Advanced Automata Theory 8
Groups, Monoids and Automata Theory

Frank Stephan
Department of Computer Science
Department of Mathematics
National University of Singapore
fstephan@comp.nus.edu.sg
Lexicographical Order
\[ a_1 a_2 \ldots a_n <_{\text{lex}} b_1 b_2 \ldots b_m \] iff either the first string is a proper prefix of the second or there is a \( k \) with \( k \leq n \wedge k \leq m \wedge a_k < b_k \wedge a_h = b_h \) for all \( h \) with \( 1 \leq h < k \).

\( \text{NUH} <_{\text{lex}} \text{NUHS} <_{\text{lex}} \text{NUS} <_{\text{lex}} \text{SOC} \).

Algorithm
Processing inputs \( x, y \) symbol by symbol.
1. If \( x \) is exhausted and \( y \) not, then \( x <_{\text{lex}} y \), Halt.
2. If \( y \) is exhausted and \( x \) not, then \( y <_{\text{lex}} x \), Halt.
3. If \( x \) and \( y \) are exhausted, then \( x = y \), Halt.
4. Read symbol \( a \) from \( x \) and \( b \) from \( y \).
5. If \( a = b \) then go to 1.
6. If \( a < b \) then \( x <_{\text{lex}} y \) else \( y <_{\text{lex}} x \). Halt.
Relations are sets of tuples of strings; they can be translated into sets of strings using convolution.

Convolution has combined characters of matching positions in the words with # used for exhausted words (# is not in the alphabet).

$$\text{conv}(00110, 0123456789) =$$

$$\begin{pmatrix}
0 \\
0 \\
1 \\
2 \\
1 \\
3 \\
0 \\
4 \\
5 \\
# \\
#
\end{pmatrix}\begin{pmatrix}
# \\
# \\
# \\
# \\
# \\
# \\
9
\end{pmatrix}.$$ 

Now one can formalise automaticity of a relation using a convolution.

For example, a ternary relation $R$ of words over $\Sigma$ is automatic iff the set $\{\text{conv}(x, y, z) : (x, y, z) \in R\}$ is regular.
For binary alphabet \( \{0, 1\} \), the following automaton recognises lexicographic ordering.

Here \( \binom{a}{b} \) on an arrow means that the automaton always goes this way.
Repetition 4

Theorem
If a relation or function is first-order definable from automatic parameters then it is automatic.

Example
Length-lexicographic ordering:

\[ x <_{ll} y \iff |x| < |y| \lor (|x| = |y| \land x <_{\text{lex}} y). \]

Length-lexicographic successor:

\[ y = \text{Succ}(x) \iff x <_{ll} y \land \forall z [z <_{ll} x \lor z = x \lor z = y \lor y <_{ll} z]. \]

Range \( R \) of a function \( f \) with domain \( D \):

\[ y \in R \iff \exists x [x \in D \land y = f(x)]. \]
The derivation relation \( \Rightarrow \) in a grammar is an automatic relation.

One can also characterise the context-sensitive languages using automatic relations.

This characterisation is used to show that the complement of a context-sensitive language is again context-sensitive. This result is due to Neil Immerman and Róbert Szelepcsényi.
Groups, Monoids and Semigroups

Groups, monoids and semigroups are mathematical objects related to automata theory in two ways:

- The derivations of a regular language can be made into a monoid;
- Many groups, monoids and semigroups can be represented by automata in various ways.

There are prominent examples of groups: the integers \((\mathbb{Z}, +, 0)\) and the rationals \((\mathbb{Q}, +, 0)\); furthermore, permutation groups or the group of all possible move-sequences on Rubik’s cube (modulo equivalence).
Definition of Groups

Let $G$ be a set and $\circ$ be an operation mapping $G \times G$ to $G$.

(a) The structure $(G, \circ)$ is called a semigroup iff $\circ$ is associative, that is, iff $x \circ (y \circ z) = (x \circ y) \circ z$ for all $x, y, z \in G$.

(b) The structure $(G, \circ, e)$ is called a monoid iff $(G, \circ)$ is a semigroup and $e \in G$ and $e$ satisfies $x \circ e = e \circ x = x$ for all $x \in G$.

(c) The structure $(G, \circ, e)$ is called a group iff $(G, \circ, e)$ is a monoid and for each $x \in G$ there is an $y \in G$ with $x \circ y = e$.

(d) A semigroup $(G, \circ)$ is called finitely generated iff there is a finite subset $F \subseteq G$ such that for each $x \in G$ there are $n$ and $y_1, y_2, \ldots, y_n \in F$ with $x = y_1 \circ y_2 \circ \ldots \circ y_n$.

(e) A semigroup $(G, \circ)$ is finite iff $G$ is finite as a set.
Examples 8.2 – 8.4

8.2 Prominent Sets

\((\{1, 2, 3, \ldots\}, +)\) is semigroup but not monoid.

\((\mathbb{N}, +, 0)\) is a monoid.

\((\mathbb{Z}, +, 0)\) is a finitely generated group.

\((\mathbb{Q}, +, 0)\) is a group but not finitely generated.

\((\mathbb{Q} - \{0\}, *, 1)\) is a group.

8.3 Semigroup

Let \(G\) have at least two elements and \(x \circ y = x\) for all \(x, y \in G\). This is a semigroup and every element is neutral from one side but not from the other.

8.4 Functions on Set

Let \(Q\) be a finite set and \(G = \{f : Q \to Q\}\) and \(\circ\) be the concatenation of functions, \(\text{id}\) the identity. \((G, \circ, \text{id})\) is a finite monoid. Let \(G' = \{f \in G : f \text{ is one-one}\}\); the submonoid \((G', \circ, \text{id})\) is a group.
8.5 Syntactic Monoid

Let \((Q, \Sigma, \delta, s, F)\) be a complete dfa, \(G = \{f : Q \rightarrow Q\}\) and \(\text{id}\) the identity function. Then

\[
G' = \{f \in G : \exists w \in \Sigma^* \forall q \in Q [\delta(q, w) = f(q)]\}
\]

defines the syntactic monoid \((G', \circ, \text{id})\) with \(\circ\) being function concatenation. Let \(f_w\) be the function generated by \(w\).

The relation \(\sim\) on \(\Sigma^*\) with \(v \sim w\) iff \(f_v = f_w\) is called a congruence: It is an equivalence relation which respects the concatenation; that is, if \(v \sim w\) and \(x \sim y\) then \(vx \sim wy\).

Theorem

A language \(L\) is regular iff it is the union of equivalence classes of a congruence with finitely many equivalence classes.


Word Problem of Semigroups

Let \((G, \circ)\) be a semigroup and \(F \subseteq G\). Let \(w \in F^*\) be a string over \(G\), say \(w = a_1 a_2 \ldots a_n\). Then \(e_{lG}(w)\) is the group element \(a_1 \circ a_2 \circ \ldots \circ a_n\).

Generators

\(F\) is called a set of generators of \(G\) iff for every \(v \in G\) exists \(w \in F^*\) with \(v = e_{lG}(w)\).

Word Problem

The word problem of a semigroup \(G\) over a set \(F\) of generators asks for an algorithm which checks for two \(v, w \in F^*\) whether \(e_{lG}(v) = e_{lG}(w)\)
A Thurston automatic semigroup is a finitely generated semigroup which is represented by a \( G \subseteq F^* \) such that the following conditions hold:

- \( G \) is a regular subset of \( F^* \);
- Each element of the semigroup has exactly one representative in \( G \);
- For each \( y \in G \) the mapping \( x \mapsto \text{el}_G(xy) \) is automatic.

Similarly for Thurston automatic monoids and groups.
Example

\( F = \{ \overline{a}, a \} \) and \( G = a^* \cup \overline{a}^* \). Then \((G, \circ, \varepsilon)\) is a Thurston automatic group with

\[
\begin{align*}
  a^n \circ a^m &= a^{n+m}; \\
  \overline{a}^n \circ a^m &= \overline{a}^{n+m}; \\
  a^n \circ \overline{a}^m &= \begin{cases} 
    a^{n-m} & \text{if } n > m; \\
    \varepsilon & \text{if } n = m; \\
    \overline{a}^{m-n} & \text{if } n < m;
  \end{cases} \\
  \overline{a}^n \circ a^m &= \begin{cases} 
    \overline{a}^{n-m} & \text{if } n > m; \\
    \varepsilon & \text{if } n = m; \\
    a^{m-n} & \text{if } n < m.
  \end{cases}
\]

This group represents \((\mathbb{Z}, +)\) with generator \(a\) for \(+1\) and \(\overline{a}\) for \(-1\).
Example 8.10

Semigroups can be described by rules. For example, let $F = \{\bar{a}, a, b, c\}$ and consider $G = (a^* \cup \bar{a}^*) \cdot \{b, c\}^*$ with rules $baa = ab$, $c = ba$, $a\bar{a} = \varepsilon$ and $\bar{a}a = \varepsilon$.

Note that words like $aaabaaa$ can be brought into the normal form in $G$ by applying the rules:

$aaabaaa = aaaaaba = aaaaac$.

This permits to eliminate $a$ after $b$ and $c$.

Similarly with $\bar{a}$: $b\bar{a} = \bar{a}ab\bar{a} = \bar{a}baa\bar{a} = \bar{a}ba = \bar{a}c$.

This semigroup is Thurston automatic.
Exercise 8.11

Let $G = a^* b^*$ define a monoid with generators $a, b$, neutral element $\varepsilon$ and $\circ$ be defined by $a^i b^j \circ a^{i'} b^{j'} = a^{i+b^{j+j'}}$ for $j > 0$ and $a^i \circ a^{i'} b^{j'} = a^{i+i'} b^{j'}$.

Show that the semigroup in this representation is Thurston automatic but not Thurston bi-automatic.

Does the monoid have a Thurston bi-automatic representation?

Use adequate pumping lemmas to prove the result.
Hodgson and independently Khoussainov and Nerode formalised automatic semigroups as follows.

- The domain $G$ is regular and represents each semigroup element exactly once;
- The function $x, y \mapsto x \circ y$ is automatic.

Some Thurston automatic semigroups can also be made automatic in the sense of Hodgson, Khoussainov and Nerode; the representation has, somehow, to be adjusted.

$(\mathbb{N}, +, 0)$ is an automatic monoid in a suitable representation;
$(\mathbb{Q}, +, 0)$ is not an automatic group (in this sense);
If $(F, +, 0)$ is a finite group then the set $G = \{ f : \mathbb{N} \to F, f \text{ is eventually constant} \}$ has an automatic representation.
Exercise 8.13 (Quiz)

Represent \((\mathbb{Z}, +, 0)\) using alphabet \(\Sigma = \{0, 1, +, -\}\) with 0 representing 0 and \(a_0a_1\ldots a_n+\) with \(a_n = 1\) and \(a_0, a_1, \ldots, a_{n-1} \in \{0, 1\}\) representing \(a_0 + 2a_1 + 4a_2 + \ldots + 2^n a_n\). Accordingly \(a_0a_1\ldots a_n-\) with \(a_n = 1\) and \(a_0, a_1, \ldots, a_{n-1} \in \{0, 1\}\) represents \(- (a_0 + 2a_1 + 4a_2 + \ldots + 2^n a_n)\).

Why is the addition automatic?

Why is the order of the binary digits inverted?
Theorem 8.15

Theorem. There is a monoid which is Thurston automatic but not automatic in the sense of Hodgson, Khoussainov and Nerode.

Monoid. \((\{0, 1\}^*, \circ, \varepsilon)\) with \(\circ\) being concatenation of strings. This monoid is Thurston automatic.

Assume that there is an automatic presentation of this monoid. Let \(F_0\) be the representatives of \(\{0, 1\}\) and \(F_{n+1} = \{v \circ w : v, w \in F_n\}\).

There is a constant \(c\) such that the members of \(F_n\) have at most length \(c \cdot (n + 1)\).

There are at most \(|\Sigma|^{1+c \cdot (n+1)}\) strings of length up to \(c \cdot (n + 1)\). There are \(2^{2^n}\) elements in \(F_n\). A contradiction.
Exercise 8.16

Free Group over two generators.
Let $\Sigma = \{a, \bar{a}, b, \bar{b}\}$. The free group with two generators can be represented with $G = \{w \in \Sigma^* : a\bar{a}, \bar{a}a, b\bar{b}, \bar{b}b$ are not substrings of $w\}$. The group operation $x \circ y$ with $x, y \in G$ takes as value the $z$ obtained by removing from the concatenation $xy$ all occurrences of $a\bar{a}, \bar{a}a, b\bar{b}, \bar{b}b$ until none of them are left. $\epsilon$ is the neutral element.

This group is Thurston automatic in this representation.

Show that this group is not automatic (in the sense of Hodgson, Khoussainov and Nerode).
Exercise 8.17

Let $a, b$ be generators of a group satisfying

$$a^h b^k \circ a^i b^j = \begin{cases} a^{h+i} b^{k+j} & \text{if } k \text{ is even;} \\ a^{h-i} b^{k+j} & \text{if } k \text{ is odd;} \end{cases}$$

where $h, k, i, j \in \mathbb{Z}$. Show that this group is Thurston bi-automatic as well as automatic; find for both results representations.

For automaticity, one can use (as parameter) a representation of $(\mathbb{Z}, +, 0)$. 