CSE396 Lecture Thu Mar 24 Spring 2016 ①

$E \to T \mid E + T$

$T \to F \mid T * F$

$F \to (E) \mid \langle \underline{\text{Variable}} \rangle \mid \underline{\text{Const}}$

var

As distinct from a Grammar Variable, e.g. var → x | y | z | ... etc.

Prove: Every formula generated by this grammar does not have two consecutive * or + signs, nor any leading or trailing + or *. Indeed, it is surrounded by formula variables or constants or ( )

P_E, P_T, P_F all ≡ Every formula f I derive begins and ends with parentheses or constants or vars.

Proof: $E \to T$: Suppose $E \Rightarrow^* f$ using this rule first. Then $T \Rightarrow^* f$. By IH P_T on RHS, f begins and ends with (,) or var or const. This upholds P_E on LHS.

$E \to E+T$: Suppose $E \Rightarrow^* f$ utrf. Then $f = g + h$ where $E \Rightarrow^* g$ and $T \Rightarrow^* h$. By IH P_E on RHS, g begins with (or var or const. By IH P_T on RHS, h ends with) or var or const. ∴ Hence f begins with (or var or const since g does, and f ends -- ditto since h does, ∴ P_E on LHS.

We could do this

By a conventional induction on the number n of { ~~operators in f~~ / steps in the derivation }

Prove $\forall n\, P(n)$, where $P(n) \equiv$ for all formulas f with n operators, f does not have two consecutive operators, nor does f begin or end with + or *.

Base case ($n=0$): const and var have no operators, ditto ((const)) ((var)) etc., and these satisfy $P(0)$.

$\boxed{\text{Let any } f \text{ with } n \text{ ops be given}}$

Ind ($n \geq 1$). Then f must have been derived from one of E, T, F by the rule $E \to E+T$ or $T \to T*F$ somewhere. Consider the rule

$E \to E+T$: Then $f = g+h$ where $E \Rightarrow^* g$ and $T \Rightarrow^* h$.

Then the numbers m_1 of ~~ops~~ steps in g and m_2 of ~~ops~~ steps in h add up to $n-1$, so both m_1 and m_2 are $< n$. By the

Principle of Strong Induction : If for all n, the truth of
Course of Values Induction $[P(m)$ for all $m < n]$ implies the truth of $P(n)$, then $\forall n P(n)$ follows.

So by IH $P(m_1)$ and $P(m_2)$, g and h have no consecutive ops
Oops! I needed to state $P(n)$ in the better positive form "begins and ends with () or var or const". $\therefore$ the same holds for $f = g+h$.
Since f was an arbitrary formula with n operators, $P(n)$ holds.
The rule $T \to T*F$ is similar .... $\forall n P(n)$ follows by induction ⊠

Using n as # steps allows $E \to T$ to be handled by IH $P(n-1)$ since if $E \Rightarrow^n f$, then $T \Rightarrow^{n-1} f$. So the proof "works" on # steps.

Larger Point: Structural Induction works automatically and avoids all this fuss!

Unfortunately (?) Comprehensiveness proofs involve you in (messy) numerical induction on the length(s) "n" of (sub)strings. As a silver lining, it does give you a parsing algorithm and sometimes info about ambiguity and/or reducing the grammar.

$G_1 = S \to SS \mid aSb \mid bSa \mid \varepsilon \quad L = \{x \in \{a,b\}^* : \#a(x) = \#b(x)\}$ ③.

Prove $L \subseteq L(G)$. (note: Soundness is "$L(G) \subseteq L$").

$|||$

$(\forall x \in \Sigma^*)$: $x \in L \Rightarrow x \in L(G)$ $\qquad G = (V, \Sigma, R, S)$ $\quad x \in L(G)$.

$|||$ $\qquad\qquad |||$

$(\forall n \geq 0)\ \underbrace{(\text{for each } x \text{ of length } n)\ x \in L \Rightarrow \overbrace{S \overset{*}{\underset{G}{\Rightarrow}} x}^{}}_{\equiv \text{ my "property } P(n)\text{"}}$

Thus proving $L \subseteq L(G)$ is the same as proving $\forall n\, P(n)$.
We can use numerical <u>strong induction</u> on n.

<u>Basis ($n=0$)</u>: $P(0)$ states "For each x of length 0, such that $x \in L$, $S \Rightarrow^* x$."
Well, there is only one string "x" of length 0, namely $x = \varepsilon$.
And $\varepsilon \in L$: $\#a(\varepsilon) = 0 = \#b(\varepsilon)$. But, $S \Rightarrow \varepsilon$, so $\varepsilon \in L(G)$. $\therefore P(0)$ holds.

Ind ($n > 0$): $P(n) \equiv$ "For each $x \in \Sigma^n$, if $x \in L$ then $S \Rightarrow^* x$."
What happens if $n = 1$, or n is odd? Then $x \in L$ <u>is always false</u>. (by default)
But that's fine: $P(n)$ for odd n is always "False $\Rightarrow \ldots$", so <u>it holds</u>.

Suppose n is even. Let any $x \in \Sigma^n$ such that $x \in L$ be given.

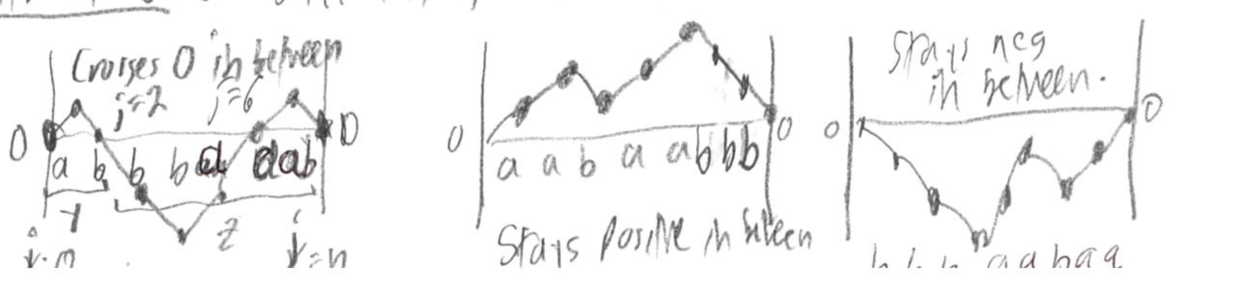
$x_j = b|3$

j fenceposts.

Define, for $0 \leq j \leq n$, <u>Diff</u>$(x, j) = \#a(x_1 \cdots x_j) - \#b(x_1 \cdots x_j)$

Diff$(x, 0)$ is <u>always 0</u>. Diff$(x, n) = 0$ if and only if $x \in L$.

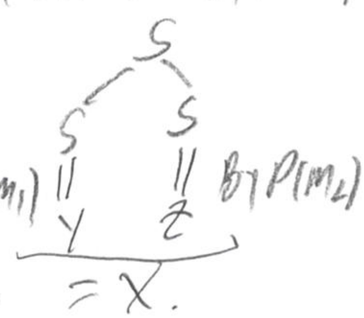
Then there are 3 possibilities for the graph of Diff(x, j) from $j = 0$ to $j = n$



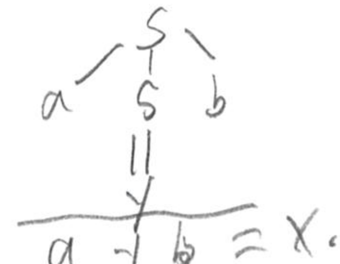
Then the following cases are mutually exhaustive: [that they be exclusive is less important] (4)

(i) $\mathrm{Diff}(x, j) = 0$ for some j, $0 < j < n$.
(ii) $\mathrm{Diff}(x, j)$ starts and stays positive until $j = n$
(iii) $\mathrm{Diff}(x, j)$ starts and stays negative until $j = n$.

Case (i): Take such a j, and define $y = x_1 \cdots x_j$, $z = x_{j+1} \cdots x_n$.
Then $x = yz$ where $\mathrm{Diff}(y, \underbrace{|y|}_{=j}) = 0$ and $\mathrm{Diff}(z, \underbrace{|z|}_{=n-j}) = 0$
Then $y \in L$ and $z \in L$, where $m_1 = |y|$ and $m_2 = |z|$ are both $< n$.
Thus we can use IH $P(m_1)$ and $P(m_2)$ to conclude $S \Rightarrow^* y \wedge S \Rightarrow^* z$.
Finally we assemble the derivation $S \Rightarrow SS \underset{\text{by } P(m_1)}{\Rightarrow^*} yS \underset{\text{by } P(m_2)}{\Rightarrow^*} yz = x$.
So $S \Rightarrow^* x$, proving $P(n)$ in this case.
We can also diagram this via parse trees:



Case ii. Then x begins with a & ends with b.
So $x = ayb$, where necessarily $\#a(y) = \#b(y)$.
Thus by $x \in L$, we have $y \in L$ and $|y| = n-2 < n$. So we can apply IH $P(n-2)$ to conclude $S \Rightarrow^* y$.
Thus $S \Rightarrow aSb \Rightarrow^* ayb = x$. So $S \Rightarrow^* x$.



Case iii Then $x = bza$ where $z \in L$.
Similar getting $S \Rightarrow bSa \underset{\text{by IH } P(n-2)}{\Rightarrow^*} bza = x$. ∴ $P(n)$ holds in all cases.

So $\forall n\, P(n)$ follows by strong induction, which means $L \subseteq L(G)$.
Since we earlier showed $L(G) \subseteq L$, we finally get $L = L(G)$.

Recitation Notes – continuing on... (5)

Sometimes a bigger grammar makes the $\left\{\begin{array}{l}\text{parsing}\\ \text{counting}\end{array}\right.$ details easier, although having more variables makes the proof "heavier" at the beginning.

Recall Gr: $\left\{\begin{array}{l} S \to \varepsilon \mid AB \mid BA \\ A \to a \mid aS \mid BAA \\ B \to b \mid bS \mid ABB \end{array}\right.$

$L = \{x : \#a(x) = \#b(x)\}$. Call this "$L_S$".

We want to prove $(\forall n \geq 0)\, P(n)$, where

$\boxed{P(n) \equiv \text{for each } x \in \Sigma^n,\ x \in L \Rightarrow S \Rightarrow^* x.}$

When we have other variables, we need to strengthen the induction by maintaining language comprehension properties of the other variables.

Define $L_A = \{x : \#a(x) = \#b(x) + 1\}$ $\quad Q(n) \equiv$ for each $w \in \Sigma^n$, if $w \in L_A$, then $A \Rightarrow^* w$.

$L_B = \{x : \#b(x) = \#a(x) + 1\}$ $\quad R(n) \equiv$ for each $w \in \Sigma^n$, $w \in L_B \Rightarrow B \Rightarrow^* w$.

Prove $(\forall n \geq 0)\, P'(n)$ where $P'(n) = P(n) \wedge Q(n) \wedge R(n)$. Basis $(n=0)$:

$P(0) \equiv$ if $\varepsilon \in L_S$ then $S \Rightarrow^* \varepsilon$. Well, $\varepsilon \in L_S$ ✓ And $S \Rightarrow \varepsilon$. ✓ OK

$Q(0) =$ if $\varepsilon \in L_A$ then $A \Rightarrow^* \varepsilon$. Well $\varepsilon \notin L_A$ so we don't care.

$R(0)$ likewise holds by default. $\therefore P'(0)$ holds.

Ind $(n \geq 1)$ Let any $x \in \Sigma^n$ be given. Now odd n are relevant for $Q(n), R(n)$

$P(n)$: If $x \in L_S$ (then n is even), then we can break into easier cases:

(i) x begins with a $\quad\}$ exhaustive since $n > 0$ so $x \neq \varepsilon$. $\quad$ Then $x = ay$ where $y \in L_B$. By IH $R(n-1)$, $B \Rightarrow^* y$. So $S \Rightarrow AB \Rightarrow^* ay = x$.

(ii) x begins with b

In (ii), $x = bz$ where $z \in L_A$. By IH $Q(n-1)$, $A \Rightarrow^* z$. So $S \Rightarrow BA \Rightarrow bA \Rightarrow^* bz = x$.

We still need to do $Q(n) \wedge R(n)$. Do they add more pain or less pain than G_1 had?

$Q(n)$: If $x \in L_A$ then either (i) x begins with a or (ii) x begins with b.

(i) $x = aw$ where $w \in L_S$. By IH $P(n-1)$, $S \Rightarrow^* w$. So $A \Rightarrow aS \Rightarrow^* aw = x$.

(ii) $x = bw$. Here $\#a(w)$ must equal $\#b(w) + 2$. But we can break $w =: uv$ such that $\#a(u) = \#b(u) + 1$ and $\#a(v) = \#b(v) + 1$. By IH, $A \Rightarrow^* u$, $A \Rightarrow^* v$. So $S \Rightarrow BAA \Rightarrow^* buv = x$.