Pushdown Automata

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

- States (Finite)
- Lang. Stack (Finite)
- Alphabet Symbols (Finite)
- Start State
- Optional: Bottom of Stack Marker
- Optional: Final States

Transition Function

\[ \delta : Q \times \Sigma \times \Gamma \rightarrow 2^{Q \times \Gamma^*} \]

CAN EXTEND TO \( \Gamma^* \)

CAN LIMIT TO \( \Gamma^e \) BY GROWING STATES

PDA Diagrams

Some texts use \( q, x / y \) which is fine also

\[ S(s, a, x) = \delta(q, p, \beta) \]

Can be \( ? \)
PDA LANGUAGES
= CFLs

INSTANTANEOUS DESCRIPTIONS (ID)
\[ [q, w, y] \]
  \( q \) - CURRENT STATE
  \( w \) - REMAINING INPUT
  \( y \) - STACK CONTENTS
  (READ LEFT (TOP) TO RIGHT (BOTTOM))

SINGLE STEP
\[ [q, ax, zα] \vdash [p, x, βα] \text{ if } δ(q, a, z) ∈ (p, β) \]

MULTISTEP \( \vdash^* \) REFLEXIVE TRANSITIVE CLOSURE OF \( \vdash \)

GIVEN \( A = (Q, Σ, Γ, δ, q_0, Z_0, F) \)

THERE CAN BE THREE NOTIONS OF ACCEPTANCE

FINAL STATE
\( L(A) = \{ w | [q_0, w, Z_0] \vdash^* [F, ϵ, \epsilon] \} \), \( \epsilon ∈ F \)

EMPTY STACK
\( N(A) = \{ w | [q_0, w, Z_0] \vdash^* [q, ϵ, \epsilon] \} \), \( q ∈ Q \)

EMPTY STACK AND FINAL STATE
\( E(A) = \{ w | [q_0, w, Z_0] \vdash^* [F, ϵ, \epsilon] \} \), \( \epsilon ∈ F \)
EQUIVALENCY OF LANGUAGE CLASSES, \( L(A), N(A), E(A) \), 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:

  - Prior bottom of stack is \( z_0 \) which might be \( \lambda \)

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

  - \( \sigma \), \( \lambda \rightarrow \lambda \)

  - Note \( \# \) unknown in old machine

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

  - \( \sigma \), \( \# \rightarrow \lambda \)

  - \( \sigma \) if \( N(A) \)

  - \( \sigma \) if \( E(A) \)
EXAMPLE PDA

$L_1 = \{ w \mid 1w1_w = 1w1_b \}$, Assume $\lambda$ on stack.

This gets $L_1 = E(Q_1)$

This is non-deterministic

$L_2 = \{ w \# w \# w \# \mid w \in \{a, b\}^* \}$

Assume stack starts with $\#$

This gets $L_2 = L(Q_2) = E(Q_2)$

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 deterministic

---

Determinism for PDAs

1) For each \( q \in Q, z \in \Gamma \) and \( a \in \Sigma \)
   - If \( |\delta(q, a, z)| > 0 \) then \( |\delta(q, a, z)| = 0 \)

2) For no \( q \in Q, z \in \Gamma \) and \( a \in \Sigma \)
   - \( |\delta(q, a, z)| > 1 \)
Top Down Parser

Given $G = (N, \Sigma, R, S)$

Define $A_G = (\Sigma^*, \Sigma, \Sigma^0, S, \emptyset)$

$\delta(q_0, \alpha, \epsilon) = \{ (q_0, \lambda) \} \quad \forall \alpha \in \Sigma$

$\delta(q_0, \lambda, A) = \{ (q_0, \lambda) \mid A \rightarrow \epsilon \in R \}$

Predictive

$N(A) = L(G)$

Note: This has one state so actual state fullness is stack content
Top-Down Parser Example

\[
\begin{align*}
E & \rightarrow E + T \mid T \\
T & \rightarrow T \ast F \mid F \\
F & \rightarrow (E) \mid \text{Int}
\end{align*}
\]

INPUT: \[7 \ast (3 + 21) \Rightarrow \text{Int} \ast (\text{Int} + \text{Int})\]

ACCEPTS BY EMPTY STACK
**Bottom-Up Parser**

Given \( G = (V, \Sigma, R, S) \)

Define \( A_G = (\delta, \epsilon, \Sigma, \delta, \epsilon, \Gamma, \lambda, \delta, \lambda, \delta, \lambda) \)

\[ S(q_0, \alpha, \lambda) = S(q_0, \alpha) \quad \forall \alpha \in \Sigma \text{ Shift} \]

\[ S(q_0, \lambda, \alpha^R) = S(q_0, \alpha) | A \rightarrow \alpha \in R \text{ Reduce} \]

Note: Looking at handle on stacks (more than one symbol is possible)

\[ S(q_0, \lambda, \lambda) \subseteq \delta(S, \lambda) \]

\[ S(S, \lambda, \lambda) = \delta(S, \lambda) \]

\[ E(A) = L(G) \]

Could use

\[ S(q_0, \lambda, S\#) = \delta(q_0, \lambda) \]

\[ NCA(T) = L(G) \]
Bottom-Up Parser Example

E → E + T | T
T → T * F | F
F → (E) | Int

Input: 7 * (3 + 21) ⇒ Int * (Int + Int)

Remaining Input: Int + Int

Remaining Input: T

Accepts by final state and empty stack or just empty stack (can alter for final)
Look at Parsing.pptx
LIMITING PDA TO Push/Pop

**Push:** Push(α) is equivalent to
\[ S(q, a, z) \equiv S(p, xz) \]
where we just use
\[ S(q, a, z) \equiv S(p, \text{Push}(\alpha)) \]

**Pop:** Pop is equivalent to
\[ S(q, a, z) \equiv S(p, \lambda) \]
where we just use
\[ S(q, a, z) \equiv S(p, \text{Pop}) \]

If want to simulate standard operation of
\[ S(q, a, z) \equiv S(p, x) \]

Can do
\[ S(q, a, z) \equiv S(p', \text{Pop} \#) \]
\[ S(p', \lambda, x) \equiv S(p, \text{Push}(\alpha)) \]
\[ \xrightarrow{\text{Any element of } \Sigma} \]

\[ [q_0, \omega, \#] \stackrel{\ast}{\rightarrow} [\varepsilon, \lambda, \lambda] \]
**PDA to CFG**

\[ Q = (Q, \Sigma, \Gamma, \delta, q_0, Z, F) \]

**Technique #1:**

- **Non-Terminals are of form**
  - \( \langle p, z, q \rangle \)
- **Goal is**
  - \( \langle p, z, q \rangle \Rightarrow w \in \Sigma^* \)
  - When \( \Sigma(p, w, z) \Rightarrow \Sigma(q, \lambda, \lambda) \)
- **Start Symbol is**
  - \( S \rightarrow \langle q_0, \#, s \rangle \)

**Can actually do for empty stack only by having**

\( S \rightarrow \langle q_0, \#, \rangle \quad \forall q \in Q \)

**Rules, other than start, are**

- \( \langle q, x, p \rangle \rightarrow \langle s, y, t \rangle \langle t, x, p \rangle \quad \forall t \in Q \)
- **Whenever** \( S(q, q, x) \Rightarrow \{ L(s, \text{push}(y)) \} \)
- \( \langle q_0, x, p \rangle \rightarrow \langle s, y, t \rangle \)
- **Whenever** \( S(q_0, q, x) \Rightarrow \{ L(p, \text{pop}) \} \)

**Goal:**

\( \langle q_0, \#, s \rangle \Rightarrow w, \) whenever \( w \in F(q) \)
CFL CLOSURE

EASY: UNION

\[ G_A = (V_A, \Sigma, R_A, S_A) \]
\[ G_B = (V_B, \Sigma, R_B, S_B) \]
\[ G = (\{ S \} \cup V_A \cup V_B, \Sigma, \{ S \Rightarrow S_A | S_B, R_A \cup R_B \}, S) \]

MODERATE: INTERSECTION WITH REGULAR

\[ Q_0 = (Q_0, \Sigma, \Gamma, q_0, \delta_0, \rho_0, \delta_0^0, \rho_0^0, F_0) \quad \text{PDA, } L = L(Q_0) \]
\[ Q_1 = (Q_1, \Sigma, \delta_1, q_1, F_1) \quad \text{DFA, } L = L(Q_1) \]

DEFINE

\[ Q_2 = (Q_0 \times Q_1, \Sigma, \Gamma, \delta_2, \rho_2, \delta_0 \times \delta_1, \rho_0 \times \rho_1) \]
\[ \delta_2(q_0, s, a, x) = \begin{cases} \delta_1(q_1, s, a, x) & \text{if } a \in \Sigma \\ \delta_0(q_0, s, a, x) & \text{if } a = \lambda \\ \end{cases} \]

NOW INDUCTIVELY CAN SHOW

\[ \left[ q_0, q, w, \# \right] \xrightarrow{\ast} [t, s, \gamma, B] \quad \text{iff} \]
\[ \left[ q_0, q, w, \# \right] \xrightarrow{\ast} [t, \gamma, B] \quad \text{in } Q_0 \]
\[ \text{and} \left\{ s, \gamma \right\} \xrightarrow{\ast} \left\{ s, \gamma \right\} \text{in } Q_1 \]

So we \( F(Q_2) \) iff \( t \in F(Q_0) \land s \in F(Q_1) \)
CFL CLOSURE

MODERATE: SUBSTITUTION

\[ G = (V, \Sigma, R, S), \quad L = \mathcal{L}(G) \]

SUBSTITUTION

\[ S(\alpha) = L_\alpha, \quad \alpha \in \Sigma \]

\[ G_\alpha = (V_\alpha, \Sigma, R_\alpha, S_\alpha), \quad L_\alpha = \mathcal{L}(G_\alpha) \]

IN \( R \), CHANGE ALL INSTANCES OF \( \alpha \in \Sigma \) IN RHS's TO \( S_\alpha \)

Thus, if originally,

\[ S \Rightarrow \alpha_1, \ldots, \alpha_k \]

THEN, IN NEW

\[ S \Rightarrow S_\alpha_1, \ldots, S_\alpha_k \]

AND THEN

\[ S \Rightarrow \alpha_1, \ldots, \alpha_k \]

WHERE \( \alpha_1 \in S(G_\alpha) \)

\[ G' = (V_\alpha U V_\beta, \Sigma, R', S) \]

\[ R' = R \text{CHANGED} U R_\alpha \quad \alpha \in \Sigma \]

WHERE \( R \text{CHANGED} \) IS AS ABOVE
CFL Non-Closure

Intersection

By Example of Contradictory Case

\[ L_1 = \{a^n b^n c^n | n, m > 0 \} \]
\[ L_2 = \{a^m b^n c^n | n, m > 0 \} \]

\[ S_1 \rightarrow S_1 c | T_1 c \quad S_2 \rightarrow aS_2 1aT_2 \]
\[ T_1 \rightarrow aT_1 b | \lambda a b \quad T_2 \rightarrow bT_2 c \lambda b c \]

Both are CFLs

However,
\[ L_1 \cap L_2 = \{a^n b^n c^n | n > 0 \} \]

Which is not a CFL

Complement

By fact that closure under union and complement implies closure under intersection

\[ \sim (\sim A \cup \sim B) = \sim A \cap \sim B \]
CSG

\[ \alpha \rightarrow \beta \quad |\alpha| \leq |\beta|, \]
\[ \delta \in (\nu \nu \varepsilon)^* \nu (\nu \varepsilon)^* \]
\[ \beta \in (\nu \varepsilon)^+ \]

One exception is

\[ S \rightarrow \chi \quad \text{if} \quad \chi \in L \]

and then \( S \) cannot be on RHS

\[
L = \{ a^n b^n c^n \mid n > 0 \}
\]

\[ G = (\Sigma, \Gamma, \delta, \\Sigma_0, \Gamma, \delta_0, A) \]

\[ A \rightarrow aBb \quad \text{or} \quad \text{abc} \]

\[ B \rightarrow aBbC \quad \text{or} \quad \text{abc} \]

\[ C_b \rightarrow BC \]

\[ C_c \rightarrow CC \]

---

\[ L = \{ \omega \mid \omega \in \{0, 1\}^+ \} \]

\[ S \rightarrow 001111 \quad \text{or} \quad \text{abc} \]

\[ R \rightarrow OA2 \quad \text{or} \quad 1AX \]

\[ Z0 \rightarrow OZ \]

\[ X1 \rightarrow 1Z \]

\[ XD \rightarrow 0X \]

\[ XI \rightarrow X \]

\begin{align*}
\text{SHUTTLE} & \\
\text{L \rightarrow 0} & \\
\text{L \rightarrow 1} & \\
\text{L \rightarrow 0} &
\end{align*}
LBA

Simple View
R/W Tape

$0 \leq w \leq 8$

Accept by final state

Actions are
Read, Write, Move (Left/Right/Stay)

Often easiest to view
Operations as being able
to "look left or right" (basically
moves either way)

Can also view as multi-track
(Finite # of tracks)
For example \((\text{\$0, \$1, \$2}) \times (\text{\$0, \$1, \$2})\)
As tape alphabet with channel 1 having input,
Quick Example of LBA

$L = \{a^n b^n c^n \mid n \geq 0\}$

$\begin{align*}
q_0 \$ & \rightarrow \$ \# \\
q_0 a & \rightarrow x q_0 \\
q_0 b & \rightarrow y q_0 \\
q_0 c & \rightarrow z q_0 \\
q_5 y & \rightarrow y q_0 \\
q_3 z & \rightarrow z q_3 \\
q_3 y & \rightarrow y q_0 \\
q_4 y & \rightarrow y q_0 \\
q_4 x & \rightarrow x q_0 \\
q_5 z & \rightarrow z q_3 \\
q_5 y & \rightarrow y q_0 \\
q_3 z & \rightarrow z q_3 \\
q_3 y & \rightarrow y q_0 \\
q_4 y & \rightarrow y q_0 \\
q_4 x & \rightarrow x q_0 \\
q_5 z & \rightarrow z q_3 \\
\end{align*}$

Tape Alphabet is

$\{\$, \#, \$, x, y, z, a, b, c\}$
All RE sets are homomorphic images of CSLs

Let $G = (V, \Sigma, R, S)$, construct $G' = (V, \Sigma, D^2, \Sigma B^2 \{ \star \}, R', S')$

For each rule $\alpha \rightarrow \beta \in R$

Where $|\alpha| \leq |\beta|$

$\alpha \rightarrow \beta \in R'$

For each rule $\alpha \rightarrow \beta \in R$

Where $|\alpha| > |\beta|$, let $d = |\alpha| - |\beta|$

$\alpha \rightarrow \beta \in D^d \in R'$

Also $R'$ contains $Dx \rightarrow xD$ for all $x \in \Sigma U V$

$D\star \rightarrow \star\star$

$S' \rightarrow S*$

$L(G') = \{ w \in \{ \star\}^+ \text{ where } |w| \text{ mod } k \neq 0 \text{ but } k \text{ unknown} \}$

$h(L(G')) = L(G)$; $h(\alpha) = \delta$, $\alpha \in \Sigma$

$h(\star) = \lambda$