Buttom-up- Parssing

example

$S \rightarrow aABe$

$A \rightarrow Abc \mid b$

$B \rightarrow d$

abbcde (A → b)

aAbcde (A → Abc)

aAde (B → d)

aABe (S → aABe)

S → is start symbol so input accept.

example

$S \rightarrow E$

$E \rightarrow E+T \mid E*T \mid T$

$T \rightarrow id$ (*id is any non terminal)

a + b * c (T → id)

T + b * c (E → T)

E + b * c (T → id)

E + T * c (E → E + T) * (Here be careful)

E * c (T → id)

E * T (E → T)

E (S → E)

S → is strand symbol so input accepted

example

$E \rightarrow E+T \mid T$

$T \rightarrow T*F \mid F$

$F \rightarrow (E) \mid id$

id * id (F → id)

F * id (T → F)

T * id (F → id) *(Here be carefull)

T * F (T → T*F)

T (E → T)

E ✓ is start symbol

what about id+id ? is it valid.

try to Solve it yourself.

LL(k) grammer

L L (k)

→ reading (k) input at a time
where k = 1, 2, 3 ---

↳ Leftmost derivation

↳ Reading input string from left to right

- This is used for Top Down Parsing

Example — is the following grammer is LL(1) ?

$S \rightarrow aA \mid bB$

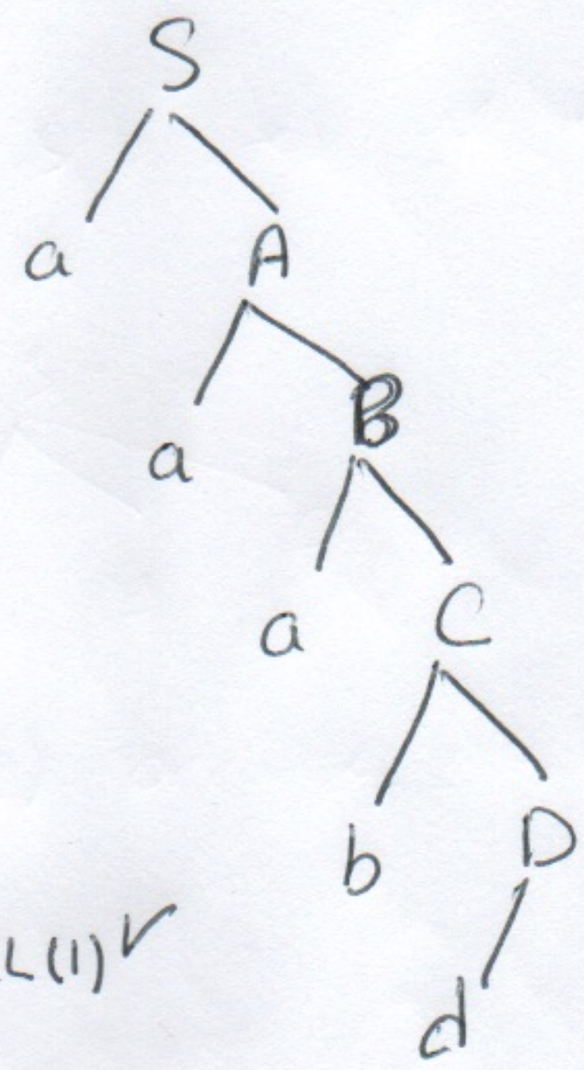
$A \rightarrow aB \mid cB$

$B \rightarrow bC \mid aC$

$C \rightarrow bD$

$D \rightarrow d$

input aaabd



Yes it is LL(1) ✓

[redacted]

is it LL(2) also? Think about it.

TB: agar grammer'ek LL(1) bet awa LL(2) w LL(3) w LL(k+1)'isha balam ba pechawanawa rast nia

LL(k) grammer

example

$S \rightarrow abB \mid aaA$

$B \rightarrow d$

$A \rightarrow c \mid d$

input

abd

- is this grammer LL(1)? lets check

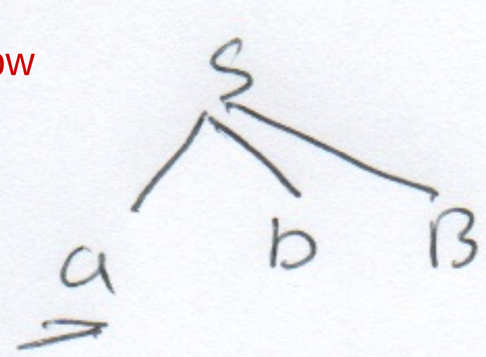
abd read on input at a time

Trying to produce a here we have two possibilites we have

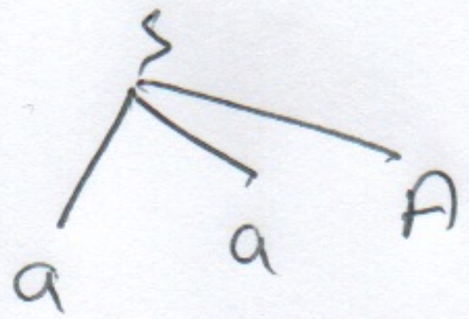
TB: la har hangawekda zyatr la halbzhardnekm habw awa yaksar dalem LL(1) nia,
balam gar hatw la yakam
hangawawa hich regayakm nabw
awa aw grammer'a aw input'a
accept nakat

ex/
input : aaabd

grammer:
S --> cA | bB
A --> d | B
B --> a



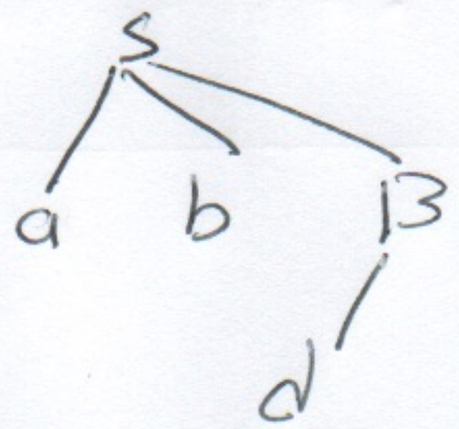
and



so this grammer is not LL(1) because we have more than one choice.

- is it LL(2)? Lets check. (Reading two symbols)

ab d



so it is LL(2)