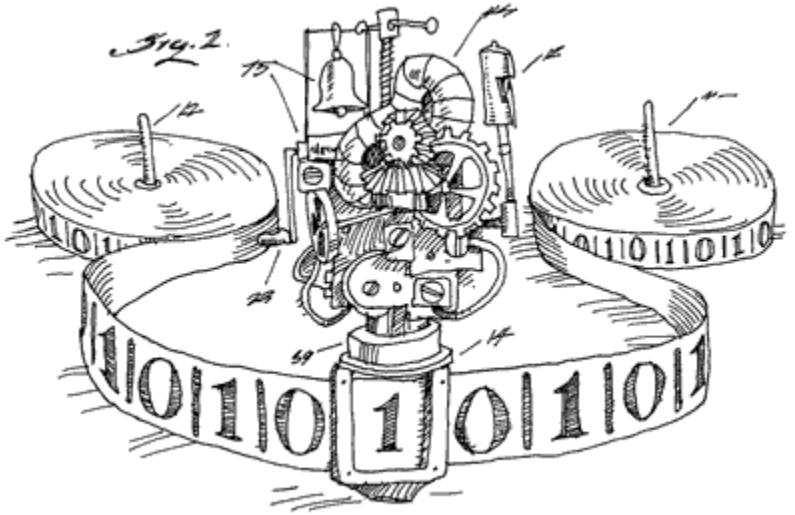
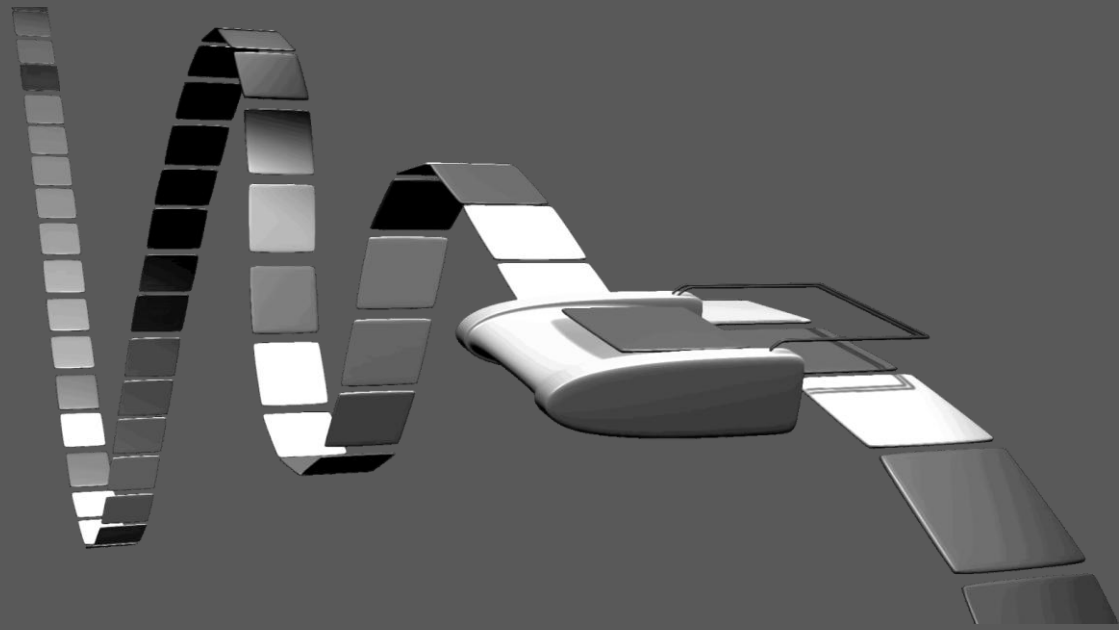
Gramáticas livres de contexto

Introdução à Teoria da Computação

Nicollas Mocelin Sdroievski



Relembrando

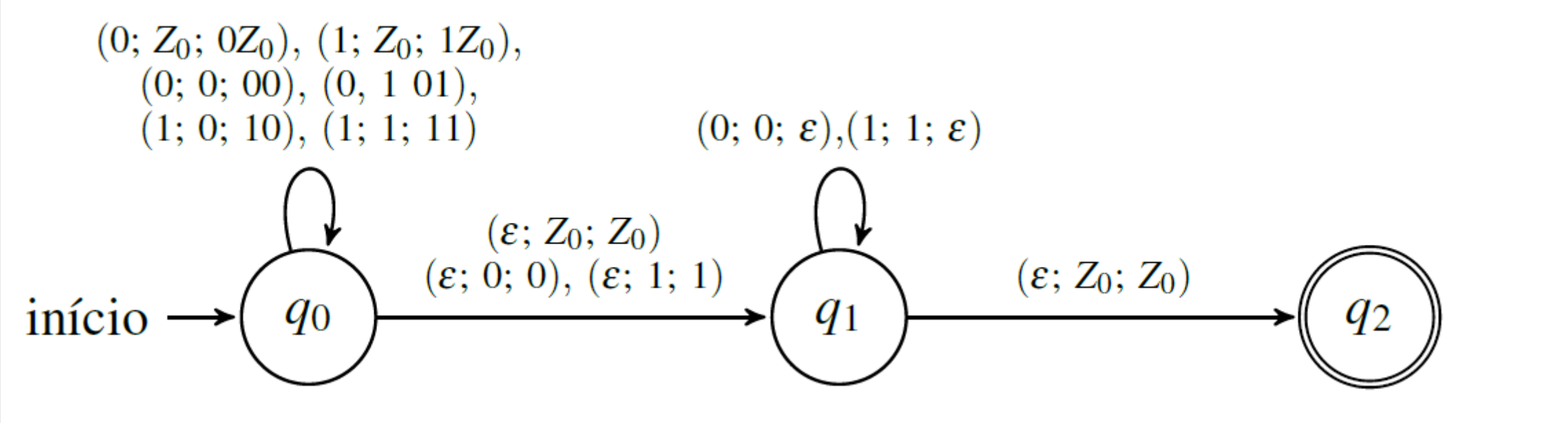


Intuição

- ε-AFNs que possuem uma memória adicional no formato de uma pilha de dados
- ε-AFN
  - Processa um ou zero símbolos da string de entrada
  - Faz uma transição de estado
- Autômato com pilha
  - Consome um ou zero símbolos da string de entrada
  - Desempilha o símbolo do topo da pilha
  - Faz uma transição de estado
  - Empilha uma quantidade finita de símbolos na pilha

Exemplo

- AP para a linguagem L_{RR} das strings binárias da forma ww^R



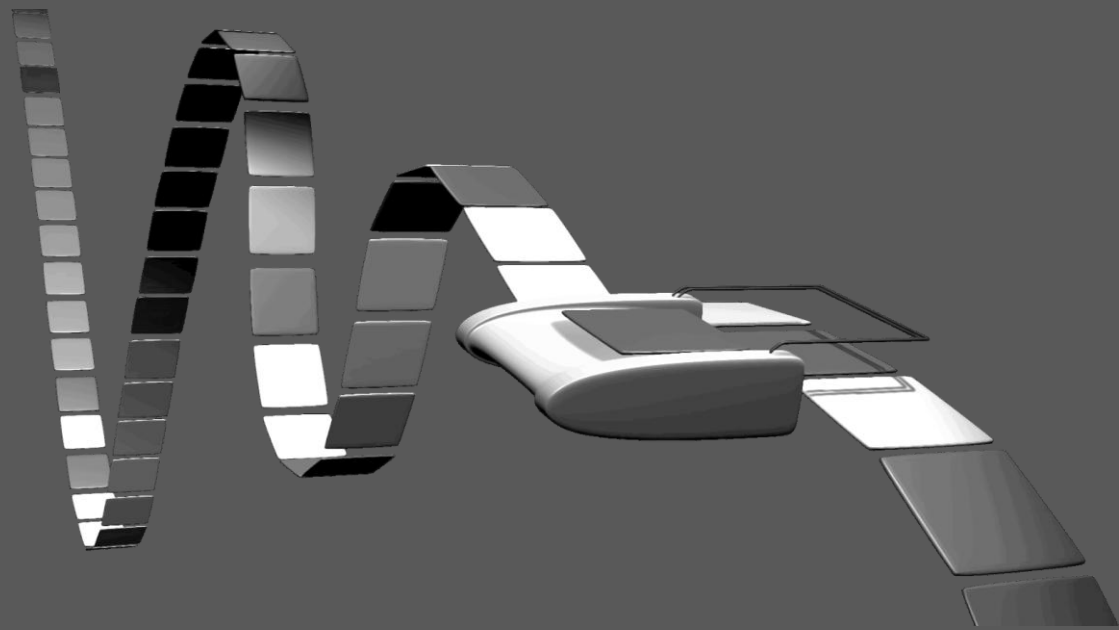
Projetando APs

- É comum sempre manter Z_0 no fundo da pilha
  - Ajuda a saber quando a pilha está vazia
- **Observação**. Um AP *morre* se tenta realizar uma transição com a pilha vazia (sem nem mesmo Z_0)

Exercício 4.8 Apresente a definição formal e o diagrmaa de um AP que aceite a linguagem $L = \{0^n 1^n \ : \ n \geq 1\}$. ■

- **Definição.** Dada uma linguagem L, se existe um AP P tal que $L = L(P)$, então L é dita uma **linguagem livre de contexto**.

Gramáticas livres de contexto



Intuição

- Expressões regulares fornecem regras para construir linguagens regulares
- Gramáticas livres de contexto fornecem regras para construir linguagens livres de contexto
- **Ideia principal**
  - Começar com uma variável como S
  - Aplicar regras do tipo $S \to 00S1$ ou $S \to \varepsilon$
    - Podem haver outras variáveis também
  - Eventualmente terminar o processo quando não houverem mais variáveis
  - $S \to 00S1 \to 0000S11 \to 0000\varepsilon11 = 000011$

Definição

- **Definição.** Uma gramática livre de contexto (ou apenas gramática) é uma quádrupla $G = (V, T, P, S)$ tal que:
  - V é o conjunto de *variáveis*.
  - T é o conjunto de *símbolos terminais*.
  - P é o conjunto de *regras de produção*, sendo que cada elemento de P é uma expressão da forma $X \to W$, onde $X \in V$ e W é uma string de $(V \cup T)^*$
  - S é a variável inicial, onde $S \in V$.

Exemplo

- $G_{PAL} = (\{P\}, \{0,1\}, A, P)$ tal que os elementos do conjunto A são as seis regras de produção listadas abaixo

  $P \rightarrow \varepsilon$
  $P \rightarrow 0$
  $P \rightarrow 1$
  $P \rightarrow 0P0$
  $P \rightarrow 1P1$

- Exemplos de strings geradas por essa gramática?
- $L(G_{PAL})$?

Outro exemplo

- Podemos condensar as regras de uma gramática em uma linha por variável
- **Exemplo**: a gramática $G_{EX} = (\{I, E\}, \{a, b, 0, 1, +, *, (,)\}, A_{EX}, E)$, com as regras A_{EX} a seguir:

  $E \rightarrow I \mid E + E \mid E * E \mid (E)$

  $I \rightarrow a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

  - Essa gramática gera strings como $a * (a + b00)$

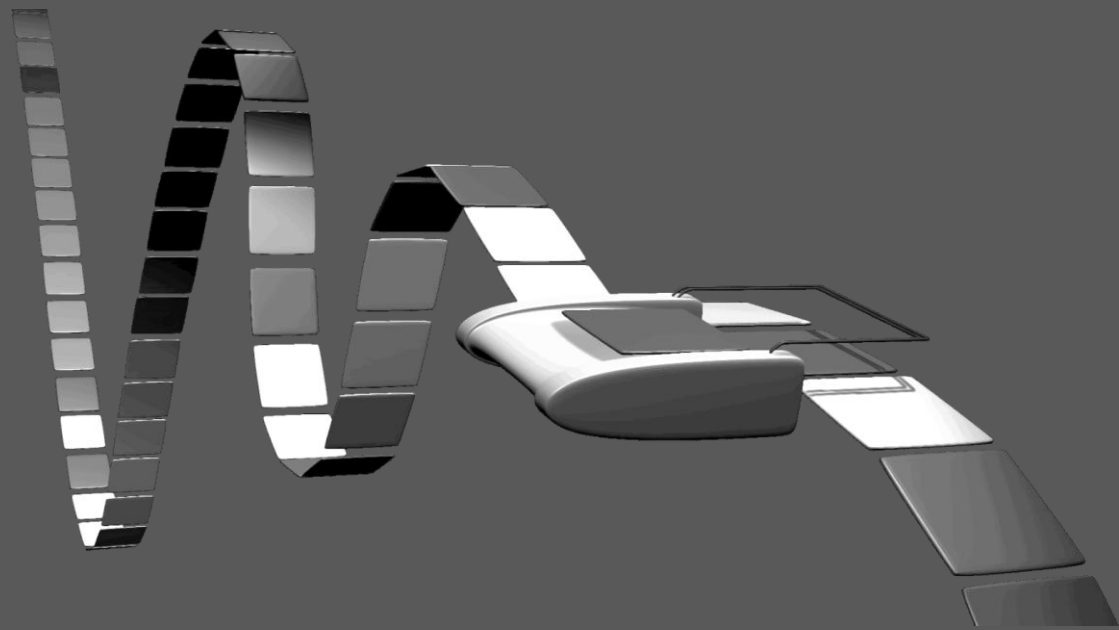
Produção de strings

- Strings deriváveis por uma gramática são aquelas que podem ser produzidas a partir de S ao aplicar as regras de produção
- Vamos definir esse conceito passo a passo
- **Definição (Produção de string – um passo).** Seja $G = (V, T, P, S)$ uma gramática e seja uma string $\alpha A \beta$, onde $A \in V$ e $\alpha, \beta \in (V \cup T)^*$. Seja $A \to \gamma$ uma regra de P. Então dizemos que a string $\alpha\gamma\beta$ pode ser *produzida em um passo* da string $\alpha A \beta$ na gramática G. Neste caso, escrevemos $\alpha A \beta \Rightarrow_G \alpha\gamma B$.
- Quando não houver ambiguidade, escrevemos $\alpha A \beta \Rightarrow \alpha\gamma\beta$

Produção de strings

- **Definição (Produção de string – múltiplos passos).** Seja $G = (V, T, P, S)$ e sejam $\alpha, \beta, \gamma \in (V \cup T)^*$ strings. Dizemos que a string γ pode ser produzida em múltiplos passos a partir de α, escrito $\alpha \Rightarrow_G^* \gamma$, se o seguinte vale:
  - **Base:** Se $\gamma = \alpha$, então vale que $\alpha \Rightarrow_G^* \gamma$.
  - **Indução:** Suponha que $\gamma \neq \alpha$ e que $\alpha \Rightarrow_G^* \beta$ e $\beta \Rightarrow_G^* \gamma$, então $\alpha \Rightarrow_G^* \gamma$.
- **Definição (Linguagem de uma gramática)**. Dada uma gramática $G = (V, T, P, S)$, a linguagem de G é $L(G) = \{w \in T^* \mid S \Rightarrow_G^* w\}$. Neste caso também dizemos que $L(G)$ é *gerada* pela gramática G.

Ambiguidade



Um exemplo

- Duas sequências que levam à $a \ * b$ em G_{EX}

  - $E \Rightarrow E + E \Rightarrow I + E \Rightarrow a + E \Rightarrow a + I \Rightarrow a + b$
  - $E \Rightarrow E + E \Rightarrow E + I \Rightarrow I + \ I \Rightarrow a + I \Rightarrow a + b$

- Cada uma das sequências é chamada de uma *derivação*

Derivação

Definição (Derivação). Seja $G = (V, T, P, S)$ uma gramática. Uma derivação é uma sequência de passos $\alpha_1 \Rightarrow \alpha_2 \Rightarrow \cdots \Rightarrow \alpha_k$ tal que $S = \alpha_1$ e cada $\alpha_i \Rightarrow \alpha_{i+1}$ é uma produção válida na gramática G.

Definição (Árvores sintáticas). Seja $G = (V, T, P, S)$ uma gramática e seja $w = w_1 w_2 \ldots w_k$ uma string de $L(G)$. Seja $\mathcal{D}$ uma derivação de w em G. A *árvore sintática da derivação* $\mathcal{D}$ é uma árvore com raiz S e as seguintes relações entre nós:

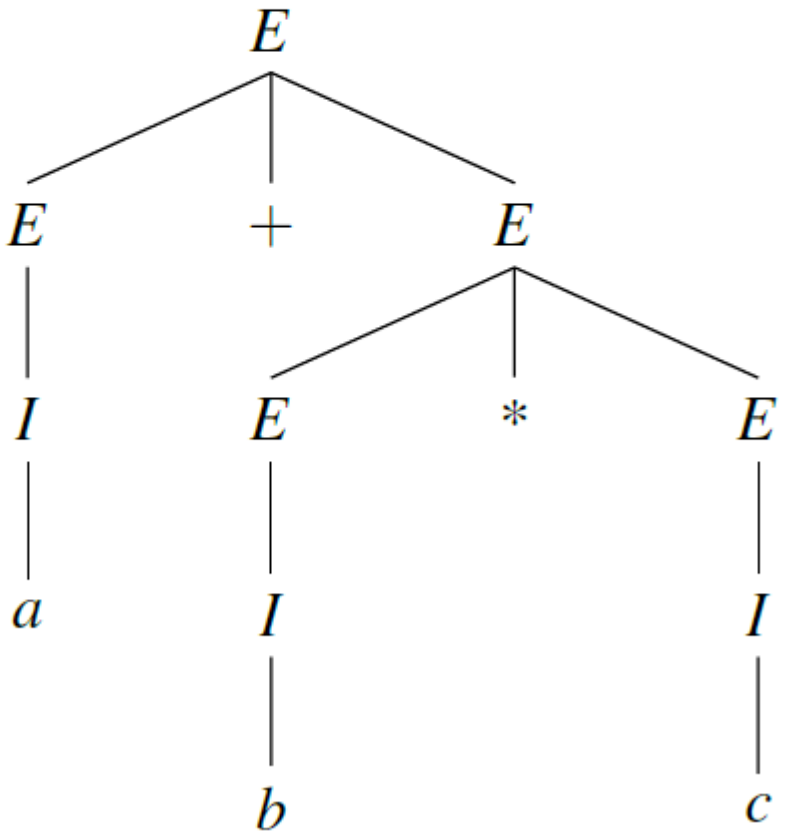
- Se o passo $\alpha A \beta \Rightarrow \alpha\gamma\beta$ aparece na derivação $\mathcal{D}$ e $\gamma = \gamma_1\gamma_2 \ldots \gamma_\ell$, então o nó A é pai dos nós $\gamma_1\gamma_2 \ldots \gamma_\ell$ da esquerda para a direita.
- O conjunto das árvores sintáticas de todas as possíveis derivações de w é chamado de conjunto das árvores sintáticas de w.

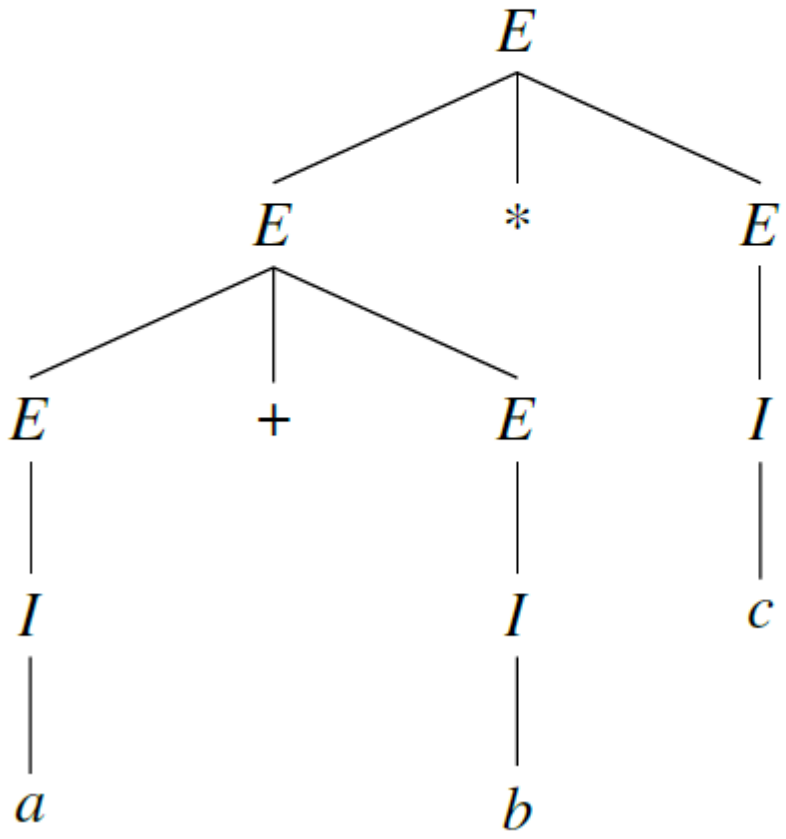
■ **Exemplo 4.8** Condidere abaixo duas possíveis derivações da string $a+b*c$:

$(i)\ E \Rightarrow E+E \Rightarrow E+E*E \Rightarrow E+E*I \Rightarrow E+E*c \Rightarrow I+E*c \Rightarrow a+E*c \Rightarrow a+I*c \Rightarrow a+b*c$

$(ii)\ E \Rightarrow E*E \Rightarrow E+E*E \Rightarrow E+E*I \Rightarrow E+E*c \Rightarrow I+E*c \Rightarrow a+E*c \Rightarrow a+I*c \Rightarrow a+b*c$

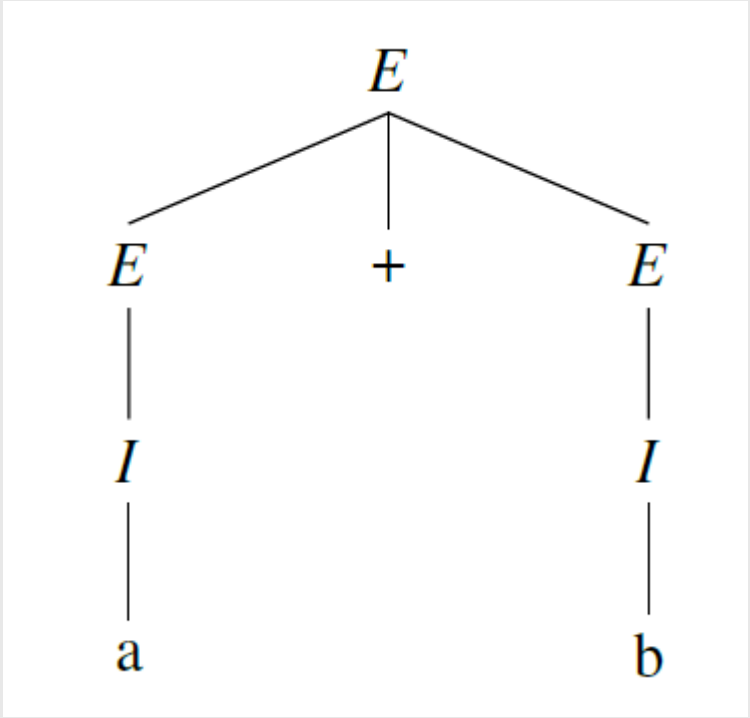
Abaixo à esquerda a árvore sinática da derivação *(i)* e à derivação de *(ii)*.





Voltando ao primeiro exemplo

- Duas derivações para $a * b$
  - $E \Rightarrow E + E \Rightarrow I + E \Rightarrow a + E \Rightarrow a + I \Rightarrow a + b$
  - $E \Rightarrow E + E \Rightarrow E + I \Rightarrow I + I \Rightarrow a + I \Rightarrow a + b$
- As duas possuem a mesma árvore sintática



Ambiguidade

- **Definição.** Uma gramática G é dita ambígua se existe mais de uma árvore sintática para alguma string $w \in L(G)$.
- Concluímos que G_{EX} é ambígua
- Multiplicidade de derivações versus ambiguidade
- **Teorema.** Seja G uma gramática. Toda string de $L(G)$ tem exatamente uma derivação mais à esquerda se e somente se G não é ambígua.
- **Teorema.** Seja G uma gramática. Toda string de $L(G)$ tem exatamente uma derivação mais à direita se e somente se G não é ambígua.

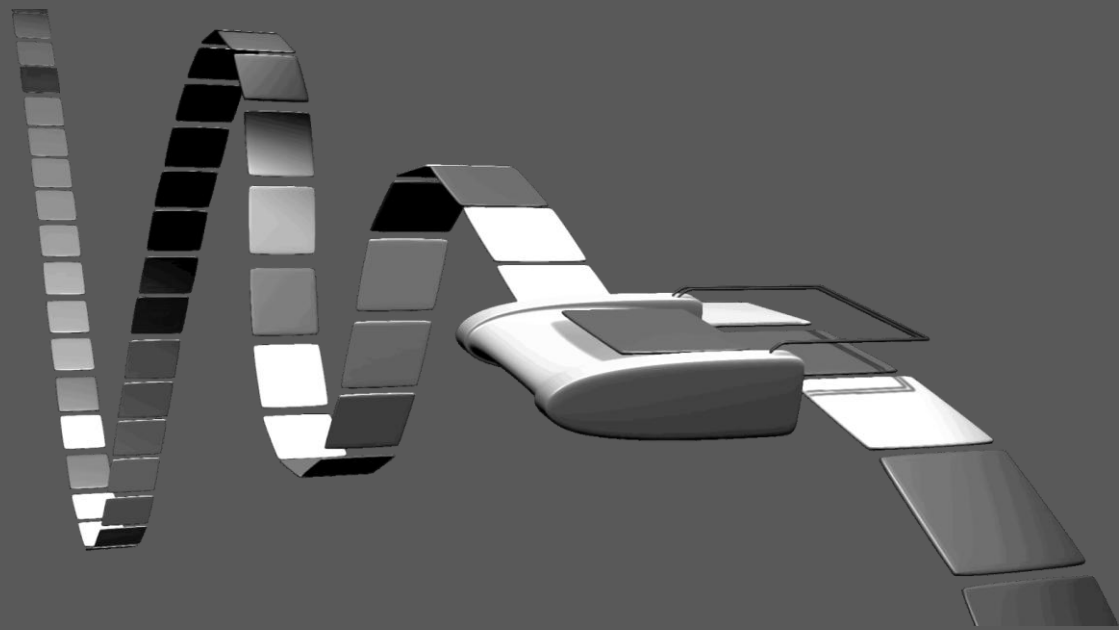
Um pouco mais sobre ambiguidade

- Existem linguagens que são *inerentemente ambíguas*
  - Qualquer gramática para a linguagem é ambígua
  - Exemplo

$$L = \{ a^n b^n c^m d^m \mid n > 0,\ m > 0 \} \cup \{ a^n b^m c^m d^n \mid n > 0,\ m > 0 \}$$

- **Teorema.** Se $L = L(P)$ para algum APD P, então existe uma gramática livre de contexto não ambígua tal que $L(G) = L$.
- **Teorema.** Existe uma gramática livre de contexto não ambígua G tal que não existe nenhum APD que aceite $L(G)$.

Para fechar



Para fechar

- Hoje
  - Gramáticas livres de contexto
    - Definem regras para produzir strings
    - Equivalentes a APs
- Próxima aula
  - Revisão para a avaliação

Referências

- [SIL25] – SILVA, M.V.G. **Autômatos, Computabilidade e Complexidade Computacional** , 2025.