$(a+\varepsilon)(b+\varepsilon)(ab+bc)^* c.$

$$\begin{array}{cccc} a & b & ab & c. \\ \varepsilon & \varepsilon & bc & \end{array}$$

w in $\{a,b\}^*$ |w|_a is odd.

$\Rightarrow b^*(ab^*ab^*)^*ab^*$

Oct 11

COMPSCI 3331

Fall 2022

What's next?

- Quiz 2 grades not available yet, solution available.
- Assignment 1: due TONIGHT at 11:59 PM
- Assignment 2: available by end of day today, due Oct 26.
- Quiz 3: tomorrow, end of Lecture 6.
- Midterm: October 25.

Closure Properties

- Easy: use regular expressions to show that regular languages are closed under concatenation, Kleene star and union.
- What about intersection, reversal, complement?

Intersection

$M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ DFAS

$M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$.

DFA for the intersection:

$M = (\underbrace{Q_1 \times Q_2}_{\uparrow}, \Sigma, \delta, (q_1 + q_2), F)$

Cartision Product

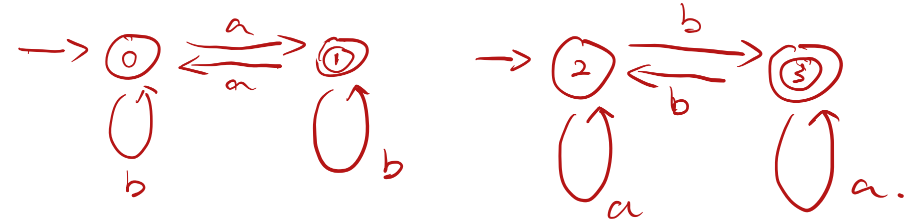
$\delta((q_i, q_j), a)$

$= \delta(\delta_1(q_i, a), \delta_2(q_j, a))$

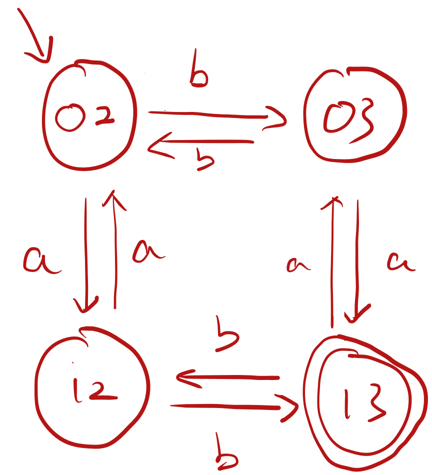
parallel run two states.

$F = F_1 \times F_2$

Example:



Construct a DFA that is accept in both languages



Union: everything stays the same except F.

$F' = F_1 \times Q_2 \cup F_2 \times Q_1$.

Stages: $A \times B$

transition: intersection: $A \cap B$; union: $A \cup B$.

final stages: similar to the rule of finding transitions

Complement

$L \subseteq \Sigma^*$, $\bar{L} = \Sigma^* - L$

Flip the final state.

DFA $M = \{Q, \Sigma, \delta, q_0, F\}$.

Complement $\bar{M} = \{Q, \Sigma, \delta, q_0, Q - F\}$.

n be —

$z =$ — $\in L$

$z = uvw$. $|uv| \leq n$ $v \neq \varepsilon$

$u =$ $v =$ $w =$ $i + j < n$ $j \geq 0$.

consider $k =$ — $z = uv^k w =$ —

— Thus

Showing languages aren't regular

- Intuition: does this language take a lot of memory?

Ex: $L = \{a^n b^n \; n \geq 0\}$.

such language could not be translate to a DFA, it is not a regular language.

Showing languages aren't regular

- Language B of "balanced parentheses". $\Sigma = \{(,)\}$.
- Every open parenthesis is matched with a closing parenthesis, no extra parentheses.
- So $B = \{x \in \Sigma^* \ : \ |x|_(= |x|_) \text{ and } \forall \text{ prefixes } y \text{ of } x, |y|_(\geq |y|_)\}$

!!

)(is invalid.

Showing languages aren't regular

Let n be the constant defined by pumping lemma.

pick the string $z = a^n b^n \in L$

we can decompose $z = uvw$, $|uv| \leq n$, $|v| > 0$.

$u = a^i \quad v = a^j \quad w = a^{n-i-j} b^n$

consider $uv^2w = a^i a^{2j} a^{n-i-j} b^n$

$= a^{n+j} b^n \notin L$

Therefore, L is not regular by the pumping lemma.

Showing languages aren't regular

- $L_1 = \{a^n b^m \ : \ n \neq m\}$.
- $L_2 = \{w\#y \ : \ w, y \in \{a,b\}^*, |w| < |y|\} \ (\subseteq \{a,b,\#\}^*)$
- $L_3 = \{w\#y \ : \ w, y \in \{a,b\}*, |w|_a = |y|_b\}$.
- $L_4 = \{w\#y \ : \ w, y \in \{0,1\}^*, w < y$ as binary numbers. $\}$.

Show L_1 is not regular:

let n be the constant defined

The pumping lemma.

$z = \underline{a^n b^{n+1}} \in L_1$

in the language.

$u = a^i \quad v = a^j \quad w = \varepsilon$.

$w = a^{n-i-j} b^{n+1} \quad \begin{cases} j > 0 \\ i+j \leq n \end{cases}$

try to show $uv^k w$ is not in the language

$k = i+1$

$uv^k w = a^i a^{(i+1)j} a^{n-i-j} b^{n+1}$

$\underline{= a^{n+ij} b^{n+1}.}$ can guarantee. $\Rightarrow$ X

$\Downarrow$

pick a different string.

$z = a^n b^{n+n!} \in L_1$

$w = a^{n-i-j} + b^{n+n!}$

$uv^k w = a^i + a^{jk} + a^{n-i-j} b^{n+n!}$

$= a^{n+(k-1)j} b^{n+n!}$

$k = \frac{n!}{j} + 1$

L_2: $z = w^i \# y^j \in L_1$ $(|w| < |y|)$.

uvw

$u = w^x$

$v = w^y$

$w = w^{i-x-y} \# y^j$

$uv^k w = w^{i+(k-1)y} \# y^j$

$i + (k-1)y = j$

$k = \frac{j-i}{y} + 1$.

$i + \left(\frac{j-1}{y} \cancel{+1-1}\right) y$

$= i + j - 1$.

$z = w^{i+j-1} \# y^j \notin L$

Showing languages aren't regular

- $L_5 = \{w \in \{a,b,c,d\}^* : \forall x \in \{a,b,c,d\}, |w|_x = 0 \text{ or } |w|_x \geq 5\}$
- $L_6 = L^*$ where $L = \{a^n b^n : n \geq 0\}$
- $L_7 = \{w \in \{a,b\}^* : |w|_a \equiv |w|_b (\bmod 10)\}$