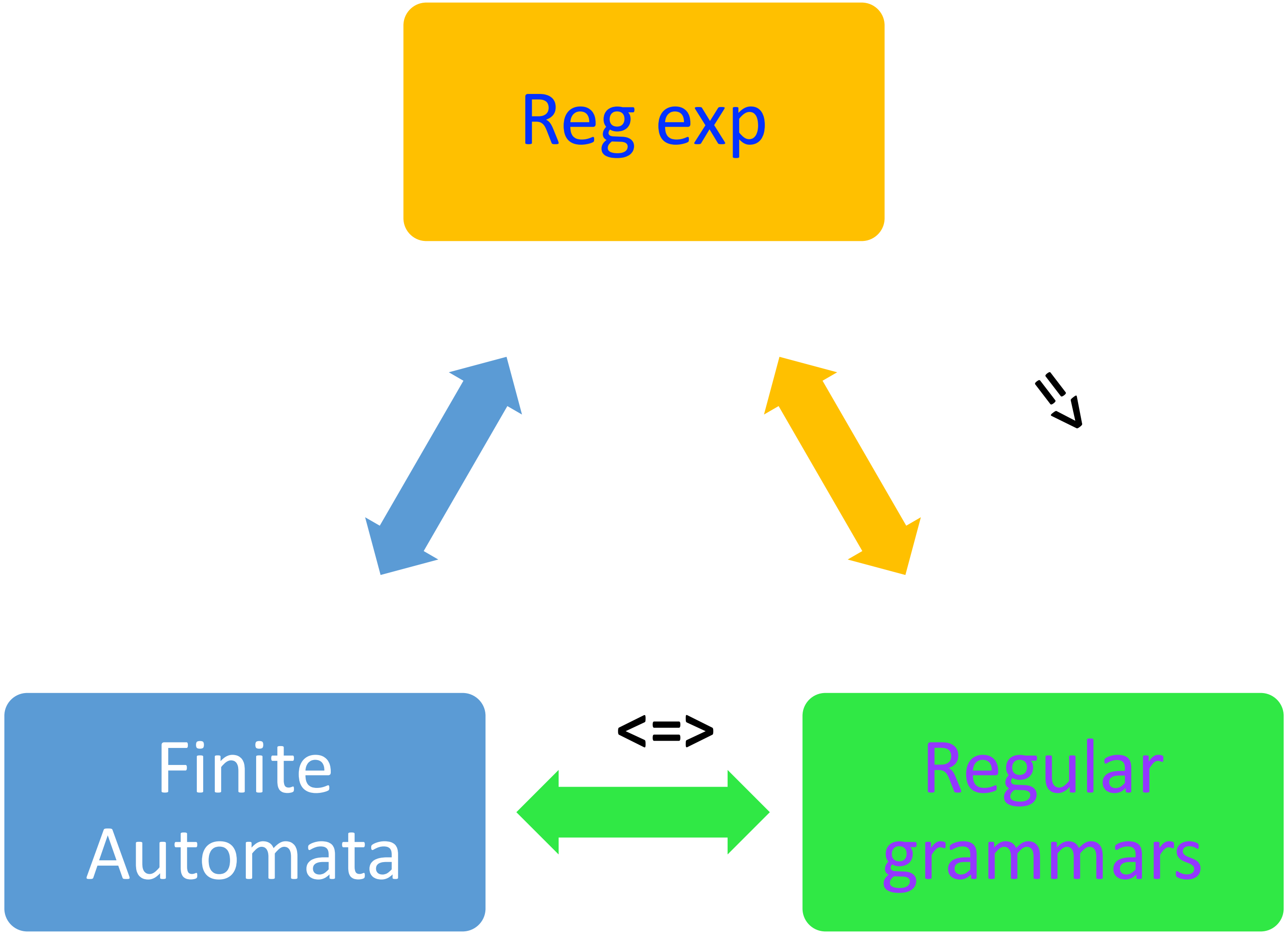
Course 4

Formal Languages

- Regular Languages -


Reg exp
=>
<=>
Finite Automata
Regular grammars

Prop: *Regular sets are right linear languages*

Lemma 1': $\boldsymbol{\Phi}$,{$\boldsymbol{\varepsilon}$}, {a},∀a∈$\boldsymbol{\Sigma}$ are right linear languages

Proof: constructive

i. G = ({S}, $\boldsymbol{\Sigma}$, $\boldsymbol{\Phi}$, S) – regular grammar such that L(G) = $\boldsymbol{\Phi}$

ii. G = ({S}, $\boldsymbol{\Sigma}$,{S→$\boldsymbol{\varepsilon}$}, S) – regular grammar such that L(G) ={$\boldsymbol{\varepsilon}$}

iii. G = ({S}, $\boldsymbol{\Sigma}$,{S→a}, S) – regular grammar such that L(G) ={a}

Lemma 2': If L_1 and L_2 are right linear languages then:

$L_1 \cup L_2$, L_1L_2 and L_1^* are right linear languages.

Proof: constructive

L_1, L_2 right linear languages => $\exists G_1, G_2$ such that

$G_1 = (N_1, \Sigma_1, P_1, S_1)$ and $L_1 = L(G_1)$

$G_2 = (N_2, \Sigma_2, P_2, S_2)$ and $L_2 = L(G_2)$ assume $N_1 \cap N_2 = \emptyset$

i. $G_3 = (N_3, \boldsymbol{\Sigma}, P_3, S_3)$

$N_3 = N_1 \cup N_2 \cup \{S_3\}$; $\Sigma_3 = \Sigma_1 \cup \Sigma_2$

$P_3 = P_1 \cup P_2 \cup \cancel{\{S_3 \rightarrow S_1 \mid S_2\}}$

$\{S_3 \rightarrow \boldsymbol{\alpha}_1 \mid S_1 \rightarrow \boldsymbol{\alpha}_1 \in P_1\} \cup \{S_3 \rightarrow \boldsymbol{\alpha}_2 \mid S_2 \rightarrow \boldsymbol{\alpha}_2 \in P_2\}$

G_3 – right linear language

and

$L(G_3) = L(G_1) \cup L(G_2)$

PROOF!!! Homework

ii. $G_4 = (N_4, \boldsymbol{\Sigma}, P_4, S_4)$

$N_4 = N_1 \cup N_2$; $S_4 = S_1$; $\Sigma_4 = \Sigma_1 \cup \Sigma_2$

$P_4 =$ $\{A \to aB \mid \text{if } A \to aB \in P_1\} \cup$
$\{A \to aS_2 \mid \text{if } A \to a \in P_1\} \cup$
$P_2 \cup$
$\{S_1 \to \boldsymbol{\alpha}_2 \mid \text{if } S_1 \to \boldsymbol{\varepsilon} \in P_1 \text{ and } S_2 \to \boldsymbol{\alpha}_2 \in P_2\}$

G_4 – right linear language

and

$\mathbf{L(G_4) = L(G_1)\, L(G_2)}$

PROOF!!! Homework

iii. $G_5 = (N_5, \boldsymbol{\Sigma}_1, P_5, S_5)$

//IDEA: concatenate L_1 with itself

$N_4 = N_1 U \{S_5\};$

$P_5 =$ P_1 U $\{S_5 \rightarrow \boldsymbol{\varepsilon}\}$ U

$\{S_5 \rightarrow \boldsymbol{\alpha}_1 | S_1 \rightarrow \boldsymbol{\alpha}_1 \in P_1\}$ U

$\{A \rightarrow aS_1 | \text{ if } A \rightarrow a \in P_1\}$

G_5 – right linear language

and

$L(G_5) = L(G_1)^*$

PROOF!!! Homework

Theorem: A language is a regular set if and only if is a right linear language

Proof:

=> Apply lemma 1' and lemma 2'

<= construct a system of regular exp equations where:

- Indeterminants – nonterminals
- Coefficients – terminals
- Equation for A: all the possible rewritings of A

Example: G=({S,A,B},{0,1}, P, S)

P: $S \to 0A \mid 1B \mid \boldsymbol{\varepsilon}$

$A \to 0B \mid 1A$

$B \to 0B \mid 1$

$$\begin{cases} S = 0A + 1B + \boldsymbol{\varepsilon} \\ A = 0B + 1A \\ B = 0B + 1 \end{cases}$$

Regular exp = solution corresponding to S

Pumping Lemma

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

- Idea: pump symbols

Example: $L = \{0^n1^n \mid n >= 0\}$

Theorem: (Pumping lemma, Bar-Hillel)

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

$$w = xyz, \text{ where } 0<|y|<=p$$

and

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

Proof

L regular => $\exists$ M = (Q,$\boldsymbol{\Sigma}$,$\boldsymbol{\delta}$, q_0, F) such that L= L(M)

Let |Q| = p

If w $\in$ L(M): $(q_0,w) \overset{*}{\vdash} (q_f , \boldsymbol{\varepsilon})$, $q_f \in F$

and

|w|>p

process at least p+1 symbols

p states

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

$(q_0,xyz) \overset{*}{\vdash} (q1,yz) \overset{+}{\vdash} (q1,z) \overset{*}{\vdash} (q_f , \boldsymbol{\varepsilon})$, $q_f \in F$ => 0<=|y|<=p

Proof (cont)

$(q_0,xy^iz) \overset{*}{\vdash} (q_1,y^iz)$

$\overset{*}{\vdash} (q_1,y^{i-1}z)$

$\overset{*}{\vdash} \ldots$

$\overset{*}{\vdash} (q_1,yz)$

$\overset{*}{\vdash} (q_1, z)$

$\overset{*}{\vdash} (q_f , \boldsymbol{\varepsilon}) , q_f \in F$

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

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

Example: L = $\{0^n1^n \mid n >= 0\}$

Suppose L is regular => w= xyz = 0^n1^n

Consider all possible decomposition =>

Case 1. $y = 0^k$

$$xyz = 0^{n-k}0^k1^n;\ xy^iz = 0^{n-k}0^{ik}1^n \notin L$$

Case 2. $y = 1^k$

$$xyz = 0^n1^k1^{n-k};\ xy^iz = 0^n1^{ik}1^{n-k} \notin L$$

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

$$xyz = 0^{n-k}0^k1^l1^{n-l};\ xy^iz = 0^{n-k}(0^k1^l)^i1^{n-l} \notin L$$

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

$$xyz = 0^{n-k}0^k1^k1^{n-k};\ xy^iz = 0^{n-k}0^k1^k0^k1^k\ldots1^{n-l} \notin L$$

=> L is not regular

Context free grammars (cfg)

Context free grammar (cfg)

- Procdutions of the form: A $\rightarrow \alpha$, A$\in$N, $\alpha \in (N \cup \boldsymbol{\Sigma})^*$

- More powerful

- Can model programming language:

  G = (N, $\boldsymbol{\Sigma}$,P,S) s.t. L(G) = 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 $X_1, X_2, \ldots, X_n$ only if $A \rightarrow X_1X_2\ldots X_n \in P$

Remarks:

a) Parse tree = syntax tree – result of parsing (syntatic analysis)

b) Derivation tree – condition 2.2 not satisfied

c) Abstract syntax tree (AST) ≠ 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-> aSbS | c; w = aacbcbc

Leftmost derivations	Rightmost derivations
S => aSbS => aaSbSbS => aacbSbS => aacbcbS => aacbcbc	S => aSbS => aSbc => aaSbSbc => aaSbcbc => aacbcbc

Definition: A cfg G = (N, Σ,P,S) is ambigous if for a w ∊ L(G) there exists 2 distinct syntax tree with frontier w.

Example: