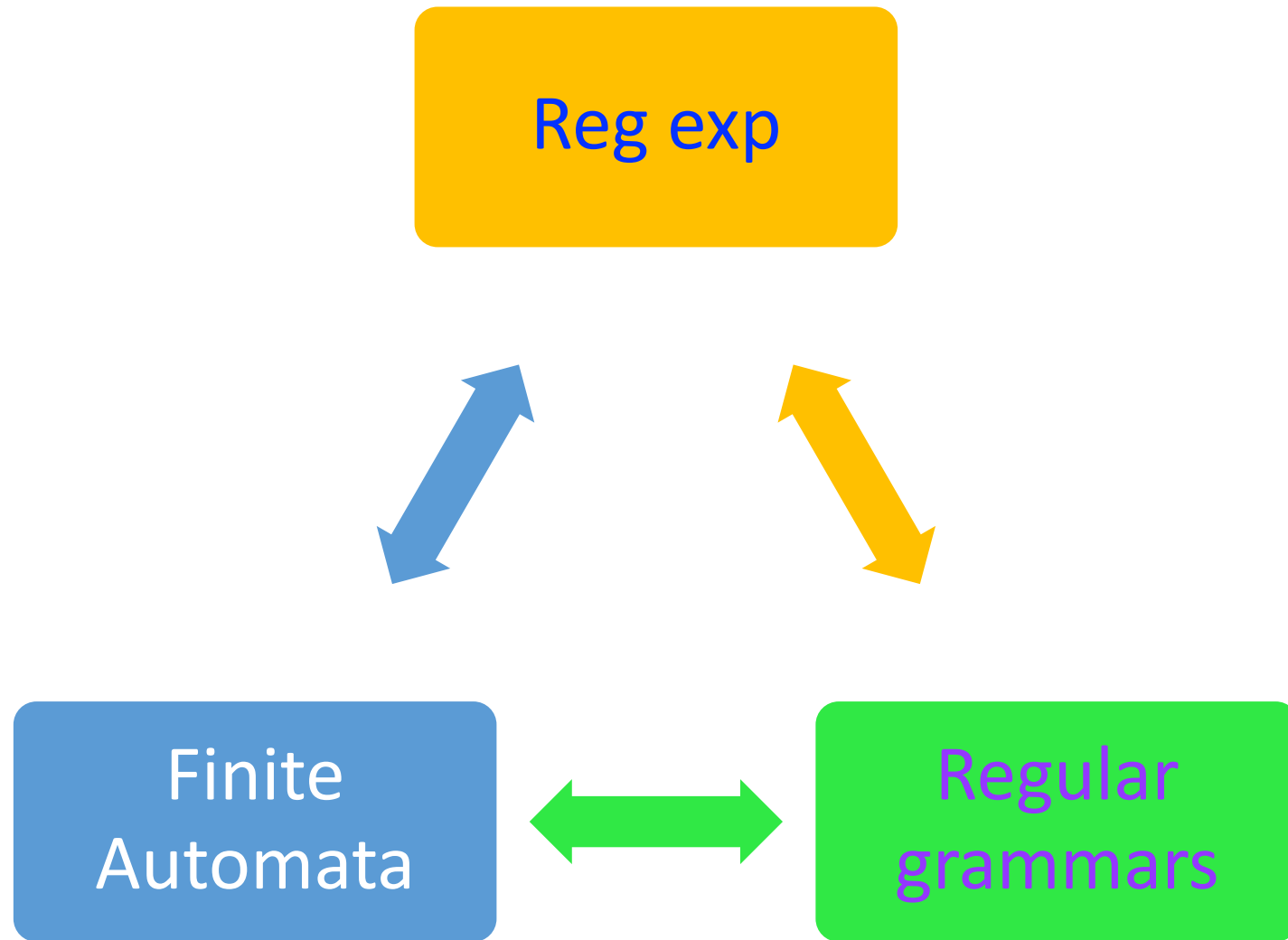


Course 4

Formal Languages

- *Regular Languages* -



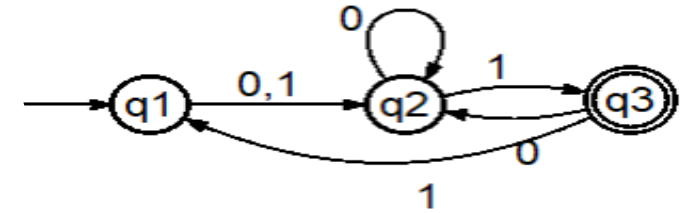
Theorem: A language is a regular set if and only if it is accepted by a FA

Proof:

=> Apply lemma 1 and lemma 2 (to follow, similar to RG)

<= construct a system of regular exp equations where:

- Indeterminants – states
- Coefficients – terminals
- Equation for A: all the possibilities that put the FA in state A
- Equation of the form: $X = Xa + b \Rightarrow$ solution $X = ba^*$



$$\begin{cases} q_1 = q_3 0 + \epsilon \\ q_2 = q_1 0 + q_1 1 + q_2 0 + q_3 0 \\ q_3 = q_2 1 \end{cases}$$

Regular exp = union of solutions corresponding to final states

Lemma 1': $\emptyset, \{\epsilon\}, \{a\}, \forall a \in \Sigma$ are accepted by FA

Reg exp	FA
\emptyset	$M = (Q, \Sigma, \delta, q_0, \emptyset)$
ϵ	$M = (Q, \Sigma, \emptyset, q_0, \{q_0\})$
$a, \forall a \in \Sigma$	$M = (\{q_0, q_1\}, \Sigma, \{\delta(q_0, a) = q_1\}, q_0, \{q_1\})$

Lemma 2': If L_1 and L_2 are accepted by a FA then:
 $L_1 \cup L_2$, $L_1 L_2$ and L_1^* are accepted by FA

Proof:

$M_1 = (Q_1, \Sigma_1, \delta_1, q_{01}, F_1)$ such that $L_1 = L(M_1)$

$M_2 = (Q_2, \Sigma_2, \delta_2, q_{02}, F_2)$ such that $L_2 = L(M_2)$

$M_3 = (Q_3, \Sigma_{1 \cup 2}, \delta_3, q_{03}, F_3)$

$Q_3 = Q_1 \cup Q_2 \cup \{q_{03}\}; \Sigma_3 = \Sigma_1 \cup \Sigma_2$

$F_3 = F_1 \cup F_2 \cup \{q_{03} \mid \text{if } q_{01} \in F_1 \text{ or } q_{02} \in F_2\}$

$\delta_3 = \delta_1 \cup \delta_2 \cup \{\delta_3(q_{03}, a) = p \mid \exists \delta_1(q_{01}, a) = p\} \cup$
 $\{\delta_3(q_{03}, a) = p \mid \exists \delta_2(q_{02}, a) = p\}$

$$L(M_3) = L(M_1) \cup L(M_2)$$

PROOF!!! Homework

$$M_4 = (Q_4, \Sigma_4, \delta_4, q_{04}, F_4)$$

$$Q_4 = Q_1 \cup Q_2; \quad q_{04} = q_{01};$$

$$F_3 = F_2 \cup \{q \in F_1 \mid \text{if } q_{02} \in F_2\}$$

$$\begin{aligned} \delta_3(q,a) &= \delta_1(q,a), \text{ if } q \in Q_1 - F_1 \\ &\quad \delta_1(q,a) \cup \delta_2(q_{02},a) \text{ if } q \in F_1 \\ &\quad \delta_2(q,a), \text{ if } q \in Q_2 \end{aligned}$$

$$L(M_3) = L(M_1)L(M_2)$$

PROOF!!! Homework

$$M_5 = (Q_5, \Sigma_1, \delta_5, q_{05}, F_5)$$

$$Q_5 = Q_1; \quad q_{05} = q_{01}$$

$$F_5 = F_1 \cup \{q_{01}\}$$

$$\begin{aligned} \delta_5(q,a) &= \delta_1(q,a), \text{ if } q \in Q_1 - F_1 \\ &\quad \delta_1(q,a) \cup \delta_1(q_{01},a) \text{ if } q \in F_1 \end{aligned}$$

//IDEA: concatenate with itself

$$L(M_3) = L(M_1)^*$$

PROOF!!! Homework

Pumping Lemma

- Not all languages are regular
- How to decide if a language is regular or not?
- Idea: pump symbols

Example: $L = \{0^n 1^n \mid n \geq 0\}$

Theorem: (Pumping lemma, Bar-Hillel)

Let L be a regular language. $\exists p \in \mathbb{N}$, such that if $w \in L$ with $|w| > p$, then

$w = xyz$, where $0 < |y| \leq p$

and

$xy^iz \in L, \forall i \geq 0$

Proof

L regular $\Rightarrow \exists M = (Q, \Sigma, \delta, q_0, F)$ such that $L = L(M)$

Let $|Q| = p$

If $w \in L(M)$: $(q_0, w) \xrightarrow{*} (q_f, \epsilon)$, $q_f \in F$ } process at least $p+1$ symbols
and }
 $|w| > p$ } p states

$\Rightarrow \exists q_1$ that appear in at least 2 configurations

$(q_0, xyz) \xrightarrow{*} (q_1, yz) \xrightarrow{*} (q_1, z) \xrightarrow{*} (q_f, \epsilon)$, $q_f \in F \Rightarrow 0 \leq |y| \leq p$

Proof (cont)

$$\begin{aligned}(q_0, xy^iz) & \vdash^* (q_1, y^iz) \\ & \vdash^* (q_1, y^{i-1}z) \\ & \vdash^* \dots \\ & \vdash^* (q_1, yz) \\ & \vdash^* (q_1, z) \\ & \vdash^* (q_f, \varepsilon), q_f \in F\end{aligned}$$

So, if $w=xyz \in L$ then $xy^iz \in L$, for all $i>0$

If $i=0$: $(q_0, xz) \vdash^* (q_1, z) \vdash^* (q_f, \varepsilon), q_f \in F$

Example: $L = \{0^n 1^n \mid n \geq 0\}$

Suppose L is regular $\Rightarrow w = xyz = 0^n 1^n$

Consider all possible decomposition \Rightarrow

Case 1. $y = 0^k$

$$xyz = 0^{n-k} 0^k 1^n; xy^i z = 0^{n-k} 0^{ik} 1^n \notin L$$

Case 2. $y = 1^k$

$$xyz = 0^n 1^k 1^{n-k}; xy^i z = 0^n 1^{ik} 1^{n-k} \notin L$$

Case 3. $y = 0^k 1^l$

$$xyz = 0^{n-k} 0^k 1^l 1^{n-l}; xy^i z = 0^{n-k} (0^k 1^l)^i 1^{n-l} \notin L$$

Case 4. $y = 0^k 1^K$

$$xyz = 0^{n-k} 0^k 1^K 1^{n-k}; xy^i z = 0^{n-k} 0^k 1^K 0^k 1^K \dots 1^{n-l} \notin L$$

$\Rightarrow L$ is not regular

Context free grammars (cfg)

Context free grammar (cfg)

- Productions of the form: $A \rightarrow \alpha$, $A \in N$, $\alpha \in (N \cup \Sigma)^*$
- More powerful
- Can model programming language:
 $G = (N, \Sigma, P, S)$ s.t. $L(G) = \text{programming language}$

Syntax tree

Definition: A syntax tree corresponding to a cfg $G = (N, \Sigma, P, S)$ is a tree obtained in the following way:

1. Root is the starting symbol S
2. Nodes $\in N \cup \Sigma$:
 1. Internal nodes $\in N$
 2. Leaves $\in \Sigma$
3. For a node A the descendants in order from left to right are X_1, X_2, \dots, X_n only if $A \rightarrow X_1 X_2 \dots X_n \in P$

Remarks:

- a) Parse tree = syntax tree – result of parsing (syntactic analysis)
- b) Derivation tree – condition 2.2 not satisfied
- c) Abstract syntax tree (AST) \neq syntax tree (semantic analysis)

Syntax tree (cont)

Property: In a cfg $G = (N, \Sigma, P, S)$, $w \in L(G)$ if and only if there exists a syntax tree with frontier w .

Proof: HomeWork

Example: $S \rightarrow aSbS \mid c$; $w = aacbcabc$

Leftmost derivations

$S \Rightarrow aSbS \Rightarrow aaSbSbS \Rightarrow aacbSbS$
 $\Rightarrow aacbcS \Rightarrow aacbcabc$

Rightmost derivations

$S \Rightarrow aSbS \Rightarrow aSbc \Rightarrow aaSbSbc$
 $\Rightarrow aaSbcbcb \Rightarrow aacbcabc$