CSE 376
Top Hat 3557
Lecture Thu 3/28
Spring 2019

G: $S \Rightarrow SS | bAA | aBb | llB|
A \Rightarrow aS1bAA
B \Rightarrow bS1aBB

Target $E = \{ x \in \{0, 1\}^*: \#a(x) = \#b(x) \}$

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

$P_S$: "Every $x$ such that $S \Rightarrow^* x$ has $\#a(x) = \#b(x)$!"

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

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

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

$S \Rightarrow SS$: 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 $\#a(v) = \#b(v)$. since $x = uv$ $P_S$ on LHS for $x$.

$S \Rightarrow bA$: 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.

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

$S \Rightarrow e$: Suppose $S \Rightarrow^* utrf$. Then $x = e$. And $\#a(e) = 0 = \#b(e)$

and this upholds $P_S$ on LHS. No...
$S \rightarrow SS \mid aB \mid bA \mid e$

$A \rightarrow aS \mid bAA$

$B \rightarrow bS \mid aBB$

$A \rightarrow bAA$ Suppose $A \Rightarrow^* x$ utrf. Then $x = a^u b^v$ where $u$ and $v$ each have one more 'a' than 'b'. The leading b thus brings the count in x down from two to one in the uv part to one more 'a' overall. $\therefore P_A$ on LHS is upheld.

$B \rightarrow bS, B \rightarrow aBB$: Similar to the last two cases. $\therefore L(G) \subseteq E$ by SR.

How about $E \subseteq L(G)$? Since G is sound, $\vdash G$, G is comprehensive

Sketch: Idea is induction on the length of strings, also using the languages $E_1 = \{x : \#A(x) = \#B(x) + 1 \}$ and $E_2 = \{x : \#A(x) = 2 \#B(x) - 1 \}$

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

Example: $x = a b b b a b a a$

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

We also need B verify other variable recursively. So with $y = b b b a b a a$, 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 $x' = b b b a b a a$. Next we do $S \Rightarrow bA$ and we're in A mode on $y' = b a b a a$. This is OK because we can parse $y'$ as $b \cdot a \cdot b a a$. The pairs $u, a, v, b a a$ are both in
Hence by induction hypothesis on comprehensiveness of $A$ for $E$, we can derive both of them from $A$. This gives us:

$$A \Rightarrow b AA$$

Whole parse:

$$a - S - b - B - b$$

The full proof:

$$b U V = \gamma'$$

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 $S \Rightarrow SS$. So we can delete it from $G$.

$$G' = S \Rightarrow aB \mid bA$$

Deleting the rule left:

$$A \Rightarrow aS \mid bAA$$

$G'$ still sound and we:

$$B \Rightarrow bS \mid aBB$$

showed $G'$ is still comprehensive.

**Defn:** A grammar $G$ is in ChNF if all of its rules have the form $A \Rightarrow C (c \in \Sigma)$ or $A \Rightarrow BC$ with $B, C \in V$, (either can be $A$ itself or $C = B$ allowed)

The original defn (as in the text) forbids $D$ or $C = S$.

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

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

---

**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 $\varepsilon \notin \Sigma(L(G))$, except that the step of bypassing $\varepsilon$-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.