Morning

Equivalence of states

- *equivalent states*

$$\forall w \in \Sigma^*, \hat{\delta}(p,w) \in F \Leftrightarrow \hat{\delta}(q,w) \in F$$

- *notice*

We never mentioned $\hat{\delta}(p,w)$ = $\hat{\delta}(q,w)$

- *distinguishable states*

$$\exists w \in \Sigma^*, \hat{\delta}(p,w) \in F \Leftrightarrow \neg\hat{\delta}(q,w) \in F$$

Table-filling algorithm

- *Basis* If p is accepting and q is not accepting, then p and q are distinguishable.

- *Induction* Let $r = \delta(p, a)$, $s = \delta(q, a)$, r and s are distinguishable. Then p and q are distinguishable.

- *Example*

	ε	0	1	00	01	10	11	000	001	010	100	011	101	110
0														
1	×	×												
00			×											
01	×	×		×										
10	×	×	×	×	×									
11	×	×	×	×	×	×								
000			×		×	×	×							
001	×	×		×		×	×	×						
010	×	×			×		×	×	×					
100			×		×	×	×		×	×				
011	×	×	×	×	×	×		×	×	×	×			
101	×	×	×	×	×	×		×	×	×	×			
110	×	×	×	×	×	×		×	×	×	×			
111	×	×	×	×	×	×		×	×	×	×			

Minimization of DFA's

- what is minimization of DFA

- algorithm for minimization
  - partition remaining states into equivalent blocks
  - take blocks as states

- minimum-state DFA for a regular language is unique

Left/Right Most Derivations

$L = \{a^{2n}b^m \mid n \geq 0, m \geq 0\}$

$S \rightarrow AB$, $A \rightarrow \varepsilon | aaA$, $B \rightarrow \varepsilon | Bb$

for $w = aabb$:

$S \Rightarrow AB \Rightarrow aaAB \Rightarrow aaABb \Rightarrow aaBb \Rightarrow aaBbb \Rightarrow aabb$

Left most :

$S \Rightarrow AB \Rightarrow aaAB \Rightarrow aaB \Rightarrow aaBb \Rightarrow aaBbb \Rightarrow aabb$

Right most :

$S \Rightarrow AB \Rightarrow ABb \Rightarrow ABbb \Rightarrow Abb \Rightarrow aaAbb \Rightarrow aabb$

Parse Tree

Let $G = (V, T, S, P)$ be a CFG. A tree is a parse tree for G if :

1. Each interior node is labeled by a variable in V

2. Each leaf is labeled by a symbol in $T \cup \{\varepsilon\}$. Any ε-labeled leaf is the only child of its parent.

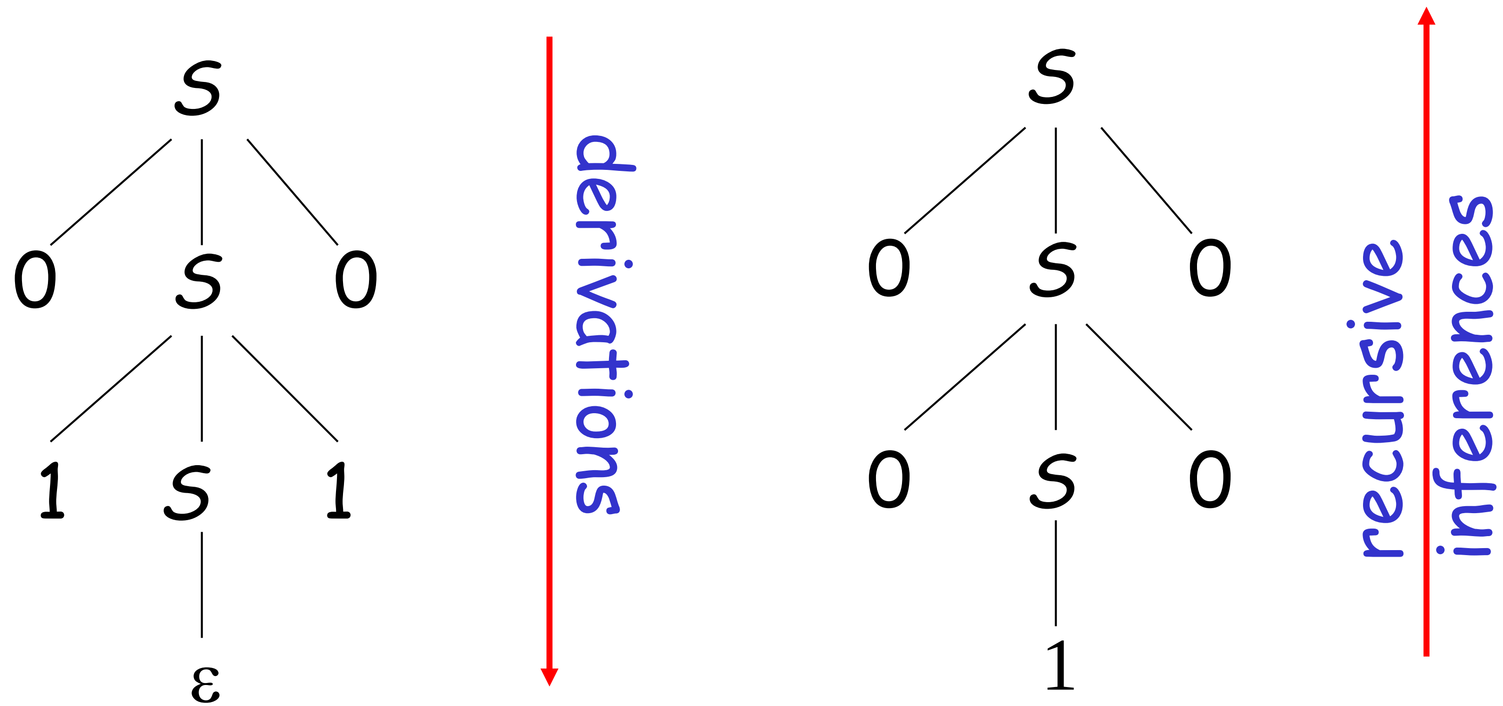
3. If an interior node is labeled A, and its children (from left to right) labeled $x_1, x_2, \ldots, x_k$,

Then $A \to x_1, x_2, \ldots, x_k \in P$.

Parse Tree

Example $L=\{\, w \mid w \in \{0,1\}^* \text{ and } w = w^R \,\}$

$$S \rightarrow \varepsilon \mid 0 \mid 1 \mid 0S0 \mid 1S1$$



w=0110 w=00100

Ambiguity

$G = (\{E, I\}, \{a, b, (,), +, *\}, E, P)$

$E \to I \mid E + E \mid E * E \mid (E), \quad I \to a \mid b$

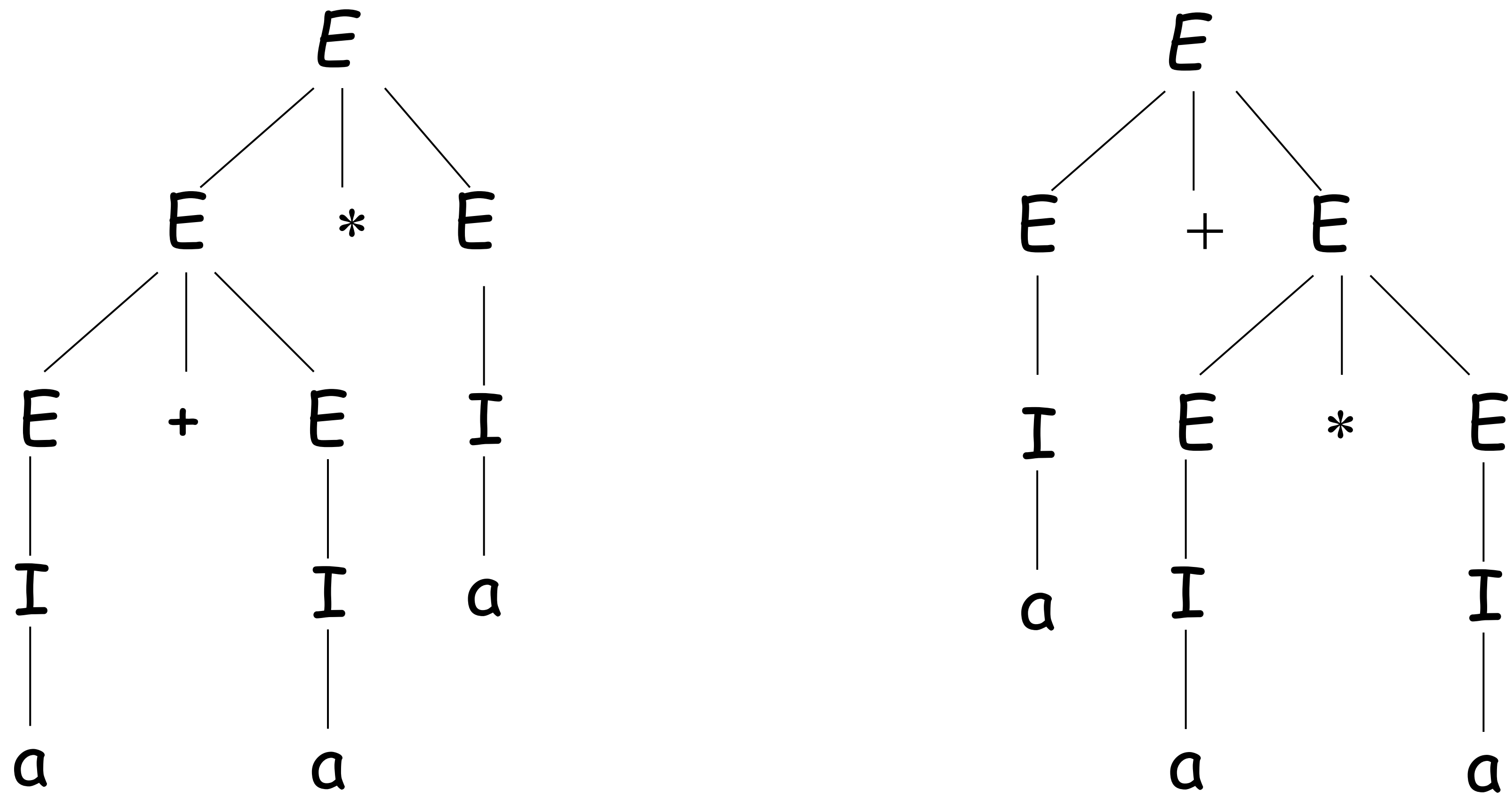
Derivation for $w = a + a * a$:

$E \Rightarrow E*E \Rightarrow E+E*E \Rightarrow I+E*E \Rightarrow a+E*E \overset{*}{\Rightarrow} a+a*a$

$E \Rightarrow E+E \Rightarrow I+E \Rightarrow a+E \Rightarrow a+E*E \overset{*}{\Rightarrow} a+a*a$

Ambiguity

parse-tree for $w = a + a * a$:



Removing Ambiguity

$E \to I \mid E+E \mid E*E \mid (E)$, $I \to a \mid b$

$E \to T|E+T$, $T \to F|T*F$, $F \to I|(E)$, $I \to a|b|Ia|Ib$

Left most derivation for $w = a + a * a$:

$$E \Rightarrow E+T \Rightarrow T+T \Rightarrow F+T \Rightarrow I+T \Rightarrow a+T \Rightarrow a+T*F$$

$$\Rightarrow a+F*F \Rightarrow a+I*F \Rightarrow a+a*F \Rightarrow a+a*I \Rightarrow a+a*a$$

$$E \Rightarrow T \Rightarrow T*T \Rightarrow (E)*T \Rightarrow (E+T)*T \overset{*}{\Rightarrow} (a+a)*a$$

Inherent Ambiguity

- What is inherent ambiguity

A CFL *L* is said to be *inherently ambiguous* if all grammars that generate it is ambiguous.

Example Let $L=\{ w \mid w \in \{0,1\}^* \text{ and } n_0(w)=n_1(w) \}$

L is not inherently ambiguous ,because there is an unambiguous CFG :

$$S \rightarrow \varepsilon \mid 0S1 \mid 1S0 \mid 0S11S0 \mid 1S00S1$$

Example

$L=\{a^n b^n c^m d^m \mid n\geq 1, m\geq 1\} \cup \{a^n b^m c^m d^n \mid n\geq 1, m\geq 1\}$

The CFG for L is :

$S\rightarrow AB \mid C$, $A\rightarrow aAb \mid ab$, $B\rightarrow cBd \mid cd$

$C\rightarrow aCd \mid aDd$, $D\rightarrow bDc \mid bc$

Let w= *abcd* , there are two left most derivations

$S\Rightarrow AB \Rightarrow abB \Rightarrow abcd$

$S\Rightarrow C \Rightarrow aDd \Rightarrow abcd$

Simplification of CFG

Why & what :

$S \to A \mid B,\ \ A \to 1CA \mid 1DE \mid \varepsilon,\ \ B \to 1CB \mid 1DF,$

$C \to 1CC \mid 1DG \mid 0G,\ D \to 1CD \mid 1DH \mid 0H,$

$E \to 0A,\ F \to 0B,\ G \to \phi,\ H \to 1$

- ε-productions
- unit productions
- useless symbols and productions

Example $G: S \to A \mid B,\ A \to 1CA \mid 1DE \mid \varepsilon$

$B \to 1CB \mid 1DF,\ C \to 1CC \mid 1DG \mid 0G,\ D \to 1CD \mid 1DH \mid 0H,$

$E \to 0A,\ F \to 0B,\ G \to \phi,\ H \to 1$

- eliminating ε-productions

  the only one : $A \to \varepsilon$

  $S \to A \mid B,\ A \to 1CA \mid 1C \mid 1DE,\ B \to 1CB \mid 1DF,$

  $C \to 1CC \mid 1DG \mid 0G,\ D \to 1CD \mid 1DH \mid 0H,$

  $E \to 0A \mid 0,\ F \to 0B,\ G \to \phi,\ H \to 1$

$S \to A \mid B$, $A \to 1CA \mid 1C \mid 1DE$, $B \to 1CB \mid 1DF$,

$C \to 1CC \mid 1DG \mid 0G$, $D \to 1CD \mid 1DH \mid 0H$,

$E \to 0A \mid 0$, $F \to 0B$, $G \to \phi$, $H \to 1$

- eliminating unit productions

  the only two : $S \to A$ and $S \to B$

  $S \to 1CA \mid 1C \mid 1DE \mid 1CB \mid 1DF$,

  $A \to 1CA \mid 1C \mid 1DE$, $B \to 1CB \mid 1DF$,

  $C \to 1CC \mid 1DG \mid 0G$, $D \to 1CD \mid 1DH \mid 0H$,

  $E \to 0A \mid 0$, $F \to 0B$, $G \to \phi$, $H \to 1$

$S \to 1CA \mid 1C \mid 1DE \mid 1CB \mid 1DF$,

$A \to 1CA \mid 1C \mid 1DE$, $B \to 1CB \mid 1DF$,

$C \to 1CC \mid 1DG \mid 0G$, $D \to 1CD \mid 1DH \mid 0H$,

$E \to 0A \mid 0$, $F \to 0B$, $G \to \phi$, $H \to 1$

- eliminating useless symbols and productions

$S \to 1DE$,$A \to 1DE$, $D \to 1DH \mid 0H$, $E \to 0A \mid 0$, $H \to 1$

Chomsky Normal Form(CNF)

1. $A \to BC$;
2. $A \to a$.

$S \to 1DE$,$A \to 1DE$, $D \to 1DH \mid 0H$, $E \to 0A \mid 0$, $H \to 1$

Chomsky normal form :

$S \to IE$,$A \to IE$, $D \to IH|EH$, $E \to EA|0$, $I \to HD$, $H \to 1$

$D \to IH|FH$, $E \to FA|0$, $F \to 0$

Automata		Languages		Grammars
Construction		Properties		Design
Finite Automaton	→	Regular Language	←	Regular Expression
↓	Recognize	↓	Generate	↓
Push Down Automaton	→	Context Free Language	←	Context Free Grammar
↓		↓		↓
Turing Machine	→	Recursively Enumerable	←	(Phrase Grammar)

Good good study

day day up !