Properties of Context-Free Languages

A Pumping Lemma for CFLs and CFL Closure Properties

Tuesday, November 9, 2010
Reading: Sipser 2.3, Stoughton 4.7, 4.10;

CS235 Languages and Automata

Department of Computer Science
Wellesley College

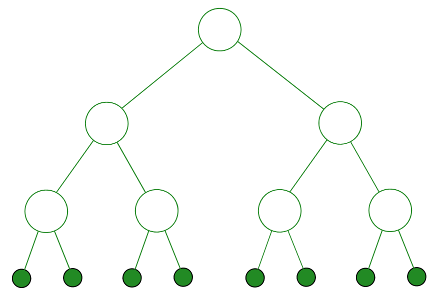
Overview of Today's Lecture

- Develop a pumping lemma for CFLs.
- Use the pumping lemma for CFLs to show that certain languages are not CFLs.
- Review closure properties for regular languages and discuss closure properties for context-free languages.

Height and # Leaves of Binary Trees

What is the maximum number of leaves in a binary tree of height h?

height	# leaves
0	
1	
2	
3	
h	



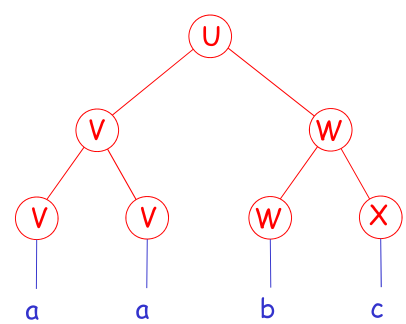
CFL Properties 26-3

Chomsky Normal Form Parse Trees

What is the maximum length of a string yielded by a Chomsky Normal Form parse tree of height h?

$U \rightarrow VW$	$V \rightarrow a$
$V \rightarrow VV$	$W \rightarrow b$
$W \rightarrow WX$	$X \rightarrow c$

height	length
1	
2	
3	
h	



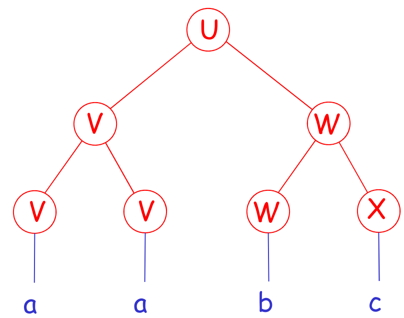
CFL Properties 26-4

Minimum Height of CNF Parse Tree

What is the minimum height of a CNF parse tree for a string of length 2^k?

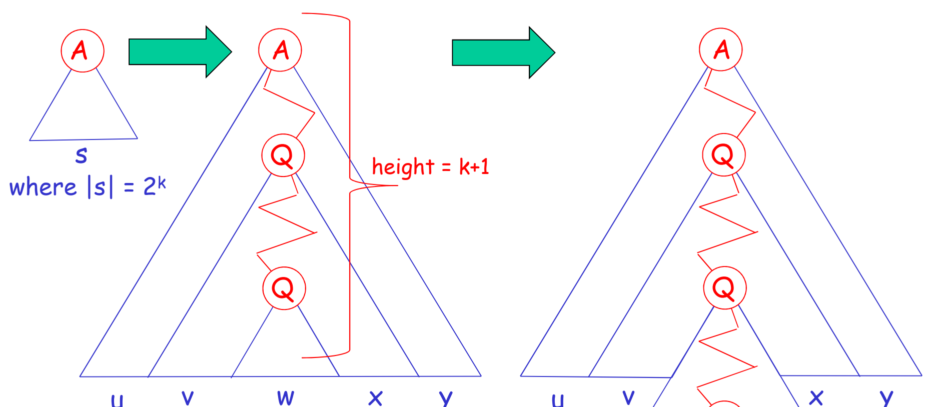
U → VW	V → a
V → VV	W → b
W → WX	X → c

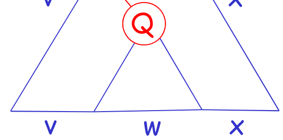
length	height
$2^0 = 1$	
$2^1 = 2$	
$2^2 = 4$	
2^k	



Idea Behind Pumping Lemma For CFLs

Assume a Chomsky Normal Form grammar with k variables and start variable *A*





Any sufficiently long string *s* in a CFL *L* must have a CNF parse tree containing a path from the root *A* repeating some variable Q and so can be decomposed into *uvwxy*, where at least one of v or *x* is nonempty and $uv^iwx^iy \in L$ for any $i \in Nat$.

The Pumping Lemma for CFLs

The Pumping Lemma for Context-Free Languages

If L is a context-free language, there is a number p (the pumping length) such that any string s with length $\geq p$ can be expressed as $uvwxy$, where:

1. $|vx| > 0$ (at least one of v or x is nonempty)
2. $|vwx| \leq p$
3. $uv^iwx^iy \in L$ for each $i \in Nat$.

CF-pumpability

Let's rephrase the CFL pumping lemma in terms of a property we'll call **CF-pumpability** (Lyn's term, not standard).

A language is CF-pumpable iff

There exists some p (the pumping length) s.t.
for all strings s with length $\geq p$
there exists a parsing of s into $uvwxy$, where

1. $|vx| > 0$ (at least one of v or x is nonempty)
2. $|vwx| \leq p$

s.t. for all $i \in Nat$, $uv^iwx^iy \in L$

Then the pumping lemma for CFLs can be rephrased as:

L is context-free $\Rightarrow$ L is CF-pumpable

Proving that a Language isn't Context Free

CFL Pumping Lemma: *L* is context-free ⇒ L is CF-pumpable

Contrapositive: *L* is not CF-pumpable ⇒ L is not context free

How to show *L* is not CF-pumpable?

For all p (the pumping length)
there exists some string s with length $\geq p$ *s.t.*
for all parsings of s into $uvwxy$, where:

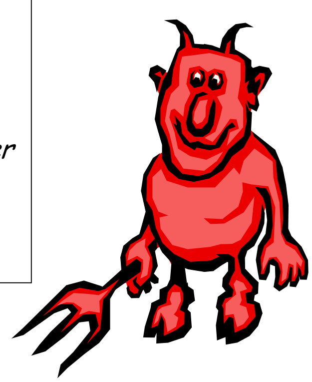
1. $|vx| > 0$ (at least one of v or x is nonempty)
2. $|vwx| \leq p$

there exists some $i \in Nat$ s.t. $uv^iwx^iy \notin L$

Games vs. Demons, Revisited

Suppose you want to prove that a language L is not context-free.
The proof (by contradiction) can be viewed as a game with a demon:

1. *You:* give the demon the language L.
2. *Demon:* gives you the pumping length p.
3. *You:* give the demon a string s with $|s| \geq p$.
4. *Demon:* divides s into $uvwxy$ such that $|vx| > 0$ and $|vwx| \leq p$.
   (Warning: typically more cases to consider than in nonregular language proofs!)
5. *You:* win if you can give the demon an i such that $uv^iwx^iy \notin L$.



$L_1 = 0^n1^n2^n$ is not a CFL

Viewed as game vs. a demon:

1. *You:* give the demon the language $L_1 = \{0^n1^n2^n \mid n \text{ in Nat}\}$
2. *Demon:* gives you the pumping length p
3. *You:* give the demon what string s with $|s| \geq p$?
4. *Demon:* divides s into $uvwxy$ such that $|vx| > 0$ and $|vwx| \leq p$.
   What are the possible divisions for your s?
5. *You:* win if you give the demon an i such that $uv^iwx^iy \notin L_1$.

$L_2 = \{0^j1^k2^n \mid j \leq k \leq n\}$ is not a CFL

Viewed as game vs. a demon:

1. *You:* give the demon the language $L_2 = \{0^j1^k2^n \mid j \leq k \leq n\}$
2. *Demon:* gives you the pumping length p
3. *You:* give the demon what string s with $|s| \geq p$?
4. *Demon:* divides s into $uvwxy$ such that $|vx| > 0$ and $|vwx| \leq p$.
   What are the possible divisions for your s?
5. *You:* win if you give the demon an i such that $uv^iwx^iy \notin L_2$.

L_{ww} = {ww | w ∈ {0,1}*} is not a CFL

Viewed as game vs. a demon:

1. *You:* give the demon the language L_{ww} = {ww | w in {0,1}*}
2. *Demon:* gives you the pumping length *p*
3. *You:* give the demon what string *s* with |*s*| ≥ *p*?
   (Hint: 0^p10^p1 doesn't work!)
4. *Demon:* divides *s* into *uvwxy* such that |*vx*| > 0 and |*vwx*| ≤ *p*.
   What are the possible divisions for your s?
5. *You:* win if you give the demon an *i* such that $uv^iwx^iy \notin L_{ww}$.

$\overline{L_{ww}}$ *is* a CFL!

Surprise: L_{ww} = {ww | w ∈ {0,1}*} isn't a CFL, but its complement is!

Why? Can construct a CFG that accepts $\overline{L_{ww}}$.

$\overline{L_{ww}}$ = {s | s ∈ {0,1}* isn't of the form ww for some w ∈ {0,1}*}

- Which odd-length strings are in $\overline{L_{ww}}$? Write a CFG generating them.

- Which even-length strings are in $\overline{L_{ww}}$? Write a CFG generating them.

Review: Closure Properties for Regular Languages

Suppose L, L_1, and L_2 are regular languages.
Then all the following are regular languages:

- L_1L_2
- L^*
- $\overline{L}$
- $L_1 \cup L_2$
- $L_1 \cap L_2$
- $L_1 - L_2$
- L^R

Closure Properties for CFLs

Suppose L, L_1, and L_2 are CFLs.
Then all the following are CFLs:

- L_1L_2
- L_1^*
- $L_1 \cup L_2$
- L_1^R

How can we prove these closure properties?
(Suppose A generates L_1 and B generates L_2.
How to generate results of operations?)

More Forlan Gram Bindings

```
val union : gram * gram -> gram      (* union of two grammars *)
val concat : gram * gram -> gram     (* concatentation of two grammars *)
val closure : gram -> gram           (* Kleene closure of two gramars *)
val rev : gram -> gram               (* reversal of two grammars *)
```

```
- Gram.output ("", L1gram);
{variables}
A, B, S
{start variable}
S
{productions}
A -> % | 0A1; B -> % | 1B0; S -> AB
val it = () : unit

- Gram.output ("", Gram.rev L1gram);
{variables}
A, B, S
{start variable}
S
{productions}
A -> % | 1A0; B -> % | 0B1; S -> BA
val it = () : unit
```

CFL Properties 26-17

What about Complement & Intersection?

CFLs are not closed under intersection.

Counterexample:

Both $0^n1^n2^m$ and $0^m1^n2^n$ are CFLs,,
but $0^n1^n2^m \cap 0^n1^n2^m = 0^n1^n2^n$ is not.

CFLs are not closed under complement.

Counterexample:

Let L_{ww} = {ww | w in {0,1}*}.

We've seen $\overline{L_{ww}}$ is a CFL but L_{ww} is not.

Can also prove this by contradiction: use deMorgan to show that CFLs closed under complement implies CFL closed under intersection.

CFLs are not closed under difference.

Counterexample: {0,1}* and $\overline{L_{ww}}$ are CFLs, but not L_{ww} = {0,1}* - $\overline{L_{ww}}$

CFL Properties 26-18

Intersecting a CFL with a Regular Language

CFLs **are** closed under intersection with a regular language:

If L is a CFL and R is a regular language, then $L \cap R$ ***is*** *a CFL*

We won't prove this yet – need our next topic (pushdown automata) to do this.

Examples:

Let L_{0eq1} = {w | w in {0,1}* contains equal # of 0s & 1s}

- $L_{0eq1} \cap$ 0*1* =
- $L_{0eq1} \cap$ 0*1*0* =
- $\underline{L_{ww}} \cap$ 0*110* =
- Use this property to prove that the following language is not context free: {w | w in {0,1,2}* and w contains equal numbers of 0s, 1s, and 2s}.