CSCI 301 - Lecture 25: Parsing II

LL(1) Parsing:

- Left-to-right
- leftmost derivation
- 1 symbol lookahead

G_0: $A \to aA$ (1)
$\quad | \; b$ (2)

Parse aaab: $S \Rightarrow aA$
aaab $\quad aaA$
aaab $\quad aaaA$
aaab $\quad aaab$

G_1: $S \to aS \mid bS \mid \varepsilon$

G_2: $S \to Sa \mid Sb \mid \varepsilon$

$w = aabb$

Ex. Pt A

G_1:

$S \Rightarrow aS$
aaS
aabS
aabbS
aabb

G_2: $S \Rightarrow Sb$
Sbb
Sabb
Saabb
aabb

Eliminating Left Recursion

$$S \to Sa \mid Sb \mid c \mid d$$

$$\begin{cases} S \to cS' \mid dS' \\ S' \to aS' \mid bS' \mid \varepsilon \end{cases}$$

optional but helpful intuition:

Regex: $(c \cup d)(a \cup b)^*$

General approach:

- Make new variable and append to all non-recursive rules
- Swap recursion order for recursive rules

Ex. Pt 3

$$S \to Sa \mid Sb \mid AB \mid C \mid \varepsilon$$

$$S \to ABS' \mid CS' \mid S'$$
$$S' \to aS' \mid bS' \mid \varepsilon$$

Exercise

$$S \to Sa \mid Sb \mid AB \mid C \mid \varepsilon$$

$$\begin{cases} S \to ABS' \mid CS' \mid S' \\ S' \to aS' \mid bS' \mid \varepsilon \end{cases}$$

Common Prefixes : Left Factoring

$$S \to aCbS \mid aCbSeS$$

$$S \to aCbSE \qquad S \to aCbSE$$
$$E \to eS \mid \varepsilon \qquad E \to eS \mid \varepsilon$$

Why?

$S \to$ if C then S
| if C then S else S

Ex. Pt C.

It's even worse!

$A \to da \mid abB$

$B \to abB \mid daA \mid Af$

$abBf$

Ex. C:

$S \to SS+ \mid SS* \mid a$

$S \to SSM \mid a$

$M \to + \mid *$

$A \to B$

$B \to Ax$

$A \overset{*}{\Rightarrow} Ax$

Reverse Polish Notation

math → Infix: 1 + (2-4) * 3

racket → Prefix (Polish) : (+ 1 (* (- 2 4) 3))

easiest to parse! → Postfix (Reverse Polish) : 1 4 2 - 3 * +