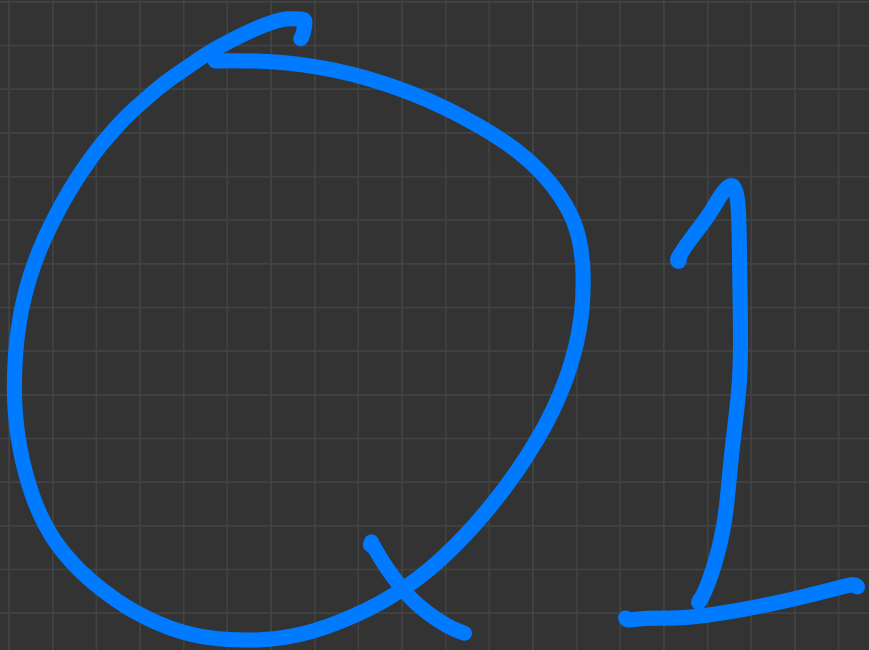


Pumping Lemma (For CFL)

Pumping Lemma (for CFL) is used to prove that a language is NOT context free.

IF A is a context free language, then, A has a pumping length 'p' such that any string 's', where $|s| \geq N$ may be divided into 5 pieces $s = uvxyz$ such that the following conditions must be true.

(1) uv^ixy^iz is in A for every $i \geq 0$
(2) $vz \neq \epsilon$. At least one of the outer symbols cannot be empty
(3) $|vxy| \leq N$

Step 1: Assume L is a context-free language, then $\exists\, G = \{N, T, P, S\}$ such that $L = L(G)$

Step 2: Define variable values in terms of n so it satisfies L

Step 3: So $z = \{...\}$ we have $|z| > \{...\}$ and $z \in L$ then by pumping Lemma for CFL, we have that $z = uvwxy$ and $|vx| \geq 1$ and $uv^iwx^iy \in L \; \forall i \geq 0$

Step 4: Consider every possible way for 's' to be divided into 'uvwxy' Show that none of those combinations satisfy all three pumping conditions above.

Step 5: If none of the above combinations satisfy all three pumping condions, then we have a CONTRADICTION, thus L is NOT context free.

Spring 2008

$L = \{0^k 1^j 2^i \mid i > j > k \geq 0\}$

Assume L is a context free language, then $\exists\ G(N,T,P,S)$ is in chomsky normal form such that $L = L(G)$

The string $S = 0^{2P} 1^{2P+1} 2^{2P+2}$ satisfies L such that P is the pumping length such that $|S| > 2^P$

By pumping lemma $S = uvwxy$ where $|vy| \geq 1$ and $uv^i wx^i y \in L(G)$ for all $i \geq 0$

Case 1: V and X are all zeros =>

Note: When we choose how to partition S into uvwxy, make sure it satisfies $|vwx| \leq P$ and $|vx| \geq 1$

* We can choose any P value as long as $|vwx| \leq P$ and $|vx| \geq 1$

for $i = 0$,

If we choose an arbitrary $P = 2$ then

$S = 0000\,1111\,2222$

u v x y

$uv^0wx^0y = 0\ 0^0\ 0111\ 1\ 0^0\ 2222$

u v x y

order → 0 0 1111 2222

We cannot choose $i = 0$ because this string satisfies L such that $2 > 5 > 6 \geq 0$

for $i = 1$

If we choose an arbitrary $P = 2$ then

$S = 0000\,1111\,2222$

u v x y

$uv^1wx^1y = 0\ 0^1\ 0111\ 1\ 0^2\ 2222$

u v x y

order → 0000 1111 2222

We cannot choose $i = 0$ because this string satisfies L such that $4 > 5 > 6 \geq 0$

for $i = 2$ uv^2wx^2y

If we choose an arbitrary

$S = 0000\,1111\,122222$

$u\,v\,x$ | w | y

$uv^2wx^2y = 0\,0^2\,1\,1\,1\,1\,1\,0^2\,2\,2\,2\,2\,2\,2$

$u\;v$ | w | x | y

$= 0001111100222222$

order it $\rightarrow$ $00000\,11111\,222222$

0^k 1^j 2^i

This does not satisfy L's conditions

$i > j > k \geq 0$ because $5 > 5 > 6 \geq 0$ is false

This means for $i = 2 \in L(G) \notin L$

Case 2: V and X are all ones

for $i=0$, $S = 0000\,11111\,22222$

$uv^0wx^0y =$ We cannot choose $i=0$ because this violates L such that $2 > 5 > 6 > 5$

order $\rightarrow$

Case 3: V & X are all twos $\Rightarrow$ for $i=0 \in L(G), \notin L$

Case 4: No twos in V or X $\Rightarrow$ for $i=2 \in L(G) \notin L$

Case 5: No ones in V or X $\Rightarrow$ for $i=2 \in L(G) \notin L$

Case 6: N zeros in V or X $\Rightarrow$ for $i=0 \in L(G) \notin L$
because # of ones decreases to be # of zeros

Case 7: At least one 0, one 1, one two in V or X
for $i>1$ the pattern will change i.e. zeros can follow twos which is not accepted by L.

Summer 2017

Prove that the following language is context free.

$$L = \{a^i b^j c^i \mid j \geq i \geq 1\}$$

Assume L is a Context Free Language. Then $\exists\ G = (M, T, P, S)$ in CNF such that $L = L(G)$

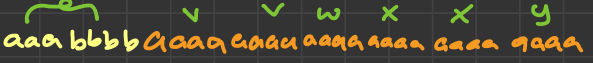
Let $Z \in L(G)$ it can be shown as $Z = uvwxy$ with $|vx| \geq 1$ then by $uv^s wx^s y \in L(G)$ $\forall s \geq 0$

Consider $Z = a^{2^n} b^{2^n} a^{2^n}$

Case 1: V and X are only a's on the right, for $s=2$

$uv^2wx^2y = uvvwxxy$

$a^{2^n} b^{2^n}$ a^{2^n}



uv^2wx^2y is not in $L(G)$

Too many right a's.

Case 2: V and X are only a's on the left, for $s=2$

$uv^2wx^2y = uvvwxxy$

Too many left a's

Case 3: V and X are only b's, for $s=0$

$uv^0wx^0y = uwy$

$a^{2^n} b^{2^n} a^{2^n}$

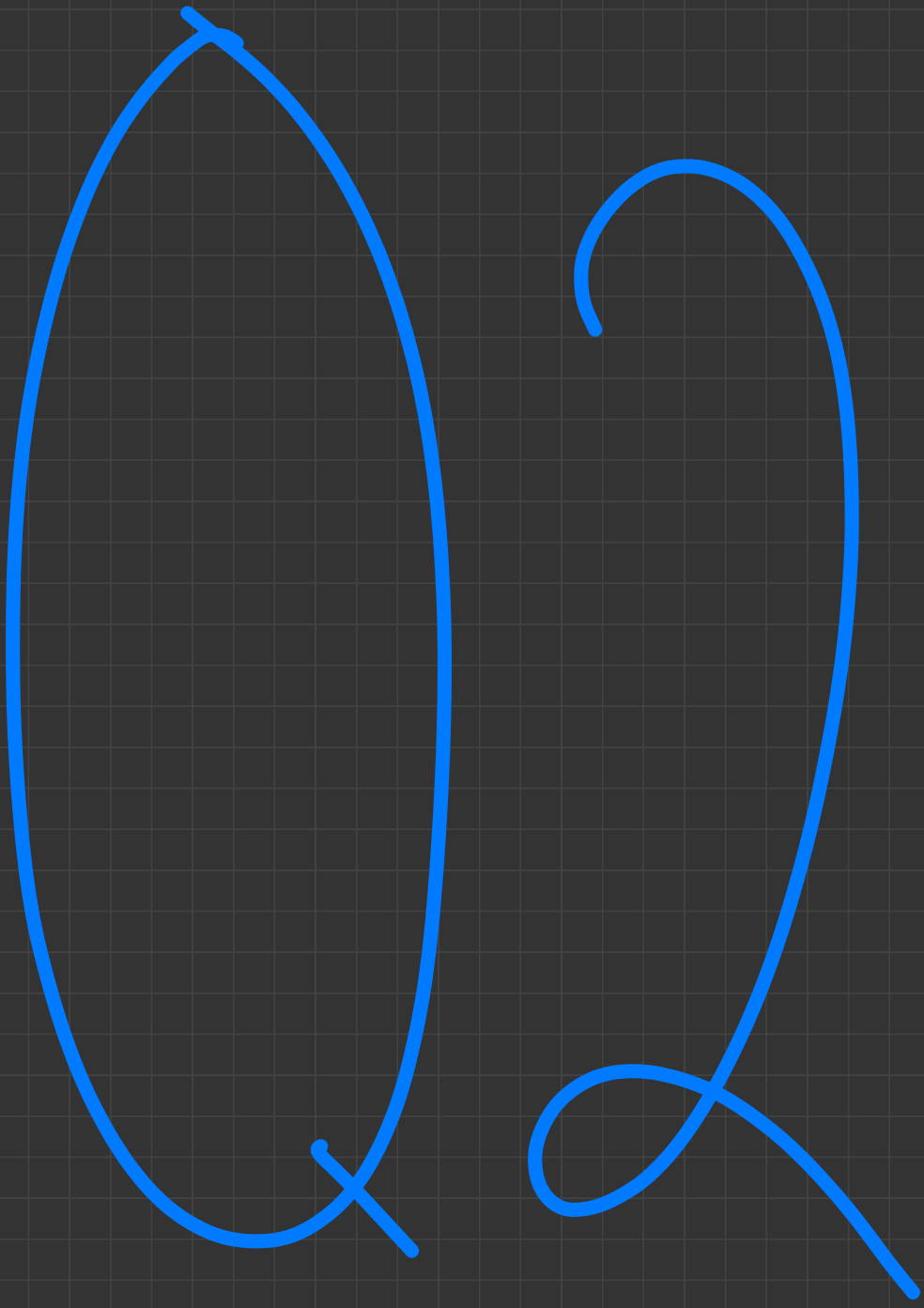
Case 4

Case 5

Case 6

$Z \in L(G)$ but $\notin L$ $\Rightarrow$ Contradiction.

For every case, we found contradictions in which proves that L is not context free.



Anita R Video: PDA Example $a^n b^{2n}$

$\{abb, aabbbb, \ldots\}$

Give a PDA to accept the following language
$L = \{a^n b^{2n} \mid n \geq 1\}$ by final state

δ(current state, current input, top of stack) = (current state, operation representation)

① $\delta(q_0, a, Z_0) = (q_0, aZ_0)$

② $\delta(q_0, a, a) = (q_0, aa)$

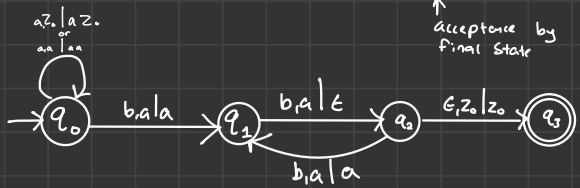
③ $\delta(q_0, b, a) = (q_1, \epsilon)$ — If state changes Read Operation

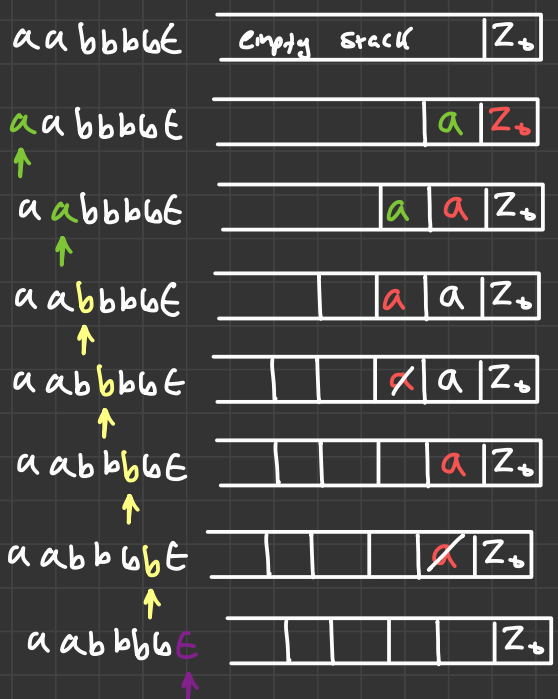
④ $\delta(q_1, b, a) = (q_2, \epsilon)$

⑤ $\delta(q_2, b, a) = (q_1, a)$

Repeat ④

⑥ $\delta(q_2, \epsilon, Z_0) = (q_3, Z_0)$ ← acceptance by final state





Spring 2009

Construct a PDA P for the following language.

$L = \{0^i 1^{4i} \mid i \geq 0\}$ where $L = L_f(P)$
(Acceptance by final state)

State on which side you write the top of stack, Left ⊐ or right ⊏.

Note: Put 4 markers on the stack for every 0

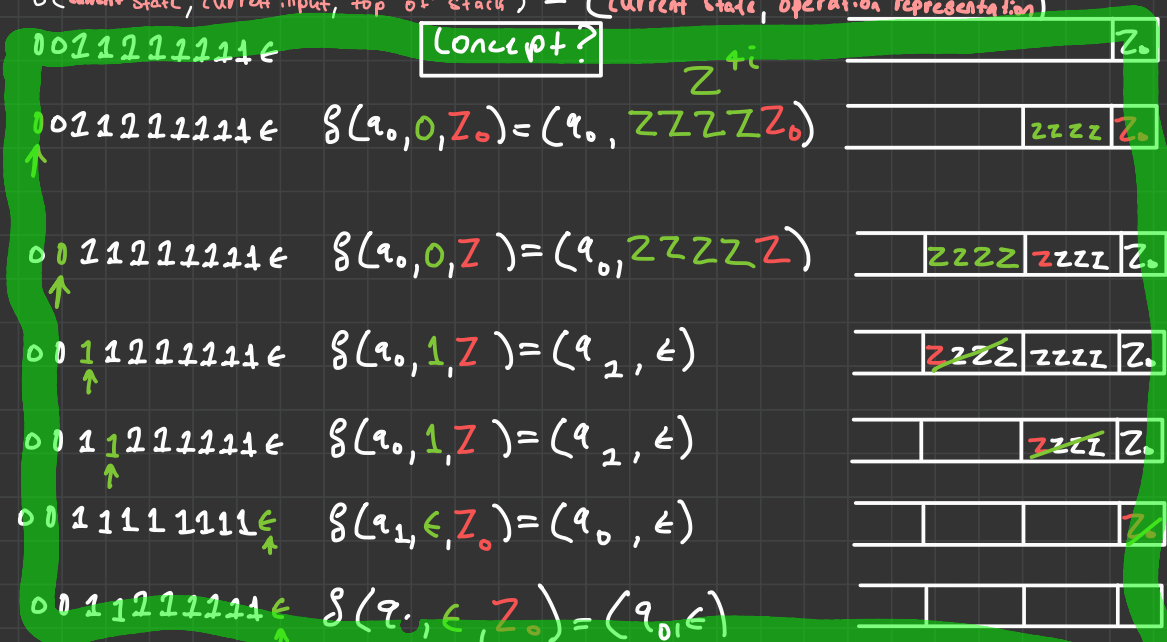
Notations used:

Z_0 → initial stack symbol
q_0 → initial state
q_1 → final state
Z → Stack symbol

Left ⊐ side will be used for the top of the stack

δ(current state, current input, top of stack) = (current state, operation representation)

Concept? Z^{4i}



$P = (\{0,1\}, \{q_0, q_1\}, \{Z_0, Z\}, \delta, q_0, Z_0, \emptyset)$

		0	1	ϵ
q_0	Z_0	$(q_0, ZZZZZ_0)$		(q_1, ϵ)
	Z	$(q_0, ZZZZZ)$	(q_1, ϵ)	
q_1	Z_0			(q_1, ϵ)
	Z		(q_1, ϵ)	

		0	1	ϵ
q_0	Z_0	$\{(q_0, ZZZZZ_0)\}$	/	$\{(q_1, \epsilon)\}$
	Z	$\{(q_0, ZZZZZ)\}$	$\{(q_1, \epsilon)\}$	/
q_1	Z_0	/	/	$\{(q_1, \epsilon)\}$
	Z	/	$\{(q_1, \epsilon)\}$	/

← copied from last Page

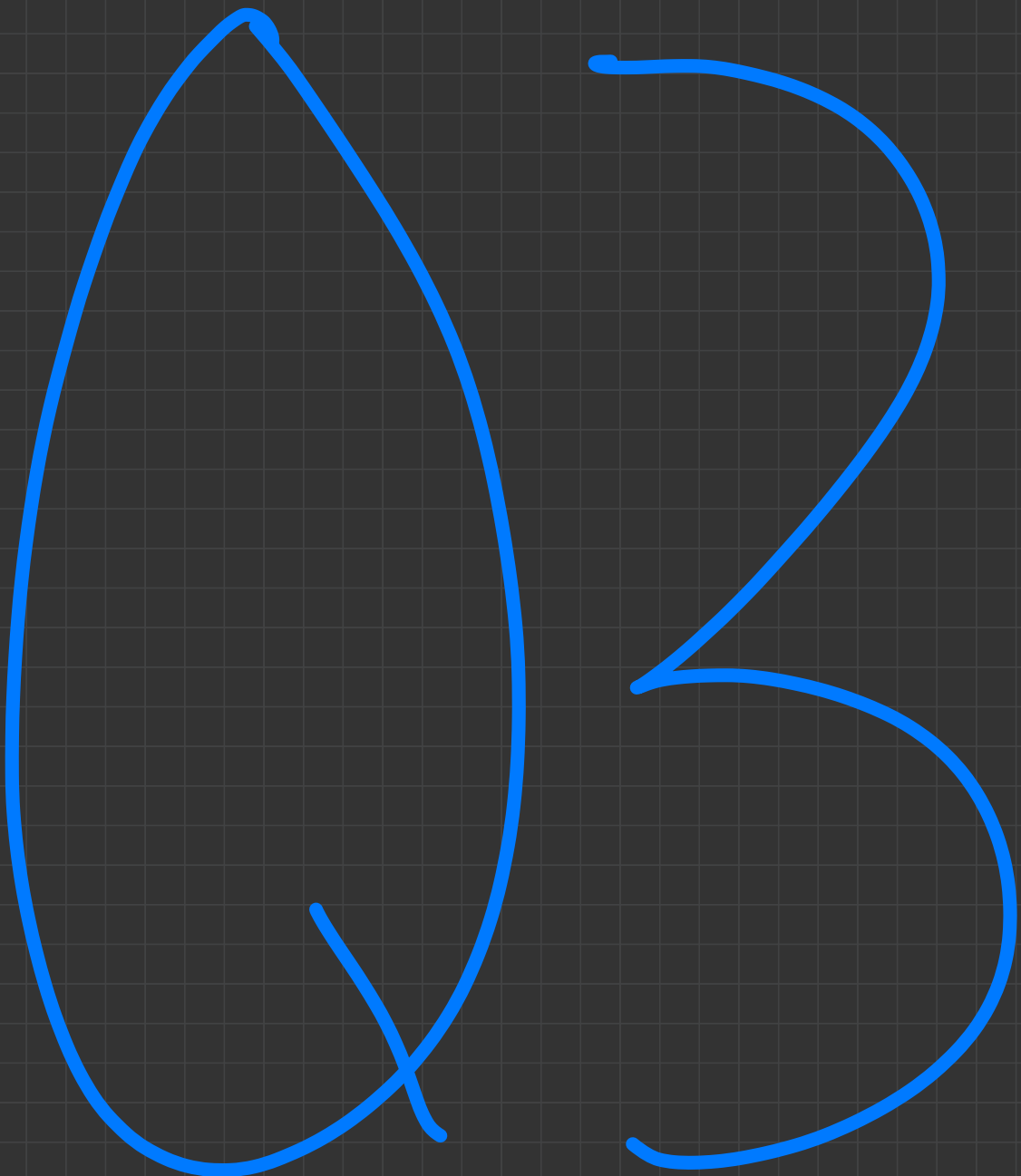
Final State Acception

$P_f = (\{0,1\}, \{q_0', q_0, q_1, q_f\}, (Z_0', Z_0, Z), \delta', q_0', q_f, Z_0')$

δ'

		0	1	ϵ
q_0'	Z_0'	/	/	$\{(q_0, Z_0 Z_0')\}$
	Z_0	/	/	/
	Z	/	/	/
q_0	Z_0'	/	/	/
	Z_0	$\{(q_0, ZZZZZ_0)\}$	/	$\{(q_1, \epsilon)\}$
	Z	$\{(q_0, ZZZZZ)\}$	$\{(q_1, \epsilon)\}$	/
q_1	Z_0'	/	/	$\{(q_f, \epsilon)\}$
	Z_0	/	/	$\{(q_1, \epsilon)\}$
	Z	/	$\{(q_1, \epsilon)\}$	/
q_f	Z_0'	/	/	/
	Z_0	/	/	/
	Z	/	/	/

Final State



Spring 2009

Construct a PDA P that accepts the following language by empty stack.

$L = L(G)$ where $G = (T, N, P, S)$ with $T = \{<, >, [,]\}$, $N = \{S, A\}$ and $P = \{S \to <S>A \mid [A]A,\ A \to [A]S \mid <S>S \mid \epsilon\}$

Starting Productions

$S \to <S>A \mid [A]A$
$A \to [A]S \mid <S>S \mid \epsilon$

Eliminate $A \to \epsilon$. Add an option to replace all 'A' with 'ε'.

$S \to <S>A \mid <S> \mid [A]A \mid [\,]A \mid [A] \mid [\,]$
$A \to [A]S \mid [\,]S \mid <S>S$

Replace closing symbols with S_1, S_2

$S \to <SS_1A \mid <SS_1 \mid [AS_2A \mid [S_2A \mid [AS_2 \mid [S_2$
$A \to [AS_2S \mid [S_2S \mid <SS_1S$
$S_1 \to >$
$S_2 \to]$

Left ⊐ side will be used for top of stack.

Rewriting productions for clarity.

$S \to <SS_1A \mid <SS_1 \mid [AS_2A \mid [S_2A \mid [AS_2 \mid [S_2$
$A \to [AS_2S \mid [S_2S \mid <SS_1S$ $\quad S_1 \to >$ $\quad S_2 \to]$

Traverse each column and add matching productions.

		<	>	[	]	ε
q	S	$\{(q, SS_1A)(q, SS_1)\}$	/	$\{(q, AS_2A)(q, S_2A)$ $(q, AS_2)(q, S_2)\}$	/	/
	A	(q, SS_1S)	/	$\{(q, AS_2S)(q, S_2S)\}$	/	/
	S_1	/	(q, ϵ)	/	/	/
	S_2	/	/	/	(q, ϵ)	/

Summer 2017

Construct a PDA P that accepts the following language by empty stack.

$L = L(G)$ where $G = (T, N, P, E)$ with
$T = \{id, *, /, (,)\}$, $N = \{E\}$
$P = \{E \to E*E \mid E/E \mid (E) \mid id\}$

Starting Productions

$E \to \overbrace{E*E}^{R.} \mid \overbrace{E/E}^{R.} \mid \underbrace{(E)}_{\text{Non Recursive}} \mid \underbrace{id}_{\text{Non Recursive}}$

Recursive

Eliminate left recursion $E \to E\ldots$

$E \to (E) \mid id \mid (E)E' \mid id\,E'$
$E' \to *E \mid /E \mid *EE' \mid /EE'$

Replace) with X_1

$E \to (EX_1 \mid id \mid (EX_1E' \mid id\,E'$
$E' \to *E \mid /E \mid *EE' \mid /EE'$
$X_1 \to)$

Left $\sqsupset$ side will be used for top of stack.
Rewriting for clarity.

$E \to (EX_1 \mid id \mid (EX_1E' \mid id\,E'$
$E' \to *E \mid /E \mid *EE' \mid /EE'$
$X_1 \to)$

Traverse each column and add matching productions.

q	*	/	(	)	id	ϵ
E			(q, EX) (q, EX_1E')		(q, ϵ) (q, E')	
E'	(q, E) (q, EE')	(q, E) (q, EE')				
X_1				(q, ϵ)		

$P = \{\{\epsilon\}, \{*, /, (,), id, \epsilon\}, \{E, E', X_1\}, \delta, q, E, \varnothing\}$

Summer 2022

Starting Productions

Construct a PDA P that accepts the following language by empty stack.

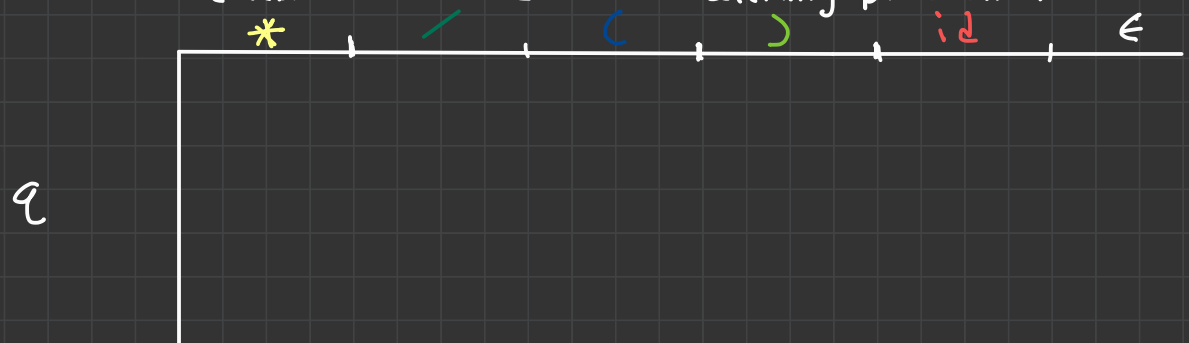
L = L(G) where G = (T, N, P, with
T = {
P = {

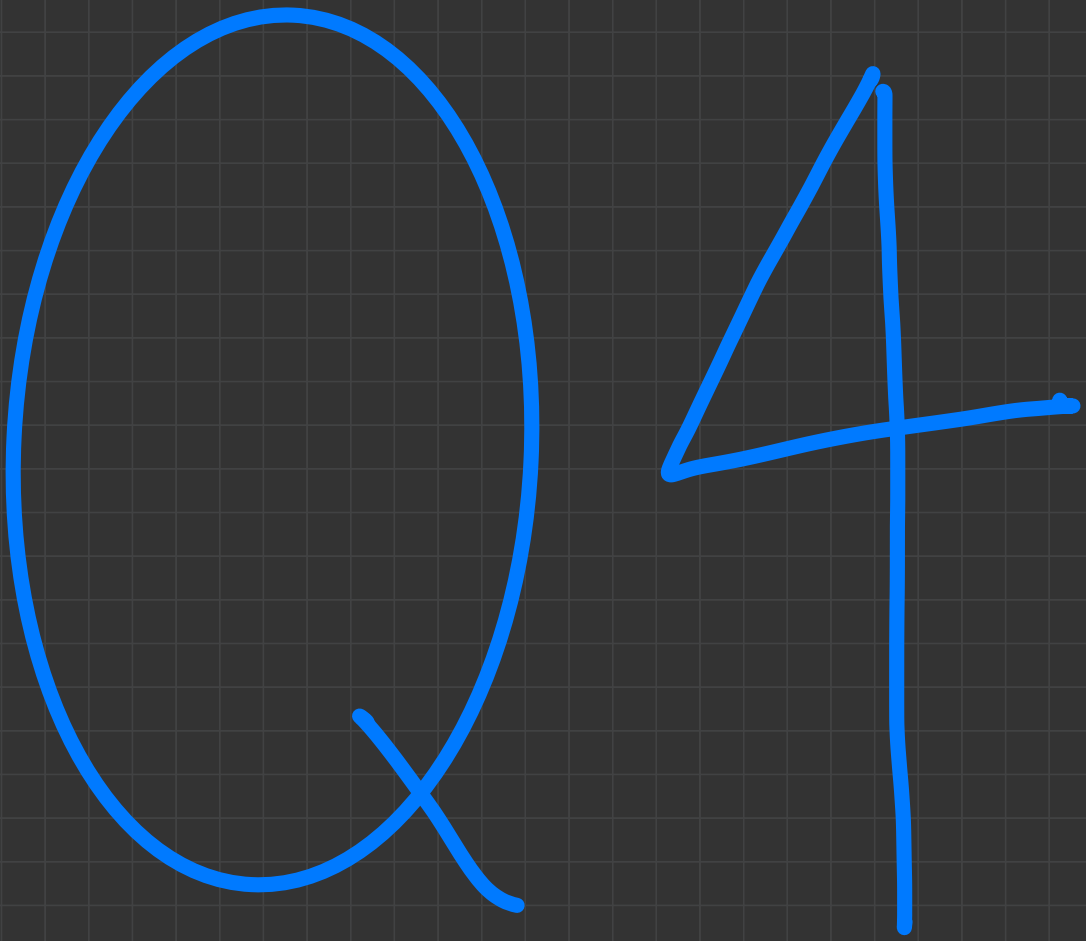
Eliminate left recursion ⇒ F...

Replace with

Left ⊐ side will be used for top of stack.
Rewriting for clarity.

Traverse each column and add matching productions.





Notes on Leiss Lecture

PDA format

$= \{$States, input symbols, stack symbols, move function, initial state, initial stack, final$\}$

$= \{Q, T, P, \delta, q_0, Z_0, \emptyset\}$

$G = (N, T, P, S)$

$$N = \{S\} \cup \{[q, A, p] \mid p, q \in Q, A \in T$$

$$P: 1.\ S \to [q_0, Z_0, q]\ \forall_q \in Q$$

Final States

Free Variable q_0 and q_1

		0	1	ϵ
q_0	Z	(q_0, XZ)	/	/
	X	(q_0, XX)	(q_1, ϵ)	/
q_1	Z	/	/	(q_1, ϵ)
	X	/	(q_1, ϵ)	(q_1, ϵ)

rule 1. $S \to [q_0, Z, \begin{matrix} q_0 \\ q_1 \end{matrix}]$

Spring 2008/2009

Construct a grammar for L(G) for the following language N(P)

$P = (\{p, q\}, \{a, b\}, \{Z, X\}, \delta, p, Z, \emptyset)$

Where the move function δ is given by:

$\delta(p, a, Z) = \{(p, XZ)\}$ $\quad \delta(p, \epsilon, Z) = \{(p, \epsilon)\}$ $\quad \delta(p, a, X) = \{(p, XX)\}$

$\delta(q, a, Z) = \{(q, \epsilon)\}$ $\quad \delta(p, b, X) = \{(q, X)\}$ $\quad \delta(q, b, Z)\} = \{(p, Z)\}$

		a = 1	b = 0	ϵ
p	Z	(p, XZ)		(p, ϵ)
	X	(p, XX)	(q, X)	
q	Z	(q, ϵ)	(p, Z)	
	X			

1) $S \rightarrow [p, \text{initial stack symbol}, \text{all state symbols}]$

$S \rightarrow [p, Z, \begin{matrix} p \\ q \end{matrix}]$

Therefor we have two productions

$S \rightarrow [p, Z, p][p, Z, q]$

2) $(p, XZ) \in \delta(p, a, Z)$

$[p, Z, \begin{smallmatrix} p \\ q \end{smallmatrix}] \to a[p, X, \begin{smallmatrix} p \\ q \end{smallmatrix}][\begin{smallmatrix} p \\ q \end{smallmatrix}, Z, \begin{smallmatrix} p \\ q \end{smallmatrix}]$

We will have 4 productions.

$[p, Z, p] \to a[p, X, p][p, Z, p]\, a[p, X, q][q, Z, p]$

$[p, Z, q] \to a[p, X, p][p, Z, q]\, a[p, X, q][q, Z, q]$

$(p, XX) \in \delta(p, a, X)$ $\quad XX \cdot pq = 4$

$[p, X, \begin{smallmatrix} p \\ q \end{smallmatrix}] \to a[p, X, \begin{smallmatrix} p \\ q \end{smallmatrix}][\begin{smallmatrix} p \\ q \end{smallmatrix}, X, \begin{smallmatrix} p \\ q \end{smallmatrix}]$

We will have 4 productions.

$[p, X, p] \to a[p, X, p][p, X, p]\, a[p, X, q][q, X, p]$

$[p, X, q] \to a[p, X, p][p, X, q]\, a[p, X, q][q, X, q]$

$(q, X) \in \delta(p, b, X)$ $\quad X \cdot pq = 2$

We will have two productions

$[p, X, \begin{smallmatrix} p \\ q \end{smallmatrix}] \to b[q, X, \begin{smallmatrix} p \\ q \end{smallmatrix}]$

$[p, X, p] \to b[q, X, p]$

$[p, X, q] \to b[q, X, q]$

$(p, Z) \in \delta(q, b, Z)$ $\quad Z \cdot qz = 2$

We will have two productions

$[q, Z, \begin{smallmatrix} p \\ q \end{smallmatrix}] \to b[p, Z, \begin{smallmatrix} p \\ q \end{smallmatrix}]$

$[q, Z, p] \to b[p, Z, p]$

$[q, Z, q] \to b[p, Z, q]$

$(q, \epsilon) \in \delta(q, a, Z)$

We will have 1 production

$[q, Z, q] \rightarrow a$

$(p, \epsilon) \in \delta(p, \epsilon, Z)$

We will have one production.

$[p, Z, p] \rightarrow \epsilon$

So total we have 16 productions

Summer 2017

Construct a grammar for $L(G)$ for the language $N(P)$:

$P = (\{p, q\}, \{a, b\}, \{Z, X\}, \delta, p, Z, \emptyset)$

2 $\delta(p, b, Z) = \{(p, XZ)\}$ 3 $\delta(q, \epsilon, Z) = \{(q, \epsilon)\}$ 4 $\delta(p, b, X) = \{(p, XX)\}$

5 $\delta(q, b, Z) = \{(p, XZ)\}$ 6 $\delta(q, a, X) = \{(q, \epsilon)\}$ 7 $\delta(p, a, X) = \{(p, \epsilon)\}$

		a	b	ϵ
p	Z	/	(p, XZ)	/
	X	(p, ϵ)	(p, XX)	/
q	Z	/	$(p, XZ)(q, \epsilon)$	/
	X	(q, ϵ)	/	/

1) $S \to [p, Z, \begin{matrix} p \\ q \end{matrix}]$ Therefor we have $2^{(2^1)}$ productions

$S \to [p, Z, p][p, Z, q]$

2) $(p, XZ) \in \delta(p, b, Z)$ Therefor we have $4^{(2^2)}$ productions

$[p, Z, \begin{matrix} p \\ q \end{matrix}] \to b\,[p, X, \begin{matrix} p \\ q \end{matrix}][\begin{matrix} p \\ q \end{matrix}, Z, \begin{matrix} p \\ q \end{matrix}]$

$[p, Z, p] \to b\,[p, X, p][p, Z, p] \mid b\,[p, X, q][q, Z, p]$

$[p, Z, q] \to b\,[p, X, p][p, Z, q] \mid b\,[p, X, q][q, Z, q]$

3) $(p, XZ) \in \delta(q, b, Z)$ Therefor we have $4^{(2^2)}$ productions.

$[q, Z, \begin{matrix} p \\ q \end{matrix}] \to b\,[p, X, \begin{matrix} p \\ q \end{matrix}][\begin{matrix} p \\ q \end{matrix}, Z, \begin{matrix} p \\ q \end{matrix}]$

$[q, Z, p] \to b\,[p, X, p][p, Z, p] \mid b\,[p, X, q][q, Z, p]$

$[q, Z, q] \to b\,[p, X, p][p, Z, q] \mid b\,[p, X, q][q, Z, q]$

4) $(p, XX) \in \delta(p, b, X)$ (2^1)

$[p, X, \begin{matrix} p \\ q \end{matrix}] \to b\,[p, X, \begin{matrix} p \\ q \end{matrix}]\,[\begin{matrix} p \\ q \end{matrix}, X, \begin{matrix} p \\ q \end{matrix}]$ Therefor we have 4 productions

$[p, X, p] \to b\,[p, X\,p]\,[p, X, p] \mid b\,[p, X, q]\,[q, X, p]$

$[p, X, q] \to b\,[p, X\,p]\,[p, X, q] \mid b\,[p, X, q]\,[q, X, q]$

5) $(p, \epsilon) \in \delta(p, a, X)$ We have one production $[p, X, p] \to a$ 2°

6) $(q, \epsilon) \in \delta(q, \epsilon, Z)$ We have one production $[q, Z, q] \to \epsilon$ 2°

7) $(q, \epsilon) \in \delta(q, a, X)$ We have one production $[q, X, q] \to a$ 2°

We have 15 total productions

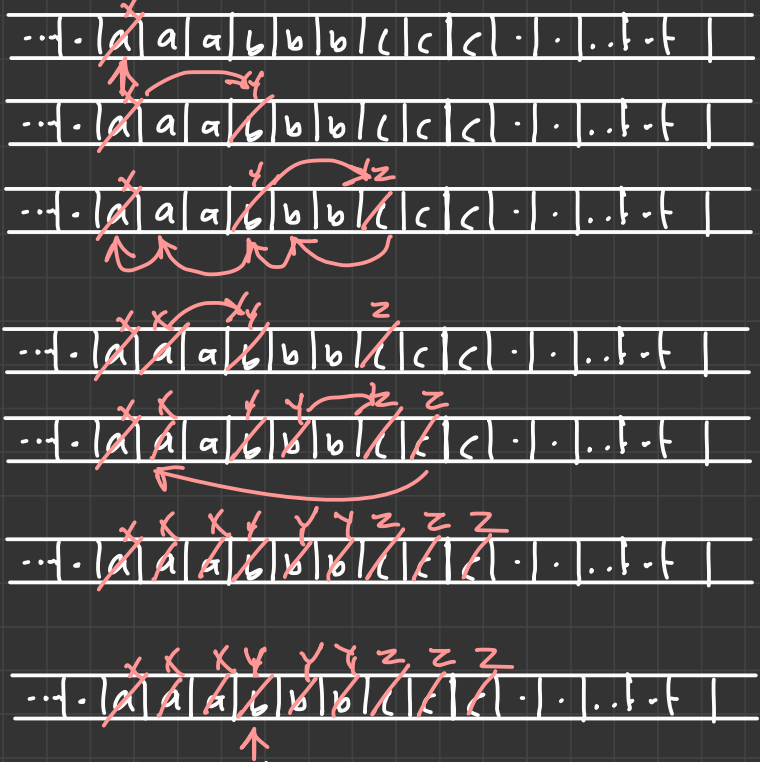
Q6

Construct a turing machine to accept

$L = \{ a^n b^n c^n \mid n \geq 1 \}$

$L = \{ abc, aabbcc, aaabbbccc \}$

Load input string on tape



States	a	b	c	X	Y	Z	∅
q_0	(q_1, X, R)	—	—	—	(q_0, Y, R)	—	—
q_1	(q_1, a, R)	(q_2, Y, R)	—	—	(q_1, Y, R)	—	—
q_2	—	(q_2, b, R)	(q_3, Z, L)	—	—	(q_2, Z, R)	—
q_3	(q_3, a, L)	(q_3, b, L)	—	(q_0, X, L)	(q_3, Y, L)	(q_3, Z, L)	—
q_4	—	—	—	—	(q_4, Y, R)	(q_4, Z, R)	(q_5, B, R)

$TM, M = (\{q_0, q_1, q_2, q_3, q_4, q_5\}, \{a, b, c\}, \{a, b, c, X, Y, Z, \emptyset\}, \delta, q_0, \emptyset, \{q_5\})$

Summer 2017

Construct a Turing Machine for $L = \{ a^i b^j a^i \mid j \geq i \geq 1 \}$
Describe in words what you are doing, then formulate the process.

Let the right a's be renamed to a^+ and $\not b$ represents blank. We know that the number of a's on the left must be equal to the number of a's on the right and the number of b's must be greater than or equal to the number a's on the left or right. The number of a's must be greater than or equal to one.

To formulate the turing machine, start from the left hand side of the tape. Prime a for left a's and continue right. Once a b is reached, prime it. Continue right until $\not b$ is reached. Now start moving left until a^+ is reached and mark it as $\$$. Move left until a' is reached. Now repeat the process above unless a' is in the current state and b' is in the next state. Find and mark the remaining b's. All characters should be marked and moved to final accepting state if the first condition is met.

For $i = j = 1$

Move from left to right while marking left a's and b's. Pass a^+ and reach $\not b$. Turn right and mark a^+ as $\$$ and continue moving until a'. Move left to b' and go to final accepting state.

Each Step of the tape

1) a a b b a a → a' a b b a a
2) a a b b a a → a' a b' b a a
3) a a b b a a → a' a b' b a a
4) a a b b a a → a' a b' b a a
5) a a b b a a → a' a b' b a' a

Marked versions

	a	b	a^+	a'	b'	$	↙
q_0	1 (q_1, a', R)						
q_1	2 (q_1, a, R)	3 (q_2, b', R)					
q_2		4 (q_2, b, R)	5 (q_2, a^+, R)				
q_3							
q_4							
q_5							
q_6							
q_7							
q_8							
q_9							
q_{10}							
q_{11}							
q_{12}							
q_F							

Summer 2022

Construct a Turing Machine for $L = \{a^i b^i a^j \mid j \geq i \geq 1\}$
Describe in words what you are doing, then formulate the process.

Let the right a's be renamed to a^+ and $\not b$ represents blank. We know that the number of a's on the left must be equal to the number of b's. And the number of right a's must be greater than or equal to the number a's on the left and b's.

To formulate the turing machine, start from the left hand side of the tape. Prime a for left a's and continue right. Once a b is reached, prime it. Continue right until $\not b$ is reached. Now start moving left until a^+ is reached and mark it as $\$$. Move left until a' is reached. Now repeat the process above unless a' is in the current state and b' is in the next state. Find and mark the remaining b's. All characters should be marked and moved to final accepting state if the first condition is met.

For $i = j = 1$

Move from left to right while marking left a's and b's. Pass a^+ and reach $\not b$. Turn right and mark a^+ as $\$$ and continue moving until a'. Move left to b' and go to final accepting state.

(state moving to, State changing to, Right / Left)

Marked versions

	a	b	A	B	↙
q_0	(q_2, A, R)	⌜	/	(q_5, B, R)	
q_1	(q_1, a, R)	(q_3, B, R)	/	(q_1, B, R)	
q_2	(q_3, A, L)	(q_2, b, R)	(q_2, A, R)	(q_2, B, R)	
q_3	(q_3, a, L)	(q_3, b, L)	(q_3, A, L)	(q_4, B, L)	
q_4	(q_4, a, L)	(q_4, b, L)	(q_0, A, R)	(q_4, B, L)	
q_5	(q_5, A, R)	/	(T_5, A, R)	(q_5, B, R)	
q_f	/ Accepting			State	