PUMPING LEMMA FOR CFLs

IF L IS A CFL, THEN $\exists N>0$, SUCH THAT IF $z \in L$ AND $|z| \geq N$ THEN $z = uvwxy$, WHERE $|vwx| \leq N$, $|vx|>0$ AND $\forall i \geq 0$ $uv^i w x^i y \in L$

NOTE THAT IF $|\Sigma| = 1$ THEN PUMPING LEMMA FOR CFL DEGENERATES TO PUMPING LEMMA FOR REGULAR

THUS, IF $L \subseteq \Sigma^*$, $|\Sigma|=1$, AND L IS NOT REGULAR THEN L IS NOT A CFL

WEEK #7

$L = \{ww \mid w \in \{a,b\}^*\}$ IS NOT A CFL

1. ASSUME L IS A CFL
2. PL GIVES YOU N
3. YOU CHOOSE $a^N b^N a^N b^N$
4. PL SPLITS $a^N b^N a^N b^N$ INTO $uvwxy$
   WHERE $|vwx| \leq N$, $|vx| > 0$
   AND STATES $\forall i \geq 0 \; uv^i w x^i y \in L$
5. a.) IF vwx IS OVER a'S ONLY
   THEN SINCE $|vwx| \leq N$
   vwx CANNOT SPAN FROM ONE OF
   THE a^N SUBSTRINGS TO OTHER,
   AND SO FOR $i=0$, WE GET EITHER
   $a^{N-|vx|} b^N a^N b^N \in L$ OR $a^N b^N a^{N-|vx|} b^N \in L$
   AND BOTH ARE NOT SO.

   b) IF vwx IS OVER b'S ONLY, THE
   SAME ARGUMENTS APPLY

   c) IF vwx SPANS a'S AND b'S THEN
   THIS MUST BE WITHIN ONE OF
   THE $a^N b^N$ OR $b^N a^N$ SUBSTRINGS
   AS $|vwx| \leq N$. IF $i=0$ THEN WE
   REDUCE SOME a'S AND SOME b'S, BUT NOT ALL,
   WITHIN JUST ONE OF THE CONSECUTIVE
   a'S AND b'S, BUT NOT THE OTHER
   AND, SO THE RESULTING STRING IS NOT
   $\in L$

PUSHDOWN AUTOMATA
PDA

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

Q: STATES (FINITE)
Σ: LANG. ALPHABET (FINITE)
Γ: STACK SYMBOLS (FINITE
q_0: START STATE
Z_0: OPTIONAL BOTTOM OF STACKMARKER
F: OPTIONAL FINAL STATES

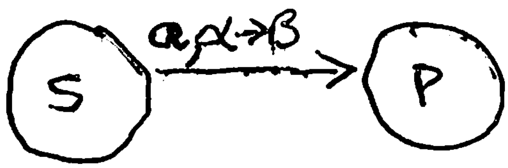
δ: TRANSITION FUNCTION

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

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

PDA DIAGRAMS

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



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

10/13/2016

PDA LANGUAGES
$\equiv$ CFLs

INSTANTANEOUS DESCRIPTIONS (ID)

$[q, w, \gamma]$

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

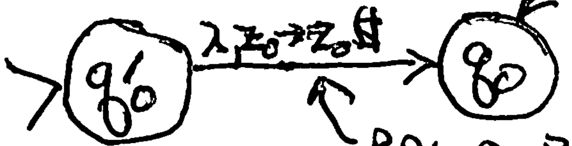
EMPTY STACK AND FINAL STATE
$E(A) = \{w \mid [q_0, w, Z_0] \vdash^* [f, \lambda, \lambda]\}, f \in F$

10/13/2016

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

OLD START

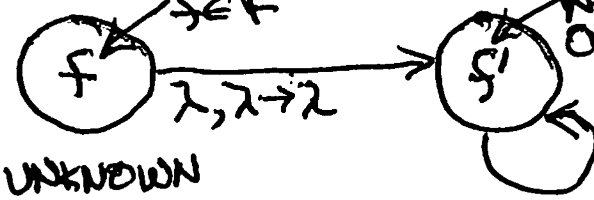


PRIOR BOTTOM OF STACK IS Z_0 WHICH MIGHT BE λ

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

$f \in F$

NEW AND ONLY FINAL STATE OR JUST NEW STATE



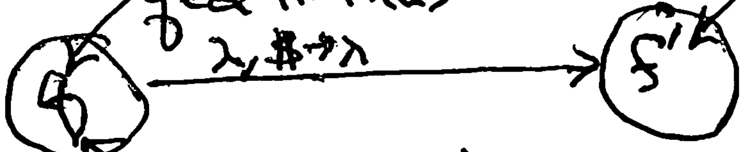
NOTE \$ UNKNOWN IN OLD MACHINE

IN $\Gamma \cup \{\$\}$

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

$q \in Q$ IF $N(Q)$

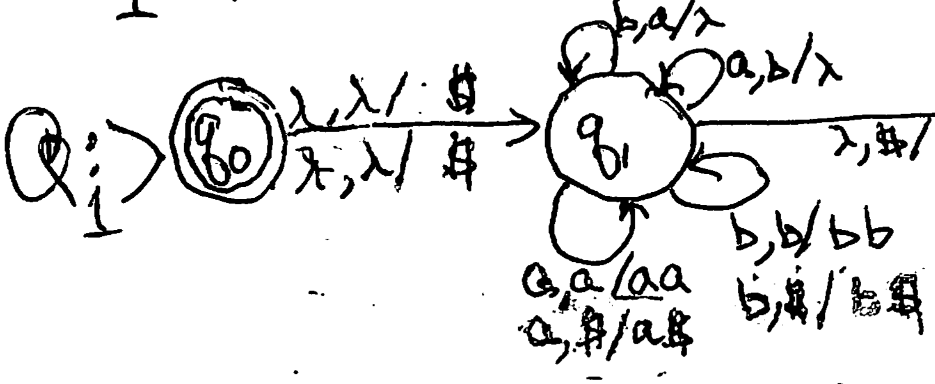
NEW AND ONLY FINAL STATE



$q \in F$ IF $E(Q)$

EXAMPLE PDA

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



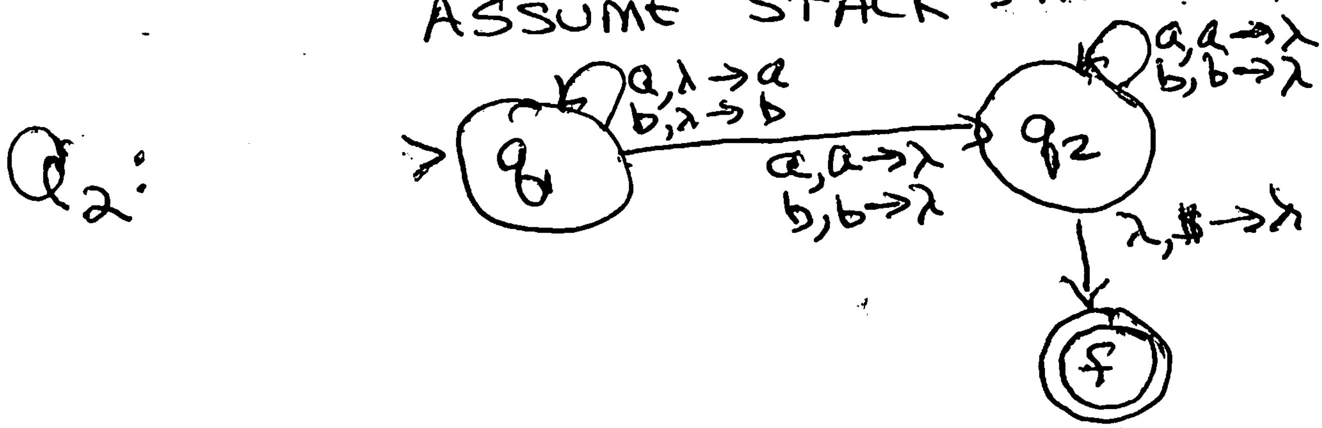
I USED / WHERE BOOK USES $\rightarrow$

THIS GETS $L_1 = E(Q_1)$

THIS IS NON-DETERMINISTIC

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

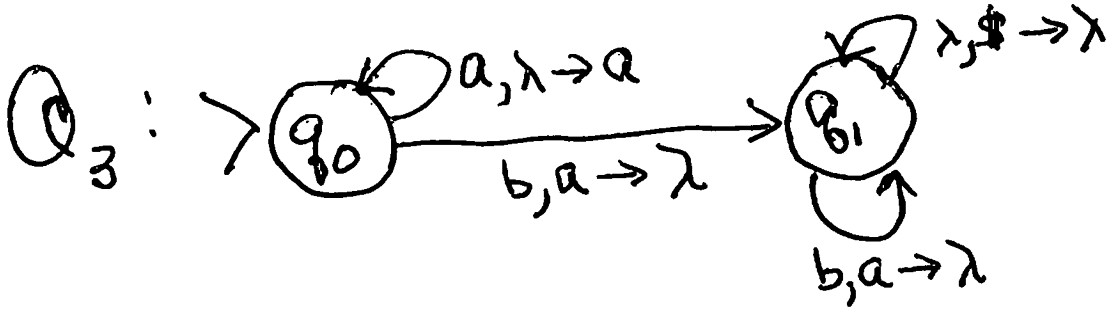
ASSUME STACK STARTS WITH \$



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

THIS IS ALSO NON-DETERMINISTIC

$L_3 = \{a^n b^n \mid n > 0\}$. 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 $|\delta(q, a, z)| > 1$

10/13/2016

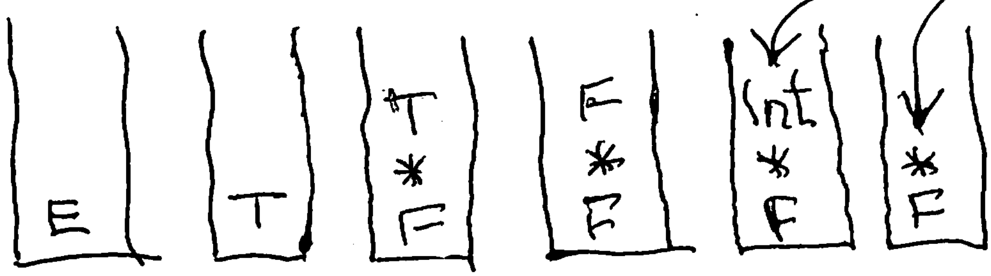
TOP-DOWN PARSER

$E \to E+T \mid T$

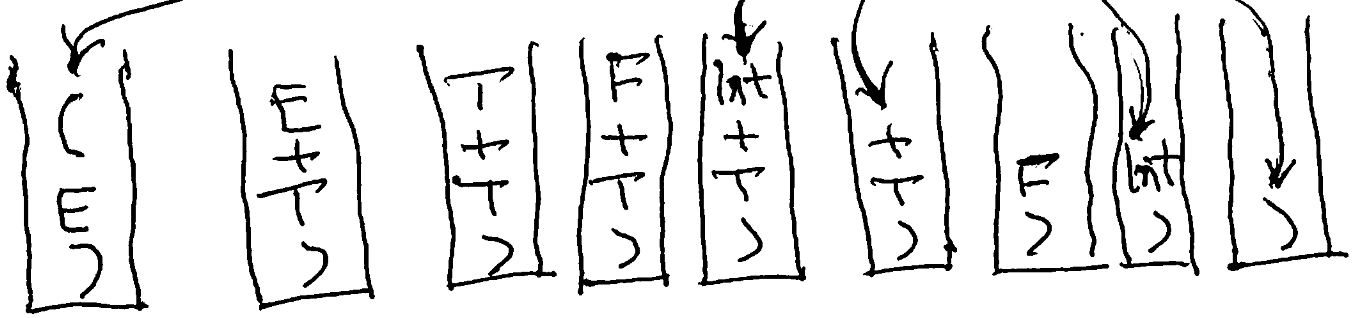
$T \to T*F \mid F$

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

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



REMAINING INPUT: (Int + Int)



ACCEPTS BY EMPTY STACK

10/13/2016

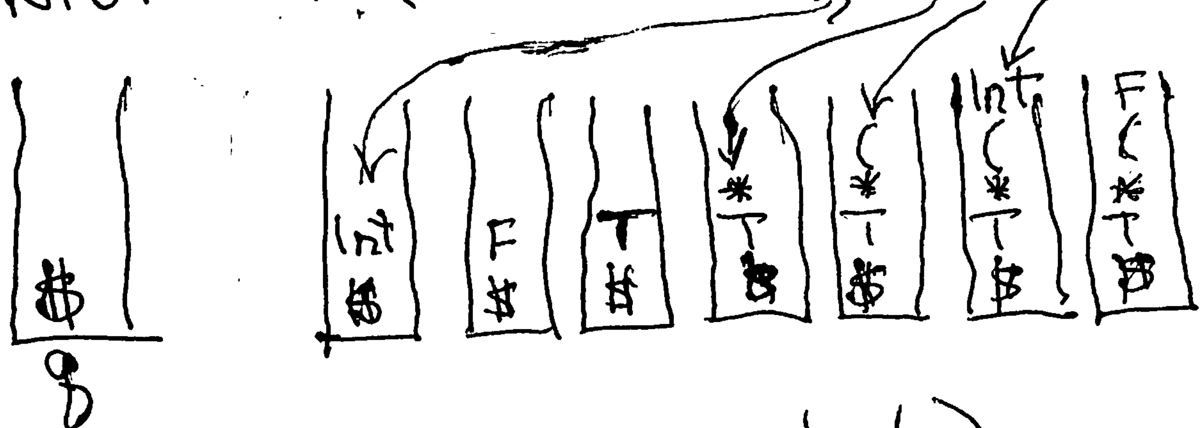
BOTTOM-UP PARSER

$E \to E + T \mid T$

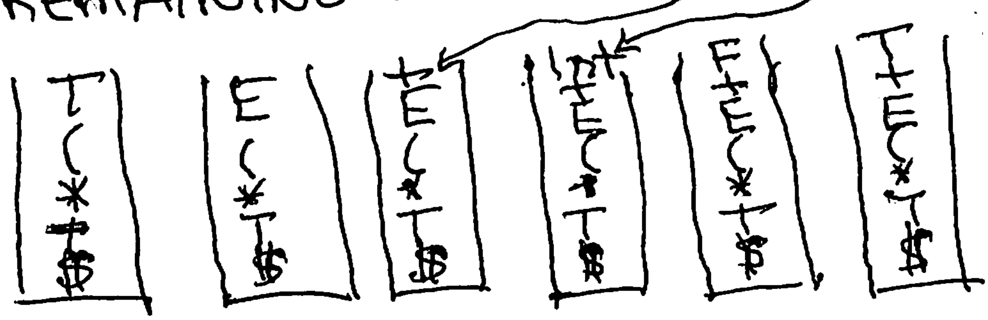
$T \to T * F \mid F$

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

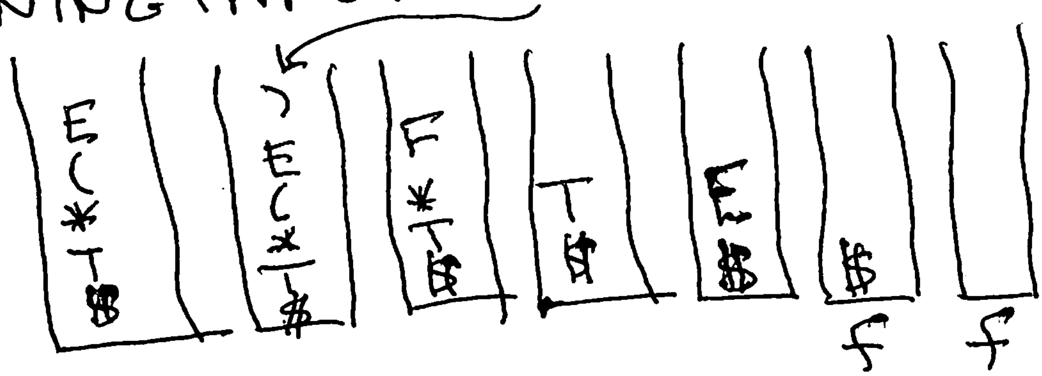
INPUT: $7 * (3+21) \Rightarrow \text{Int} * (\text{Int} + \text{Int})$



REMAINING INPUT $+ \text{Int})$



REMAINING INPUT)



ACCEPTS BY FINAL STATE AND EMPTY STATE

OR JUST EMPTY STACK (CAN ALTER FOR FINAL)