Algebraic Approach to Automata Theory

Deepak D’Souza

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

11 September 2018
Outline

1. Overview
2. Recognition via monoid morphisms
3. Transition monoid
4. Syntactic Monoid
5. First-Order Definable Languages
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.
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: $(X \rightarrow X, \circ, id)$, where
- $X \rightarrow X$ 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 \(\varphi : M \rightarrow N\), satisfying
  - \(\varphi(1_M) = 1_N\), and,
  - \(\varphi(m \circ_M m') = \varphi(m) \circ_N \varphi(m')\).

- **Example**: \(\varphi : A^* \rightarrow \mathbb{N}\), given by
  \[\varphi(w) = |w|\]

  is a morphism from \((A^*, \cdot, \epsilon)\) to \((\mathbb{N}, +, 0)\).
A language $L \subseteq A^*$ is said to be recognizable if there exists a monoid $(M, \circ, 1)$ and a morphism $\varphi$ 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 $M = \langle \{1, m\}, \circ, 1 \rangle$ where $\circ$ is given by:

<table>
<thead>
<tr>
<th>$\circ$</th>
<th>1</th>
<th>m</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>m</td>
</tr>
<tr>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
</tbody>
</table>

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

- $\epsilon \mapsto 1$
- $w \mapsto m$ 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:

\[
\begin{array}{ccc}
\circ & 1 & m \\
1 & 1 & m \\
m & m & m \\
\end{array}
\]

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

\[
\begin{align*}
\epsilon & \mapsto 1 \\
w & \mapsto m & \text{for } w \in A^+.
\end{align*}
\]

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

- For $w \in A^*$, define $f_w : Q \rightarrow Q$ by
  \[ f_w(q) = \hat{\delta}(q, w). \]

- Consider the monoid
  \[ M(A) = (\{f_w \mid w \in A^*\}, \circ, 1). \]

- $M(A)$ is called the **transition monoid** of $A$. 
Distinct elements of $M(\mathcal{A})$ are \{$f_\varepsilon, f_a, f_b, f_{aa}, f_{ab}, f_{ba}$\}.

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?
Let $A_{\equiv L} = (Q, s, \delta, F)$ be the canonical automaton for a language $L \subseteq A^*$. The transition monoid of $A_{\equiv L}$ is called the syntactic monoid of $L$. We denote the syntactic monoid of $L$ by $M(L)$. 
Overview Recognition via monoid morphisms Transition monoid Syntactic Monoid First-Order Definable Languages

Syntactic Congruence of a language

- Define an equivalence relation \( \equiv_L \) on \( A^* \), induced by \( L \), as
  \[
  u \equiv_L v \text{ iff } \forall x, y \in A^*: xuy \in L \text{ iff } xvy \in L.
  \]
- \( \equiv_L \) is called the **syntactic congruence** of \( L \).
- Check that \( \equiv_L \) is a **two-sided congruence**:
  - That is, \( \equiv_L \) is both a **left-congruence** (i.e \( u \equiv_L v \) implies \( wu \equiv_L wv \), for each \( w \in A^* \)) and a **right-congruence** (i.e \( u \equiv_L v \) implies \( uw \equiv_L vw \)).
  - Equivalently, \( u \equiv_L u' \) and \( v \equiv_L v' \) imples \( uv \equiv_L u'v' \).
- \( \equiv_L \) refines the canonical MN relation, \( \equiv_L \), for \( L \).
- Example?
Syntactic Congruence of a language

- Define an equivalence relation $\equiv_L$ on $A^*$, induced by $L$, as
  $$u \equiv_L v \iff \forall x, y \in A^*: xuy \in L \iff xvy \in L.$$  
- $\equiv_L$ is called the **syntactic congruence** of $L$.
- Check that $\equiv_L$ is a **two-sided congruence**:
  - That is, $\equiv_L$ is both a **left-congruence** (i.e. $u \equiv_L v$ implies $wu \equiv_L vw$, for each $w \in A^*$) and a **right-congruence** (i.e. $u \equiv_L v$ implies $uw \equiv_L vw$).
  - Equivalently, $u \equiv_L u'$ and $v \equiv_L v'$ imples $uv \equiv_L u'v'$.
- $\equiv_L$ refines the canonical MN relation, $\equiv_L$, for $L$.
- Example? Consider the language $(a + b)^*bb$:

```
   ε  b
----  ----
(a + b)*a  (a + b)*ab
----  ----
(a + b)*bb
```
Characterization of the syntactic monoid

Claim

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

$$f_u = f_v \iff u \equiv_L v.$$  

Proof:

By definition the element $f_u$ of the transition monoid of the canonical DA is $\hat{\delta}(\_, u)$.

$xuy \in L$ iff $(f_x \circ f_u \circ f_y)(s) \in F$ (in other words, $\hat{\delta}(s, xuy) \in F$) in the canonical DA iff $(f_x \circ f_v \circ f_y)(s) \in F$ in the canonical DA iff

$xvy \in L$

Thus the syntactic congruence $u \equiv_L v$ matches the equality $f_u = f_v$ derived from the canonical DA.
Syntactic monoid via syntactic congruence

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

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

and identity element is $[\epsilon]$.

**Claim**

The monoids $M(L)$ and $A^*/\sim_L$ 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:
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) \( \Rightarrow \) (2): since \( A_{\equiv_L} \) 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 $A^*$ is finite, and hence so is $M(L)$.

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


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 $A_{\equiv_L}$ 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 $\varphi$ and $X \subseteq M$. Define a DFA $A = (M, 1, \delta, X)$, 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 $M$.
Theorem (Schützenberger (1965), McNaughton-Papert (1971))

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

1. \( L \) is definable in \( \text{FO}(\text{<}) \).
2. \( L \) is accepted by a counter-free DFA.
3. \( A \equiv_L \) 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

- DFA
- MSO
- CF
- AP
- FO
- Fin Monoid
- ERE
- SF
- LTL
- Schutzenberger 1965
- McNaughton, Paper 1971
- Kamp 1968
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 $n > 0$.

Examples:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$\cdot$</td>
<td>1</td>
<td>$m$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>$m$</td>
</tr>
<tr>
<td>$m$</td>
<td>$m$</td>
<td>1</td>
</tr>
</tbody>
</table>

A periodic monoid.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$\cdot$</td>
<td>1</td>
<td>$m$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>$m$</td>
</tr>
<tr>
<td>$m$</td>
<td>$m$</td>
<td>$m$</td>
</tr>
</tbody>
</table>

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 $i$, $\hat{\delta}(q_i, u) = q_{i+1}$, and $\hat{\delta}(q_k, u) = q_0$.

- 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 \( L \):
  \[
  \forall x \ ( \text{zero}(x) \implies Q_a(x) \land \\
  (Q_a(x) \implies \exists y (\text{succ}(x, y) \land Q_b(y))) \land \\
  (Q_b(x) \land \neg \text{last}(x) \implies \exists y (\text{succ}(x, y) \land Q_a(y))))
  \]

- 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 \( \{ \epsilon \} \)?).