Definition: An **alphabet** is a non-empty finite set, e.g., $\Sigma = \{0, 1\}$, $\Gamma = \{a, b, c\}$, etc.

Definition: A **string** over an alphabet Σ is a finite sequence of zero or more symbols from Σ. The unique string with zero symbols is called ϵ. The set of all strings over Σ is called Σ^*.

Definition: A **language** over Σ is any subset of Σ^*. The decision problem for a language L is to input a string w and determine whether $w \in L$.

In formal language theory we look at various kinds of *machines* that take a string as input, look at one letter at a time, and decide whether the string is in some language.

We also look at various formal ways to *specify* a language, such as regular expressions and context-free grammars, that are used in the real world.

In the 1950's and 1960's it was discovered that each of the most natural machine models corresponded to a specification system: the languages that could be decided by the machines were exactly those that could be specified in a certain way. Here we'll see some examples of that phenomenon.

Finally, we will always be interested in when a language *cannot be decided* by any machine in some class, or *cannot be specified* within some system. Such a result is called a *lower bound*, because we show that some particular amount of resources is insufficient.

Definition: The set of **regular expressions** $\mathbf{R}(\Sigma)$ over alphabet Σ is the set of strings made up from Σ, "ϵ", and "$\emptyset$" and using the operations $\cup$, $\circ$, and *.

Meaning of a Regular Expression:

1. if $a \in \Sigma$ then $a \in \mathbf{R}(\Sigma)$; $\mathcal{L}(a) = \{a\}$
2. $\epsilon \in \mathbf{R}(\Sigma)$; $\mathcal{L}(\epsilon) = \{\epsilon\}$
3. $\emptyset \in \mathbf{R}(\Sigma)$; $\mathcal{L}(\emptyset) = \emptyset$
4. if $e, f \in \mathbf{R}(\Sigma)$ then so are $(e \cup f)$, $(e \circ f)$, $(e^\star)$:

$$\begin{aligned} \mathcal{L}(e \cup f) &= \mathcal{L}(e) \cup \mathcal{L}(f) \\ \mathcal{L}(e \circ f) &= \mathcal{L}(e)\mathcal{L}(f) = \{uv \mid u \in \mathcal{L}(e), v \in \mathcal{L}(f)\} \\ \mathcal{L}(e^\star) &= (\mathcal{L}(e))^\star \end{aligned}$$

Definition: A **deterministic finite automaton (DFA)** is a tuple,

$$D = (Q, \Sigma, \delta, s, F)$$

- Q is a finite set of states,
- Σ is a finite alphabet,
- $\delta : Q \times \Sigma \to Q$ is the transition function,
- $s \in Q$ is the start state, and
- $F \subseteq Q$ is the set of final or accept states.

A DFA executes the following pseudo-Java algorithm:

```
public boolean isAccepted (String w) {
   State s = startState;
   for (int i=0; i < w.length(); i++)
      s = delta(s, w.charAt(i));
   return isFinalState(s);}
```

We will be interested in the following results about DFA's and regular languages.

- **Kleene's Theorem:** A language is decided by some DFA iff it is regular.
- **Myhill-Nerode Theorem:** There is a *minimal* DFA for any regular language, definable in terms of a purely language-theoretic property.
- **Non-Regularity Proofs:** If a language is not regular, we can usually prove that fact.

Definition: A **nondeterministic finite automaton (NFA)** is a tuple,

$$N = (Q, \Sigma, \Delta, s, F)$$

- Q is a finite set of states,
- Σ is a finite alphabet,
- $\Delta : (Q \times (\Sigma \cup \{\epsilon\}) \to \wp(Q)$ is the transition function,
- $s \in Q$ is the start state, and
- $F \subseteq Q$ is the set of final or accept states.

$$\mathcal{L}(N) \quad = \quad \{w \mid s \xrightarrow[w]{\star} q \in F\}$$

So $\Delta(q, a)$ is the *set* of states to which N *might* go if it reads a when in state q. There might be zero, one, or more than one.

Proposition 3.1 *Every NFA N can be translated into an NFA without ϵ-transitions N' such that $\mathcal{L}(N) = \mathcal{L}(N')$.*

Proposition 3.2 *For every NFA, N, with n states, there is a DFA, D, with at most 2^n states s.t. $\mathcal{L}(D) = \mathcal{L}(N)$.*

Theorem 3.3 (Kleene's Theorem) *Let $A \subseteq \Sigma^\star$ be any language. Then the following are equivalent:*

1. *$A = \mathcal{L}(D)$, for some DFA D.*
2. *$A = \mathcal{L}(N)$, for some NFA N with no ϵ-transitions*
3. *$A = \mathcal{L}(N)$, for some NFA N.*
4. *$A = \mathcal{L}(e)$, for some regular expression e.*
5. *A is regular.*

Proof: Obvious that $1 \to 2 \to 3$.

$3 \to 2$ by Prop. 1.2 (ϵ-elimination).

$2 \to 1$ by Prop. 1.3 (subset construction).

$4 \leftrightarrow 5$ by definition

$4 \to 3$: We showed by induction on all regular expressions e that there is an NFA N with $\mathcal{L}(e) = \mathcal{L}(N)$.

$3 \to 4$: (Cf. *state elimination* proof [S, Lemma 1.32])

Let $N = (\{1, \ldots, n\}, \Sigma, \Delta, 1, F)$, $F = \{f_1, \ldots, f_r\}$

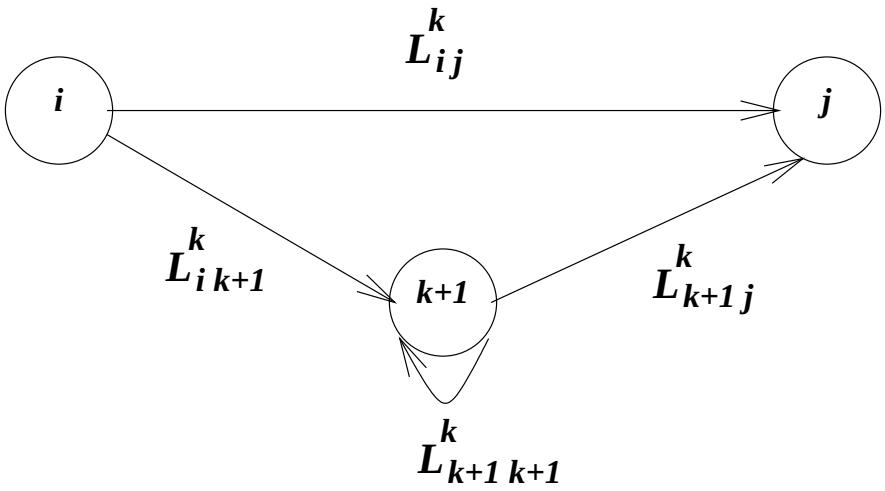
$$L_{ij}^k \equiv \{w \mid j \in \Delta^\star(i, w);\ \text{no intermediate state} \# > k\}$$

$$L_{ij}^0 = \{a \mid j \in \Delta(i, a)\} \cup \{\epsilon \mid i = j\}$$

$$L_{ij}^{k+1} = L_{ij}^k \cup L_{i\,k+1}^k (L_{k+1\,k+1}^k)^\star L_{k+1,j}^k$$

$$e = L_{1\,f_1}^n \cup \cdots \cup L_{1\,f_r}^n$$

$$\mathcal{L}(e) = \mathcal{L}(N)$$



♠

Let $A \subseteq \Sigma^\star$ be any language.

Define the **right-equivalence relation** $\sim_A$ on $\Sigma^\star$:

$$x \sim_A y \quad \Leftrightarrow \quad (\forall w \in \Sigma^\star)(xw \in A \leftrightarrow yw \in A)$$

$x \sim_A y$ iff x and y cannot be distinguished by concatenating some string w to the right of each of them and testing for membership in A.

Example: $A_1 = \{w \in \{a,b\}^\star \mid \#_b(w) \equiv 0 \pmod 2\}$

$$\epsilon \sim_{A_1} a \sim_{A_1} aa \qquad b \sim ab \sim bbb$$

Claim: $x \sim_{A_1} y$ iff $\#_b(x) \equiv \#_b(y) \pmod 2$.

Proof: Suppose $x \sim_{A_1} y$. Let $w = \epsilon$.

$$xw = x \in A_1 \qquad \leftrightarrow \qquad yw = y \in A_1$$

Thus, $\#_b(x) \equiv \#_b(y) \pmod 2$.

Suppose, $\#_b(x) \equiv \#_b(y) \pmod 2$.

$$(\forall w)\#_b(xw) \equiv \#_b(yw) \pmod 2 \ .$$

$$(\forall w)(xw \in A_1 \quad \leftrightarrow \quad yw \in A_1)$$

Thus, $x \sim_{A_1} y$. ♠

$$\begin{aligned} [u]_{\sim_A} &= \{w \in \Sigma^\star \mid u \sim_A w\} \\ [a] &= \{w \in \{a,b\}^\star \mid \#_b(w) \equiv 0 \pmod 2\} \\ [b] &= \{w \in \{a,b\}^\star \mid \#_b(w) \equiv 1 \pmod 2\} \end{aligned}$$

Review: Show that for any language A, $\sim_A$ is an equivalence relation. Recall that an equivalence relation is a binary relation that is reflexive, symmetric, and transitive.

Proof: Reflexive: $(\forall x \in \Sigma^\star)(x \sim_A x)$

Let $x, w \in \Sigma^\star$ be arbitrary.

$$(xw \in A \leftrightarrow xw \in A)$$

$(\forall w \in \Sigma^\star)(xw \in A \leftrightarrow xw \in A)$ because w was arbitrary.

$$x \sim_A x$$

$(\forall x \in \Sigma^\star)(x \sim_A x)$ because x was arbitrary.

Symmetric: $(\forall x, y \in \Sigma^\star)(x \sim_A y \to y \sim_A x)$

Let $x, y, \in \Sigma^\star$ be arbitrary.

Suppose $x \sim_A y$.

$$(\forall w)(xw \in A \ \leftrightarrow \ yw \in A)$$

$$(\forall w)(yw \in A \ \leftrightarrow \ xw \in A)$$

$$y \sim_A x$$

$$x \sim_A y \to y \sim_A x$$

$$(\forall x, y \in \Sigma^\star)(x \sim_A y \to y \sim_A x)$$

Transitive:

$$(\forall x, y, z \in \Sigma^\star)((x \sim_A y \wedge y \sim_A z) \rightarrow x \sim_A z)$$

Let $x, y, z \in \Sigma^\star$ be arbitrary.

Suppose $x \sim_A y \wedge y \sim_A z$.

$$(\forall w)(xw \in A \leftrightarrow yw \in A)$$

$$(\forall w)(yw \in A \leftrightarrow zw \in A)$$

Let $w \in \Sigma^\star$ be arbitrary.

$$(xw \in A \leftrightarrow yw \in A)$$

$$(yw \in A \leftrightarrow zw \in A)$$

$$(xw \in A \leftrightarrow zw \in A)$$

$(\forall w \in \Sigma^\star)(xw \in A \leftrightarrow zw \in A)$ because w was arbitrary.

$$x \sim_A z$$

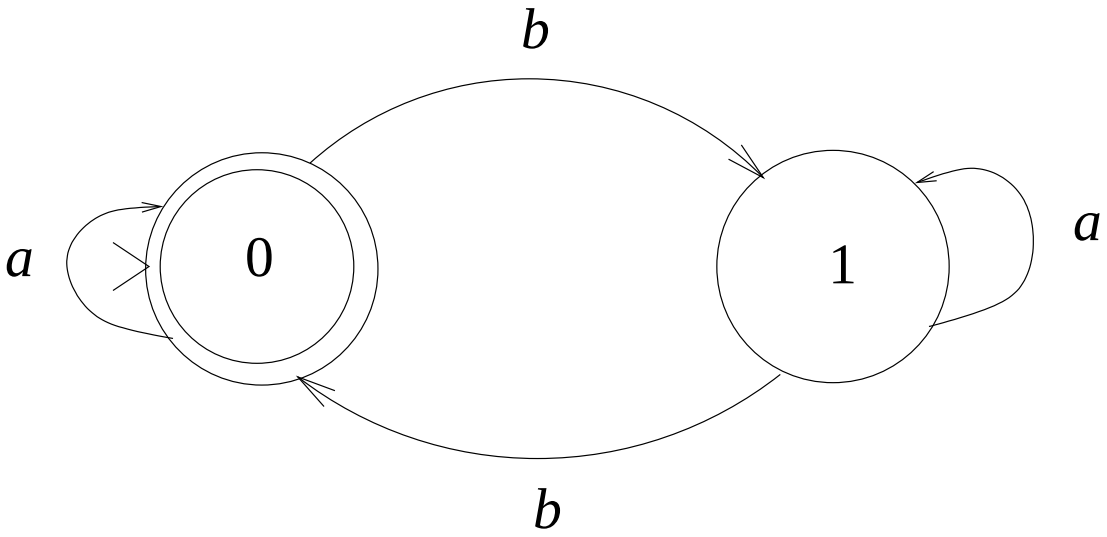
$$(x \sim_A y \wedge y \sim_A z) \rightarrow x \sim_A z$$

$(\forall x, y, z \in \Sigma^\star)(x \sim_A y \wedge y \sim_A z) \rightarrow x \sim_A z$ because x, y, z were arbitrary. ♠

- To prove $(\forall x)\varphi$: let x be arbitrary, prove φ, conclude $(\forall x)\varphi$.
- To prove $\varphi \rightarrow \psi$: assume φ, prove ψ, conclude $\varphi \rightarrow \psi$.
- From $\varphi \wedge \psi$ may conclude φ, ψ.
- From φ, ψ may conclude $\varphi \wedge \psi$.
- To prove φ: assume $\neg\varphi$, prove $A \wedge \neg A$, conclude φ.

An Example:

$$x \sim_{A_1} y \quad \Leftrightarrow \quad \#_b(x) \equiv \#_b(y) \pmod 2$$



This language has two equivalence classes, and the DFA operates by reading each letter in turn and determining the class of the part of the string seen so far. The correspondence between states and classes is the essence of the theorem.

Myhill-Nerode Theorem: The language A is regular iff $\sim_A$ has a finite number k of equivalence classes. Furthermore, this number k is equal to the number of states in the minimum-state DFA that decides A.

Proof: Suppose $A = \mathcal{L}(D)$ for some DFA,

$$D = (\{q_1, q_2, \ldots, q_n\}, \Sigma, \delta, q_1, F)$$

Let $S_i = \{w \mid \delta^\star(q_1, w) = q_i\}$

Claim: Each S_i is contained in a single $\sim_A$ equivalence class.

Let $x, y \in S_i$, $w \in \Sigma^\star$ be arbitrary.

$$\delta^\star(q_1, xw) = \delta^\star(\delta^\star(q_1, x), w) = \delta^\star(\delta^\star(q_1, y), w) = \delta^\star(q_1, yw)$$

$$\mathcal{L}(D) = \{z \mid \delta^\star(q_1, z) \in F\}$$

$$xw \in A \leftrightarrow \delta^\star(q_1, xw) \in F \leftrightarrow \delta^\star(q_1, yw) \in F \leftrightarrow yw \in A$$

$$(\forall w)(xw \in A \leftrightarrow yw \in A)$$

$$x \sim_A y$$

Thus, there are *at most* n equivalence classes.

Conversely, suppose that there are finitely many equivalence classes of $\sim_A$: $E_1, \ldots, E_m$.

Let $[x]$ be the equivalence class that x is in.

Define $D = (\{E_1, \ldots, E_m\}, \Sigma, \delta, [\epsilon], F)$ where

$$F \;=\; \{[x] \;\mid\; x \in A\}$$

$$\delta([x], a) \;=\; [xa]$$

We must show that δ is well defined, i.e.,

$$([x] = [y]) \quad \Rightarrow \quad ([xa] = [ya])$$

Suppose $x \sim_A y$.

$$(\forall w)(xw \in A \;\leftrightarrow\; yw \in A)$$

$$(\forall w)(xaw \in A \;\leftrightarrow\; yaw \in A)$$

Thus, $xa \sim_A ya$.

Claim: $\delta^\star([\epsilon], x) = [x]$.

Proof: by induction on $|x|$ [exercise].

$$x \in \mathcal{L}(D) \;\leftrightarrow\; \delta^\star([\epsilon], x) \in F \;\leftrightarrow\; [x] \in F \;\leftrightarrow\; x \in A$$

Example: The following language is regular, and its minimal DFA has seven states:

$$A_7 = \{w \in \{0,1,\ldots,9\}^\star \mid 7|w\}$$

$$D_7 = (\{0,1,\ldots,6\}, \Sigma, \delta_7, 0, \{0\})$$

$$\delta_7(q,d) = (10q+d) \bmod 7 \quad = \quad (3q+d) \bmod 7$$

It is straightforward to show that $\mathcal{L}(D_7) = A_7$ (try this as an exercise). To show that D_7 is minimal, we must show that there are seven different equivalence classes. Since the strings 0, 1, $\ldots$, 6 each take D_7 to a different state, they are our candidates to be in different classes. We must show:

$$(\forall i \neq j \in \{0,1,\ldots,6\})(i \not\sim_{A_7} j)$$

Let $i \neq j \in \{0, 1, \ldots, 6\}$ be arbitrary. We can find a number d such that $3i + d \equiv 0 \pmod 7$ (why?). If $i \sim_{A_7} j$ were true, then we would have $3j + d \equiv 0 \pmod 7$ because $\delta^*(id)$ would have to equal $\delta^*(jd)$. But in that case:

$$\begin{aligned} 3i + d &\equiv 3j + d \pmod 7 \\ 3i &\equiv 3j \pmod 7 \\ 15i &\equiv 15j \pmod 7 \\ i &\equiv j \pmod 7 \end{aligned}$$

Thus, we must have $i \not\sim_{A_7} j$. Since i and j were arbitrary, no two of the seven strings are equivalent and there are seven classes.

By Kleene, a language is non-regular if it has no DFA.

By Myhill-Nerode, then, A is non-regular if $\sim_A$ has infinitely many equivalence classes.

To *prove* that A is non-regular, we find an infinite set of strings, no two of which are $\sim_A$-equivalent.

Example: Show $E = \{a^n b^n \mid n \in \mathbf{N}\}$ is not regular.

Proof: Let $i \neq j \in \mathbf{N}$ be arbitrary.

We can easily show from the definition that $a^i \not\sim_E a^j$.

Let $w = b^i$

$$a^i w \in E; \qquad a^j w \notin E$$

So each element of $\{a^i : i \in \mathbf{N}\}$ is in a separate equivalence class.

Another Example: $PAL = \{w \mid w = w^R\}$ is not regular. (Here w^R is w written backwards.)

It's *almost* true that no two different strings are equivalent, but not quite. (Can you find a counterexample?) All we need for the proof, though, is to find an infinite set of strings. $\{a^i b : i \in \mathbf{N}\}$ is such a set, if there are two distinct letters a and b in Σ.

What if Σ contains only a single letter? (This is called a *unary* language.) Then PAL *is* a regular language (why?). It's easy to miss cases like this if you're not careful. But look at our proof! In order to finish it, we had to *assume* that there were two different letters in Σ. To justify this step, we must add a condition to the *statement* of the problem.

Harder Example: The set $UPRIME$, defined as $\{a^i : i \text{ is prime}\}$, is not regular.

We have seen that a single class of formal languages has several alternate specifications: by Kleene's Theorem a language has a DFA iff it has an NFA iff it has a regular expression. (There are more!)

This suggests that this class has mathematical importance. The alternate characterizations are also *useful* in proofs.

For example, we can ask what operations, when applied to regular languages, *always* give us more regular languages. Cases of this phenomonon are called **closure properties**.

Closure Theorem for Regular Sets: Let $A, B \subseteq \Sigma^\star$ be regular languages and let $h : \Sigma^\star \to \Gamma^\star$ and $g : \Gamma^\star \to \Sigma^\star$ be homomorphisms. Then the following languages are regular:

1. $A \cup B$
2. AB
3. $\overline{A} = (\Sigma^\star - A)$
4. $A \cap B$
5. $h(A)$
6. $g^{-1}(A)$

The last two require a new definition:

A language **homomorphism** is a function $h : \Sigma^\star \to \Gamma^\star$ s.t.

$$(\forall x, y \in \Sigma^\star)(h(xy) = h(x)h(y)) \tag{3.3}$$

Examples:

$$\begin{aligned} h : \{0,1,2,3\}^\star &\to \{a,b\}^\star \\ h(0) = aa,\ h(1) = b, &\quad h(2) = aba,\ h(3) = \epsilon \\ h(012310) &= aababababaa \end{aligned}$$

$$\begin{aligned} g : \{a,b\} &\to \{a,b,c\} \\ g(a) = a, &\quad g(b) = cbc \\ g(baa) &= cbcaa \end{aligned}$$

Notation: for function $f : A \to B$, sets $S \subseteq A, T \subseteq B$,

$$f(S) = \{f(a) \mid a \in S\}; \quad f^{-1}(T) = \{a \in A \mid f(a) \in T\}$$

Example:

$$A_1 = \{w \in \{a,b\}^\star \mid \#_b(w) \equiv 0 \pmod 2\}$$

$$\begin{aligned} h^{-1}(A_1) &= \{w \in \{0,1,2,3\}^\star \mid \#_1(w) + \#_2(w) \equiv 0 \pmod 2\} \\ g(A_1) &= \{w \in \{a,b,c\}^\star \mid \#_{cbc} \equiv 0 \pmod 2;\ \text{no other b or c}\} \end{aligned}$$

Proofs of Closure Properties:

(1,2): Let $\mathcal{L}(e) = A$, $\mathcal{L}(f) = B$.

Thus $\mathcal{L}(e \cup f) = A \cup B$; $\mathcal{L}(e \circ f) = AB$

(3): Let $\mathcal{L}(D) = A$, DFA $D = (Q, \Sigma, \delta, s, F)$.

Let $\overline{D} = (Q, \Sigma, \delta, s, Q - F)$.

Thus $\mathcal{L}(\overline{D}) = \overline{A}$

(4): $A \cap B = \overline{\overline{A} \cup \overline{B}}$

(5): Let $A = \mathcal{L}(e)$.

Thus $h(A) = \mathcal{L}(h(e))$.

Example:

$$\begin{aligned} g(a) &= a, \quad g(b) = cbc \\ A &= \mathcal{L}(a^\star(ba^\star ba^\star)^\star) \\ g(A) &= \mathcal{L}(a^\star(cbca^\star cbca^\star)^\star) \end{aligned}$$

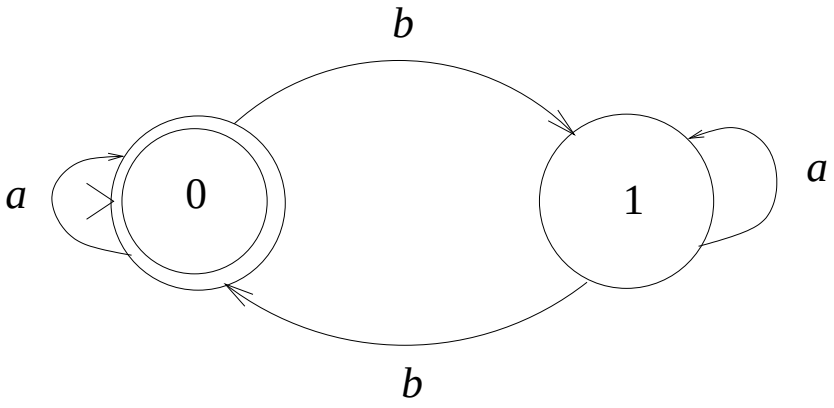
(6): Let $A = \mathcal{L}(D)$, DFA, $D = (Q, \Sigma, \delta, s, F)$.
Let $D' = (Q, \Gamma, \delta', s, F)$.

$$\delta'(q, \gamma) \quad = \quad \delta^\star(q, h(\gamma))$$

Example:

$$h(0) = aa, \quad h(1) = b, \quad h(2) = aba, \quad h(3) = \epsilon$$

D



D’