

Parsing - Part II

Complete Course on Compiler Design

$$E \rightarrow E + E \mid E * E \mid E / E \mid E - E \mid (E) \mid id$$

$\begin{pmatrix} + < * \\ - < / \end{pmatrix}$ \Downarrow E can be according to C-Lang.

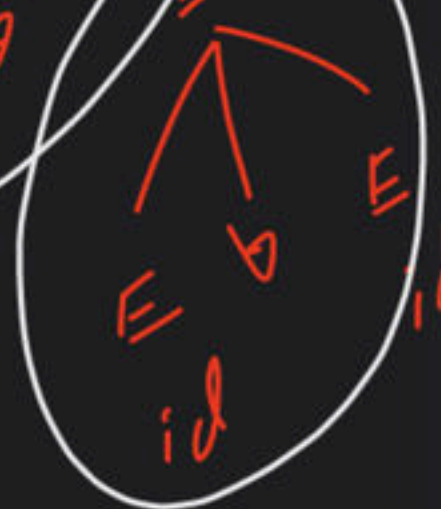
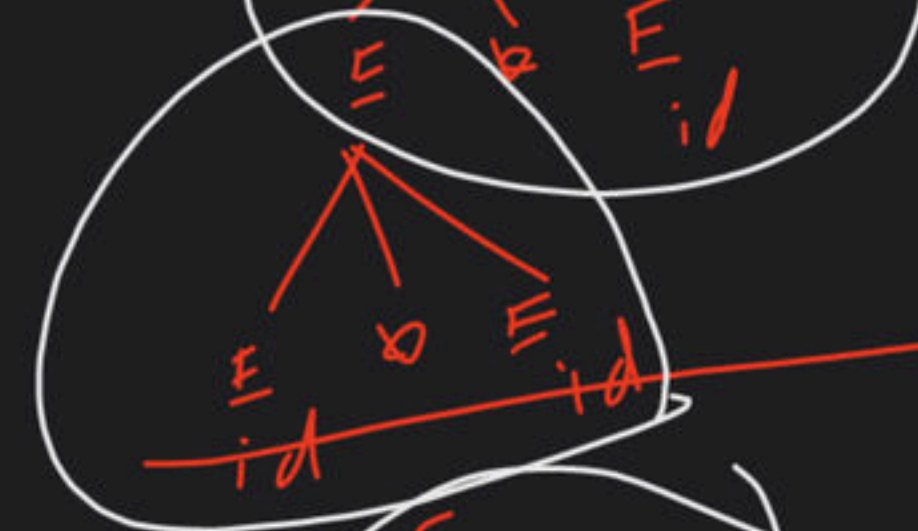
$$E \rightarrow E + T \mid E - T \mid T$$

$$T \rightarrow T * F \mid T / F \mid F$$

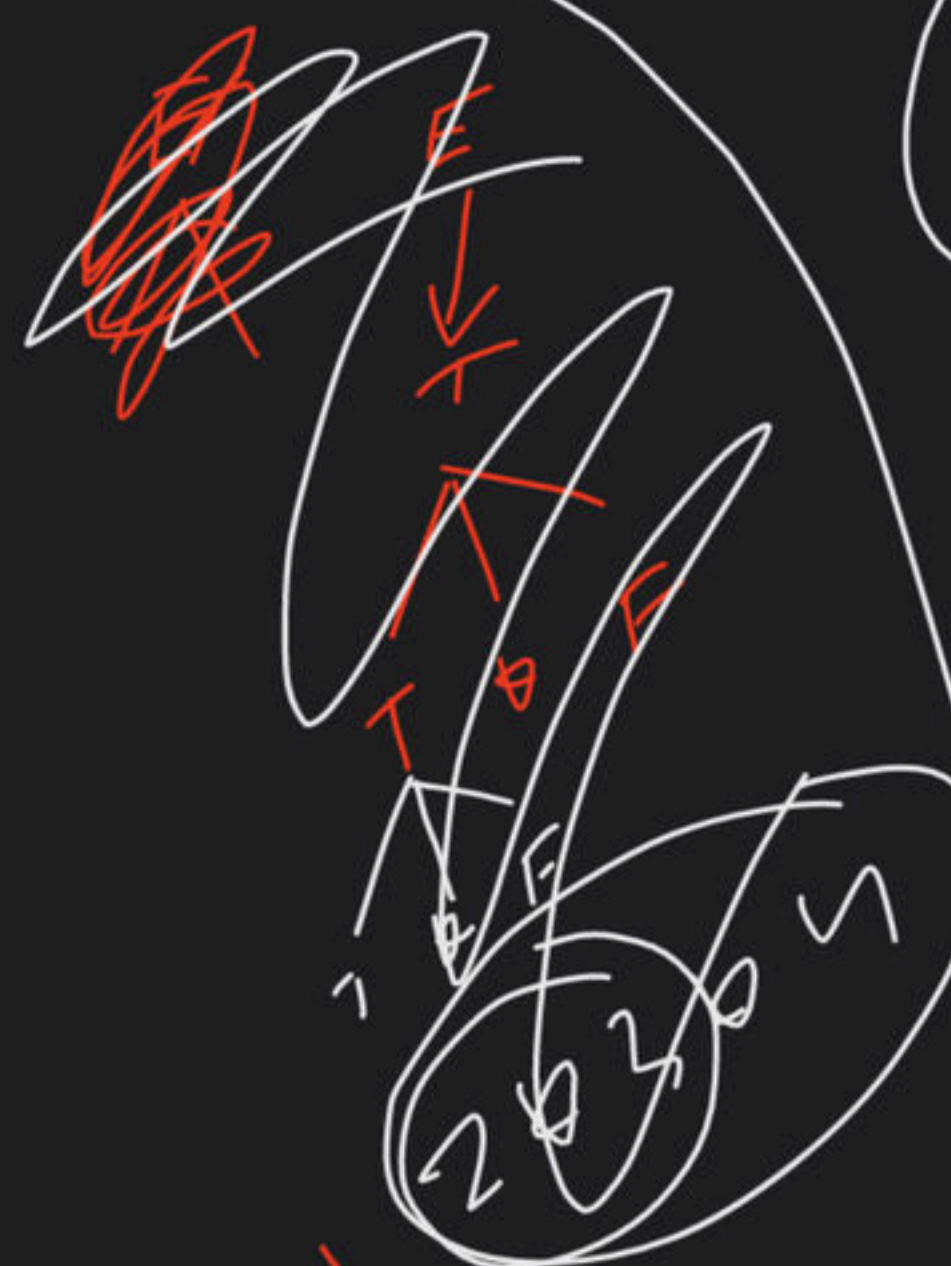
$$E \rightarrow id \mid (E)$$

$\leftarrow id * id * id$

L-R



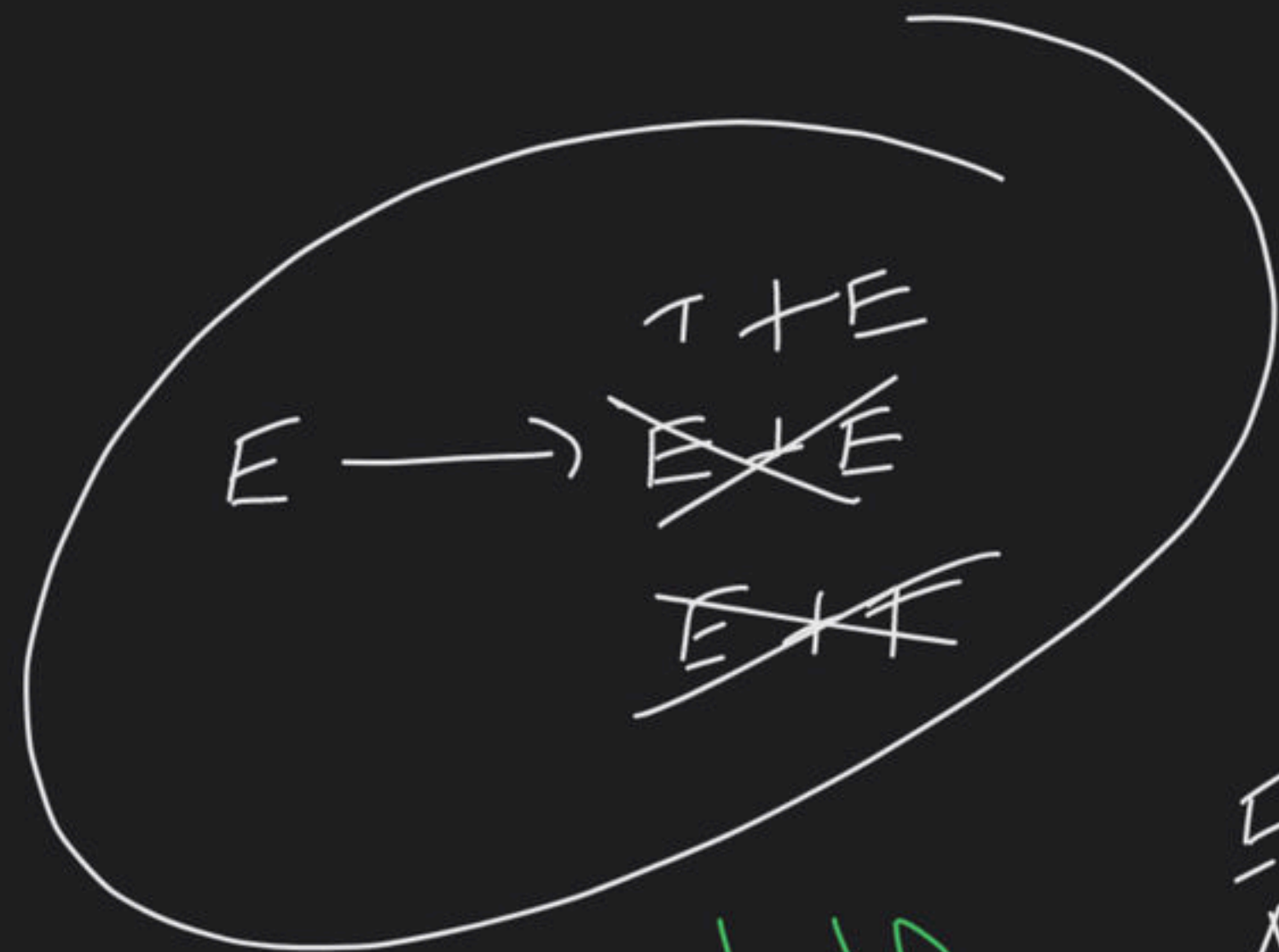
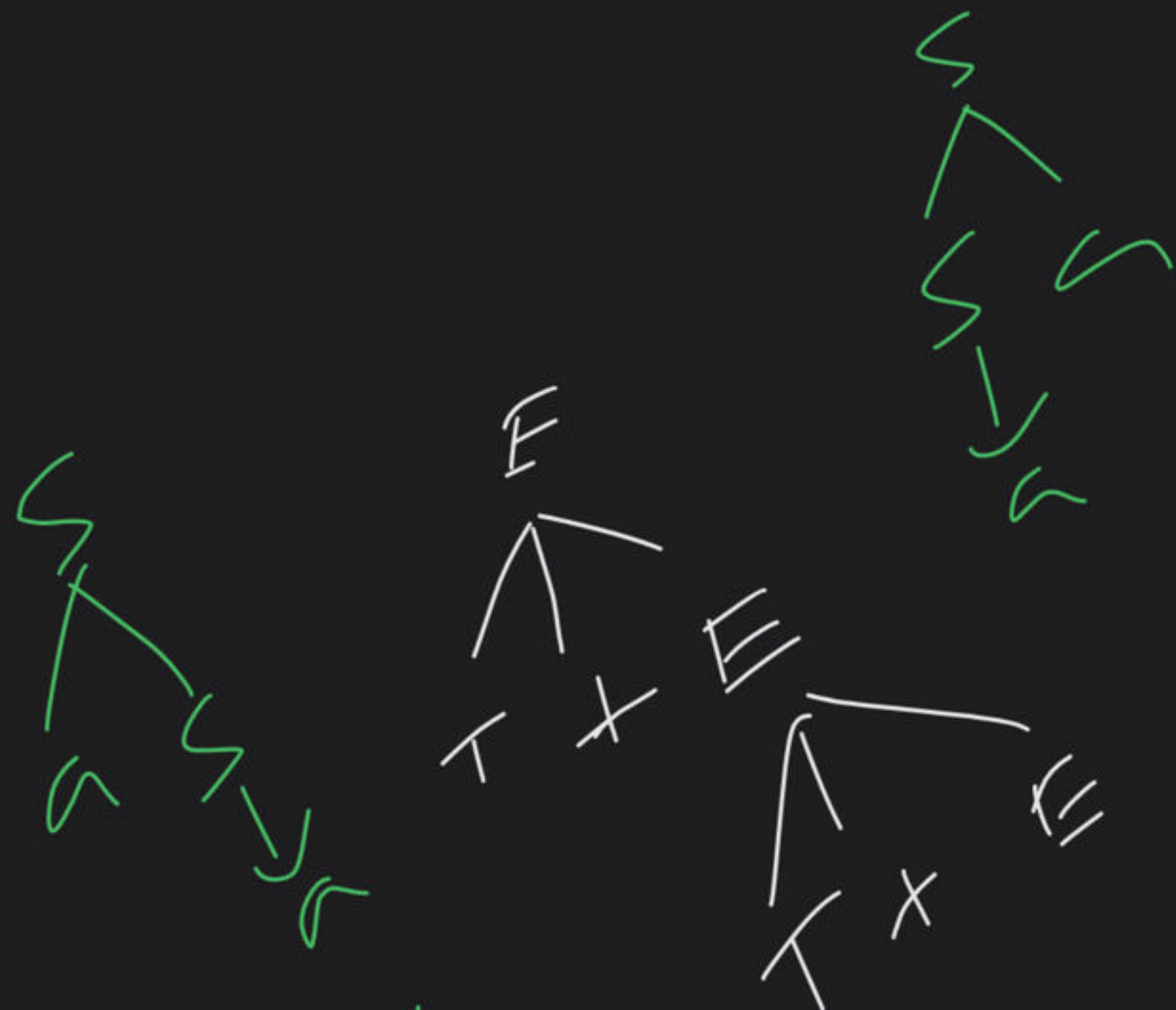
R-L



ex

Write Unambiguous CGL for the following rules

Priority	Operator	Associativity	Rule
3	$\&$	L-R	$A \rightarrow B - A / B$
1 (L)	-	R-L	$B \rightarrow B @ C / C$
5	+, /	L-R	$C \rightarrow C \& D / D$
2	@	L-R	$D \rightarrow E \oplus D / E$
4	\oplus	R-L	$E \rightarrow E + F / E / F / F$
6 (H)	$\#, \$$	L-R	$F \rightarrow F \# G / F \$ G / G$
			$G \rightarrow \text{id} (A)$



$S \rightarrow a s b a / a$
aa

$S \rightarrow A B C D$
 $D \rightarrow E F G H$
 $E \rightarrow B C D$



ex

$$B \rightarrow B \cup B \mid B \text{ and } B \mid \text{not } B \mid (B) \mid 0 \mid 1$$

\Downarrow Equivalent grammar CFL
 according to C-Lang (not(1)
 and(2) L-R
 or(3)

$$B \rightarrow B \cup C \mid C$$

$$C \rightarrow C \text{ and } D \mid D$$

$$D \rightarrow \text{not } D \mid E$$

$$E \rightarrow 0 \mid 1 \mid (B)$$

Left Factoring

$S \rightarrow \underline{a}b \mid \underline{a}c \mid \underline{a}d$

i/p: ad



Elimination of
Left Factoring

$S \rightarrow as$
 $s \rightarrow b \mid c \mid d$

Left Factored
Grammar

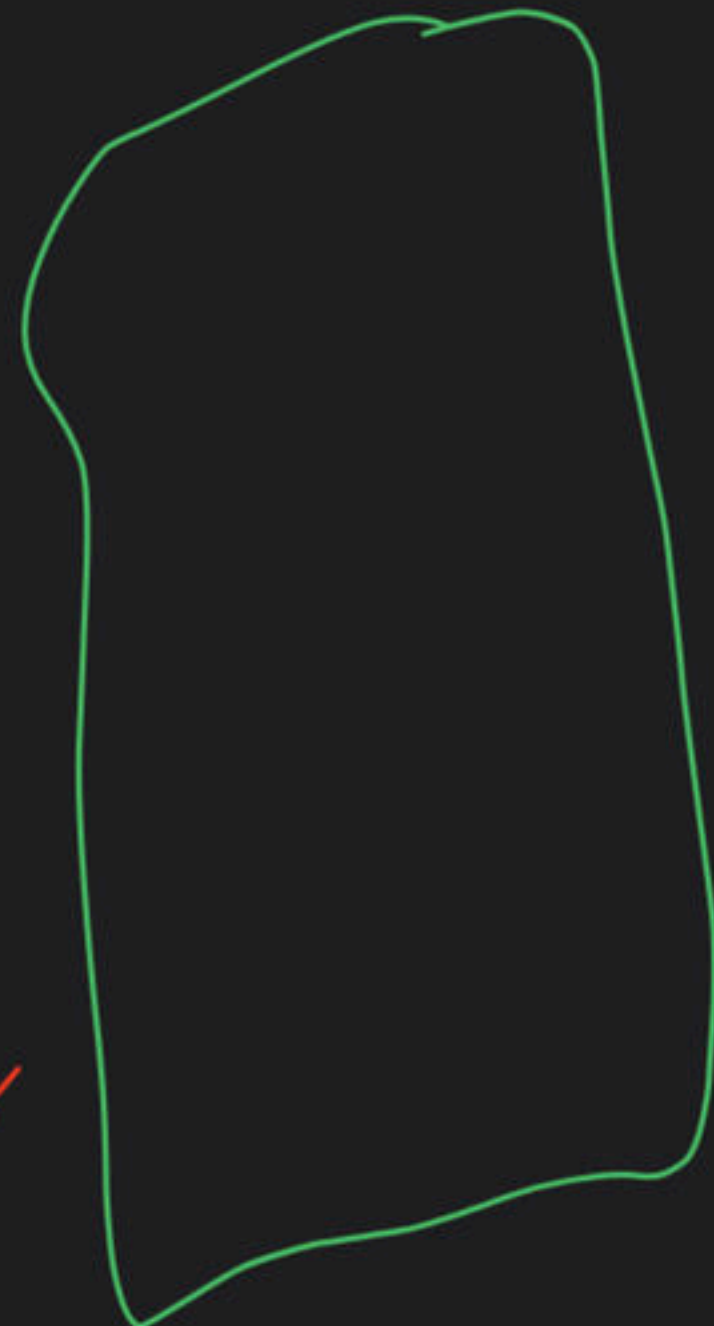
Proof

$\rightarrow a$

then pro

~~ad~~ \rightarrow

or i



ex

$S \rightarrow i E k S \mid i E k S \mid a \mid b$

$E \rightarrow d$

\downarrow

$S \rightarrow a \mid b \mid i E k$

$S \rightarrow \epsilon \mid e s$

$E \rightarrow d$

~~$S \rightarrow a \mid b \mid i E k S$~~

~~$S \rightarrow \epsilon \mid e s$~~

$a b c d . \epsilon \Rightarrow a b c d$

ex

$S \rightarrow \underline{a} / \underline{ab} / \underline{abc} / \underline{abcd} / \epsilon / f / g$

$S \rightarrow \epsilon / f / g / a S'$

$S' \rightarrow \epsilon / b S''$

$S'' \rightarrow \epsilon / c S'''$

$S''' \rightarrow \epsilon / d$

Extended Backus Normal Form
Left Recursive

Recursive gram

Right Recursive

$S \rightarrow \underline{sa}/b$ $[b, ba, baa]$

$S \rightarrow as/b$

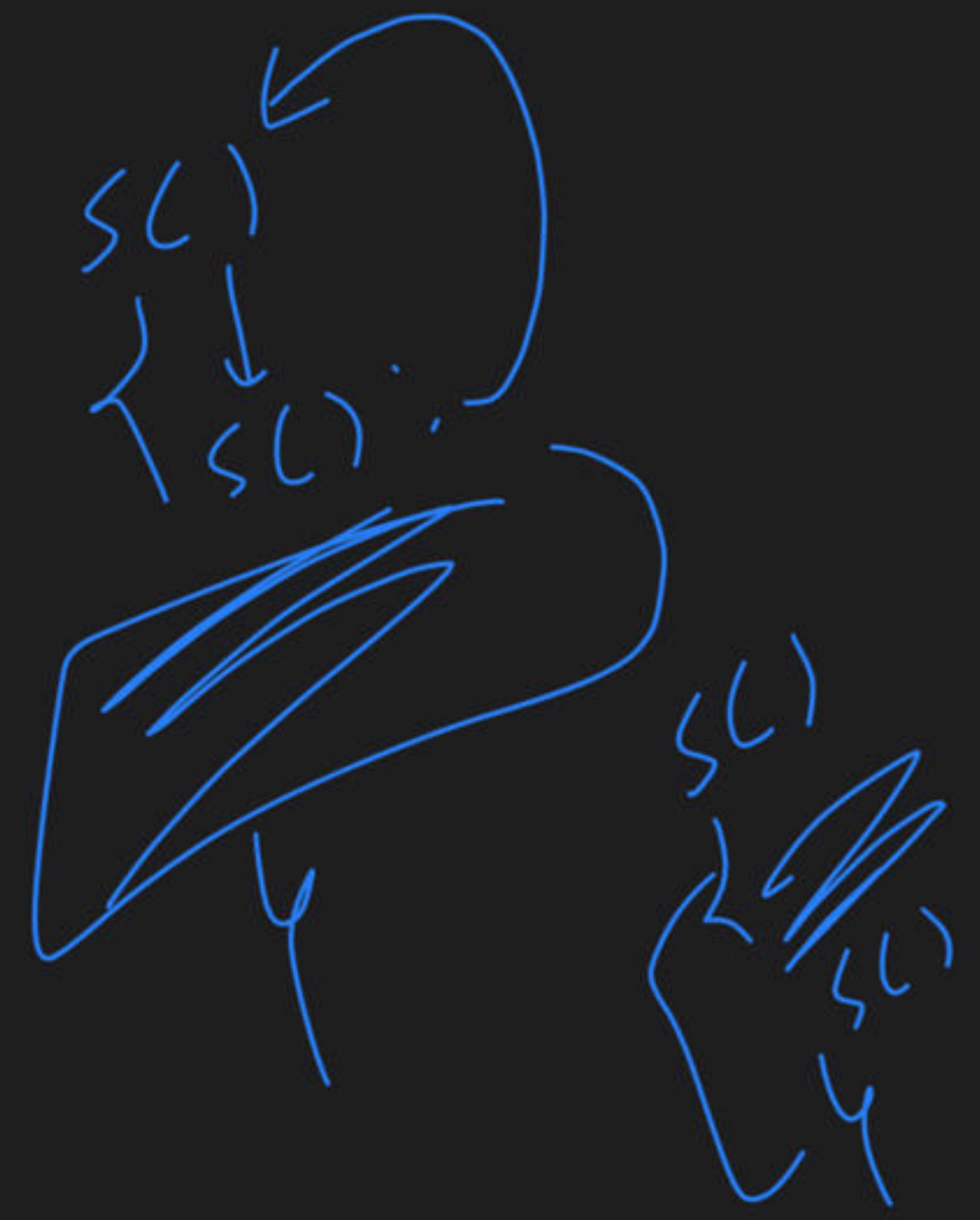
eliminate LR using ERNF

$S \rightarrow b|a|$

(a)

$b \Rightarrow ba \Rightarrow b|a|$

$S \rightarrow bs$
 $\underline{S} \rightarrow a\underline{s}|e$



ex

$S \rightarrow sa|b|c \Rightarrow b|c|ba|ca \Rightarrow$

$S \rightarrow b|a|c|d$

EBNF

\Downarrow

$S \rightarrow bs'|cs'$

$s' \rightarrow as'|e$

$S \rightarrow sa|sb|c|d|e$

\Downarrow

$S \rightarrow cs'|ds'|es'$

$s' \rightarrow as'|e|bs'$

Parser

Top Down Parser

Bottom-Up parser



S → ABC

ABCD