RE-minder: PS 5 due Thursday. Exam Key posted backdoor.

Let L be a language defined by a logical "prose specification". E.g.:

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

Def'ns (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 s.t. $x \notin L$. (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: $G_0 = S \rightarrow aSb \mid bSa \mid \epsilon$

- Claim ("by inspection") G_0 is sound: $L(G_0) \subseteq L$
- Is G_0 comprehensive? Try $x = abba$. Then $x \notin L(G_0)$ ("by trial and error, or—?) So G_0 is not comprehensive.

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

$L(G_0) \subseteq L' = \{x \in L : x$ does not begin & end with the same letter$\}$. $\#a(x) = \#b(x)$ &

Is G_0 comprehensive for L'? sound yes No: $y = ab|abba|ab \in L' \setminus L(G_0)$.

Hence certainly G_0 is not comprehensive for the original L.

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

Say $S \rightarrow SS$ Then $S \Rightarrow SS \Rightarrow aSbS \Rightarrow abS \Rightarrow abbSa \Rightarrow abba$

Thus: G_1 is not sound for L', so it has a chance of being comprehensive for (L' and) L.

$G_1 = S \to \epsilon \mid aSb \mid bSa \mid SS$

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

For Thu: Is G_1 comprehensive for the (original) language L?

"Structural Induction Proof Script" [for Soundness proofs]:

$\boxed{\text{Theorem: } L(G_1) \subseteq L.}$

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 (Might need a stronger P'_S.)

Always need: x obeys $P_S \Rightarrow x \in L$.

② For each rule $A \to X$, 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 $\#a(x) = \#b(x)$" $\left(S \overset{*}{\Rightarrow} x \Rightarrow x \in L\right)$

② $S \to \epsilon$: Suppose $S \overset{*}{\Rightarrow} x$ using this production rule first (utpf) (utrf). Then $x = \epsilon$ ("duh!") And $\epsilon \in L$. So P_S is upheld on LHS.

$S \to aSb$: Suppose $S \overset{*}{\Rightarrow} x$ utrf. Then $x = ayb$ where $S \overset{*}{\Rightarrow} y$. By IH P_S on RHS, $\#a(y) = \#b(y)$. Hence $\#a(x) = 1 + \#a(y)$ $= 1 + \#b(y)$ (by IH) $= \#b(x)$. So $\#a(x) = \#b(x)$. $\therefore P_S$ on LHS.

$S \to bSa$: OK to say "Similar to last rule" and move on. ($\therefore L(G_0) \subseteq L$)

③

To finish with G_1, analyze the rule $\underline{S \to SS}$:

Suppose $S \Rightarrow^* x$ utrf. Then $x =: yz$ where $S \Rightarrow^* y$ & $S \Rightarrow^* z$

By IH P_S on RHS (twice) $\#a(y) = \#b(y) \wedge \#a(z) = \#b(z)$.

Thus $\#a(x) = \#a(y) + \#a(z)$ by $x = yz$

$= \#b(y) + \#b(z)$ by IH P_S (twice)

$= \#b(x)$ again by $x = yz$

$\therefore P_S$ on LHS holds in this case too.

Since P_S on LHS is upheld by each rule, $L(G_1) \subseteq L$ follows by "$\underline{SI}$".

<u>A Multi-Variable Example</u>: G_2: $S \to \varepsilon \mid AB \mid BA \mid$

$A \to a \mid aS \mid BAA$

$B \to b \mid bS \mid ABB$.

Same L, same P_S.

What to choose for P_A & P_B?

Suggestion: $P_A \equiv$ "Every x I derive has $\#a(x) = 1$ $\wedge \#b(x) = 0$"

$P_B \equiv$ "Every x I derive has $\#b(x) = 1$. $\wedge \#a(x) = 0$"

OK for S but unsound for A & B rules.

① P_A: "Every y I derive has $\#a(y) = \#b(y) + 1$."

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

② $\underline{S \to \varepsilon}$ OK as before.

$\underline{S \to AB}$: Suppose $S \Rightarrow^* x$ utrf. Then $x =: yz$ where $A \Rightarrow^* y$

and $B \Rightarrow^* z$. By IH P_A on RHS, $\#a(y) = \#b(y) + 1$, and by IH P_B on RHS, $\#b(z) = \#a(z) + 1$. Hence $\#a(z) = \#b(z) - 1$

$\#a(x) = \#a(y) + \#a(z)$ by $x = yz$

$= \#b(y) + 1 + \#a(z)$ by IH P_A

$= \#b(y) + 1 + \#b(z) - 1 = \#b(y) + \#b(z) = \#b(x)$.

$\therefore P_S$ is upheld on LHS.

$\underline{S \to BA}$: OK to say "Similar".

Is it OK to stop here?

<u>No</u>: We also need to show the rules for A & B uphold P_A & P_B!

<u>$A \to a$</u>: Immediate since $\#a(\text{"a"}) = 1 = 1 + 0 = 1 + \#b(\text{"a"})$.

<u>$A \to aS$</u>: Suppose $A \Rightarrow^* w$ utrf. Then $w = ax$ where $S \Rightarrow^* x$.
By IH P_S on RHS, $\#a(x) = \#b(x)$. Hence $\#a(w) = 1 + \#a(x)$
so $\#a(w) = 1 + \#b(x)$ (by P_S on RHS)
so <u>$\#a(w) = 1 + \#b(w)$ (by $x = aw$)</u> which upholds P_A on LHS.

<u>$A \to BAA$</u>: Suppose $A \Rightarrow^* w$ utrf. Then $w =: xyz$ where

$B \Rightarrow^* x$ By P_B and $\#b(x) = \#a(x)+1$ Adds $\therefore$ P_A on LHS
$A \Rightarrow^* y$ P_A (twice) $\#a(y) = \#b(y)+1$ up to for w.
$A \Rightarrow^* z$. on RHS: $\#a(z) = \#b(z)+1$ $\#a(w) = \#b(w)+1$.

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

$\therefore$ P_S, P_A, P_B are upheld by all rules, $\therefore$ $L(G_2) \subseteq L$. ☒

¿ Is G_2 comprehensive ? → <u>Thu</u>.

<u>Added Note</u> (spoken early in the lecture): The concepts "sound" and "comprehensive" (or historically, "complete") apply to more general kinds of <u>string rewriting systems</u> than CFGs. The granddaddy of them all is the notion of a <u>proof system</u> (taught in CSE191!). A proof system F has "items" that are well-formed formulas (WFFs) over some logical and/or arithmetical syntax (which itself can be defined by a CFG/BNF grammar) and ("meta-") <u>rules</u> typified by <u>Modus Ponens</u>: if X and X→Y are theorems then so is Y. We begin with an axiom set A_F; then <u>L(F)</u> is the set of theorems. The language L, often called V for <u>veritas</u> (truth in Latin), is the set of WFFs that are objectively true. F is <u>sound</u> if $L(F) \subseteq V$. <u>Gödel's Incompleteness Theorem</u> is that for F = "set theory," "arithmetic," $L(F) \subsetneq V$.