Properties of Context-Free Languages

An easy way to prove a bunch of properties of Context-Free languages is through the idea of a *substitution*. Let Σ be a finite alphabet and suppose that for each letter a in Σ we have a language S(a). If $w=a_1...a_n$ is a string in Σ^* we can say that S(w) is the concatenation $S(a_1)...S(a_n)$. If L is a language over Σ we say that $S(L) = \bigcup_{w \in L} S(w)$

For example, if we let $\Sigma=\{0,1\}$ and $S(0)=\{a^n b^n \mid n>= 1\}$ and $S(1) = \{a^n \mid n>=1\}$ then $S(001) = \{a^n b^n a^m b^m a^k \mid n,m,k >= 1\}$

Theorem: If $\mathcal{L}$ is a context-free language over Σ and S(a) is context-free for each a in Σ, then S($\mathcal{L}$) is context-free.
Proof: Start with the grammars for each S(a) and rewrite them so they have no nonterminal symbols in common. Take a Chomsky Normal Form grammar for $\mathcal{L}$ and rewrite it so it has no nonterminal symbols in common with any of the S(a) grammars. Each grammar rule for $\mathcal{L}$ has either the form A => BC or A => a. Replace each A => a rule by A => Start(a), where Start(a) is the start symbol for the S(a) grammar. This gives a context free grammar for S($\mathcal{L}$). (Two simple inductions show that this grammar derives w if and only if w is in S($\mathcal{L}$).

Theorem: If languages $\mathcal{L}_1$ and $\mathcal{L}_2$ are context-free then so are $\mathcal{L}_1 \cup \mathcal{L}_2$, $\mathcal{L}_1\mathcal{L}_2$ and $(\mathcal{L}_1)^*$.

Proof: Let Σ be {0,1}, let S(0)=$\mathcal{L}_1$ and let S(1)=$\mathcal{L}_2$. Then

a) {0,1} is context-free, and S({0,1}) = $\mathcal{L}_1 \cup \mathcal{L}_2$.
b) {01}) is context-free, and S({01}) = $\mathcal{L}_1\mathcal{L}_2$
c) 0* is context-free and S(0*) = $(\mathcal{L}_1)^*$.

However, note that context-free languages are not closed under intersection.

Example: Let $\mathcal{L}_1=\{0^n1^n2^j \mid n,j >= 0\}$ and let $\mathcal{L}_2 =\{0^k1^m2^m \mid k,m >= 0\}$
These are both context-free languages but $\mathcal{L}_1 \cap \mathcal{L}_2 = \{0^n1^n2^n \mid n >= 0\}$ and this is not context-free.

Note that this tells us that complements and differences of context-free languages are not necessarily context-free, for if they were intersections would also be context-free.

Theorem: If $\mathcal{L}$ is context-free and $\mathcal{R}$ is regular, then $\mathcal{L} \cap \mathcal{R}$ is context-free.

Proof: Start with a PDA that accepts $\mathcal{L}$ by final state and a DFA that accepts $\mathcal{R}$. Make a new PDA whose states are pairs of states from $\mathcal{L}$ and $\mathcal{R}$. If $\mathcal{L}$ has transition $\delta(q,a,X)=(q',y)$ and $\mathcal{R}$ has transition $\delta(r,a)=r'$ then make transition for the new PDA $\delta((q,r),a,X)=((q',r'),Y)$. The final states of the new PDA are $\{(q,r) \mid q \text{ is final for } \mathcal{L} \text{ and } r \text{ is final for } \mathcal{R}\}$ This new PDA accepts string w if and only if w is accepted by both $\mathcal{L}$ and $\mathcal{R}$.

Why can't we do this with 2 PDAs?

Theorem: If $\mathcal{L}$ is context-free and $\mathcal{R}$ is regular then $\mathcal{L}$-$\mathcal{R}$ is context-free.

Proof: $\mathcal{L}-\mathcal{R} = \mathcal{L} \cap \mathcal{R}^c$ and $\mathcal{R}^c$ is regular.

Theorem: If $\mathcal{L}$ is context-free then $\mathcal{L}^{rev}$ is also context-free.

Proof: Start with a Chomsky Normal Form grammar for $\mathcal{L}$. Replace any rule A => BC with the rule A => CB. An induction on the length of derivations shows that this is a grammar for $\mathcal{L}^{rev}$.

See example next slide

For example, a grammar for $\{a^n b^m | n>0, m >= 0\}$ is

A => AB | AA | a

B => BB | b

The grammar

A => BA | AA | a

B => BB | b

creates the language $\{b^m a^a | n>0, m >= 0\}$

Here is an example of a language that is pumpable but not context-free. This is just a variation of the language that was pumpable in the regular sense but not regular:

$\mathcal{L} = \{a^i b^j c^k d^l \mid i,j,k,l >= 0$ and if $i=1$ then $j=k=l\}$

First, if $\mathcal{L}$ was context-free then, since $ab^*c^*d^*$ represents a regular language, the intersection $\mathcal{L} \cap ab^*c^*d^* = \{ab^j c^j d^j \mid j >= 0\}$ would also have to be context-free, which it clearly isn't.

(continued next slide)

Second, note that $\mathcal{L}$ is the union of three languages:

$\mathcal{L}_0=\{b^j c^k d^l \mid j,k,l >= 0\} = b^*c^*d^*$

$\mathcal{L}_1 =\{a b^j c^j d^j \mid j,k,l >= 0\}$

$\mathcal{L}_2=\{a^i b^j c^k d^l \mid i >=2, j,k,l >= 0\} = a^2a^*b^*c^*d^*$

Note that $\mathcal{L}_0$ and $\mathcal{L}_2$ are regular, so they are certainly pumpable. $\mathcal{L}_1$ is not pumpable in itself, but if we take any string z in $\mathcal{L}_1$, such as $z= ab^jc^jd^j$ we can let $u=\varepsilon$, $v=a$, $w=x=\varepsilon$, $y=b^jc^jd^j$. Then $z=uvwxy$ and $uv^nwx^ny = a^nb^jc^jd^j$.is an elements of $\mathcal{L}$ for every n. So every long string in $\mathcal{L}$ can be pumped; the pumping constant for $\mathcal{L}$ is the longer of the constants for $\mathcal{L}_0$ and $\mathcal{L}_2$.

Decision Algorithms for Context-Free Languages:

We can determine if a given string w is in a given context-free language: convert the grammar to CNF and generate all possible parse trees of height $|w|-1$. Since a binary tree of height n has at least n+1 leaves, this will find all strings in the language of length $|w|$ or less.

We can determine if a context-free language is empty or infinite; these are homework questions.

Most other questions regarding context-free languages are undecidable, including:

- Are two context-free languages the same?
- Is the intersection of two context-free languages empty?
- Is a context-free language Σ^*?
- Is a given grammar ambiguous?
- Is a given language inherently ambiguous?