

$E \rightarrow E + E$
 $E \rightarrow E * E$
 $E \rightarrow id$

$id + id * id$

D_1
 D_2

$D_1 < D_2$

$E \rightarrow E + E \rightarrow id + E \rightarrow id + E * E \rightarrow id + id * E \rightarrow id + id * id$
 $E \rightarrow E + E \rightarrow id + E \rightarrow id + E * E \rightarrow id + E * id \rightarrow id + id * id$

$D_1 > D_2$

Ambiguity

A grammar G is said to be *ambiguous* if it can generate strings which have more than one distinct parse tree.

LMD and RMD

Inherently ambiguous languages: CFL which have all CFGs that can generate them must be ^Iambiguous.

Fortunately programming languages are never inherently ambiguous.

non programming lang

PL

ambiguous

$G_1 \rightarrow L$
 $G_2 \rightarrow L$

eliminate left recursion or sth

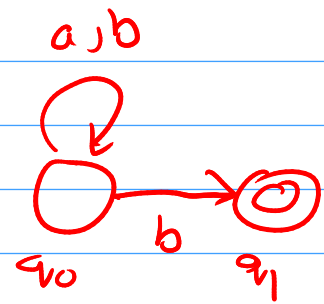
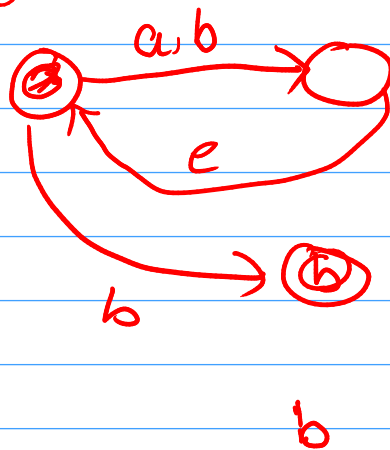
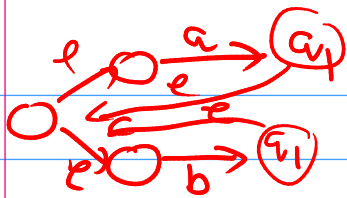
$(a \cup b) b$

$(a \cup b)(a(a \cup b))^* b a^*$

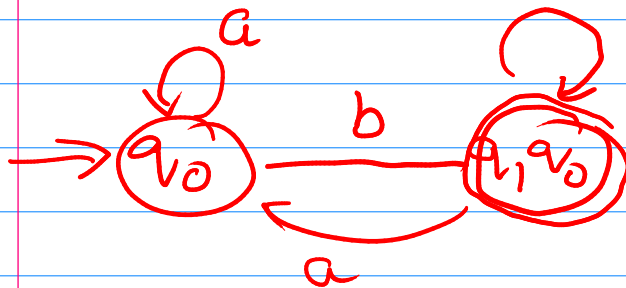
$[A B C G] [D E F]$

$\{A\} \{B\} \{C\} \{G\} \quad \{D\} \{E\} \{F\}$

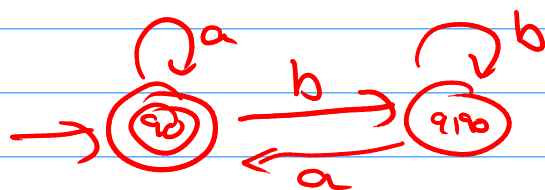
$\{A\} \{B\} \{C\} \{G\} \quad \{D\} \{E\} \{F\}$



	a	b
q ₀	q ₀	q ₁ , q ₀
q ₁	∅	∅
∅	∅	∅
q ₁ , q ₀	q ₀	q ₁ , q ₀



ABCG, DEFF



bbb

	a	b
A	B	C
B	B	D
C	E	B
D	E	F
E	F	D
F	G	E
G	E	C

A, B, CG, DE, F
A, B, C, G, D, E, F