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

Repetition 1

Lexicographical Order

$a_1 a_2 \ldots a_n <_{lex} b_1 b_2 \ldots b_m$ 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 $1 \leq h < k$.

$NUH <_{lex} NUHS <_{lex} NUS <_{lex} SOC$.

Algorithm

Processing inputs x, y symbol by symbol.

1. If x is exhausted and y not, then $x <_{lex} y$, Halt.
2. If y is exhausted and x not, then $y <_{lex} x$, 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 $x <_{lex} y$ else $y <_{lex} x$. Halt.

Repetition 2

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).

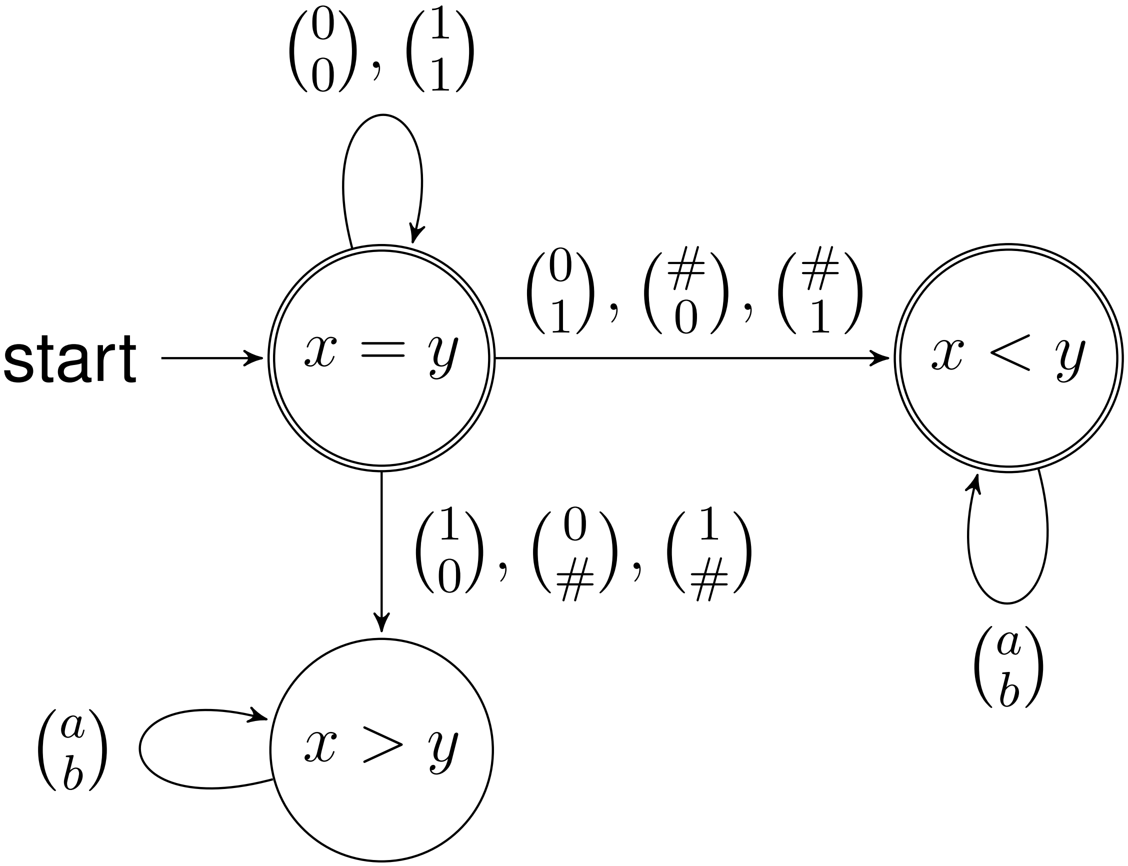
$\mathbf{conv(00110, 0123456789) =}$
$\binom{\mathbf{0}}{\mathbf{0}}\binom{\mathbf{0}}{\mathbf{1}}\binom{\mathbf{1}}{\mathbf{2}}\binom{\mathbf{1}}{\mathbf{3}}\binom{\mathbf{0}}{\mathbf{4}}\binom{\#}{\mathbf{5}}\binom{\#}{\mathbf{6}}\binom{\#}{\mathbf{7}}\binom{\#}{\mathbf{8}}\binom{\#}{\mathbf{9}}$.

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

For example, a ternary relation $\mathbf{R}$ of words over $\mathbf{\Sigma}$ is automatic iff the set $\{\mathbf{conv(x, y, z)} : \mathbf{(x, y, z)} \in \mathbf{R}\}$ is regular.

Repetition 3

For binary alphabet $\{0, 1\}$, the following automaton recognises lexicographic ordering.


$\binom{0}{0}, \binom{1}{1}$
start
$x = y$
$\binom{0}{1}, \binom{\#}{0}, \binom{\#}{1}$
$x < y$
$\binom{1}{0}, \binom{0}{\#}, \binom{1}{\#}$
$\binom{a}{b}$
$\binom{a}{b}$
$x > y$

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:

$$\mathbf{x} <_{\mathbf{ll}} \mathbf{y} \Leftrightarrow |\mathbf{x}| < |\mathbf{y}| \vee (|\mathbf{x}| = |\mathbf{y}| \wedge \mathbf{x} <_{\mathbf{lex}} \mathbf{y}).$$

Length-lexicographic successor:

$$\mathbf{y} = \mathbf{Succ}(\mathbf{x}) \Leftrightarrow \mathbf{x} <_{\mathbf{ll}} \mathbf{y} \wedge \forall \mathbf{z}\, [\mathbf{z} <_{\mathbf{ll}} \mathbf{x} \vee \mathbf{z} = \mathbf{x} \vee \mathbf{z} = \mathbf{y} \vee \mathbf{y} <_{\mathbf{ll}} \mathbf{z}].$$

Range $\mathbf{R}$ of a function $\mathbf{f}$ with domain $\mathbf{D}$:

$$\mathbf{y} \in \mathbf{R} \Leftrightarrow \exists \mathbf{x}\, [\mathbf{x} \in \mathbf{D} \wedge \mathbf{y} = \mathbf{f}(\mathbf{x})].$$

Repetition 5

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 derivatives 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 $(\mathbb{Z}, +, \mathbf{0})$ and the rationals $(\mathbb{Q}, +, \mathbf{0})$; furthermore, permutation groups or the group of all possible move-sequences on Rubik's cube (modulo equivalence).

Definition of Groups

Let $\mathbf{G}$ be a set and $\circ$ be an operation mapping $\mathbf{G} \times \mathbf{G}$ to $\mathbf{G}$.
(a) The structure $(\mathbf{G}, \circ)$ is called a semigroup iff $\circ$ is associative, that is, iff $\mathbf{x} \circ (\mathbf{y} \circ \mathbf{z}) = (\mathbf{x} \circ \mathbf{y}) \circ \mathbf{z}$ for all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbf{G}$.
(b) The structure $(\mathbf{G}, \circ, \mathbf{e})$ is called a monoid iff $(\mathbf{G}, \circ)$ is a semigroup and $\mathbf{e} \in \mathbf{G}$ and $\mathbf{e}$ satisfies $\mathbf{x} \circ \mathbf{e} = \mathbf{e} \circ \mathbf{x} = \mathbf{x}$ for all $\mathbf{x} \in \mathbf{G}$.
(c) The structure $(\mathbf{G}, \circ, \mathbf{e})$ is called a group iff $(\mathbf{G}, \circ, \mathbf{e})$ is a monoid and for each $\mathbf{x} \in \mathbf{G}$ there is an $\mathbf{y} \in \mathbf{G}$ with $\mathbf{x} \circ \mathbf{y} = \mathbf{e}$.
(d) A semigroup $(\mathbf{G}, \circ)$ is called finitely generated iff there is a finite subset $\mathbf{F} \subseteq \mathbf{G}$ such that for each $\mathbf{x} \in \mathbf{G}$ there are $\mathbf{n}$ and $\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_n} \in \mathbf{F}$ with $\mathbf{x} = \mathbf{y_1} \circ \mathbf{y_2} \circ \ldots \circ \mathbf{y_n}$.
(e) A semigroup $(\mathbf{G}, \circ)$ is finite iff $\mathbf{G}$ 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 moniod.
$(\mathbb{Z}, +, 0)$ 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, id the identity. $(G, \circ, \mathrm{id})$ is a finite monoid. Let $G' = \{f \in G : f$ is one-one$\}$; the submonoid $(G', \circ, \mathrm{id})$ is a group.

8.5 Syntactic Monoid

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

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

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

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

Theorem

A language $\mathbf{L}$ is regular iff it is the union of equivalence classes of a congruence with finitely many equivalence classes.

Example 8.7

Let a dfa have the alphabet $\Sigma = \{0, 1\}$ and states $Q = \{s, s', q, q', q''\}$ and the following state transition table:

state	s	s'	q	q'	q''
successor at 0	s'	s	q	q'	q''
successor at 1	q'	q'	q'	q''	q

Now the syntactic monoid contains the transition functions f_ε (the identity), f_0, f_1, f_{11} and f_{111}. The functions f_0 and f_1 are as in the state transition diagramme and f_{11} and f_{111} are given by the following table.

state	s	s'	q	q'	q''
f_{11}	q''	q''	q''	q	q'
f_{111}	q	q	q	q'	q''

Other functions are equal to these, for example $f_{110} = f_{11}$ and $f_{0000} = f_\varepsilon$.

Word Problem of Semigroups

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

Generators

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

Definition 8.8: Word Problem

The word problem of a semigroup $\mathbf{G}$ over a set $\mathbf{F}$ of generators asks for an algorithm which checks for two $\mathbf{v}, \mathbf{w} \in \mathbf{F}^*$ whether $\mathbf{el_G(v)} = \mathbf{el_G(w)}$.

Application

Definition 8.9. A language $\mathbf{L}$ defines the congruence $\{(\mathbf{v},\mathbf{w}) : \mathbf{L_v} = \mathbf{L_w}\}$ and furthermore $\mathbf{L}$ also defines the syntactic monoid $(\mathbf{G_L},\circ)$ of its minimal deterministic finite automaton.

Theorem 8.10. Every finite group $(\mathbf{G},\circ)$ is the syntactic monoid of a language $\mathbf{L}$.

Proof. Let $\mathbf{L} = \{\mathbf{v} \in \mathbf{G}^* : \mathbf{el_G}(\mathbf{v}) = \varepsilon\}$ be the set of words which are equal to the neutral element.

The corresponding dfa is $(\mathbf{G},\mathbf{G},\delta,\varepsilon,\{\varepsilon\})$ with $\delta(\mathbf{a},\mathbf{b}) = \mathbf{a}\circ\mathbf{b}$ for all $\mathbf{a},\mathbf{b} \in \mathbf{G}$. For each $\mathbf{a} \in \mathbf{G}$, the inverse $\mathbf{b}$ of $\mathbf{a}$ satisfies that $\mathbf{b}$ is the only one-letter word in $\mathbf{G}^*$ such that $\mathbf{L_{ab}} = \mathbf{L}_\varepsilon$ and therefore, for all $\mathbf{a} \in \mathbf{G}$, the languages $\mathbf{L_a}$ are different. Thus the dfa is a minimal dfa and has a congruence $\sim$ satisfying $\mathbf{v} \sim \mathbf{w}$ iff $\mathbf{el_G}(\mathbf{v}) = \mathbf{el_G}(\mathbf{w})$ iff $\mathbf{L_v} = \mathbf{L_w}$.

When $(\mathbf{G}, \circ)$ differs from all $(\mathbf{G_L}, \circ)$

Let $(\{\varepsilon, \mathbf{0}, \mathbf{1}, \mathbf{2}\}, \circ)$ be the semigroup with $\mathbf{a} \circ \varepsilon = \varepsilon \circ \mathbf{a} = \mathbf{a}$ and $\mathbf{a} \circ \mathbf{b} = \mathbf{b}$ for all $\mathbf{a}, \mathbf{b} \in \{\mathbf{0}, \mathbf{1}, \mathbf{2}\}$.

The set $\{\mathbf{0}, \mathbf{1}, \mathbf{2}\}$ generates the given monoid. Consider all words over $\{\mathbf{0}, \mathbf{1}, \mathbf{2}\}^*$. Two such words $\mathbf{v}, \mathbf{w}$ define the same member of the semigroup iff either both $\mathbf{v}, \mathbf{w}$ are empty or both $\mathbf{v}, \mathbf{w}$ end with the same digit $\mathbf{a} \in \{\mathbf{0}, \mathbf{1}, \mathbf{2}\}$.

The monoid is the syntactic monoid of the dfa which has alphabet $\{\mathbf{0}, \mathbf{1}, \mathbf{2}\}$, states $\{\mathbf{s}, \mathbf{0}, \mathbf{1}, \mathbf{2}\}$, start state $\mathbf{s}$ and transition function $\delta(\mathbf{q}, \mathbf{a}) = \mathbf{a}$ for all symbols $\mathbf{a}$ and states $\mathbf{q}$.

However there is no language $\mathbf{L}$ with the syntactic monoid being equal to $(\mathbf{G_L}, \circ)$. The reason is that for two different digits $\mathbf{a}, \mathbf{b}$, the corresponding states with the same name are either both accepting or both rejecting and thus the minimal automaton of the dfa belonging to this monoid unifies these two states into one.

Exercises of Syntactic Monoids

Determine the syntactic monoids $(\mathbf{G_{L_k}}, \circ)$ for the following languages $\mathbf{L_k}$.

Exercise 8.12: Do this for $\mathbf{L_1}$ and $\mathbf{L_2}$:

1. $\mathbf{L_1 = \{0^n 1^m : n + m \leq 3\}}$;
2. $\mathbf{L_2 = \{w : w}$ has an even number of $\mathbf{0}$ and an odd number of $\mathbf{1\}}$.

Exercise 8.13: Do this for $\mathbf{L_3}$ and $\mathbf{L_4}$:

1. $\mathbf{L_3 = \{00\}^* \cdot \{11\}^*}$;
2. $\mathbf{L_4 = \{0, 1\}^* \cdot \{00, 11\}}$.

Quiz 8.14

Do the same for $\mathbf{L_5 = \{0\}^* \cdot \{1\}^*}$.

Automatic Semigroups

An automatic semigroup (as introduced by Epstein, Cannon, Holt, Levy, Paterson and Thurston in 1992) is a finitely generated semigroup which is represented by a $\mathbf{G} \subseteq \mathbf{F}^*$ such that the following conditions hold:

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

Similarly for automatic monoids and groups.

Example 8.16

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

$$\begin{array}{rcl}
\mathbf{a}^{\mathbf{n}} \circ \mathbf{a}^{\mathbf{m}} & = & \mathbf{a}^{\mathbf{n}+\mathbf{m}}; \\
\overline{\mathbf{a}}^{\mathbf{n}} \circ \overline{\mathbf{a}}^{\mathbf{m}} & = & \overline{\mathbf{a}}^{\mathbf{n}+\mathbf{m}}; \\
\mathbf{a}^{\mathbf{n}} \circ \overline{\mathbf{a}}^{\mathbf{m}} & = & \begin{cases} \mathbf{a}^{\mathbf{n}-\mathbf{m}} & \text{if } \mathbf{n} > \mathbf{m}; \\ \varepsilon & \text{if } \mathbf{n} = \mathbf{m}; \\ \overline{\mathbf{a}}^{\mathbf{m}-\mathbf{n}} & \text{if } \mathbf{n} < \mathbf{m}; \end{cases} \\
\overline{\mathbf{a}}^{\mathbf{n}} \circ \mathbf{a}^{\mathbf{m}} & = & \begin{cases} \overline{\mathbf{a}}^{\mathbf{n}-\mathbf{m}} & \text{if } \mathbf{n} > \mathbf{m}; \\ \varepsilon & \text{if } \mathbf{n} = \mathbf{m}; \\ \mathbf{a}^{\mathbf{m}-\mathbf{n}} & \text{if } \mathbf{n} < \mathbf{m}. \end{cases}
\end{array}$$

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

Example 8.17

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

Note that words like $\mathbf{aaabaaa}$ can be brought into the normal form in $\mathbf{G}$ by applying the rules: $\mathbf{ca} = \mathbf{baa} = \mathbf{ab}$ and $\mathbf{ba} = \mathbf{c}$ and $\mathbf{aaabaaa} = \mathbf{aaaaba} = \mathbf{aaaac}$.
This permits to eliminate $\mathbf{a}$ after $\mathbf{b}$ and $\mathbf{c}$.
Similarly with $\bar{\mathbf{a}}$: $\mathbf{b}\bar{\mathbf{a}} = \bar{\mathbf{a}}\mathbf{ab}\bar{\mathbf{a}} = \bar{\mathbf{a}}\mathbf{baa}\bar{\mathbf{a}} = \bar{\mathbf{a}}\mathbf{ba} = \bar{\mathbf{a}}\mathbf{c}$ and $\mathbf{c}\bar{\mathbf{a}} = \mathbf{ba}\bar{\mathbf{a}} = \mathbf{b}$.

This semigroup is automatic.

Exercise 8.18

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

Show that the monoid in this representation is automatic but not biautomatic.

Does the monoid have a biautomatic representation?

Use adequate pumping lemmas to prove the result.

Automatic Semigroups by Hodgson

Hodgson and independently Khoussainov and Nerode formalised fully automatic semigroups as follows (they used the term “automatic semigroup” and “fully” is added here to indicate that the full group operation is automatic).

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

Some automatic semigroups can also be made fully automatic; the representation has, however, to be adjusted.

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

Quiz 8.20

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 fully automatic?

Why is the order of the binary digits inverted?

Automaton

Verification of $\mathbf{a}+\mathbf{b}=\mathbf{c}$ with $\mathbf{a},\mathbf{b},\mathbf{c} \geq \mathbf{0}$. For ease of notation, identify $\#$ and $\mathbf{0}$ and omit $+$. Three states: $\mathbf{n},\mathbf{c},\mathbf{e}$ for no carry (accepting, start), carry (rejecting) and error (rejecting).

Assume that input is $(\mathbf{a_k},\mathbf{b_k},\mathbf{c_k})$ in $\mathbf{k}$-th step.
From $\mathbf{n}$, if $\mathbf{a_k}+\mathbf{b_k}=\mathbf{c_k}$ then go to $\mathbf{n}$;
if $\mathbf{a_k}+\mathbf{b_k}=\mathbf{2}+\mathbf{c_k}$ then go to $\mathbf{c}$; otherwise go to $\mathbf{e}$.
From $\mathbf{c}$, if $\mathbf{a_k}+\mathbf{b_k}+\mathbf{1}=\mathbf{c_k}$ then go to $\mathbf{n}$;
if $\mathbf{a_k}+\mathbf{b_k}+\mathbf{1}=\mathbf{2}+\mathbf{c_k}$ then go to $\mathbf{c}$; otherwise go to $\mathbf{e}$.
From $\mathbf{e}$ one can only go to $\mathbf{e}$.

```
a  |   1   1   1   0   |    1   0   1   1   0
b  |   1   1   0   0   |    0   1   0   1   0
c  |   0   1   0   0   |    1   1   1   0   1
st | n   c   c   c   e |  n   n   n   n   c   n
       7 + 3 = 2?           13 + 10 = 23?
```

Theorem 8.22

Theorem. There is a monoid which is automatic but not fully automatic.

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

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

There is a constant $c \geq \max\{|x| : x \in F_0\}$ such that, for all $x, y \in G$, $|x \circ y| \leq \max\{|x|, |y|\} + c$. By induction, the members of F_n have as members of G at most length $c \cdot (n+1)$; but they represent binary strings of length 2^n.

There are at most $|\Sigma|^{1+c\cdot(n+1)}$ members of G of length up to $c \cdot (n+1)$, where G uses alphabet Σ. There are 2^{2^n} elements in F_n. A contradiction.

Exercise 8.23

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

This group is automatic in this representation.

Show that this group is not fully automatic.

Example 8.24

Let $\mathbf{a}, \mathbf{b}, \mathbf{c}$ be generators of the monoid satisfying $\mathbf{c} \circ \mathbf{b} = \mathbf{c}$ and $\mathbf{b} \circ \mathbf{a} = \mathbf{a}$ and $\mathbf{a} \circ \mathbf{c} = \mathbf{c} \circ \mathbf{a}$ which gives the equation

$$\mathbf{a^i b^j c^k} \circ \mathbf{a^{i'} b^{j'} c^{k'}} = \begin{cases} a^i b^{j+j'} c^{k'} & \text{if } \mathbf{k} = \mathbf{0} \wedge \mathbf{i'} = \mathbf{0}; \\ a^{i+i'} b^{j'} c^{k'} & \text{if } \mathbf{k} = \mathbf{0} \wedge \mathbf{i'} > \mathbf{0}; \\ a^i b^j c^{k+k'} & \text{if } \mathbf{k} > \mathbf{0} \wedge \mathbf{i'} = \mathbf{0}; \\ a^{i+i'} c^{k+k'} & \text{if } \mathbf{k} > \mathbf{0} \wedge \mathbf{i'} > \mathbf{0}; \end{cases}$$

where $\mathbf{i}, \mathbf{j}, \mathbf{k}, \mathbf{i'}, \mathbf{j'}, \mathbf{k'} \in \mathbb{N}$. This monoid is fully automatic: One can represent the group as a convolution of three copies of $(\mathbb{N}, +)$ and use above formula for $\circ$.

However, the monoid is not automatic. Intuitively, the reason is that when multiplying with $\mathbf{c}$ from the front or with $\mathbf{a}$ from the back, the corresponding deletions of the entries for $\mathbf{b}$ cannot be done.

Exercise 8.25

Let $\mathbf{a}, \mathbf{b}$ be generators of a group satisfying

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

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

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

Additonal Exercises

8.26: Assume that $(G, \circ)$ is an infinite automatic semigroup. Show that it cannot be that $\circ$ is, in the given automatic representation, a fully automatic semigroup operation.

8.27: Construct an automatic representation of the group H of all rationals (positive or negative) with multiplication which have only one-digit prime factors – so they are of the form $2^h \cdot 3^i \cdot 5^j \cdot 7^k$ or $-2^h \cdot 3^i \cdot 5^j \cdot 7^k$ with $h, i, j, k \in \mathbb{Z}$.

8.28: Construct a fully automatic representation for the group H from Exercise 8.27. Note that one cannot use the same representation by Exercise 8.26.

8.29: Let a representation $(G, \circ)$ of a group be given and let f map each $x \in G$ to its inverse. Are the following true:
(a) If $(G, \circ)$ is automatic then f is automatic;
(b) If $(G, \circ)$ is fully automatic then f is automatic?

Group Size

Let $(G, \circ)$ be an automatic group. Furthermore, let $f_G(n)$ be the number of elements which have representatives in G of length shorter than n. Construct automatic groups for the following f.

8.30: Prove that every automatic representation of $(\mathbb{Z}, +)$ satisfies that the function f_G is bounded by the expression cn for some constant c.

8.31: Provide an example of an automatic group with quadratic function f_G.

8.32: Determine the function f_G of the semigroup from Example 8.17.

8.33: For fully automatic semigroups, the above questions would be quite easy by showing the following fact: For every nonempty regular set G there is an Abelian semigroup operation $\circ$ such that $(G, \circ)$ is fully automatic.