CSCE 355
3/24/2025

CFG $\longrightarrow$ PDA in general

1

Definition: Let $G = \langle \underbrace{V}_{\text{nonterminals}}, \overbrace{\Sigma}^{\text{terminals}}, \underbrace{S}_{\text{start symbol}}, \overbrace{P}^{\text{productions}} \rangle$ be a CFG

Define a PDA $M := \langle \underbrace{\{q\}}_{\text{state set}}, \overbrace{\Sigma}^{\text{input alphabet}}, \underbrace{\Sigma \cup V}_{\text{stack alphabet (grammar symbols)}}, \underbrace{\delta}_{\text{transition fcn}}, \overbrace{q}^{\text{start state}}, \overbrace{S}^{\text{bottom stack marker}}, \underbrace{\emptyset}_{\text{accepting state set}} \rangle$

where $\forall a \in \Sigma$, $\delta(q, a, a) = \{(q, \varepsilon)\}$ (matching transition)

~~$\forall$ production $A \to \alpha$~~

and $\forall A \in V$, $\delta(q, \varepsilon, A) = \{(q, \alpha) : (A \to \alpha) \in P\}$ (expansion step)

and $\delta(\cdot, \cdot, \cdot) = \emptyset$ for all other inputs.

WTS: $L(G) = N(M)$ (empty stack acceptance)

Prop: $N(M) \subseteq L(G)$, i.e., given $w \in N(M)$, show that $w \in L(G)$

Suppose $w \in N(M)$ is arbitrary.

Let (q, y, γ) be some config of M along the accepting path of M on input w.

y is a suffix of w. Let $x \in \Sigma^*$ such that

$w = xy$ [x is the read portion of w] 2

Claim: In G $S \underbrace{\Rightarrow^*}_{\text{0 or more steps.}} x\gamma$.

Proof of the claim is by induction on the number of steps of M to get to (q, y, γ):

Base case: Initially: (q, w, S) (initial configuration)

$x = \varepsilon$ and $\gamma = S$. $S \Rightarrow^* \varepsilon S = S$ (zero steps)

Inductive case 1: $(q, \overbrace{ay}, \overbrace{a\beta}) \overset{\text{matching step}}{\vdash} (q, y, \beta)$

some $a \in \Sigma$, $y \in \Sigma^*$, $\beta \in (V \cup \Sigma)^*$

~~Assuming~~ $\cancel{S \Rightarrow^* ay}$

Let x be such that $w = xay$

Assuming $S \Rightarrow^* \underbrace{x}_{\text{read portion}}\overbrace{a\beta}^{\text{stack contents}}$ (inductive hypothesis

then $S \Rightarrow^* \underbrace{xa}_{\text{read portion}}\overbrace{\beta}^{\text{stack contents}}$ (no additional derivation steps)

Inductive case 2: $(q, y, A\beta) \vdash (q, y, \alpha\beta)$

where $A \to \alpha \in P$, $\beta \in (V \cup \Sigma)^*$

Let x be such that $w = xy$.

If $S \Rightarrow^* xA\beta$ then $S \Rightarrow^* x\alpha\beta$ (one more step in the derivation).

By induction, the claim is proven. //

To show that $w \in L(G)$, note the last config of ~~M~~ M accepting w is $(q, \varepsilon, \varepsilon)$. By the claim, $S \Rightarrow^* \underbrace{w}_{\text{read portion}}\varepsilon = w$

$\therefore w \in L(G)$

$\therefore N(M) \subseteq L(G)$ since $w \in N(M)$ was chosen arbitrarily. ∎

Prop: $L(G) \subseteq N(M)$.

Lemma: Suppose $A \to \alpha$ is a production. Write α uniquely as $x\beta$ where $x \in \Sigma^*$ and either $\beta = \varepsilon$ or β starts with some nontermin

Then, for any $y \in \Sigma^*$ and $\gamma \in (\Sigma \cup V)^*$,

In M $(q, xy, A\gamma) \underbrace{\vdash^*}_{\text{zero or more computation steps}} (q, y, \beta\gamma)$

Pf of the lemma:

$$(q, xy, A\gamma) \underset{\substack{\text{expand} \\ \text{on } A \to \underbrace{x\beta}_{\alpha}}}{\vdash} (q, xy, x\beta\gamma) \underbrace{\vdash \cdots \vdash}_{\text{matching steps}} (q, y, \beta\gamma)$$

//

Suppose $w \in L(G)$. Then there is a leftmost derivation of w: ~~A~~

$$S \Rightarrow x_1 A_1 \beta_1 \Rightarrow x_2 A_2 \beta_2 \Rightarrow \cdots \Rightarrow w$$

$x_1, x_2, \ldots \in \Sigma^*$, $A_1, A_2, \ldots \in V$

(leftmost vars in each sentential form.)

Then by the lemma, in M, $w = x_1 y_1$ (some $y_1 \in \Sigma^*$)

$$(q, w, S) \vdash^* (q, y_1, A_1\beta_1)$$

$$\underbrace{\vdash \cdots\cdots \vdash}_{\text{proof omitted}} (q, \varepsilon, \varepsilon)$$

applying the lemma repeatedly. //// proof sketch

M is called a top-down parser, or an ~~LL~~ LL-parser (Left-to-right, Leftmost derivation)

Ex: $S \to (S)S \mid \varepsilon$

$\Sigma = \{ '(' , ')' \}$

$V = \{ S \}$

$M = \langle \{q\}, \Sigma, \Sigma \cup V, \delta, q, S, \emptyset \rangle$

where $\delta(q, '(', '(') = \{(q, \varepsilon)\}$ matching

$\delta(q, ')', ')') = \{(q, \varepsilon)\}$

$\delta(q, \varepsilon, S) = \{(q, '(S)S'), (q, \varepsilon)\}$

Leftmost derivation of (())() in G

$S \Rightarrow (S)S \Rightarrow ((S)S)S \Rightarrow (()S)S \Rightarrow (())S$

$\Rightarrow (())(S)S$

$\Rightarrow (())()S$

$\Rightarrow (())()$

Accepting computation of M:

$(q, (())(), S) \vdash^{exp} (q, (())(), (S)S) \vdash^{match} (q, ())(), S)S)$

$\vdash (q, ())(), (S)S)S) \vdash (q,))(), S)S)S)$

$\vdash (q,))(),)S)S) \vdash (q,)(), S)S)$

$\vdash (q,)(),)S) \vdash (q, (), S)$

$\vdash (q, (), (S)\cancel{S}) \vdash (q,), S)S)$

$\vdash (q,),)S) \vdash (q, \varepsilon, S)$

$\vdash (q, \varepsilon, \varepsilon)$ accept.

☆ Leftmost derivation steps in G correspond to expansion steps in M in the same order, with matching steps in between to clear the terminals from the top of the stack.

Unambiguous grammar for arith exprs with constants (c), and binary ops +, *, & parentheses.

$E \to T\,T'$

$T' \to \varepsilon \mid \cancel{F\,E\,T'}$ $+\,T\,T'$

$T \to F\,F'$

$F' \to \varepsilon \mid *\,F\,F'$

$F \to c \mid (E)$

T' – optionally more terms

F' – optionally more factors

Leftmost derivation of $c * (c + c)$

$E \Rightarrow TT' \Rightarrow FF'T' \Rightarrow cF'T'$

$(q, c*(c+c), E) \vdash (q, c*(c+c), TT') \vdash (q, c*(c+c), FF'T')$

$\Rightarrow c * FF'T' \Rightarrow c*(E)F'T'$

$\vdash (q, c*(c+c), cF'T') \vdash$ match $\vdash \ldots$

$\Rightarrow c * (TT')F'T' \Rightarrow c*(FF'T')F'T'$

$\Rightarrow c*(cF'T')F'T' \Rightarrow c*(cT')F'T'$

$\Rightarrow c*(c+TT')F'T' \Rightarrow c*(c+FF'T')F'T'$

$\Rightarrow c*(c+cF'T')F'T' \Rightarrow \underset{\substack{\text{all } T' \to \varepsilon \\ \text{or } F' \to \varepsilon \\ \text{productions}}}{\ldots} \Rightarrow c*(c+c)$