Lecture Tue Mar 31

$Diff(X,i) = \#'('[X_1..X_i] - \#')'[X_1..X_i]$ $0 \le i \le n$ (1) $n = |X|$

$Bal = \{X \in \{(,)\}^* : Diff(X,n) = 0 \wedge (\forall i : 0 \le i < n)\ Diff(X,i) \ge 0\}$.

G_1: $S_1 \to \varepsilon \mid S_1 S_1 \mid (S_1)$

G_2: $S_2 \to \varepsilon \mid (S_2) S_2$.

Prove $L(G_1) = L(G_2) = Bal$.

(Both grammars have $S \to \varepsilon$)

Note: G_1 simulates G_2 rule-wise via $S \Rightarrow SS \Rightarrow (S)S$.

Hence: Soundness of G_1 $\Rightarrow$ soundness of G_2. I.e. $L(G_1) \subseteq Bal \Rightarrow L(G_2) \subseteq Bal$.

But, G_2 does not simulate G_1 rule-wise: in G_2 you cannot get the so-called "sentential form" SS.

(So the "expanded" languages of the two grammars are not the same, but we don't care)

any derivable string in $(V \cup \Sigma)^*$.

Hence also: If G_2 is comprehensive, then G_1 is comprehensive. I.e.

$L(G_2) \supseteq Bal \Rightarrow L(G_1) \supseteq Bal$. Earlier we proved G_1 is sound, so we only need $L(G_2) \supseteq Bal$.

Proof by induction on the length n of strings:

Prove $\forall n\, P(n)$, where $P(n) \equiv$ for each $X \in \{(,)\}^n$, $X \in Bal \Rightarrow S_2 \Rightarrow^* X$.

G: $S \to \varepsilon \mid (S)S$

Basis $(n=0)$ $\varepsilon \in Bal$. $S \Rightarrow \varepsilon$ so that checks.

Another Basis $(n=1)$: The only strings $X \in \{(,)\}^1$ are $X =$ (and $X =$). Neither string belongs to Bal. So this holds by default – nothing to do.

Induction $(n \ge 2)$: Assume (IH) the statement $\boxed{(\forall m < n)\ P(m)}$.

Given n, goal: show $P(n)$. Let any $X \in \{(,)\}^n$ be given.

- If $X \notin Bal$ (as always happens when n is odd), there is nothing to do.
- Let $X \in Bal$. Our goal is to show $S \Rightarrow^* X$ in G_2. Consider how to parse X.

We know $Diff(X,i) = 0$ when $i = 0$ and when $i = n = |X|$, and $Diff(X,i) \ge 0$ for all $i \le n$.

$Diff$ 0 1 2 1 2 1 0

We want to isolate a "critical" value of i.

$X = (()())$: Parse as $S \Rightarrow (S)\$ \to \varepsilon$ $(i = n = 6)$

$X' = (((())()))$ $i = 6$ won't work for X': The parse $S \Rightarrow (S\ \ )\$ \to$ leaves $(((())C)$ & Bal.

"Mathematical Rhetoric" to locate the matching mate of the opening '(': ②

"Take i to be the least $i>0$ such that $Diff(X,i)=0$."

By $Diff(X,n)=0$ we know that the loop while($Diff[X,i]!=0$){i++;} must exit. [Side Remark: When we don't have such a bound, this kind of "Impredicative" math definition is highly controversial.]

Clearer rewrites: • Take i to be least such that $Diff(X,i)=0$ $(i>0)$.
• Take i_0 to be the least $i>0$ such that $Diff(X,i)=0$.

$X =$ (- - - - -) $i=0$ - - - $Diff(i)=0$ at $i=n$
$Diff(i)>0$. $Diff$

Then we can parse $X =: (Y)_i Z_n$ where $|Y| = i-2$ and:

• $Diff[-1, |Y|) = Diff(X, i-1) - 1 = 1 - 1 = 0$
($Diff(X,i-1)=1$: taking away the opening '(' ; again, taking away the opening 'i')

• For each j, $0 \le j \le i-2$, $Diff(Y,j) = Diff(X, j+1) - 1$.

Because i is least such that $Diff(X,i)=0$ (and $i>0$), we have that for all j, $j+1$ runs from 1 to $i-1$ as indices of X, so that

$\boxed{Diff[X, j+1] \ge 1}$ $\therefore \forall j, 0 \le j \le |Y| : Diff(Y,j) \ge 0$.

$\therefore Diff[Y, |Y|) = 0$ and $\forall j, 0 \le j \le |Y|, Diff[Y,j) \ge 0$ $\therefore \boxed{Y \in Bal}$

Moreover, for all K, $i \le K \le n$ $Diff(X, K) = Diff(Z, K-i)$.

$\therefore Diff[Z, |Z|) = Diff(X,n) = 0$, and for all l, $0 \le l \le z$, $Diff(Z,l) \ge 0$.

$\therefore Y \in Bal$ and $Z \in Bal$. And $|Y| = i-2 < n$ (even when $i=n$), $|Z| = n-i < n$ since $i>0$!

Hence we can apply IH $P(i-2)$ and $P(n-i)$ to get $\therefore P(n)$

$S \Rightarrow^* Y$ and $S \Rightarrow^* Z$ in G_2. Thus $S \Rightarrow (S)S \Rightarrow^* (Y)S \Rightarrow^* (Y)Z = X$. done ☒

Q: Do we need $S \to \varepsilon$? G_2: $S \to \varepsilon \mid (S)S$. $L[G_2] = Bal$. ③

Yes since $\varepsilon \in Bal$, but can we get all of $Bal - \{\varepsilon\}$ without ε-rules?

How about $G' = S \to () \mid (S)S$. G_1 is sound, but is it comprehensive for $Bal - \{\varepsilon\}$?

Try $x = (())$. $S \Rightarrow (S)S \Rightarrow ((S)S \Rightarrow (())S$ alas leaves an S we can no longer get rid of.

Theorem [First part of conversion to Chomsky Normal Form]

For any CFG $G = (V, \Sigma, R, S)$, we can build a CFG G' without ε-rules such that $L(G') = L(G) - \{\varepsilon\}$. [Optionally, as the text does, if $\varepsilon \in L(G)$ we can finally add a new start symbol S' with rules $S' \to \varepsilon \mid S$.]

Proof: ① Identify the subset $W \subseteq V$ of nullable variables, meaning $A \in V$ such that $A \Rightarrow^* \varepsilon$. In G_2: $W = V = \{S\}$.

② For every rule $B \to X$ in G, add to R all rules obtained by deleting 1 or more nullable variables. [As literally described, this can be an exponential explosion]

$S \to \varepsilon \mid (S)S$. Add: occurrences of.

* $S \to ()S$
* $S \to (S)$ and
* $S \to ()$ deleting both occurrences.

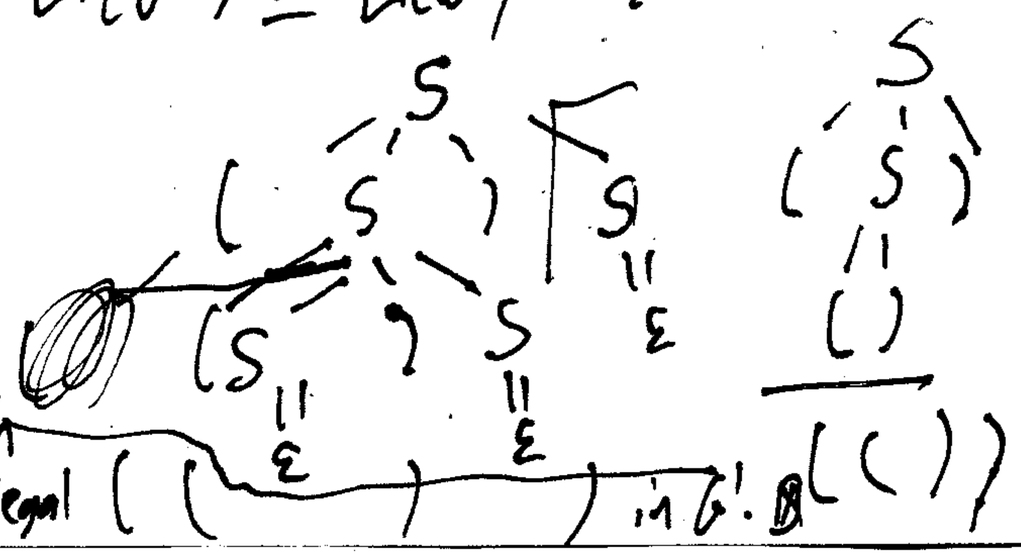
Finally, define G' by deleting all ε-rules.

Now we can do $S \Rightarrow (S) \Rightarrow (())$.

This is sound because all added rules could be simulated by deriving the deleted occurrences to ε.

Hence to show that $L(G') = L(G) - \{\varepsilon\}$, we have to show that all nonempty strings $x \in L(G)$ can be derived in G'.

Consider any parse tree for x in the original grammar G. Just delete all subtrees that yielded ε. G' covers all such deletions with its rules, so the new parse is legal in G'. ∎



Def'n: A CFG G is in Chomsky Normal Form (ChNF) if all rules in R have the form $A \to c$ or $A \to BC$ with $A, B, C \in V$, $c \in \Sigma$. ($c \in \Sigma$; possibly $B, C = A$)

Theorem: For every CFG G we can build a CFG G'' in ChNF such that $L(G'') = L(G) \setminus \{\varepsilon\}$. [optionally add $S'' \to \varepsilon \mid$ right-hand sides of S to allow $L(G'') = L(G)$.]

Note: ChNF involves the step of classifying every terminal $c \in \Sigma$ to a variable X_c and adding the rules $X_c \to c$ as the only productions with terminals on the RHS. $S \to X_L S X_R S \mid X_L X_R S \mid X_L S X_R \mid X_L X_R$, $X_R \to$ ')' $X_L \to$ '('.

Extra The rest of the algorithm:

① Eliminate ε-rules as above (don't do the optional $S' \to S \mid \varepsilon$ yet: save it for the end.

② Now find all pairs of variables (A, B) such that $A \Rightarrow^* B$. This might happen by transitivity: $A \to C$, $C \to D$, $D \to B$. Add all right-hand sides of B as rule options for A. Finally delete all "unit productions" $A \to C$ etc. Call the result G''.

③ Finally, the "silly steps": Alias every $c \in \Sigma$ to X_c as above, except in rules $A \to b$.

④ For every right-hand side of length $\ell \geq 3$, make variables $Y_1, \ldots, Y_{\ell-1}$. If the rule is $A \to V_1 V_2 \cdots V_\ell$, the new rules are $A \to V_1 Y_1$, $Y_1 \to V_2 Y_2$, $Y_2 \to V_3 Y_3$, ..., up to $Y_{\ell-1} \to V_{\ell-1} V_\ell$. Use different "$Y$"-variables for each rule. Call that G'' too. Optionally, add $S'' \to \varepsilon \mid$ right-hand sides of S in G''! Then $L(G'') = L(G)$ in ChNF.

Example of step ④ for our parentheses grammar:

$S \to X_L Y_{11} \mid X_L Y_{21} \mid X_L Y_{31} \mid X_L X_R$, $X_R \to$ ')', $X_L \to$ '('

$Y_{11} \to S Y_{12}$, $Y_{12} \to X_R S$, $Y_{21} \to X_R S$, $Y_{31} \to S X_R$. Option: $S'' \to \varepsilon \mid X_L Y_{11} \mid X_L Y_{21} \mid X_L Y_{31} \mid X_L X_R$

Then G'' is in Chomsky NF. (Can combine Y_{12} and Y_{21}, but this is hideously unreadable anyway.)