

**Universidade Federal do Rio Grande do Norte**  
**Departamento de Informática e Matemática Aplicada**  
**DIM0548 - Engenharia de Linguagens**

**Problema:** projetar e implementar uma linguagem de programação.

**Subproblema 7:** segunda fase do compilador - Análise Sintática: aprendizado das técnicas de parsing *bottom-up*, mediante a construção de um analisador sintático para a linguagem definida pelo grupo.

**Produtos deste problema:**

- *Implementação do analisador sintático bottom-up (LALR) da linguagem do grupo;*
- *Programas de teste abrangendo todos os construtores sintáticos da linguagem;*
- *Manual de uso do analisador produzido.*

**Instruções:** consulte o capítulo 3 do livro *Modern Compiler Implementation in C* (APPEL, Andrew W.) e resolva os exercícios propostos associados a este problema.

**Data de entrega via SIGAA:** 05/11/2025.

**Questões:**

“O parsing deve ser eficiente, pois caso contrário o tempo de compilação para programas de grande porte faria impraticável a própria existência da linguagem de programação.”

1. A frase acima é o motivo do desenvolvimento de técnicas eficientes para o parsing. Uma dessas técnicas é o parsing LR(1). Que significa a sigla LR(1)? OBS: Consultar o livro de Appel e Palsberg para uma resposta concisa a esta pergunta.
2. Os algoritmos de parsing LR mais comuns são: SLR, LR(1) e LALR. Quais são as diferenças entre eles?
3. Yacc é uma ferramenta para gerar analisadores sintáticos LALR. Qual é a sintaxe de um programa Yacc?
4. Implementar o parser LALR para a sua linguagem, usando Yacc.
5. Qual é o resultado obtido pelo seu parser para os programas de teste?