03/03/2025 Quiz4.

so far

1. eliminating left recursion

(a) direct left recursion / immediate.

$A \to A\alpha \mid \beta$ | $A \to \beta B$, $B \to \varepsilon \mid \alpha B$

(b) indirect left recursion

$A \to Bb \mid a$

$B \to Ab$

$A \Rightarrow Bb$

$\Rightarrow Abb$

idea: reduce to direct recursion.

$A \to Bb \mid a$

$B \to Bbb \mid ab$

$A \to Bb \mid a$

$B \to abB'$

$B' \to bbB' \mid \varepsilon$

or. $B \to Ab$

$A \to Abb \mid a$

$B \to Ab$

$A \to aA'$

$A' \to bbA' \mid \varepsilon$

General algorithm

arrange the N_s into some order $A_1, A_2 \cdots A_n$

for $i=1$ to n // consider rules $A_i \to \cdots \mid \cdots \mid \cdots$

for $s=1$ to $i-1$

a) replace each rule $A_i \to A_s \gamma$

with $A_i \to \delta_1\gamma \mid \delta_2\gamma \mid \cdots \mid \delta_k\gamma$

where $A_s \to \delta_1 \mid \delta_2 \cdots \mid \delta_k$

b) eliminate direct left recursion

(assume no $A \to \varepsilon$, no $A \xrightarrow{+} A$) $\times$

$A \to BC$
$B \to A$
$C \to \varepsilon$

ex: $A \to Ba$

$B \to Ab \mid c$

order A, B

$A \to Ba$

$B \to Bab \mid c$

$\Big\}$

$A \to Ba$

$B \to cB'$

$B' \to \varepsilon \mid abB'$

order B, A

$B \to Ab \mid C$

$A \to Ba \Rightarrow A \to Aba \mid ca$

$B \to Ab \mid C$

$A \to caA'$

$A' \to \varepsilon \mid baA'$

ex2. $S \to A$

$A \to Ba \mid a$

$B \to Ab$.

ex3: $A \to Cd$

$B \to Ce$

$C \to A \mid B \mid f$.

A, B, C

$A \to Cd$

$B \to Ce$

$C \to Cd \mid Ce \mid f$

$A \to cd$

$B \to ce$

$C \to fc' \mid c' \to dc' \mid ec' \mid \varepsilon$

A, B

ex4 $A \to Ba \mid Aa \mid C$

$B \to Bb \mid Ab \mid d$

$A \to cA' \mid BaA'$

$A' \to aA' \mid \varepsilon$

$B \to Bb \mid d \mid cA'b \mid BaA'b$

$A \to CA' \mid BaA'$

$A' \to aA' \mid \varepsilon$

$B \to Bb \mid d \mid cA'b \mid BaA'b$

$\Longleftrightarrow$

$B \to Bb \mid BaA'b \mid cA'b \mid d$

$A \to CA' \mid BaA'$

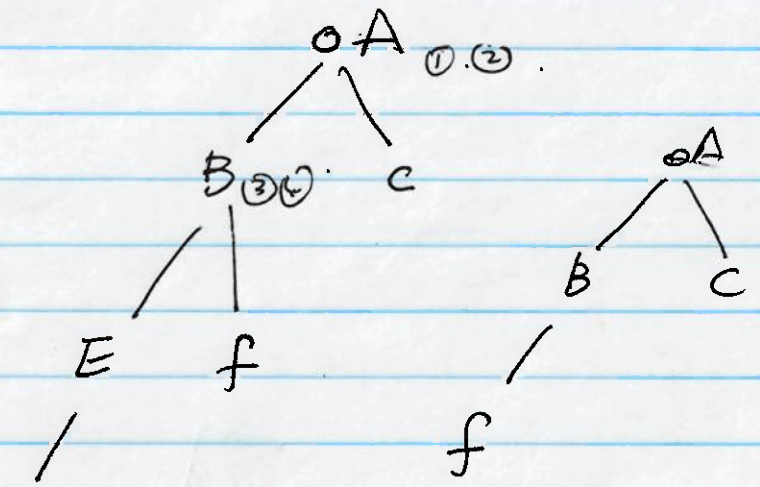
$A' \to aA' \mid \varepsilon$

$B \to dB' \mid cA'bB'$

$B' \to bB' \mid aA'bB' \mid \varepsilon$

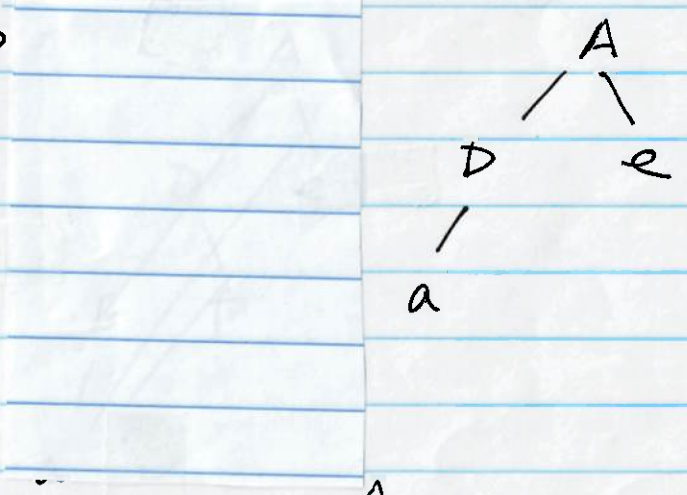
2. pick "right" rule by looking ahead.

$A \to Bc$ ①
$\quad | De$ ②
~~$\quad | Be$~~
$B \to Ef$ ③
$\quad | f$ ④
$D \to a$ ⑤
$E \to b$ ⑥



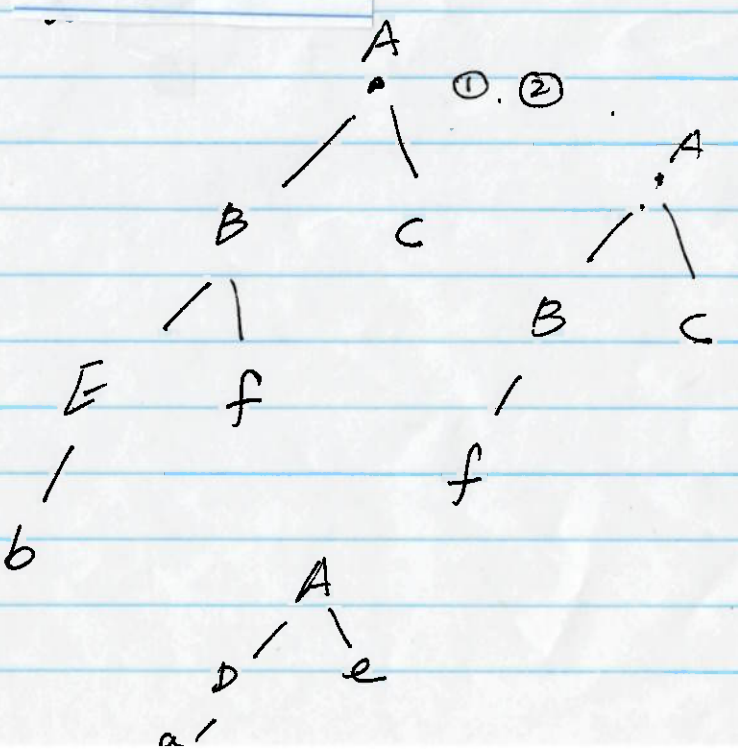
input 1: c

X



input 2. a e

② ✓



back-track free grammar / predictive grammar: can always predict the correct rule with bounded look ahead.

LL(1). back track free.
grammar: look ahead one word / token
can be parsed in linear time

2. first(α) $\alpha \in (N \cup T \cup EOF \cup \varepsilon)^*$

$x \in first(\alpha)$ iff $\alpha \overset{*}{\Longrightarrow} x\gamma$, $x \in \{T, \varepsilon, EOF\}$

* $\alpha \in \{T \cup \varepsilon \cup EOF\}$ $first(\alpha) = \alpha$

* $\alpha \in N$ $\alpha \longrightarrow B_1 B_2 \cdots B_k$

$first(\alpha) = first(B_1 B_2 \cdots B_k)$

$= first(B_1) \cup first(B_2) \cup \cdots first(B_i)$

① where B_i is the first symbol whose first set does not contain ε.

② $\varepsilon \in first(\alpha)$ if and only if $\varepsilon \in first(B_i)$ for every $1 \le i < k$

$S \to ABDC$

$A \to a \mid \varepsilon$

$B \to b \mid \varepsilon$

$D \to d$

first(S) =

$S \overset{*}{\Rightarrow} a \cdots$
$\overset{*}{\Rightarrow} b \cdots$
$\overset{*}{\Rightarrow} d$

$A \to B_1 B_2 \cdots B_k$

$A \to C_1 C_2 \cdots C_m$

// compute. first.

// init.

for each $\alpha \in T \cup \{\varepsilon\} \cup \{EOF\}$.

$first(\alpha) = \alpha$

for each $\alpha \in N$

$first(\alpha) = \{\ \}$

[redacted]

while first sets are still changing do

// iterate all production rules

for each production $A \longrightarrow B_1 B_2 \cdots B_k$ do

firstrhs = first(B_1) − {ε}

trailing = true

// $B_1, B_2 \ldots B_i$ have been processed.

for$_1$ ($i = 1$ to $k-1$)

if ε ∈ first(B_i)

firstrhs = firstrhs ∪ first(B_{i+1}) − {ε}

else

trailing = false

break

//end for$_1$.

if trailing = true and $\varepsilon \in first(B_k)$

    firstrhs = {ε} ∪ firstrhs.

first(A) = first(A) ∪ firstrhs

    // end of for.

// end of while.

ex1. 0 ~Goal → list

1 list → pair list

2 | ε

3 pair → (list)

First sets.

	init	round 1	
Goal	ϕ	ϕ	
list	ϕ	~~ϕ~~	{ε}
pair	{ }	(	
(	(		
)	)		
ε	ε		
EOF	EOF		