Lecture 20

Balanced Parentheses

Intuitively, a string of parentheses is *balanced* if each left parenthesis has a matching right parenthesis and the matched pairs are well nested. The set PAREN of balanced strings of parentheses [ ] is the prototypical context-free language and plays a pivotal role in the theory of CFLs.

The set PAREN is generated by the grammar

$$S \to [S] \mid SS \mid \epsilon.$$

This is not obvious, so let's give a proof. First we need a formal characterization of *balanced*. To avoid confusing notation, we'll use

$$L(x) \stackrel{\text{def}}{=} \#[(x) = \text{the number of left parentheses in } x,$$

$$R(x) \stackrel{\text{def}}{=} \#](x) = \text{the number of right parentheses in } x.$$

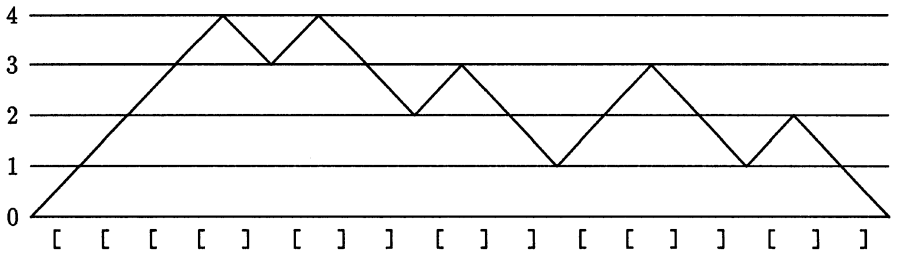
We will define a string x of parentheses to be *balanced* if and only if

(i) $L(x) = R(x)$, and

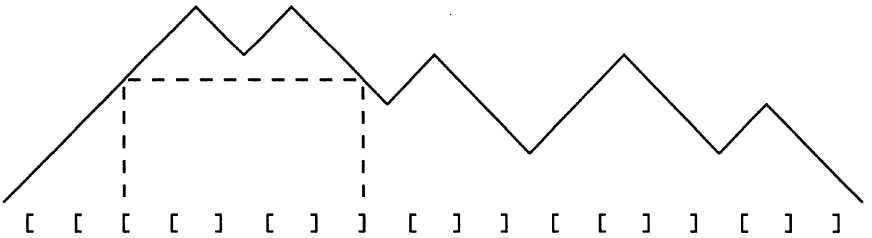
(ii) for all prefixes y of x, $L(y) \geq R(y)$.

(Recall that a *prefix* of x is a string y such that $x = yz$ for some z.) To see that this definition correctly captures the intuitive notion of *balanced*, note that property (i) says that there must be the same number of left

parentheses as right parentheses, which must certainly be true if x is balanced; and property (ii) says that for any way of partitioning x into yz, there cannot be more right parentheses in y than left parentheses, because right parentheses can only match left parentheses to the left of them. Thus (i) and (ii) are certainly necessary conditions for a string to be balanced. To see that they are sufficient, draw the graph of the function $L(y) - R(y)$ as y ranges over prefixes of x:



Property (i) says that the graph ends with value 0 (i.e., $L(x) - R(x) = 0$), and (ii) says that it never dips below 0 on the way across. If the graph satisfies these two properties, then given any parenthesis in the string, one can find the matching parenthesis by shooting upward and ricocheting off the graph twice.



Thus we will take (i) and (ii) as our formal definition of balanced strings of parentheses and prove that the given grammar generates exactly the set of such strings, no more and no less.

Theorem 20.1 *Let G be the CFG*

$$S \to [S] \mid SS \mid \epsilon.$$

Then

$$L(G) = \{x \in \{[,]\}^* \mid x \textit{ satisfies (i) and (ii)}\}.$$

Proof. We show the inclusion in both directions. Both arguments will be by induction, but induction on different things.

First we show the forward inclusion: if $S \xrightarrow[G]{*} x$, then x satisfies (i) and (ii). Thus any string generated by G is balanced.

We would like to use induction on the length of the derivation of x from S, but since the intermediate sentential forms in this derivation will contain nonterminals, we need to strengthen our induction hypothesis to allow nonterminals. Thus we will actually show that for any $\alpha \in (N \cup \Sigma)^*$, if $S \xrightarrow[G]{*} \alpha$, then α satisfies (i) and (ii). This will be proved by induction on the length of the derivation $S \xrightarrow[G]{*} \alpha$.

Basis

If $S \xrightarrow[G]{0} \alpha$, then $\alpha = S$ by definition of the relation $\xrightarrow[G]{0}$. But the sentential form S satisfies (i) and (ii) trivially.

Induction step

Suppose $S \xrightarrow[G]{n+1} \alpha$. Let β be the sentential form immediately preceding α in the derivation. Then

$$S \xrightarrow[G]{n} \beta \xrightarrow[G]{1} \alpha.$$

By the induction hypothesis, β satisfies (i) and (ii). There are now three cases, corresponding to the three productions in the grammar that could have been applied in the last step to derive α from β. We will show in each case that properties (i) and (ii) are preserved.

The first two cases, corresponding to productions $S \to \epsilon$ and $S \to SS$, are easy because neither production changes the number or order of parentheses. In either case there exist $\beta_1, \beta_2 \in (N \cup \Sigma)^*$ such that

$$\beta = \beta_1 S \beta_2 \quad \text{and} \quad \alpha = \begin{cases} \beta_1\beta_2 & \text{if } S \to \epsilon \text{ was applied,} \\ \beta_1 S S \beta_2 & \text{if } S \to SS \text{ was applied;} \end{cases}$$

and in either case α satisfies (i) and (ii) iff β does.

If the last production applied was $S \to [S]$, then there exist $\beta_1, \beta_2 \in (N \cup \Sigma)^*$ such that

$$\beta = \beta_1 S \beta_2 \quad \text{and} \quad \alpha = \beta_1 [S] \beta_2,$$

and by the induction hypothesis (i) and (ii) hold of β. Then

$$\begin{aligned} L(\alpha) &= L(\beta) + 1 \\ &= R(\beta) + 1 \quad \text{since } \beta \text{ satisfies (i)} \\ &= R(\alpha), \end{aligned}$$

thus (i) holds of α. To show that (ii) holds of α, let γ be an arbitrary prefix of α. We want to show that $L(\gamma) \geq R(\gamma)$. Either

- γ is a prefix of β_1, in which case γ is a prefix of β, so (ii) holds for the prefix γ by the induction hypothesis; or

- γ is a prefix of $\beta_1 [S$ but not of β_1, in which case

$$\begin{aligned} L(\gamma) &= L(\beta_1) + 1 \\ &\geq R(\beta_1) + 1 \quad \text{induction hypothesis, since } \beta_1 \text{ is a prefix of } \beta \\ &> R(\beta_1) \\ &= R(\gamma); \quad \text{or} \end{aligned}$$

- $\gamma = \beta_1 [S] \delta$, where δ is a prefix of β_2, in which case

$$\begin{aligned} L(\gamma) &= L(\beta_1 S \delta) + 1 \\ &\geq R(\beta_1 S \delta) + 1 \quad \text{induction hypothesis} \\ &= R(\gamma). \end{aligned}$$

Thus in all cases $L(\gamma) \geq R(\gamma)$. Since γ was arbitrary, (ii) holds of α. This concludes the inductive proof that if $S \xrightarrow[G]{*} x$, then x is balanced.

Now we wish to show the other direction: if x is balanced, then $S \xrightarrow[G]{*} x$. This is done by induction on $|x|$. Assume that x satisfies (i) and (ii).

Basis

If $|x| = 0$, we have $x = \epsilon$ and $S \xrightarrow[G]{*} x$ in one step using the production $S \rightarrow \epsilon$.

Induction step

If $|x| > 0$, we break the argument into two cases: either

(a) there exists a *proper* prefix y of x (one such that $0 < |y| < |x|$) satisfying (i) and (ii), or

(b) no such prefix exists.

In case (a), we have $x = yz$ for some z, $0 < |z| < |x|$, and z satisfies (i) and (ii) as well:

$$L(z) = L(x) - L(y) = R(x) - R(y) = R(z),$$

and for any prefix w of z,

$$\begin{aligned} L(w) &= L(yw) - L(y) \\ &\geq R(yw) - R(y) \quad \text{since } yw \text{ is a prefix of } x \text{ and } L(y) = R(y) \\ &= R(w). \end{aligned}$$

By the induction hypothesis, $S \xrightarrow[G]{*} y$ and $S \xrightarrow[G]{*} z$. Then we can derive x by starting with the production $S \to SS$, then deriving y from the first S, then deriving z from the second S:

$$S \xrightarrow[G]{1} SS \xrightarrow[G]{*} yS \xrightarrow[G]{*} yz = x.$$

In case (b), no such y exists. Then $x = [z]$ for some z, and z satisfies (i) and (ii). It satisfies (i) since

$$L(z) = L(x) - 1 = R(x) - 1 = R(z),$$

and it satisfies (ii) since for all nonnull prefixes u of z,

$$L(u) - R(u) = L([u) - 1 - R([u) \geq 0$$

since $L([u) - R([u) \geq 1$ because we are in case (b). By the induction hypothesis, $S \xrightarrow[G]{*} z$. Combining this derivation with the production $S \to [S]$, we get a derivation of x:

$$S \xrightarrow[G]{1} [S] \xrightarrow[G]{*} [z] = x.$$

Thus every string satisfying (i) and (ii) can be derived. □