PUMPING LEMMA FOR CFL

IF L IS A CFL THEN $\exists N > 0$,
SUCH THAT IF $z \in L$, $|z| \geq N$, THEN

$\forall_{z_{|z| \geq N}} \exists u, v, w, x, y \; \forall i \; [z = uvwxy, |vwx| \leq N, |vx| > 0$
$\& \; uv^i w x^i y \in L$

COMPLEMENT

$\exists_{z_{|z| \geq N}} \forall u, v, w, x, y \; \exists i \; [z = uvwxy, |vwx| \leq N, |vx| > 0,$
$\& \; uv^i w x^i y \underline{\underline{\notin}} L]$

NOTE: i IS DEPENDENT ON z, u, v, w, x, y

SOME TEXTS ONLY SHOW EXAMPLES
WHERE i IS FIXED NO MATTER HOW
$uvwxy$ PARSES z

FAILS FOR $\{a^i b^j c^k \mid k = \max(i,j)\}$
$\&$ $\{a^i b^j c^k \mid k = \min(i,j)\}$

EVEN THOUGH BOTH ARE NON-CFLs

Phrase Structured Grammar

We previously defined PSGs. The language generated by a PSG is a Phrase Structured Language (PSL) but is more commonly called a recursively enumerable (re) language. The reason for this will become evident a bit later in the course.

The recognizer for a PSL (re language) is a Turing Machine, a model of computation we will soon discuss.

QUICK EXAMPLE OF LBA

$L = \{ a^n b^n c^n \mid n > 0 \}$ $\qquad q_0 ¢ \; w \; \$$

$q_0 ¢ \to ¢ q_1$

$q_1 a \to x q_2$

$q_2 b \to y q_3$

$q_3 c \to q_4 z$

$q_2 a \to a q_2$

$q_3 b \to b q_3$

$z q_4 \to q_4 z$

$q_2 y \to y q_2$

$q_3 z \to z q_3$

$b q_4 \to q_4 b$

$y q_4 \to q_4 y$

$a q_4 \to q_4 a$

$x q_4 \to x q_1$

$q_f z \to z q_f$

$q_1 y \to y q_f$

$q_f y \to y q_f$

$q_f \$ \to \$ q_f$

$$q_0 ¢ a^n b^n c^n \$ \vdash^* ¢ x^k q_1 a^{n-k} y^k b^{n-k} z^k c^{n-k} \$$$

$$\vdash^* ¢ x^n q_1 y^n z^n \$ \vdash^* ¢ x^n y^n z^n \$ q_f$$

TAPE ALPHABET IS

$\{¢, \$, x, y, z, a, b, c\}$

A NASTY VARIANT OF

$L_1 = \{ww \mid w \in \{a,b\}^+\}$

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

$\cup \{ww^R \mid w \in \{a,b\}^+\}$ CFL

TO SHOW THIS IS NOT A CFL WE EMPLOY PUMPING LEMMA

- LET $N>0$ BE FROM PL
- I CHOOSE $a^N b^N a^N b^N$ BECAUSE IT WORKED FOR L_1
- PL SAYS $a^N b^N a^N b^N = uvwxy$, $|vwx| \le N$, $|vx| > 0$ & $\forall i \ge 0\ uv^iwx^iy \in L$
- I CHOOSE $i=0$, AGAIN BECAUSE IT WORKED FOR L_1 AND STATE WE CAN REDUCE EITHER SOME a'S ~~AND MAYBE~~ SOME b'S OR SOME b'S AND MAYBE SOME a'S
I THEN NOTE THAT IF I REDUCE a'S AT BEGINNING, I CANNOT ALSO REDUCE a'S STARTING AT MIDDLE AND VICE VERSA. I MAKE SAME STATEMENT ABOUT b'S AND FEEL DONE, BUT THIS JUST SHOWS $uv^0wx^0y \notin L_1$. WHAT ABOUT L_2?

MORE OF WW VARIANT

IT MIGHT BE THAT $|vx| = N$ AND $|w| = 0$. IN THIS CASE, vx COULD BE OVER ALL a'S IN BEGINNING, LEAVING $b^N a^N b^N$ WHICH IS IN $\{ww^R \mid w \in \{a,b\}^+\}$ PROVIDED N IS AN EVEN. SO IS $a^N b^N a^N$ (ERASE FINAL b'S)

OOPS!!

SOLUTION

$i = 2$ WORKS

ALSO, COULD CHOOSE

$a^{N+1} b^{N+1} a^{N+1} b^{N+1}$

GREIBACH NORMAL FORM

ALL RULES, EXCEPT PERHAPS, $S \to \lambda$

LIMITED TO

$$A \to a\alpha \qquad A \in V, a \in \Sigma, \alpha \in V^*$$

PROVIDES LINEAR PARSE IF

WE CAN AVOID

SHIFT/REDUCE

&

REDUCE/REDUCE

CONFLICTS

Context Sensitive

Context Sensitive Grammar

$G = (V, \Sigma, R, S)$ is a PSG where

Each member of R is a rule whose right side is no shorter than its left side.

The essential idea is that rules are length preserving, although we do allow $S \to \lambda$ so long as S never appears on the right hand side of any rule.

A context sensitive grammar is denoted as a CSG and the language generated is a Context Sensitive Language (CSL).

The recognizer for a CSL is a Linear Bounded Automaton (LBA), a form of Turing Machine (soon to be discussed), but with the constraint that it is limited to moving along a tape that contains just the input surrounded by a start and end symbol.

CSG Example#1

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

$G = (\{A,B,C\}, \{a,b,c\}, R, A)$ where R is

$A \rightarrow aBbc \mid abc$

$B \rightarrow aBbC \mid abC$

Note: $A \Rightarrow aBbc \Rightarrow n\ a^{n+1}(bC)^n\ bc$ // n>0

$Cb \rightarrow bC$ // Shuttle C over to a c

$Cc \rightarrow cc$ // Change C to a c

Note: $a^{n+1}(bC)^n\ bc \Rightarrow^* a^{n+1}b^{n+1}c^{n+1}$

Thus, $A \Rightarrow^* a^n b^n c^n$, n>0

$A \Rightarrow aBbc$

$\Rightarrow aaBbbCc \Rightarrow aaBbbcc$

$\Rightarrow aaabCbbcc$

$\Rightarrow^* aaabbbccc$

CSG Example#2

$L = \{ ww \mid w \in \{0,1\}^+ \}$

$G = (\{S,A,X,Z,<0>,<1>\}, \{0,1\}, R, S)$ where R is

$S \rightarrow 00 \mid 11 \mid 0A<0> \mid 1A<1>$		
$A \rightarrow 0AZ \mid 1AX \mid 0Z \mid 1X$		
$Z0 \rightarrow 0Z$	$Z1 \rightarrow 1Z$	// Shuttle Z (for owe zero)
$X0 \rightarrow 0X$	$X1 \rightarrow 1X$	// Shuttle X (for owe one)
$Z<0> \rightarrow 0<0>$	$Z<1> \rightarrow 1<0>$	// New 0 must be on rhs of old 0/1's
$X<0> \rightarrow 0<1>$	$X<1> \rightarrow 1<1>$	// New 1 must be on rhs of old 0/1's
$<0> \rightarrow 0$		// Guess we are done
$<1> \rightarrow 1$		// Guess we are done

CSG AND QUOTIENT AND HOMOMORPHISM (λ ALLOWED)

START WITH PSG, FOR LANG. L

IF $\alpha \to \beta$ & $|\alpha| \le |\beta|$ keep

$\alpha \to \beta$

IF $\alpha \to \beta$ & $|\alpha| > |\beta|$ INCLUDE

$$\underbrace{\alpha \to \beta \, *^{|\alpha|-|\beta|}}_{\text{LENGTH PRESERVING}}$$

WHERE $*$ IS NEW NON-TERMINAL

INCUDE ALSO

$S' \to S¢\$$

OLD START

S' NEW START

\$ NEW SYMBOL TERMINAL

AND

$$*X \to X*$$
$$*\$ \to \$\$$$

$X \in V \cup \Sigma \cup \{¢\}$

IF $h(a) = a$ $\quad a \in \Sigma$

$h(\$) = \lambda$ $\quad h(¢) = \lambda$

THEN $h(L') = L$

L' IS OUTPUT OF NEW GRAMMAR

AND $L' / ¢\$^+ = L$

LBA

SIMPLE VIEW
R/W TAPE

q_0 ¢ $\boxed{\quad w \quad}$ $\$$

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 MULTITRACK
(FINITE # OF TRACKS)
FOR EXAMPLE $(\{¢,\$\} \cup \Sigma) \times (\{¢,\$\} \cup \Gamma)$
AS TAPE ALPHABET WITH CHANNEL
1 HAVING INPUT.