

RESUMO CAPÍTULO 3 e 4

UNIC – Universidade de Cuiabá

Curso: Ciências da Computação

Disciplina: Linguagens Formais e Autômatos

Professor: Felipe Douglas

MATHEUS HENRIQUE GODOY LEITE

CUIABÁ - MT

Capítulo 3 – Linguagens e Gramáticas Livres do Contexto e Autômatos com Pilha

O capítulo explica como as **gramáticas livres de contexto** e os **autômatos com pilha** expandem o poder dos modelos regulares, permitindo representar estruturas hierárquicas.

Esses conceitos são fundamentais para **interpretação de linguagens formais, XML, linguagens de programação e análise sintática**.

Seção 3.1 – Gramáticas Livres de Contexto

As **gramáticas livres de contexto (GLC)** são sistemas de regras capazes de gerar linguagens mais complexas do que as regulares.

Em uma GLC, todas as regras têm a forma $A \rightarrow w$, onde A é uma variável e w é uma sequência de variáveis e/ou terminais.

Elas são usadas para descrever estruturas com dependências hierárquicas, como **parênteses balanceados** e **sintaxe de linguagens de programação**.

Exemplo:

$S \rightarrow aSb \mid ab$

gera as cadeias **ab, aabb, aaabbb**, etc., representando igualdade entre o número de a's e b's.

Nem toda linguagem livre de contexto é regular, pois GLCs têm maior poder expressivo. Elas são muito utilizadas em compiladores e interpretadores de linguagens.

Seção 3.2 – Linguagens Livres de Contexto

As **linguagens livres de contexto (LLC)** são aquelas geradas por uma GLC. Possuem propriedades importantes, como **fechamento por união e concatenação**, mas **não são fechadas por interseção ou complemento**.

O estudo dessa seção mostra como modificar GLCs para adaptar-se a diferentes estruturas, por exemplo, arquivos **XML** com sintaxe variada.

Essas linguagens permitem a modelagem de **estruturas aninhadas**, como tags de abertura e fechamento em arquivos ou sentenças com múltiplos níveis de dependência.

Seção 3.3 – Autômatos com Pilha (AP)

O **autômato com pilha (AP)** é o **reconhecedor** das linguagens livres de contexto — o equivalente do **AFD** para as linguagens regulares.

Ele funciona com uma **memória auxiliar (pilha)** que permite armazenar símbolos e verificar correspondências, como abertura e fechamento de estruturas.

Há dois tipos:

- **AP não determinístico (APND)** – mais geral e usado em teoria.
- **AP determinístico (APD)** – usado em compiladores (analisadores sintáticos).

O AP pode ser construído a partir de uma GLC e vice-versa, e seu funcionamento é base para **análise sintática** em compiladores e processadores de linguagens.

Capítulo 4 – Linguagens Sensíveis ao Contexto e Recursivamente Enumeráveis

O capítulo mostra que as **gramáticas sensíveis ao contexto** e as **máquinas de Turing** representam o **nível mais alto da hierarquia de linguagens formais**. Ele introduz conceitos centrais da **computabilidade** e dos **limites da automação**, fundamentais para entender como e até onde a computação pode ir.

Seção 4.1 – Linguagens Sensíveis ao Contexto

As **gramáticas sensíveis ao contexto (GSC)** são mais poderosas que as livres de contexto. Suas regras possuem a forma $\alpha \rightarrow \beta$, com $|\alpha| \leq |\beta|$, ou seja, o lado direito nunca é menor que o esquerdo. Isso garante que o processo de derivação **nunca diminui o tamanho da cadeia**.

Toda GLC pode ser transformada em uma GSC, mas o contrário não é verdadeiro. Um exemplo clássico é a linguagem $L = \{a^n b^n c^n \mid n \geq 0\}$, que **não é livre de contexto**, mas **é sensível ao contexto**.

Essas linguagens são **fechadas sob união e concatenação** e podem ser reconhecidas por **máquinas linearmente limitadas (ALL)** — versões restritas das máquinas de Turing.

Seção 4.2 – Máquinas de Turing

A **máquina de Turing (MT)** é o modelo teórico mais poderoso da computação, equivalente ao que um **computador moderno pode fazer**. Ela lê e escreve símbolos em uma fita infinita, podendo mover-se para frente e para trás.

As MTs servem para:

- Definir o que é **computável**.
- Estudar os limites da computação (problemas **decidíveis** e **indecidíveis**).
- Demonstrar que certas linguagens não podem ser reconhecidas por nenhum autômato finito.

A MT é a base para conceitos como **decidibilidade**, **funções recursivas** e a **tese de Church-Turing**, que afirma que tudo que pode ser calculado por um algoritmo pode ser feito por uma máquina de Turing.

Seção 4.3 – Linguagens Recursivas e Recursivamente Enumeráveis

As **linguagens recursivas** são aquelas para as quais existe uma máquina de Turing que **sempre termina e responde “sim” ou “não”** sobre a pertinência de uma cadeia.

Já as **linguagens recursivamente enumeráveis (LRE)** são aquelas para as quais existe uma máquina de Turing que **gera todas as cadeias da linguagem**, mas pode **nunca parar** se a entrada não pertence à linguagem.

Nem toda linguagem é recursiva ou recursivamente enumerável.

Essas classes de linguagens estão relacionadas a **problemas computáveis**, **indecidíveis** e **paradoxos** clássicos, como o **problema do barbeiro** e o **problema da parada**.