CSE 396 | Lecture Thu 3/28 | Spring 2019

TopHat 3557

G: $S \to SS \mid bA \mid aB \mid \epsilon$
$A \to aS \mid bAA$
$B \to bS \mid aBB$.

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

Proof of $L(G) \subseteq E$ by "Structural Induction":
Define the properties

Note $S \to \epsilon$ is the basis for all three variables, but will not need any special treatment.

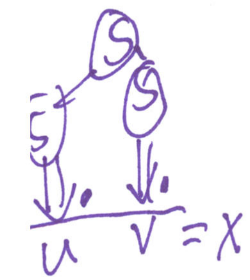
$P_S \equiv$ "Every x such that (that I derive) $S \Rightarrow^* x$ has $\#a(x) = \#b(x)$," (belongs to E.)

$P_A \equiv$ "Every y s.t. $A \Rightarrow^* y$ has $\#a(y) = \#b(y) + 1$. one more a than b.

$P_B \equiv$ "Every z s.t. $B \Rightarrow^* z$ has $\#b(z) = \#a(z) + 1$. one more b than a.

Go through all rules for each variable and show that if the variables on the RHS derive strings that obey their properties, then the resulting string obeys ("upholds") the property of the variable on the LHS.

$S \to SS$: Suppose $S \Rightarrow^* x$ using this rule first. Then $x =: uv$ where $S \Rightarrow^* u$ and $S \Rightarrow^* v$. By IH P_S on RHS (twice), we have



$\#a(u) = \#b(u)$ and $\#a(v) = \#b(v)$. $\therefore$ $\#a(x) = \#b(x)$, since $x = uv$, which upholds P_S on LHS for x.

$S \to bA$: Suppose $S \Rightarrow^*$ utrf. Then $x =: by$ where $A \Rightarrow^* y$. By IH P_A on RHS, y has 1 more a than b. The leading 'b' in x thus equalizes the count: $\#a(x) = \#b(x)$, and this upholds P_S on LHS.

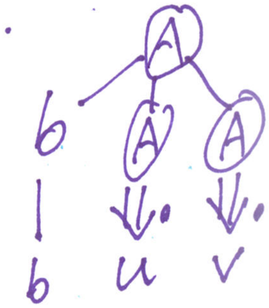
$S \to aB$: "Similar to $S \to bA$ case." This finishes S. Are we done?

$S \to \epsilon$: Suppose $S \Rightarrow^* x$ utrf. Then $x = \epsilon$. And $\#a(\epsilon) = 0 = \#b(\epsilon)$ and this upholds P_S on LHS. No...

$S \to SS \mid aB \mid bA \mid \epsilon$
$A \to aS \mid bAA$
$B \to bS \mid aBB$

<u>$A \to aS$</u>: Suppose $A \Rightarrow^* y$ utrf. Then $y =: \underline{a}x$ where $S \Rightarrow^* x$. By IH <u>P_S</u> on <u>RHS</u> $\#a(x) = \#b(x)$. ∴ y has one more a than b. ∴ P_A on LHS is upheld.

<u>$A \to bAA$</u>: Suppose $A \Rightarrow^* x$ utrf. Then $x =: buv$ where u and v each have one more 'a' than 'b'. The leading b thus brings the count in x down from two more a's in the uv part to one more 'a' overall. ∴ <u>P_A on LHS.</u>



<u>$B \to bS$</u>, <u>$B \to aBB$</u>: "Similar to the last two cases." ∴ <u>$L(G) \subseteq E$ by SR</u>

How about $E \subseteq L(G)$? Since G is sound, if G is comprehensive then we get $L(G) = E$.

<u>Sketch</u>: Idea is induction on the length of strings, also using the languages $E_1 = \{x : \#a(x) = \#b(x) + 1\}$ and $E_2 = \{x : \#a(x) = \#b(x) - 1\}$

Think of this as a task of parsing when you're in "S mode", "A mode", or "B mode". These "modes" stand for routines that build parse trees from the given variable.

Example: $x = abbbabaa$

$S \to aB \mid bA \mid \epsilon$ $B \to bS$
$A \to aS \mid bAA$ $B \to aBB$.

$S \Rightarrow aB$. $x = ay$ with $y = bbbabaa$.

$\Downarrow^*$ y

∴ $S \Rightarrow^* x$.

Key point: $n' = |y|$ is $< n = |x|$. So we can use an IH for E_2 saying that for all y of length $< n$, <u>if</u> $\#b(y) = \#a(y) + 1$ <u>then</u> $B \Rightarrow^* y$. "B is comprehensive for E_2."

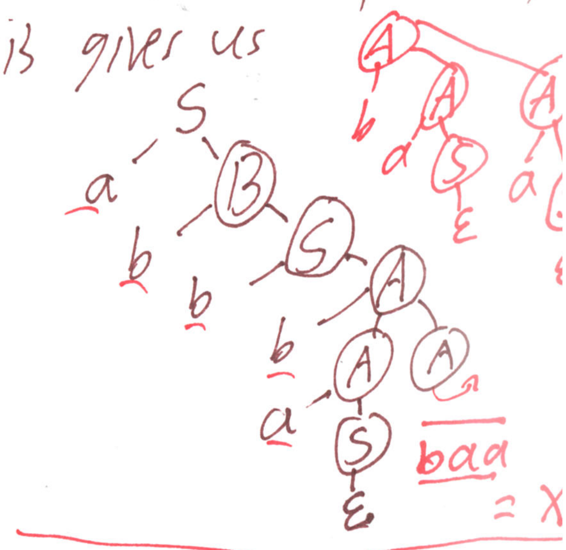
We also need to verify other variables recursively. So with $y = \underline{bbbabaa}$, we have $y \in E_2$ so in "B mode" we need to parse it from B. We can use $B \Rightarrow bS$ and go back to S mode on $x' = bbabaa$. Next we do $S \Rightarrow bA$ and we're in A-mode on $y' = babaa$. What now

This is OK because we can parse y' as $b \cdot \underline{a} \cdot \underline{baa}$. The parts $u = a$, $v = baa$ are both in E

Hence by induction hypothesis on comprehensiveness of A for E 1, we can derive both of them from A. This gives us

$$A \Rightarrow b\,A\,A, \quad A \Rightarrow^* u, \quad A \Rightarrow^* v, \quad b\,u\,v = y'.$$

Whole parse so far is:



The full proof that the other rules $B \Rightarrow aBB$ and $A \Rightarrow aS$ and $S \Rightarrow aB$ make all three variables comprehensive is similar and never needs the rule $S \to SS$. So we can delete it from G.

$$G' = \begin{array}{l} S \to aB \mid bA \mid \varepsilon \\ A \to aS \mid bAA \\ B \to bS \mid aBB. \end{array}$$

Deleting the rule left G' still sound, and we showed G' is still comprehensive.

Because y' could have been broken as $b \cdot aba \cdot a$, G and G' are ambiguous!

Def^n: A grammar G is in ChNF (Chomsky normal form) if all of its rules have the form $A \to c$ ($c \in \Sigma$) or $A \to BC$ with $B, C \in V$, (either can be A itself or $C = B$ allowed)

The original def^n (as in the text) forbids B or $C = S$.

It also disallows $S \to \varepsilon$. But both can be tolerated by re-defining a new start symbol S_0 with rules $S_0 \to \varepsilon \mid$ any r.h.s. of S.

Hence in lecture, ChNF will allow S on right-hand sides.

$\boxed{\text{ADDED}}$: I have decided this time not to cover the proof that every CFG G can be converted to G' in ChNF (strict ChNF if $\varepsilon \notin L(G)$), except that the step of bypassing ε-rules will be covered where relevant in Ch. 4. The *fact* of the conversion will still be used to make the CFL Pumping Lemma (§2.3) easier to visualize on Tue