Pushdown Automata
PDA

$$A = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$$

- Q: STATES (FINITE)
- Σ: LANG. ALPHABET (FINITE)
- Γ: STACK SYMBOLS (FINITE
- δ: TRANSITION FUNCTION
- q_0: START STATE
- Z_0: OPTIONAL BOTTOM OF STACK MARKER
- F: OPTIONAL FINAL STATES

TRANSITION FUNCTION

$$\delta : Q \times \Sigma_e \times \Gamma_e \to 2^{Q \times \Gamma^*}$$

CAN EXTEND TO Γ^* CAN LIMIT TO Γ_e BY GROWING STATES

PDA Diagrams

SOME TEXTS USE $a, X / Y$ WHICH IS FINE ALSO

$$(s) \xrightarrow{a, \alpha \to \beta} (p)$$

$$\delta(s, a, \alpha) \supseteq \{(p, \beta)\}$$

PDA LANGUAGES
$\equiv$ CFLs

INSTANTANEOUS DESCRIPTIONS (ID)

$[q, w, \gamma]$ q - CURRENT STATE
w - REMAINING INPUT
γ - STACK CONTENTS
READ LEFT (TOP) TO RIGHT (BOTTOM)

SINGLE STEP

$[q, ax, Z\alpha] \vdash [p, x, \beta\alpha]$ IF $\delta(q, a, Z) \in (p, \beta)$

MULTISTEP $\vdash^*$ REFLEXIVE TRANSITIVE CLOSURE OF $\vdash$

GIVEN $A = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

THERE CAN BE THREE NOTIONS OF ACCEPTANCE

FINAL STATE

$\mathcal{L}(A) = \{w \mid [q_0, w, Z_0] \vdash^* [f, \lambda, \beta]\}, f \in F$

EMPTY STACK

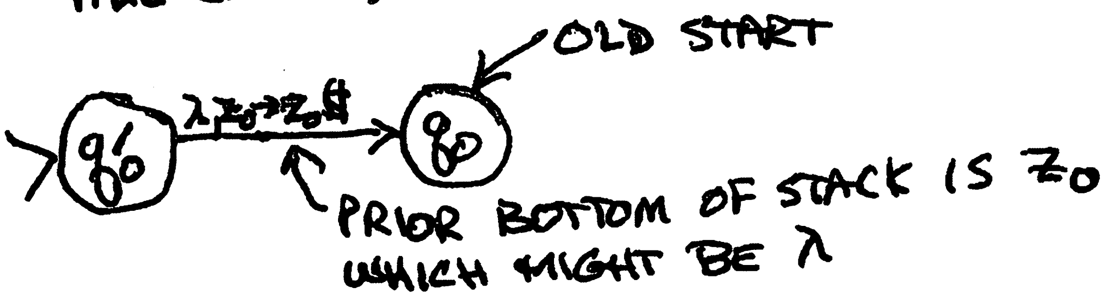
$N(A) = \{w \mid [q_0, w, Z_0] \vdash^* [q, \lambda, \lambda]\}, q \in Q$

EMPTY STACK AND FINAL STATE

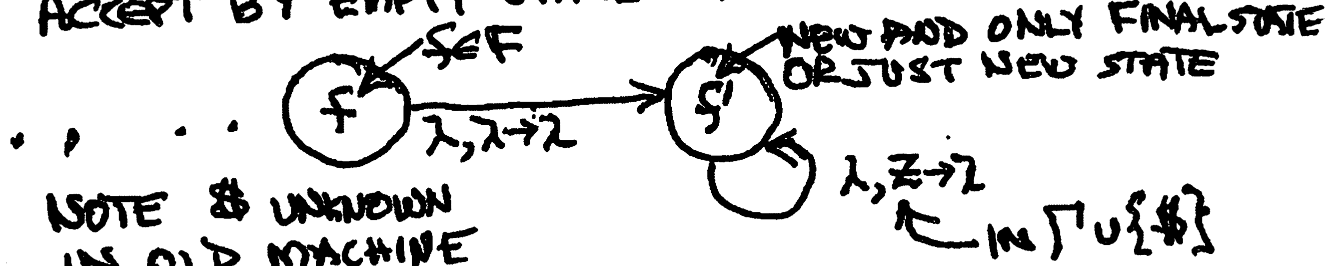
$E(A) = \{w \mid [q_0, w, Z_0] \vdash^* [f, \lambda, \lambda]\}, f \in F$

EQUIVALENCY OF LANGUAGE CLASSES, $\mathcal{L}(A)$, $N(A)$, $E(A)$, WHERE A RANGES OVER ALL PDAs

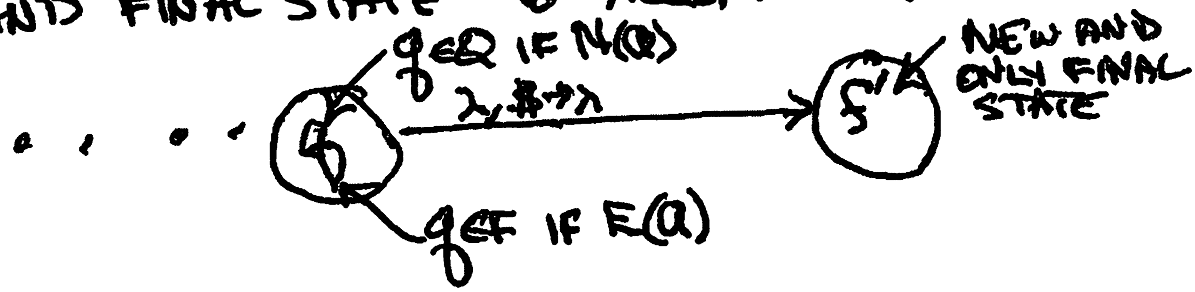
- CONVERTING ONE FORM TO ANOTHER
  FOR EACH CASE, ASSUME q_0' IS A NEW STATE (NEW START) AND \$ IS A NEW STACK SYMBOL PLACED ON BOTTOM OF STACK. FOR ALL CASES, START WITH



- CHANGE ACCEPT BY FINAL STATE TO ACCEPT BY EMPTY STACK OR EMPTY AND FINAL STATE



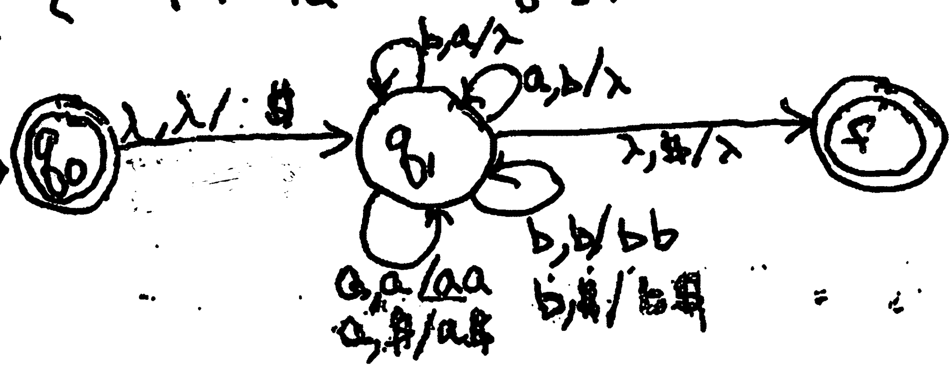
- CHANGE ACCEPT BY EMPTY STACK OR EMPTY AND FINAL STATE TO ACCEPT BY FINAL STATE



EXAMPLE PDA

$L_1 = \{ w \mid |w|_a = |w|_b \}$. ASSUME λ ON STACK

Q_1:



I USED / WHERE BOOK USES →

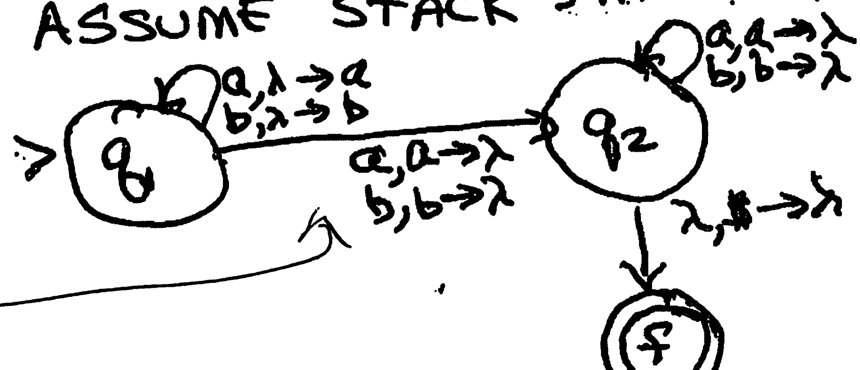
THIS GETS $L_1 = E(Q_1)$

THIS IS NON-DETERMINISTIC

$L_2 = \{ ww^R \mid w \in \{a,b\}^+ \}$

ASSUME STACK STARTS WITH $

Q_2:



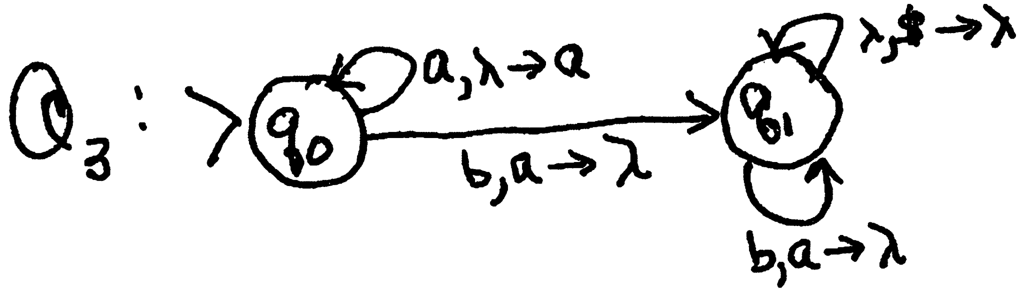
THIS GETS $L_2 = \mathcal{L}(Q_2) = E(Q_2)$

THIS IS ALSO NON-DETERMINISTIC

GUESS MIDPOINT

ANOTHER PDA

$L_3 = \{a^n b^n \mid n > 0\}$. ASSUME \$ ON STACK



THIS GETS $L_3 = N(Q_3)$

THIS IS DETERMINSTIC

DETERMINISM FOR PDAs

1) FOR EACH $q \in Q$ & $Z \in \Gamma$ & $a \in \Sigma$
   IF $|\delta(q, \lambda, Z)| > 0$ THEN $|\delta(q, a, Z)| = 0$

2) FOR NO $q \in Q$, $Z \in \Gamma$ & $a \in \Sigma_\epsilon$
   IS $|\delta(q, a, Z)| > 1$

Bottom Up Parsing by PDA

- Given $G = (V, \Sigma, R, S)$, define
  $A = (\{q,f\}, \Sigma, \Sigma \cup V \cup \{\$\}, \delta, q, \$, \{f\})$
- $\delta(q,a,\lambda) = \{(q,a)\}$ for all $a \in \Sigma$, SHIFT
- $\delta(q,\lambda,\alpha^R) \supseteq \{(q,A)\}$ if $A \rightarrow \alpha \in R$, REDUCE
  Cheat: looking at more than top of stack
- $\delta(q,\lambda,S) \supseteq \{(f,\lambda)\}$
- $\delta(f,\lambda,\$) = \{(f,\lambda)\}$, ACCEPT
- $E(A) = \mathscr{L}(G)$
- Could also do $\delta(q,\lambda,S\$) \supseteq \{(f,\lambda)\}$, $N(A) = \mathscr{L}(G)$

Bottom Up Parsing by PDA

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

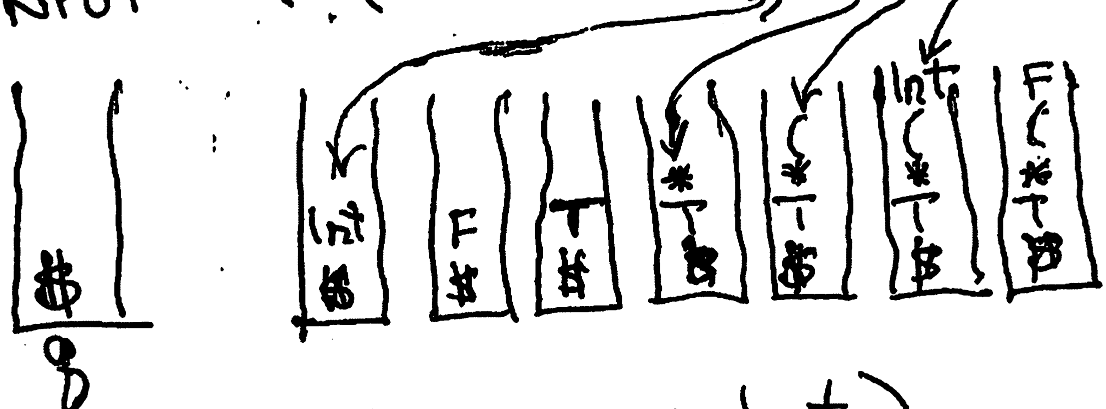
$F \rightarrow (E) \mid Int$

- $\delta(q,+,\lambda)=\{(q,+)\}$, $\delta(q,*,\lambda)=\{(q,*)\}$, $\delta(q,Int,\lambda)=\{(q,Int)\}$, $\delta(q,(,\lambda)=\{(q,()\}$, $\delta(q,),\lambda)=\{(q,))\}$
- $\delta(q,\lambda,T+E) = \{(q,E)\}$, $\delta(q,\lambda,T) \supseteq \{(q,E)\}$ $(E+T)^R$
- $\delta(q,\lambda,F*T) \supseteq \{(q,T)\}$, $\delta(q,\lambda,F) \supseteq \{(q,T)\}$
- $\delta(q,\lambda,)E() \supseteq \{(q,F)\}$, $\delta(q,\lambda,Int) \supseteq \{(q,F)\}$
- $\delta(q,\lambda,E) \supseteq \{(f,\lambda)\}$
- $\delta(f,\lambda,\$) = \{(f,\lambda)\}$
- $E(A) = \mathscr{L}(G)$

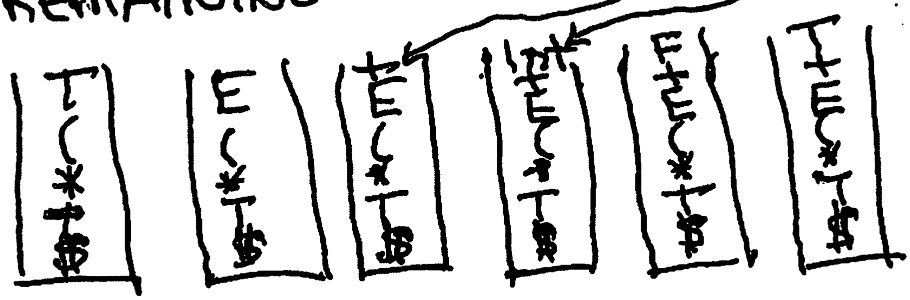
BOTTOM-UP PARSER

$E \to E + T \mid T$
$T \to T * F \mid F$
$F \to (E) \mid \text{Int}$

INPUT: $7 * (3 + 21) \Rightarrow \text{Int} * (\text{Int} + \text{Int})$



REMAINING INPUT + Int)



REMAINING INPUT)

ACCEPTS BY FINAL STATE AND
EMPTY STATE
OR JUST EMPTY STACK (CAN ALTER FOR FINAL)

Top Down Parsing by PDA

- Given $G = (V, \Sigma, R, S)$, define
  $A = (\{q\}, \Sigma, \Sigma \cup V, \delta, q, S, \phi)$
- $\delta(q,a,a) = \{(q,\lambda)\}$ for all $a \in \Sigma$
- $\delta(q,\lambda,A) = \{(q,\alpha) \mid A \to \alpha \in R \text{ (guess)} \}$
- $N(A) = \mathscr{L}(G)$

- Give just one state, this is essentially stateless, except for stack

Top Down Parsing by PDA

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid Int$

- $\delta(q,+,+) = \{(q,\lambda)\}$, $\delta(q,*,*) = \{(q,\lambda)\}$,
- $\delta(q,Int,Int) = \{(q,\lambda)\}$,
- $\delta(q,(,() = \{(q,\lambda)\}$, $\delta(q,),)) = \{(q,\lambda)\}$
- $\delta(q,\lambda,E) = \{(q,E+T), (q,T)\}$
- $\delta(q,\lambda,T) = \{(q,T*F), (q,F)\}$
- $\delta(q,\lambda,F) = \{(q,(E)), (q,Int)\}$

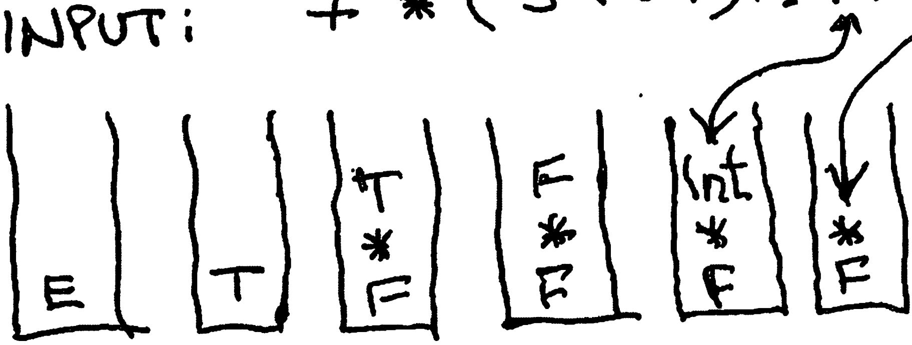
TOP-DOWN PARSER

$E \rightarrow E + T \mid T$

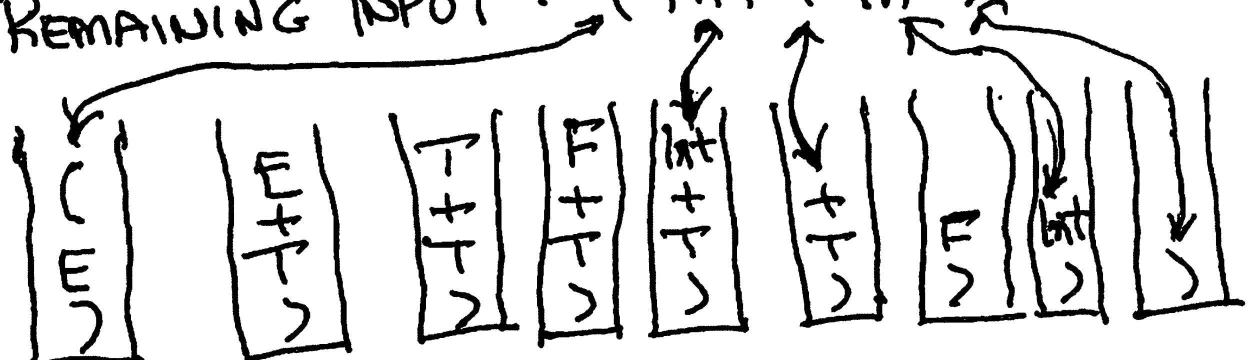
$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid Int$

INPUT: $7 * (3 + 21) \Rightarrow Int * (Int + Int)$



REMAINING INPUT: $(Int + Int)$



ACCEPTS BY EMPTY STACK

LIMITING PDA TO PUSH/POP

PUSH: PUSH (α) IS EQUIVALENT TO

$$\delta(q, a, z) \supseteq \{(p, \alpha z)\}$$

WHERE WE JUST USE

$$\delta(q, a, z) \supseteq \{(p, \text{PUSH}(\alpha))\}$$

POP: POP IS EQUIVALENT TO

$$\delta(q, a, z) \supseteq \{(p, \lambda)\}$$

WHERE WE JUST USE

$$\delta(q, a, z) \supseteq \{(p, \text{POP})\}$$

IF WANT TO SIMULATE STANDARD OPERATION OF

$$\delta(q, a, z) \supseteq \{(p, \alpha)\}$$

CAN DO

$$\delta(q, a, z) \supseteq \{(p', \text{POP})\}$$

$$\delta(p', \lambda, X) \supseteq \{(p, \text{PUSH}(\alpha))\}$$

ANY ELEMENT OF Γ OR λ IF ALLOWED

$$[q_0, \omega, \$] \vdash^* [f, \lambda, \lambda]$$

FOR PDA TO CFG

ASSUME PDA OPERATIONS

POP & PUSH

$$\delta(q, a, Z) \subseteq (p, \mathrm{PUSH}(Y))$$

$$[q, aw, Z\alpha] \vdash [p, w, YZ\alpha]$$

$$\delta(q, a, Z) \subseteq (p, \mathrm{POP})$$

$$[q, aw, Z\alpha] \vdash [p, w, \alpha]$$

PDA $\Rightarrow$ CFG

TECHNIQUE I LIKE

$Q = (Q, \Sigma, \Gamma, \delta, q_0, \$, F)$

$\langle q, Z, p \rangle \stackrel{+}{\Rightarrow} w \quad , w \in \Sigma^*$

IFF $[q, w, Z] \vdash^{+} [p, \lambda, \lambda]$

START WITH

$S \rightarrow \langle q_0, \$, f \rangle \qquad \{f\} = F$

OR

$S \rightarrow \langle q_0, \$, f_1 \rangle \mid \cdots \mid \langle q_0, \$, f_k \rangle$

$\{f_1, \ldots, f_k\} = F$

$w \in E(Q)$

IFF $[q_0, w, \$] \vdash^{*} [f, \lambda, \lambda] \quad f \in F$

IFF $S \stackrel{+}{\Rightarrow} w$

BASICS OF PREFERRED (BY ME) TECHNIQUE

$$\langle q, Z, p\rangle \rightarrow a$$
$$\text{IFF } \delta(q, a, Z) \supseteq \{(p, POP)\}$$

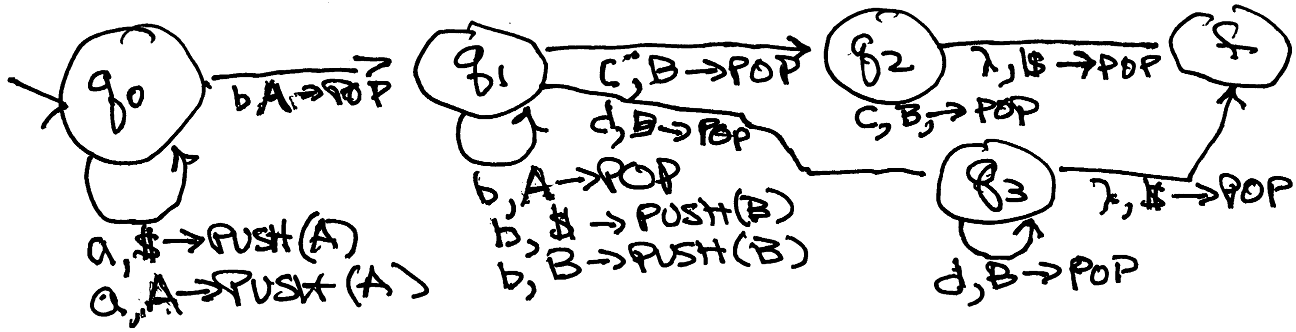
$$\langle q, Z, p\rangle \rightarrow a\langle s, Y, t\rangle\langle t, Z, p\rangle$$
$$\text{IFF } \delta(q, a, Z) \supseteq \{(s, PUSH(Y))\}$$

$$a \in \Sigma_e$$

ONE SHOULD NEVER BOTHER WITH A t
THAT IS CLEARLY INACCESSIBLE FROM S
OR CLEARLY CANNOT POP Y ON PATH
FROM S TO t
OR WITH A t THAT CANNOT GET TO P
OR CLEARLY CANNOT POP X ON PATH
FROM t TO P

$$L = \{a^i b^t c^j \mid t = i+j\} \quad i,j > 0$$
$$\cup \{a^i b^t d^j \mid t = i+j\}$$

$$Q = (\{q_0, q_1, q_2, q_3, f\}, \{a,b,c,d\}, \{A,B,\$\}, \delta, q_0, \$, \{f\})$$



$S \to \langle q_0, \$, f\rangle$

$\langle q_0, \$, f\rangle \to a\langle q_0, A, q_1\rangle\langle q_1, \$, f\rangle$

$\langle q_0, A, q_1\rangle \to a\langle q_0, A q_1\rangle\langle q_1, A q_1\rangle$

$\langle q_0, A, q_1\rangle \to b$

$\langle q_1, A, q_1\rangle \to b$

$\langle q_1, \$, f\rangle \to b\langle q_1, B, q_2\rangle\langle q_2, \$, f\rangle$

$\langle q_1, B, q_2\rangle \to b\langle q_1, B, q_2\rangle\langle q_2, B, q_2\rangle$

$\langle q_1, \$, f\rangle \to b\langle q_1, B, q_3\rangle\langle q_3, \$, f\rangle$

$\langle q_1, B, q_3\rangle \to b\langle q_1, B, q_3\rangle\langle q_3, B, q_3\rangle$

$\langle q_1, B, q_2\rangle \to c \qquad \langle q_1, B, q_3\rangle \to d$

$\langle q_2, B, q_2\rangle \to c \qquad \langle q_3, B, q_3\rangle \to d$

$\langle q_2, \$, f\rangle \to \lambda \qquad \langle q_3, \$, f\rangle \to \lambda$

BOOK'S APPROACH

$$\mathcal{Q} = (Q, \Sigma, \Gamma, \delta, q_0, \lambda, F)$$

$$A_{t;u} \overset{+}{\Rightarrow} w, \quad w \in \Sigma^*$$

IFF $[t, w, \alpha] \overset{*}{\vdash} [u, \lambda, \alpha]$

START WITH

$$S \to A_{q_0, f} \qquad \{f\} = F$$

OR

$$S \to A_{q_0, f_1} \mid \cdots \mid A_{q_0, f_k}$$

$$\{f_1, \ldots, f_k\} = F$$

$$w \in \mathcal{L}(\mathcal{Q}) = E(\mathcal{Q})$$

IFF $[q_0, w, \lambda] \overset{*}{\vdash} [f, \lambda, \lambda] \quad f \in F$

IFF $S \overset{+}{\Rightarrow} w$

BASIC OF BOOK'S TECHNIQUE

$$A_{q,q} \rightarrow \lambda \quad \forall q \in Q$$

$$A_{q,p} \rightarrow A_{q,s} A_{s,p}$$

AND

$$A_{q,p} \rightarrow a\, A_{s,t}\, b$$

WHEN $\delta(p, a, \lambda) \supseteq \{(s, x)\}$
& $\delta(t, b, x) \supseteq \{(p, \lambda)\}$

AS WITH PREFERRED TECHNIQUE,
AVOID IMPOSSIBLE CASES

We assume that initial PDA was normalized to one with a single final state and that each step is either a PUSH (push new symbol on top of current top of stack) or POP (remove current top of stack). For the method from Hopcroft and Ullman, we start with a \$ on bottom of stack and accept by final state and empty stack, For Sipser, we start with empty stack and accept by final state and empty stack. Also, Sipser's method requires read of element from Σ_e, whereas HU allows reading Σ_e*.

Consider the pushdown automaton $A = (\{ q, f \} , \{ 0 , 1 , c \} , \{ 0, 1, c, \$ \}, \delta, q, \$, \{ f \})$, where δ defines transitions:

$\delta(q , 0 , \$) = \{ (q , \mathrm{PUSH}(1)) \}$
$\delta(q , 1 , \$) = \{ (q , \mathrm{PUSH}(0)) \}$
$\delta(q , 0 , 0) = \{ (q , \mathrm{PUSH}(1)) \}$
$\delta(q , 0 , 1) = \{ (q , \mathrm{PUSH}(1)) \}$
$\delta(q , 1 , 0) = \{ (q , \mathrm{PUSH}(0)) \}$
$\delta(q , 1 , 1) = \{ (q , \mathrm{PUSH}(0)) \}$
$\delta(q , c , 0) = \{ (p , \mathrm{PUSH}(c)) \}$
$\delta(q , c , 1) = \{ (p , \mathrm{PUSH}(c)) \}$
$\delta(p , \lambda , c) = \{ (p , \mathrm{POP}) \}$
$\delta(p , 0 , 0) = \{ (p , \mathrm{POP}) \}$
$\delta(p , 1 , 1) = \{ (p , \mathrm{POP}) \}$
$\delta(p , \lambda , \$) = \{ (f , \mathrm{POP}) \}$

This generates the language $E(A) = \{ w\, c\, h(w)^R \mid w \in \{0,1\}^+ \}$ and $h(0)=1$; $h(1)=0$

Write the equivalent grammar using our class's variant of the construction in Hopcroft, Motwani and Ullman.

Hint: the starting non-terminal is: $\langle q , \$, f \rangle$, meaning generate all string that are consumed when we start in **q**, and end up in **f**, having uncovered what's below **\$**.

$\langle q, \$, f\rangle \rightarrow 0 \langle q, 1, q\rangle \langle q, \$, f\rangle$ **X**
$\quad \mid 0 \langle q, 1, p\rangle \langle p, \$, f\rangle$
$\quad \mid 0 \langle q, 1, f\rangle \langle f, \$, f\rangle$ **X**
$\quad \mid 1 \langle q, 0, q\rangle \langle q, \$, f\rangle$ **X**
$\quad \mid 1 \langle q, 0, p\rangle \langle p, \$, f\rangle$
$\quad \mid 1 \langle q, 0, f\rangle \langle f, \$, f\rangle$ **X**
$\langle q, 0, p\rangle \rightarrow 0 \langle q, 1, p\rangle \langle p, 0, p\rangle$
$\quad \mid 1 \langle q, 0, p\rangle \langle p, 0, p\rangle$
$\quad \mid c \langle p, 0, p\rangle$
$\langle q, 1, p\rangle \rightarrow 0 \langle q, 1, p\rangle \langle p, 1, p\rangle$
$\quad \mid 1 \langle q, 0, p\rangle \langle p, 1, p\rangle$
$\quad \mid c \langle p, 1, p\rangle$
$\langle p, 0, p\rangle \rightarrow 0$
$\langle p, 1, p\rangle \rightarrow 1$
$\langle p, \$, f\rangle \rightarrow \lambda$

$\langle q, 0, q\rangle$, $\langle q, 1, q\rangle$, $\langle f, \$, f\rangle$ can lead nowhere as states q and f never entered after popping the stack.

Rewrite as

$S \rightarrow 0\,T \mid 1\,U$
$T \rightarrow 0\,T\,1 \mid 1\,U\,1 \mid c\,1$ **// owe you a 1**
$U \rightarrow 0\,T\,0 \mid 1\,U\,0 \mid c\,0$ **// owe you a 0**

Consider the pushdown automaton $A = (\{q, f\}, \{0, 1, c\}, \{0, 1, c\}, \delta, q, \Phi, \{f\})$, where δ defines transitions:

$$
\begin{aligned}
\delta(q, 0, \lambda) &= \{(q, \mathrm{PUSH}(1))\} \\
\delta(q, 1, \lambda) &= \{(q, \mathrm{PUSH}(0))\} \\
\delta(q, c, \lambda) &= \{(p, \mathrm{PUSH}(c))\} \\
\delta(p, \lambda, c) &= \{(p, \mathrm{POP})\} \\
\delta(p, 0, 0) &= \{(p, \mathrm{POP}), (f, \mathrm{POP})\} \\
\delta(p, 1, 1) &= \{(p, \mathrm{POP}), (f, \mathrm{POP})\}
\end{aligned}
$$

This generates the language $E(A) = \{ w\, c\, h(w)^R \mid w \in \{0,1\}^+ \}$ **and** $h(0)=1$; $h(1)=0$

Write the equivalent grammar using Sipser's construction.

The starting non-terminal is: **Aq,f**, meaning generate all string that are consumed when we start in **q**, and end up in **f** with the stack having the same contents as when we started in **q**. Note that the stack is empty at start and we allow top of stack to be ignored in transitions.

There are two cases for any **At,u**. Either something is pushed on stack and it gets back to its starting point at some intermediate state, **v**, and then back to the start at **u** (**At,u→ At,v Av,u**) or the first transition from **t** involves a **PUSH** and the last to **u** involves a **POP**. In this case, the input is of the form **xwy**, where the **x** is read at the **PUSH** and the y at the **POP**, where $x,y \in \Sigma_e$ and $w \in \Sigma_e^*$

Aq,f → Aq,q Aq,f	**Useless as becomes Aq,f→Aq,f since can never pop and end in q**
| Aq,p Ap,f	**Useless as can never push starting in p**
| Aq,f Af,f	**Useless as becomes Aq,f→Aq,f (see below)** **Note that: Aq,f Af,p is impossible**
| 0 Aq,p 1	**Would have pushed a 1 on stack when in q and must match it in p**
| 1 Aq,p 0	**Would have pushed a 0 on stack when in q and must match it in p**
Aq,p → Aq,q Aq,p	**Useless as becomes Aq,p→Aq,p since can never pop and end in q**
| Aq,p Ap,p	**Useless as becomes Aq,p→Aq,p (see below)**
| 0 Aq,p 1	**Would have pushed a 0 on stack when in q and must match it with 1 in p**
| 1 Aq,p 0	**Would have pushed a 1 on stack when in q and must match it with 0 in p**
| c Ap,p λ	**This reduces to Aq,p→c**
Aq,q → λ	**No other options as never pop and end in q**
Ap,p → λ	**No other options as never p pushes onto stack**
Af,f → λ	**No other options as never p pushes onto stack**

Rewrite as

S →0 T 1 | 1 T 0

T →0 T 1 | 1 T 0 | c

Closure Properties

Context Free Languages

$S \rightarrow S_1 | S_2$

$S_2 \rightarrow AB \mid BA$

$A \rightarrow CAC \mid a$

$C \rightarrow a \mid b$

$B \rightarrow CBC \mid b$

CFL NOT CLOSED UNDER COMPLEMENT

$\{a^n b^n c^m \mid n, m > 0\}$

$\cap \{a^m b^n c^n \mid n, m > 0\}$

$\{a^n b^n c^n\}$ NOT A CFL

NOT CLOSED UNDER $\cap$

MIN + MAX OF CFLs

$L_1 = \{a^i b^j c^k \mid k \le i \text{ OR } k \le j\}$
$L_2 = \{a^i b^j c^k \mid k \ge i \text{ OR } k \ge j\}$

$MAX(L_1) = \{a^i b^j c^k \mid k = MAX(i,j)\}$
$MIN(L_1) = \{\lambda\}$

$MAX(L_2) = \{\ \} = \emptyset$
$MIN(L_2) = \{a^i b^j c^k \mid k = MIN(i,j)\}$

$MAX(L_1)$ IS <u>NOT</u> A CFL
$MIN(L_2)$ IS <u>NOT</u> A CFL

SHOWS CFLs ARE <u>NOT</u> CLOSED UNDER EITHER MAX OR MIN

NOTE: REGULAR ARE CLOSED UNDER MAX AND MIN

CHALLENGING LANGUAGES

$L_1 = \{a^i b^j c^k \mid k = \min(i,j)\}$
$L_2 = \{a^i b^j c^k \mid k = \max(i,j)\}$

CONSIDER L_1 AND LET $N>0$ BE FROM PL
WE MIGHT CHOOSE $a^N b^N c^N$
PL SPLITS IN $uvwxy \ni |vwx| \leq N, |vx| > 0$
AND SAYS $\forall i\ uv^iwx^iy \in L$

WE CHOOSE $i = 0$
CASE 1 IF vx CONTAINS AT LEAST ONE 'a',
WE SEE THAT $uwy \notin L$ AS IT HAS
FEWER 'a''s THAN 'c''s SINCE CONSTRAINT
$|vwx| \leq N$ MEANS vx CANNOT CONTAIN
ANY 'c''s IF IT CONTAINS 'a''s

CASE 2 ASSUME vx CONTAINS NO 'a''s THEN IT
CONTAINS AT LEAST ONE 'b' OR AT LEAST
ONE 'c'. NASTY CASE: IT CONTAINS
THE SAME NUMBER OF b's AS c's
IN WHICH CASE $uwy \in L$

FAIL!!!

WE CONSTRAINED OURSELVES
IF WE REQUIRE SAME i FOR ALL CASES

STATEMENT ABOUT z, $|z| \geq N$,

$$\exists u, v, w, x, y \ \forall i \ P(z, u, v, w, x, y, i)$$

COMPLEMENT IS

$$\forall u, v, w, x, y \ \exists i \ \neg P(z, u, v, w, x, y, i)$$

WE MOVED i TO START, DOING

$$\exists i \ \forall u, v, w, x, y \ \neg P(z, u, v, w, x, y, i)$$

BUT THAT LIMITS US TO CHOOSING i INDEPENDENT OF u, v, w, x, y

P.S. $P(z, u, v, w, x, y, i) = z = uvwxy, \ |vwx| \leq N, \ |y| > 0$
AND $\forall i \geq 0 \ uv^i w x^i y \in L$

REDO OF MIN

CASE 1:

IF vx CONTAINS NO 'c's THEN IT CONTAINS AT LEAST ONE 'b' OR ONE 'a'. CHOOSE $i=0$ AND THEN WE ERASE SOME 'a's AND 'b's (AT LEAST ONE OF ONE OF THESE) BUT NO 'c's, SO WE HAVE MORE 'c's THAN ONE OF 'a's OR 'b's AND SO $uv^0wx^0y \notin L$

CASE 2:

IF vx CONTAINS A 'c' THEN IT CANNOT CONTAIN ANY a's.

CHOOSE $i=2$ AND THEN WE HAVE MORE THAN N 'c's BUT JUST N 'a's SO $k \neq \min(i,j)$ - MORE 'c's THAN 'a's AND THEREFORE $uv^2wx^2y \notin L$

CASES 1&2 COVER ALL POSSIBILITIES SO L IS NOT A CFL BY PUMPING LEMMA

MAX

$$L = \{a^i b^j c^k \mid k = \max(i,j)\}$$

P.L. GIVE ME $N>0$

ME: $z = a^N b^N c^N$

PL: $z = uvwxy$, $|vwx| \leq N$, $|vx|>0$

& $\forall i \geq 0$ $uv^i w x^i y \in L$

ME: ASSUME vwx CONTAINS NO c's

(1) MUST CONTAIN AT LEAST ONE a OR ONE b.

$i=2$ $uv^2wx^2y \notin L$

SINCE THIS WILL ~~HAVE EITHER~~ MORE a's THAN c's OR MORE b's THAN c's

(2) ASSUME vwx CONTAINS AT LEAST ONE c. THEN IT CANNOT CONTAIN ANY a's

$i=0$ REDUCES # OF c's BUT STILL HAVE N a's SO

$$uv^0wx^0y \notin L$$

Intersection with Regular

- CFLs are closed under intersection with Regular sets
  - To show this we use the equivalence of CFGs generative power with the recognition power of PDAs.
  - Let $A_0 = (Q_0, \Sigma, \Gamma, \delta_0, q_0, \$, F_0)$ be an arbitrary PDA
  - Let $A_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ be an arbitrary DFA
  - Define $A_2 = (Q_0 \times Q_1, \Sigma, \Gamma, \delta_2, \langle q_0,q_1\rangle \$, F_0 \times F_1)$ where
    - $\delta_2(\langle q,s\rangle, a, X) \supseteq \{(\langle q',s'\rangle, \alpha)\}$, $a\in\Sigma\cup\{\lambda\}$, $X\in\Gamma$ iff
      $\delta_0(q, a, X) \supseteq \{(q', \alpha)\}$ and
      $\delta_1(s,a) = s'$ (if $a=\lambda$ then $s' = s$).
  - Using the definition of derivations we see that
    $[\langle q_0,q_1\rangle, w, \$] \vdash^* [\langle t,s\rangle, \lambda, \beta]$ in A_2 iff
    $[q_0, w, \$] \vdash^* [t, \lambda, \beta]$ in A_0 and
    $[q_1, w] \vdash^* [s, \lambda]$ in A_1
    But then $w\in\mathcal{F}(A_2)$ iff $t\in F_0$ and $s\in F_1$ iff $w\in\mathcal{F}(A_0)$ and $w\in\mathcal{F}(A_1)$

Substitution

- CFLs are closed under CFL substitution
  - Let $G=(V,\Sigma,R,S)$ be a CFG.
  - Let f be a substitution over Σ such that
    - $f(a) = L_a$ for $a \in \Sigma$
    - $G_a = (V_a,\Sigma_a,R_a,S_a)$ is a CFG that produces L_a.
    - No symbol appears in more than one of V or any V_a
  - Define $G_f = (V \cup_{a\in\Sigma}V_a,\ \cup_{a\in\Sigma}\Sigma_a,\ R' \ \cup_{a\in\Sigma}R_a,\ S)$
    - $R' = \{ A \rightarrow g(\alpha)$ where $A \rightarrow \alpha$ is in $R \}$
    - $g: (V\cup\Sigma)^* \rightarrow (V \cup_{a\in\Sigma}S_a)^*$
    - $g(\lambda) = \lambda$; $g(B) = B$, $B \in V$; $g(a) = S_a$, $a \in \Sigma$
    - $g(\alpha X) = g(\alpha)\, g(X)$, $|\alpha| > 0$, $X \in V\cup\Sigma$
  - Claim, $f(\mathscr{L}(G)) = \mathscr{L}(G_f)$, and so CFLs closed under substitution and homomorphism.

More on Substitution

- Consider G'_f. If we limit derivations to the rules $R' = \{ A \rightarrow g(\alpha)$ where $A \rightarrow \alpha$ is in $R \}$ and consider only sentential forms over the $\cup_{a\in\Sigma} S_a$, then $S \Rightarrow^* S_{a1}\ S_{a2} \ldots S_{an}$ in G' iff $S \Rightarrow^*$ a1 a2 … an iff a1 a2 … an $\in \mathscr{L}(G)$. But, then $w \in \mathscr{L}(G)$ iff $f(w) \in \mathscr{L}(G_f)$ and, thus, $f(\mathscr{L}(G)) = \mathscr{L}(G_f)$.

- Given that CFLs are closed under intersection, substitution, homomorphism and intersection with regular sets, we can recast previous proofs to show that CFLs are closed under
  - Prefix, Suffix, Substring, Quotient with Regular Sets

- Later we will show that CFLs are <u>not</u> closed under Quotient with CFLs.

LOTS OF CLOSURE

SUBSTITUTION & INTERSECTION W/REGULAR

$f(a) = \{a, a'\}$

$g(a) = \{a'\}$

PREFIX

$h(a) = \{a\}$; $h(a') = \{\lambda\}$

INFIX

$L/R = h(f(L) \cap (\Sigma^* \cdot g(R)))$

SUFFIX

$x \cdot y'$ $x \in \Sigma^*$, $y \in R$

QUOTIENT WITH REGULAR

$x \cdot y'$ $xy \in L$, $y \in R$

ETC.

BUT <u>NOT</u>

QUOTIENT WITH CFL