$N_{\overline{2}}1$ 

Описание языка:

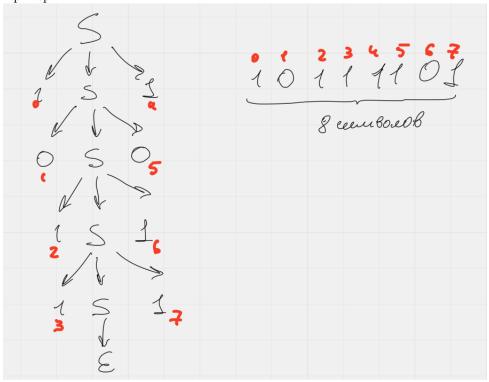
 $V_T = \{S\}$ 

 $V_N = \{1, 0\}$ 

 $P = \{S - > 1S1 \mid 0S0 \mid \varepsilon \}$ 

 $G = <V_T, V_N, P, S>$ 

Пример:



 $N_{2}$ 

Начинаем с нетерминала S, если мы идем в ветку, где S заменяется на aSA, то у нас получается строка, в которой слева от S стоит столько терминалов a, сколько и справа от S нетерминалов A. И так мы можем набирать справа и слева много раз, пока не заменим S на aT, сделав +1 к терминалу а справа. Дальше слово кодируется однозначно:

$$a^{n+1}TA^n - > a^{n+1}bTaA^{n-1}$$

в следующем переходе мы просто много раз свапаем A c , чтобы получился нетерминал TA, если же мы решил сократить нетерминал T не в конце, то дальше раскодировать не сможем

В итоге мы приходим к такому виду:

$$a^{n+1}b^nTa^n - > a^{n+1}b^{n+1}a^{n+1}$$

Следовательно язык, порождаемый данной грамматикой, описывается так:

 $a^m b^m a^m$ , где  $m \ge 2$ , самый маленький случай m=2 получается из случая, когда мы сразу S заменяем на aT.

№3

я изучал спецификацию синтаксиса языка Go, ссылка на документ вот тут, и отметил 3 новые особенности синтаксиса, а именно:

- 1) defer statements
- 2) panic
- 3) йота (iota)