&+ \pi_b z u v L_{33} &&+ 1
\end{aligned}$$

$$\mathbf{P}(a) = \pi_a \quad \mathbf{P}(b) = \pi_b$$

Getting the Multivariate generating Function



$U = aa + b, \qquad V = b^\star aab^\star$

Chomsky-Schützenberger again

$$
\begin{aligned}
L_{00} &= \pi_a z L_{11} + \pi_b z u L_{30} + 1\\
L_{11} &= \pi_a z u v L_{22} + \pi_b z u L_{30} + 1\\
L_{30} &= \pi_a z L_{11} + \pi_b z u L_{30} + 1\\
L_{22} &= \pi_a z u v L_{22} + \pi_b z u v L_{33} + 1\\
L_{33} &= \pi_a z L_{11} + \pi_b z u v L_{33} + 1
\end{aligned}
$$

$\mathbf{P}(a) = \pi_a \quad \mathbf{P}(b) = \pi_b$

Assume $\pi_a = \pi_b = \frac{1}{2}$ $\begin{cases} U_n \text{ number of occurrences of } U \text{ in texts of length } n \\ V_n \text{ number of occurrences of } V \text{ in texts of length } n \end{cases}$

$$
\begin{aligned}
F(z,u,v) &= \sum_{n\geq 0} z^n \sum_{\substack{u\geq 0\\ v\geq 0}} \mathbf{P}(U_n = r, V_n = s) u^r v^s \\
&= \frac{8 + 4z - 8uvz - 2uv(1-uv)z^2}{8 - 4uz - 8uvz - 2u(1 - 2uv - uv^2)z^2 - u^2v^2(1+u)z^3}
\end{aligned}
$$

Covariance of U_n and V_n

$$F(z,u,v) = \sum_{n\geq 0} z^n \sum_{\substack{u\geq 0\\ v\geq 0}} \mathbf{P}(U_n = r, V_n = s) u^r v^s$$

$$= \frac{8+4z-8uvz-2uv(1-uv)z^2}{8-4uz-8uvz-2u(1-2uv-uv^2)z^2-u^2v^2(1+u)z^3}$$

Covariance of U_n and V_n

$$F(z,u,v) = \sum_{n\geq 0} z^n \sum_{\substack{u\geq 0\\ v\geq 0}} \mathbf{P}(U_n = r, V_n = s) u^r v^s$$

$$= \frac{8+4z-8uvz-2uv(1-uv)z^2}{8-4uz-8uvz-2u(1-2uv-uv^2)z^2-u^2v^2(1+u)z^3}$$

By differentiation:

$$\sum_{n\geq 0} \mathbf{E}(U_n V_n) z^n = \frac{\partial}{\partial u}\frac{\partial}{\partial v} F(z,u,v)\bigg|_{\substack{u=1\\ v=1}} = \frac{z^2}{8} \times \frac{8+8z-14z^2+5z^3-z^4}{(1-z)^3(2-z)^2}$$

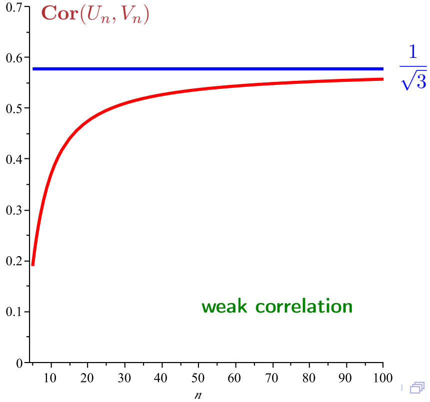
$$\mathbf{E}(U_n V_n) = \frac{3}{8}n^2 - \frac{3n+1}{4} + 2^{-n}n \qquad \begin{cases} \mathbf{E}(U_n) = \dfrac{3n-1}{4} \\[2ex] \mathbf{E}(V_n) = \dfrac{n-2}{2} + 2^{-n} \end{cases}$$

$$\mathbf{Cov}(U_n, V_n) = \mathbf{E}(U_n V_n) - \mathbf{E}(U_n)\mathbf{E}(V_n) = \frac{n-4}{8} + 2^{-n}\frac{n+1}{4}$$

Correlation of $U_n = aa + b$ and $V_n = b^\star aab^\star$

$$\mathbf{Cor}(U_n, V_n) = \frac{\mathbf{Cov}(U_n, V_n)}{\sigma_{U_n}\sigma_{V_n}} = \frac{\mathbf{E}(U_n V_n) - \mathbf{E}(U_n)\mathbf{E}(V_n)}{\sigma_{U_n}\sigma_{V_n}}$$

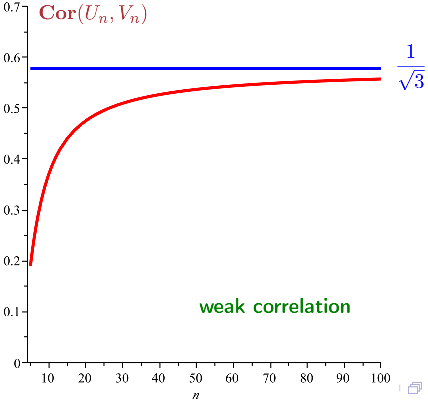
$$= \frac{n - 4 + 2^{-(n-1)}(n+1)}{\sqrt{(n+1)(3n - 6 - 2^{-n}(4n-12) - 4^{-(n-1)})}}$$



Correlation of $U_n = aa + b$ and $V_n = b^\star aab^\star$

$$\mathbf{Cor}(U_n, V_n) = \frac{\mathbf{Cov}(U_n, V_n)}{\sigma_{U_n}\sigma_{V_n}} = \frac{\mathbf{E}(U_n V_n) - \mathbf{E}(U_n)\mathbf{E}(V_n)}{\sigma_{U_n}\sigma_{V_n}}$$

$$= \frac{n - 4 + 2^{-(n-1)}(n+1)}{\sqrt{(n+1)(3n - 6 - 2^{-n}(4n-12) - 4^{-(n-1)})}}$$



Remark:
$2^{100} \approx 1.27 \times 10^{30}$

More on Marked-Automata

1. The **Marked-States** have the **same properties** as the **Accepting-States**, with respect to
   - **determinization** of NFAs
   - **minimization** of DFAs

More on Marked-Automata

1. The **Marked-States** have the **same properties** as the **Accepting-States**, with respect to
   - **determinization** of NFAs
   - **minimization** of DFAs

2. It is possible to make the **product of any finite number of automata**; this is not limited to the product of two automata. The automata need only be complete.

Reg-Exp to NFA by Glushkov (1961) or Berry-Sethi (1986) algorithm

$$R = (a+b)^* aba$$

1. **Index** the occurrences of letters $R' = (a_1 + b_1)^* a_2 b_2 a_3$
2. Use the **constructors** $\mathrm{first}, \mathrm{last}, \mathrm{follow}$
   $\mathrm{first}(R') = \{a_1, b_1, a_2\}$
   $\mathrm{last}(R') = \{a_3\}$
   $\mathrm{follow}(R', b_1) = \{a_1, b_1, a_2\}$
3. **Automaton**
   - **indexed letters** $\rightarrow$ **states**
   - **suppression of the indices** $\rightarrow$ **transitions**

   $\delta(b_1, a) = \{a_1, a_2\}, \qquad \delta(b_1, b) = \{b_1\}, \quad$ *etc.*

Glushkov and Berry-Sethy algorithm

Recursive definition of $\mathrm{first}, \mathrm{last}, \mathrm{follow}$ and $\mathrm{nullable}$

$\mathrm{nullable}(R) = \texttt{true}$ $\quad$ $\texttt{if}$ $\epsilon \in$ language of R

$$\mathrm{first}(R_1 R_2) = \begin{cases} \mathrm{first}(R_1) \cup \mathrm{first}(R_2) & \texttt{if} \quad \mathrm{nullable}(R_1), \\ \mathrm{first}(R_1) & \text{otherwise} \end{cases}$$

$$\mathrm{follow}(R_1 R_2, x) = \begin{cases} \mathrm{follow}(R_2, x) & \texttt{if} \quad x \in R_2, \\ \mathrm{follow}(R_1, x) \cup \mathrm{first}(R_2) & \texttt{if} \quad x \in \mathrm{last}(R_1) \\ \mathrm{follow}(R_1, x) & \texttt{otherwise} \end{cases}$$

$$\mathrm{follow}(R^*, x) = \begin{cases} \mathrm{follow}(R, x) \cup \mathrm{first}(R) & \texttt{if} \quad x \in \mathrm{last}(R), \\ \mathrm{follow}(R, x) & \texttt{otherwise} \end{cases}$$

Technical Condition $\Rightarrow$ quadratic complexity

Fast exact extraction of Taylor coefficients

$$F(z,u) = \frac{P(z,u)}{Q(z,u)} \Longrightarrow E(z) = \frac{U(z)}{V(z)}, \quad M_2(z) = \frac{H(z)}{K(z)}$$

$$\mathbf{E}(X_n) = [z^n]E(z), \quad \mathbf{E}(X_n^2) = [z^n]M_2(z)$$

Aim: **fast extraction** of the **nth Taylor coefficient** of a **rational function**

Method

$$E(z) = \frac{\sum_{0\le i\le j} u_i z^i}{\sum_{0\le i\le k} v_i z^i} = \sum_{n\ge 0} e_n z^n \Longrightarrow \sum_{0\le i\le k} v_i z^i \sum_{n\ge 0} e_n z^n = \sum_{0\le i\le j} u_i z^i$$

$$\Longrightarrow e_m v_0 + e_{m-1} v_1 + \cdots + e_{m-k} v_k = 0 \qquad (m > j)$$

$$\begin{cases} E_m = (e_m, e_{m-1}, \ldots, e_{m-k}) \\ E_{m+1}^t = \mathbb{A} \times E_m^t \end{cases} \quad \text{with } \mathbb{A} = \begin{pmatrix} -v_1/v_0 & -v_2/v_0 & \ldots & -v_k/v_0 \\ 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \ldots & & & \end{pmatrix} \quad \text{square matrix}$$

$$E_m^t = \mathbb{A}^{m-k} E_k^t$$

binary exponentiation to compute $\mathbb{A}^{m-k}$: $\mathbb{A}^4 = (\mathbb{A}^2)^2$, $\mathbb{A}^8 = (\mathbb{A}^4)^2, \ldots$

Example - $R = aba$, $\mathbf{P}(a) = \mathbf{P}(b) = 0.5$ - $\mathbf{E}(400000)$?

$$\sum_{n\geq 0} \mathbf{E}(X_n) z^n = \frac{z^3/2}{4 - 8z + 5z^2 - 2z^3 + z^4}$$

$$e_n = 2e_{n-1} - \frac{5}{4}e_{n-2} + \frac{1}{2}e_{n-3} - \frac{1}{4}e_{n-4}$$

$$E^t_{400000} = \begin{pmatrix} 2 & -5/4 & 1/2 & -1/4 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}^{399997} \begin{pmatrix} 1/8 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

399997
$= 1100001101001111101$
(base 2) (19 bits)

19 matrix products, 11 matrix by vector products (number of bits equal to 1)

$$E(X_{400000}) = \frac{399998}{8} \ (0.001\text{sec}), \quad E(X_{4000000}) = \frac{3999998}{8} \ (0.002\text{sec})$$

Complexity $O(\log n)$ **number of operations** for the computation of the **nth coefficient**

$\log(4000000)/\log(400000) \approx 1.179$ beware of **bit complexity**

Automatic computations - Lib. regexpcount (N.-Salvy)

```
> with(regexpcount):
> GRAM:={a=Atom,b=Atom,R=Prod(a,Sequence(b),a)};
```

$$GRAM := \{R = Prod(a, Sequence(b), a), a = Atom, b = Atom\}$$

```
> autoR:=regexptomatchesgram(GRAM,S,[[R,u,'overlap']]);
```

$$autoR := \{S = Union(\mathrm{E}, Prod(a, w3), Prod(b, S)), a = Atom, b = Atom, u = \mathrm{E}, w2 = Union(\mathrm{E}, Prod(a, u, w2), Prod(b, w3)), w3 = Union(\mathrm{E}, Prod(a, u, w2), Prod(b, w3))\}$$

```
> EQS:={seq(eval(subs(Prod=`*`,Union=`+`,Epsilon=1,Atom=var,i)),i=
  autoR)};
```

$$EQS := \{S = 1 + a\,w3 + b\,S, a = var, b = var, u = 1, w2 = 1 + a\,u\,w2 + b\,w3, w3 = 1 + a\,u\,w2 + b\,w3\}$$

```
> for i in {u,p} do EQS:=EQS minus {i=1} end do:for i in {a,b} do
  EQS:=EQS minus {i=var} end do:EQS;
```

$$\{S = 1 + a\,w3 + b\,S, w2 = 1 + a\,u\,w2 + b\,w3, w3 = 1 + a\,u\,w2 + b\,w3\}$$

```
> VAR:={seq(op(1,i),i=EQS)};
```

$$VAR := \{S, w2, w3\}$$

```
> SOLabu:=subs(solve(EQS,VAR),S);
```

$$SOLabu := -\frac{-a - 1 + b + a\,u}{a\,u\,b + 1 - 2\,b + b^2 - a\,u}$$

```
> SOLzu:=subs(a=z/2,b=z/2,SOLabu);
```

$$SOLzu := -\frac{-1 + \frac{1}{2}\,z\,u}{\frac{1}{4}\,z^2\,u + 1 - z + \frac{1}{4}\,z^2 - \frac{1}{2}\,z\,u}$$

```
> E(z):=subs(u=1,diff(SOLzu,u));
```

$$E(z) := -\frac{1}{2}\,\frac{z}{\frac{1}{2}\,z^2 + 1 - \frac{3}{2}\,z} + \frac{\left(-1 + \frac{1}{2}\,z\right)\left(\frac{1}{4}\,z^2 - \frac{1}{2}\,z\right)}{\left(\frac{1}{2}\,z^2 + 1 - \frac{3}{2}\,z\right)^2}$$

Automatic computations - Lib. gfun (Salvy-Zimmerman)

```
> E(z):=subs(u=1,diff(SOLzu,u));
```

$$E(z) := -\frac{1}{2}\frac{z}{\frac{1}{2}z^2+1-\frac{3}{2}z}+\frac{\left(-1+\frac{1}{2}z\right)\left(\frac{1}{4}z^2-\frac{1}{2}z\right)}{\left(\frac{1}{2}z^2+1-\frac{3}{2}z\right)^2}$$

```
> with(gfun):
> rec:=diffeqtorec(E(z)-y(z),y(z),u(n));
```

$$rec := \left\{-2\,u(n)+4\,u(1+n)-n,\, u(0)=0,\, u(1)=0,\, u(2)=\frac{1}{4}\right\}$$

```
> PROC:=rectoproc(rec,u(n)):
> time();
```

28.099

```
> t:=time():evalf(PROC(4000));time()-t;
```

1999.000000

0.017

```
> t:=time():evalf(PROC(40000));time()-t;
```

19999.00000

0.050

```
> evalf(log(40000)/log(4000));
```

1.277618919

```
> BITS_NUMER_4000:=evalf(log(numer(PROC(4000)))/log(10));
```

$BITS\_NUMER\_4000 := 1207.420796$

```
> BITS_NUMER_40000:=evalf(log(numer(PROC(40000)))/log(10));
```

$BITS\_NUMER\_40000 := 12045.50083$

Automatic computations - Lib. gfun (Salvy-Zimmerman)

```
> rec:=diffeqtorec(E(z)-y(z),y(z),u(n));
> PROC:=rectoproc(rec,u(n)):
> PP[4000]:=PROC(4000);
PP_4000 :=
2635089982768455257107675698678946135004854303025452265416514097033272099297556075760842449422107194424874057220495292925385518857078692198218039384257185
930508042143293910237016123149012158661431365192955747794631755345954422489188643985605602952832759661189294017486176163962476680465148128629219226693451 5
7616564579653326533152116751987169088120903711326493141941024724576377193712598746393325621172177608837740203359481599889055366167377732651823103641252738
3414595947305921112047909415647520431952859914117438164391026906958299508209849625986824965595364911397136016960799660468743294709168796848751485808242686
7241375223153740399360830333474596148120365212952448142610805840455064559134667030298312790704687079353135482454175849720688564093174157630595504000635446
6721761910115113310158034393947696250371314763046987742511356760633590078090299313664684231393397949869929645078593616544758290908108701063787860954672055
0588031354659109364559093701888447103900767515491188517270411454003385284483931369862245370562401767813732876381923948403361697372455955466938975834412295
69969818738803692554948517286086011692293139371521278025991623604578896409740965332055287304082286646838114987802625 /
1318204093430943100103889794236591363184019161093272769092803450241756928112834455107975212317212203314094075648071682303844681769424058128173106245251218
3854467444438688895632897064277199393003658655292424951448883218338941583237562000928492260894611103857875407791326544091858312558605043164728460363649082
5000782681167246890021068910448808948534719215270882011976500612594485839776187466930127874523350479658699451405443521705380373270324028340081592616934836
9947271609457689400724316866256888660306583248683060612501764335646973240725287456721773369482423667532334175568183922195469382045607202025388437122682684
5863619421287513956658744539006801474797581397174811477043924882668866712923795412855584187446066572963049265860017933827257911002088122876736120060347897
2016889399757435372765399896922309279825570166606797269890623692162876477283791552608646438916157053461695670374484050297527909408758729896842351653162609
9838935144902005685122107904896671887894330923207197857563987720862123704094012691276761065814107937875804340361142545474418057715085520493716346090251273
5126053963922145700597724726667634401815564750951539671135148754606247944459277905555421362722504575706910949376
>
> evalf(log10(numer(PP[4000])));
                                        1207.420796
> evalf(log10(denom(PP[4000])));
                                        1204.119983
```

An application to biology - Protein Motifs Statistics

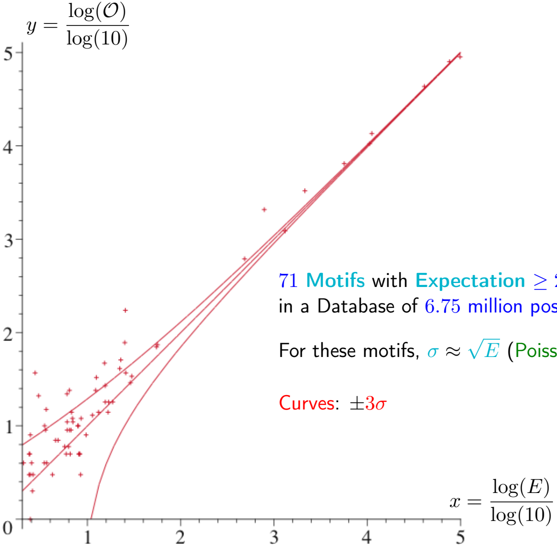
Motif PS00844 (1998): DALA_DALA_LIGASE_2

[LIV]-x(3)-[GA]-x-[GSAIV]-R-[LIVCA]-D-[LIVMF](2)-x(7,9)-[LI]-x-E-[LIVA]-N-[STP]-x-P-[GA]

- $\mathcal{A}$: alphabet of the proteins (20 letters)
- [LIV] $= L + I + V$
- [LIVMF](2) $= (L + I + V + M + F)^2$
- x $= \mathcal{A}$
- x(3) $=$ x^3
- x(7,9) $=$ x^7 $+$ x^8 $+$ x^9

The **automaton** recognizing $\mathcal{A}^\star$.PS00844 and **counting the matches of the motif** in a **random non-uniform Bernoulli** text has **946 states** while the **number of words** of the **finite language** generated by the **motif** is about 2×10^{26}

Comparison of Observed and Predicted Counts



71 **Motifs** with **Expectation** ≥ 2
in a Database of 6.75 million positions

For these motifs, $\sigma \approx \sqrt{E}$ (Poisson-like)

Curves: $\pm 3\sigma$

From [N.,Salvy,Flajolet] - Motif Statistics, TCS2002

Open problems

- **Definition** of a **random model** of **NFA**

- **Limit distribution** of the **number of occurrences** of **two regular expressions** (use Heuberger's theorem)
- **Generalization** of **Hwang's Large Powers theorem** to **dimensions larger than two**

- **Limit distribution** when the **number of occurrences** is $O(1)$ (one **regular expression**) - **Conjecture:** Poisson

Short Bibliography

- Kelley, D. *Automata and Formal Languages, an Introduction*. Prentice Hall, 1995
- Kozen, D. C. *Automata and Computability*, Springer Verlag, 1997
- Nicodème,P. , Salvy, B., Flajolet, F. *Motif Statistics*, TCS 2002
- Nicodème, P. *Regexpcount, a symbolic package for counting problems on regular expressions and words*, Fundamentae Informaticae, 2003
- Nicaud, C., Pivoteau, C., Razet, B. *Average Analysis of Glushkov Automata under a BST-Like Model*, FSTTCS'10, 2010
- Nuel, G., Dumas, J.-G. *Sparse approaches for the exact distribution of patterns in long state sequences generated by a Markov source*, TCS 2012