Algebraic Approach to Automata Theory

Deepak D'Souza

Department of Computer Science and Automation
Indian Institute of Science, Bangalore.

11 September 2018

Outline

Algebraic approach to automata: Overview

- Defines language recognition via morphisms into a monoid.
- Analogous result to canonical automaton in the setting of monoids.
- Helps in characterising class of FO-definable languages.

Monoids

- A monoid is a structure $(M, \circ, 1)$, where
  - M is a base set containing the element "1",
  - $\circ$ is an associative binary operation on M, and
  - 1 is the identity element with respect to $\circ$.
- Examples of monoids: $(\mathbb{N}, +, 0)$, $(A^*, \cdot, \epsilon)$.
- Another Example: $(X \to X, \circ, id)$, where
  - $X \to X$ denotes the set of all functions from a set X to itself,
  - $f \circ g$ is function composition:

$$(f \circ g)(x) = g(f(x)).$$

Monoid morphisms

- A morphism from a monoid $(M, \circ_M, 1_M)$ to a monoid $(N, \circ_N, 1_N)$ is a map $\varphi : M \to N$, satisfying
  - $\varphi(1_M) = 1_N$, and,
  - $\varphi(m \circ_M m') = \varphi(m) \circ_N \varphi(m')$.
- Example: $\varphi : A^* \to \mathbb{N}$, given by

$$\varphi(w) = |w|$$

is a morphism from $(A^*, \cdot, \epsilon)$ to $(\mathbb{N}, +, 0)$.

Language recognition via monoid morphisms

- A language $L \subseteq A^*$ is said to be recognizable if there exists a monoid $(M, \circ, 1)$ and a morphism φ from $(A^*, \cdot, \epsilon)$ to $(M, \circ, 1)$, and a subset X of M such that

$$L = \varphi^{-1}(X).$$

- In this case, we say that the monoid M recognizes L.

Example of language recognition via monoid

Consider monoid $M = (\{1, m\}, \circ, 1\}$ where $\circ$ is given by:

$\circ$	1	m
1	1	m
m	m	m

Consider the morphism $\varphi : A^* \to M$ given by

$$\epsilon \mapsto 1$$
$$w \mapsto m \quad \text{for } w \in A^+.$$

Then M recognizes A^+, since $\varphi^{-1}(\{m\}) = A^+$.
M also recognizes $\{\epsilon\}$ (taking $X = \{1\}$), A^* (taking $X = \{1, m\}$), and $\emptyset$ (taking $X = \{\}$).

Example of language recognition via monoid

Consider monoid $M = (\{1, m\}, \circ, 1\}$ where $\circ$ is given by:

$\circ$	1	m
1	1	m
m	m	m

Consider the morphism $\varphi : A^* \to M$ given by

$$\begin{aligned} \epsilon &\mapsto 1 \\ w &\mapsto m \quad \text{for } w \in A^+. \end{aligned}$$

Then M recognizes A^+, since $\varphi^{-1}(\{m\}) = A^+$.
M also recognizes $\{\epsilon\}$ (taking $X = \{1\}$), A^* (taking $X = \{1, m\}$), and $\emptyset$ (taking $X = \{\}$).

Question: Is every language recognizable?

Exercise

Show that the language of odd a's over the alpbabet $A = \{a, b\}$ is recognizable.

Transition Monoid of a DA

Let $\mathcal{A} = (Q, s, \delta, F)$ be a deterministic automaton (DA).

- For $w \in A^*$, define $f_w : Q \to Q$ by

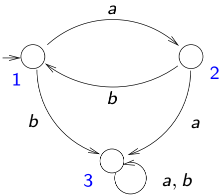
$$f_w(q) = \widehat{\delta}(q, w).$$

- Consider the monoid

$$M(\mathcal{A}) = (\{f_w \mid w \in A^*\}, \circ, 1).$$

- $M(\mathcal{A})$ is called the transition monoid of $\mathcal{A}$.

Example DA and Transition Monoid



Distinct elements of $M(\mathcal{A})$ are $\{f_\epsilon, f_a, f_b, f_{aa}, f_{ab}, f_{ba}\}$.

We write f_a as $\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 3 \end{pmatrix}$, or simply (2 3 3).

Question: If Q is finite, how many elements can $M(\mathcal{A})$ have?

Syntactic Monoid of a language

- Let $\mathcal{A}_{\equiv_L} = (Q, s, \delta, F)$ be the canonical automaton for a language $L \subseteq A^*$.
- The transition monoid of $\mathcal{A}_{\equiv_L}$ is called the syntactic monoid of L.
- We denote the syntactic monoid of L by $M(L)$.

Syntactic Congruence of a language

- Define an equivalence relation $\cong_L$ on A^*, induced by L, as

$$u \cong_L v \text{ iff } \forall x, y \in A^* : \; xuy \in L \text{ iff } xvy \in L.$$

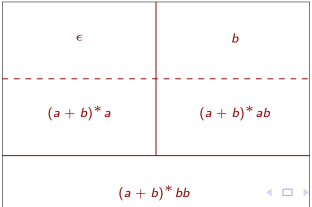
- $\cong_L$ is called the syntactic congruence of L.
- Check that $\cong_L$ is a two-sided congruence:
  - That is, $\cong_L$ is both a left-congruence (i.e $u \cong_L v$ implies $wu \cong_L wv$, for each $w \in A^*$) and a right-congruence (i.e $u \cong_L v$ implies $uw \cong_L vw$).
  - Equivalently, $u \cong_L u'$ and $v \cong_L v'$ imples $uv \cong_L u'v'$.
- $\cong_L$ refines the canonical MN relation, $\equiv_L$, for L.
- Example?

Syntactic Congruence of a language

- Define an equivalence relation $\cong_L$ on A^*, induced by L, as

$$u \cong_L v \text{ iff } \forall x, y \in A^* : \; xuy \in L \text{ iff } xvy \in L.$$

- $\cong_L$ is called the syntactic congruence of L.
- Check that $\cong_L$ is a two-sided congruence:
  - That is, $\cong_L$ is both a left-congruence (i.e $u \cong_L v$ implies $wu \cong_L wv$, for each $w \in A^*$) and a right-congruence (i.e $u \cong_L v$ implies $uw \cong_L vw$).
  - Equivalently, $u \cong_L u'$ and $v \cong_L v'$ imples $uv \cong_L u'v'$.
- $\cong_L$ refines the canonical MN relation, $\equiv_L$, for L.
- Example? Consider the language $(a+b)^*bb$:



Characterization of the syntactic monoid

Claim

For a canonical DA $\mathcal{A} = (Q, s, \delta, F)$,

$$f_u = f_v \text{ iff } u \cong_L v.$$

Syntactic monoid via syntactic congruence

For a language $L \subseteq A^*$, consider the monoid $A^*/\cong_L$, whose elements are equivalence classes under $\cong_L$, operation $\circ$ is given by

$$[u] \circ [v] = [uv],$$

and identity element is $[\epsilon]$.

Claim

The monoids $M(L)$ and $A^*/\cong_L$ are isomorphic.

(Use the morphism $f_w \mapsto [w]$.)

Algebraic definition of regular languages

Theorem

Let $L \subseteq A^*$. Then the following are equivalent:

1. *L is regular*
2. *The syntactic monoid of L, i.e. $M(L)$, is finite.*
3. *L is recognized by a finite monoid.*

Proof:

Algebraic definition of regular languages

Theorem

Let $L \subseteq A^*$. Then the following are equivalent:

1. *L is regular*
2. *The syntactic monoid of L, i.e. $M(L)$, is finite.*
3. *L is recognized by a finite monoid.*

Proof:

(1) $\implies$ (2): since $\mathcal{A}_{\equiv_L}$ is finite, and hence so is $M(L)$.

Algebraic definition of regular languages

Theorem

Let $L \subseteq A^*$. Then the following are equivalent:

1. *L is regular*
2. *The syntactic monoid of L, i.e. $M(L)$, is finite.*
3. *L is recognized by a finite monoid.*

Proof:

(1) $\implies$ (2): since $\mathcal{A}_{\equiv_L}$ is finite, and hence so is $M(L)$.

(2) $\implies$ (3): Define morphism $\varphi : A^* \to M(L)$, given by $w \mapsto f_w$.

Algebraic definition of regular languages

Theorem

Let $L \subseteq A^*$. Then the following are equivalent:

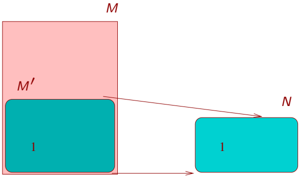
1. *L is regular*
2. *The syntactic monoid of L, i.e. $M(L)$, is finite.*
3. *L is recognized by a finite monoid.*

Proof:
(1) $\implies$ (2): since $\mathcal{A}_{\equiv_L}$ is finite, and hence so is $M(L)$.
(2) $\implies$ (3): Define morphism $\varphi : A^* \to M(L)$, given by $w \mapsto f_w$.
(3) $\implies$ (1): Let L be recognized by a finite monoid $(M, \circ, 1)$, via a morphism φ and $X \subseteq M$. Define a DFA $\mathcal{A} = (M, 1, \delta, X)$, where

$$\delta(m, a) = m \circ \varphi(a).$$

Canonicity of syntactic monoid/congruence

Let M and N be monoids. We say N divides M if there is a submonoid M' of M, and a surjective morphism from M' to N.



Theorem

Let $L \subseteq A^*$. Then L is recognized by a monoid M iff $M(L)$ divides M.

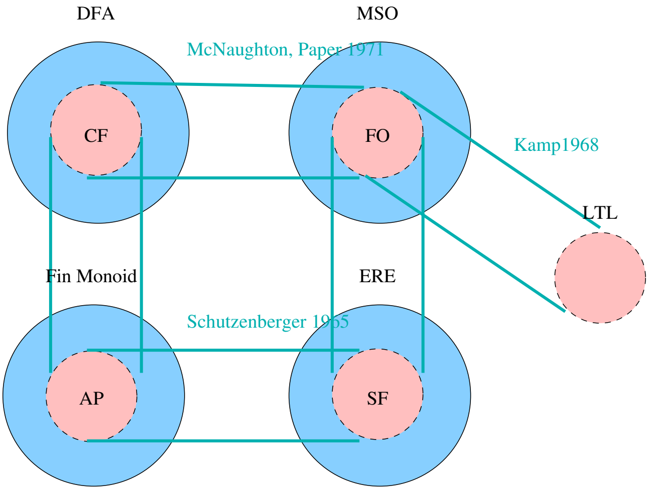
First-Order definable languages

Theorem (Schutzenberger (1965), McNaughton-Papert (1971))

Let $L \subseteq A^*$. Then the following are equivalent

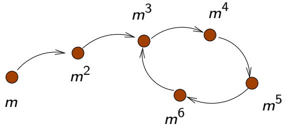
1. *L is definable in $FO(<)$.*
2. *L is accepted by a counter-free DFA.*
3. *$\mathcal{A}_{\equiv_L}$ is a counter-free DFA.*
4. *L is definable by a star-free extended regular expression.*
5. *L is recognized by an aperiodic finite monoid.*
6. *$M(L)$ is aperiodic.*

FO-definable languages

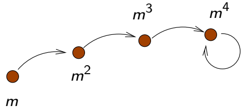


Definitions: Aperiodic Monoids

- A finite monoid is called aperiodic, if it does not contain a non-trivial group, or equivalently, for each element m in the monoid, $m^n = m^{n+1}$, for some $n > 0$.



Periodic



Aperiodic

- Examples:

∘	1	m
1	1	m
m	m	1

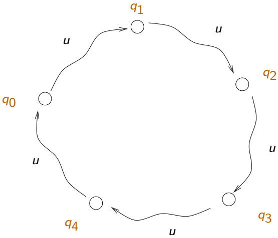
A periodic monoid.

∘	1	m
1	1	m
m	m	m

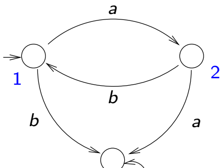
An aperiodic monoid.

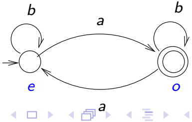
Definitions: Counter in a DFA

- A counter in a DFA $\mathcal{A} = (Q, s, \delta, F)$ is a string $u \in A^*$ and distinct states $q_0, q_1, \ldots, q_k$ in Q, with $k \geq 1$, such that for each i, $\widehat{\delta}(q_i, u) = q_{i+1}$, and $\widehat{\delta}(q_k, u) = q_0$.



- A DFA is counter-free if it does not have any counters.





Definitions: Star-Free Regular Expressions

- A star-free regular expression is an extended regular expression obtained using the syntax:

$$s ::= \emptyset \mid a \mid s + s \mid s \cdot s \mid s \cap s \mid \overline{s},$$

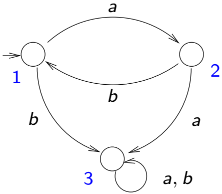
  where $a \in A$, and $\overline{s}$ denotes the language $A^* - L(s)$.
- A language $L \subseteq A^*$ is called star-free if there is a star-free regular expression defining it.
- For example the language A^* is star-free since the star-free expression $\overline{\emptyset}$ denotes it.

Illustrative example: $L = (ab)^*$

- FO(<) sentence for L:

$$\begin{array}{lll} \forall x & (& zero(x) \implies Q_a(x) \wedge \\ & & (Q_a(x) \implies \exists y(succ(x,y) \wedge Q_b(y))) \wedge \\ & & (Q_b(x) \wedge \neg last(x) \implies \exists y(succ(x,y) \wedge Q_a(y)))) \\ &) & \end{array}$$

- Counter-Free DFA for L:



- Star-Free ERE:

$$\{\epsilon\} \cup (aA^* \cap A^* b \cap \overline{A^*(aa+bb)A^*}).$$

Note that A^* is short-hand for $\overline{\emptyset}$ (what about $\{\epsilon\}$?).