Vimos que a inclusão {linguagens regulares} ⊆ {linguagens livres de contexto} é própria, ou seja, existem linguagens livres de contexto que não são regulares. Por exemplo:

$$\mathcal{L} = \{ 0^i 1^i \mid i = 0,1,2,\dots \}$$

Temos que $\mathcal{L} = \mathcal{L}(G)$ onde G: $S \to 0S1$ $\quad S \to \varepsilon$

Será que existem linguagens que não são livres de contexto? Certamente existem, por um argumento de enumerabilidade. O conjunto das possíveis linguagens é não enumerável e o conjunto das gramáticas livres de contexto é enumerável.

Queremos exibir um exemplo, e construir um método para verificar se uma linguagem é, ou não, livre de contexto. No fim de tudo vamos acabar tendo um resultado equivalente ao "pumping lemma" para linguagens livres de contexto.

O exemplo que vamos usar é:

$$\mathcal{L} = \{ 0^i 1^i 2^i \mid i = 0,1,2,\dots \}$$

que não é livre de contexto. Para provar isso, usaremos o seguinte lema, que não vamos demonstrar:

Lema: Se G é uma gramática livre de contexto, existe uma gramática G', também livre de contexto, tal que:

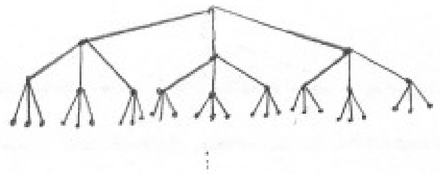
i) $\mathcal{L}(G') = \mathcal{L}(G)$

ii) G' é sensível ao contexto

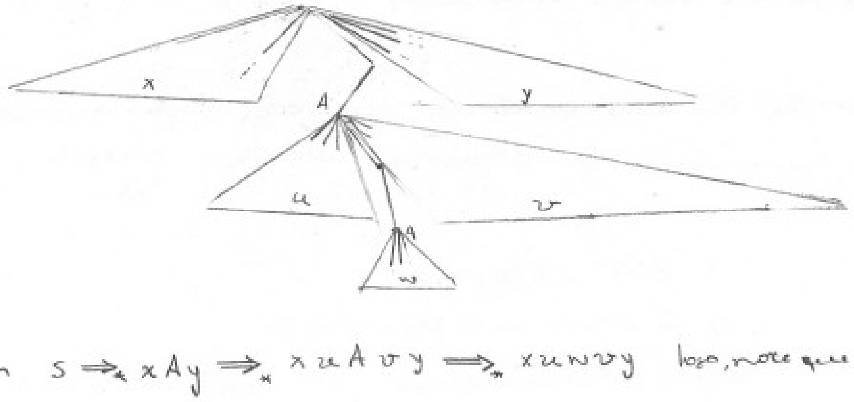
iii) G' não tem regras simples, ou seja, regras da forma $A \to B$ com A e B não terminais.

Lema do Bombeamento: Suponha, por hipótese, que $\mathcal{L}$ é livre de contexto, então existe uma gramática livre de contexto G tal que $\mathcal{L} = \mathcal{L}(G)$. Pelo lema anterior, existe uma gramática G' livre de contexto, sensível ao contexto e sem regras simples tal que $\mathcal{L} = \mathcal{L}(G')$.

Seja $(n-1)$ o número de símbolos não terminais de G' e m o tamanho da maior regra, então, no primeiro nível da árvore, temos no máximo m símbolos, no segundo símbolo no máximo m^2 símbolos, no terceiro, no máximo m^3 símbolos e assim por diante.



Assim, a árvore de derivação de uma sentença w com $|w| > m^k$ tem altura maior que k. Assim, tomando uma sentença de comprimento maior que m^n, sua árvore de derivação tem altura maior que n. Consideremos o seu ramo de maior comprimento (maior que n), e analise os nós terminais nesse ramo. Certamente existem dois nós terminais iguais, digamos A.



Assim $S \Rightarrow_* xAy \Rightarrow_* xuAvy \Rightarrow_* xuwvy$ logo, note que.

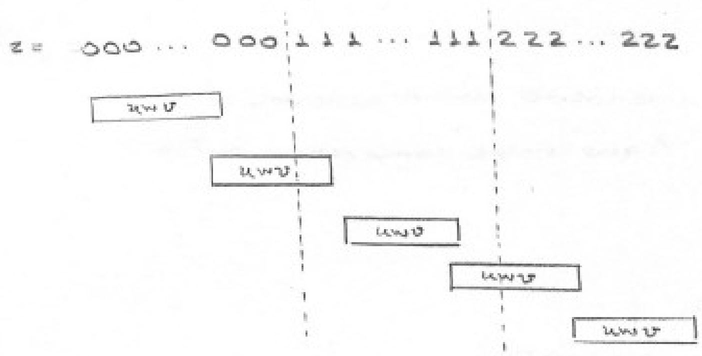
$$S \Rightarrow_* xu^k w v^k y \ , \forall k \in \mathbb{N}.$$

Isso é, em essência, a prova do Lema do Bombeamento. Vamos enunciá-lo formalmente agora:

Se $\mathcal{L}$ é uma linguagem livre de contexto, então existe um natural n tal que:

$$\left.\begin{array}{l} z \in \mathcal{L} \\ |z| > n \end{array}\right] \; z = xuwvy \text{ tal que } \begin{cases} uv \neq \varepsilon \\ |uwv| \leq n \\ xu^k w v^k y \in \mathcal{L}, \; k = 0,1,2,\ldots \end{cases}$$

Usando o "pumping lemma", vamos provar que $\mathcal{L} = \{0^i 1^i 2^i \mid i = 0,1,2,\ldots\}$ não é livre de contexto. Seja n a constante do "pumping lemma" e seja $z = 0^n 1^n 2^n$.



Note que a subsentença uwv pode estar em vários pontos da sentença. Analisemos dois casos:

i) se uwv contém 0's então ele não contém 2's, assim: $xu^k w v^k y$ contém mais 1's ou mais 0's do que 2's para $k > 1$

ii) se uvw não contém 0's então para $k > 1$, a sentença $xu^k w v^k y$ contém mais 1's ou mais 2's do que 0's.

Nos dois casos $xu^k w v^k y \notin \mathcal{L}$ para $k > 1$.

EXTRA (Problema de Turing): Turing propôs um problema bastante interessante: dado um programa P e um dado de entrada d, determinar se o programa P com dado de entrada d para ou não (isto é, se entra em loop infinito ou não).

Esse problema não é computável, ou seja, não existe como escrever um programa de computador que resolva esse problema.

Suponha, por absurdo, que existe uma rotina PARA(P,d) que retorna:

$$\text{PARA}(P,d) = \begin{cases} 0, \text{ se } & P(d) \text{ não para} \\ 1, \text{ se } & P(d) \text{ para} \end{cases}$$

Usando essa rotina, vamos escrever um programa X que recebe como dado outro programa:

```
X(P) {
   if PARA(P,P) {
      while(1);
   }else {
         return 1;
   }
}
```

Note que o dado que P recebe pode ser o programa P (ele mesmo), convertido em um número ou em uma sequência de bits se necessário. Agora vamos mostrar uma contradição. Façamos uma pergunta:

X(X) para?

Note que:

$$X(X) \text{ para} \iff PARA(X,X) = 0 \iff X(X) \text{ não para}$$

o que nos dá uma contradição. Note que essa prova é muito semelhante ao paradoxo de Russel, onde provamos que o conjunto:

$$S = \{x \text{ conjunto} \mid x \notin x\}$$

não existe, fazendo a pergunta $S \in S$?