CONTEXT-FREE GRAMMARS FOR THE BALANCED AND OVERBALANCED BINARY LANGUAGES

by

George Epstein

Computer Science Department

Indiana University

Bloomington, Indiana 47401

TECHNICAL REPORT No. 71

CONTEXT-FREE GRAMMARS FOR THE
BALANCED AND OVERBALANCED
BINARY LANGUAGES

GEORGE EPSTEIN

NOVEMBER, 1977

1. Introduction

The exercise of giving grammars for the balanced and over-balanced binary languages was performed approximately 15 years ago by S. Ginsburg, this author, and others. The feature of the author's work given in this technical report is its transparent structure.

First, a candidate grammar is hypothesized which preserves key features of the language to be generated. Second, there is an attempt to show that an arbitrary sentence of the language can be generated by this grammar (if the attempt fails, the candidate grammar must be widened, then sharpened). Third, to obtain generalization of the language, add appropriate production rule(s) to the grammar as it stands.

CONTEXT-FREE GRAMMARS FOR THE BALANCED AND OVERBALANCED BINARY LANGUAGES

G. Epstein

Balanced Binary Language

The balanced binary language is defined as the set of all strings over the symbols 0 and 1 for which the number of 0's in any string is equal to the number of 1's in that string. This language is a context-free language given by the following rewriting rules:

(1) $S \longrightarrow 0S1$

(2) $S \longrightarrow 1S0$

(3) $S \longrightarrow 01$

(4) $S \longrightarrow 10$

(5) $S \longrightarrow SS$.

Proof

It is obvious that this grammar generates only those strings for which the number of 0's is equal to the number of 1's. To prove that this grammar generates all such strings, note that

I. Rule(5) followed by rules (3) and (4) produce the rule $S \longrightarrow 0110$

II. Rule(5) followed by rules (1) and (4) produce the rule $S \longrightarrow 0S110$

III. Rule(5) followed by rules (3) and (2) produce the rule $S \longrightarrow 011S0$

IV. Rule(5) followed by rules (1) and (2) produce the rule $S \longrightarrow 0S11S0$

In similar fashion, one obtains these rules with 1's and 0's interchanged:

Ia. $S \longrightarrow 1001$

IIa. $S \longrightarrow 1S001$

IIIa. $S \longrightarrow 100S1$

IVa. $S \longrightarrow 1S00S1$.

Now the rules 1-4, I-IV, and Ia-IVa exhaust all possible rewriting rules for a string with the number of 0's equal to the number of 1's. For if the string has a 0 at one end and a 1 at the other, then apply one of the rules (1)-(4) to obtain it; if the string has 0's at both ends, then apply one of the rules I-IV to obtain it; if the string has 1's at both ends, then apply one of the rules Ia-IVa to obtain it.

The first of these instructions is obviously applicable. Suppose that the string has 0's at both ends. Then the inner string must have an excess of two 1's and be one of the forms

(i) 11

(ii) S11

(iii) 11S

(iv) S11S ,

where in each of these forms S is again a string with the number of 0's equal to the number of 1's. To prove this statement, consider the string $\alpha = \alpha(1)\alpha(2) \cdots \alpha(n)$ which has two more 1's than 0's, where each $\alpha(i)$, $i = 1, 2, \cdots n$ is either 0 or 1. Let $N(i)$ be the excess in the number of 1's over the number of 0's in the string $\alpha(1)\alpha(2) \cdots \alpha(i)$, $i = 1, 2, \cdots n$. It is obvious that $|N(j+1) - N(j)| = 1$ for $j = 1, 2, \cdots n-1$; i.e., the function $N(i)$ must either increase or decrease by 1 as i moves from j to j + 1. In the case when $N(i) = 0$ for some i, there must be a maximum $i = i_o$ such that $N(i_o) = 0$ and $N(i) > N(i_o)$ for all $i > i_o$. Then it is clear that $\alpha(i_o + 1) = \alpha(i_o + 2) = 1$, for otherwise $i = i_o$ is not maximum. Hence, in this case the inner string is of the form (ii) or (iv) and either rule II or rule IV is applicable. Since $2 - N(i)$ is the excess in the number of 1's over the number of 0's in the string $\alpha(i+1)\alpha(i+2) \cdots \alpha(n)$, $i = 1, 2, \cdots n-1$, if there is an i such that $2 - N(i) = 0$, then there is a minimum $i = i_p$ such that $2 - N(i_p) = 0$ and $2 - N(i) > 2 - N(i_p)$ for all $i < i_p$. It is clear that $\alpha(i_p - 1) = \alpha(i_p - 2) = 1$, for otherwise $i = i_p$ is not minimum. In this instance the inner string is of the form (iii) or (iv) and either rule III or rule IV is applicable. The only remaining case is when $N(i) \neq 0$ and $2 - N(i) \neq 0$ for all i. Since $|N(j+1) - N(j)| = 1$

for all j and N(n) = 2, it follows that in this last case $N(i) > 0$ for all i. Similarly, since $\left| \left[2 - N(j)\right] - \left[2 - N(j + 1)\right] \right| = 1$ for all j and 2 - N(1) is either 1 or 3, it follows that $2 - N(i) > 0$ for all i. Hence, the only permissible value of N(i) is N(i) = 1; that is, the string is of the form (i). Hence, in this last cast rule I is applicable.

In similar fashion rules Ia-IVa can be proven to be applicable when there are 1's at the ends of the string. Continued application of these rules will yield any arbitrary string in which the number of 0's is equal to the number of 1's.

Overbalanced Binary Language

The 0- overbalanced binary language is defined as the set of all strings over the symbols 0 and 1 for which the number of 0's in any string is greater than or equal to the number of 1's in that string. This language is given by the rules (1)-(5) above and the additional rule

(6) $S \longrightarrow 0$.

Proof

These rules obviously produce only strings with this property. Let Z(i) be the excess in the number of 0's over the number of 1's in the string $\alpha(1)\,\alpha(2)\,\alpha(3)\cdots\alpha(n)$. Then, if Z(i) = 0 for some i, there is a maximum value $i = i_o$ such that $Z(i_o) = 0$. If $i_o < n$, then $\alpha(i_o + 1) = 0$ and there is a maximum value $i = i_1$, such that $Z(i_1) = 1$. Continuing in this fashion, there is a string

$$\alpha(1)\,\alpha(2)\cdots\alpha(i_o)0\,\alpha(i_o + 2)\cdots\alpha(i_1)0\,\alpha(i_1 + 2)\cdots\alpha(i_2)\cdots\alpha(n)$$

with the property that the 0's are markers for strings (possibly null) which have as many 0's as 1's. A similar string is produced in the case that $Z(i) \neq 0$ for all i by starting with 0 markers in the first m positions, where $Z(i) \neq m - 1$ for all $i > m$ and m maximum, and then proceeding in the same manner as above, finding a maximum $i = i_m$ such that $Z(i_m) = m$, and so forth.

If the number of non-null strings between markers is A, then application of rule (5), A + Z(n)-1 times, followed by application of rule (6), Z(n) times will produce a string in which the symbol S represents only strings for which the number of 0's is equal to the number of 1's. Rules (1)-(5) will then yield the final string.