Relationship between regular languages, DFA’s, and DFAO’s.

Recall: DFAO is a 6-tuple 
\((Q, \Sigma, \delta, q_0, \Delta, \tau)\)

and output on input \(w\) is \(\tau(\delta(q_0, w))\).

A DFA is similar: it is a 5-tuple 
\(M = (Q, \Sigma, \delta, q_0, F)\)

where \(F \subseteq Q\) is the set of final states (or accepting states).

We can think of a DFA as a DFAO by setting
\[\begin{align*}
\Delta &= \{0, 1\} \\
\tau(q) &= \begin{cases} 1, & \text{if } q \in F; \\
0, & \text{otherwise}. \end{cases}
\end{align*}\]

If \(\delta(q_0, w) \in F\), then we say \(M\) accepts \(w\)
and otherwise \(M\) rejects \(w\).

A language is a (finite or infinite) set of strings; a subset of \(\Sigma^*\).

A language recognized by a DFA \(M\) is defined to be
\[L(M) = \{x \in \Sigma^* : \delta(q_0, x) \in F\}\]

A language \(L\) is regular if \(L = L(M)\) for some DFA \(M\).
Example: Thue-Morse automaton as a DFA

\[ L = \{ x \in \{0,1\}^* : |x|_1 \text{ is odd} \} \]

As a DFAO it would be

Another characterization of regular languages is in terms of the operations of union, concatenation and Kleene *. We define

\[ L_1 L_2 = L_1 \cdot L_2 = \{ xy : x \in L_1, y \in L_2 \} \]

\[ L^n = \underbrace{L \cdot L \cdot \ldots \cdot L}_{n \text{ times}} \]

(alternatively, \( L^0 = \{ e \} \) and \( L^n = L \cdot L^{n-1} \))

\[ L^* = \bigcup_{i \geq 0} L_i \]

\[ = \{ x_1 x_2 \ldots x_i : i \geq 0 \text{ and each } x_i \in L \} \]
Thm. A language is regular iff it can be expressed as a finite combination of the operations of union, concatenation, and Kleene * starting from $\epsilon$ and elements of $\Sigma$.

Proof sketch: Make automata for $\epsilon$ and each element of $\Sigma$; join them using "$\epsilon$-transitions" (which allow transition from state to state using no symbols from input); remove $\epsilon$-transitions (which may result in a nondeterministic automaton (NFA)); convert NFA to DFA using the "subset construction". See Section 4.1 of course text.

Notation for regular languages: regular expressions.

- $\emptyset$ denotes empty set
- $\epsilon$ denotes \{ $\epsilon$ \}
- $a$ denotes \{ $a$ \}
- $(r_1)(r_2)$ denotes $L(r_1) \cdot L(r_2)$
- $(r)^*$ denotes $L(r)^*$
- $r_1 \cup r_2$ denotes $L(r_1) \cup L(r_2)$

Superfluous parentheses may be omitted. Precedence:
Star has highest precedence (is done first)
then concatenation, then union.
Example: a regular expression for DFA at top of page 2 is

\[ 0^* 1^* (1^* 0^*)^*. \]

Alternative characterization of automatic sequences

Definition of fibers. Let \( a = (a_n)_{n \geq 0} \) be a sequence.

Define \( \text{Id}(a) = \{ (n)_k : a_n = d \} \) with range \( \Delta \).

Theorem. \( (a_n)_{n \geq 0} \) is \( \kappa \)-automatic if and only if each of the languages \( \text{Id}(a), d \in \Delta \), is a regular language.

Proof. \( \Rightarrow \): Let \( (Q, \Sigma, \Delta, \delta, q_0, \tau) \) be a DFA computing \( a = (a_n)_{n \geq 0} \).

For each \( d \in \Delta \), define

\[ M_d = (Q, \Sigma, \delta, q_0, F_d) \]

where \( F_d = \{ q \in Q : \tau(q) = d \} \).

Then \( F_d \) recognizes the language \( 0^* \text{Id}(a) \), so this is regular. Then by next lemma, \( \text{Id}(a) \) is also regular.
Suppose each \( \text{Id}(a) \) is regular. Then \( O^* \text{Id}(a) \) is regular. So \( \exists \) a DFA \( M_d \) accepting \( O^* \text{Id}(a) \). Now combine each of these DFA’s into a single DFA \( M \) using the "direct product" construction, as follows:

\[
M_d = (Q_d, \Sigma, \delta_d, q_{0d}, F_d)
\]

\[
M = (Q, \Sigma, \delta, q_0, \Delta, \tau)
\]

\[
Q = Q_{d_1} \times Q_{d_2} \times \ldots \times Q_{d_i}
\]

\[
\Delta = \{ \delta_{1d_1,]}(p_1, a), \delta_{2d_2,]}(p_2, a), \ldots, \delta_{id_i,]}(p_i, a) \}
\]

\[
q_0 = [q_{01}, q_{02}, \ldots, q_{0i}]
\]

Define \( c([p_1, \ldots, p_i]) = d_j \) if \( p_j \) is the (unique) state in \( \{ p_1, \ldots, p_i \} \) s.t. \( p_j \in F_d \).
We need a lemma saying we can remove trailing or leading zeroes from the words in a regular language, and have it still be regular. A tool for this is the quotient of languages.

\[ L_1 / L_2 : = \{ x : \exists y \in L_2 \text{ with } xy \in L_1 \} \]

**Thm.** If \( L_1 \) is regular and \( L_2 \) is any language, then \( L_1 / L_2 \) is regular.

**Pf.** Let \( M = (Q, \Sigma, \delta, q_0, F) \) be a DFA for \( L_1 \). Define

\[ M' = (Q, \Sigma, \delta, q_0, F') \]

where

\[ F' = \{ q \in Q : \exists y \in L_2 \text{ s.t. } \delta(q, y) \in F \} \]

Define \( r_{\text{tz}}(x0^i) = x \), for \( x \in \{ 1, 2, \ldots, n-1 \}^* \), \( i \geq 0 \)

\[ r_{\text{lz}}(0^i x) = x \]

\( r_{\text{tz}} \) = remove trailing zeroes
\( r_{\text{lz}} \) = remove leading zeroes

Then \( r_{\text{tz}}(L) = (L / 0^*) \cap (\Sigma_1 \cup \Sigma_2 \cup \ldots \cup \Sigma_{n-1} \cup \{ \emptyset \}) \)

\[ r_{\text{lz}}(L) = (r_{\text{tz}}(L^R))^R \]

So both are regular.
Morphisms. Let $\Sigma, \Delta$ be alphabets.

A morphism $h$ is a map from $\Sigma^*$ to $\Delta^*$ that obeys the identity $h(xy) = h(x)h(y)$ for $x, y \in \Sigma^*$. Note that $h(e)h(e) = h(e)$ and so $h(e) = e$.

Also, the identity implies that it suffices to define a morphism on each element of $\Sigma$.

Example: The Thue-Morse morphism $m(0) = 01, m(1) = 10$. A morphism is $k$-uniform if $|h(a)| = k$ for all $a \in \Sigma$.

A 1-uniform morphism is called a coding.

If $\Sigma \supset \Delta$, we can iterate $h$. $h^0(x) = x$

$h^1(x) = h(h(x))$

$h^2(x) = h(h(h(x)))$ etc.

We say $h$ is "prolongable" on $a$.

Theorem. Suppose $h(a) = a \cdot x$ for some $x \in \Sigma^*$. and $h^i(x) \neq e$ for all $i \geq 0$. Then $h^n(a) = a \cdot h(x) \cdot h^2(x) \cdots = \lim_{n \to \infty} h^n(a)$

is a fixed point of $h$.

Proof. By induction. We claim $h^n(a) = a \cdot h(x) \cdot h^{n-1}(x)$.

This is true for $n=0$. Assume true for $n$. Then
\[ h^{n+1}(a) = h(h^n(a)) \]
\[ = h(a \times h(x) \bullet h^{-1}(x)) \]
\[ = h(a) h(x) h^2(x) \bullet h^n(x) \]
\[ = a \times h(x) h^2(x) \bullet h^n(x). \]

Furthermore, note that
\[ h(h^w(a)) = h(a \times h(x) h^2(x) \bullet \cdots) \]
\[ = a \times h(x) h^2(x) \cdots \]
\[ = h^w(a). \]

So \( h^w(a) \) is a fixed point.

**Cobham's little theorem.** Let \( b = (b_n)_{n \geq 0} \) be a sequence taking values in \( \Delta \), a finite alphabet.

Then \( b \) is \( k \)-automatic iff \exists a finite alphabet \( \Gamma \), a \( k \)-uniform morphism \( h : \Gamma^* \rightarrow \Gamma^* \), \( a \in \Gamma \), and a coding \( \tau : \Gamma^* \rightarrow \Delta^* \) such that
\[ b = \tau(h^w(a)). \]

**Proof.** Suppose \( b \) is \( k \)-automatic. Then \( b \) corresponds to a \( k \)-uniform morphism \( h : \Gamma^* \rightarrow \Gamma^* \). By Cobham's theorem, there exists a DFAO \( M = (Q, \Sigma_k, \Delta, q_0, \delta, T) \) computing \( b \). Take \( \Gamma = Q \). Define
\[ h \] as follows: \( h(q) = \delta(q_0, 0) \delta(q_1, 0) \cdots \delta(q_{k-1}, 0). \)

WLOG we may assume \( \delta(q_0, 0) = q_0 \). Take \( a = q_0 \).
We need a little lemma.

**Lemma** If $h$ is $k$-uniform and $h(w) = w$, then $h(w[n]) = W[kn] \ldots W[kn+k-1]$.

**Pf.** By length considerations we have

$h(w[0..n-1]) = W[0] \ldots W[kn-1]$

$h(w[0..n]) = W[0] \ldots W[kn+k-1]$

and the result follows.
Let \( W = h^*(a) \). We show that
\[
\delta(q_0, y) = W[ [y]_k ]
\]
(*)
for all \( y \in \Sigma^* \), by induction on \( |y| \).

The base case is \( |y| = 0 \). Then the LHS of (*) is
\[
\delta(q_0, \epsilon) = q_0^* = a \quad \text{and the RHS is } W[0] = a.
\]
Now assume (*) is true for all \( y \) with \( |y| < i \); we prove it for \( |y| = i \). Write \( y = xa \), \( a \in \Sigma_k \).

Then
\[
\delta(q_0, y) = \delta(\delta(q_0, x), a)
\]

\[
= \delta(W[[x]_k], a) \quad \text{by induction}
\]

\[
= h(W[[x]_k])[a] \quad \text{by def. of } h
\]

\[
= (W[[x]_k] \ldots k[[x]_k + k-1]) [a] \quad \text{by lemma}
\]

\[
= W[[x]_k + a]
\]

\[
= W[[x]_k]
\]

\[
= W[[y]_k].
\]

Let \( y = (n)_k \). Then
\[
\tau(W[n]) = \tau(W[[n]_k])
\]

\[
= \delta(q_0, (n)_k) = b_n
\]

So \( \tau(W) = k \).
Suppose \( b = \tau(w) \) where \( w \in \Gamma^w \)

\[ w = h^w(a) \]

for some \( \kappa \)-uniform morphism \( h : \Gamma^w \rightarrow \Gamma^w \)

let \( \alpha \). Define the DFAO

\[ M = (\Gamma, \Sigma^w, \Delta, q_0, \delta, \tau) \]

where \( q_0 = a \), and

\[ \delta(q, c) = h(q)[c] \]

for all \( q \in \Gamma, c \in \Sigma^w \).

We claim: \( \omega[n] = \delta(q_0, (n)_\kappa) \) \( \forall n \geq 0 \).

proof by induction on \( n \). Easy for \( n = 0 \). Assume true for \( n' < n \); we prove for \( n \). Write \( (n)_\kappa = x a \), where \( x \in \Sigma^w \), \( a \in \Sigma^w \). Then

\[ n = \kappa \left[ x \right] \kappa + a \]

and \( \left[ x \right] \kappa < n \). Then

\[ \delta(q_0, (n)_\kappa) = \delta(q_0, x a) \]

\[ = \delta(\delta(q_0, x), a) \]

\[ = \delta(\delta(q_0, (\frac{n-a}{\kappa})_\kappa), a) \]

\[ = \delta(\omega n, a) \]

\[ = h(\omega n)[a] \]

\[ = \omega n \].

so \( \tau(\delta(q_0, (n)_\kappa)) = \tau(\omega n) = b \). \( [n] = b_n \).
Consequence: we can trivially go from a k-DFAO computing $a$ to the corresponding representation as image under a coding, or fixed point of $k$-uniform morphism.

Example 1: from Rudin–Shapiro automaton (p. 6, lecture)

to morphism representation:

$h(q_0) = q_0 \ 9_1$  $\quad \ T(q_0) = 0$
$h(q_1) = q_0 \ 9_2$  $\quad \ T(q_1) = 0$
$h(q_2) = q_3 \ 9_1$  $\quad \ T(q_2) = 1$
$h(q_3) = q_3 \ 9_2$  $\quad \ T(q_3) = 1$

Example 2: Consider the morphism $\gamma$ defined by $\gamma(0) = 001$, $\gamma(1) = 110$.

Iterating $\gamma$ gives $001001110001001110\ldots$

The corresponding DFAO is then

\begin{tikzpicture}
  \node (q0) [state] {$q_0/0$};
  \node (q1) [state, below right of=q0] {$q_1/1$};

  \path
    (q0) edge[->, bend right] node {$0,1$} (q1)
    (q1) edge[->, bend right] node {$0,1$} (q0)
    (q0) edge[->, out=270, in=180] node {$2$} (q1)
    (q1) edge[->, out=90, in=0] node {$2$} (q0);
\end{tikzpicture}