Let $L$ be a language defined by a "phrase specification." Eg:

$L = \{ x \in \{a,b\}^* : \#a(x) = \#b(x) \} \setminus \emptyset.$

Def. (not explicit in text): A CFG $G$ is **sound** for the specification if $L(G) \subseteq L$. I.e., $G$ does not generate any string $x \not\in L$. (Logic term: "complete")

$G$ is **comprehensive** if $L \subseteq L(G)$. "$G$ has no false positives."

$G$ fails to be comprehensive if there is a string $w$ in $L \setminus L(G)$. Example: $G_0 = S \rightarrow aSb | bSa | \epsilon$

- **Claim** ("by inspection") $G_0$ is sound: $L(G_0) \subseteq L$

- Is $G_0$ comprehensive? Try $x = \text{abba}$. Then $x \not\in L(G_0)$ ("by trial and error", or -?) So $G_0$ is not comprehensive.

More generally, $G_0$ obeys a "further restriction" on the spec:

$L(G_0) \subseteq L' = \{ x \in L : x \text{ does not begin or end with the same letter} \}$

Is $G_0$ comprehensive for $L'$? No: $y = \text{abbbab} \not\in L' \setminus L(G_0)$. Hence certainly $G_0$ is not comprehensive for the original $L$.

How about adding a rule? $G_1 = S \rightarrow aSb | bSa | SS | \epsilon$

Say $S \rightarrow SS$ Then $S \rightarrow SS \Rightarrow aSbS \Rightarrow abS \Rightarrow abbab \Rightarrow abba$
Thus: \( G_1 \) is not sound for \( L' \), so it has a chance of being comprehensive for \( (L' \) and \( L \).

\[ G_1 = S \rightarrow \varepsilon | aSc | bSa | SS \]

\( L = \{ x : \#a(x) = \#b(x) \} \). First ask, is \( G_1 \) sound for \( L \)? Yes ("because: if \( SS \Rightarrow \varepsilon \) the fact that the concatenation of two strings with equal \( a \)'s and \( b \)'s has equal \( a \)'s and \( b \)'s comes into play.

"Structural induction proof script" (for soundness proofs):

**Theorem:** \( L(G_1) \subseteq L \).

1. For every variable \( A \), define a property \( P_A \)
   - Here there is only one variable \( S \), so use the spec of \( L \) as \( P_S \)
   - Always need: \( x \) obeys \( P_S \Rightarrow x \in L \).

2. For each rule \( A \rightarrow X \), show that if all variables \( B, C, D \in X \) derive substrings \( y, z, w \) that obey their properties \( P_B, P_C, P_D \) etc., then the resulting string \( X \) must obey \( P_A \).

**1) \( P_S = \) "Every \( x \) that I derive has \( \#a(x) = \#b(x) \)"**

**2) \( S \rightarrow \varepsilon : \) Suppose \( S \Rightarrow \varepsilon \) using this rule first (utrf).
   - Then \( x = \varepsilon \) ("duh!") and \( \varepsilon \in L \). So \( P_S \) is upheld on LHS.
**3) \( S \rightarrow aSc : \) Suppose \( S \Rightarrow X \) utrf. Then \( X = aYb \) where \( S \Rightarrow Y \).
   - By 2H, \( P_S \) on RHS, \( \#a(Y) = \#b(Y) \). Hence \( \#a(X) = 1 + \#a(Y) = 1 + \#b(Y) \) (by 2H) = \( \#b(\varepsilon) \). So \( \#a(\varepsilon) = \#b(\varepsilon) \). **P_S** on LHS.
   - \( S \rightarrow bSa : \) OK to say "Similar to last rule" and move on. (\( \therefore L(G_1) \subseteq L \))
To finish with $G_1$, analyze the rule $S \rightarrow SS$:  
Suppose $S \Rightarrow^* x \ uTrf$. Then $x = \gamma z$ where $S \Rightarrow^* y \ u z \Rightarrow^* z$  
By IH $P_z$ on RHS (twice) $#a(y) = #b(y) \land #a(z) = #b(z)$.  
Thus $#a(x) = #a(y) + #a(z)$ by $x = \gamma z$  
= $#b(y) + #b(z)$ by IH $P_z$ (twice)  
= $#b(x)$ again by $x = \gamma z$  
$\therefore P_z$ on LHS holds in this case too.  
Since $P_z$ on LHS is upheld by each rule, $L(G_1) \subseteq L$ follows by "SF".

A Multi-Variable Example: $G_2$: 
$S \rightarrow e \mid AB \mid BA$  
$A \rightarrow a \mid aS \mid BAA$  
$B \rightarrow b \mid bS \mid ABB$.

Same L, Same $P_z$.  
What to choose for $P_A$ & $P_B$? 

Suggestion: $P_A = \"Every x I derive has $#a(x) = #b(x) + 1.\"$  
$P_B = \"Every z I derive has $#b(z) = #a(z) + 1.\"$  

(1) $P_A$: "Every $y$ I derive has $#a(y) = #b(y) + 1."$  
(2) $S \rightarrow e$ OK  
As before.

$S \rightarrow AB$: Suppose $S \Rightarrow^* x \ uTrf$. Then $x = \gamma z$ where $A \Rightarrow^* y$  
and $z \Rightarrow^* z$. By IH $P_a$ on RHS, $#a(y) = #b(y) + 1$, and  
by IH $P_a$ on RHS, $#b(z) = #a(z) + 1$. Hence  
$#a(x) = #a(y) + #a(z)$ by $x = \gamma z$  
$= #b(y) + 1 + #a(z)$ by IH $P_A$  
$= #b(y) + 1 + #b(z) - 1 = #b(y) + #b(z) = #b(x)$.  
$\therefore P_z$ is upheld on LHS.  

$S \rightarrow BA$: OK to say "Similar":  
Is it OK to stop here?
No: We also need to show the rules for $A \& B$ uphold $PA \& PB$!

$A \to a$: Immediate since $\#a(a) = 1 = 1 + 0 = 1 + \#B(a)$.

$A \to as$: Suppose $A \Rightarrow w \text{ utrf. Then } w = ax$ where $s \Rightarrow x$.

$B + 2 \equiv B$ on $\text{RHS, } \#a(x) = \#b(x)$.

Hence $\#a(w) = 1 + \#a(x)$

So $\#a(w) = 1 + \#b(w)$ (by $B$ on $\text{RHS}$)

$\#a(w) = 1 + \#b(w)$ (by $x = aw$) which uphold $B$ on $\text{LHS}$.

$A \to BAA$: Suppose $A \Rightarrow w \text{ utrf. Then } w = xyz$ where

$B \Rightarrow x$ \hspace{1cm} $B \equiv B$ and $\#b(x) = \#a(x) + 1$ \hspace{1cm} Adds $\Rightarrow PA$ on $\text{LHS}$

$A \Rightarrow y$ \hspace{1cm} $PA$ (twice) \hspace{1cm} $\#a(\cdot) = \#b(\cdot) + 1$ \hspace{1cm} up to \hspace{1cm} for $w$.

$A \Rightarrow z$ \hspace{1cm} on $\text{RHS: } \#a(z) = \#b(z) + 1$ \hspace{1cm} $\#a(w) = \#b(w) + 1$.

We have to do the rules for $B$ too, but here they are "similar."

$\therefore B$, $PA$, $PB$ are upheld by all rules, $\therefore L(G_2) \subseteq L$.

Is $G_2$ comprehensive? $\Rightarrow$ Thy.

Added Note (spoken early in the lecture): The concepts "sound" and "comprehensive" apply to more general kinds of string rewriting systems than CFG. The granddaddy of them all is the notion of a proof system (taught in CS199).

A proof system has "items" that are well-formed formulas (WFFs) over some logical and/or arithmetical syntax (which itself can be defined by a CFG/ANN grammar) and ("meta-") rules typified by Modus Ponens: if $X \land X \to Y$ are theorems then so is $Y$.

We begin with an axiom set $\mathcal{A}$; then $L(F)$ is the set of theorems. The language $L$, often called $V$ for veritas (truth in Latin), is the set of WFFs that are objectively true. $F$ is sound if $L(F) \subseteq V$. Gödel's Incompleteness Theorem is that for $F = \text{"set theory"}, L(F) \not\subseteq V$.