Last time: formal grammars
This time: finite automata

Definition: A finite state automaton (fa), $M = (Q, S, f, s_0, F)$, where $Q$ is a set of states, $S$ is alphabet, $f: Q \times S \rightarrow Q$ (transition function), $s_0$: initial state, $F \subseteq Q$: terminal state

Example:

$S = \{0, 1\}$,
$Q = \{1, 2, 3\}$, (each number is inside a circle)
$F = \{3\}$, (3 is inside a circle)
$s_0 = 1$ (1 is inside a circle)

$f = \begin{cases} 
\text{circle 1, 0 to circle 1} & 
\text{circle 1, 1 to circle 1, 1 to circle 3} \\
\text{circle 2, 0 to circle 2} & 
\text{circle 2, 1 to circle 2} \\
\text{circle 3, 0 to circle 2} & 
\text{circle 3, 1 to circle 1} 
\end{cases}$

Input: 100
Result:
We can see that this input is not accepted because state 2 doesn’t belong to F.

Let’s expand \( f \) to handle strings.

Now, we set \( f' = Q \# S^* \sim Q \).

\[ f(s, m) = s, \quad f(s, ay) = f(f(s, a), y), \]

Definition: Let \( x \in S^* \). \( x \) is accepted by \( M \) if \( f(s_0, x) \in F \).

\( T(M) = \{ x \mid x \in S^*, f(s_0, x) \in F \} \).

\( T(M) \) is a set of all sequences accepted by \( M \).

Example:

"a, b, *:"

\[ 1 \]  
\[ 2 \]  
\[ 1 \]  
\[ 2 \]  
\[ 2 \]  

"a, :"
From the above transition graph, we can get some string like:
\[ aa, abba, a(bb)^n a, (aa)^m ab(aa)^n (bb) \]
Actually, the above transition graph demonstrates a language as the following.
\[ \text{EVENAB} = \{ x \mid x \in \{a, b\}^*, \sum_a x \text{ is even, } \sum_b x \text{ is even.} \} \]

Claim: A language accepted by a FA is called a regular language.
Claim: All finite languages are regular.

Let \( M_1 = (Q_1, S, f_1, s_1, F_1) \) and \( M_2 = (Q_2, S, f_2, s_2, F_2) \).
\( M_1 \# M_2 \): direct product machine.
\( M_1 \# M_2 = (Q_1 \# Q_2, S, f, (s_1, s_2), F) \) where \( F = F_1 \# F_2, 6 (p, q) \in Q_1 \# Q_2, a \in S, \)
\[ f((p, q), a) = (f_1(p, a), f_2(q, a)). \]

\( M_1, M_2 \): direct union machine
\( M_1, M_2 = (Q_1 \# Q_2, S, f, (s_1, s_2), (F_1 \# Q_2, F_2 \# Q_1)) \)
\[ T(M_1 \# M_2) = T(M_1) + T(M_2) \]
\[ T(M_1, M_2) = T(M_1), T(M_2) \]

The class of Regular languages is closed under intersection, union, and negation.

Negation:
Without proof: If \( A \& B \) are regular, so are \( AB \) and \( A^* \).

Definition: A non-deterministic finite automaton (NFA) is \( M = (Q, S, f, s_0, F) \), \( f: Q \times S^* \rightarrow P(Q) \).
And DFA is a special case of NFA.
Example of nfa:

\[ f(q, m) = \begin{cases} q, \\ f(q, ay) = \begin{cases} q', & f(q, a) \\ f(q', y) & \end{cases} \end{cases} \]

A string is accepted if \( f(s0, x) + F \neq z \).

\[ f: \]

<table>
<thead>
<tr>
<th>A</th>
<th>0</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>C</td>
<td>A</td>
<td>C</td>
</tr>
</tbody>
</table>

Final state C

Start in A. And the string is 1011.