Computational Models - Lecture 5[1]

Handout Mode

Iftach Haitner.

Tel Aviv University.

November 28, 2016

[1] Based on frames by Benny Chor, Tel Aviv University, modifying frames by Maurice Herlihy, Brown University.

Talk Outline

- Chomsky Normal Form (CNF)
- Checking membership in a CNF grammer
- Pumping Lemma for context free languages
- Push Down Automata (PDA)

- Sipser's book, 2.1, 2.2 & 2.3

CFGs and CFLs, reminder

A context-free grammar is a 4-tuple (V, Σ, R, S), where

- V is a finite set of variables
- Σ is a finite set of terminals $(V \cap \Sigma = \emptyset)$
- R is a finite set of rules of the form $A \to x$, where $A \in V$ and $x \in (V \cup \Sigma)^*$.
- $S \in V$ is the start symbol.

- Let $u, v \in (V \cup \Sigma)^*$. If $A \to w \in R$, then uAv yields uwv, denoted $uAv \to uwv$.
- $u \overset{*}{\to} v$ if $u = v$, or $u \to u_1 \to \ldots \to u_k \to v$ for some sequence $u_1, u_2, \ldots, u_k$

Definition 1

The language of the grammar G, denoted $\mathcal{L}(G)$, is $\{w \in \Sigma^* : S \overset{*}{\to} w\}$

where $\overset{*}{\to}$ is determined by G.

Part I

Chomsky Normal Form (CNF)

Checking membership in a CFL

Challenge

Given a CFG G and a string w, decide whether $w \in \mathcal{L}(G)$?

Initial Idea: Design an algorithm that tries all derivations.

Problem: If G does not generate w, we'll never stop.

Possible solution: Use special grammars that are:

- just as expressive!
- better for checking membership.

Chomsky Normal Form (CNF)

A simplified, canonical form of context free grammars.

$G = (V, \Sigma, R, S)$ is in a CNF, if every rule in in R has one of the following forms:

$$\begin{array}{lll} A \to a, & A \in V & \wedge \quad a \in \Sigma \\ A \to BC, & A \in V & \wedge \quad B, C \in V \setminus \{S\} \\ S \to \varepsilon. & & \end{array}$$

Simpler to analyze: each derivation adds (at most) a single terminal, S only appears once, ε appears only at the empty word

What does parse tree look like?

Most internal nodes are degree 2 (except parents of leaves, which are degree 1)

Generality of CNF

Theorem 2

Any context-free language is generated by a context-free grammar in Chomsky Normal Form.

Proof Idea:

- Add new start symbol S_0.
- Eliminate all ε rules of the form $A \to \varepsilon$.
- Eliminate all "unit" rules of the form $A \to B$.
- Convert remaining "long rules" to proper form.

Add new start symbol

Add new start symbol S_0 and rule $S_0 \to S$

(Guarantees that new start symbol is never on right hand side of a rule)
e.g.

$$\begin{aligned} S &\to A \mid ab \mid \varepsilon \\ A &\to baA \mid S \end{aligned}$$

becomes

$$\begin{aligned} S_0 &\to S \\ S &\to A \mid ab \mid \varepsilon \\ A &\to baA \mid S \end{aligned}$$

Convert "long rules": terminals

$$
\begin{aligned}
S &\rightarrow ccAbA \mid bc \mid b \\
A &\rightarrow a \mid bb
\end{aligned}
$$

becomes

$$
\begin{aligned}
S &\rightarrow CCABA \mid BC \mid b \\
A &\rightarrow a \mid BB \\
B &\rightarrow b \\
C &\rightarrow c
\end{aligned}
$$

Convert "long rules": multiple nonterminals

$$S \rightarrow AAAB$$

becomes

$$\begin{aligned} S &\rightarrow AN_1 \\ N_1 &\rightarrow AN_2 \\ N_2 &\rightarrow AB \end{aligned}$$

Eliminate "ε-rules"

Repeat until all $A \to \epsilon$ ($A \neq S$) rules are gone:

- remove $A \to \varepsilon$
- for any rule of form $C \to AB$ or $C \to BA$, add $C \to B$.
- for any rule of form $C \to AA$ add $C \to A$ and $C \to \varepsilon$ (unless $C \to \varepsilon$ has already been removed).
- for any rule of form $C \to A$ add $C \to \varepsilon$ (unless $C \to \varepsilon$ has already been removed.)

Eliminate "unit rules"

Repeat until all unit rules removed

- remove some $A \to B$
- for each $B \to U$ (where $U \in (\Sigma \cup \Gamma)^*$), add $A \to U$ (unless $A \to U$ was a previously removed unit rule)

CNF: Example

$$
\begin{aligned}
S &\rightarrow ASA \mid aB \\
A &\rightarrow B \mid S \\
B &\rightarrow b \mid \varepsilon
\end{aligned}
$$

Is transformed into:

$$
\begin{aligned}
S_0 &\rightarrow AA_1 \mid UB \mid a \mid SA \mid AS \\
S &\rightarrow AA_1 \mid UB \mid a \mid SA \mid AS \\
A &\rightarrow b \mid AA_1 \mid UB \mid a \mid SA \mid AS \\
A_1 &\rightarrow SA \\
U &\rightarrow a \\
B &\rightarrow b
\end{aligned}
$$

CNF has bounded derivation length

Lemma 3

Let G be a CFG in CNF and let $w \in \mathcal{L}(G)$ be with $|w| = n \geq 1$. Then every derivation of w by G has a derivation of length $2n - 1$.

Proof? consider the parsing tree for w

Advantage: Easier to check whether $w \in \mathcal{L}(G)$, see next part.

Part II

Checking Membership for CFGs in Chomsky Normal Form

Checking membership for CFG in CNF form

Let $G = (V, \Sigma, R, S)$ be CFG in CNF and let $A \in V$ and $w \in \Sigma^*$

Algorithm 4 (Derive(A, w))

- $w = \varepsilon$: if $A \to \varepsilon \in R$ (i.e., $A = S$) return `TRUE`, otherwise return `FALSE`.
- $|w| = 1$: if $A \to w \in R$ return `TRUE`, otherwise return `FALSE`.
- $|w| > 1$: for each $A \to BC$ and **each** non-trivial partition $w = w_1 w_2$:
  - Call Derive(B, w_1) and Derive(C, w_2).
  - Return `TRUE` if *both* return `TRUE`.
- Return `FALSE`.

Claim: $A \stackrel{*}{\to} w \iff$ Derive$(A, w) =$ `TRUE`. Proof?

$A \stackrel{*}{\to} w \implies$ Derive$(A, w) =$ `TRUE`, by induction on # of derivation steps.

Derive$(A, w) =$ `TRUE` $\implies A \stackrel{*}{\to} w$, by induction on $|w|$.

- Hence, Derive$(S, w) =$ `TRUE` $\iff w \in \mathcal{L}(G)$.
- Procedure Derive can also output a parse tree for w
- Where have we used the fact that G is in CNF?

Time complexity of Derive

What is the time complexity $T : \mathbb{N} \mapsto \mathbb{N}$ of Derive?

- Each recursive call tests $|R|$ rules and n partitions.
- $T(n) \leq |R| \cdot n \cdot 2T(n-1)$
- $T(n) \in O((|R| \cdot n)^n)$.

Still exponential...

Efficient Algorithm

- Keep in memory the results of Derive(A, w).
  - Number of different inputs: $|V| \cdot n^2$.
  - Only $|V| \cdot n^2$ calls, each takes $O(|R| \cdot n)$.
  - $T(n) \in O(|R| \cdot n^3 \cdot |V|)$.
- Polynomial time!

- This approach is called Dynamic Programming

  Basic idea:

  - If number of different inputs is limited, say $I(n)$.
  - Each run (excluding recursive calls) takes at most $R(n)$ time
  - Total running time is bounded by $T(n) \leq R(n)I(n)$.

Part III

Non-Context-Free Languages

Proving a Language is not a CFL

- The pumping lemma for finite automata and Myhill-Nerode theorem are our tools for showing that languages are not regular.
- We will now show a similar pumping lemma for context-free languages.
- It is slightly more complicated . . .

Pumping Lemma for CFL (also known as, the *uvxyz* Theorem)

Theorem 5

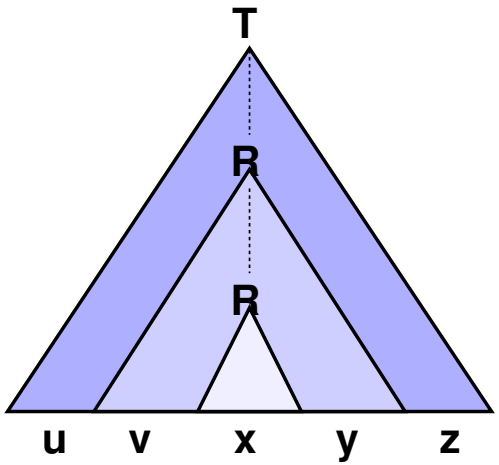
For any CFL $\mathcal{L}$ there exists $\ell \in \mathbb{N}$ ("critical length"), such that for any $w \in \mathcal{L}$ with $|w| \geq \ell$, there exit $u, v, x, y, z \in \Sigma^*$ such that $w = uvxyz$ and

- *For every $i \geq 0$: $uv^i xy^i z \in \mathcal{L}$*
- *$|vy| > 0$, ("non-triviality")*
- *$|vxy| \leq \ell$*

Basic Intuition:
Let $\mathcal{L}$ be a CFL and a let w be a "very long" string in $\mathcal{L}$. Then w must have a "tall" parse tree.

Hence, some root-to-leaf path must repeat a symbol. Why is that so?

Basic Intuition cont.



We have: $T \stackrel{*}{\rightarrow} uRz$, $R \stackrel{*}{\rightarrow} vRy$, and $R \stackrel{*}{\rightarrow} x$.

Proof of Thm 5

Let G be a CFG and let $\mathcal{L} = \mathcal{L}(G)$.

- Let b be the max number of symbols in right-hand-side of any rule (what is b for a CNF grammar?).

  Since no node in a parse tree of G has more than b children, at depth d such tree has at most b^d leaves.

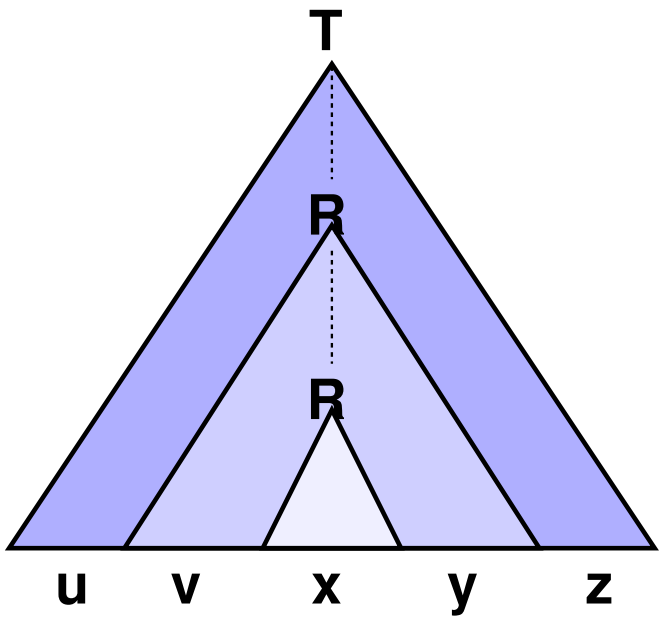
- Let $|V|$ be the number of variables in G, and set $\ell = b^{|V|+2}$.

Let w be a string with $|w| \geq \ell$, and let T be parse tree for w (with respect to G) with fewest nodes

- T has height $\geq |V| + 2$
- Some path in T has length $\geq |V| + 2$
- Such path repeats a variable R

Proof of Thm 5 cont.

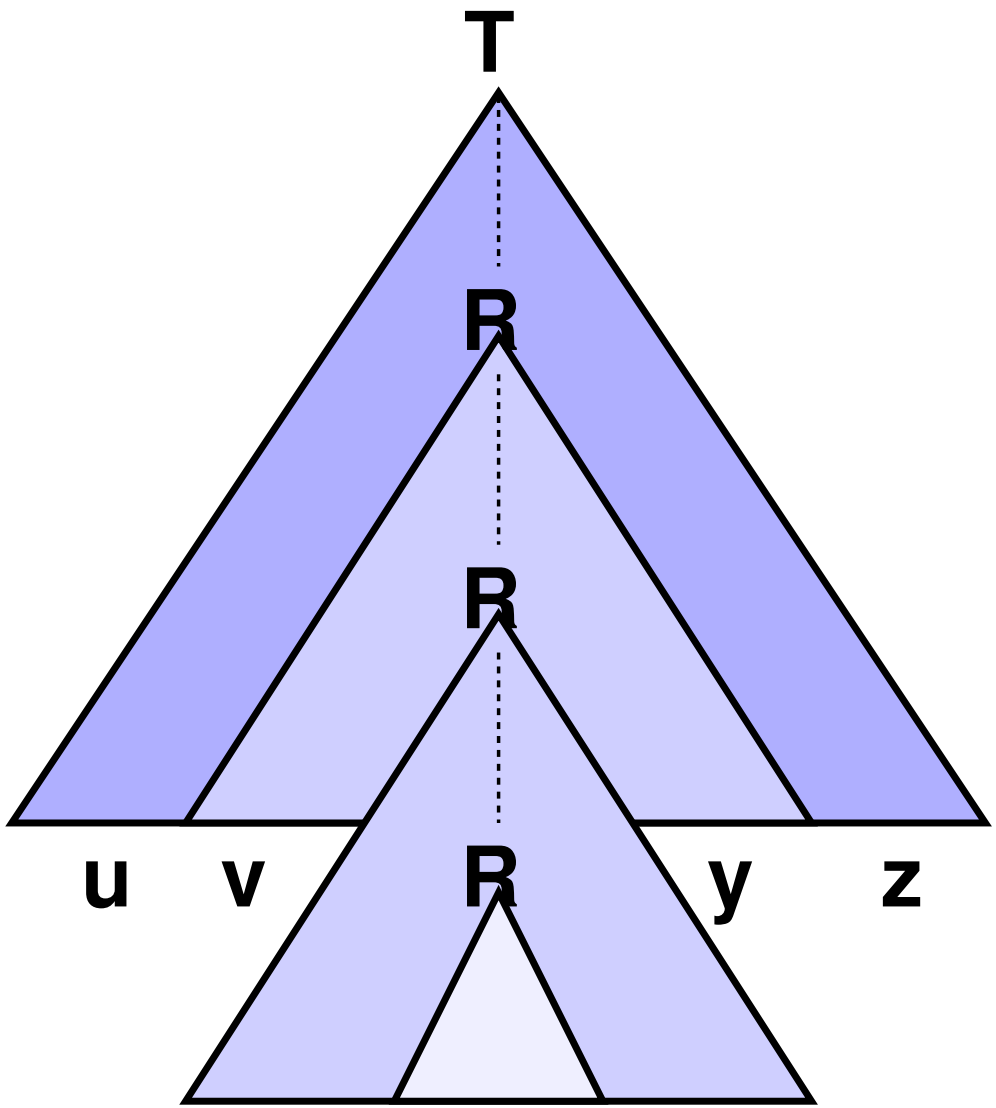
Set $w = uvxyz$



- Each occurrence of R produces a string
- Upper produces string vxy
- Lower produces string x

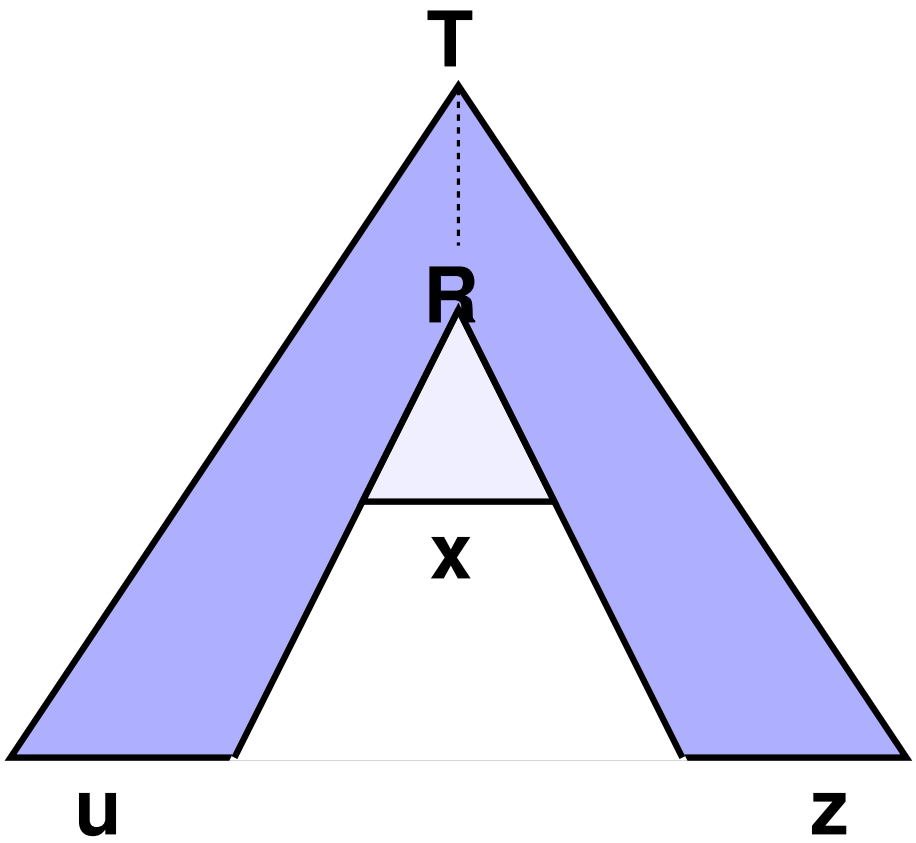
Proving $uv^ixy^iz \in \mathcal{L}$ for all $i > 1$

Replacing smaller by larger yields uv^ixy^iz, for $i > 0$.



Proving $uv^i xy^i z \in \mathcal{L}$ for $i = 0$

Replacing larger by smaller yields uxz.

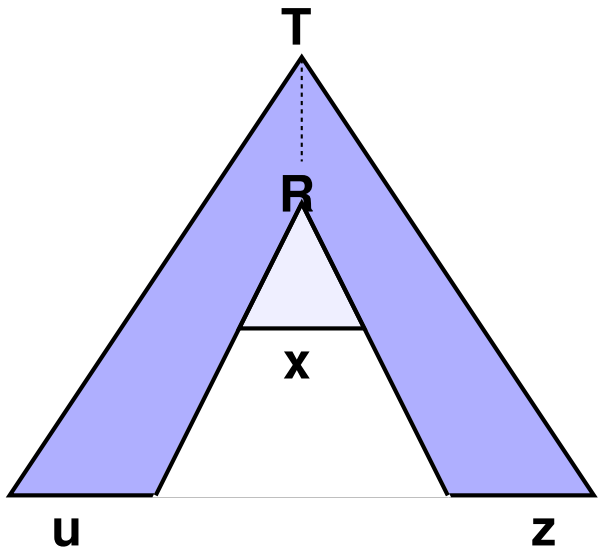


Together, they establish:

- $uv^i xy^i z \in \mathcal{L}$ for all $i \geq 0$

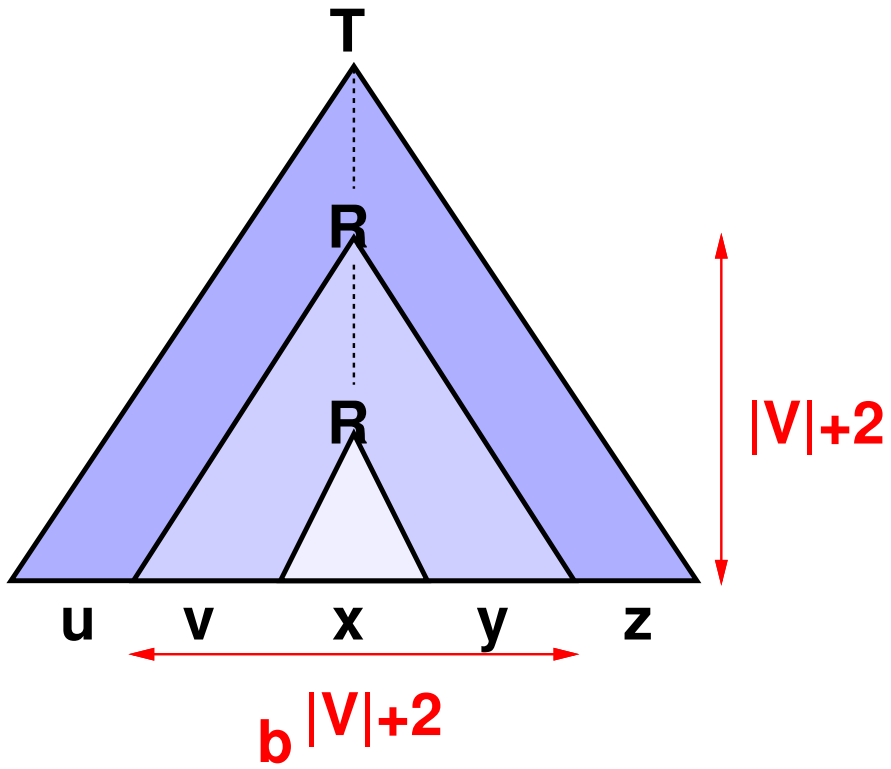
Proving $|vy| > 0$

If v and y are both ε, then



is a parse tree for w with fewer nodes than T, a contradiction.

Proving $|vxy| \leq \ell$



- Without loss of generality both occurrences of R lie in bottom $|V| + 1$ variables on the path.
- The upper occurrence of R (from now on R^1) generates vxy.
- Subtree rooted at R^1 is of height at most $|V| + 2$.

  Hence, $|vxy| \leq b^{|V|+2} = \ell$.

Non CFL Example (1)

Claim 6

$\mathcal{L}_1 = \{a^n b^n c^n : n \in \mathbb{N}\}$ is not a CFL.

Proof: By contradiction. Assume $\mathcal{L}_1$ is a CFL with grammar G, let ℓ be the critical length of G and consider $w = a^\ell b^\ell c^\ell$. Let u, v, x, y, z be the strings with $w = uvxyz$ guaranteed by Thm 5 for w.

- Both v and y are monochromatic — repetitions of of a single letter.

  Proof: Otherwise, uv^2xy^2z would have out-of-order symbols.

- Hence, uv^2xy^2z is imbalanced

♣

Non CFL Example (2)

Claim 7

$\mathcal{L}_2 = \{a^i b^j c^k : 0 \leq i \leq j \leq k\}$ is not context free.

Proof: By contradiction. Assume $\mathcal{L}_2$ is a CFL with grammar G, let ℓ be the critical length of G and consider $w = a^\ell b^\ell c^\ell$. Let u, v, x, y, z be the strings with $w = uvxyz$ guaranteed by Thm 5 for w.

Both v and y are monochromatic.

- If neither v nor y contains a, then uv^0xy^0z has too many a's.
- If neither v nor y contains c, then uv^2xy^2z has too few c's.
- If neither v nor y contains b, then
  - either $|v| > 0$, and then uv^2xy^2z has more a's than b's.
  - or $|y| > 0$, and then uv^0xy^0z has more b's than c's

♣

Non CFL Example (3)

Claim 8

$\mathcal{L}_3 = \{ww : w \in \{0,1\}^*\}$ is not context-free.

Proof:
By contradiction. Assume $\mathcal{L}_3$ is a CFL with grammar G, let ℓ be the critical length of G and consider $w = 0^\ell 1^\ell 0^\ell 1^\ell$. Let u, v, x, y, z be the strings with $w = uvxyz$ guaranteed by Thm 5 for w.

- Assuming vxy is in the first half of w, then uv^2xy^2z "moves" a 1 into the first position of second half.
- Assuming vxy is in the second half, then uv^2xy^2z "moves" a 0 into the last position of first half.
- Assuming vxy straddles the midpoint, then pumping *down* to uxz yields $0^\ell 1^i 0^j 1^\ell$ where i and j cannot both be ℓ.

♣

Note that $\{ww^R : w \in \{0,1\}^*\}$ is a CFL.