Lecture 11: CFL normal forms & pumping lemma

CSCI 101
Spring, 2019

Kim Bruce

Take-home midterm coming in 2 weeks!

Normal Forms

- Because of ε-productions, can be hard to determine if w in L.
  - Parsers recognize terms of language and build abstract syntax tree *(thinned down parse tree)*
- Normal forms can make it easier.
- Chomsky and Greibach Normal Forms
  - Do only Chomsky

Chomsky Normal Form (CNF)

- A grammar is in Chomsky Normal Form if all productions are of form $A \to BC$, or $A \to a$. Advantages:
  - Parse trees are all binary
  - To see if w in L try all derivations of length < $2|w|$
  - Efficient parsing algorithms (CYK)
    - but watch blowup in size!
  - Disadvantage: Leave out ε!

Converting grammar to CNF

- Theorem: If L is a cfl, there is a cfg G' in Chomsky Normal Form such that L(G') = L - {ε}
- Proof:
  - Eliminate ε-productions from all vbles
    - If $A \to \varepsilon$ is a rule, then drop it and for all rules of the form $B \to w$, add all rules of the form $B \to w'$ where w' formed by dropping one or more A's from right side of w.
    - Note: Don't add $B \to \varepsilon$ if it has already been dropped.

Converting grammar to CNF

- Eliminate unit productions (size of right is 1)
  - If $A \to B$ is a rule, then drop it, and for each production $B \to w$, add $A \to w$.
  - Note: If $A \to w$ is a unit production that was already eliminated, then don't add it back.
- Eliminate long right sides:
  - If $A \to W_1 \ldots W_n$ where each $W_i \in V$, replace by
    - $A \to W_1X_1$, $X_1 \to W_2X_2$, ..., $X_{n-2} \to W_{n-1} W_n$ where X_i are new.

Converting grammar to CNF

- Eliminate terminals on the right side:
  - For each terminal $a \in \Sigma$, add new non-terminal N_a and production $N_a \to a$. For each production of the form $U \to w$, for $|w| = 2$, replace all terminals a in w by corresponding N_a.
- Should be clear get same language except for ε.

Example

- Start with $S \to UaabU$,
  $U \to aU \mid bU \mid \varepsilon$
- Eliminate ε-productions:
  - $U \to aU \mid bU \mid a \mid b$
    $S \to UaabU \mid aabU \mid Uaab \mid aab$
- Eliminate unit productions:
  - None -- so nothing to do

Example

- Shorten long productions & eliminate terminals

$S \to U\,C_0 \mid AD_0 \mid UE_0 \mid AF_0$
$C_0 \to AC_1$
$C_1 \to AC_2$
$C_2 \to BU$
$D_0 \to AD_1$
$D_1 \to BU$
$E_0 \to AE_1$
$E_1 \to AB$
$F_0 \to AB$

$U \to AU \mid BU \mid a \mid b$
$A \to a$
$B \to b$

$U \to aU \mid bU \mid a \mid b$
$S \to UaabU \mid aabU \mid Uaab \mid aab$

Pumping Lemma for CFLs

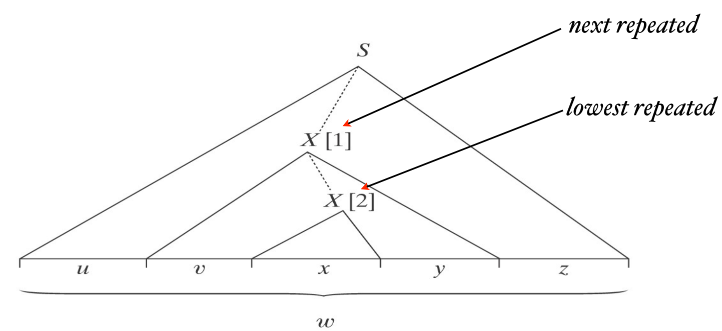
- For each CFL L, there is a $k > 1$ s.t. for all $w \in L$ of length at least k, there are u, v, x, y, and z s.t.
  - $w=uvxyz$;
  - $|vxy| \le k$;
  - $vy \ne \varepsilon$; and
  - for each non-negative integer i, $uv^ixy^iz \in L$.
- More complex than regular, but same idea of repetition

Use parse trees

- Theorem: Length of the yield of tree of height h and branching factor b is $\le b^h$
- Let G be in CNF w/ n non-terminals. If T generated by G and no non-terminal appears more than once on any path. Then
  - Max height of T is n
  - Max length of T's yield is 2^n
- *Equivalently*: if w in L(G) s.t. $|w| > 2^n$ then the parse tree height is greater than n.

Proof of Pumping

- Let G be grammar, and G' be equivalent grammar in CNF. Thus branching factor = 2.
- Proof by picture: let $k = 2^{n+1}$ and w s.t. $|w| \ge k$. Therefore height > n & hence exists repeated non-terminal on a path



PL Proof

- From picture
  - $S \Rightarrow^* uXz$
  - $X \Rightarrow^* vXy$ (*Note: $vy \neq \varepsilon$*)
  - $X \Rightarrow^* x$
  - $|vxy| \leq k$ ($= 2^{n+1}$) because height $\leq n+1$
- Hence can get
  - $S \Rightarrow^* uXz \Rightarrow^* uvXyz \Rightarrow^* uv^iXy^iz \Rightarrow^* uv^ixy^iz$ for any i
  - Can also get $S \Rightarrow^* uXz \Rightarrow^* uxz$

Using Pumping Lemma

- To show L not cfl
  - *Opponent* picks k
  - *I* pick w s.t. $|w| \geq k$
  - *They* pick decomposition $w = uvxyz$ s.t. $|vxy| \leq k$, $vy \neq \varepsilon$
  - *I* show there is some i s.t. $u\, v^i\, x\, y^i\, z \notin L$
- Note: *I* can't predict where vxy starts!

Example

- Show $L = \{a^nb^nc^n \mid n \geq 0\}$ is not a cfl.
  - Assume cfl w/k for P.L. Choose $w = a^k\, b^k\, c^k$.
  - They break into $w = uvxyz$ such that $|vxy| \leq k$, $vy \neq \varepsilon$.
  - vxy cannot contain both a's *and* c's
    - Spose vxy contains no c's: get contradiction if pump!
    - Similarly if no a's

Example

Text does wcw

- $L = \{ww : w \in \{0,1\}^*\}$ is not cfl
  - Spose cfl w/ k for P.L. Let $w = 0^k1^k0^k1^k$
  - They choose u,v,x,y,z s.t. $|vxy| < k$, $vy \neq \varepsilon$
  - If vy all in first 0 section then pumping by 2 disallows split: $uvvxyyz = 0^{k+j}1^k0^k1^k$ where $i > 0$. Can't be split
    - Same for other 3 homogeneous sections: 1^*, 0^*, 1^*
    - Can't reach around 3 sections as $|vxy| < k$
    - Therefore straddle 2 sections. Pump by 0.
    - E.g., if in middle 1^*0^*, get $uxz = 0^k1^i0^j1^k \notin L$ where i or j < k

Closure properties of CFL's

- Already shown closed under
  - concatenation, union, Kleene*, reversal, substitution
- Also closed under intersection with regular set.
  - Product machine
- Not closed under intersection, difference, or complement.
  - Why doesn't product work for intersection?