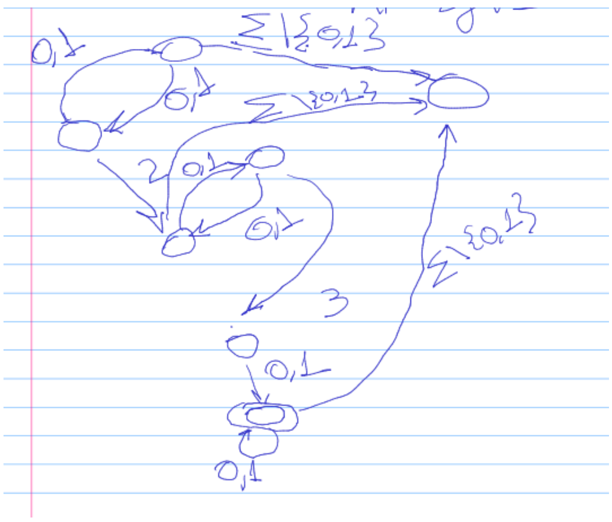
0.1.1. 1. Considere $L_1 = (0 \bigcup 11)^m 2(0 \bigcup 1)^n 3(0 \bigcup 1)^+$, com $m, n \geq 0, m$ par e n impar. Prove que L_1 é uma linguagem regular construindo um AFD com no máximo 7 estados que a reconheça



0.1.2. 2. Diga se $L_2 = \{w \in \{0,1\}^* \mid w \text{ possui menos 0s que 1s}\}$ é ou não é uma linguagem regular, mostrando uma expressão regular (caso seja) ou usando lema do bombeamento (caso não seja)

O enunciado do lema do bombeamento

tal que x é prefixo, y é uma concatenação de cadeias e z é o sufixo e m e k pertencem aos naturais com m>0
$|xy| \leq p$
$|y| \geq 1$
$(\forall n \geq 0)(xy^n z \in L)$

1. Considere que o número de 1′s seja k e o número de 0′s seja k-1.
2. Considere a seguinte cadeia que pertence a $L_2 = 10^{k-m-1}0^m 1^{k-1}$
3. Como m pertence aos naturais e é maior que 0 então existe um número menor de 0′s que 1′s como gostaríamos.
4. Escolha $p = k - m$ como comprimento de bombeamento, temos

$xy^n = 1\left(0^{k-m-1}\right)^n$, expandindo temos: $xy^n = \left(0^{nk-nm-n}\right) = 10^{n(k-m-1)}$
5. Concatenando com o sufixo temos $10^{n(k-m-1)}0^m 1^{k-1}$
6. Temos pelo menos n zeros a mais que 1′s onde n é a quantidade de concatenações da cadeia bombeada, assim

a cadeia não pertence a L, visto que há mais zeros que 1′s, portante não é regular.

0.1.3. 3. Prove que a interseção entre linguagens regulares e linguagens livre de contexto são linguagens livre de contexto.

Sabemos que toda linguagem regular pode ser representada como expressão regular.

0.1.4. 4. Contrua uma GLC na forma normal de chomsky para o conjunto de todos palíndromos binários

a definição da forma normal de chomsky:
$S \rightarrow \epsilon$
$A \rightarrow BC$

$A \to a$
tal que B e C não sejam S, embora A possa ser.

$$
\begin{aligned}
&S \to \epsilon \\
&S \to 0 \\
&S \to 1 \\
&S \to CC \\
&S \to DD \\
&S \to CA' \\
&S \to DA'' \\
&A \to CA' \\
&A' \to AC \\
&A \to DA'' \\
&A'' \to AD \\
&C \to 0 \\
&D \to 1 \\
&A \to 0 \\
&A \to 1
\end{aligned}
$$

0.1.5. 5. Prove que uma máquina de Turing é equivalente a um autômato com duas ou mais pilhas

Essa prova consiste da ida que tem como enunciado "uma máquina de turing atua como autômato de duas pilhas" e a volta "um autômato de duas pilhas atua como máquina de turing", provemos primeiranmente a ida

$\Longrightarrow$

Sabemos que uma MT é definida pela seguinte 7-upla
$M := (\Sigma, \Gamma, Q_o, Q_{\text{accept}}, Q_{\text{reject}}, \delta, Q)$

e um autômato com duas pilhas pela seguinte 7-upla
$A_p := (\Sigma, \Gamma_1, \Gamma_2, Q_o, Q_{\text{accept}}, \delta, Q)$

$\delta_M := Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$
$\delta_p := Q \times \Sigma \times \Gamma_1 \times \Gamma_2 \to P(Q \times \Gamma_1 \times \Gamma_2)$

vamos converter a MT num autômato com pilha através do seguinte mapeamento
- Os estados Q da MT são os mesmos estados Q da AP
- O alfabeto Σ da MT é o mesmo da AP
- O alfabeto Γ da MT é o alfabeto Γ_1 da AP
-

0.1.6. 6. Resolva os seguintes problemas:

0.1.6.1. a. Explique brevemente o lema de Church-Turing

Máquina de Turing é equivalente a um algoritmo, isto é dada uma entrada w para um algoritmo se o mesmo iria "entrar em loop" então a MT tambem iria, se o mesmo iria eventualmente parar a MT tambem, essa breve explicação é válida para todo autômato que tem capacidade computacional equivalente a uma máquina de Turing, reconhece todas as linguagens que uma MT reconhece, embora a complexidade computacional varie de autômato a autômato.

0.1.6.2. b. Prove que $L_3 = \{\langle G, A\rangle$

autômato que tem capacidade computacional equivalente a uma máquina de Turing, reconhece todas as linguagens que uma MT reconhece, embora a complexidade computacional varie de autômato a autômato.