

## Descrição do Projeto Prático

Braulio Mello  
Última atualização: 15-08-19

### Objetivo:

Projeto e implementação das etapas de compilação para uma linguagem de programação hipotética.

### Etapas:

- (a)Análise Léxica (0,2);
- (b)Análise Sintática (0,5);
- (c)Análise Semântica (0,1);
- (e)Geração de código intermediário (0,1);
- (e)Otimização (0,1).

As etapas (a) e (b) devem ser implementadas na íntegra. Para as demais etapas, é suficiente a implementação de uma demonstração para uma característica semântica, para a geração de código intermediário de uma estrutura sintática, e para demonstrar um procedimento de otimização.

### Requisitos:

Gerais:

A especificação da linguagem pode ser baseada nas linguagens convencionais ou adicionar características diferenciadas (estrutura sintática, tokens, características semânticas, etc).

Etapa (a):

- definir conjunto de tokens;
- implementar reconhecedor léxico;
- o reconhecedor deve gerar informações na tabela de símbolos para uso nas etapas subsequentes; p.ex: rótulo dos identificadores
- entrada: arquivo com tokens (cadeia ou GR, conforme trabalho feito em LFA);
- saída: fita com id de cada token reconhecido, ou erro, terminada no marcador de final de sentença \$, e tabela de símbolos.
- se fita de saída conter estados de erro, emitir mensagens de erro com linha de ocorrência e rótulo da cadeia que provocou o erro;

Etapa (b):

- construir as regras sintáticas da linguagem: GLC;
- aplicar eliminação de inúteis e fatoração na GLC;
- Construção do conjunto de itens válidos, transições e follow
- Construção da tabela de parsing SLR ou LALR
  - SLR Parser Generator (<http://jsmachines.sourceforge.net/machines/slr.html>)
  - GoldParser (<http://www.goldparser.org/>)
- Implementação do algoritmo de mapeamento da tabela para reconhecimento sintático
- Gerenciamento da tabela de símbolos (suporte para as próximas etapas)
- Integrar procedimentos (ações semânticas – TDS) para os seguintes itens:
  - tratamento de erros (pelos menos o tipo de erro e linha de ocorrência);
  - adição de atributos ao símbolos (id de tokens reconhecidos);
  - valores nos atributos;

- entrada: fita de saída do reconhecedor léxico;
- saída: aceite ou mensagem(ns) de erro(s) sintático(s);
- agragar informações na tabela de símbolos para uso nas etapas subsequentes.

Etapas (c):

- definir uma característica semântica cujas informações adicionadas na TS sejam suficientes para a análise;
- implementar a análise semântica da característica especificada;
- entrada: código / tabela de símbolos
- saída: aceite ou mensagem(ns) de erro(s) semântico(s);

Etapas (d):

- cada redução de uma produção (reconhecimento de uma estrutura sintática) é seguida de uma chamada a uma ação que gera o correspondente trecho de código intermediário com os respectivos rótulos dos identificadores, constantes, etc.
- basta aplicar a geração de código intermediário para uma das regras sintáticas (demonstração);
- entrada: reconhecimento de uma estrutura sintática conforme regras de produção.
- saída: código intermediário;

Etapas (e):

- aplicar uma estratégia de otimização de código sobre o código intermediário gerado na etapa anterior
- entrada: código intermediário
- saída: código intermediário otimizado

### **Texto:**

Elaborar texto técnico (formato de artigo) entre 4 e 8 páginas contendo:

- Título, autores e instituição
- Resumo: breve apresentação do teor do texto
- Introdução: Contextualização sobre reconhecedores léxicos (propósitos, aplicação e características), apresentação de problema e objetivo do trabalho.
- Referencial teórico: Breve explanação sobre os conceitos, técnicas e/ou teoremas fundamentais para o desenvolvimento do trabalho.
- Implementação e resultados: Apresentação dos detalhes de especificação do analisador léxico (tokens, decisões de projeto, GLC, etc), implementação e validação.
- Conclusões: O que foi feito, dificuldades, resultados finais e perspectivas para continuidade do trabalho (por exemplo, sugerir alterações futuras para utilizar a implementação no ensino de compiladores).

### **Apresentação:**

- Trabalho individual ou em duplas
- Aplicação em funcionamento: 50% da nota
- Apresentação (demonstração da aplicação e arguição): 50% da nota
- Resultados mínimos para que o trabalho possa ser apresentado: etapas (a) e (b) concluídas e relatório no formato requerido (artigo).
- Qualidade da solução, requisitos contemplados, domínio do processo de especificação e implementação da aplicação, teor/clareza/conteúdo do artigo são os principais referenciais para composição da nota.