Pumping lemma for CFL

$\forall L$ CFL $\exists p$ (pumping length)
s.t $\forall w, |w| \geq p$ then
$\exists$ $u,v,x,y,z \in \Sigma^*$
s.t $w = uvxyz$ (1)
$|vxy| \leq p$ (2)
$|vy| \geq 1$ (3)
$uv^ixy^iz \in L \ \forall i \geq 0$ (4)

Ex 2.36 $B = \{a^n b^n c^n \mid n \geq 0\}$ is not CFL

P: suppose it is and let $p = p(L)$ be given from the pumping lemma
we take $w = a^p b^p c^p$
adversary show us $u,v,x,y,z \in \Sigma$ s.t
$w = uvxyz$ and (1), (2), (3), (4) hold.

$aaa \ldots a bb \ldots b c \ldots c$
use that $|vxy| \leq p$ so cannot have both a and c in vxy
If all letters in vxy are the same then $uxz \notin L$ since it has fewer a's if $vxy = a^k$ than b's and c's

$$a\,a \cdots \underbrace{a\,b \cdots}_{vxy} b\,c \cdots c$$

$uxz \notin L$ as we loose either some a's or some b's

Conclusion: adversary cannot find u, v, x, y, z as claimed

So L is not a CFL

Ex 2.37

$C = \{a^i b^j c^k \mid 0 \le i \le j \le k\}$ is not a CFL

use $s = a^p b^p c^p$, where $p = p(L)$ (assuming L was a CFL and using PL)

assume adversary gives us

$s = uvxyz$ s.t. (1)–(4) hold in PL

Observation: neither v nor y can contain two different symbols

$$aa\cdots \underbrace{aabb}_{v}\cdots bcc\cdots c$$

$$uv^2xy^2z \notin L$$

Case 1: no a in vy

$$\underbrace{aa\cdots a}_{u}\underbrace{bb\cdot}_{v}\underbrace{-bc}_{x}\underbrace{\cdots}_{y}\underbrace{c}_{z}$$

$uxz \notin L$

Case 2: no b in vy

$uv^2xy^2z \notin L$

$$\underbrace{aa\cdots a}_{uvxy}\underbrace{bb\cdots bcccc}_{z}$$

~~$uxz \notin L$ if $|y| \ge 1$~~

$$\underbrace{\underbrace{\ }_{u}\underbrace{\ }_{v}\underbrace{\ }_{x}\underbrace{\ }_{y}}_{|vxy| > p \rightarrow \Leftarrow}\underbrace{\ }_{z}$$

otherwise ~~$(|y| = 0)$~~ $|v| \ge 1$

so ~~$uv^2xy^2z \notin L$~~ since it has too many a's

So far we know that vy contains an a and a b

$$\underbrace{aa\cdots a}_{u}\underbrace{\ }_{v}\underbrace{bb\cdots}_{x}\underbrace{b}_{y}\underbrace{cc\cdots c}_{z}$$

$uv^2xy^2z \notin L$ more a's than c's and more b's than c's

Ex 2.38 $\quad D = \{ww \mid w \in \{0,1\}^*\}$

look at $\quad s = \underbrace{0^p 1^p}_{w}\underbrace{0^p 1^p}_{w}$ $\quad$ where $p = p(D)$

suppose there were $u, v, x, y, z \in \Sigma^*$ s.t

$s = uvxyz$ and (1) – (4) from PL hold

then we must have

$$0\cdots0\,1\cdots\underbrace{1\,0\cdots}_{vxy}0\,1\cdots1$$

Since if not then

$$0\cdots\underbrace{0\,1\cdots}_{vxy}1\,0\cdots\underbrace{0\,1\cdots}_{vxy}1$$

So $uv^2xy^2z \notin L$

So we conclude that

$$0\cdots0\,1\cdots\underbrace{1\,0\cdots}_{vxy}0\,1\cdots1$$

but then $uxz = 0^p 1^i 0^j 1^p$ for some i, j where $i + j < 2p$

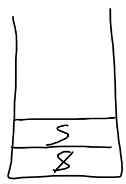
so $uxz \notin L$

Lemma 2.21 $L = L(G)$ for a CFG G
$\Downarrow$ $L = L(M)$ for some PDA M

left derivation in G: always replace leftmost variable

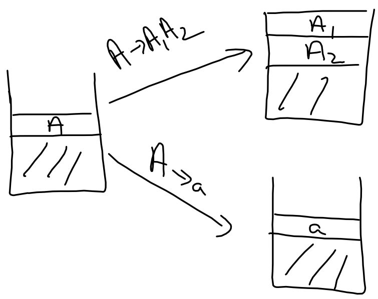
informal proof of L2.21: NB may assume G is chomsky

1. place \$ on stack and
   S on top of this
   (S is start symbol of G)

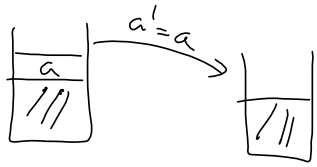


2. Repeat

   a. if top symbol on stack
   is some Variable A
   Select non-deterministically
   a rule $A \to B$ from R
   and replace A on stack by B
   which is either $A_1 A_2 \in V$
   or some $a \in \Sigma$

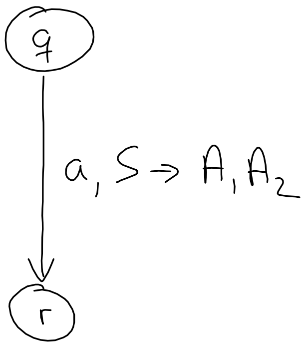


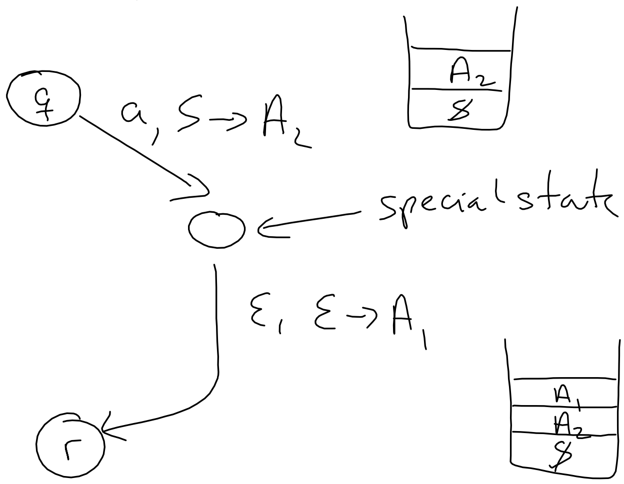
   b. If top of stack is $a \in \Sigma$
   read next input char 'a''
   If $a' = a$ remove a
   from stack. otherwise reject
   this branch of the calculation



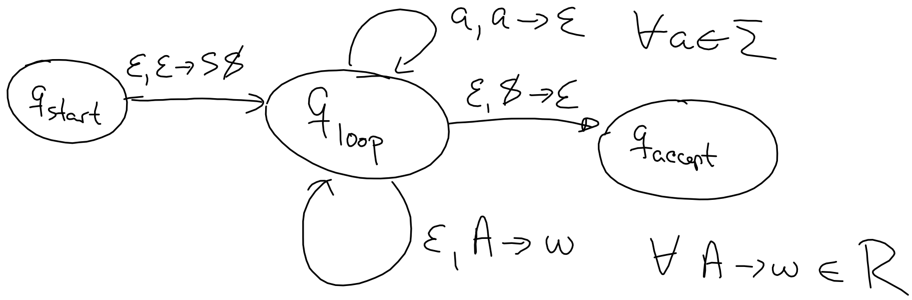
   c. if top symbol is \$
   enter accept state

pushing long strings on stack





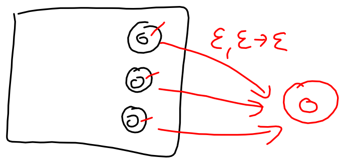
states {q_{start}, q_{loop}, q_{accept}} (+ special states)



Lemma 2.27 If $L = L(M)$ for some PDA M then $L = L(G)$ for some CFG G.

First define a restricted PDA:

1. single accept state



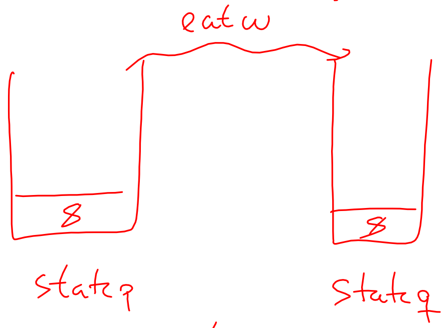
2. always empties stack before stopping.

3. Every transition either pops symbol or pushes a symbol

$$p \xrightarrow{-,\ \alpha\to\beta} q \quad \to \quad p \xrightarrow{-,\ \alpha\to\varepsilon} r_{pq\alpha\beta} \xrightarrow{\varepsilon,\ \varepsilon\to\beta} q$$

$$p \xrightarrow{a,\ \varepsilon\to\varepsilon} q \qquad p \xrightarrow{a,\ \varepsilon\to\alpha} r_{pq\varepsilon} \xrightarrow{\varepsilon,\ \alpha\to\varepsilon} q$$

Note that if M can do



then