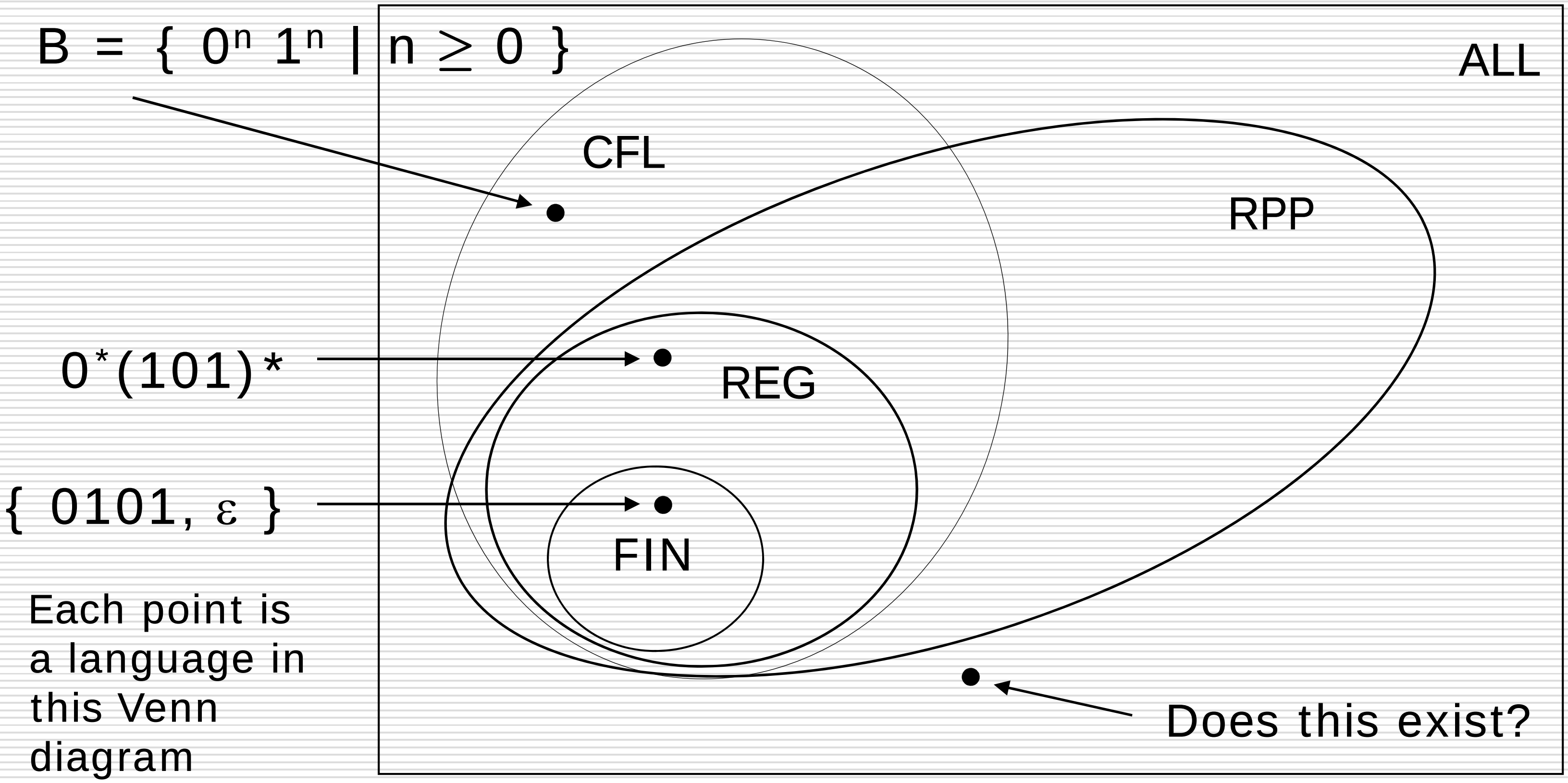
91.304 Foundations of (Theoretical) Computer Science

Chapter 2 Lecture Notes (Section 2.3: Non-Context-Free Languages)

David Martin
dm@cs.uml.edu

With some modifications by Prof. Karen Daniels, Fall 2012

Picture so far



Each point is a language in this Venn diagram

Strategy for finding a non-CFL

- Just as we produced non-regular languages with the assistance of RPP, we'll produce non-context-free languages with the assistance of the *context-free pumping property*
  - First we show that CFL $\subseteq$ CFPP
  - And then show that a particular language L is not in CFPP
  - Hence L can not be in CFL either

The Context-Free Pumping Property, CFPP

Definition L is a member of CFPP if

- There exists $p \geq 0$ such that
  - For every $s \in L$ satisfying $|s| \geq p$,
    - There exist **u,v,**x,y,z $\in \Sigma^*$ such that
      1. s=**uv**xyz
      2. |**v**y|>0
      3. |**v**xy|**≤** p
      4. For all $i \geq 0$,

         $\mathbf{u\ v^i\ x\ y^i\ z \in L}$

bold, red text shows differences from RPP

The non-CFPP

Rephrasing L is **not** in CFPP if

- For *every* $p \geq 0$
  - There exists some $s \in L$ satisfying $|s| \geq p$ such that
    - For every **u,v,**x,y,z $\in \Sigma^*$ satisfying 1-3:
      1. s=**uv**xyz,
      2. |**v**y|>0, and
      3. |**v**xy|**≤** p
    - There exists some $i \geq 0$ for which

$$\mathbf{u\ v^i\ x\ y^i\ z} \not\subseteq \mathbf{L}$$

Game theory formulation

- The direct (non-contradiction) proof of non-context-freeness can be formulated as a two-player game
  - **You** are the player who wants to establish that L is not CF-pumpable
  - Your **opponent** wants to make it difficult for you to succeed
  - Both of you have to play by the rules
  - Same setup as with regular pumping (RPP)

Game theory continued

The game has just four steps.

1. Your **opponent** picks $p \geq 0$
2. **You** pick $s \in L$ such that $|s| \geq p$
3. Your **opponent** chooses $u,v,x,y,z \in \Sigma^*$ such that $s=uvxyz$, $|vy|>0$, and $|vxy| \leq p$
4. **You** produce some $i \geq 0$ such that $uv^ixy^iz \notin L$

Game theory continued

- If you are able to succeed through step 4, then you have won only one round of the game
- To show that a language is not in CFPP you must show that you can **always** win, regardless of your opponent's legal moves
  - Realize that the opponent is free to choose the most inconvenient or difficult p and u,v,x,y,z imaginable that are consistent with the rules

Game theory continued

- So you have to present a *strategy* for always winning — and convincingly argue that it will always win
  - So your choices in steps 2 & 4 have to depend on the opponent's choices in steps 1 & 3
  - And you don't know what the opponent will choose
  - So your choices need to be framed in terms of the variables p, u, v, x, y, z

Towards proving CFL $\subseteq$ CFPP

- To prove the claim that CFL $\subseteq$ CFPP we'll simplify things by using Chomsky Normal Form (CNF)
- **Recall:** a CFG $G=(V,\Sigma,R,S_0)$ is in Chomsky Normal Form if each rule is of one of these forms:
  - $A \rightarrow BC$, where A, B and C $\in$ V, and $B \neq S_0$ and $C \neq S_0$ (neither B nor C is the start symbol)
  - $A \rightarrow c$, where $A \in V$ and $c \in \Sigma$
  - $S_0 \rightarrow \varepsilon$, where S_0 is the grammar's start symbol (this is the only ε production allowed)
- **Recall:** Every context-free language L has a grammar G that is in Chomsky Normal Form

Towards proving CFL $\subseteq$ CFPP: Length constraints

We will use some handy facts about CNF grammars.

Definition. Suppose s is some string generated by a CNF grammar G. Then let **minheight**(s) be the height (number of levels - 1) in the shortest parse tree for s in the grammar G.

Example: minheight(ε) $\geq$ 1 for every G

Towards proving CFL $\subseteq$ CFPP: Length constraints

- **Lemma** Suppose G is in Chomsky Normal Form. Then

1. For all $n \geq 1$, if minheight(s)$\leq$n then $|s| \leq 2^n$. In other words, constraining the height of a parse tree also constrains the length of the string.
   1. Recall length of string = # terminals = # leaves of parse tree.
2. For all $n \geq 0$, if $|s| > 2^n$, then minheight(s)$>$ n. In other words, large strings come from tall trees.

- (The 2 in 2^x comes from the fact that each node in a parse tree for s has at most two children, because the grammar is in CNF.)

The Context-Free Pumping Property, CFPP (repeat)

Definition L is a member of CFPP if

- There exists $p \geq 0$ such that
  - For every $s \in L$ satisfying $|s| \geq p$,
    - There exist $u,v,x,y,z \in \Sigma^*$ such that
      1. $s=uvxyz$
      2. $|vy|>0$
      3. $|vxy| \leq p$
      4. For all $i \geq 0$,

         $u\ v^i\ x\ y^i\ z \in L$

Theorem 2.19: CFL $\subseteq$ CFPP: Proof Idea

Let:

A be a CFL and

G be a CFG generating A

s be a "very" long string in A

- s has a parse tree for its derivation
  - Parse tree is "very" long and contains a "long" path.
  - Pigeon-hole principle:
    - "Long" path contains repetition of some variable **R**.
    - Repetition of **R** allows substitution of first occurrence of **R**'s subtree where second occurrence of **R**'s subtree occurs.
  - Result is a legal parse tree for language A.
  - Due to substitution we can cut s into 5 pieces uvxyz.
    - Occurrences of v and y can be "pumped" to yield uv^ixy^iz.

Theorem 2.19: CFL $\subseteq$ CFPP: Proof Idea

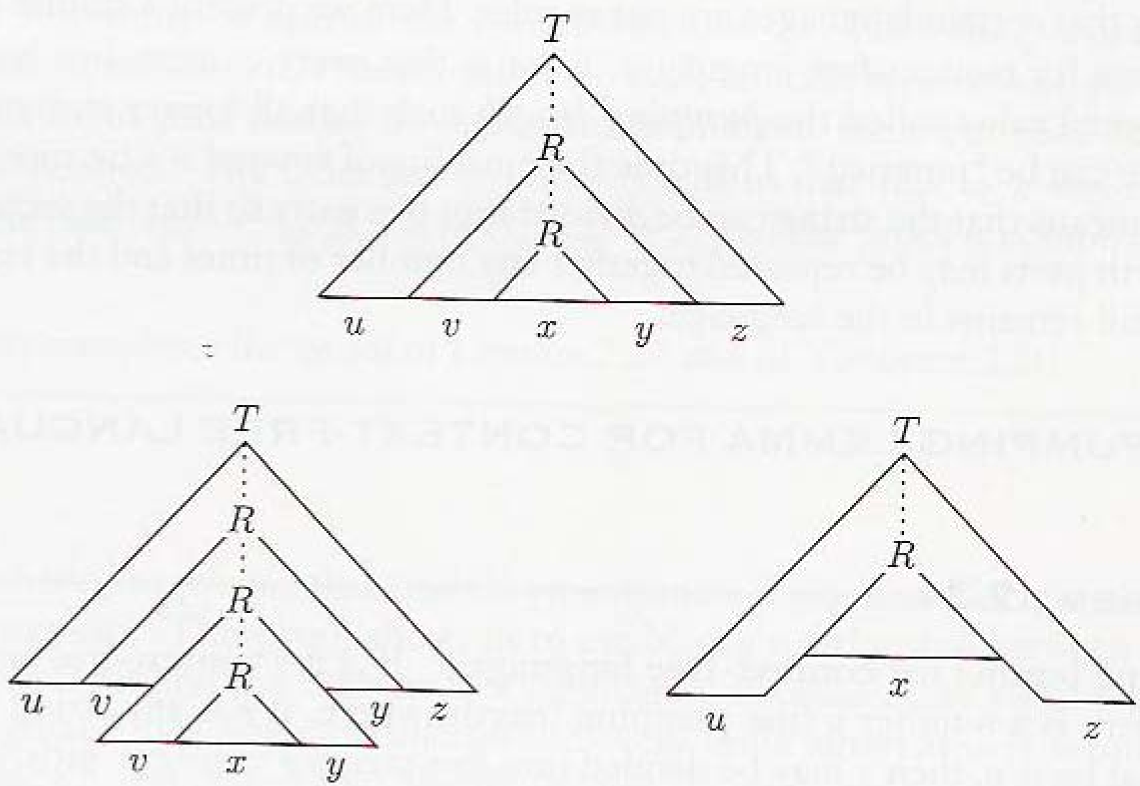


FIGURE 2.35
Surgery on parse trees

Source: Sipser textbook

Theorem 2.19: CFL $\subseteq$ CFPP

Proof. Suppose L$\in$CFL and let G=(V,Σ,R,S_0) be any CNF grammar that generates it.

- We set $p=2^{|V|+1}$.
- Now suppose s$\in$L where $|s| \geq p$. We must show how to produce u,v,x,y,z etc.
- Since $|s| \geq 2^{|V|+1} > 2^{|V|}$, we can apply the length fact to conclude that minheight(s)> |V|. But there are only |V| variables in the grammar. So looking at the parse tree for |s|, some variable **R** must be used more than once.
  - For convenience later, pick **R** to be a variable that repeats on the bottom |V| +1 internal nodes (corresponding to variables) of that path of the tree.

CFL $\subseteq$ CFPP continued

- We know that $S_0 \Rightarrow^* s$ and that **R** appears within this derivation twice
- So let u,v,x,y,z be strings satisfying
  - uvxyz=s
  - $S_0 \Rightarrow^* u\mathbf{R}z$ (first appearance)
  - $\mathbf{R} \Rightarrow^* v\mathbf{R}y$ (second appearance)
  - $\mathbf{R} \Rightarrow^* x$ (then turning into x)
- So $S_0 \Rightarrow^* u\mathbf{R}z \Rightarrow^* uv\mathbf{R}yz \Rightarrow^* uvxyz = s$ (we knew that $S_0 \Rightarrow^* s$ already)
- But the grammar is *context free*, so we can apply any of the **R** substitutions at any point
- Thus $S_0 \Rightarrow^* u\mathbf{R}z \Rightarrow^* u\mathbf{x}z = u\ v^0\ x\ y^0\ z$
- And $S_0 \Rightarrow^* uv\mathbf{R}yz \Rightarrow^* uv\mathbf{vRy}yz \Rightarrow^* uvvxyyz = uv^2xy^2z$ and so on. Hence, the pumping property holds.

CFL $\subseteq$ CFPP continued

- We still have to see the length constraints $|vy|>0$ and $|vxy| \leq p$ though.
- Recall $s = uvxyz$.
- Suppose that $|vy|=0$ (to get a contradiction). Then the parse tree has to include
  $S_0 \Rightarrow^* u\mathbf{R}z \Rightarrow^{\geq 1} u\mathbf{R}z \Rightarrow^* uxz$ (})
  (≥ 1 meaning "at least one substitution") This is because we know that **R** is actually repeated in the tree.
- But CNF rules *always* add to the string. The only exception is the optional rule $S_0 \rightarrow \varepsilon$, but we've already assumed that $|s|$ is long, so it isn't ε. Thus line (}) above can't be true, and hence $|vy|=0$ is impossible.

CFL $\subseteq$ CFPP continued

- We still have to see the length constraint $|vxy| \leq p$.
- We know that **R** repeats somewhere within the *bottom* $|V|+1$ internal nodes (representing variables) of the tree while producing the **vxy** part of s. Let h be the actual height of this subtree. Then
  - minheight(vxy) $\leq$ h $\leq$ $|V|+1$ (length of longest branch) $\Rightarrow$ $|vxy| \leq 2^{|V|+1} = p$ (by lemma (1.0 on slide 12)).
- Q.E.D.

Game theory (repeat)

The game has just four steps.

1. Your **opponent** picks $p \geq 0$
2. **You** pick $s \in L$ such that $|s| \geq p$
3. Your **opponent** chooses $u,v,x,y,z \in \Sigma^*$ such that $s=uvxyz$, $|vy|>0$, and $|vxy| \leq p$
4. **You** produce some $i \geq 0$ such that $uv^ixy^iz \notin L$

Example 2.36

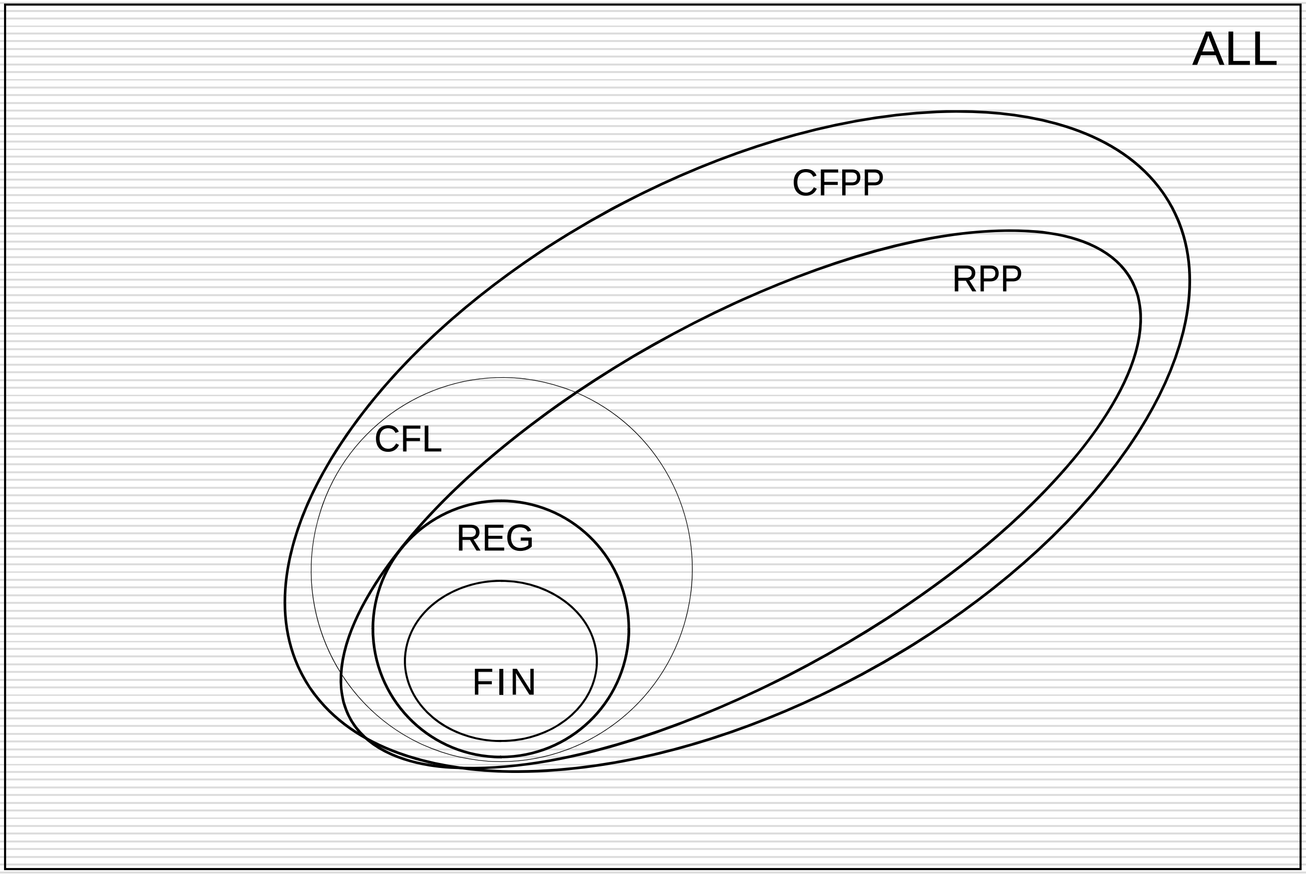
- $L = \{a^n b^n c^n \mid n \geq 0\}$ is not a CFL
- To see this: let opponent choose p, then we set s = $a^p b^p c^p$. Clearly $|s| > p$ and $s \in L$.
- So opponent breaks it up into u,v,x,y,z subject to the length constraints $|vy| > 0$ and $|vxy| \leq p$.
- We need to show that some i exists for which $uv^i xy^i z$ is not in L.
  - Note: the first character of v must be no more than p chars away from the last character of y, because $|vxy| \leq p$.
  - So in the string $uv^0 xy^0 z$, we have removed at least one char and at most p chars — but we have removed at most 2 *types* of characters: that is, some "a"s & "b"s, or some "b"s & "c"s. It's impossible to remove 3 types ("a"s & "b"s & "c"s) this way.
  - So the resulting string isn't in L. i=0 is our exponent.

Closure properties of CFL

- Reminder: closure properties can help us measure whether a computation model is reasonable or not
- CFL is closed under
  - Union, concatenation
    - Thus, exponentiation and *
- CFL is *not* closed under
  - Intersection
  - Complement
- Weak intersection:

  If $A \in$ CFL and $R \in$ **REG**, then $A \cap R \in$ CFL

Revised Picture



Each point is a language in this Venn diagram