2.16: Finite automata modelling

Daniela Petrisan
Sam van Gool
Amaury Pouly
Matthieu Picantin

2021 – 2022

Contents

Part I

Weighted automata and transducers (Daniela Petrisan)

1 Weighted automata, rational and recognizable series

1.1 Weighted automata

Definition 1.1 Semiring

A semiring is a tuple $(K, +, \cdot, 0, 1)$ such that

- $(K, +, 0)$ is a commutative monoid
- $(K, \cdot, 1)$ is a monoid
- $\forall x, y, z \in K, x \cdot (y + z) = x \cdot y + x \cdot z$
- $\forall x, y, z \in K, (y + z) \cdot x = y \cdot x + y \cdot z$
- $\forall x \in K, x \cdot 0 = 0 \cdot x = 0$

Example 1.2

- The numerical semiring: $(\mathbb{N}, +, \cdot, 0, 1)$ (or with $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$)
- The boolean semiring: $\mathbb{B} = \{\{0, 1\}, \wedge, \vee, 1, 0\}$
- Tropicals semirings:
  - $\mathbb{N}_{\min} = (\mathbb{N} \cup \{\infty\}, \min, +, \infty, 0)$
  - $\mathbb{N}_{\max} = (\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$

Result 1.3

If $(M, \cdot, 1)$ is a monoid, then $(\mathcal{P}(M), \cup, \cdot, \emptyset, \{1\})$ is a semiring, where $\cdot$ is the complex multiplication $\forall A, B \subseteq M, A \cdot B = \{a \cdot b \,|\, a \in A, b \in B\}$

Example 1.4

With the monoid $(A^*, \cdot, \varepsilon)$ we have the semiring $(\mathcal{P}(A^*), \cup, \cdot, \emptyset, \{\varepsilon\})$.
$(\mathrm{Rat}(A^*), \cup, \cdot, \emptyset, \{\varepsilon\})$ is a subsemiring of that semiring.

Result 1.5

If $(K, +, \cdot, 0, 1)$ is a semiring, then the set of matrices of size n $K^{n\times n}$ can be equipped with a semiring structure $(K^{n\times n}, +, \cdot, 0, I)$ with 0 the null matrix and I the identity

Definition 1.6 **Morphism of semiring**

A morphism between semirings $(K, +, \cdot, 0, 1)$ and $(K', +', \cdot', 0', 1')$ is a function $f : K \to K'$ such that

- $\forall x, y \in K, f(x + y) = f(x) +' f(y)$ and $f(x \cdot y) = f(x) \cdot' f(y)$
- $f(0) = 0'$ and $f(1) = 1'$

Definition 1.7 **Weighted automata**

A weighted automaton over a semiring $(K, +, \cdot, 0, 1)$ is a tuple $\mathcal{A} = (A, Q, I, \delta, F)$ where

- A is a finite set (the alphabet)
- Q is a finite set of states
- $I : Q \to K$ is the initial values function
- $\delta :\subseteq Q \times A \times Q \to K$ is the weighted transition function
- $F : Q \to K$ is the final (or accepting) states function

Remark 1.8

If K is the boolean semiring $\mathbb{B}$ then the weighted automaton is just a non-deterministic finite automaton.

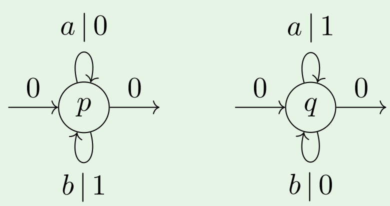
Notation 1.9

For $q \in Q$, if $I(q) = k$ we draw $\xrightarrow{k} \bigcirc\!\!\!\!\!q\,$ and if $F(q) = k$ we draw $\bigcirc\!\!\!\!\!q\, \xrightarrow{k}$.

For $p, q \in Q$, if $\delta(p, a, q) = k$ we draw $p \xrightarrow{a \,|\, k} q$ or $p \xrightarrow{k,\, a} q$.

Example 1.10

With $\mathbb{N}_{\max}$-automata we omit the 1.



Definition 1.11 **Path**

A path in a weighted automaton is a sequence of the form

$$p : p_0 \xrightarrow{a_1 \,|\, k_1} p_1 \xrightarrow{a_2 \,|\, k_2} \ldots \xrightarrow{a_n \,|\, k_n} p_n$$

where $p_i \in Q$, $a_i \in A$ and $k_i \in K$.
The labels of p is $\ell(p) = a_1 a_2 \ldots a_n$.
The weight of p is $w(p) = k_1 \cdot k_2 \cdot \ldots \cdot k_n$.
The weighted label of p is $w\ell(p) = w(p)\ell(p)$.

Definition 1.12 Computation

A computation in a weighted automaton $\mathcal{A}$ is a sequence

$$C : \xrightarrow{k_0} p_0 \xrightarrow{a_1 \,|\, k_1} p_1 \xrightarrow{a_2 \,|\, k_2} \ldots \xrightarrow{a_1 \,|\, k_1} p_n \xrightarrow{k_{n+1}}$$

where $p_0 \xrightarrow{a_1 \,|\, k_1} p_1 \xrightarrow{a_2 \,|\, k_2} \ldots \xrightarrow{a_1 \,|\, k_1} p_n$ is a path and $I(p_0) = k_0$ and $F(p_n) = k_{n+1}$.
The labels of C is $\ell(C) = a_1 \ldots a_n$.
The weight of C is $w(C) = k_0 \cdot \ldots \cdot k_{n+1}$.

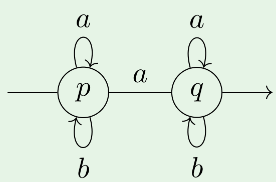
Definition 1.13 Behaviour of automata

The behaviour of a K-weighted $\mathcal{A}$ is a function $|\mathcal{A}| : A^* \to K$ defined on $w \in A^*$ by

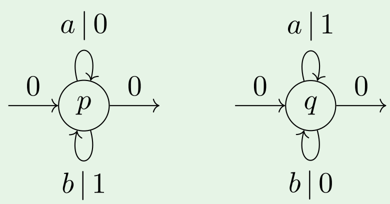
$$|\mathcal{A}|(w) = \sum_{\substack{C \text{ computation in } \mathcal{A} \\ \ell(C)=w}} w(C)$$

Example 1.14

Consider the following $\mathbb{N}$-automaton $\mathcal{A}_1$ (with only 1s on transitions)



then $|\mathcal{A}_1|(w) = |w|_a$ because all the paths are of weight 1.
Consider the following $\mathbb{N}_{\max}$-automaton $\mathcal{A}_2$



then $|\mathcal{A}_2|(w) = \max(|w|_a, |w|_b)$ because p counts the bs and q counts the as.

1.2 K-series

Definition 1.15 K-series

A K-series over A^* is a function $s : A^* \to K$.
The set of K-series is denoted by $K\langle\langle A^* \rangle\rangle$.

Definition 1.16 K-series operations

$K\langle\langle A^* \rangle\rangle$ can be equipped with the following operations.

- Pointwise addition: $\forall s, t \in K\langle\langle A^* \rangle\rangle, \forall w \in A^*, (s+t)(w) = s(w) + t(w)$
- Cauchy product: $\forall s, t \in K, \forall w \in A^*, (s \cdot t)(w) = \sum_{\substack{u,v \in A^* \\ uv=w}} s(u) \cdot_K t(v)$
- Scalar multiplications: $\forall s \in K\langle\langle A^* \rangle\rangle, \forall w \in A^*, \forall k \in K, (k \cdot s)(w) = k \cdot_K s(w)$ and $(s \cdot k)(w) =$

$s(w) \cdot_K k$

The Cauchy product and pointwise addition allows to define a semiring.

Proposition 1.17

- $\forall s, t, r \in K\langle\langle A^*\rangle\rangle$,
  - $(s + t) \cdot r = s \cdot r + t \cdot r$
  - $r \cdot (s + t) = r \cdot s + r \cdot t$
- $\forall k \in K, \forall s, t \in K\langle\langle A^*\rangle\rangle$,
  - $k \cdot (s + t) = k \cdot s + k \cdot t$
  - $(s + t) \cdot k = s \cdot k + t \cdot k$
  - $k \cdot (s \cdot t) = (k \cdot s) \cdot t$
  - $(s \cdot t) \cdot k = s \cdot (t \cdot k)$

$K\langle\langle A^*\rangle\rangle$ is a K-algebra.

Definition 1.18 **Support of a series**

The support of a K-series $s \in K\langle\langle A^*\rangle\rangle$ is defined as $\mathrm{supp}(s) = \{w \in A^* \mid s(w) \neq 0_K\}$.

Given a K-weighted automaton $\mathcal{A}$ we can obtain a Boolean automaton $\mathrm{supp}(\mathcal{A})$ by replacing all non zero weights in $\mathcal{A}$ with $1_{\mathbb{B}}$.

Exercise 1.19

Prove that $\mathrm{supp}|\mathcal{A}| \subseteq |\mathrm{supp}(\mathcal{A})|$.
That is that the support of the series $|\mathcal{A}| : A^* \to K$ is included in the language accepted by the Boolean automaton $\mathrm{supp}(\mathcal{A})$.

Proof.
Let $w \in \mathrm{supp}|\mathcal{A}|$.
Then $\displaystyle\sum_{\substack{C \text{ computation} \\ \ell(C)=w}} w(C) \neq 0$.
So $\exists C_0, w(C_0) \neq 0$, so C_0 is a path of $\mathrm{supp}(\mathcal{A})$ since the values of C_0 are non 0.

Exercise 1.20

Find a sufficient condition such that $\mathrm{supp}|\mathcal{A}| = |\mathrm{supp}(\mathcal{A})|$ and an example where the inclusion is strict.

Notation 1.21

For $s \in K\langle\langle A^*\rangle\rangle$ we write it $s = \displaystyle\sum_{w \in A^*} s(w) \cdot w$.
If $\mathrm{supp}(s)$ is finite we call s a polynomial over K. The set of polynomials is denoted by $K\langle A^*\rangle$.

Example 1.22

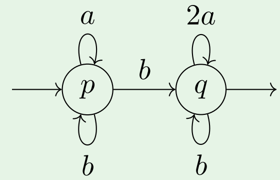
$2aa + 3abba \in \mathbb{Z}\langle A^*\rangle \subseteq \mathbb{Z}\langle\langle A^*\rangle\rangle$

For $\mathcal{A} = (A, Q, I : Q \to K, F : Q \to K, \delta : Q \times A \times Q \to K)$, δ generates a matrix of size $|Q|$.
$\delta^\# : Q \times Q \to K^A \subseteq \underbrace{K\langle\langle A^*\rangle\rangle}_{\text{semiring}}$.

$I : Q \to K$ represents row vectors of size $|Q|$.
$F : Q \to K$ represents column vectors of size $|Q|$.

Example 1.23

Consider the following $\mathbb{N}$-automaton $\mathcal{A}_3$.



It's matrix representation is the following.

$$\begin{array}{ccc} I & \Delta & F \\ \begin{pmatrix} 1 & 0 \end{pmatrix} & \begin{pmatrix} a+b & b \\ 0 & 2a+b \end{pmatrix} & \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{array}$$

Lemma 1.24

Consider the matrix representation (I, Δ, F) of a K-weighted automaton $\mathcal{A}$.
Then for $w \in A^*$ with $|w| = n$, we have

$$|\mathcal{A}|(w) = (\underbrace{I \cdot \Delta^n \cdot F}_{\in K\langle\langle A^* \rangle\rangle})(w).$$

Exercise 1.25

Check this in the previous example by computing $|\mathcal{A}_3|(ab)$ in two different ways.

Corollary 1.26

Consider the matrix representation (I, Δ, F) of a K-weighted automaton $\mathcal{A}$. Then

$$|\mathcal{A}| = \sum_{n \geqslant 0} I \cdot \Delta^n \cdot F = I \cdot \left(\underbrace{\sum_{n \geqslant 0} \Delta^n}_{=\Delta^*} \right) \cdot F.$$

The question is to know if Δ^* is well defined, if it exists. So the goal is to define the $*$ operation on $K\langle\langle A^* \rangle\rangle$.

1.3 Topological semiring

Definition 1.27 **Topological semiring**

K is called a topological semiring if it is equipped with a semiring structure $(K, +, \cdot, 0, 1)$ and a topological structure such that $+$ and $\cdot$ are both continuous.

In practice we will consider topologies generated by some distance.

Observation

A distance d on K induces a distance $\tilde{d}$ on $K\langle\langle A^*\rangle\rangle$.

$$\tilde{d}(s,t) = \frac{1}{2}\sum_{n\geqslant 0}\frac{1}{2^n}\max_{|w|=n}(d(s(w),t(w))).$$

Definition 1.28 Summable family

A summable family of a semiring K is a family $(k_i)_{i\in I}$ with $k_i \in K$ such that

$$\exists k \in K, \forall \varepsilon > 0, \exists I_\varepsilon \subseteq I \text{ finite}, \forall J \subseteq I \text{ finite}, \text{with } I_\varepsilon \subseteq J, d(\sum_{i\in J} k_i, k) < \varepsilon$$

Then $k = \sum_{i\in I} k_i$.

Definition 1.29 Locally finite family

A family $(s_i)_{i\in I} \subseteq K\langle\langle A^*\rangle\rangle$ is called locally finite if $\forall w \in A^*, \{i \in I \mid s_i(w) \neq 0_K\}$ is finite.

Proposition 1.30

A locally finite family $(s_i)_{i\in I} \subseteq K\langle\langle A^*\rangle\rangle$ is summable.

Definition 1.31 Proper series

A series $s \in K\langle\langle A^*\rangle\rangle$ is called proper when $s(\varepsilon) = 0_K$.

Proposition 1.32

Consder $s \in K\langle\langle A^*\rangle\rangle$ a proper series.
Then $(s^n)_{n\in\mathbb{N}}$ is summable family.

Proof.
$s^2(w) = \sum_{\substack{u,v\in A^* \\ uv=w}} s(u)\cdot s(v)$
So for $w = a \in A$, $s^2(a) = s(\varepsilon)s(a) + s(\varepsilon)s(a) = 0 + 0 = 0$.
By induction we can prove that $s^n(w) = 0$ for all words $w \in A^*$ such that $|w| < n$.
We conclude that $(s^n)_{n\in\mathbb{N}}$ is locally finite, hence summable.

We denote $s^* = \sum_{n\geqslant 0} s^n$ when the sum on the right is well defined.

We obtain a partial operation $(\_)^* : K\langle\langle A^*\rangle\rangle \to K\langle\langle A^*\rangle\rangle$. Let's consider $w \in A^*$. We need to show that $s^n(w) = 0$ for all w for all but finitely many ns.

Claim $s^n(w) = 0$ for all $n > |w|$.

Lemma 1.33

Let K be a topological semiring and $k \in K$.
If k^* exists then $k^* = k\cdot k^* + 1$ and $k^* = k^*\cdot k + 1$.

Proof.
$$\sum_{n\geqslant 0} k^n = k\sum_{n\geqslant 0} k^n + \underbrace{1}_{=k^0}$$

Lemma 1.34 **Arden's lemma**

Let $s, t \in K\langle\langle A^*\rangle\rangle$ with s and t proper.
Then the equation $X = sX + t$ has a unique solution in $K\langle\langle A^*\rangle\rangle$, namely the series s^*t.
Similarly, the equation $X = Xs + t$ has a unique solution ts^*.

Definition 1.35 **Rational closed sets**

A subset of $K\langle\langle A^*\rangle\rangle$ is called rationnaly closed when it is closed under

- pointwise addition
- Cauchy product
- left and right scalar multiplication
- the $(\_)^*$ operation when defined

Remark that the intersection of rationally closed sets is rationally closed.

Definition 1.36 **Rational closure**

For $X \subseteq K\langle\langle A^*\rangle\rangle$, we define its rational closure as the intersection of all the rationally closed subsets of $K\langle\langle A^*\rangle\rangle$ that contain X.

Definition 1.37 **Rational series**

The set of rational series over A^*, denoted $\mathrm{Rat}\langle A^*\rangle$, is defined as the rational closure of $K\langle A^*\rangle$ (the set of polynomials).

Theorem 1.38

If K is a strong semiring, then the rational series are the behaviours of K-weighted automata $\mathrm{Rat}_K\langle A^*\rangle$.

Exercise 1.39

Let $s't \in K\langle\langle A^*\rangle\rangle$ be a proper series.
Prove that $(s+t)^* = s^*(ts^*)^*$.

Proof.
$(s+t)^*$ is a solution to the equation $X = (s+t)X + 1$.

$$\begin{aligned}(s+t)s^*(ts^*)^* + 1 &= ss^*(ts^*)^* + ts^*(ts^*)^* + 1\\ &= ss^*(ts^*)^* + (ts^*)^*\\ &= (ss^* + 1)(ts^*)^*\\ &= s^*(ts^*)^*\end{aligned}$$

So by unicity of the solution of the equation, $(s+t)^* = s^*(ts^*)^*$.

2 A unifying algebraic framework for minimization

2.1 Categories

$$\begin{tikzcd} \mathbf{1} \arrow[r] & Q \arrow[r] \arrow[loop above, "\delta_a"] & \mathbf{2} \end{tikzcd}$$

- Q is the set of states
- $\mathbf{1}$ is a singleton set ($\{*\}$)
- $\mathbf{2}$ is a two elements set ($\{0, 1\}$)

Giving an initial state $q_0 \in Q$ is equivalent to give a function $\triangleright : \begin{cases} \mathbf{1} & \to & Q \\ & \mapsto & q_0 \end{cases}$.

Giving a set $F \subseteq Q$ of final states in Q is equivalent to giving a function $\triangleleft : \begin{cases} Q & \to & \mathbf{2} \\ q & \mapsto & \begin{matrix} 0, \text{ if } q \notin F \\ 1, \text{ if } q \in F \end{matrix} \end{cases}$.

$$\begin{tikzcd} \mathbf{1} \arrow[r, "/" marking] & Q \arrow[r, "/" marking] \arrow[loop above, "\delta_a"] & \mathbf{1} \end{tikzcd}$$

If X and Y are sets, we denote a relation $R \subseteq X \times Y$ by a negated arrow $R : X \nrightarrow Y$.

Giving a subset $I \subseteq Q$ of initial states is equivalent to giving a relation $\triangleright : \mathbf{1} \nrightarrow Q$. $\triangleright \subseteq \mathbf{1} \times Q$, $\triangleright = \{(*, q) \mid q \in I\}$.

Similarly, a subset $F \subseteq F$ of finite states can be described as a relation $\triangleleft : Q \nrightarrow \mathbf{1}$.

Definition 2.1 **DFA, NFA**

A DFA is a tuple $(Q, \triangleright : \mathbf{1} \to Q, \triangleleft : Q \to \mathbf{2}, (\delta_a : Q \to Q)_{a \in A})$.
A NFA is a tuple $(Q, \triangleright : \mathbf{1} \nrightarrow Q, \triangleleft : Q \nrightarrow \mathbf{1}, (\delta_a : Q \nrightarrow Q)_{a \in A})$.

$$\begin{tikzcd} K \arrow[r, "\triangleright"] & Q \arrow[r, "\triangleleft"] \arrow[loop above, "\delta_a"] & K \end{tikzcd}$$

Let K be a field and A a finite alphabet. Let Q be a K-vector space.

Previously we had a set of states Q_0 and an initial map $I : Q_0 \to K$.

Let Q be the vector space with basis Q_0: K^{Q_0}. The initial map $I : Q_0 \to K$ can be seen as a vector in Q.

Giving a vector $v_I \in Q$ is equivalent to giving a linear transformation $\triangleright : \begin{cases} K & \to & Q \\ 1_K & \mapsto & v_I \end{cases}$.

For a weighted automaton witf a finite set of states Q_0, we also have a final map $F : Q_0 \to K$. Such a final map corresponds to a linear transformation $\triangleleft : Q = K^{Q_0} \to K$.

The weighted transitions $\delta_a \subseteq Q_0 \times K \times Q_0$ can be encoded as linear transformations $\delta_a : K^{Q_0} \to K^{Q_0}$.

Summary

$\begin{tikzcd} \mathbf{1} \arrow[r] & Q \arrow[r] \arrow[loop above, "\delta_a"] & \mathbf{2} \end{tikzcd}$	DFA	$Q, \mathbf{1}, \mathbf{2}$ are sets. $\to$ are functions.	$\mathbf{Set}$
$\begin{tikzcd} \mathbf{1} \arrow[r, "/" marking] & Q \arrow[r, "/" marking] \arrow[loop above, "\delta_a"] & \mathbf{1} \end{tikzcd}$	NFA	$Q, \mathbf{1}$ are sets. $\nrightarrow$ are relations.	$\mathbf{Rel}$
$\begin{tikzcd} K \arrow[r, "\triangleright"] & Q \arrow[r, "\triangleleft"] \arrow[loop above, "\delta_a"] & K \end{tikzcd}$	Weighted automata	K, Q are K-vector spaces. The arrows are K-linear transformations.	K-$\mathbf{Vec}$

Definition 2.2 **Category**

A category $\mathcal{C}$ consists of the following data:

- a class of objects $\mathrm{Ob}(\mathcal{C})$
- for each object $X \in \mathrm{Ob}(\mathcal{C})$, the identity morphism
- for each pair $X, Y \in \mathrm{Ob}(\mathcal{C})$, a set of morphisms $\mathcal{C}(X,Y)$. For $f \in \mathcal{C}(X,Y)$ we write $X \underset{f}{\to} Y$.
- for each triple $X, Y, Z \in \mathrm{Ob}(\mathcal{C})$, a composition relation $\circ : \mathcal{C}(Y,Z) \times \mathcal{C}(X,Y) \to \mathcal{C}(X,Z)$, that is associative.

For the DFA, the objects are sets, the identities are the identity functions and the morphisms are the functions.

For the NFA, the objects are sets, the identities are the diagonal relations and the morphisms are the relations.

For the weighted automata, the objects are K-vector spaces, the identities are the identity linear transformations and the morphisms are the linear transformations.

Example 2.3

Let $(X, \leqslant)$ be a preordered set. We can see X as a category $\mathcal{C}_{(X,\leqslant)}$ as follows:

- the objects are the elements of X
- for $x, y \in X, \mathcal{C}_{(X,\leqslant)}(x,y) = \begin{cases} \emptyset & \text{if } x \nleqslant y \\ \{*_{(x,y)}\} & \text{if } x \leqslant y \end{cases}$

Example 2.4

Let A be a finite set.
Consider a category $\mathcal{I}$ with three objects $\{\mathsf{in}, \mathsf{st}, \mathsf{out}\}$ and whose morphisms are generated via composition from the following ones:

- $\triangleright : \mathsf{in} \to \mathsf{st}$
- $\triangleleft : \mathsf{st} \to \mathsf{out}$
- $a : \mathsf{st} \to \mathsf{st}$ for $a \in A$

$$\begin{tikzcd} \mathsf{in} \arrow[r, "\triangleright"'] & \mathsf{st} \arrow[loop above, "a"] \arrow[r, "\triangleleft"'] & \mathsf{out} \end{tikzcd}$$

$\mathcal{I}(\mathsf{in}, \mathsf{in}) = \{\mathrm{id}_{\mathsf{in}}\}$
$\mathcal{I}(\mathsf{in}, \mathsf{st}) = \{\triangleright w \mid w \in A^*\}$
$\mathcal{I}(\mathsf{st}, \mathsf{st}) = \{w \mid w \in A^*\}$
$\mathcal{I}(\mathsf{st}, \mathsf{out}) = \{w \triangleleft \mid w \in A^*\}$
$\mathcal{I}(\mathsf{in}, \mathsf{out}) = \{\triangleright w \triangleleft \mid w \in A^*\}$

Definition 2.5 **Functor**

Consider categories $\mathcal{C}_1$ and $\mathcal{C}_2$.
A functor $F : \mathcal{C}_1 \to \mathcal{C}_2$ consist of the following data:

- a function $F : \mathrm{Ob}(\mathcal{C}_1) \to \mathrm{Ob}(\mathcal{C}_2)$
- for each pair $X, Y \in \mathrm{Ob}(\mathcal{C}_1)$, a function $F_{X,Y} : \mathcal{C}_1(X,Y) \to \mathcal{C}_2(FX, FY)$

and that preserves identity and composition:

- $F_{X,X}(\mathrm{id}_X) = \mathrm{id}_{FX}$
- given $X \xrightarrow{f} Y \xrightarrow{g} Z$, $F_{Y,Z}(g) \circ F_{X,Y}(f) = F_{X,Z}(g \circ f)$.

Example 2.6

Consider the functor $G : \mathbf{Set} \to \mathbf{Rel}$ defined as follows:

- $\forall X \in \mathrm{Ob}(\mathbf{Set}), GX = X$
- $\forall f : X \to Y, G(f)$ is the graph of f, which is a relation $X \not\to Y$.

Exercise 2.7

Find a functor $\mathbf{Rel} \to \mathbf{Set}$.

Proof.
$F : X \to \mathcal{P}(X)$ and for $X \xrightarrow{R} Y, F(R)(X) = \bigcup x \in X\{y \,|\, (x, y) \in R\}$.
If $\mathrm{id}_X : X \not\to X$ is the identity relation, $F(\mathrm{id}_X)(X) = \bigcup_{x\in X}\{y \,|\, (x, y) \in \mathrm{id}_X\} = \bigcup_{x\in X}\{x\} = X$.

Definition 2.8 $(\mathcal{C}, I, F)$**-automata**

Let $\mathcal{C}$ be a category and $I, F \in \mathrm{Ob}(\mathcal{C})$.
A $(\mathcal{C}, I, F)$-automaton is a functor $\mathcal{A} : \mathcal{I} \to \mathcal{C}$ such that $\mathcal{A}(\text{in}) = I$ and $\mathcal{A}(\text{out}) = F$ and $\mathcal{I}$ is the three objects category of Example 2.4.

$$\begin{tikzcd} \text{in} \arrow[r, "\triangleright"'] & \text{st} \arrow[loop above, "a"] \arrow[r, "\triangleleft"'] & \text{out} \arrow[r, "\mathcal{A}"] & \mathcal{C} \end{tikzcd}$$

The functor $\mathcal{A}$ "interprets" the object st $\in \mathrm{Ob}(\mathcal{I})$ as an object of $\mathcal{C}$. This will be the object of states of the $(\mathcal{C}, I, F)$-automaton, $\mathcal{A}(\text{st})$.

For each input letter $a \in A$, the functor $\mathcal{A}$ interprets the morphism a : st $\to$ st as a morphism $\mathcal{A}(a) : \mathcal{A}(\text{st}) \to \mathcal{A}(\text{st})$ in the category $\mathcal{C}$. This is the a-transition of the $(\mathcal{C}, I, F)$-automaton.

$\mathcal{A}(\triangleright) : \mathcal{A}(\text{in}) \to \mathcal{A}(\text{st})$ is the initial state morphism. $\mathcal{A}(\triangleleft) : \mathcal{A}(\text{st}) \to \mathcal{A}(\text{out})$ is the final values morphism.

Example 2.9

1. $(\mathbf{Set}, \mathbf{1}, \mathbf{2})$-automata are the deterministic automata.
2. $(\mathbf{Rel}, \mathbf{1}, \mathbf{1})$-automata are the non-deterministic automata.
3. $(K\text{-}\mathbf{Vec}, K, K)$-automata are the weighted automata.

2.2 Minimization

2.2.1 Factorization system

A deterministic automaton A is a minimal when it *divides*, i.e. if it is a quotient of a sub-automaton, any other automaton accepting the same language.

If B is an arbitrary DFA, minimizing it constist in considering the sub-automaton of the states that are reachable from the initial state of B, and take the quotient of initial states.

$$\begin{tikzcd} \mathrm{Reach}(B) \arrow[r, "\text{inclusion}"] \arrow[dr, "\text{quotient}"'] & B \\ & \mathrm{Obs}(\mathrm{Reach}(B)) = \mathrm{Min}(B) \end{tikzcd}$$

Example 2.10

A morphism between two deterministic automata $(Q, q_0 \in Q, F \subseteq Q, (\delta_a : Q \to Q)_{a \in A})$ and $(Q', q'_0 \in Q', F' \subseteq Q', (\delta'_a : Q' \to Q')_{a \in A})$ is a function $f : Q \to Q'$ such that

- $f(q_0) = q'_0$
- $\delta'_a \circ f = f \circ \delta_a$
- $f(q) \in F' \Leftrightarrow q \in F$

f is called a quotient of automata when $f : Q \to Q'$ is surjective.

Definition 2.11 **Factorization system**

Let $\mathcal{C}$ be a category. A factorization system on $\mathcal{C}$ is a pair (E, M) where E and M are classes of morphisms such that

1. E and M contain all the isomorphisms, i.e. morphisms $f : X \to Y$ such that there exists $g : Y \to X$ with $g \circ f = \mathrm{id}_X$ and $f \circ g = \mathrm{id}_Y$
2. E and M are closed under composition and every function $f : X \to Y$ in $\mathcal{C}$ can be written as a composition:

$$\begin{tikzcd} X \arrow[r, two heads, "e"] \arrow[rr, bend right, "f"'] & \mathcal{C} \arrow[r, tail, "m"] & Y \end{tikzcd}$$

3. If $e : X \to Y$ is in E, $m : Z \to W$ is in M, and $f : X \to Z$ and $g : Y \to W$ are so that $g \circ e = m \circ f$, then there exists a unique morphism $d : Y \to Z$ such that $d \circ e = f$ and $m \circ d = g$:

$$\begin{tikzcd} X \arrow[r, two heads, "e"] \arrow[d, "f"'] & Y \arrow[d, "g"] \arrow[dl, dashed, "d"] \\ Z \arrow[r, tail, "m"'] & W \end{tikzcd}$$

Notation 2.12

We write a two-headed arrow for morphisms in E and a tail arrow for morphisms in M.

$$\xrightarrow{e}\!\!\!\!\to \ \in E \qquad\qquad \rightarrowtail^{m} \ \in M$$

Example 2.13

- In $\mathbf{Set}$, the pair (Surjective Functions, Injective Functions) is a factorization system.
- In $\mathbb{R}$-$\mathbf{Vec}$, the pair (Surjective Linear Functions, Injective Linear Functions) is a factorization system.

Exercise 2.14

Let $\mathcal{C}$ be a category with a factorization system (E, M). Consider two factorizations of a morphism $f : A \to B$

$$\begin{tikzcd}
 & \mathcal{C} \arrow[dr, tail, "m"] \arrow[dd, dashed, bend left, "\varphi"] & \\
A \arrow[ur, two heads, "e"] \arrow[rr, "f"] \arrow[dr, two heads, "e'"'] & & B \\
 & \mathcal{C}' \arrow[ur, tail, "m'"'] &
\end{tikzcd}$$

Prove that there exists an isomorphism $\varphi : \mathcal{C} \to \mathcal{C}'$ such that $g \circ e = e'$ and $m' \circ g = m$. That is factorizations are unique up to isomorphism.

Proof.
In the diagrams

$$\begin{tikzcd}
A \arrow[r, two heads, "e"] \arrow[d, "e'"'] & \mathcal{C} \arrow[d, "m"] \arrow[dl, dashed, "\varphi"] \\
\mathcal{C}' \arrow[r, tail, "m'"'] & B
\end{tikzcd}
\qquad
\begin{tikzcd}
A \arrow[r, two heads, "e'"] \arrow[d, "e"'] & \mathcal{C}' \arrow[d, "m'"] \arrow[dl, dashed, "\varphi'"] \\
\mathcal{C} \arrow[r, tail, "m"'] & B
\end{tikzcd}$$

we have the existence and uniquenesses of φ and φ', and we have that $\varphi \circ \varphi' = \mathrm{id}$ by uniqueness in the following diagram.

$$\begin{tikzcd}
A \arrow[r, two heads, "e"] \arrow[d, "e"'] & \mathcal{C} \arrow[d, "m"] \arrow[dl, bend right, "\varphi \circ \varphi'"'] \arrow[dl, dashed, "\mathrm{id}"] \\
\mathcal{C} \arrow[r, tail, "m"'] & B
\end{tikzcd}$$

Definition 2.15 **Initial object**

An object I of $\mathcal{C}$ is called initial if for every object X of $\mathcal{C}$ there is a unique morphism $I \to X$.

Example 2.16

In **Set**, $\emptyset$ is an initial object.

Exercise 2.17

Show that initial objects are unique up to isomorphism.

Definition 2.18 **Final object**

An object I of $\mathcal{C}$ is called final if for every object X of $\mathcal{C}$ there is a unique morphism $X \to I$.

Example 2.19

In **Set**, any singleton is a final set.

Definition 2.20 Natural transformation

If $\mathcal{C}$ and $\mathcal{D}$ are two categories and $F, G : \mathcal{C} \to \mathcal{D}$ are two functors, a natural transformation $\alpha : F \Rightarrow G$ is a family of morphisms $(\alpha_X : FX \to GX)$ such that for any morphism $f : X \to Y$ in $\mathcal{C}$, the following diagram commutes.

$$\begin{tikzcd} FX \arrow[r, "\alpha_X"] \arrow[d, "Ff"'] & GX \arrow[d, "Gf"] \\ FY \arrow[r, "\alpha_Y"'] & GY \end{tikzcd}$$

Consider a regular language $\mathcal{L} \subseteq \mathcal{A}^*$ and the category of deterministic automata that accept the language $\mathcal{L}$. This will be a subcategory of the category of $(\mathbf{Set}, \mathbf{1}, \mathbf{2})$-automata.

$$\begin{tikzcd} \text{in} \arrow[r, "\triangleright"'] & \text{st} \arrow[loop above, "a"] \arrow[r, "\triangleleft"'] & \text{out} \arrow[r, "\mathcal{A}"] & \mathbf{Set} \end{tikzcd}$$

$\mathcal{A}$ is a functor such that $\mathcal{A}(\text{in}) = \mathbf{1}$ and $\mathcal{A}(\text{out}) = \mathbf{2}$.

Definition 2.21 Morphism of automata

A morphism of $(\mathbf{Set}, \mathbf{1}, \mathbf{2})$-automata $f : \mathcal{A} \to \mathcal{A}'$ is a natural transformation between two functors such that $f_{\text{in}} : \mathcal{A}(\text{in}) \to \mathcal{A}'(\text{in})$ and $f_{\text{out}} : \mathcal{A}(\text{out}) \to \mathcal{A}'(\text{out})$ are the identities on $\mathbf{1}$ respectively $\mathbf{2}$.

Concretely, a morphism $f : \mathcal{A} \to \mathcal{A}'$ is given by a morphism $f_{\text{st}} : \mathcal{A}(\text{st}) \to \mathcal{A}'(\text{st})$ such that all the diagrams below commute.

$$\begin{tikzcd} & \mathcal{A}(\text{st}) \arrow[dd, "f_{\text{st}}"] \\ \mathbf{1} \arrow[ur, "\mathcal{A}(\triangleright)"] \arrow[dr, "\mathcal{A}'(\triangleright)"'] & \\ & \mathcal{A}'(\text{st}) \end{tikzcd}$$

f_{st} preserves the initial state

$$\begin{tikzcd} \mathcal{A}(\text{st}) \arrow[r, "\mathcal{A}(a)"] \arrow[d, "f_{\text{st}}"'] & \mathcal{A}(\text{st}) \arrow[d, "f_{\text{st}}"] \\ \mathcal{A}'(\text{st}) \arrow[r, "\mathcal{A}'(a)"'] & \mathcal{A}'(\text{st}) \end{tikzcd}$$

f_{st} commutes with the transition functions

$$\begin{tikzcd} \mathcal{A}(\text{st}) \arrow[dd, "f_{\text{st}}"'] \arrow[dr, "\mathcal{A}'(\triangleleft)"] & \\ & \mathbf{2} \\ \mathcal{A}'(\text{st}) \arrow[ur, "\mathcal{A}(\triangleleft)"'] & \end{tikzcd}$$

f_{st} preserves the accepting states

Let O be the full subcategory of $\mathcal{I}$ over the objects in and out, that is the objects of O are "in" and "out", and for every $w \in \mathcal{A}^*$ we have a morphism $\triangleright w \triangleleft : \text{in} \to \text{out}$ in O.

$$\begin{tikzcd} \text{in} \arrow[r, "\triangleright w \triangleleft"] & \text{out} \arrow[r] & \text{in} \arrow[r, "\triangleright"'] & \text{st} \arrow[loop above, "a"] \arrow[r, "\triangleleft"'] & \text{out} \end{tikzcd}$$

2.2.2 Language accepted by an automaton

Definition 2.22 Language accepted by a $(\mathcal{C}, I, F)$-automaton

Consider a category $\mathcal{C}$ and two objects I and F. The "language" accepted by a $(\mathcal{C}, I, F)$-automaton $\mathcal{A} : \mathcal{I} \to \mathcal{C}$ is the composite $O \underset{\subseteq}{\to} \mathcal{I} \xrightarrow{\mathcal{A}} \mathcal{C}$.

Example 2.23 Language accepted by a $(\mathbf{Set}, \mathbf{1}, \mathbf{2})$-automaton

Let $\mathcal{C} = \mathbf{Set}$, $I = \mathbf{1}$ and $F = \mathbf{2}$.
A language accepted by a $(\mathbf{Set}, \mathbf{1}, \mathbf{2})$-automaton is a functor $\mathcal{L} : O \to \mathbf{Set}$ such that $\mathcal{L}(\text{in}) = \mathbf{1}$ and $\mathcal{L}(\text{out}) = \mathbf{2}$.

$$\begin{tikzcd} O \arrow[d] & \text{in} \arrow[r, "\rhd w \lhd"] \arrow[d, mapsto] & \text{out} \arrow[d, mapsto] \\ \mathbf{Set} & \mathbf{1} \arrow[r, "\mathcal{L}(\rhd w \lhd)"'] & \mathbf{2} \end{tikzcd}$$

For every $w \in \mathcal{A}^*$, there is a function $\mathcal{L}(\rhd w \lhd) : \begin{array}{ccc} \mathbf{1} & \longrightarrow & \mathbf{2} \\ \{*\} & \longmapsto & \{0,1\} \end{array}$ such that $\mathcal{L}(\rhd w \lhd)(*) = 0$ if $w \in \mathcal{L}$ and 1 otherwise.

$$\mathbf{1} \xrightarrow[\mathcal{A}(\rhd)]{q_0} Q \xrightarrow[\mathcal{A}(a_1)]{\delta_{a_1}} Q \xrightarrow[\mathcal{A}(a_2)]{\delta_{a_2}} \dots \xrightarrow[\mathcal{A}(\lhd)]{\lhd} \mathbf{2}$$

Recall that a K-weighted automaton can be represented as a $(K\text{-}\mathbf{Vec}, K, K)$-automaton, that is a functor $\mathcal{A} : \mathcal{I} \to K\text{-}\mathbf{Vec}$ with $\mathcal{A}(\text{in}) = K$ and $\mathcal{A}(\text{out}) = K$.

Given $w = a_1 \dots a_n \in \mathcal{A}^*$, the behaviour of the automaton on w is computed as the composite

$$K \xrightarrow[\mathcal{A}(\rhd)]{} Q \xrightarrow[\mathcal{A}(a_1)]{} Q \longrightarrow \dots \xrightarrow[\mathcal{A}(a_n)]{} Q \xrightarrow[\mathcal{A}(\lhd)]{} K.$$

For every $w \in \mathcal{A}^*$, we obtain a linear transformation $\mathcal{A}(\rhd w \lhd) : K \to K$. Since the set of linear transformations from K to K verifies $K\text{-}\mathbf{Vec}(K, K) \simeq K$, we can think of $\mathcal{A}(\rhd w \lhd)$ as an element of K. This is exactly $|\mathcal{A}|(w)$.

$$\overset{\rhd w \lhd : \text{in} \to \text{out}}{O} \xrightarrow[\subseteq]{} \overset{\rhd w \lhd : \text{in} \to \text{out}}{\mathcal{I}} \xrightarrow[\mathcal{A}]{} \overset{\mathcal{A}(\rhd w \lhd)}{K\text{-}\mathbf{Vec}}$$

The language recognized by the automaton is $\mathcal{L} : \begin{array}{ccc} A^* & \longrightarrow & K\text{-}\mathbf{Vec}(K, K) \\ w & \longmapsto & \mathcal{L}(\rhd w \lhd) \end{array}$ that is, $\mathcal{L} : O \to K\text{-}\mathbf{Vec}$ corresponds to a series $A^* \to K$ in $K\langle\langle A^* \rangle\rangle$.

Notation 2.24 $\mathrm{Auto}(\mathcal{L})$

Lec $\mathcal{C}$ be a category and I, F be two objects of $\mathcal{C}$. Let $\mathcal{L} : O \to \mathcal{C}$ be a functor with $\mathcal{L}(\text{in}) = I$ and $\mathcal{L}(\text{out}) = F$.
We denote by $\mathrm{Auto}(\mathcal{L})$ the category whose objects are the $(\mathcal{C}, I, F)$-automaton accepting $\mathcal{L}$ and morphisms are $(\mathcal{C}, I, F)$-automata morphisms.

$$\mathcal{A} : \begin{array}{ccc} \mathcal{I} & \longrightarrow & \mathcal{C} \\ \text{in} & \longmapsto & I \\ \text{out} & \longmapsto & F \end{array}$$

$\mathcal{A}$ accepts a given language

$$\begin{tikzcd} O \arrow[d] \arrow[dr, "\mathcal{L}"] & \\ \mathcal{I} \arrow[r, "\mathcal{A}"'] & \mathcal{C} \end{tikzcd}$$

The question is how to obtain the minimal automaton accepting $\mathcal{L}$ by looking at the properties of $\mathcal{C}$.

Example 2.25 $(\mathbf{Set}, \mathbf{1}, \mathbf{2})$**-automata**

Consider a language $L \subseteq A^*$ seen as a functor $\mathcal{L} : O \to \mathbf{Set}$.
We conider the minimal automaton $\mathrm{Min}(\mathcal{L})$ by factorizing in the category $\mathrm{Auto}(\mathcal{L})$ the unique morphism from the initial object $\mathrm{Auto}(\mathcal{L})$ to the final object of $\mathrm{Auto}(\mathcal{L})$.

2.2.3 Minimization of deterministic automata

Definition 2.26 **Minimal object**

Let $\mathcal{C}$ be a category with an initial object X, a final object Y and a factorization system (E, M).

The minimal object of $\mathcal{C}$ is a factorization of the unique morphism $\begin{tikzcd} X \arrow[rr] \arrow[dr, two heads] & & Y \\ & \mathrm{Min} \arrow[ur, tail] & \end{tikzcd}$.

Definition 2.27 Reach**,** Obs

Let $\mathcal{C}$ be a category with an initial object X, a final object Y and a factorization system (E, M).
Given an object A in $\mathcal{C}$, define $\mathrm{Reach}(A)$ as a factorization of the unique morphism from X to A and define $\mathrm{Obs}(A)$ as a factorization of the unique morphism.

$$\begin{tikzcd} X \arrow[rr] \arrow[dr, two heads] & & A \\ & \mathrm{Reach}(A) \arrow[ur, tail] & \end{tikzcd} \qquad \begin{tikzcd} A \arrow[rr] \arrow[dr, two heads] & & Y \\ & \mathrm{Obs}(A) \arrow[ur, tail] & \end{tikzcd}$$

Exercise 2.28

Let $\mathcal{L}$ be a language $O \to \mathbf{Set}$.
Prove that the initial object in $\mathrm{Auto}(\mathcal{L})$ is the $(\mathbf{Set}, \mathbf{1}, \mathbf{2})$-automaton $\mathcal{A}_{\mathsf{init}} : \mathcal{I} \to \mathbf{Set}$ where

- $\mathcal{A}_{\mathsf{init}}(\mathsf{st}) = A^*$
- $\mathcal{A}_{\mathsf{init}}(\triangleright) = \begin{array}{ccc} \mathbf{1} & \longrightarrow & A^* \\ * & \longmapsto & \varepsilon \end{array}$
- $\mathcal{A}_{\mathsf{init}}(a) = \begin{array}{ccc} A^* & \longrightarrow & A^* \\ w & \longmapsto & wa \end{array}$
- $\mathcal{A}_{\mathsf{init}}(\triangleleft) = \begin{array}{ccc} A^* & \longrightarrow & \mathbf{2} \\ w & \longmapsto & \begin{cases} 0 & \text{if } w \notin \mathcal{L} \\ 1 & \text{if } w \in \mathcal{L} \end{cases} \end{array}$

$$\begin{tikzcd} \mathbf{1} \arrow[r, "\varepsilon"] & A^* \arrow[loop above, "w \mapsto wa"] \arrow[r, "\mathcal{L}?"] & \mathbf{2} \end{tikzcd}$$

Proof.

$$\begin{tikzcd} & A^* \arrow[loop above, "a"] \arrow[dr, "\mathcal{L}?"] \arrow[dd, dashed, "f"] & \\ \mathbf{1} \arrow[ur, "\varepsilon"] \arrow[dr, "q_0"'] & & \mathbf{2} \\ & Q \arrow[loop below, "\delta_a"] \arrow[ur, "\mathsf{acc}"'] & \end{tikzcd}$$

$$\begin{tikzcd} & A^* \arrow[dd, dashed, "f"] \\ \mathbf{1} \arrow[ur, "\varepsilon"] \arrow[dr, "q_0"'] & \\ & Q \end{tikzcd} \qquad \begin{tikzcd} A^* \arrow[r, "a"] \arrow[d, dashed, "f"'] & A^* \arrow[d, dashed, "f"] \\ Q \arrow[r, "\delta_a"'] & Q \end{tikzcd} \qquad \begin{tikzcd} A^* \arrow[dd, dashed, "f"'] \arrow[dr, "\mathcal{L}?"] & \\ & \mathbf{2} \\ Q \arrow[ur, "\mathsf{acc}"] & \end{tikzcd}$$

$f(\varepsilon) = q_0$ $\qquad$ $f(wa) = \delta_a(f(w))$

Notice that f is the unique morphism of automata from A^* to Q.
Given $w \in A^*$, $f(w)$ is the state of Q reached by reading the word w from the initial state q_0 of Q.

Since $\mathbf{1} \xrightarrow[q_0]{} Q \xrightarrow[\mathsf{acc}]{} \mathbf{2}$ accepts the same language $\mathcal{L}$, the triangle $\begin{tikzcd} A^* \arrow[dd, "f"'] \arrow[dr, "\mathcal{L}?"] & \\ & \mathbf{2} \\ Q \arrow[ur, "\mathsf{acc}"] & \end{tikzcd}$ also commutes.

Exercise 2.29

Let $\mathcal{L}$ be a language $O \to \mathbf{Set}$.
Prove that the final object in $\mathrm{Auto}(\mathcal{L})$ is the $(\mathbf{Set}, \mathbf{1}, \mathbf{2})$-automaton $\mathcal{A}_{\mathsf{final}} : \mathcal{I} \to \mathbf{Set}$ where

- $\mathcal{A}_{\mathsf{final}}(\mathsf{st}) = 2^{A^*}$
- $\mathcal{A}_{\mathsf{final}}(\triangleright : \mathsf{in} \to \mathsf{out}) = \begin{array}{ccc} \mathbf{1} & \longrightarrow & 2^{A^*} \\ * & \longmapsto & \varepsilon \end{array}$
- $\mathcal{A}_{\mathsf{final}}(a : \mathsf{st} \to \mathsf{st}) = \begin{array}{ccc} 2^{A^*} & \longrightarrow & 2^{A^*} \\ K & \longmapsto & a^{-1}K \end{array}$
- $\mathcal{A}_{\mathsf{final}}(\triangleleft) = \begin{array}{ccc} 2^{A^*} & \longrightarrow & \mathbf{2} \\ w & \longmapsto & \begin{cases} 0 & \text{if } \varepsilon \notin \mathcal{L} \\ 1 & \text{if } \varepsilon \in \mathcal{L} \end{cases} \end{array}$

$$\begin{tikzcd} \mathbf{1} \arrow[r, "\mathcal{L}"] & 2^{A^*} \arrow[loop above, "K \mapsto a^{-1}K"] \arrow[r, "\varepsilon?"] & \mathbf{2} \end{tikzcd}$$

Proof.

$$\begin{tikzcd} & Q \arrow[loop above, "\delta_a"] \arrow[dd, dashed, "h"] \arrow[dr, "\mathcal{L}?"] & \\ \mathbf{1} \arrow[ur, "\varepsilon"] \arrow[dr, "q_0"'] & & \mathbf{2} \\ & A^* \arrow[loop below, "a"] \arrow[ur, "\mathsf{acc}"'] & \end{tikzcd}$$

$$\begin{tikzcd} & Q \arrow[dd, dashed, "h"] \\ \mathbf{1} \arrow[ur, "q_0"] \arrow[dr, "\mathcal{L}"'] & \\ & 2^{A^*} \end{tikzcd} \qquad \begin{tikzcd} Q \arrow[r, "\delta_a"] \arrow[d, dashed, "h"'] & Q \arrow[d, dashed, "h"] \\ 2^{A^*} \arrow[r, "a"'] & 2^{A^*} \end{tikzcd} \qquad \begin{tikzcd} Q \arrow[dd, dashed, "h"'] \arrow[dr, "\mathsf{acc}"] & \\ & \mathbf{2} \\ 2^{A^*} \arrow[ur, "\varepsilon?"'] & \end{tikzcd}$$

$$K \longmapsto a^{-1}K$$

For $q \in Q$, defined $h(q)$ as the language accepted from the state q, that is, $h(q) = \{w \in A^* \mid \delta_w(q) \text{ is accepting}\}$.

$$\begin{tikzcd}
q \arrow[r, mapsto, "\delta_a"] \arrow[d, dashed, mapsto, "h"'] & \delta_a(q) \arrow[dr, dashed, mapsto, "h"] & \\
\mathcal{L}(q) \arrow[rr, mapsto, "a"'] & & a^{-1}\mathcal{L}(q) = \{u \mid au \in \mathcal{L}(q)\}
\end{tikzcd}$$

Exercise 2.30

Let E be the class of $\mathrm{Auto}(\mathcal{L})$ morphisms such that the component for the object st is a surjection.

$$\begin{tikzcd}
& Q \arrow[loop above, "a"] \arrow[dd, "e"] \arrow[dr] & \\
\mathbf{1} \arrow[ur] \arrow[dr] & & \mathbf{2} \\
& Q \arrow[loop below, "a"] \arrow[ur] &
\end{tikzcd}$$

$$\begin{tikzcd}
\mathcal{A} \arrow[r, two heads, "e"] & \mathcal{A}'
\end{tikzcd}$$

$$e \in E \iff e_{\mathsf{st}} : \mathcal{A}(\mathsf{st}) \to \mathcal{A}(\mathsf{st}) \text{ is a surjection}$$

Consider M the class of morphisms in $\mathrm{Auto}(\mathcal{L})$ such that the st-component is an injection.

$$\begin{tikzcd}
& Q \arrow[loop above, "a"] \arrow[dd, tail, "m"] \arrow[dr] & \\
\mathbf{1} \arrow[ur] \arrow[dr] & & \mathbf{2} \\
& Q \arrow[loop below, "a"] \arrow[ur] &
\end{tikzcd}$$

Show that (E, M) is a factorization system in the category $\mathrm{Auto}(\mathcal{L})$.

Consider the unique morphism from $\mathcal{A}_{\mathsf{init}}(\mathcal{L})$ to the final object $\mathcal{A}_{\mathsf{final}}(\mathcal{L})$ of $\mathrm{Auto}(\mathcal{L})$.

$$
\begin{tikzcd}
 & A^* \arrow[loop above, "w \mapsto wa"] \arrow[drr, "\mathcal{L}?"] \arrow[dr, two heads] \arrow[dd, dashed, "\exists! g"'] & & \\
\mathbf{1} \arrow[ur, "\varepsilon"] \arrow[dr, "\mathcal{L}"'] & & \mathrm{Im}(g) \arrow[dl, tail] & \mathbf{2} \\
 & 2^{A^*} \arrow[loop below, "K \mapsto a^{-1}K"] \arrow[urr, "\varepsilon?"'] & &
\end{tikzcd}
$$

$g(w) = w^{-1}L = \{u \mid wu \in \mathcal{L}\}$

If $w = a_1 a_2 \ldots a_n$,

$$
\begin{array}{ccccccccc}
\mathbf{1} & \xrightarrow[\mathcal{L}]{} & 2^{A^*} & \xrightarrow[a_1]{} & 2^{A^*} & \xrightarrow[a_1]{} & \ldots & \xrightarrow{\varepsilon?} & \mathbf{2} \\
* & \longmapsto & \mathcal{L} & \longmapsto & a^{-1}\mathcal{L} & \longmapsto & \ldots & \overset{\varepsilon?}{\longmapsto} &
\end{array}
$$

$$
\begin{tikzcd}
 & & A^* \arrow[dl, two heads] \arrow[dd, two heads] \arrow[ddr, two heads] \arrow[ddrr] & & \\
 & \mathrm{Reach}(Q) \arrow[dd, tail] & & & \\
\mathbf{1} \arrow[uurr] \arrow[dr, "q_0"] \arrow[ddrr] & & \mathrm{Obs}(\mathrm{Reach}(Q)) \arrow[dd, tail] & \mathrm{Min}(\mathcal{L}) \arrow[ddl, tail] & \mathbf{2} \\
 & Q \arrow[loop below, "\delta_a"] \arrow[urrr, "\mathsf{acc}"] \arrow[dr] & & & \\
 & & 2^{A^*} \arrow[uurr] & &
\end{tikzcd}
$$

$$
\begin{tikzcd}
 & \mathrm{Obs}(\mathrm{Reach}(Q)) \arrow[dr, tail] & \\
A^* \arrow[ur, two heads] \arrow[rr] \arrow[dr, two heads] & & 2^{A^*} \\
 & \mathrm{Min}(\mathcal{L}) \arrow[ur, tail] &
\end{tikzcd}
$$

These are the two factorizations of the unique morphism of automata from A^* to 2^{A^*}. Hence $\mathrm{Obs}(\mathrm{Reach}(Q)) \simeq \mathrm{Min}(\mathcal{L})$. So $\mathrm{Min}(\mathcal{L})$ divides Q.

Exercise 2.31

Show that $\mathrm{Min} \simeq \mathrm{Obs}(\mathrm{Reach}(\mathcal{A})) \simeq \mathrm{Reach}(\mathrm{Obs}(A))$.

2.2.4 Minimization of K weighted automata

Let $L : A^* \to K$ be a series, equivalently seen as a functor $\mathcal{L} : O \to K\text{-}\mathbf{Vec}$. Consider the category $\mathrm{Auto}(\mathcal{L})$ of $(K\text{-}\mathbf{Vec}, K, K)$-automata accepting $\mathcal{L}$.

The minimal object of $\mathrm{Auto}(\mathcal{L})$ corresponds to the minimal K-weighted automata accepting $\mathcal{L}$.

$$\begin{tikzcd}
 & \mathrm{Reach}(Q) \arrow[dl, tail] \arrow[dr, two heads] & \\
Q \arrow[dr, two heads] & & \mathrm{Min} \arrow[dl, tail] \\
 & \mathrm{Obs}(Q) &
\end{tikzcd}$$

$$\begin{tikzcd}
0 \arrow[d, "i"'] \arrow[drr, "\mathcal{L}=\mathcal{A}\circ i", "\mathcal{A}"'] & & \\
\mathcal{I} & & K\text{-}\mathbf{Vec}
\end{tikzcd}$$

$$\begin{tikzcd}
\mathsf{in} \arrow[r, "\triangleright"'] & \mathsf{st} \arrow[loop above, "a"] \arrow[r, "\triangleleft"'] & \mathsf{out} \arrow[r, mapsto] & K \arrow[r, "\mathcal{A}(\triangleright)"'] & Q \arrow[loop above, "\mathcal{A}(a)"] \arrow[r, "\mathcal{A}(\triangleleft)"'] & K
\end{tikzcd}$$

$$L : A^* \to K \iff \mathcal{L} : \begin{array}{ccc} O & \to & K\text{-}\mathbf{Vec} \\ \mathsf{in} & \mapsto & K \\ \mathsf{out} & \mapsto & K \end{array}$$

What is the minimal $(K\text{-}\mathbf{Vec}, K, K)$-automaton accepting $\mathcal{L}$?

1. $\mathcal{A}_{\mathsf{init}}(\mathcal{L})$ is the initial object of $\mathrm{Auto}(\mathcal{L})$.
2. $\mathcal{A}_{\mathsf{final}}(\mathcal{L})$ is the final object of $\mathrm{Auto}(\mathcal{L})$.
3. For a factorization system (E, M) of $\mathrm{Auto}(\mathcal{L})$ we obtain a minimal automaton as the factorization of the unique morphism

$$\begin{tikzcd}
\mathcal{A}_{\mathsf{init}}(\mathcal{L}) \arrow[rr] \arrow[dr, two heads] & & \mathcal{A}_{\mathsf{final}}(\mathcal{L}) \\
 & \mathrm{Min}(\mathcal{L}) \arrow[ur, tail] &
\end{tikzcd}$$

1. $\mathcal{A}_{\mathsf{init}}(\mathcal{L})$. K^{A^*} is the vector space with base A^*. It is all the finitely supported functions $A^* \to K$ with a suitable K-vector space structure.

$\mathcal{A}_{\mathsf{init}}(\mathcal{L})$ is of the shape $\begin{array}{ccc} \mathcal{I} & \to & K\text{-}\mathbf{Vec} \\ \mathsf{in}, \mathsf{out} \to & \mapsto & K \\ \mathsf{st} & \mapsto & K^{A^*} \end{array}$.

$\mathcal{A}_{\mathsf{init}}(\mathcal{L})(\triangleright : \mathsf{in} \to \mathsf{st}) : \begin{array}{ccc} K & \to & K^{A^*} \\ 1_K & \mapsto & \varepsilon \end{array}$ is a linear transformation.

With a an input letter, $\mathcal{A}_{\mathsf{init}}(\mathcal{L})(a : \mathsf{st} \to \mathsf{st}) : \begin{array}{ccc} K^{A^*} & \to & K^{A^*} \\ w & \mapsto & wa \end{array}$ is a linear transformation.

$\mathcal{A}_{\mathsf{init}}(\mathcal{L})(\triangleleft : \mathsf{st} \to \mathsf{out}) : \begin{array}{ccc} K^{A^*} & \to & K \\ w & \mapsto & L(w) \end{array}$ is a linear transformation.

$$\begin{tikzcd}
K \arrow[r, "\varepsilon"'] \arrow[dr, bend right, "q_0"'] & K^{A^*} \arrow[loop above, "w \mapsto wa"] \arrow[r, "\mathcal{L}?"'] & K \\
 & Q \arrow[loop below, "a"] \arrow[ur, bend right, "f"'] &
\end{tikzcd}$$

Exercise 2.32

Prove that $\mathcal{A}_{\mathsf{init}}(\mathcal{L})$ is an initial object in $\mathrm{Auto}(\mathcal{L})$.

2. $\mathcal{A}_{\mathsf{final}}(\mathcal{L})$. $\mathcal{A}_{\mathsf{final}}(\mathcal{L})$ is of the shape $\begin{array}{rcl} \mathcal{I} & \to & K\text{-}\mathbf{Vec} \\ \mathsf{in}, \mathsf{out} \to & \mapsto & K \\ \mathsf{st} & \mapsto & K\langle\langle A^*\rangle\rangle \end{array}$.

$$\mathcal{A}_{\mathsf{final}}(\mathcal{L})(\triangleright : \mathsf{in} \to \mathsf{st}) : \begin{array}{rcl} K & \to & K\langle\langle A^*\rangle\rangle \\ 1_K & \mapsto & \sum_{w \in A^*} L(w)w = L \end{array}.$$

For $a \in A$, $\mathcal{A}_{\mathsf{final}}(\mathcal{L})(a : \mathsf{st} \to \mathsf{st}) : \begin{array}{rcl} K\langle\langle A^*\rangle\rangle & \to & K\langle\langle A^*\rangle\rangle \\ R & \mapsto & (w \mapsto R(aw)) \end{array}$.

$$\mathcal{A}_{\mathsf{final}}(\mathcal{L})(\triangleleft : \mathsf{st} \to \mathsf{out}) : \begin{array}{rcl} K\langle\langle A^*\rangle\rangle & \to & K \\ R & \mapsto & R(\varepsilon) \end{array}.$$

$$\begin{tikzcd} K \arrow[r, "\mathcal{L}"'] & K\langle\langle A^*\rangle\rangle \arrow[loop above, "R \mapsto a^{-1}R"] \arrow[r, "\varepsilon?"'] & K \end{tikzcd}$$

Exercise 2.33

Show that $\mathcal{A}_{\mathsf{final}}(\mathcal{L})$ is the final object of $\mathrm{Auto}(\mathcal{L})$.

$$\begin{tikzcd} & K^{A^*} \arrow[dd, "\exists! h"'] \arrow[dr, two heads] \arrow[drr, bend left, "\mathcal{L}?"] & & \\ K \arrow[ur, bend left, "\varepsilon"] \arrow[dr, bend right, "\mathcal{L}"'] & & \mathrm{Min}(\mathcal{L})(\mathrm{st}) \arrow[dl, tail] & K \\ & K\langle\langle A^*\rangle\rangle \arrow[urr, bend right, "\varepsilon?"'] & & \end{tikzcd}$$

The unique morphism $h : K^{A^*} \to K\langle\langle A^*\rangle\rangle$ is defined by $\forall w \in A^*, h(w) = w^{-1}\mathcal{L}$, with

$$w^{-1}\mathcal{L} : \begin{array}{rcl} A^* & \longrightarrow & K \\ u & \longmapsto & \mathcal{L}(wu) \end{array}.$$

Lemma 2.34

If $\mathcal{C}$ has a factorization system (E, M) then $\mathrm{Auto}(\mathcal{L})$ has a factorization system $(E_{\mathrm{Auto}}, M_{\mathrm{Auto}})$ where a natural transformation α is in E_{Auto} if all its components are in E (that is $\alpha_{\mathsf{st}} \in E$). And similarly, $\alpha \in M_{\mathrm{Auto}}$ iff all its components are in M (that is $\alpha_{\mathsf{st}} \in M$).

3 Basic definition of transducers

3.1 (Sub)Sequential transducers

Definition 3.1 **Sequential transducer**

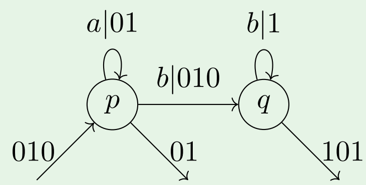
Let A and B be two finite sets, the input and output alphabets.
A sequential transducer is a tuple $(Q, q_0, u_0, \delta_a : Q \to Q, u_a : Q \to B^*, f : Q \to B^*)$ where

- Q is a finite set of states
- q_0 is either an initial state in Q or may be undefined

- $u_0 \in B^*$ is the initial output (when q_0 is defined)
- $\delta_a : Q \to Q$ is a partial transition function for $a \in A$
- $u_a : Q \to B^*$ is a partial output function for $a \in A$, such that $u_a(q)$ is defined iff $\delta_a(q)$ is defined
- $f : Q \to B^*$ is a partial final output function

Example 3.2

Let $A = \{a, b\}$ and $B = \{0, 1\}$. Consider the following transducer.



We have

- $Q = \{p, q\}$
- $q_0 = p$
- $u_0 =$
- $\delta_a(p) = p$, $\delta_a(q)$ is undefined
  $\delta_b(p) = \delta_b(q) = q$
- $u_a(p) = 01$ and $u_a(q)$ is undefined
  $u_b(p) = 010$ and $u_b(q) = 1$
- $f(p) = 01$ and $f(q) = 101$.

And $|\mathcal{A}| : \begin{array}{ccc} A^* & \longrightarrow & B^* \\ ab & \longmapsto & \underbrace{010}_{\text{initial}}\underbrace{01}_{p\to p}\underbrace{010}_{p\to q}\underbrace{101}_{\text{final}} \\ aba & \longmapsto & \perp \end{array}$ is a partial function.

In general, given $w \in A^*$, say $w = a_1 \ldots a_n$,

$$|\mathcal{A}|(w) = \begin{cases} u_0 u_{a_1}(q_0) u_{a_2}(q_1) \ldots u_{a_n}(q_{n-1}) f(q_n) & \text{if } q_0 \text{ is defined}, \delta_{a_i}(q_{i-1}) = q_i \in Q, f(q_n) \in B^* \\ \perp & \text{if some of the computations above is undefined} \end{cases} .$$

3.2 Categorical definition

Consider the category $\mathcal{S}$ whose objects are sets and, given two sets X and Y, the set of morphisms $\mathcal{S}(X, Y)$ is the set of all functions $X \to B^* \times Y + \mathbf{1}$, i.e. partial functions from X to $B^* \times Y$.
Notice that sequential transducers are just $(\mathcal{S}, \mathbf{1}, \mathbf{1})$-automata.

The identity morphisms are the $x \mapsto (\varepsilon, x)$.

Notation 3.3

Given $f : X \to B^* \times Y + \mathbf{1}$, we write $f_1 : X \to B^* + \mathbf{1}$ the function such that $f(x) = \begin{cases} \pi_1(f(x)) & \text{if } f(x) \neq \perp \\ \perp & \text{if } f(x) = \perp \end{cases}$.
And f_2 similarly.

Definition 3.4 **Composition of morphism in $\mathcal{S}$**

Let X, Y, Z be sets, $f \in \mathcal{S}(X,Y)$ and $g \in \mathcal{S}(Y,Z)$.

$$\begin{tikzcd} X \arrow[r, "f"] \arrow[rr, bend right, "g \circ f"'] & Y \arrow[r, "g"] & Z \end{tikzcd}$$

Then $g \circ f$ is defined by $g \circ f(x) = \begin{cases} ((f_1(x) \cdot g_1(y)), g_2(y)) & \text{if } f_2(x) = y \text{ and } g_2(y) \text{ are defined} \\ \perp & \text{otherwise} \end{cases}$.

Notation 3.5

We write

$$\begin{tikzcd} X \arrow[r, "f", "\circ" marking, no head] & Y \end{tikzcd}$$

for a morphism $f \in \mathcal{S}(X,Y)$, that is a function $X \xrightarrow{f} B^* \times Y + \mathbf{1}$ in $\mathbf{Set}$.

$$\begin{tikzcd} \text{in} \arrow[r, "\triangleright"'] & \text{st} \arrow[loop above, "a"] \arrow[r, "\triangleleft"'] & \text{out} \arrow[r, "\mathcal{A}"] & \mathcal{S} \end{tikzcd}$$

$$\text{in,out} \longmapsto \mathbf{1}$$
$$\text{st} \longmapsto Q$$

$$\begin{tikzcd} \mathbf{1} \arrow[r, "\mathcal{A}(\triangleright)", "\circ" marking] & Q \arrow[loop above, "\mathcal{A}(a)", "\circ" marking] \arrow[r, "\mathcal{A}(\triangleleft)", "\circ" marking] & \mathbf{1} \end{tikzcd}$$

- $\mathcal{A}(\triangleright) : \begin{array}{ccl} \mathbf{1} & \longrightarrow & B^*Q + \mathbf{1} \\ * & \longmapsto & (u_0, q_0) \text{ or } \perp \end{array}$

- $\mathcal{A}(a) : \begin{array}{ccl} Q & \longrightarrow & B^*Q + \mathbf{1} \\ q & \longmapsto & (u_a(q), \delta_a(q)) \text{or } \perp \end{array}$

- $\mathcal{A}(\triangleleft) : \begin{array}{ccl} Q & \longrightarrow & B^*Q + \mathbf{1} \\ q & \longmapsto & f(q) \end{array}$

A language is a function $\mathcal{L} : A^* \to B^* + \mathbf{1}$.

$$\begin{array}{ccc} O & \xrightarrow{\mathcal{L}} & \mathcal{S} \\ \text{in, out} & \longrightarrow & \mathbf{1} \end{array}$$

$\triangleright w \triangleleft : \text{in} \to \text{out} \longmapsto 1 \nrightarrow 1$ in $\mathcal{S}$, $1 \to B^* \times 1 + \mathbf{1} \simeq B^* + \mathbf{1}$.

$\mathrm{Auto}(\mathcal{L})$ is the category of $(\mathcal{S}, \mathbf{1}, \mathbf{1})$-automata accepting $\mathcal{L}$.

Given $\mathcal{L} : A^* \to B^* + \mathbf{1}$, we have to

1. prove that $\mathcal{A}_{\text{init}}(\mathcal{L})$ is the initial object of $\mathrm{Auto}(\mathcal{L})$
2. prove that $\mathcal{A}_{\text{final}}(\mathcal{L})$ is the final object of $\mathrm{Auto}(\mathcal{L})$
3. find a suitable factorization system on $\mathcal{S}$ (and hence also on $\mathrm{Auto}(\mathcal{L})$)

3.2.1 $\mathcal{A}_{\text{init}}(\mathcal{L})$

$\mathcal{A}_{\text{init}}(\mathcal{L})$ is of the shape $\begin{array}{ccc} \mathcal{I} & \to & \mathcal{S} \\ \text{in, out} \to & \mapsto & \mathbf{1} \\ \text{st} & \mapsto & A^* \end{array}$.

$$\begin{tikzcd} \mathbf{1} \arrow[r, "\mathcal{A}_{\mathsf{init}}(\triangleright)"'] & A^* \arrow[loop below, "\mathcal{A}_{\mathsf{init}}(a)"] \arrow[r, "\mathcal{A}_{\mathsf{init}}(\triangleleft)"'] & \mathbf{1} \end{tikzcd}$$

- $\mathcal{A}_{\mathsf{init}}(\mathcal{L})(\triangleright : \mathsf{in} \to \mathsf{st}) : \begin{array}{ccc} \mathbf{1} & \nrightarrow & A^* \\ 1 & \to & B^* \times A^* + \mathbf{1} \\ * & \mapsto & (\varepsilon_{B^*}, \varepsilon_{A^*}) \end{array}$.
- For $a \in A$, $\mathcal{A}_{\mathsf{init}}(\mathcal{L})(a : \mathsf{st} \to \mathsf{st}) : \begin{array}{ccc} A^* & \nrightarrow & A^* \\ A^* & \to & B^* \times A^* + \mathbf{1} \\ w & \mapsto & (\varepsilon_{B^*}, wa) \end{array}$.
- $\mathcal{A}_{\mathsf{init}}(\mathcal{L})(\triangleleft : \mathsf{st} \to \mathsf{out}) : \begin{array}{ccc} A^* & \nrightarrow & \mathbf{1} \\ A^* & \to & B^* + \mathbf{1} \\ w & \mapsto & L(w) \end{array}$.

$$\begin{tikzcd} \mathbf{1} \arrow[r, "\mathcal{A}_{\mathsf{init}}(\triangleright)"] & A^* \arrow[loop above, "w \mapsto wa"] \arrow[r, "\mathcal{L}?"] & \mathbf{1} \end{tikzcd}$$

Exercise 3.6

Prove that $\mathcal{A}_{\mathsf{init}}(\mathcal{L})$ is an initial object of $\mathrm{Auto}(\mathcal{L})$.

$$\begin{tikzcd} & A^* \arrow[loop above] \arrow[dr, bend left] \arrow[dd, dashed, "\exists! h"'] & \\ \mathbf{1} \arrow[ur, bend left] \arrow[dr, bend right] & & \mathbf{1} \\ & Q \arrow[loop below] \arrow[ur, bend right] & \end{tikzcd}$$

With $h : \begin{array}{ccc} A^* & \nrightarrow & Q \\ A^* & \longrightarrow & B^* \times Q + \mathbf{1} \end{array}$.

3.2.2 $\mathcal{A}_{\mathsf{final}}(\mathcal{L})$

Definition 3.7 **Longest common prefix**

Consider a function $K : A^* \to B^* + \mathbf{1}$. Define the longest common prefix of K, $\mathrm{lcp}(K)$ as follows

$$\mathrm{lcp}(K) = \begin{cases} \perp & \text{if } \forall w \in A^*, K(w) = \perp \\ \text{longest common prefix}\{K(w) \,|\, w \in A^*, K(w) \neq \perp\} \in B^* & \text{otherwise} \end{cases},$$

where the longest common prefix is

$$u \iff \begin{cases} \forall w \in A^*, K(w) \neq \perp, \exists v \in B^*, K(w) = uv, \text{ that is } u \text{ is a prefix for all the defined } K(w) \\ u \text{ is the longest word in } B^* \text{ with this property} \end{cases}.$$

Example 3.8

$K : A^* \to B^* + \mathbf{1}$
$K(\varepsilon) = 010$
$K(a) = 0110$
$K(b) = 0101$
$K(w) = \perp$ if $|w| \geqslant 2$
Then $\mathrm{lcp}(K) = 01$.

Definition 3.9 **Irreductible function**

A function $K : A^* \to B^* + \mathbf{1}$ is called irreductible if $\mathrm{lcp}(K) = \varepsilon$.

Notation 3.10

Given a function $f : A^* \to B^* + \mathbf{1}$ and $u \in B^*$, we write $u \cdot K$ for the function $A^* \to B^* + \mathbf{1}$ defined by

$$u \cdot K(w) = \begin{cases} u \cdot K(w) & \text{if } K(w) \neq \bot \\ \bot & \text{otherwise} \end{cases}$$

Definition 3.11 **Reduced part**

Given a function $K : A^* \to B^* + \mathbf{1}$, which is not constant $\bot$, we define its reduced part $\mathrm{red}(K)$ as the unique function $\mathrm{red}(K) : A^* \to B^* + \mathbf{1}$ such that $K = \mathrm{lcp}(K) \cdot \mathrm{red}(K)$.

Notation 3.12 **Irreducible functions**

We denote by $\mathrm{Irr}(A^*, B^*)$ the set of irreducible functions $A^* \to B^* + \mathbf{1}$.

Notice that we have an isomorphism

$$(B^* + \mathbf{1})^{A^*} \simeq B^* \times \mathrm{Irr}(A^*, B^*) + \mathbf{1}.$$

Now we can define $\mathcal{A}_{\mathsf{final}}(\mathcal{L}) : \begin{array}{rcl} \mathcal{I} & \to & \mathcal{S} \\ \mathsf{in}, \mathsf{out} \to & \mapsto & \mathbf{1} \\ \mathsf{st} & \mapsto & \mathrm{Irr}(A^*, B^*) \end{array}$.

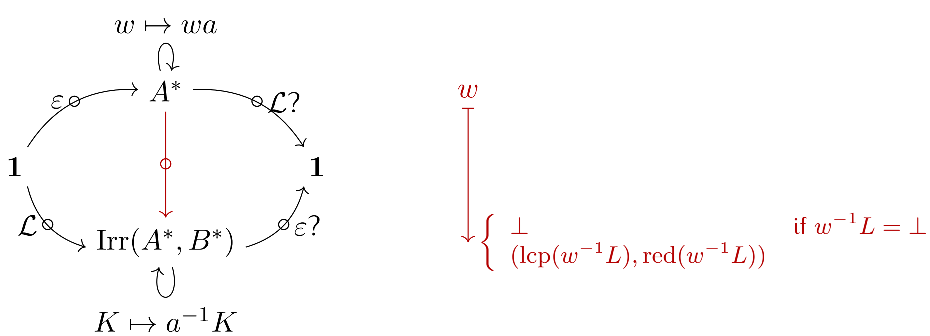
- $\mathcal{A}_{\mathsf{final}}(\mathcal{L})(\triangleright : \mathsf{in} \to \mathsf{st}) : \begin{array}{rcl} \mathbf{1} & \nrightarrow & \mathrm{Irr}(A^*, B^*) \\ \mathbf{1} & \to & B^* \times \mathrm{Irr}(A^*, B^*) + \mathbf{1} \\ * & \mapsto & \begin{cases} \bot & \text{if } L \text{ is constant } \bot \\ (\mathrm{lcp}(L), \mathrm{red}(L)) \end{cases} \end{array}$.

- For $a \in A$, $\mathcal{A}_{\mathsf{final}}(\mathcal{L})(a : \mathsf{st} \to \mathsf{st}) : \begin{array}{rcl} \mathrm{Irr}(A^*, B^*)^* & \nrightarrow & \mathrm{Irr}(A^*, B^*) \\ \mathrm{Irr}(A^*, B^*) & \to & B^* \times \mathrm{Irr}(A^*, B^*) + \mathbf{1} \\ K & \mapsto & \begin{cases} \bot \text{ if } K \text{ is constant } \bot \\ (\mathrm{lcp}(a^{-1}K), \mathrm{red}(a^{-1}K)) \end{cases} \end{array}$,

  with $a^{-1}K : \begin{array}{rcl} A^* & \to & B^* \\ w & \mapsto & K(aw) \end{array}$.

- $\mathcal{A}_{\mathsf{final}}(\mathcal{L})(\triangleleft : \mathsf{st} \to \mathsf{out}) : \begin{array}{rcl} \mathrm{Irr}(A^*, B^*) & \nrightarrow & \mathbf{1} \\ \mathrm{Irr}(A^*, B^*) & \to & B^* + \mathbf{1} \\ K & \mapsto & K(\varepsilon) \end{array}$.

$$\begin{tikzcd} \mathbf{1} \arrow[r, "\mathcal{L}"] & A^* \arrow[loop above, "K \mapsto a^{-1}K"] \arrow[r, "\varepsilon?"] & \mathbf{1} \end{tikzcd}$$



3.2.3 Factorization system on $\mathcal{S}$

Given a morphism $f : X \nrightarrow Y$ in $\mathcal{S}$, we say that

1. $f \in E$ iff $f_2 : X \to Y + \mathbf{1}$ is surjective
2. $f \in M$ iff $f_2 : X \to Y + \mathbf{1}$ is injective and $f_1 : X \to B^* + \mathbf{1}$ is constant and maps all $x \in X$ to $\varepsilon \in B^*$.

Exercise 3.13

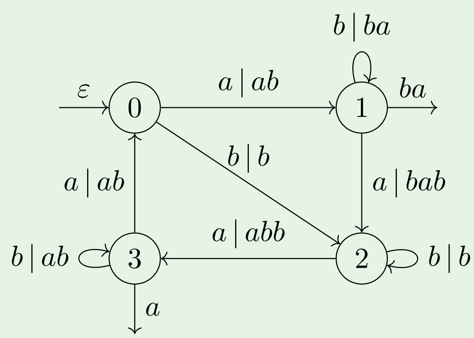
Prove that (E, M) is a factorization system in $\mathcal{S}$.

$$\begin{tikzcd} A^* \arrow[dd] \arrow[dr, two heads] & \\ & A^*/r \arrow[dl, tail] \\ \mathrm{Irr}(A^*, B^*) & \end{tikzcd}$$

4 TD

Exercise 4.1

Let $\mathcal{A}$ be the sequential transducer



For each state $q \in \{0, 1, 2, 3\}$ let $\mathcal{L}_q : \{a, b\}^* \to \{a, b\}^* \cup \{\perp\}$ denote he function realized by $\mathcal{A}$ taking q as initial state.

1. Compute $\mathrm{lcp}(\mathcal{L}_q)$ for each q.
2. Show that $\mathcal{L}_0$ and $\mathcal{L}_2$, repetively $\mathcal{L}_1$ and $\mathcal{L}_3$, have the same reduced part.
3. Compute the minimal sequential transducer accepting $\mathcal{L}_0$.

Exercise 4.2

Definition 4.3 **Normalized matrix representation**

Let $\mathcal{A}$ be a K-automaton of size n with matrix representation $(\underset{\in K^n}{I}, \underset{\in K^{n\times n}}{E}, \underset{\in K^n}{F})$.

We assume I is of the form

$$\begin{array}{|c|c|c|c|}\hline 1 & 0 & \dots & 0 \\ \hline\end{array}$$

and that E is of the form

$$\begin{array}{|c|c|}\hline 1 & y \\ \hline 0 & \\ \vdots & E \\ 0 & \\ \hline\end{array}$$

We call this a normalized matrix representation.

If $\mathcal{A}$ of size n and $\mathcal{A}'$ of size m have normalized matrix representations (I, E, F), resp. (I', E', F'), find matrix representations for automata accepting:

1. $|\mathcal{A}| + |\mathcal{A}'|$ (the pointwise sum)
2. $|\mathcal{A}| \cdot |\mathcal{A}'|$ (the Cauchy product of the series $|\mathcal{A}|$ and $|\mathcal{A}'|$)
3. $k|\mathcal{A}|$
4. $|\mathcal{A}| \cdot k$
5. $|\mathcal{A}|^*$ (if defined)

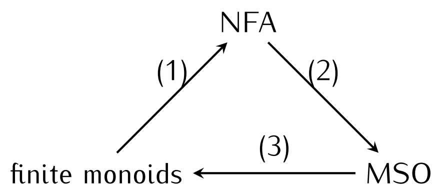
Part II
Automata, monoids and logic (Sam van Gool)

References

- Bojańzyk 2020
- Pin - MPRI 2020

5 Monoids and logic

5.1 First triangle



5.1.1 Finite monoids

Definition 5.1 Semigroup

A semigroup is a set equipped with a binary associative operation.

Definition 5.2 Monoid

A monoid is a semigroup equipped with a neutral element.

Example 5.3

- For Σ an alphabet, the set Σ^+ of non-empty finite words is a semigroup and the set Σ^* of finite words is a monoid.

  Fact 5.4

  For any function $f : \Sigma \to S$ with S a semigroup, there is a unique homomorphism $\overline{f} : \Sigma^+ \to S$ extending f.

- For any $c, d \geqslant 1$, the set $\{x, x^2, ..., x^c, x^{c+1}, ..., x^{c+d-1}\}$ subject to $x^{c+d} = x^c$ is a finite semigroup generated by x.
- For any set Q, the set $\mathrm{Rel}(Q)$ of relations on Q is a monoid under relational composition, with neutral element Δ.

Definition 5.5 NFA

A NFA is a tuple $(Q, \Sigma, \delta : \Sigma \to \mathrm{Rel}(Q), I \subseteq Q, F \subseteq Q)$.

Definition 5.6 Language recognized by a homomorphism

A language $L \subseteq \Sigma^+$ is recognized by a homomorphism $h : \Sigma^+ \to S$, with S a semigroup, if $h^{-1}(h(L)) = L$.

Definition 5.7 Language recognized by a NFA

The language recognized by a NFA $\mathcal{A} = (Q, \Sigma, \delta, I, F)$ is

$$\overline{\delta}^{-1}(\{\mathcal{R} \in \mathrm{Rel}(Q) \,|\, \mathcal{R} \cap (I \times F) \neq \emptyset\}).$$

Proposition 5.8

Any language L recognized by a finite monoid is recognized by a NFA.

Proof.
If $h : \Sigma^* \to M$ recognizes L then so does $(M, \Sigma, \delta, \{1_M\}, h(L))$ where $\delta(a) = \{(m, mh(a)) \,|\, m \in M\}$.

5.1.2 Monadic second order logic

Monadic second order logic is the extension of first-order logic by monadic second-order quantifiers.

Definition 5.9 Formulas of MSO

Given a vocabulary, i.e. a set of relation symbols, each with an associative arity, we build the formulas of MSO as follows:

- Atomic formulas $\mathcal{R}(x_1, ..., x_n)$ for $\mathcal{R}$ an n-ary relation
- Boolean connectives $\wedge$, $\vee$, $\neg$, $x_i = x_j$
- First-order quantifiers $\forall x$, $\exists x$
- Monadic second order quantifiers $\forall X$, $\exists X$
- Membership $x \in X$

Definition 5.10

A structure A together with n elements $p_1, ..., p_n$ and m subsets $P_1, ..., P_m$ satisfies an MSO formula $\varphi(x_1, ..., x_n, X_1, ..., X_m)$ if it is true.
Then we write $A, p_1, ..., p_n, P_1, ..., P_m \vDash \varphi$.

In particular, for Σ a finite alphabet, consider the signature $\{\leqslant^2\} \cup \{a^1\}_{a \in \Sigma}$.

Definition 5.11 Ordered structure

For $w \in \Sigma^*$, the ordered structure M_w associated to $w = w_1 ... w_n$ is the linear order $[\![1, n]\!]$ equipped with the partition

$$a^{M_w} := \{i \in [\![1, n]\!] \,|\, w_i = a\}.$$

Definition 5.12 Language defined by an MSO sentence

The language defined by an MSO sentence φ is $L_\varphi = \{w \in \Sigma^* \mid w \vDash \varphi\}$.

Example 5.13

- a^*bc^* is defined by $\exists x(b(x) \wedge \forall y(x < x \Rightarrow a(y) \wedge y > x \Rightarrow c(y)))$.
- $(aa)^*a$ is defined by

$$\begin{aligned}\exists X \forall x(&\text{first}(x) \Rightarrow x \in X \wedge \\ &\text{last}(x) \Rightarrow x \in X \wedge \\ &\forall y(y = \text{succ}(x) \Rightarrow (x \in X \Leftrightarrow y \notin X)))\end{aligned}$$

Theorem 5.14 Trakhtenbrot, Büchi-Elgot theorem

Any language recognized by an NFA is MSO-definable.

Proof.
Given $\mathcal{A}(Q, \Sigma, \delta, I, F)$, write a sentence of the following shape:
$\exists X_1 ... \exists X_{|Q|}$(the X_i form a partition such that the automaton is in state i immediatly after reading the letter at position p on a successful run iff $p \in X_i$).

5.1.3 From an MSO-formula to a finite monoid

For any finite Σ-word w and any sequence of subsets $P_1, ..., P_m \subseteq |w|$, we define the marked Σ-word $(w, P_1, ..., P_m)$ as the word in the alphabet $\Sigma \times 2^{\{X_1, ..., X_m\}}$ which has at position p the letter (w_p, b_p) where $b_p(X_i) = 1$ iff $p \in P_i$.

Now any MSO formula $\varphi(X_1, ..., X_m)$ defines a language in the alphabet $\Sigma \times 2^n$:

$$L_\varphi = \{(w, P_1, ..., P_m) \in (\Sigma \times 2^m)^*\}$$

such that $w_1, P_1, ..., P_m \vDash \varphi$.

Proposition 5.15

For any MSO-formula φ there exists a homomorphism $h_\varphi : (\Sigma \times 2^m)^* \to M$ for M a finite monoid which recognizes L_φ.

Proof.
By induction on the complexity of φ:

- Atomic formulas: $X \subseteq a$ $(\forall x(x \in X \Rightarrow a(x)))$, $X \leqslant Y$ $(\forall x(x \in X \Rightarrow \forall y(y \in Y \Rightarrow x \leqslant y)))$, $X \subseteq Y$ $(\forall x \in X, x \in Y)$

  For $X_i \subseteq a$, let $h : (\Sigma \times 2^m) \to (\{0, 1\}, \wedge)$ be defined by $h((c, \bar{b})) = 0$ iff $b_i = 1$ and $c \neq a$.

  Then $L_{X_i \subseteq a} = h^{-1}(1)$.

- Induction step:

  - $\varphi = \neg\psi$: $h_\varphi := h_\psi$. Then $L_\varphi = \overline{L_\psi}$
  - $\varphi = \psi_1 \wedge \psi_2$: given $h_{\psi_i} : \tilde{\Sigma}^* \to M_i$ we construct $h_\varphi : \tilde{\Sigma}^* \to M_1 \times M_2$ by putting $h_\varphi(a) = (h_{\psi_1}(a), h_{\psi_2}(a))$. Then $w \vDash \varphi \Leftrightarrow h_\varphi(w) \in F_1 \times F_2$.
  - $\varphi = \exists X_n \psi(X_1, ..., X_n)$: by induction we have $h_\psi(\Sigma \times 2^n)^* \to M$ recognizing L_ψ, $F := h_\psi(L_\psi)$. Let $\mathcal{P}(M)$ be the powerset monoid of M with multiplication $A \cdot B := \{ab \mid a \in A, b \in B\}$. Now define $h_\varphi : (\Sigma \times 2^{n-1})^* \to \mathcal{P}(M)$ by $h_\varphi(a, \bar{b}) := \{h_\psi(a, \bar{b}0), h_\psi(a, \bar{b}1)\}$. Then

  $$h_\varphi((a_1, \bar{b}_1), ..., (a_m, \bar{b}_m)) = \{h_\psi((a_1, \bar{b}_1 c_1)...(a, \bar{b} c_m)) \mid c_1 ... c_m \in 2^m\}.$$

  Hence $h_\varphi(w) \cap F \neq \emptyset$ iff $w \vDash \exists X_n \psi$. So $F = \{A \in \mathcal{P}(M) \mid A \cap F \neq \emptyset\}$ recognizes L_φ.

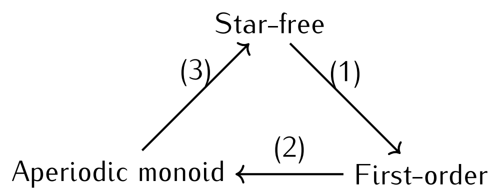
5.2 Logic fragments and classes of monoids

Theorem 5.16 **Schützenberger, Mcaughhton and Papert, Kamp theorem**

For any $L \subseteq \Sigma^*$, the following are equivalent:

1. L is recognized by a finite aperiodic monoid
2. L can be described by a starfree expression
3. L is definable in first order logic

3'. L is definable in linear temporal logic



Definition 5.17 **Subgroup of a semigroup**

A subgroup of a semigroup a subsemigroup which is a group.

Definition 5.18 **Subgroup of a monoid**

A subgroup of a monoid is a subgroup of the underlying semigroup.

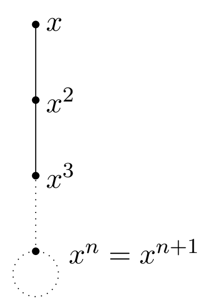
The subgroup of a monoid may have a different neutral element.

Definition 5.19 **Aperiodic monoid**

A monoid is aperiodic if all of its subgroups are trivial.

Fact 5.20

A finite semigroup is aperiodic iff $\exists n \geqslant 1, \forall x \in S, x^n = x^{n+1}$.



Definition 5.21 **Starfree expression**

A starfree expression E is built from $\emptyset$, $\{\varepsilon\}$ and singleton letters using concatenation and boolean operations.

5.2.1 From starfree to first order

The important lemma is the following.

Lemma 5.22

If L_1 and L_2 are FO-definable, then so is $L_1 a L_2$ for every $a \in \Sigma$.

Proof.
Let φ_i be a FO sentence defining L_i.
We define φ as $\exists x(a(x) \wedge \varphi_1^{<x} \wedge \varphi_2(x)^{>x})$ where $\psi^{<x}$ is an FO-formula with free variable x such that for any word w and $p \in |w|$, $p \vDash \psi^{<x}(x)$ iff $w_1, \ldots, w_{p-1} \vDash \psi$. Such an FO-formula exists by exercice 9 (induction on ψ).

5.2.2 First-order definable to aperiodic recognizable

Definition 5.23 Quantifier depth

The quantifier depth of an FO formula is the largest number of nestings of quantifiers.

We write, for marked structures A, $\bar{a} \in A^n$ and B, $\bar{b} \in B^n$, $A, \bar{a} \equiv_{n,k} B, \bar{b}$ iff the two satisfy the same FO formulas of $\mathrm{FO}_{n,k}$, i.e. with free variables among $x_1, \ldots, x_n$ of depth $\leqslant k$.

Theorem 5.24 Hintikka's theorem

For every n, k, the equivalence relation $\equiv_{n,k}$ has finite index, and each class can be described by a formula of $\mathrm{FO}_{n,k}$.

Proof sketch (see exercice 10).
For $k = 0$, $n \geqslant 0$, $\mathrm{FO}_{n,0}$ can describe a length n word plus a total ordering on the variables.
By induction, if a set $F_{n,k}$ is given for all $n \geqslant 0$, then we construct $F_{n,k+1} = \{\varphi(S) \mid S \subseteq F_{n+1,k}\}$, where

$$\varphi(S) = \bigwedge_{\psi \in S} \exists x \psi \wedge \forall x \left(\bigvee_{\psi \in S} \psi \right).$$

Lemma 5.25

Let $(u, \overline{p})$ and $(v, \overline{q})$ be marked Σ-words.
Then we have $u, \overline{p} \equiv_{n,k+1} v, \overline{q}$ iff for every $p \in |u|$ there exists $q \in |v|$ such that $u, \overline{p}p \equiv_{n+1,k} v, \overline{q}q$ and for every $q \in |v|$ there exists $p \in |u|$ such that $u, \overline{p}p \equiv_{n+1,k} v, \overline{q}q$.
This is like "Ehrenfeucht-Fraïssé games".

Proposition 5.26

For any words $u, v \in \Sigma^*$, if $u \equiv_{0,k} v$ then for any $\alpha \in \Sigma^*$, $\alpha u \equiv_{0,k} \alpha v$ and $u\alpha \equiv_{0,k} v\alpha$.

Proof.
$k = 0$: use the definitions and exercice 10.a.
By induction on k, assume $u \equiv_{0,k+1} v$. To show $\alpha u \equiv_{0,k+1} \alpha v$ we use Lemma 5.25: let $p \in |\alpha u|$.
If $p \leqslant |\alpha|$, choose $q = p$ and otherwise choose q so that $u, p \equiv_{1,k} v, q$.
Now by the correct induction hypothesis, $\alpha u, p \equiv_{1,k} \alpha v, q$. Then $\alpha u \equiv_{0,k+1} \alpha v$.

Definition 5.27 Congruence

An equivalence relation $\equiv$ on Σ^* is a congruence if $u \equiv v$ and $w \equiv x$ implies $uw \equiv vx$.

Proposition 5.28

For every $k \geqslant 0$, the equivalence relation $\equiv_{0,k}$ on Σ^* is a congruence.

Lemma 5.29 is exercice 1.c of sheet 2.

Lemma 5.29

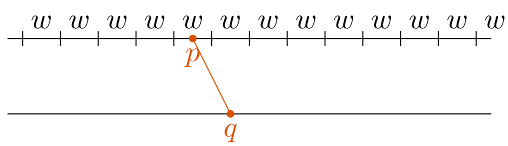
Two marked words $u, \overline{p}$ and $v, \overline{q}$ are $\equiv_{n,k}$-equivalent iff the markings look exactly the same, and the factors between marked positions are $\equiv_{0,k}$-equivalent.

Proposition 5.30 is exercice 11.a of sheet 1.

Proposition 5.30

For any $k \geqslant 1$ and finite word w, $w^{2^k-1} \equiv_{0,k} w^{2^k}$.

Proof.
The proof is by induction on k.
For $k = 1$, $w^2 \equiv_{0,1} w$, since w and w^2 contain the same letters. And this is all that $\mathsf{FO}_{0,1}$ can express.
For the induction step, let $n = 2^{k+1}$. To show that $w^n \equiv_{0,k+1} w^{n-1}$ it suffices by exercice 10 to prove that for every position $p \in |w^n|$ there is a position $q \in |w^{n-1}|$ with $w^n, p \equiv_{1,k} w^{n-1}, q$ and vice versa.



Suppose without loss of generality (by symmetry) that $p < |w|^{2^k}$.
Then choose $q := p$. The first block in w^n, p is equal to the first block in w^{n-1}, q, and the second blocks in w^n, p and w^{n-1}, q are of length $\geqslant |w|^{2^k-1}$, so by the induction hypothesis, they are $\equiv_{0,k}$-equivalent. Also p and q have the same letters.
So by Lemma 5.29, $w^n, p \equiv_{1,k} w^{n-1}, q$.

Corollary 5.31

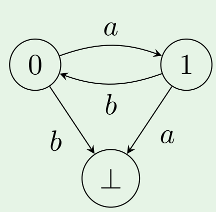
If L is $\mathsf{FO}_{0,k}$-definable, then L is recognized by the monoid $\Sigma^* / \equiv_{0,k}$, which is aperiodic by Proposition 5.30.

Proof.
If $\varphi \in \mathsf{FO}_{0,k}$ defines L then $L = \bigcup_{w \models \varphi} [w]_{\equiv_{0,k}}$.

5.2.3 Aperiodic to starfree

Example 5.32

$(ab)^*$ is a starfree language. An aperiodic monoid recognizing it an be obtained from the DFA



Theorem 5.33

If S is a finite aperiodic semigroup generated by $\Sigma \subseteq S$, then $L_S := \{w \in \Sigma^+ \mid (w)_S = s\}$ is starfree for all $s \in S$.

Proof.
The proof is by induction on the pair $(|S|, |\Sigma|)$, ordered lexicographicaly where $|S|$ takes precedence. When $|S| = 1$, the language is either $\emptyset$ or Σ^+.
When $|\Sigma| = 1$, then S is cyclic of the form $C_{c,d}$ for some $c, d \geqslant 1$ and S is aperiodic so $d = 1$. So $S = \{x, x^2, ..., x^{c-1}\}$ where $\Sigma = \{x\}$. The language L_{x^n} is $\{x^n\}$ if $n < c-1$, and $\Sigma^+ \setminus \{x, x^2, ..., x^{c-2}\}$ when $n = c-1$.

Assume $|S|$ and $|\Sigma| \geqslant 2$. We distinguish two cases

1. S is right simple: for every $s \in S$, $sS = S$. Then S is in fact right zero: $st = t$ for all $s, t \in S$.

   Indeed, for every $s \in S$, the multiplication function $\lambda_S : t \mapsto st$ is surjective, and thus a permutation, because S is finite. Since S is aperiodic, $S^n = S^{n+1}$, so $(\lambda_S)^n = (\lambda_S)^{n+1}$. So $\lambda_S = \mathrm{id}_S$. So for any $w \in \Sigma^+$, if $w = s_1, ..., s_\ell$, then $(w)_S = s_\ell$. So $L_s = \Sigma^* s$ for all $s \in \Sigma$.

2. S it not right simple. Then there exists a generator $a_0 \in \Sigma$ such that $a_0 S \subsetneq S$.

   We well now describe L_s as $A_s + B_s + \sum_{s_1 s_2 = s} A_{s_1} B_{s_2}$ where $A_s = \{w \in (\Sigma - a_0)^+ \mid (w)_S = s\}$ and $B_s = \{w \in a_0\Sigma^+ \mid (w)_S = s\}$.

   Note that $T := a_0 S$ is a subsemigroup of S and $|T| < |S|$. Take as alphabet $\Delta := T$. By induction we may obtain a starfree expression in the alphabet Δ for the language $C_S := \{v \in \Delta^+ \mid (v)_T = s\}$. In this expression, replace every occurrence of a letter $a_0 x \in \Delta$ by $\left(\sum_{a_0 y = a_0 x} a_0 E(y)\right)$ where $E(y)$ is a starfree expression in $\Sigma - a_0$ for the element y, which again exists by induction the alphabet size.

Also see Bojańczyk 2.2.2.

5.3 Green's relations and characterizations for $\mathcal{L}$-trivial, $\mathcal{J}$-trivial monoids

Definition 5.34 Prefix preorder

The prefix preorder on a finite monoid M is defined by $u \leqslant_{\mathcal{R}} v$ if there exists $\alpha \in M$ such that $u = v\alpha$, or equivalently, $uM \subseteq vM$.

Definition 5.35 Suffix preorder

The prefix preorder on a finite monoid M is defined by $u \leqslant_{\mathcal{L}} v$ if there exists $\alpha \in M$ such that $u = \alpha v$.

We write $u\mathcal{R}v$ for $u \leqslant_{\mathcal{R}} v$ and $v \leqslant_{\mathcal{R}} u$ and $u\mathcal{L}v$ for $u \leqslant_{\mathcal{L}} v$ and $v \leqslant_{\mathcal{L}} u$.

Definition 5.36 $\mathcal{H}$-preorder

The $\mathcal{H}$-preorder is defined by $u \leqslant_{\mathcal{H}} v$ if $u \leqslant_{\mathcal{L}} v$ and $u \leqslant_{\mathcal{R}} v$.

Example 5.37

M is aperiodic iff $\mathcal{H}$ is a partial order.

Example 5.38

If $M = (X^X, \circ)$ then $f \leqslant_{\mathcal{R}} g \Leftrightarrow \mathrm{Im}(f) \subseteq \mathrm{Im}(g)$ and $f \leqslant_{\mathcal{L}} g \Leftrightarrow \ker(g) \subseteq \ker(f)$.

Definition 5.39 $\mathcal{L}$-triviality

A monoid is $\mathcal{L}$-trivial if $x\mathcal{L}y$ implies $x = y$.

Theorem 5.40

For any language L, the following propositions are equivalent:

1. L is recognized by a finite $\mathcal{L}$-trivial monoid
2. L is a finite union of suffix-unambiguous languages, i.e. languages of the form $\Sigma_0^* a_1 \Sigma_1^* ... a_n \Sigma_n^*$ with $\Sigma_i \subseteq \Sigma - a_i$ and $\Sigma_0 \subseteq \Sigma$.

Remark 5.41

Suffix-unambiguous $\subsetneq$ starfree.

Definition 5.42 Left action

A left action of a monoid M on a set X is a homomorphism $\lambda : M \to X^X$.
When λ is fixed we write $m \cdot q = \lambda(m)(q)$ for $m \in M$ and $q \in X$.

The fact that λ is a homomorphism means that $1_M \cdot q = q$ and $m \cdot (n \cdot q) = (mn) \cdot q$.

Definition 5.43 Faithful

An action is faithful if λ is injective, i.e. if $m \neq n$ then there is $q \in X$ with $m \cdot q \neq n \cdot q$.

Example 5.44

M has a faithful action on the set M via $m \cdot n = mn$.

Proposition 5.45

Let M be a finite monoid. Then the following propositions are equivalent:

1. M is $\mathcal{L}$-trivial
2. There exists a faithful action of M on a post $(X, \leqslant)$ such that $m \cdot x \leqslant x$ for all $x \in X$
3. If e is idempotent in M and $e \leqslant_{\mathcal{L}} m$ then $me = e$

Proof.

1. $\Rightarrow$ 2. M acts faithfully on $(M, \leqslant_{\mathcal{L}})$, and $mn \leqslant_{\mathcal{L}} n$.

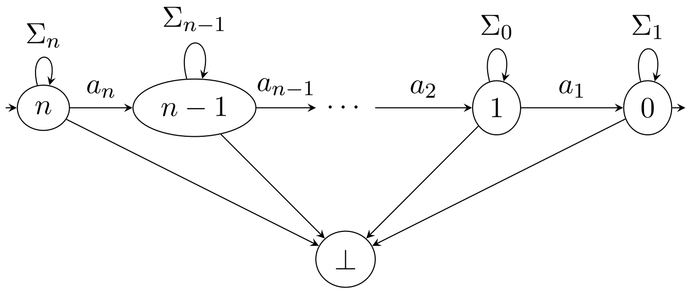
2. $\Rightarrow$ 3. Suppose e is idempotent and $e \leqslant_{\mathcal{L}} m$.
Write $e = sm$. Then for any $x \in X$, $me \cdot x \leqslant e \cdot x$, and also $e \cdot x = ee \cdot x = sme \cdot x \leqslant me \cdot x$.
Since $\leqslant$ is a partial order, $e \cdot x = me \cdot x$. Since the action is faithful, $e = me$.

3. $\Rightarrow$ 1. Suppose that $u \leqslant_{\mathcal{L}} v$ and $v \leqslant_{\mathcal{L}} u$.
Pick α such that $u = \alpha v$ and β such that $v = \beta u$.
So $u = \alpha v = \alpha\beta u$, so $u = (\alpha\beta)^n u$ for every n. Now pick n such that $(\alpha\beta)^n$ is idempotent.
Then $\beta(\alpha\beta)^n = (\alpha\beta)^n$ by 3.

So $u = (\alpha\beta)^n u = \beta(\alpha\beta)^n u = \beta u = v$.

Proof of Theorem 5.40.

2. $\Rightarrow$ 1. If L is suffix-unambiguous, then the following DFA recognizes L^{reverse}



The homomorphism $\overline{\delta} : \Sigma^* \to (Q^Q, \circ)$ recognizes L. Indeed, $\overline{\delta}(w) \cdot n = 0$ iff $w \in L$.
The image of $\overline{\delta}$ is a monoid M that acts faithfully on Q and $m \cdot q \leqslant q$ where $n > n-1 > ... > 0 > \perp$.
Thus M is $\mathcal{L}$-trivial by the proposition.

2. $\Rightarrow$ 1. We have to show that if M is a finite $\mathcal{L}$-trivial monoid, then for every $m \in M$, $\{w \in M^* \mid (w)_M = m\}$ is suffix unambiguous.
The idea is that if $w \in M^*$ is any word evaluating to some $m \in M$ then the "computation" of the product $(w)_M = w_1 \cdot w_2 \cdot ... \cdot w_n$ gives a decreasing chain in the $\mathcal{L}$-ordering on M:

$$1 \geqslant_{\mathcal{L}} w_n \geqslant_{\mathcal{L}} w_{n-1}w_n \geqslant_{\mathcal{L}} ... \geqslant_{\mathcal{L}} w_1 w_2 ... w_n = m,$$

so we need to describe the "computations" leading to m.

Write $\mathcal{F}_m$ for the set of strictly decreasing $\mathcal{L}$-chains in M with first element 1 and last element m. For any $\overline{q} = (q_0, ..., q_n) \in \mathcal{F}_n$, define the expression

$$E(\overline{q}) := \sum \{\Sigma_0^* a_1 \Sigma_1^* a_2 ... a_n \Sigma_n^* \mid a_i \in M, a_i q_{i-1} = q_i\}$$

where $\Sigma_i = \{a \in \Sigma \mid a q_i = q_i\}$.
Then (claim) if $w \in M^*$, write $\overline{q}(w)$ for the strict $\mathcal{L}$-chain obtained from $(1, w_n, w_{n-1}w_n, ..., w_1...w_n)$ by only keeping the first element among a block of equal elements, e.g. if $w = aabbbaaa$ and $1 = a = a^2 = a^3 [_{\mathcal{L}} ba^3 = b^2a^3 = b^3a^3 >_{\mathcal{L}} ab^3a^3 = a^2b^3a^3$ then $\overline{q}(w) = (1, ba^3, ab^3a^3)$ and w is in $E(\overline{q})$.
Therefore if $(w)_M = n$ then $w \in \bigcup_{\overline{q} \in \mathcal{F}_n} E(\overline{q})$. Let $w \in M^*$ such that $(w)_M = m$. Then $w \in E(\overline{q}(w))$.

Definition 5.46 $\mathcal{J}$**-preorder**

In a finite monoid M, we write $u \leqslant_{\mathcal{J}} v$ if there exists $x, y \in M$ such that $u = xvy$ and $u \mathcal{J} v$ if $u \leqslant_{\mathcal{J}} v \leqslant_{\mathcal{J}} u$.

Definition 5.47 $\mathcal{J}$**-triviality**

A finite monoid is called $\mathcal{J}$-trivial if $u \mathcal{J} v$ implies $u = v$.

Definition 5.48 **Subword**

A word $v = a_1 ... a_n \in \Sigma^*$ is a subword of $w \in \Sigma^*$ if $w \in \Sigma^* a_1 \Sigma^* ... \Sigma^* a_n \Sigma^*$.

Theorem 5.49 **Simon's theorem**

For any language $L \subseteq \Sigma^*$, the following propositions are equivalent:

1. L is recognized by a finite $\mathcal{J}$-trivial monoid
2. L is BC of sets upward closed in the subword ordering
3. L is definable by an FO formula with no quantifier alternation, i.e. a BC of existential FO-formulas.

Such a language is called piecewise testable.

6 Varieties and profiniteness

6.1 Varieties

For any finite alphabet Σ, the class $\mathrm{Reg}(\Sigma)$ of regular languages in the alphabet Σ is a Boolean algebra and for any $a \in \Sigma$, if $L \in \mathrm{Reg}(\Sigma)$, then both $a^{-1}L := \{w \in \Sigma^* \mid aw \in L\}$ and $La^{-1} = \{w \in \Sigma^* \mid wa \in L\}$ are in $\mathrm{Reg}(\Sigma)$.

Moreover, if $f : \Delta^* \to \Sigma^*$ is a homomorphism, then $f^{-1}(L)$ is in $\mathrm{Reg}(\Delta)$.

In short, $\mathrm{Reg} : \mathbf{FreeMon}_{fg}^{\mathrm{op}} \longrightarrow \mathbf{qBA}$ is a well-defined functor.

Definition 6.1 **Variety of regular languages**

A variety of regular languages is a subfunctor of Reg, i.e. an assignment $\mathcal{V}$ which assigns to every finite set Σ a Boolean subalgebra $\mathcal{V}(\Sigma)$ of $\mathrm{Reg}(\Sigma)$ such that for every $a \in \Sigma$ and $L \in \mathcal{V}(\Sigma)$, both $a^{-1}L$ and La^{-1} are in $\mathcal{V}(\Sigma)$ and for any hom. $f : \Delta^* \to \Sigma^*$, $f^{-1}(L) \in \mathcal{V}(\Delta)$.

Definition 6.2 **Variety of finite monoid**

A variety of finite monoids is a non-empty collection $\mathbb{V}$ of finite monoids such that

- for any $M_1, M_2 \in \mathbb{V}$, $M_1 \times M_2 \in \mathbb{V}$
- for any $M \in \mathbb{V}$, if $N \leqslant M$ is a submonoid, then $N \in \mathbb{V}$
- for any $M \in \mathbb{V}$, if there is a surjective homomorphism $M \twoheadrightarrow N$, then $N \in \mathbb{V}$.

Theorem 6.3 **Eidenberg's theorem**

For any variety of regular languages $\mathcal{V}$, define

$$\mu(\mathcal{V}) := \{M \text{ finite monoid} \mid \forall h : \Sigma^* \to M, \forall P \subseteq M, h^{-1}(P) \in \mathcal{V}(\Sigma)\}.$$

For any variety of finite monoids $\mathbb{V}$, define for any finite set Σ

$$\lambda(\mathbb{V})(\Sigma) = \{h^{-1}(P) \mid h : \Sigma^* \to M \text{ hom}, M \in \mathbb{V}, P \subseteq M\}.$$

Then

$$\mu : \text{Language Varieties} \rightleftarrows \text{Finite Monoid Varieties} : \lambda$$

Profinite equations

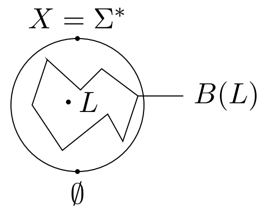
is a well-defined bijection.

Proof.
We will use syntactic monoids.

Proposition 6.4

Let $L \in \mathrm{Reg}(\Sigma)$. Then the Boolean algebra generated by the language $u^{-1}Jv^{-1}$, where u, v range over Σ^* is finite, and the equivalence relation induced by the atoms of $B(L)$ is a congruence on Σ^*.

Proof.



Concretely, $x \equiv_L y$ iff for all $u, v \in \Sigma^*$, $uxv \in L \Leftrightarrow uyv \in L$
This is called the syntactic congruence of L.

Suppose $\mathcal{A} = (Q, \Sigma, \delta, I, F)$ recognizes L. Then for any $a \in \Sigma$, the language $a^{-1}L$ is recognized by the automaton $\mathcal{A}' = (Q, \Sigma, \delta, I', F)$ where $I' = \{q \,|\, \exists i \in I, i \xrightarrow{a} q\}$ and La^{-1} is recognized by an automaton $\mathcal{A}'' = (Q, \Sigma, \delta, I, F')$...
Now a simple induction on $|u|+|v|$ shows that for any $u, v \in \Sigma^*$, $u^{-1}Lv^{-1}$ is recognized by some automaton of the form $(Q, \Sigma, \delta, I', F')$ for $I', F' \subseteq Q$. There are finitely many such automata so the set $\{u^{-1}Lv^{-1} \,|\, u, v \in \Sigma\}$ is finite, so $B(L)$ is finite.
If $x \equiv_L y$ and $w \in \Sigma^*$, then $xw \equiv_L yw$: if u, v are such that $u(xw)v \in L$.

Observe that μ is well-defined: if $M \overset{f}{\twoheadrightarrow} N$ and $h : \Sigma^* \to N$ is a homomorphism, where $M \in \mu(\mathcal{V})$, then for each $a \in \Sigma$, pick some $h'(a) \in M$ such that $f(h'(a)) = h(a)$.
Then $\overline{h'} : \Sigma^* \to M$ is a homomorphism and $f \circ \overline{h'} = h$ because both $f \circ \overline{h'}$ and h are homomorphisms $\Sigma^* \to N$ and they are equal on Σ. So for any $P \subseteq N$, $h^{-1}(P) = \overline{h'}^{-1}(f^{-1}(P))$, and this is in $\mathcal{V}(\Sigma)$ since $M \in \mu(\mathcal{V})$.
It's a submonoid: direct from the definitions.
It's a finite product: if $h : \Sigma^* \to M_1 \times M_2$ where $M_1, M_2 \in \mu(\mathcal{V})$, then both $\pi_1 \circ h$ and $\pi_2 \circ h$ are homomorphisms. So for any $(m_1, m_2) \in M_1 \times M_2$ we have $h^{-1}((m_1, m_2)) = (\pi_1 \circ h)^{-1}(m_1) \cap (\pi_2 \circ h)^{-1}(m_2)$, which is the intersection of two languages in $\mathcal{V}(\Sigma)$. Now for $P \subseteq M_1 \times M_2$, $h^{-1}(P) = \bigcup_{(m_1,m_2) \in P} h^{-1}((m_1, m_2))$, this is a finite union on languages in $\mathcal{V}(\Sigma)$.

λ is well-defined: for any Σ, $\lambda(\mathbb{V})(\Sigma)$ is a BA: if $h_1 : \Sigma^* \to M_1$, $h_2 : \Sigma^* \to M_2$, $P_1 \subseteq M_1$, $P_2 \subseteq M_2$, then $h_1^{-1}(P_1) \cap h_2 vv{-}1(P_2)$ is recognized by $h : w \mapsto (h_1(w), h_2(w))$ as it is equal to $h^{-1}(P_1 \times P_2)$.
Also, $\lambda(\mathbb{V})(\Sigma)$ is closed under quotients, and if $f : \Delta^* \to \Sigma^*$ and $L \in \lambda(\mathbb{V})(\Sigma)$ then $f^{-1}(L) \in \lambda(\mathbb{V})(\Delta)$.

To show λ, μ form a bijection, we show $\lambda\mu(\mathcal{V}) = \mathcal{V}$ and $\mu\lambda(\mathbb{V}) = \mathbb{V}$ for any varieties $\mathcal{V}, \mathbb{V}$.
Clearly, $\lambda\mu(\mathcal{V})(\Sigma) \subseteq (\mathcal{V}(\Sigma))$. For the other direction, we use the syntactic monoid.
Let $L \in \mathcal{V}(\Sigma)$. Then we define $M_L := \Sigma^* / \equiv_L$ is in $\mu(\mathcal{V})$. By Proposition 6.4, the languages recognized by $\begin{array}{ccc} \Sigma^* & \longrightarrow & M_L \\ u & \longmapsto & [u]_{\equiv_L} \end{array}$ are in $B(L)$, and therefore in $\mathcal{V}(\Sigma)$ because they are Boolean combination of languages $u^{-1}Lv^{-1}$, which are all in $\mathcal{V}(\Sigma)$. If $h : \Delta^* \to M_L$ with $[g(a)]_{\equiv_L} = h(a)$, then

$$\begin{tikzcd} \Delta^* \arrow[rr, "\overline{g}"] \arrow[dr, "h"'] & & \Sigma^* \arrow[dl, "s"] \\ & M_L & \end{tikzcd}$$

commutes so $h^{-1}(P) = \overline{g}^{-1}(s^{-1}(P))$ and $s^{-1}(P)$ is in $\mathcal{V}(\Sigma)$, and so is $\overline{g}^{-1}(K)$ for any $K \in \mathcal{V}(\Sigma)$. The crucial skip is to thow $\lambda(\mathbb{V})(\Sigma) = \{L \in \mathrm{Reg}(\Sigma) \mid \Sigma^*/\equiv_L \in \mathbb{V}\}$.

Lemma 6.5

If $h : \Sigma^* \to M$ recognizes a language L, then there exists a surjective homomorphism $f : m(h) \twoheadrightarrow M_L$ such that $f \circ h = s$.

$$\begin{tikzcd} & \Sigma^* \arrow[dl, "h"'] \arrow[dr, "s"] & \\ \mathrm{Im}(h) \arrow[rr, two heads, "f"'] & & M_L \end{tikzcd}$$

Proof.
For any $u, v \in \Sigma^*$, $u^{-1}Lv^{-1}$ is also recognized by h, because it is $h^{-1}(\{m \in M \mid h(u)mh(v) \in h(L)\})$. So the Boolean algebra $B(L)$ is contained in $\{h^{-1}(P), P \subseteq M\}$. In particular if $h(w) = h(w')$, $w \equiv_L w'$. This allows us to define the factorization f.

We now prove $\mu\lambda(\mathbb{V}) \subseteq \mathbb{V}$.

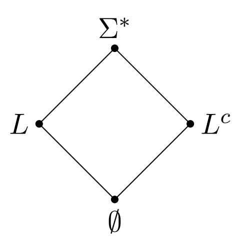
Let $M \in \mu\lambda(\mathbb{V})$, that is, for any $h : \Sigma^* \to M$, and $m \in M$, there is some monoid in $\mathbb{V}$ recognizing $h^{-1}(m)$. In particular, choose $h = \overline{\mathrm{id}} : \begin{array}{ccc} M^* & \longrightarrow & M \\ w & \longmapsto & (w)_M \end{array}$. For each $m \in M$, pick a monoid $M_m \in \mathbb{V}$, and $h_m : M^* \to M_m$ recognizing $L_m = h^{-1}(m)$. Define $f : M^* \to \prod_{m \in M} M_m$ by sending $m \in M$ to $\langle h_m(m) \rangle_{m \in M}$. Then if $f(u) = f(v)$, then $(u)_M = (v)_M$ (using the definition of f). Therefore, the monoid M is the homomorphic image of a submonoid of $\prod_{m \in M} M_m$, so it is in $\mathbb{V}$. (cf. Birkhoff's HSP theorem)

Exercise 6.6

Compute the syntactic monoid for $L = (aa)^*$, $L = \Sigma^* a \Sigma^*$ where $\Sigma = \{a, b, c\}$, and $L = (ab)^*$ for $\Sigma = \{a, b\}$.

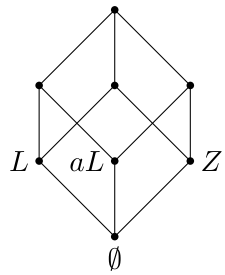
Proof.
For $L = \Sigma^* a \Sigma^*$, $a^{-1}L = \Sigma^* = La^{-1}$, $b^{-1} = L = Lb^{-1} = \ldots$
$x^{-1}(L^c) = (x^{-1}L)^c$

	0	1
0	0	0
1	0	1



For $L = (aa)^*$, $b^{-1}L = \emptyset$, $Lb^{-1} = \emptyset$, $a^{-1}L = La^{-1} = aL$
$Z = \Sigma^* \setminus (L \cup aL) = \Sigma^* b \Sigma^*$

	1	a	b
1	1	a	b
a	a	1	b
b	b	b	b



Theorem 6.7 Krohn-Rhocles' theorem

Every finite monoid can be obtained by H,S of an iterated semidirect product of groups and aperiodic monoids.

6.2 Profinite monoids and equations

Definition 6.8 Boolean space

A boolean space is a pair (X, B) where X is a set and B is a Boolean algebra of subsets of X such that

1. if $x \neq y$ then $\exists b \in B, x \in b, y \notin b$
2. if $X = \bigcup_{b \in S} b$ for some $S \subseteq B$, then there is a finite $F \subseteq S$ such that $X = \bigcup_{b \in F} b$

The sets in B are called "clopen", closed and open.

Example 6.9

1. If X is a finite set then $(X, \mathcal{P}(X))$ is the only Boolean space on X.
2. If $X = \mathbb{N} \cup \{\omega\}$, then $B = \{\text{finite subsets of } \mathbb{N}\} \cup \{\text{co-finite subsets of } X \text{ containing } \omega\}$ makes a Boolean space on X.
3. $(\mathbb{N}, \{\text{finite or cofinite } S \subseteq \mathbb{N})$ is not a Boolean space
4. With $B = \mathrm{Reg}(\Sigma^*)$, (Σ^*, B) is not compact.

Given a Boolean algebra B, can we always find a Boolean space (X, A) with $A \simeq B$?
Yes, we can! There is a Boolean space (X, A) with $A \simeq \mathrm{Reg}(\Sigma^*)$. Moreover, X is a monoid, known as the free profinite monoid on Σ which will allow us to define "profinite equations".

Definition 6.10 Profinite space

A topological space is profinite if it is a boolean space.

Definition 6.11 Profinite object

An object in a category is profinite if it is a limit of a directed diagram of finite objects.

Definition 6.12 Profinite monoid

A monoid is profinite if it is equipped with a Boolean topology such that the multiplication is continuous.

Fact 6.13

A topological monoid M is profinite iff for any $x \neq y \in M$ there exists a continuous homomorphism $h : M \to F$ with F a finite monoid, and $h(x) \neq h(y)$, iff M admits a continuous embedding into a product of finite monoids.

A class of potientially infinite monoids that is closed under H, S, P, potientially infinite can always be characterized by a set of equations (Birkhoff).

Theorem 6.14 **Reiterman's theorem**

A variety of finite monoids can always be characterized by a *profinite* equational theory.

Example 6.15

$\mathbb{A} = [\![x^\omega = x^{\omega+1}]\!]$
A finite monoid is aperiodic iff it "satisfies" $x^\omega = x^{\omega+1}$.
$\mathbb{L} = [\![(xy)^\omega = y(xy)^\omega]\!]$ (exercise 5 on TD 2)

Theorem 6.16 **Store's theorem**

For any Boolean algebra B, there exists a Boolean space (X, A) with $A \simeq B$, which is unique up to homeomorphism.

Proof sketch (see exercice 4 of sheet 3).
Let $X := \{h : B \to 2 \text{ a homomorphism}\}$.
For every $b \in B$, consider $\widehat{b} := \{h \in X \mid h(b) = 1\}$ and define $A := \{\widehat{b} \mid b \in B\}$.
Then (X, A) is a Boolean space as required.

In the last example, there is a function $\eta : \Sigma^* \to X$, defined by $w \mapsto \{x \in X \mid \}$.

Note that η is injective but not surjective. For example when $\Sigma = \{a\}$, let $x = \mathrm{Reg}(a^*) \to 2$ be the homomorphism defined by $x(L) = 1$ if there is an $n \in \mathbb{N}$ such that $a^m \in L$ for all $m \geqslant n$. Then $x \neq \eta(w)$ for any $w \in \Sigma^*$, because if w is of length n, then $\eta(w)$ sends $a^{n+1}\Sigma^*$ to 0, while x sends it to 1.

Definition 6.17 **Free profinite monoid**

The Boolean space (X, A) with $A \simeq \mathrm{Reg}(\Sigma^*)$ is called the free profinite monoid on Σ.

Proposition 6.18

Let (X, A) be a Boolean space with A the free profinite monoid on Σ.
There exists a unique continuous multiplication $\cdot : X^2 \to X$ such that $\eta : \Sigma^* \to X$ is a homomorphism.
Moreover, for any function $\Sigma \to M$ (a finite monoid), there is a unique continous homomorphism $\widehat{f} : X \to M$ such that the following diagram commutes.

$$\begin{tikzcd} X \arrow[r, "\widehat{f}"] & M \\ \Sigma \arrow[u, "\eta"] \arrow[ur, "f"'] & \end{tikzcd}$$

Remark

A function $f : (Y, C) \to (X, B)$ between Boolean spaces is continuous if $f^{-1}(b) \in C$ for every $b \in B$, given (X, B) and (Y, C) Boolean spaces, their product is defined as $(X \times Y, B \oplus C)$ where $B \oplus C$ is the Boolean algebra generated by $b \times c$ for $b \in B, c \in C$.

Prrof (sketch).
For a fixed $w \in \Sigma^*$, define $\lambda_w : X \to X$ to be the function sending an arbitrary $x \in X$ to $\lambda_w(x) : L \mapsto x(w^{-1}L) = \begin{cases} 1 & \text{if } x(w^{-1}L) = 1 \\ 0 & \text{otherwise} \end{cases}$.
Then $\lambda_w(\eta(v)) = \eta(wv)$, exercise, and λ_w is continuous. Now for $x \in X$, the function $\rho_x : \Sigma^* \to X$ defined by $\rho_x(w) = \lambda_w(x)$ extends uniquely to some $\overline{\rho_x} : X \to X$ by density. Now, multiplication may be defined by $y \cdot x = \overline{\rho_x}(y)$.
This is a continuous function $\cdot : X^2 \to X$ satisfying all the stated properties. (see JEP-MPRI Ch. XI or XII)

Notation 6.19

$\widehat{\Sigma^*}$ denotes the free profinite monoid on Σ.

Remark

$\widehat{\Sigma^*}$ is isomorphic, as a topological monoid, to a submonoid of $\prod\limits_{\substack{f:\Sigma^* \twoheadrightarrow M \\ M \text{ finite monoid} \\ f \text{ homomorphism}}} M$. Indeed, if $x \in \widehat{\Sigma^*}$, then for any $f : \Sigma^* \twoheadrightarrow M$, we get an element $x_f = \widehat{f}(x)$, and the sequence $(x_f)_{f:\Sigma^* \twoheadrightarrow M}$ uniquely determines x.

Proposition 6.20

For any $x \in \widehat{\Sigma^*}$, there is a unique idempotent element in the closure of $\{x^n \mid n \in \mathbb{N}\}$.

Remark

An element y is a Boolean space (X, B) is in the closure of a set $S \subseteq X$ if for every $b \in B$ that contains y, we have $b \cap S \neq \emptyset$.

Proof.
Let $x \in \widehat{\Sigma^*}$. Then such an element x^ω can be defined by posing $(x^\omega)_f := f(x)^{|M|!}$ for every $f : \Sigma^* \to M$ homomorphism. To prove that this again defined an element of $\widehat{\Sigma^*}$, one need to show that for any homomorphism : $M \to M'$ of finite monoids, $h((x^\omega)_f) = (x^\omega)_{h \circ f}$.

$$\begin{tikzcd} & \Sigma^* \arrow[dl, "f"'] \arrow[dr, "h \circ f"] & \\ M \arrow[rr, "h"'] & & M' \end{tikzcd}$$

Exercise 6.21

Using that a homomorphism preserves idempotents and the idempotent power of an element is unique in the finite monoid M'.

Example 6.22

If $\Sigma = \{a\}$, the element a^ω can be defined as $a^\omega(L) = 1$ iff $\exists n_0 \in \mathbb{N}, \forall n \geqslant n_0, a^n \in L$.

Definition 6.23 **Profinite equality**

A pair of elements of $\widehat{\Sigma^*}$ is called a profinite equality.
A finie monoid M is said to satisfy a profinite equality (s, t) if for any $f : \Sigma \to M$, $\widehat{f}(s) = \widehat{f}(t)$.

Fact 6.24

If E is a set of profinite equalities, then $\mathbb{V}_E := \{M \text{ a finite monoid} \mid \forall (s,t) \in E, M \text{ satisfies } (s,t)\}$ is a variety of finite monoids.

Proof of closure under $\mathcal{P}_{\text{fin}}$.
If M_1, M_2 satisfy E, let $f : \Sigma \to M_1 \times M_2$ be any function. Then for each $i = 1, 2$, $\pi_1 \circ f : \Sigma \to M_i$ is such hat $\widehat{\pi_i \circ f}(s) = \widehat{\pi_i \circ f}(t)$ for all $(s,t) \in E$. Both $\widehat{\pi_i \circ f}$ and $\pi_i \circ \widehat{f}$ are continuous monoid homomorphisms $\widehat{\Sigma^*} \to M_i$ which extends $\pi_i \circ f$, so they are equal. So we have $\pi_i \circ \widehat{f}(s) = \pi_i \circ \widehat{f}(t)$, and thus $\widehat{f}(s) = \widehat{f}(t)$.

Theorem 6.25 **Rieterman's theorem**

If $\mathbb{V}$ is a variety of finite monoids, then $\mathbb{V} = \mathbb{V}_E$, where $E := \{(s,t) \in \widehat{\Sigma^*} \mid \forall M \in \mathbb{V}, M \text{ satisfies } (s,t)\}$ and Σ is a countable infinite set.

Proof.
If $M \in \mathbb{V}$, then certainly M satisfies every equation in E. Suppose M is a finite monoid and M satisfies all equations in E.
Let f be any sujective function $\Sigma \twoheadrightarrow M$. Consider the set $D := (\widehat{f} \times \widehat{f})^{-1}(\Delta_M) = \{(s,t) \in \widehat{\Sigma^*}^2 \mid \widehat{f}(s) = \widehat{f}(t)\}$.
By assumption, $E \subseteq D$. For every pair $(s,t) \in \widehat{\Sigma^*}^2 \setminus D$ in particular we can pick a homomorphism $f_{(s,t)} : \Sigma \to M_{(s,t)}$ with $M_{(s,t)} \in \mathbb{V}$ such that $\widehat{f}_{(s,t)}(s) \neq \widehat{f}_{(s,t)}(t)$. For each such (s,t), the set $(\widehat{f}_{(s,t)} \times \widehat{f}_{(s,t)})^{-1}(\Delta^c_{M_{(s,t)}}) = \{(x,y) \mid \widehat{f}_{(s,t)}(x) \neq \widehat{f}_{(s,t)}(y)\}$ is clopen. Pick a finite subset, by compactness, of the covering $\{D, C_{(s,t)} \mid (s,t) \in \widehat{\Sigma^*}^2 \setminus D$ which is still covering $\widehat{\Sigma^*}^2$ and show that $\prod_{(s,t) \in F} M_{(s,t)}$ has a submonoid that maps onto M.

$$\begin{array}{ccc}
\text{Boolean algebras} & & \text{Boolean spaces} \\
A & \longmapsto & (X, A') \\
h : A_1 \xrightarrow{\text{hom}} A_2 & \longmapsto & f : (X_2, A_2') \xrightarrow{\text{cts}} (X_1, A_1')
\end{array}$$

6.3 Logics and profiniteness

A consequence of Theorem 6.25 is the following.

Theorem 6.26

Let $\mathbb{V}$ be a variety of finite monoids. Then for any alphabet Σ define

$$E_{\mathbb{V}}(\Sigma) = \{(u,v) \in \widehat{\Sigma^*} \mid \forall h : \widehat{\Sigma^*} \xrightarrow{\text{hom}} V \in \mathbb{V}, h(u) = h(v)\}$$

Then $\widehat{F}_{\mathbb{V}}(\Sigma) := \widehat{\Sigma^*}/E_{\mathbb{V}}(\Sigma)$ is a profinite monoid, which is relatively free for $\mathbb{V}$, that is that for any $h : \Sigma \to V \in \mathbb{V}$ there is a unique $\widehat{h} : \widehat{F}_{\mathbb{V}}(\Sigma) \to V$ extending h.
The BA of clopen sets of $\widehat{F}_{\mathbb{V}}(\Sigma)$ is isomorphic to the $\mathbb{V}$-recognizable sets in Σ^*.

Application

The profinite monoid $\widehat{F}_{\mathbb{A}}(\Sigma)$ is the topological monoid of characters of the Boolean algebra FO(Σ), because $\mathrm{Clopens}(\widehat{F}_{\mathbb{A}}(\Sigma)) \cong \{\mathbb{A}\text{-recognizable in } \Sigma^*\} = \text{FO}(\Sigma)$.

Example 6.27

Take $\Sigma = \{a\}$. Note that, if $x : \mathsf{FO}(\Sigma) \to \mathbf{2}$ is a character, then either $x(\{a^n\}) = 1$ for some n, or $x(\{a^n\}) = 0$ for every n. But then $x(L) = 1$ iff L is co-finite.

$$x_0 \quad x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_\infty$$

We use this characterization to study the top monoid $\widehat{F}_{\mathbb{A}}(\Sigma)$ using tools from logic, in particular the compactness theorem of FO logic:

Theorem 6.28 **Compactness Theorem**

If T is a set of sentences such that for any finite subset F ot T there is a word w such that $w \vDash \varphi$ for every $\varphi \in F$, there is a word w_T such that $w_T \vDash \varphi$ for all $\varphi \in T$.

But... "word" here is not necessarily a finite word!

Example 6.29

$$T = \left\{ \exists x_1, ... \exists x_n, \left(\bigwedge_{1 \leqslant i < j \leqslant n} (x_i \neq x_j) \right) \mid n \in \mathbb{N} \right\}$$

Definition 6.30 **Word**

A word on Σ is a linear[a] order W equipped with a coloring $(W_a)_{a \in \Sigma}$.

[a] total

Example 6.31

The FO-sentence $\forall x, \forall y, (x < y)$ holds in any word with domain $(\mathbb{N}, <)$ but not in any finite word.

Definition 6.32 **Pseudofiniteness**

A word W is pseudofinite if every FO-sentence φ that holds in W also holds in some finite word w.

Example 6.33

$\mathbb{N} + \mathbb{N}^{\mathrm{op}}$ labeled wth "a" everywhere is pseudofinite

$$a \quad a \quad a \quad a \quad a \quad \cdots \qquad \cdots \quad a \quad a \quad a \quad a \quad a$$

The pseudofinite words form a monoid under conatenation (just the Σ^*) and, if $U \equiv_{\mathsf{FO}} U'$ then $U \cdot V \equiv_{\mathsf{FO}} U'V$ and $V \cdot U \equiv_{\mathsf{FO}} V \cdot U'$, so the $\equiv_{\mathsf{FO}}$-classes of pseudofinite words also form a monoid.

Proposition 6.34

This monoid is isomorphic to $\widehat{F}_{\mathbb{A}}(\Sigma)$.

Example 6.35

With $W =$

$$a \quad a \quad a \quad a \quad a \quad \cdots \qquad \cdots \quad b \quad b \quad b \quad b \quad b$$

Then $W \vDash = \forall x(a(x) \to \exists y(x < y \wedge a(y))) \wedge \exists x(a(x))$ but not no finite word does.

Proposition 6.36

"Being pseudofinite" is not finitel axiomatizable, but a word is pseudofinite iff for every FO-formula $\varphi(x)$ the word satisfies $\mathrm{Last}_\varphi : \exists x, \varphi(x) \to (\exists x_0, \varphi(x_0) \wedge \forall y(y > x_0 \to \neg\varphi(y)))$.

Part III

Probabilistic automata and Markov chains (Amaury Pouly)

7 Probabilistic automata

7.1 Definition

Definition 7.1 Stochastic matrix

A matrix $M \in \mathbb{R}^{n \times m}$ is stochastic iff for each i,

- $\forall j, M_{ij} \in [0, 1]$
- $\sum_{j=1}^{m} M_{ij} = 1$

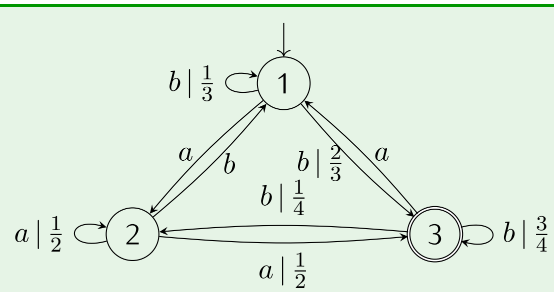
We extend the function μ: for $w \in A^*$, $\mu(w) = \mu(w_1)...\mu(w_{|w|})$ and $\mu(\varepsilon) = Id$.

Definition 7.2 Probabilistic automata

A probabilistic automata $\mathcal{A}$ is a tuple (A, Q, S, μ, T) where

- A is a finite alphabet
- Q is a finite set of states
- $S \in [0, 1]^{1 \times Q}$ is a stochastic vector (it sums at 1)
- $\mu(a) \in [0, 1]^{Q \times Q}$ is a transition stochastic matrix
- $T \in \{0, 1\}^{Q \times 1}$ is a column telling the accepting states

Example 7.3



- $A = \{a, b\}$
- $Q = \{1, 2, 3\}$
- $S = \begin{bmatrix} \frac{1}{3} & \frac{2}{3} & 0 \end{bmatrix}$
- $T = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

- $\mu(a) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} \\ 1 & 0 & 0 \end{bmatrix}$
- $\mu(b) = \begin{bmatrix} \frac{1}{3} & 0 & \frac{2}{3} \\ 1 & 0 & 0 \\ 0 & \frac{1}{4} & \frac{3}{4} \end{bmatrix}$

The probability that bb is accepted is
$1 \xrightarrow{b\,|\,\frac{1}{3}} 1 \xrightarrow{b\,|\,\frac{2}{3}} 3 = \frac{2}{9}$
$1 \xrightarrow{b\,|\,\frac{2}{3}} 3 \xrightarrow{b\,|\,\frac{3}{4}} 3 = \frac{1}{2}$
so the probability is $\frac{2}{9} + \frac{1}{2}$.

With matrices we have $S\mu(bb)T$.

Definition 7.4 **Stochastic language / Cut-point language**

Let $\mathcal{A}$ be a probablistic automaton and $\lambda \in [0, 1]$.
The stochastic language accepted by the automaton is

$$\mathcal{L}_{\mathcal{A}}(\lambda) = \{w \in A^* \mid \mathcal{A}(w) > \lambda\}.$$

There are variants of these languages: $\mathcal{L}^{\bowtie}_{\mathcal{A}}(\lambda) = \{w \in A^* \mid \mathcal{A}(w) \bowtie \lambda\}$, with $\bowtie \in \{<, \leqslant, =, >, \geqslant\}$.

7.2 Relation to regular language

Lemma 7.5

Every regular language L is stochastic, i.e. there exists a PA $\mathcal{A}$ such that $L = \mathcal{L}_{\mathcal{A}}(\lambda)$ for every $\lambda < 1$.

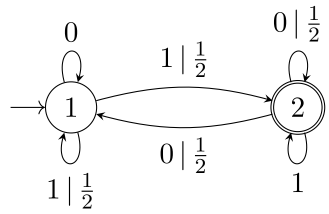
Proof.
Take a DFA and turn it into a PA with transition in $\{0, 1\}$. Then $\forall w \in A^*$, $\mathcal{A}(w) = 1$ iff $w \in L$. Thus $\forall \lambda < 1, \mathcal{L}_{\mathcal{A}}(\lambda) = L$.

Theorem 7.6

There are stochastic languages that are non regular.

Proof.
The idea is that we take $A = \{0, 1\}$ and we want $\mathcal{A}(w) = \sum_{i=1}^{|w|} w_i 2^{i-|w|-1}$.
Let $\mathcal{A} = (A, Q, S, \mu, T)$ with

- $A = \{0, 1\}$
- $Q = \{1, 2\}$
- $S = \begin{bmatrix} 1 & 0 \end{bmatrix}$
- $\mu(0) = \begin{bmatrix} 1 & 0 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$
- $\mu(1) = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 1 \end{bmatrix}$
- $T = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$



For every $w \in A^*$, define $[w] = \sum_{i=1}^{|w|} w_i 2^{i-|w|-1}$. For example $[10110] = \overline{0.01101}^2$.
Therefore $[\varepsilon] = 0$, $[w0] = \frac{[w]}{2}$ and $[w1] = \frac{1+[w]}{2}$.
We prove that $\mathcal{A}(w) = [w]$ by showing that $S\mu(w) = \begin{bmatrix} 1-[w] & [w] \end{bmatrix}$ by induction.

Then for any $\lambda < \mu$, $\mathcal{L}_\mathcal{A}(\lambda) \supsetneq \mathcal{L}_\mathcal{A}(\mu)$ because $\{[w] \mid w \in \mathcal{A}^*\}$ is dense in $[0,1]$. In particular there exists w such that $\lambda < [w] < \mu$. Then $w \in \mathcal{L}_\mathcal{A}(\lambda)$ and $w \notin \mathcal{L}_\mathcal{A}(\mu)$.

Therefore $\{\mathcal{L}_\mathcal{A}(\lambda) \mid \lambda \in [0,1]\}$ is uncountable, but there are countably many regular languages.

The proof still works if the transition probabilities are rational.

7.3 Universally non-regular languages

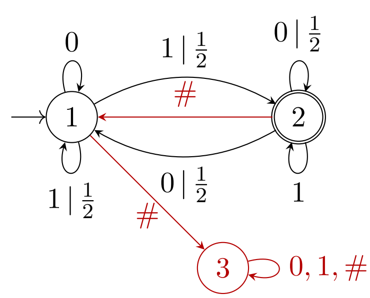
Remark 7.7

For any PA $\mathcal{A}$, $\{w \in A^* \mid \mathcal{A}(w) = 0\}$ is regular.

Theorem 7.8

There exists a PA $\mathcal{A}$ such that $\forall \lambda \in]0,1[$ and $\mathcal{L}_\mathcal{A}(\lambda)$ is not regular.

Proof.
We consider $A = \{0, 1, \#\}$, the same automaton $\mathcal{A}$ as in the proof of Theorem 7.6 and the following automaton $\mathcal{B}$.



Then for every $u, v \in \{0,1\}^*$, $\mathcal{B}(u\#v) = \mathcal{A}(u)\mathcal{A}(v)$ (proba $\mathcal{A}(u)$ for $1 \xrightarrow{u} 2$, 1 for $2 \xrightarrow{\#} 1$ and $\mathcal{A}(v)$ for $1 \xrightarrow{v} 2$).

Theorem 7.9 **Myhill-Nerode's theorem**

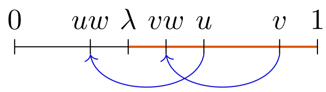
Let $L \subseteq A^*$ be a language.

Definition 7.10 **L-equivalence**

We say that $u, v \in A^*$ are L-equivalent, written $u \equiv_L v$ iff $\forall w \in A^*, uw \in L \Leftrightarrow vw \in L$.

Then L is regular iff the number of equivalence classes of A^* with respect to $\equiv_L$ is finite. And the number of states of a minimal DFA for L is the number of classes.

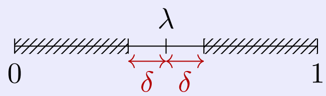
Fix $\lambda \in]0,1[$. Take $u, v \in A^*$ such that $\lambda < [u] < [v]$.
$\{[w] \mid w \in A^*\}$ is dense in $[0,1]$ so in particular there exists $w \in A^*$ such that $\frac{\lambda}{[u]} > [w] > \frac{\lambda}{[v]}$.
But then $\mathcal{B}(u\#w) = [u][w] < \lambda$ and $\mathcal{B}(v\#w) = [v][w]\lambda$, so $u\#w \not\equiv_{\mathcal{L}_\mathcal{A}(\lambda)} v\#w$.



7.4 Isolated cut-points

Definition 7.11 Isolated λ

We say that $\lambda \in [0,1]$ is isolated for $\mathcal{A}$ if $\exists \delta > 0, \forall w \in A^*, |\mathcal{A}(w) - \lambda| \geqslant \delta$.
δ can be called the isolation threshold.



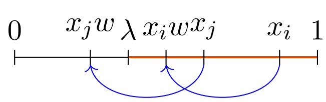
In particular, $\mathcal{L}_{\mathcal{A}}(\lambda) = \mathcal{L}_{\mathcal{A}}(\lambda + \varepsilon)$ for $|\varepsilon| \leqslant \delta$.

Theorem 7.12

If λ is isolated for $\mathcal{A}$, then $\mathcal{L}_{\mathcal{A}}(\lambda)$ is regular. Furthurmore, $\mathcal{L}_{\mathcal{A}}(\lambda)$ is recognized by a DFA with $(1 + \frac{r}{\delta})^{n-1}$ where δ is the isolation threshold, r is the number of final states and n is the number of states of $\mathcal{A}$.

Proof.
Let $\mathcal{A} = (A, Q, S, \mu, T)$. Without loss of generality, assume that $Q = \{1, ..., n\}$, only 1 is initial and only n is accepting.
Let $x_1, ..., x_k \in A^*$ such that $x_i \not\equiv_L x_j$ for $i \neq j$, where $L = \mathcal{L}_{\mathcal{A}}(\lambda)$.
Fix $i \neq j$, then by isolation threshold there exists $x \in A^*$ such that

- $x_i y \in L \implies \mathcal{A}(x, y) \geqslant \lambda + \delta$
- $x_j y \notin L \implies \mathcal{A}(x, y) \leqslant \lambda - \delta$



$\mathcal{A}(x_i, y) = S\mu(x_i)\mu(y)T$
Let $\xi_1^{(i)}, ..., \xi_n^{(i)} \in [0,1]$ be the first row of $\mu(x_i)$ and $\eta_1, ..., \eta_n \in [0,1]$ be the last column of $\mu(y)$. (y depends on i and j)

$$\mathcal{A}(x_i y) = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \xi_1^{(i)} & \cdots & \cdots & \xi_n^{(i)} \\ & & * & \end{bmatrix} \begin{bmatrix} & \eta_1 \\ & \vdots \\ * & \vdots \\ & \eta_n \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

Therefore $\mathcal{A}(x_i y) = \xi_1^{(i)}\eta_1 + ... + \xi_n^{(i)}\eta_n$ and $\mathcal{A}(x_j y) = \xi_1^{(j)}\eta_1 + ... + \xi_n^{(j)}\eta_n$, so

$$\mathcal{A}(x_i y) - \mathcal{A}(x_j y) = \sum_k (\xi_k^{(i)} - \xi_k^{(j)})\eta_k.$$

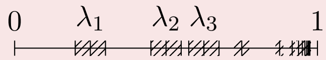
But $\mathcal{A}(x_i y) - \mathcal{A}(x_j y) \geqslant 2\delta$ so with $\|\cdot\|$ the 1-norm, $\|\xi^{(i)} - \xi^{(j)}\| \geqslant 2\delta$.
For $R > 0$, $p \in \mathbb{R}^n$, let $B_R(p) = \{x \in \mathbb{R}^n \mid \|x - p\| \leqslant R\}$. Then $B_\delta(\xi^{(i)}) \cap B_\delta(\xi^{(j)}) = \emptyset$. But $B_\delta(\xi^{(i)}) \subseteq B_{\frac{1}{2}+\delta}((0, ..., 0x))$ so by taking the volume of the union,

$$\sum_{i=1}^{k} \mathrm{vol}(B_\delta(\xi^{(i)})) \leqslant \mathrm{vol}(B_{\frac{1}{2}+\delta}((0, ..., 0))$$

so $k \cdot \delta^n \leqslant (\frac{1}{2} + \delta)^n$ and $k \leqslant (1 + \frac{1}{2\delta})^n$.
So the number of equivalence classes of $\equiv_L$ is bounded.

Theorem 7.13

There exists a PA $\mathcal{A}$ with two states and a sequence $(\lambda_n)_{n\in\mathbb{N}}$ of isolated threshold such that $\mathcal{L}_{\mathcal{A}}(\lambda_n)$ cannot be recognized by a DFA with $< n$ states.



Proof.
We are going to encode the set of cantor.
Let $\mathcal{A} = (A, Q, S, \mu, T)$ with

- $A = \{0, 2\}$
- $Q = \{1, 2\}$
- $S = \begin{bmatrix} 0 & 1 \end{bmatrix}$
- $\mu(0) = \begin{bmatrix} 1 & 0 \\ \frac{2}{3} & \frac{1}{3} \end{bmatrix}$
- $\mu(2) = \begin{bmatrix} \frac{1}{3} & \frac{2}{3} \\ 0 & 1 \end{bmatrix}$
- $\mu(2) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

For every $w \in A^*$, $\mathcal{A}(w) = \sum_{i=1}^{|w|} w_i 3^{i-|w|-1}$.
With $P = \{\mathcal{A}(w) \mid w \in A^*\}$, if $\lambda \in \overline{P}$ then λ is isolated. (P is not the Cantor set, only with finite expansion)

For $n \in \mathbb{N}$, let $\lambda_n = 0.\underbrace{2...2}_{n-1}11 = \sum_{i=1}^{n-1} 2 \cdot 3^{-i} + 3^{-n} + 3^{-n-1}$. Then $\lambda_n \notin \overline{P}$ and $2^n \in \mathcal{L}_{\mathcal{A}}(\lambda_n)$.

If $w \in A^*$ such that $|w| \leqslant n-1$ then $w \notin \mathcal{L}_{\mathcal{A}}(\lambda_n)$, so $\mathcal{A}(w) = \sum_{i=1}^{n-1} w_i 3^{i-|w|-1} < \lambda_n$.
So $\mathcal{L}_{\mathcal{A}}(\lambda_n)$ is non-empty and does not contain words of length $\leqslant n-1$, and therefore it cannot be recognized with less than n states.

7.5 Operations on PA

Given two PA $\mathcal{A}$ and $\mathcal{B}$, there are some operations we would like to compute.

- $1 - \mathcal{A}$
- $\alpha\mathcal{A} + (1-\alpha)\mathcal{B}$, for $\alpha \in [0, 1]$
- $\mathcal{A} \cdot \mathcal{B}$
- change the probability of one word (ε in particular)

Lemma 7.14

If $\mathcal{A}$ and $\mathcal{B}$ are PA on the same alphabet, and $\alpha \in [0, 1]$, then there exists $\mathcal{C}$ such that $\mathcal{C}(w) = \alpha\mathcal{A}(w) + (1-\alpha)\mathcal{B}(w)$.
$\mathcal{C}$ is denoted $\alpha\mathcal{A} + (1-\alpha)\mathcal{B}$.

Proof.

$$\begin{array}{cc} \downarrow \alpha & \downarrow 1-\alpha \\ \boxed{\mathcal{A}} & \boxed{\mathcal{B}} \end{array}$$

Lemma 7.15

If $\mathcal{A}$ is a PA, then there exists $\mathcal{C}$ such that $\mathcal{C}(w) = 1 - \alpha\mathcal{A}(w)$.
$\mathcal{C}$ is denoted $1 - \mathcal{A}$.

Proof.
We just swap the final states.

Lemma 7.16

For $\mathcal{A}$ and $\mathcal{B}$ PA on the same alphabet, there exists $\mathcal{C}$ such that $\mathcal{C}(w) = \mathcal{A}(w)\mathcal{B}(w)$.
$\mathcal{C}$ is denoted $\mathcal{A} \cdot \mathcal{B}$.

Proof.

$$(q, q') \xrightarrow{a \,|\, \mu(a)_{q \to p} \cdot \mu'(a)_{q' \to p'}} (p, p')$$

Let $\mathcal{A} = (A, Q, S, \mu, T)$ and $\mathcal{B} = (A, Q', S', \mu', T')$.
With $\otimes$ the Kroenecker product, such that $(M \otimes M')_{(i,i'),(j,j')} = M_{ij}M'i'j'$, define $\mu''(a) = \mu(a) \otimes \mu'(a)$, and this product satisfies $\mu''(w) = \mu(w) \otimes \mu'(w)$.

Lemma 7.17

If $\mathcal{A}$ is a PA and $p \in [0, 1]$, then there exists $\mathcal{C}$ such that $\mathcal{C}(w) = \begin{cases} \alpha\mathcal{A}(w) & \text{if } w \neq \varepsilon \\ p & \text{if } w = \varepsilon \end{cases}$.
$\mathcal{C}$ is denoted $\mathcal{A}[\varepsilon \leftarrow p]$.

Proof.
We add two initial state, one accepting with initial probability p and another with probability $1 - p$.

7.6 Decision problems

We only consider rational transition probabilities.

Definition 7.18 Strict emptiness problem

Given a PA $\mathcal{A}$ and $\lambda \in \mathbb{Q}$, decide whether $\mathcal{L}_{\mathcal{A}}(\lambda) \neq \emptyset$, i.e. $\exists w \in A^*, \mathcal{A}(w) > \lambda$?

Definition 7.19 Emptiness problem

Given a PA $\mathcal{A}$ and $\lambda \in \mathbb{Q}$, decide whether $\mathcal{L}^{\geqslant}_{\mathcal{A}}(\lambda) \neq \emptyset$, i.e. $\exists w \in A^*, \mathcal{A}(w) \geqslant \lambda$?

Definition 7.20 Universality problem

Given a PA $\mathcal{A}$ and $\lambda \in \mathbb{Q}$, decide whether $\mathcal{L}_{\mathcal{A}}(\lambda) = A^*$, i.e. $\forall w \in A^*, \mathcal{A}(w) \geqslant \lambda$?

Definition 7.21 Equality problem

Given a PA $\mathcal{A}$ and $\lambda \in \mathbb{Q}$, decide whether $\mathcal{L}^{=}_{\mathcal{A}}(\lambda) = A^*$, i.e $\exists w \in A^*, \mathcal{A}(w) = \lambda$.

All these problems are undecidable.
We will reduce these problems to PCP.

Definition 7.22 PCP

Given a finite alphabet A and $\phi_1, \phi_2 : A \to \{0,1\}^*$, decide whether $\exists w \in A, \phi_1(w) = \phi_2(w)$, i.e. $\phi_1(w_1)...\phi_1(w_{|w|}) = \phi_2(w_1)...\phi_2(w_{|w|})$.

Theorem 7.23

Equality problem is undecidable.

Proof.
Let ϕ_1, ϕ_2 be an instance of PCP.
Change ϕ_1, ϕ_2 into ψ_1, ψ_2 such that if $\phi_1(a) = w$ then $\psi_1(a) = w_1 1 w_2 1 ... 1 w_{|w|} 1$, and same for ϕ_2 and ψ_2.

Claim 7.24

$\forall w, \phi_1(w) = \phi_2(w) \Leftrightarrow \psi_1(w) = \psi_2(w)$

Like in the proof of Theorem 7.6, we can build $\mathcal{A}$ such that $\forall u \in \{0,1\}^*, \mathcal{A}(u) = [u]$ where $[u] = \sum_{i=1}^{|u|} u_i 2^{-i}$.
$\psi_1(w) \in \{01, 11\}^*$ so $[\cdot]$ is injective, i.e. $\forall w \in A^*, \psi_1(w) = \psi_2(w) \Leftrightarrow [\psi_1(w)] = [\psi_2(w)]$.
Fix $i \in \{1,2\}$, $\mathcal{A}_i = \langle A, Q, S, \mu_i, T \rangle$ with

- $Q = \{1,2\}$
- $S = \begin{bmatrix} 1 & 0 \end{bmatrix}$
- $\mu_i(a) = \begin{bmatrix} 2^{-[\psi_i(a)]} & [\psi_i(a)] \\ 0 & 1 \end{bmatrix}$
- $T = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Claim 7.25

$\mathcal{A}_i(w) = [\psi_i(w)]$

Claim 7.26

$\forall w \in A^*, \mu_i(w) = \begin{bmatrix} 2^{-[\psi_i(w)]} & [\psi_i(w)] \\ 0 & 1 \end{bmatrix}$

Proof.
By induction, $\mu_i(a)\mu_i(b) = \begin{bmatrix} 2^{-n} & [u] \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2^{-m} & [v] \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2^{-n-m} & 2^{-n}[v] + [u] \\ 0 & 1 \end{bmatrix}$.

Let $\mathcal{B} = \frac{1}{2}\mathcal{A}_1 + \frac{1}{2}(1 - \mathcal{A}_2) = \frac{1}{2} + \frac{1}{2}(\mathcal{A}_1 - \mathcal{A}_2)$.
For any $w \in A^*, \mathcal{B}(w) = \frac{1}{2}$ iff $\mathcal{A}_1(w) = \mathcal{A}_2(w)$ iff $[\psi_1(w)] = [\psi_2(w)]$ iff $\psi_1(w) = \psi_2(w)$.
Therefore $\mathcal{L}_{\mathcal{B}}^{=}(\frac{1}{2}) = \emptyset$ iff PCP on ψ_1, ψ_2 has no solution.
So equility in undecidable even for $\lambda = \frac{1}{2}$ and simple automata.

Definition 7.27 Simple automata

A PA $\mathcal{A}$ is simple if all probabilities in $\mathcal{A}$ are in $\{0, \frac{1}{2}, 1\}$.

Definition 7.28 Dyadic automata

A PA $\mathcal{A}$ is simple if all probabilities in $\mathcal{A}$ are in $\{[w] \,|\, w \in A^*\} \cup \{1\}$.

Example 7.29

$\{0, \frac{1}{8}, \frac{1}{4}, \frac{3}{8}, \frac{1}{2}, \frac{5}{8}, \frac{3}{4}, \frac{7}{8}, 1\}$

Exercise 7.30

Let $\mathcal{A}$ be dyadic. Show that there exists $\mathcal{B}$ simple such that $\{\mathcal{A}(w) \mid w \in A^*\} = \{\mathcal{B}(w) \mid w \in A^*\}$.

Proposition 7.31

Given $\mathcal{A}$, simple, one can compute $\mathcal{B}$ and $\mathcal{C}$ such that the following are equivalent

- $\exists w, \mathcal{A}(w) = \frac{1}{2}$
- $\exists w, \mathcal{B}(w) \geqslant \frac{1}{4}$
- $\exists w, \mathcal{C}(w) > \frac{1}{8}$

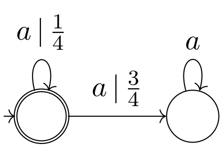
Proof.
Let $\mathcal{B} = \mathcal{A} \cdot (1 - \mathcal{A})$.

Claim 7.32

$\exists w, \mathcal{A}(w) = \frac{1}{2} \Leftrightarrow \exists w, \mathcal{B}(w) \geqslant \frac{1}{4}$

Proof.
The unique max of $x \mapsto x(x-1)$ is $x = \frac{1}{2}$.

Note that $\mathcal{B}$ only uses multiples of $\frac{1}{4}$ in its probabilities. Therefore, if $w \in A^*$, then either $\mathcal{B}(w) = \frac{1}{4}$ or $\mathcal{B}(w) \leqslant \frac{1}{4} - 4^{-|w|}$.
Observe that $\mathcal{B}(w) \geqslant \frac{1}{4}$ iff $\mathcal{B}(w) > \frac{1}{4} - 4^{-|w|}$ iff $\mathcal{B}(w) + 4^{-|w|} > \frac{1}{4}$ iff $\frac{1}{2}\mathcal{B}(w) + \frac{1}{2}4^{-|w|} > \frac{1}{8}$.
So if we build $\mathcal{D}$ such that $\mathcal{D}(w) = 4^{-|w|}$ we have $\mathcal{C} = \frac{1}{2}\mathcal{B} + \frac{1}{2}\mathcal{D}$.



Corollary 7.33

Strict emptyness, emptyness and universality problems are undecidable, even for simple automata.

7.6.1 Isolation problem

Definition 7.34 **Isolation problem**

Given $\mathcal{A}$ and λ, decide whether λ is isolated for $\mathcal{A}$.

Theorem 7.35

The isolation problem is undecidable.

Here we are going to use the infinite variant of PCP.

Definition 7.36 ω-PCP

Given A and $\phi_1, \phi_2 : A \to \{0,1\}^*$, decide whether there exists $w \in A^\omega$ such that $\phi_1(w) = \phi_2(w)$, i.e. $\forall i \in \mathbb{N}, (\phi_1(w))_i = (\phi_2(w))_i$.

If there is a finite solution, it can be repeated to form an infinite solution, but an infinite solution does not imply a finite one.

Theorem 7.37

ω-PCP is also undecidable.

Lemma 7.38

Let $\phi_1, \phi_2 : A \to \{0,1\}^*$. Assume that ϕ_1, ϕ_2 has no ω-solution. Then there exists $n_0 \in \mathbb{N}$ such that $\forall w \in A^* \cup A^\omega, \exists i \leqslant n_0, (\phi_1(w))_i \neq (\phi_2(w))_i$.

Proof.
Consider a tree labelled by pair of words such that

- The root is labelled by ε, ε
- The node u, v has a son $u\phi_1(a), v\phi_2(a)$ iff $u\phi_1(a)$ and $v\phi_2(a)$ do not differ, i.e. $\forall i \leqslant \min(|u\phi_1(a)|, |v\phi_2(a)|), (u\phi_1(a))_i = (v\phi_2(a))_i$.

Since there is no ω-solution to ϕ_1, ϕ_2, the tree has no infinite branch. Hence by König's lemma the tree is finite.

Proof of Theorem 7.35.
Let $\phi_1, \phi_2 : A \to \{0,1\}^*$. Modify into ψ_1, ψ_2 such that $\psi_i(w) \in \{0,1\}^*1$.

Claim 7.39

ϕ_1, ϕ_2 has a ω-solution iff ψ_1, ψ_2 has one.

Build $\mathcal{A}_i$ such that $\mathcal{A}_i(w) = [\psi_i(w)]$. Let $\mathcal{C} = (\frac{1}{2}\mathcal{A}_1 + \frac{1}{2}(1 - \mathcal{A}_2))[\varepsilon \leftarrow 0]$.

Claim 7.40

$\frac{1}{2}$ is isolated iff ϕ_1, ϕ_2 has no ω-solution.

Proof.
$\Longrightarrow$: if there is a ω-solution $w \in A^\omega$, then $\forall i \in \mathbb{N}$ let $u^{(i)} = w_1 \dots w_i \in A^*$. $\mathcal{C}(u^{(i)}) = \frac{1}{2} + \frac{1}{2}(\mathcal{A}_1(u^{(i)}) - \mathcal{A}_2(u^{(i)}))$, so $|\mathcal{C}(u^{(i)}) - \frac{1}{2}| \leqslant 2^{-i} \to 0$.
$\Longleftarrow$: if ϕ_1, ϕ_2 has no ω-solution, by Lemma 7.38, there exists n_0 such that the difference must be in the first n_0. Let $w \in A^*$. $|\mathcal{C}(w) - \frac{1}{2}| = \frac{1}{2}|[\psi_1(w)] - [\psi_2(w)]|$. Write $\psi_1(w) = x01y$ and $\psi_2(w) = x11z$ with $|x| \leqslant n_0 - 1$. Then

$$\begin{aligned}
|[x01y] - [x11z]| &= |2^{-|x|}[0] + 2^{-|x|-2}[y] - 2^{-|x|}[1] - 2^{-|x|-2}[z] \\
&= 2^{-|x|}| - \frac{1}{2} + \frac{1}{4}([y] - [z])| \\
&\geqslant 2^{-|x|}(\frac{1}{2} - \frac{1}{4}) \\
&\geqslant \frac{1}{4}2^{-n_0+1}
\end{aligned}$$

So $\frac{1}{2}$ is isolated.

7.6.2 Value problem

Definition 7.41

Let $\mathcal{A}$ be a PPA. $\mathrm{val}(\mathcal{A}) = \sup\{\mathcal{A}(w) \mid w \in A^*\}$

Definition 7.42 **Value problem**

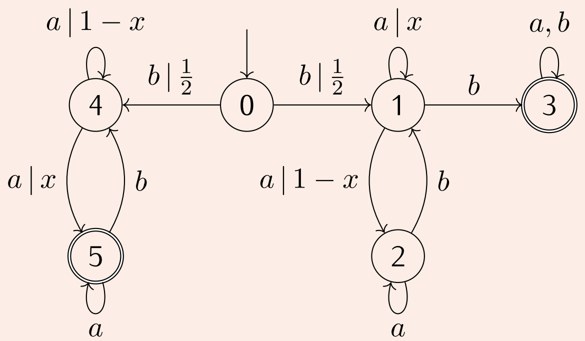
Given $\mathcal{A}$ and λ, decide whether $\mathrm{val}(\mathcal{A}) \geqslant \lambda$, i.e. $\forall \varepsilon > 0, \exists w \in A^*, \mathcal{A}(w) \geqslant \lambda - \varepsilon$.

Remark 7.43

Previous results show that this is undecidable for any $\lambda \in]0,1[$.

Proposition 7.44

Consider the automaton $\mathcal{A}_x$:



Then $\mathrm{val}(\mathcal{A}_x) = 1$ iff $x > \frac{1}{2}$, and otherwise $\mathrm{val}(\mathcal{A}_x) = \frac{1}{2}$.

Proof.

Let $n \in \mathbb{N}$. If we are not in state 4 then we cannot reach 4 by reading a^k.

$\mathcal{A}_x(4 \xrightarrow{a^n} 4) = (1-x)^n$

$\mathcal{A}_x(4 \xrightarrow{a^n b} 4) = 1 - (1-x)^n$

Let $n_0, ..., n_k$.

Claim 7.45

$$\mathcal{A}_x(1 \xrightarrow{a^{n_0}b\dots ba^{n_k}} 3) = 1 - \prod_{i=0}^{k-1}(1-x_i)$$

$$\mathcal{A}_x(4 \xrightarrow{a^{n_0}b\dots ba^{n_k}} 5) = \prod_{i=0}^{k}(1-(1-x)^{n_i})$$

Proof.
If $x > \frac{1}{2}$, and $\exists n \in \mathbb{N}$ such that $n_i = n$, $k = 2^n - 1$.

$$\begin{aligned}\mathcal{A}_x(1 \xrightarrow{a^{n_0}b\dots ba^{n_k}} 3) &= 1-(1-x^n)^k\\ &= 1-e^{k\log(1-x^n)}\\ &\geqslant 1-e^{-kx^n}\\ &= 1-e^{-(2^n-1)x^n}\\ &= 1-e^{-(2x)^n-x^n}\end{aligned}$$

If $\frac{1}{2} < x < 1$, $x^n \to 0$ ad $(2x)^n \to +\infty$ so $\mathcal{A}_x(1 \xrightarrow{a^{n_0}b\dots ba^{n_k}} 3) \to 1$.

$$\begin{aligned}\mathcal{A}_x(4 \xrightarrow{a^{n_0}b\dots ba^{n_k}} 5) &= (1-(1-x)^n)^{k+1}\\ &= e^{(k+1)\log(1-(1-x)^n)}\\ &\geqslant e^{2^n(-2(1-x)^n)}\\ &= e^{-2(2-2x)^n} \to 1\end{aligned}$$

So $\mathcal{A}((ba^n)^{2^n-1}) = \frac{1}{2}\mathcal{A}_x(4\to 5) + \frac{1}{2}\mathcal{A}_x(1\to 3) \to 1$.

If $x \leqslant \frac{1}{2}$, any word w accepted is of the form $\underbrace{ba^{n_0}b\dots ba^{n_k}}_{u}$ with $n_i \in \mathbb{N}$.

$$\begin{aligned}\mathcal{A}_x(w) &= \frac{1}{2}\mathcal{A}_x(1 \xrightarrow{u} 3) + \frac{1}{2}\mathcal{A}_x(4 \xrightarrow{u} 5)\\ &= \frac{1}{2} - \frac{1}{2}\prod_{i=0}^{k-1}(1-x^{n_i}) + \frac{1}{2}\prod_{i=0}^{k}(1-(1-x)^{n_i})\\ &\leqslant \frac{1}{2}\end{aligned}$$

$x^{n_i} \leqslant (1-x)^{n_i}$ so $1-x^{n_i} \geqslant 1-(1-x)^{n_i}$

$$\begin{aligned}\prod_{i=0}^{k-1}(1-x^{n_i}) &\geqslant \prod_{i=0}^{k-1}(1-(1/x)^{n_i})\\ &\geqslant \prod_{i=0}^{k}(1-(1-x)^{n_i})\end{aligned}$$

because $1-(1-x)^{n_k} \leqslant 1$.
Therefore $\mathrm{val}(\mathcal{A}_x) \leqslant \frac{1}{2}$.

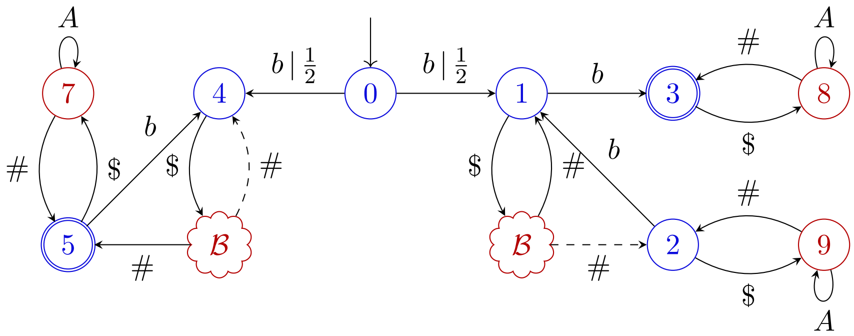
Theorem 7.46

The value 1 problem[a] is undecidable.

[a] Value problem with $\lambda = 1$

Proof.
Recall that the problem "Given $\mathcal{B}$, decide whether $\exists w, \mathcal{B}(w) > \frac{1}{2}$" is undecidable. We will replace the x of $\mathcal{A}_x$ by some value $B(w)$.

Let $\mathcal{B}$ over A, with $b, \#, \$ \notin A$. Let $\mathcal{C}$ be the following automata.



The alphabet is $A' = \{b, \$, \#\} \cup A$.

Claim 7.47

Formally, any accepted word is $v = bu_1b...bu_k$, with $u_i = \$w_{i1}\#...\$w_{in_i}\#$ where $w_ij \in A^*$.

Proof.
With 0, 1, 2, 3, 4 and 5 the type A states and 7, 8, 9, and copies of $\mathcal{B}$ the type B states,

1. From a type A state, only $b_i, \$, \#$ are valid.
2. From a type A state, only $\$$ leads to a $\mathcal{B}$.
3. From a type B, only $A \cup \{\#\}$.
4. From a type B, only $\#$ leads to a numbered state.

- If $\exists w \in A^*, \mathcal{B}(w) > \frac{1}{2}$, let $x = \mathcal{B}(w)$, $n_0, ..., n_k \in \mathbb{N}$ and $v = b(\$w\#)^{n_0}b...b(\$w\#)^{n_k}$, then $\mathcal{C}(v) = \mathcal{A}_x(ba^{n_0}b...ba^{n_k})$.

  But $\text{val}(\mathcal{A}_x) = 1$ when $x > \frac{1}{2}$, so for any $\varepsilon > 0, \exists n_0, ..., n_k, \mathcal{C}(v) \geqslant 1 - \varepsilon$. Hence $\text{val}(\mathcal{C}) = 1$.

- If $\forall w \in A^*, \mathcal{B}(w) \leqslant \frac{1}{2}$, then for any $v = bu_0b...bu_k$, with $u_i = \$w_{i1}\#...\$w_{in_i}\#$,

$$C(v) = \frac{1}{2} + \frac{1}{2}\prod_{i=0}^{k-1}(1 - \prod_{j=1}^{n}(1 - \mathcal{B}(w_{ij}))) - \frac{1}{2}\prod_{i=0}^{k}(1 - \prod_{j=1}^{n}\mathcal{B}(w_{ij})).$$

  Let $x = \max_{i,j} \mathcal{B}(w_{ij})$. Then $\mathcal{B}(w_{ij}) \leqslant x$. So

$$\prod_{j=1}^{n_i}\mathcal{B}(w_{ij}) \leqslant x^{n_i}$$

$$1 - \prod_{j=1}^{n_i}\mathcal{B}(w_{ij}) \geqslant 1 - x^{n_i}$$

$$-\prod_{i}\left(1 - \prod_{j=1}^{n_i}\mathcal{B}(w_{ij})\right) \leqslant -\prod_{i}(1 - x^{n_i})$$

  and

$$1 - \mathcal{B}(w_{ij}) \geqslant 1 - x$$

$$\prod_j (1 - \mathcal{B}(w_{ij})) \geqslant (1-x)^{n_i}$$

$$1 - \prod_j (1 - \mathcal{B}(w_{ij})) \leqslant 1 - (1-x)^{n_i}$$

$$\prod_i \left(1 - \prod_j (1 - \mathcal{B}(w_{ij}))\right) \leqslant \prod_i (1 - (1-x)^{n_i})$$

So

$$\begin{aligned} C(v) &\leqslant \frac{1}{2} + \frac{1}{2}\prod_i (1-(1-x)^{n_i}) - \prod_i (1 - x^{n_i}) \\ &= \mathcal{A}_x(ba^{n_0}b..ba^{n_k}) \\ &\leqslant \frac{1}{2}. \end{aligned}$$

So given $\mathcal{B}$, we have built $\mathcal{C}$ such that

$$\mathrm{val}(\mathcal{C}) = \begin{cases} 1 & \text{if } \mathcal{L}_{\mathcal{B}}(\frac{1}{2}) \neq \emptyset \\ \frac{1}{2} & \text{otherwise.} \end{cases}$$

Theorem 7.48

There is no algorithm such that, given $\mathcal{A}$,

- if $\mathrm{val}(\mathcal{A}) = 1$ then, the algorithm outputs "yes"
- if $\mathrm{val}(\mathcal{A}) \leqslant \frac{1}{2}$, then the algorithm outputs "no"
- otherwise the algorithm can output anything or not terminate.

Proof.
If there was such an algorithm, applying it to the previous construction (which always satisfies $\mathrm{val}(\mathcal{A}) = 1$ or $\mathrm{val}(\mathcal{A}) \leqslant \frac{1}{2}$), then it would decide whether a given $\mathcal{B}$ satisfies $\mathcal{L}_{\mathcal{B}}(\frac{1}{2})$. But this is undecidable.

Corollary 7.49

Deciding whether $\mathrm{val}(\mathcal{A}) > \lambda$ (resp. $\geqslant \lambda$), for a given $\mathcal{A}$ and $\lambda > 0$ fixed, is undecidable.

Proof.

- If $1 > \lambda \geqslant \frac{1}{2}$, then apply the previous construction, i.e. since $\mathrm{val}(\mathcal{A}) \in \{\frac{1}{2}, 1\}$, $\mathrm{val}(\mathcal{A}) = 1 \Leftrightarrow \mathrm{val}(\mathcal{A}) > \frac{1}{2}$.
- If $\lambda \geqslant \frac{1}{2}$, then assume this is decidable. Let $k \in \mathbb{N}$ such that $2^k\lambda \in]\frac{1}{2}, 1]$.

  Consider the algorithm for $\mathcal{A}$. Build $\mathcal{B}$ such that $B(w) = 2^{-k}\mathcal{A}(w)$. Run the algorithm on $\mathcal{B}$.

  Then the algorithm decides whether $\mathrm{val}(\mathcal{B}) \geqslant \lambda$ but $\mathrm{val}(\mathcal{B}) = 2^{-k}\mathrm{val}(\mathcal{A})$, so we have decided whether $\mathrm{val}(\mathcal{A}) \geqslant 2^k\lambda \in]\frac{1}{2}, 1]$. But this is impossible by previous case.

7.6.3 Decidable problem

The questions concerning the structure of the automata are in general decidable (these are weighted automata), but the questions on the stochastic languages are undecidable.

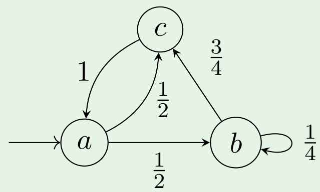
There are subclasses of PA such that most problems are decidable (leaktight). The idea is that we can make arbitrarly large loops of probability arbitrarly close to 1.

8 Markov chains and linear dynamical systems

Definition 8.1 Markov chain

A MC is a PA over the alphabet $\{a\}$.

Example 8.2



A MC $\mathcal{M}$ satisfies $\mathcal{M}(n) = SA^nT$ for some S line, A block and T column.

- Markov chain: stochastic system
- Linear dynamical system (LDS): SA^nT where the coefficients are in $\mathbb{Q}$ (not necessarily in $[0, 1]$)

  We have that MC $\subseteq$ LDS
- Linear recurrent sequences (LRS): a sequences $(u_n)_n \in \mathbb{Q}^{\mathbb{N}}$ such that $u_{n+k} = a_{k-1}u_{n+k-1} + ... + a_0u_n$ for some $a_0, ..., a_{k-1} \in \mathbb{Q}$. k is called the rank.

  A LRS in integer if $u_n \in \mathbb{N}$, $a_k \in \mathbb{N}$.

Example 8.3

$f_0 = 1, f_1 = 1, f_{n+2} = f_{n+1} + f_n$.

$V_n = \begin{bmatrix} f_n \\ f_{n+1} \end{bmatrix}$, $V_{n+1} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} f_n \\ f_{n+1} \end{bmatrix}$

$V_n = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}^n \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

Theorem 8.4 Cayley-Hamilton's theorem

Let $A \in \mathbb{C}^{d\times d}$, $p(\lambda) = \det(\lambda I_d - A)$ then $p(A) = 0$.
In particular, A^d is a linear combination of $I_d, A, A^2, ..., A^{d-1}$.

Proposition 8.5

Let $\langle S, A, T\rangle$ be a LDS. Then $(SA^nT)_n$ is a LRS.
Furthermore, if S,A,T have integer coefficients, then it is an integer LRS.

Conversly, if $(u_n)_n$ is a LRS, then there exists $\langle S, A, T\rangle$ a LDS such that $u_n = SA^nT$.
Furthermore, if $(u_n)_n$ is integer, then S, A, T have integer coefficients. In all cases, the dimension of A is the rank of the LRS.

Proof.
If $(u_n)_n$ is a LRS, then $u_{n+k} = a_{k-1}u_{n+k-1} + ... + a_0u_n$.

Let $V_n = \begin{bmatrix} u_n \\ u_{n+1} \\ \vdots \\ u_{n+k-1} \end{bmatrix} \in \mathbb{Q}^k$. Define

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & 1 \\ a_0 & a_1 & \cdots & \cdots & a_{k-1} \end{bmatrix} \in \mathbb{Q}^{d \times d}.$$

We have that $\forall n, V_{n+1} = AV_n$. Therefore, $SA^nT = u_n$ for $S = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}$ and $T = V_0$.

Let $\langle S, A, T \rangle$ be a LDS. By Cayley-Hamilton, A^d is a linear combination of $I_d, A, ..., A^{d-1}$, so there exists $a_0, ..., a_{d-1} \in \mathbb{Q}$ such that $A^d = a_0 I_d + ... + a_{d-1} A^{d-1}$.
Let $u_n = SA^nT$. Then $u_{n+d} = SA^nA^dT = \sum_{i=0} a_i \underbrace{SA^nA^iT}_{=u_{n+i}}$.

Attention

$(u_n)_n \to V_n = \begin{bmatrix} u_n \\ u_{n+1} \\ \vdots \\ u_{n+k-1} \end{bmatrix}$, $V_{n+1} = AV_n$.

$\exists S, A, T, u_n = SA^nT$ but $A^nT \neq \begin{bmatrix} u_n \\ u_{n+1} \\ \vdots \\ u_{n+k-1} \end{bmatrix}$.

Proposition 8.6

If $(u_n)_n, (v_n)_n$ are LRS, then $(u_n + v_n)_n, (u_n v_n)_n$ are also LRS, and $\forall \lambda, (\lambda u_n)_n$ is a LRS.

Example 8.7

With Fibonacci, $f_n = \frac{\phi^n + \overline{\phi}^n}{\sqrt{5}}$.

Proposition 8.8

Let $(u_n)_n$ be a LRS, $u_{n+k} = \sum a_i u_{n+i}$. Let $p(\lambda) = x^k - \sum a_i \lambda^i$. Let $\lambda_1, ...\lambda_k$ be the complex roots of p.
Then $\exists A_1, ..., A_k \in \mathbb{C}[X]$ of degree $\leqslant k$ such that $u_n = A_1(n)\lambda_1^n + ... + A_k(n)\lambda_k^n$.

Example 8.9

For Fibonacci, $k = 2$, $\lambda_1 = \phi$, $\lambda_2 = \overline{\phi}$, $A_1(n) = 1$, $A_2(n) = 1$, $u_n = (1 + \frac{n}{3})\phi^n - n\overline{\phi}^n$.

Definition 8.10 **Markov inequality problem**

Given a MC $\mathcal{M}$ and $\lambda \in \mathbb{Q}$, decide $\exists n \in \mathbb{N}, \mathcal{M}(n) \geqslant \lambda$.

Definition 8.11 **Markov equality problem**

Given a MC $\mathcal{M}$ and $\lambda \in \mathbb{Q}$, decide $\exists n \in \mathbb{N}, \mathcal{M}(n) = \lambda$.

Definition 8.12 **Skolem problem**

Given a LRS $(u_n)_n$, decide whether $\exists n, u_n = 0$.

Definition 8.13 **Positivity problem**

Given a LRS $(u_n)_n$, decide whether $\forall n, u_n > 0$.

We will prove that Skolem (even for integer LRS) is equivalent to Markov equality, and positivity (even for integer LRS) is equivalent to Markov inequality.

But the decidability of the Skolem and positivity problems have been open for over 70 years! The only things we know are the following.

- It is decidable for a rank $\leqslant 8$ (result of 2021).
- Let $(u_n)_n$, $Z = \{n \,|\, u_n = 0\}$ is of the form $F \cup \underbrace{(F' + p\mathbb{N})}_{\text{computable}}$ with F and F' finite (Skolem-Mahler-Lech).
- If Skolem was decidable at order $\simeq 10$, then one could write an algorithm that computes arbitrarly many digits of some number α, but mathematicians still cannot determine the first digit of α.

Lemma 8.14

Markov equality is a simpler problem than Skolem.

Proof.

Problem A

Given $A \in \mathbb{Q}^{d\times d}$ stochastic, and $y \in \{0,1,2\}^d$, decide whether if there exists n such that $eA^n y = 1$, where $e = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}$.

With $y \in \{0,1\}^d$ this is exactly a problem about Markov chains, but like that it is strange.

Proposition 8.15

Skolem (for rational LRS) reduces to Problem A.

Proof.
Skolem is equivalent to Given A, decide $\exists n, (A^n)_{1,2} = 0$.
Let $A \in \mathbb{Q}^{d\times d}$ from a Skolem problem of this form. Write $A = A^+ - A^-$ where A^+, A^- have nonnegative coefficients.

Let $e = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}$, $P = \begin{bmatrix} A^+ & A^- \\ A^- & A^+ \end{bmatrix}$ and $V = \begin{bmatrix} x \\ -x \end{bmatrix}$, where $x = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$.

For any $n \in \mathbb{N}$, $eP^n v = e(A^+ - A^-)^n x$.

$$\begin{aligned} Pv &= \begin{bmatrix} A^+ & A^- \\ A^- & A^+ \end{bmatrix} \begin{bmatrix} x \\ -x \end{bmatrix} \\ &= \begin{bmatrix} A^+x - A^-x \\ A^-x - A^+x \end{bmatrix} \\ &= \begin{bmatrix} z \\ -z \end{bmatrix} \end{aligned}$$

with $z = (A^+ - A^-)x$.

$$\begin{aligned} Pv \begin{bmatrix} z \\ -z \end{bmatrix} &= \begin{bmatrix} (A^+ - A^-)z \\ (A^- - A^+)z \end{bmatrix} \\ &= \begin{bmatrix} (A^+ - A^-)^2 z \\ (A^- - A^+)^2 z \end{bmatrix} \end{aligned}$$

By induction, $P^n v = \begin{bmatrix} (A^+ - A^-)^n z \\ (A^- - A^+)^n z \end{bmatrix}$.
Let $s \in \mathbb{Q}$ such that sP is substochastic.
Let $\tilde{e} = \begin{bmatrix} e & 0 \end{bmatrix}$, $\mathbf{1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$, $\tilde{P} = \begin{bmatrix} sP & \mathbf{1} - sP\mathbf{1} \\ 0 & 1 \end{bmatrix} \in \mathbb{Q}^{(2k+1)\times(2k+1)}$, $\tilde{v} = \begin{bmatrix} \mathbf{1} + v \\ 1 \end{bmatrix}$, where $v = \begin{bmatrix} x \\ -x \end{bmatrix}$.

$$\begin{aligned} \tilde{e}\tilde{P}^n\tilde{v} &= \begin{bmatrix} e & 0 \end{bmatrix} \begin{bmatrix} (sP)^n(\mathbf{1} + v) + (\mathbf{1} - (sP)^n\mathbf{1}) \\ 1 \end{bmatrix} \\ &= e(sP)^n(\mathbf{1} + v) + e(\mathbf{1} - (sP)^n\mathbf{1}) \\ &= 1 + s^n \cdot eP^n v \\ &= 1 + s^n (A^n)_{1,2}. \end{aligned}$$

So $\tilde{e}\tilde{P}^n\tilde{v} = 1$ iff $(A^n)_{1,2} = 0$.

Proposition 8.16

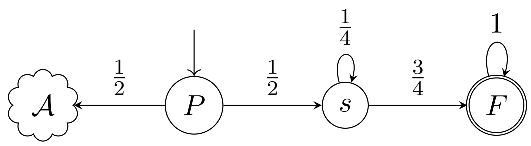
Problem A reduces to Markov equality with threshold $\frac{1}{2}$.

Proof.
Let $e = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}$, $A \in \mathbb{Q}^{d\times d}$ stochastic and $y \in \{0,1,2\}^d$.

Let $s = \begin{bmatrix} e & 0 & 0 \end{bmatrix}$, $B = \begin{bmatrix} \frac{1}{4}A & \frac{1}{4}y & \mathbf{1} - \frac{1}{4}(A\mathbf{1}+y) \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$, $t = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \end{bmatrix}$ and $M = \left[\begin{array}{c|c} A & x \\ \hline 0 & 0 \end{array}\right]$.

$M^n = \left[\begin{array}{c|c} A^n & A^{n-1}x \\ \hline 0 & 0 \end{array}\right]$.

B is stochastic ($\mathbf{1} - (A\mathbf{1}+y) \geqslant 0$). By induction, $sB^nt = \frac{1}{4}eA^{n-1}y$.
We want to decide $\exists n, eA^ny = 1$, i.e. $sB^{n+1}t = \frac{1}{4^n}$, i.e. $sB^{n+1}t + \frac{1}{2} - \frac{1}{4^n} = \frac{1}{2}$.



where $\mathcal{A} = \langle s, B, t\rangle$.
$\mathcal{B}(n+1) = \frac{1}{2}\mathcal{A}(n) + \frac{1}{2}(1 - \frac{1}{4^n}) = \frac{1}{2}(1 + sB^nt - \frac{1}{4^n})$.
On the right, after n steps, the automaton is in sort, by stochasticity, $\mathbb{P}[F] = 1 - \mathbb{P}[s]$.

So

- $\mathcal{B}(n+1) = \frac{1}{2}$ iff $sB^nt = \frac{1}{4^n}$
- $\mathcal{B}(0) = 0$
- $\mathcal{B}(1) = 0$

Corollary 8.17

Skolem and Markov equality are interreducible.

Theorem 8.18

The following are inter-reducible.

- Skolem for integer LRS
- Skolem for rational LRS
- Markov equality for $\frac{1}{2}$

Theorem 8.19

The following are inter-reducible.

- Positivity for integer/rational LRS
- Strict positivity for integer/rational LRS
- Markov inequality for $\frac{1}{2}$
- Markov strict inequality for $\frac{1}{2}$

Remark 8.20

Let $(u_n)_n \in \mathbb{Z}^{\mathbb{N}}$ integer. $\exists n, u_n = 0?$.
Let $v_n = u_n^2$ LRS. $\forall n, u_n \neq 0 \iff \forall n, v_n > 0$.
Skolem is easier than Positivity.

Definition 8.21 **Periodic set, ultimately periodic set, quasi-periodic set**

A set $A \subseteq \mathbb{N}$ is

- periodic if there exists r such that $\forall q, q \in A \iff q + r \in A$. (ex: $42\mathbb{N} \cup (3 + 42\mathbb{N})$)
- ultimately periodic if there exists q_0, r such that $\forall q \geqslant q_0, q \in A \iff q + r \in A$.
- quasi-periodic if it is the union of a finite setand a periodic set. (ex: $\{3, 7\} \cup (50 + 42\mathbb{N})$)

Theorem 8.22 **Skolem-Mahler-Lech theorem**

Let $(u_n)_n$ be a LRS. Then $Z := \{n \,|\, u_n = 0\}$ is ultimately periodic.

In fact, $A = F \cup (q_0 + P)$, but F is not constructive.

Part IV
Automata and semigroups (Matthieu Picantin)

History of the interactions between automata theory and infinite semigroups theory

In the 90s, two theories have been developped concomitantly and independently:

- Automatic (semi)groups
- Automaton (semi)group

The community and the results are disjoint.

9 Automaton semigroups

9.1 Basics

Definition 9.1 Finite deterministic complete automaton

A finite, deterministic, complete automaton is a triple (Q, Σ, τ) with

- Q the stateset
- Σ the alphabet
- $\tau = (\tau_i : Q \to Q)_{i \in \Sigma}$ a family of functions.

There are no initial and final states.

Definition 9.2 Mealy automaton

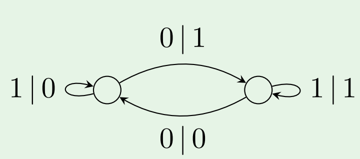
A Mealy automaton is a quadruple $(Q, \Sigma, \tau, \sigma)$ such that (Q, Σ, τ) and (Σ, Q, σ) both are finite deterministic complete automata.

In other terms, a Mealy automaton is a finite, deterministic, complete letter-to-letter transducer with the same input/output alphabet.

For every $p \in Q$ and every $x \in \Sigma$, there exists exactly one transition from p with the input letter x:

$$p \xrightarrow{x \,|\, \sigma_p(x)} \tau_x(p)$$

Example 9.3 The lamplighter automaton



A crucial point with Mealy automata is that:

- states act on letters
- letters act on states.

Definition 9.4 **Composition of actions**

Such actions can be composed in the following way: $\forall p \in Q, q \in Q^*, x \in \Sigma, u \in \Sigma^*$,

$$\sigma_{qp}(x) = \sigma_p(\sigma_q(x)) \text{ and } \tau_{ux}(p) = \tau_x(\tau_u(p)).$$

We extend recursively these actions:

$$\begin{tikzcd}
 & u \arrow[d, dash] & & x \arrow[d, dash] & \\
q \arrow[r, dash] & {} \arrow[r] \arrow[d] & \tau_u(q) \arrow[r, dash] & {} \arrow[r] \arrow[d] & \tau_x(\tau_u(q)) \\
 & \sigma_q(u) \arrow[d, dash] & & \sigma_{\tau_u(q)}(x) & \\
p \arrow[r, dash] & {} \arrow[r] \arrow[d] & \tau_{\sigma_q(u)}(p) & & \\
 & \sigma_p(\sigma_q(u)) & & &
\end{tikzcd}$$

We have $\sigma_q(ux) = \sigma_q(u)\sigma_{\tau_u(q)}(x)$ and $\tau_u(qp) = \tau_u(q)\tau_{\sigma_q(u)}(p)$.

All these mappings are length-preserving and prefix-preserving. In particular, the image of the empty word ε is itself.

Definition 9.5 **Semigroup generated by a Mealy automaton**

The composition gives a semigroup structure to the set of those transformations $\sigma_q : \Sigma^* \to \Sigma^*$ for $q \in Q^+$.
This semigroup is called the semigroup generated by the Mealy automaton $\mathcal{A} = (Q, \Sigma, \tau, \sigma)$ and is denoted $\langle \mathcal{A} \rangle_+$.

Definition 9.6 **Automaton semigroup**

An automaton semigroup is a semigroup which can be generated by some Mealy automaton.

Any element of such an automaton semigroup induces a finite-state transformation.

Notation 9.7

For any length-preserving and prefix-preserving transformation t of Σ^{*}[a] and for any $u \in \Sigma^*$, we denote

- u^t the image of u by t
- $t@u$ the unique transformation s of Σ^* satisfying $(uv)^t = u^t v^s$

[a] $t \in \mathrm{End}(\Sigma^*)$ is a tree endomorphism

Definition 9.8 **Finite-state transformation**

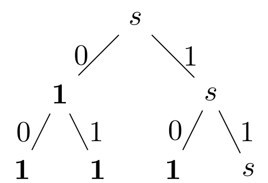
Whenever $Q(t) = \{t@u \mid u \in \Sigma^*\}$ is finite, the transformation t is said to be finite-state (and we denote $t \in \mathrm{FEnd}(\Sigma^*)$): for each state $s \in Q(t)$ we write a decomposition (traditionnaly called a wreath recursion in an automata theory). $s = (s@x_1, s@x_2, ..., s@x_{|\Sigma|})\alpha_s$ where $\alpha_s = [x_1^s, x_2^s, ..., x_{|\Sigma|}^s]$ denotes the induced transformation of s on Σ.
Whenever α_s is a permutation ($a \in \mathfrak{S}_\Sigma$), we can use parenthese instead of brackets.

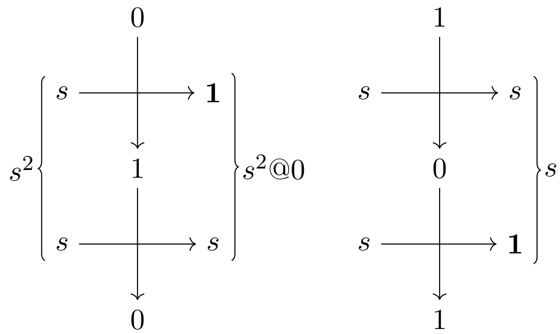
Exercise A.1

Draw the Mealy automaton $\mathcal{M}_s$ for the transformation $s = (\mathbf{1}, s)(0,1) \in \mathrm{FEnd}(\{0,1\}^*)$ where $\mathbf{1}$ denotes the identity transformation (that is, the unit of the monoid $\mathrm{FEnd}(\{0,1\}^*)$). Compute and draw its successive powers $\mathcal{M}_{s^2}, \mathcal{M}_{s^3}, \ldots$. Try to recognize the monoid or the group generated by s.

Proof.

Let $s \in \mathrm{FEnd}(\{0,1\}^*)$ be defined by $s = (\mathbf{1}, s)(0,1) = (\mathbf{1}, s)[1,0]$.
We can label the vertices of the tree $\{0,1\}^*$:





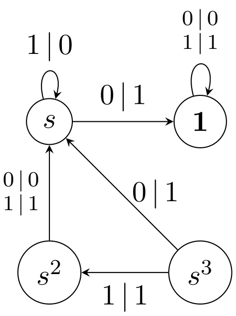
$s^2 = (s,s)[0,1] = (s,s)() = (s,s)$

We can use an "inline computation":

$$
\begin{aligned}
s^2 &= (\mathbf{1}, s)(0,1)(\mathbf{1}, s)(0,1) \\
&= (\mathbf{1}, s)(s, \mathbf{1})(0,1)(0,1) \\
&= (\mathbf{1}s, s\mathbf{1})() \\
&= (s,s)
\end{aligned}
$$

$$
\begin{aligned}
s^3 &= s^2 \times s \\
&= (s,s)(\mathbf{1}, s)(0,1) \\
&= (s, s^2)(0,1)
\end{aligned}
$$

$\mathcal{M}_s$:



Proposition 9.9

$\mathrm{FEnd}(\Sigma^*)$ is the semigroup of all those finite-state transformations of Σ^*.

Definition 9.10 Invertible, reversible Mealy automaton

A Mealy automaton $\mathcal{A} = (Q, \Sigma, \tau, \sigma)$ is

- invertible if every function σ_q is a permutation of Σ
- reversible if every function τ_x is a permutation of Q

Definition 9.11 Inverse automaton

Let $\mathcal{A}$ be an invertible Mealy automaton. We can consider its inverse $\mathcal{A}^{-1} = (Q^{-1}, \Sigma, \tau', \sigma') = i\mathcal{A}$.

$$p \xrightarrow{x\,|\,y} q \in \mathcal{A} \iff p^{-1} \xrightarrow{y\,|\,x} q^{-1} \in \mathcal{A}^{-1}$$

Definition 9.12 Dual automaton

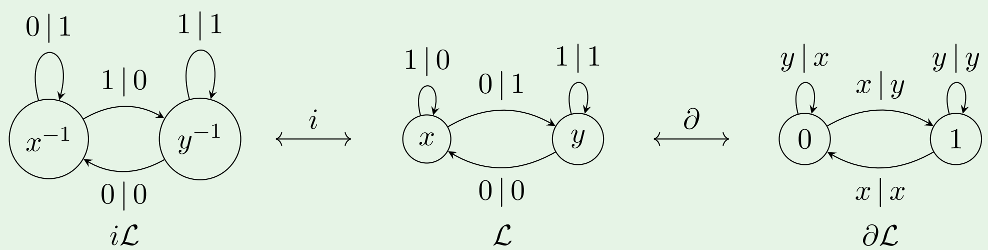
Let $\mathcal{A}$ be a Mealy automaton. We can consider its dual $\partial\mathcal{A} = (\Sigma, Q, \sigma, \tau)$.

$$p \xrightarrow{x\,|\,y} q \in \mathcal{A} \iff x \xrightarrow{p\,|\,q} y \in \partial\mathcal{A}$$

An automaton is invertible iff its dual is reversible.

Example 9.13

Let $\mathcal{L}$ be the lamplighter automaton of Example 9.3. $\mathcal{L}$ is invertible.



Definition 9.14 Bireversible automaton

An invertible automaton $\mathcal{A}$ is bireversible if both $\mathcal{A}$ and $\mathcal{A}^{-1}$ are reversible.

Proposition 9.15

When $\mathcal{A} = (S, \Sigma, \tau, \sigma)$ is invertible, the transformations σ_q for $q \in Q$ are invertible, and generate a groupe, denoted by $\langle \mathcal{A} \rangle$.

9.2 Problems

While many undecidable problems in the class of (semi)groups remain undecidable in the subclass of automaton (semi)groups, the underlying automata provide a combinatorial leverage to solve, for instance, the Word Problem.

Definition 9.16 Word problem

Does there exist an algorithm that, given a Mealy automaton and two state-words, decides whether the latter induce a same transformation?

This problem is undecidable in general but it is decidable for the class of automaton (semi)groups.

There are two possible approaches:

- minimisation of Mealy Automata
- wreath recursion.

Some other decision problems are

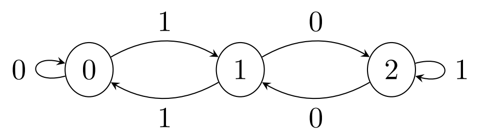
- Conjugacy problem: given $\mathcal{A} = (Q, \Sigma, \tau, \sigma)$ and $u, v \in Q^+$, decide whether there exists $w \in Q^*$ such that $\sigma_{uw} = \sigma_{wv}$
- Order problem: given $\mathcal{A} = (Q, \Sigma, \tau, \sigma)$ and $u \in Q^*$, decide whether there exists $k, l \in \mathbb{N}$ satisfying $\sigma_u^k \sigma_u^l = \sigma_u^k$
- Finiteness problem: given $\mathcal{A} = (Q, \Sigma, \tau, \sigma)$, decide whether $\langle \mathcal{A} \rangle_+$ is finite

Exercise A.3

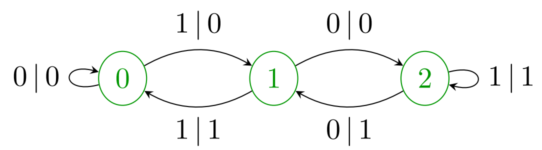
First build the minimal automaton that recognizes the language $\{u \in \{0,1\}^* \mid (u)_2 \equiv 0[3]\}$. Deduce the Mealy automaton $\mathcal{M}_{3,2}$ that allows to compute the division by 3 in base 2. Check whether $\mathcal{M}_{3,2}$ is invertible or reversible. Draw the dual automaton $\partial\mathcal{M}_{3,2}$. Experiment and find some possible relation(s) between the generators of the monoid or the group generated by $\partial\mathcal{M}_{3,2}$. Try to generalise.

Proof.

The minimal automaton that recognizes the language is the following:

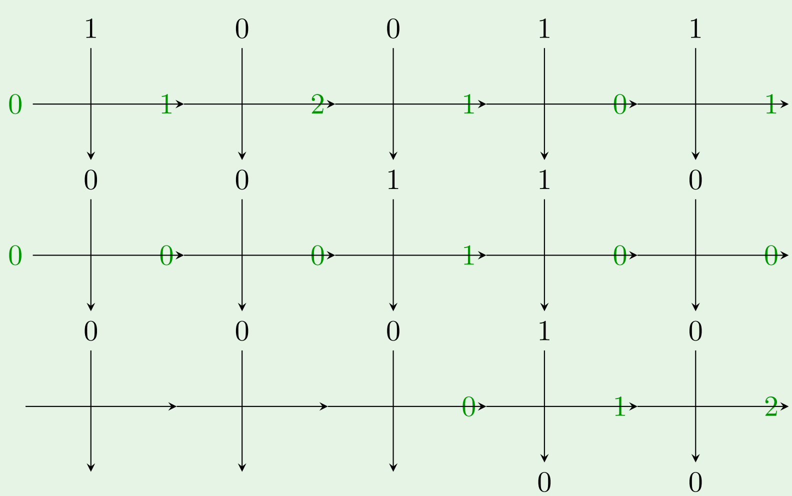


$\mathcal{M}_{3,2}$ is the following:



Example 9.17

$(19)_{10} = (10011)_2$



$\partial\mathcal{M}_{3,2}$ computes the multiplication by 2 in base 3.

9.3 Tools

9.3.1 Dualisation

Definition 9.18 Helix graph

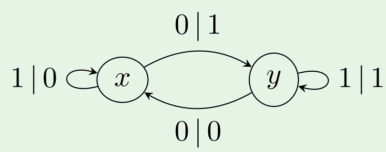
For any Mealy automaton $\mathcal{A} = (Q, \Sigma, \tau, \sigma)$, the helix graph $\mathcal{H}_{n.k}(\mathcal{A})$ is the digraph with nodes $Q^n \times \Sigma^k$ and arrows

$$\boxed{u, v} \longrightarrow \boxed{\tau_v(u), \sigma_u(v)}$$

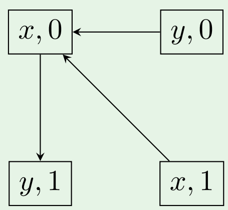
for all $(u, v) \in Q^n \times \Sigma^k$.

Example 9.19

With $\mathcal{L}$:



$\mathcal{H}_{1,1}(\mathcal{L})$ is:



(see exercises A.6 and A.7)

9.3.2 Product, conjugation, exponentiation

Definition 9.20 Product of Mealy automata

Given two Mealy automata $\mathcal{M}' = (Q', \Sigma, \tau, \sigma)$ and $\mathcal{M}'' = (Q'', \Sigma'', \tau'', \sigma'')$, the product $\mathcal{M}'\mathcal{M}''$ is the Mealy automaton $(Q' \times Q'', \Sigma, \tau, \sigma)$ with τ and σ satisfying:

$$\tau_u((q', q'')) = (\tau'_u(q'), \tau''_{\sigma'_{q'}(u)}(q'')$$

$$\sigma_{(q',q'')}(u) = \sigma''_{q''}(\sigma'_{q'}(u))$$

Definition 9.21 Conjugate of Mealy automaton

Given a Mealy automaton $\mathcal{M} = \mathcal{A}\mathcal{B}$, a conjugate of $\mathcal{M}$ is the recomposition $\mathcal{B}\mathcal{A}$.

Definition 9.22 Exponentiation of Mealy automaton

Given a Mealy automaton $\mathcal{M} = (Q, \Sigma, \tau, \sigma)$ and $n > 0$, the n-th power of $\mathcal{M}$ is $\mathcal{M}^n = (Q^n, \Sigma, (\tau'_i : Q^n \to Q^n), (\sigma_{q'} : \Sigma \to \Sigma))$.

Proposition 9.23

The connected components of a reversible Mealy automaton are strongly connected.

Proposition 9.24

All powers of a reversible Mealy automaton are reversible.

Definition 9.25 **Schreier tree**

Given a Mealy automaton $\mathcal{M}$, the Schreier tree $t(\mathcal{M})$ is the tree whose vertices are the connected components of the powers of $\mathcal{M}$ and the indicence relation is built by adding a state: for any $n \geqslant 0$, the connected components of a word $u \in Q^n$ is the parent of the connected components of uq for $q \in Q$.

9.3.3 Minimisation

Definition 9.26 **Congruence**

Let $\mathcal{M} = (Q, \Sigma, \tau, \sigma)$ be a Mealy automaton.
An equivalence relation $\cong$ of Q is a congruence for $\mathcal{M}$ if is satisfies

$$\forall p, q \in Q, p \cong q \Longrightarrow \big(\forall i \in \Sigma, \sigma_p(i) = \sigma_q(i) \text{ and } \tau_i(p) \cong \tau_i(q)\big)$$

Definition 9.27 **Nerode equivalence**

Let $\mathcal{M} = (Q, \Sigma, \tau, \sigma)$ be a Mealy automaton.
The Nerode equivalence is the coarsest[a] congruence for $\mathcal{M}$ and can be obtained as the limit of the sequence $(\equiv_k)_k$ defined by

$$\begin{cases} p \equiv_0 q & \Leftrightarrow \quad \sigma_p(i) = \sigma_q(i) \text{ for each } i \in \Sigma \\ p \equiv_{k+1} q & \Leftrightarrow \quad p \equiv_k q \text{ and } \tau_i(p) \equiv_k \tau_i(q) \text{ for each } i \in \Sigma \end{cases}$$

For each $q \in Q$, we denote $[q]$ the Nerode equivalence class of q.

[a]moins fine

Definition 9.28 **Minimisation of a Mealy automaton**

Let $\mathcal{M} = (Q, \Sigma, \tau, \sigma)$ be a Mealy automaton and $\equiv$ its Nerode equivalence.
The minimization $\mathcal{M}/\equiv$ of $\mathcal{M}$ is the Mealy automaton $(Q/\equiv, \Sigma, \tilde{\tau}, \tilde{\sigma})$ where each $(p, x) \in Q \times \Sigma$ satisfies

$$\begin{cases} \tilde{\tau}_x([p]) = [\tau_x(p)] \\ \tilde{\sigma}_{[p]}(x) = \sigma_p(x). \end{cases}$$

9.4 Finiteness problem

Proposition 9.29

A (semi)group $\langle \mathcal{M} \rangle_{(+)}$ is finite if $\langle \partial\mathcal{M} \rangle_{(+)}$ is finite.

Proof sketch.
Assume $\mathcal{M} = (Q, \Sigma, \tau, \sigma)$ and $\langle \partial\mathcal{M} \rangle_{(+)} = \{\tau_u \,|\, u \in \Sigma^*\}$ is finite. Consider the Cayley graph $\mathcal{G}$ of $\langle \partial\mathcal{M} \rangle_{(+)}$

$$\tau_u \longrightarrow \tau_{ux}$$

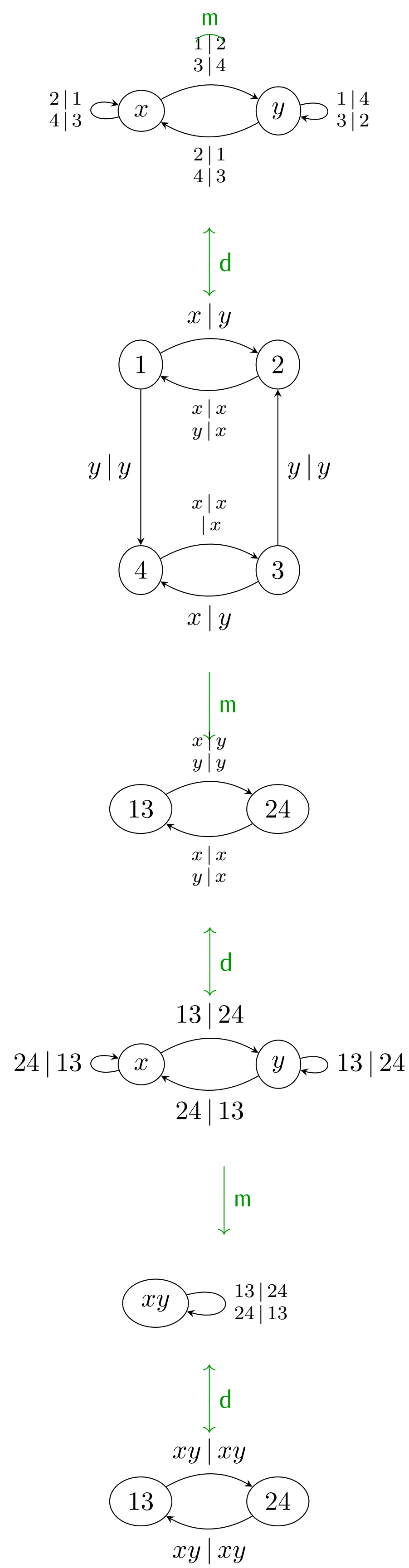
Complete this graph into a Mealy automaton and conclude by proving $|\langle \mathcal{M} \rangle| \leqslant \left|\Sigma^{\Sigma \times \langle \partial\mathcal{M} \rangle}\right|$.

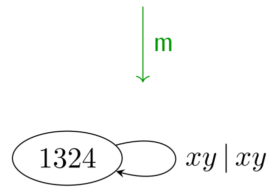
We define the md-reduction: the idea is to apply mdmd repeatedly until stabilisation.

Exercise A.12

Decide the finiteness for the (semi)group generated by $x = (y, x, y, x)(1, 2)(3, 4)$ and $y = (y, x, y, x)(1, 4, 3, 2)$.

Proof.





Corollary 9.30

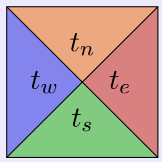
A Mealy automaton $\mathcal{M}$ and its md-reduction generates either

Definition 9.31 **Finiteness problem**

Does there exists an algorithm that, given a Mealy automaton $\mathcal{M}$ decides whether the generated (semi)group $(\langle \mathcal{M} \rangle_{(+)}$ is finite?

Definition 9.32 **Wang tile**

A Wang tile is a unit square tile with a color on each edge.



Given a Wang tileset $\mathcal{T}$, a Wang tiling of a subset P of $\mathbb{Z}^2$ is a map $P \to \mathcal{T}$.
A Wang tiling f is valid whenever for each $(x, y) \in P$, f associates a tile $f(x, y)$ such that $f(x, y)_n = f(x, y+1)_s$ and $f(x, y)_e = f(x+1, y)_w$ if $(x, y+1) \in P$ and $(x+1, y) \in P$.

Result 9.33

For any Wang tileset $\mathcal{T}$, $\mathbb{Z}^2$ admits a valid tiling for $\mathcal{T}$ iff does each finite subset P of $\mathbb{Z}^2$.

Definition 9.34 **cd-deterministic Wang tileset**

A Wang tileset $\mathcal{T}$ is cd-deterministic with $(c, d) \in \{(w, n), (n, e), (e, s), (s, w)\}$ if each tile $t \in \mathcal{T}$ is uniquely determined by its pair (t_c, t_d) of colors.

Definition 9.35 **Tiling problem**

Does there exist an algorithm that, given a Wang tileset $\mathcal{T}$, decide whether $\mathbb{Z}^2$ admits a tiling for $\mathcal{T}$.

Proposition 9.36

The tiling problem is undecidable, even on NW-deterministic tilesets.

With any NW-deterministic Wang tileset $\mathcal{T}$, we associate the Mealy automaton $\mathcal{W}_{\mathcal{T}} = (Q, \Sigma, \tau, \sigma)$ with $Q = \Sigma = \mathcal{T} \sqcup \{\square\}$, $\tau_b(a) = b$ for $(a, b) \in Q^2$ and

$$\begin{cases} c \text{ for } (a, b, c) \in \mathcal{T}^3 \text{ with } a_e = c_w \text{ and } b_s = c_n \\ \square \end{cases}$$

Lemma 9.37

If $\mathbb{Z}^2$ admits some valid Wang tiling for $\mathcal{T}$, $\langle \mathcal{W}_{\mathcal{T}} \rangle_+$ is infinite. (see exercise A.10)

Proposition 9.38

If $\mathbb{Z}^2$ does not admit a valid Wang tiling for $\mathcal{T}$, $\langle \mathcal{W}_{\mathcal{T}} \rangle_+$ is finite.

Proof.
There exists $n \in \mathbb{N}$ such that $\{0, 1, \ldots, n\}^2$ admits no valid Wang tiling for $\mathcal{T}$. Let fix $(p, q) \in \Sigma^n \times \Sigma^\omega$.
We want to prove that any word $u \in Q^{2n}$ satisfies $\sigma_u(pq) = \sigma_u(p)\square^\omega$.
We have $\langle \mathcal{W}_{\mathcal{T}} \rangle_+ = \underbrace{\{\sigma_w \mid w \in Q^{<2n}\}}_{\text{of cardinality } 1+|Q|+\ldots+|Q^{2n}|} \sqcup \underbrace{\{\sigma_w \mid w \in Q^{2n}Q^*\}}_{\text{of cardinality } \leqslant |\Sigma^{n}{}^{\Sigma^n}|}$

Theorem 9.39

The semigroup $\langle \mathcal{W}_{\mathcal{T}} \rangle$ is infinite iff $\mathbb{Z}^2$ admits a valid Wang tiling for $\mathcal{T}$.

Corollary 9.40

Finiteness problem is undecidable.

10 Automatic semigroups

10.1 Basics

We will restrain to monoids.

Definition 10.1 Normal form

Let M be a monoid with a generating set Q and $\mathrm{EV} : Q^* \longrightarrow M$ be its evaluation morphism.
A normal form for (M, Q) is a map $\mathbf{NF} : M \longrightarrow Q^*$ that assigns to each element of M a distinguished representative word over Q (with $\mathrm{EV} \circ \mathbf{NF} = \mathrm{Id}_M$).

This provides a (right-)automatic structure for M.

Definition 10.2 Automatic monoid

M is said to be a (right-)automatic monoid if for every $q \in Q \sqcup \{\#\}$,

$$\mathcal{L}_q = \{(\mathbf{NF}(a)\#^{\max(0,|\mathbf{NF}(aq)|-|\mathbf{NF}(a)|)}, \mathbf{NF}(aq)\#^{\max(0,|\mathbf{NF}(a)|-|\mathbf{NF}(aq)|)} \mid a \in M\}$$

is regular (over $(Q \sqcup \{\#\}) \setminus \{(\#,\#)\}$), where the normal forms of a pair are right-padded with an extra symbol $\# \notin Q$ to equalize the length.

Thurston shows how the whole set of those different automata recognizing the multiplication can be replaced with advantage by a single letter-to-letter transducer over Q that computes the formal form via iterated runs: each run both provides one symbol of the final normal form and outputs a word still to be normalised.

Example 10.3 Abelian free monoid

Let M_n be the rank n free abolian monoid with base $A_n = \{a_1, \ldots, a_n\}$.
We have $(M_n, \cdot) \cong (\mathbb{N}^n, +)$ with $a_i \leftrightsquigarrow (0, \ldots, 0, \underbrace{1}_{i^{\text{th}}}, 0, \ldots, 0)$.

$M_n = \langle a_1, \ldots, a_n \mid a_i a_j = a_j a_i, 1 \leqslant i < j \leqslant n \rangle$

- a lexicographic normal form $\mathbf{NF}^{\mathrm{Lex}}$ with respect to some total order $\leqslant$ on A_n ($a_1 \leqslant \ldots \leqslant a_n$): every element of M admits a unique decomposition $s_k \ldots s_1$ with $s_{i+1} \geqslant s_i$.

For example, $\mathbf{NF}^{\mathsf{Lex}}(a_4a_2^2a_3a_1a_5a_4^3) = a_5a_4^4a_3a_2^2a_1$.

- a Garside normal form $\mathbf{NF}^{\mathsf{Gar}}$ with respect to an augmented generating set

$$Q_n = \left\{ \prod_{i \in I} a_i \,|\, \emptyset \neq I \subseteq \{1, ..., n\} \right\}:$$

every element of M admits a unique decomposition $q_k...q_1$ with $\forall q \in Q_n, q > q_i \Rightarrow q_k...q_i \not\geq q$ (meaning that q_i is the maximal element of Q_n right-dividing $q_k...q_i$). $q > q_i$ means $\exists p \neq 1, q = pq_i$. $q_k...q_i \not\geq q$ means $\nexists r, q_k...q_i = rq$.

For example $\mathbf{NF}^{\mathsf{Gar}}(a_4a_2^2a_3a_1a_5a_4^3) = a_4a_4(a_2a_4)(a_1a_2a_3a_4a_5)$.

Exercise B.1

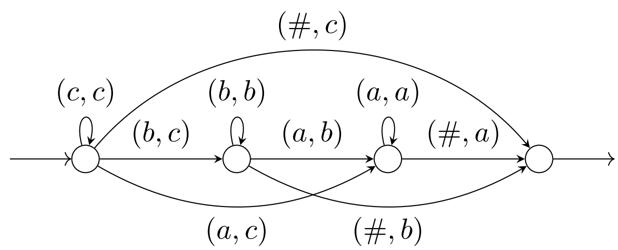
We consider the monoid $T = \langle a, b, c \,|\, ab = ba, bc = cb, ac = ca \rangle_+^1$, known as the rank 3 abelian free monoid.

1. Give a rational expression for each of the languages $\mathbf{NF}^{\mathsf{Lex}}(T)$ and $\mathbf{NF}^{\mathsf{Gar}}(T)$
2. Draw automata recognizing each of the languages $\mathcal{L}_q$ for $\mathbf{NF}^{\mathsf{Lex}}$ and $\mathbf{NF}^{\mathsf{Gar}}$ with $q \in \{a, b, c\}$, to conclude that T is automatic (via each of these structures).

Proof.

1. $\mathbf{NF}^{\mathsf{Lex}}(T) = a^*b^*c^*$
$\mathbf{NF}^{\mathsf{Gar}}(T) = ((a^* + b^*)(ab)^* + (a^* + c^*)(ac)^* + (b^* + c^*)(bc)^*)(abc)^*$

2. For $\mathcal{L}_c$



Example 10.4 ***n*-strang braid monoid**

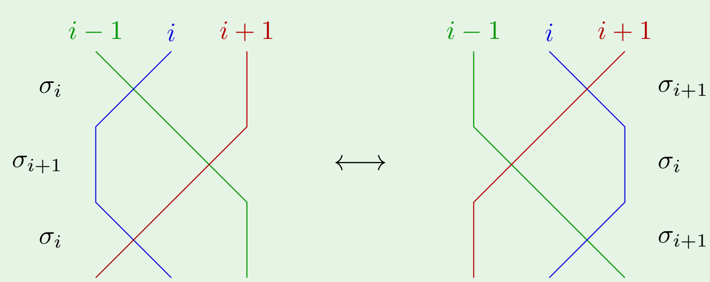
The n-strand braid monoid

$$B_n^+ = \left\langle \sigma_1, .., \sigma_{n-1} \,\middle|\, \begin{cases} \sigma_i\sigma_j = \sigma_j\sigma_i & \text{if } |i-j| > 1 \\ \sigma_i\sigma_{i+1}\sigma_i = \sigma_{i+1}\sigma_i\sigma_{i+1} & \end{cases} \right\rangle_+^1$$

Theorem 10.5

By correspondence $\sigma_i \rightsquigarrow |\cdots \times \cdots|$

and diagram concatenation (stacking), each element of B_n^+ can be interpreted as an isotopy class of a positive strand braid diagram.



We can consider $Q_n = \{\text{simple } n\text{-strand braids}\} = \{q \in B_n^+ \mid \Delta_n \geqslant q\}$. Simple means that any two strands cross at most once, and Δ_n is the half-twist.

Theorem 10.6

Every element of B_n^+ admits a unique decomposition $q_k...q_1$ with $\forall q \in Q_n, q > q_i \Rightarrow q_k...q_i \not\geqslant q$.

10.2 Garside 1

Definition 10.7 Left-divisor, right-multiple, least common right-multiple

Assume that M is a monoid.
For $a, b \in M$, we say that b is a left-divisor of a (or, a is a right-multiple of b) if there exists $d \in M$ satisfying $a = bd$.
An element $c \in M$ is a least common right-multiple of a and b (a right-lcm) if c is right-multiple of both a and b and every right-multiple of a and b is a right-multiple of c.

Definition 10.8 Cancellative, conical monoid

The monoid M is said cancellative when, for $a, b, c, d \in M$,$abc = adc \Rightarrow b = d$.
M is said to be conical if 1 is the only invertible of M: $ab = 1 \Rightarrow a = b = 1$.

These two properties imply that left and right-divisibility are orders.

Notation 10.9 Right-lcm

Whenever M is cancellative and conical, the unique right-lcm of a and b is denoted by $a \vee b$, when it exists.
$a \vee b = a(a \setminus b) = b(b \setminus a)$

Definition 10.10 Garside monoid

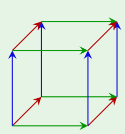
A monoid M is Garside if

- M is cancellative and conical
- every pair of elements admit a left and a right-lcm
- M admits a Garside element: an element whose left and right-divisors[a] are finite in number and coincide.

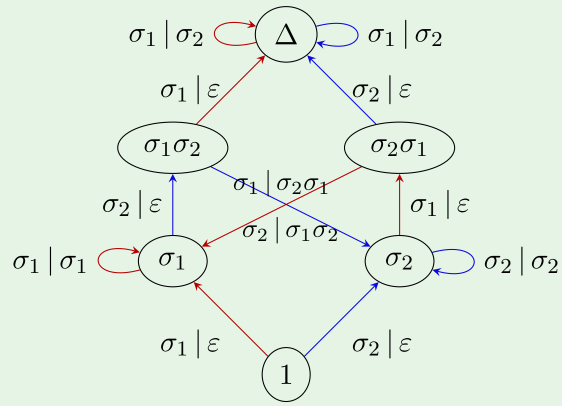
[a]These are called simples

Example 10.11

- The free abelian monoid M_k is a Garside monoid with $\Delta = a_1...a_k$. Its lattice of $\mathrm{div}(\Delta)$ (simples) is a n-dimensional hypercube.



- The n-strand braid monoid B_n^+ is a Garside monoid with Garside element the half-twist Δ_n. Its lattice of $\mathrm{div}(\Delta_n)$ is the n-dimensional permutohedron.



For example $N(\sigma_1\sigma_2\sigma_1\sigma_1\sigma_2) = N(\sigma_2\sigma_1)\Delta = (\sigma_2\sigma_1)\Delta$.

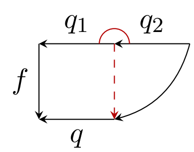
10.3 Garside 2

For any two elements f and g of a monoid M we associate arrows $\xleftarrow{f}$ for f, $\xleftarrow{g}$ for g and $\xleftarrow{fg}$ for the product fg.

Definition 10.12 Q**-normal word**

A word $q_2q_1 \in Q^2$ is Q-normal whenever we have

$$\forall q \in Q, \forall f \in M, q_2q_1f \geqslant q \Rightarrow q_1f \geqslant q.$$



Where $\frown$ is a symbol indicating the Q-normality.

Definition 10.13 Q**-normal word**

A word $q_k...q_1 \in Q^k$ is Q-normal whenever so are each $q_{i+1}q_i$.

Definition 10.14 **Garside family**

Q is a Garside family for M if each element of M admits a Q-normal decomposition.

Lemma 10.15

If M is a Garside monoid, then $\mathrm{div}(\Delta)$ is a Garside family.

Theorem 10.16

Let M be a conical, right-cancellative, and left-Noetherian (there does not exist infinite sequence for left-divisibility) monoid.
Then a generating set Q of M is a Garside family if and only if Q is closed by left-divisor and by left-lcm (common left-multiple implies least common left-multiple).

Question

How to copute a Garside normalisation with a family of Garside Q, $N : Q^* \to Q^*$ with $N = \mathbf{NF} \circ \mathrm{EV}$?

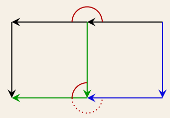
Lemma 10.17

For any pair $(q_1, q_2) \in Q^2$, the Q-normal decomposition of q_2q_1 is of length at most 2.

This leads to consider the restriction $\overline{N}$ of N to Q^2.

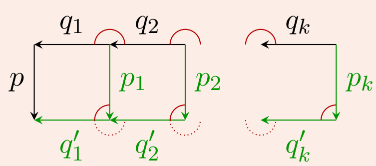
Lemma 10.18 Domino lemma

We have the following commutative diagram:



Where the dotted line represents the conclusion of the lemma.

Proposition 10.19



For $q_k..., q_1$ Q-normal, we have $N(q_k...q_1p) = \overline{N}_{k...321}(q_k...q_1p)$.

Corollary 10.20

For each $q \in Q$, there exists a transducer computing $N(wq)$ from $N(w)$.

Theorem 10.21

The Q-normal form of $w \in Q^N$ is given by $N(w) = \overline{N}_{\delta_N}(w)$ with $\delta_2 = 1$, $\delta_3 = 121$, $\delta_4 = 12321$, etc.

Corollary 10.22

Whenever Q is finite, the Word problem belongs to $\mathbf{DTIME}(n^2)$.

10.4 Quadratic normalisation

Definition 10.23 Normalisation

A normalisation is a pair (Q, N) with Q an alphabet and $N : Q^* \to Q^*$ satisfying, for all $u, v, w \in Q^*$:

- $\|N(w)\| = \|w\|$ (geodesy)
- $\|w\| = 1 \implies N(w) = w$ (atomicity)
- $N(uN(w)v) = N(uwv)$ (confluence)

Any word over Q fixed under N is called N-normal.

Such a pair (Q, N) is a normalisation for a monoid M whenever M admits the presentation

$$\langle Q : \{w = N(w) \mid w \in Q^*\} \rangle_+^1$$

Definition 10.24 **Quadratic normalisation**

A normalisation (Q, N) is quadratic if

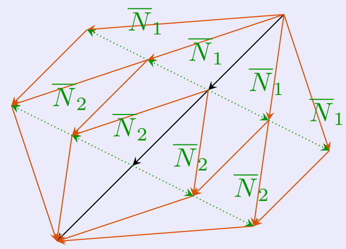
1. a word $w \in Q^*$ is N-normal iff so is each length 2 factor (static)
2. every word $w \in Q^*$ admits a finite sequence δ of positions satisfying $N(w) = \overline{N}_\delta(w)$ (dynamic)

Provided that Q is finite, the language of N-normal words is rational.

Definition 10.25 **3-complexity**

The 3-complexity of a quadratic normalisation (Q, N) is

$$(d,p) = \left(\max_{u \in Q^3} \min\{k \mid N(u) = \overline{N}_{\underbrace{212\ldots}_{k}}(u)\}, \max_{u \in Q^3} \min\{k \mid N(u) = \overline{N}_{\underbrace{121\ldots}_{k}}(u)\} \right)$$



10.5 Rewriting

Let (Q, R) be a rewriting system: Q alphabet and $R \subseteq Q^* \times Q^*$.

Notation 10.26

An element $(r, s) \in R$ is written $r \to s$ and called a rewriting rule.

We denote by $\to_R$ the closure of R with respect to product of Q:

$$u \to_R v \Leftrightarrow \exists w, w' \in Q^*, \exists (r,s) \in R, \begin{cases} u = wrw \\ v = wsw' \end{cases}$$

and by $\to_R^*$ the reflexive-transitive closure of $\to_R$:

$$u \to_R^* v \Leftrightarrow \exists p \geqslant 0, \exists w_1, \ldots, w_p \in Q^*, u \to_R w_1 \to_R \ldots \to_R w_p \to_R v$$

Definition 10.27

- A word $v \in Q^*$ is an R-normal form (of some word $u \in Q^*$) if $v \to_R^* w$ implies $v = w$ (and if $u \to_R^* v$ holds).
- (Q, R) is reduced if $u \to v \in R$ implies $\begin{cases} v & R\text{-normal} \\ u & R \setminus \{u \to v\}\text{-normal.} \end{cases}$
- (Q, R) is normalising if every word over Q admits at least an R-normal form.
- (Q, R) is confluent if for any $u, v, w \in Q^*$, with $u{}_R^* \leftarrow w \to_R^*$, there exists $w' \in Q^*$ such that $u \to_R^* w' {}_R{}^* \leftarrow v$.

  When the latter holds at least for $u_R \leftarrow w \to_R$, (Q, R) is locally confluent.
- (Q, R) is Noetherian or terminating if there is no infinite rewriting sequence $w_0 \to_R w_1 \to_r \ldots$.

  (Q, R) Noetherian implies (Q, R) normalising.

- (Q, R) is convergent or complete if (Q, R) is Noetherian and confluent.

  A convergent rewriting system (Q, R) gives a solution to the Word problem for the monoid $\langle Q : R \rangle_{+}^{1}$. The converse does not hold.

Theorem 10.28

Let (Q, R) be a Noetherian rewriting system.
Then (Q, R) is confluent iff (Q, R) is locally confluent.

Proposition 10.29

Let (Q, N) be a quadratic normalisation for a monoid M.
We obtain a quadratic, reduced, normalising and confluent rewriting system (Q, R) for M by setting

$$R = \{pq \to N(pq) \mid p, q \in Q, pq \neq N(pq)\}.$$

And reciprocally.

Theorem 10.30 **Dehornay, Suiraud's theorem**

Whenever (Q, N) has complexity at most $(4, 3)$, then the associated rewriting system is convergent.

From a word of Q^k, any rewriting sequence is of length

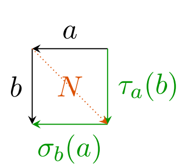
- at most $\frac{k(k-1)}{2}$ when the complexity is at most $(3, 3)$
- at most $2^k - k - 1$ when the complexity is at most $(4, 3)$.

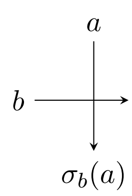
Remark 10.31

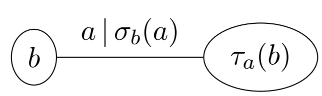
There exists some non-convergent rewriting system when complexity is $(4, 4)$.

11 Link between automaton semigroups and automatic semigroups

Assume that M is a monoid with a quadratic normalisation (Q, N) We associate a Mealy automaton $\mathcal{M}_{Q,N}(Q, Q, \tau, \sigma)$ such that, for every $(a, b) \in Q^2$, $\sigma_b(a)$ is the rightmost element in the N-normal form of ab and $\tau_a(b)$ the left one: $N(ab) = \tau_a(b)\sigma_b(a)$.







Result 11.1

If M is a monoid with a quadratic normalisation either over $Q \ni 1$ or over $Q' = Q \setminus \{1\}$, then M admits also a quadratic normalisation N over Q satisfying $N(1) = 1$ as required, and for each $q \in Q$, $N(1q) = N(q1) = 1q$ (unit condition).

Think of 1 as a dummy element that escapes the normalisation.

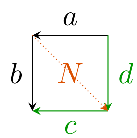
Result 11.2 **Top-approximation**

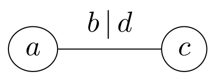
If M is a monoid with a quadratic normalisation (Q, N) satisfying the unit condition, then $\mathcal{M}_{Q,N}$ generates a monoid of which M is a quotient.

Result 11.3 **Bottom-approximation**

If M is a monoid with a quadratic normalisation (Q, N) of 3-complexity at most $(4, 3)$, then $\mathcal{M}_{Q,N}$ generates a monoid quotient of M.

To any quadratic normalisation (Q, N), we associate its Thurston transducer, defined as the Mealy automaton $\mathcal{T}_{Q,N}$ with states Q and alphabet Q.





Corollary 11.4

Let M be a monoid with a quadratic normalisation (Q, N) with complexity at most $(4, 3)$ and satisfying the unit condition.
Since $\mathcal{T}_{Q,N}$ and $\mathcal{M}_{Q,N}$ are dual automata, M possesses the explicitly dual properties of automaticity (automatic monoid) and self-similarity (automaton monoid).

Index of definitions

Index of results