

Projetos – Compiladores 2018-2

Franklin Ramalho

Professor DSC-UFCG



Motivação

- Projeto da disciplina permite:
 - Fixar e usar conceitos, técnicas e algoritmos estudados em sala de aula;
 - Conhecer e usar linguagens e ferramentas criadas para auxiliar a construção de compiladores;
 - Trabalhar em equipe;
 - Conhecer novas linguagens de programação;
 - Revisar e usar conceitos relativos às linguagens de programação;
 - Conhecer linguagem de máquina;
 - Construir um compilador!
 - Tornar-se um profissional mais capacitado e mais completo!

Etapas

- Construir compiladores
 - Análise Léxica (Completa)
 - Análise Sintática (Restrita ao escopo)
 - Análise Semântica (Restrita ao escopo)
 - Gerador de Código (Restrito ao escopo)

Linguagens

- Equipes:
 - 04 aluno(a)s
- Linguagem Fonte:
 - Pascal
- Linguagem Destino: *Assembly (versão simplificada estudada em sala de aula)*

Ferramentas

- Serão usadas duas linguagens/ferramentas:
 - JFLEX/CUP
 - Xtext
- Serão indicadas as preferências, mas caso não haja balanceamento, serão escolhidos de forma aleatória e balanceada (via sorteio)
 - Linguagem/Ferramenta

Análise

- A etapa de Análise Léxica e Sintática corresponderá à:
 - Analisador léxico;
 - Analisador sintático;
 - Entrega de analisador funcionando;
 - Uso de linguagens/ferramentas (Jflex/CUP ou Xtext);
 - Eliminar eventuais problemas das gramáticas;
 - O escopo de cada projeto vai limitar, nas gramáticas, os comandos e expressões a serem tratados.

Análise

- Todos os projetos deverão, cobrir, sintaticamente, além dos comandos e expressões do seu escopo:
 - Abstrações (funções e procedures)
 - Declaração e uso de Variáveis;
 - Comandos de atribuição;
 - Expressões aritméticas;
 - literais (inteiros, string, booleanos).

Em todas as etapas de Análise

- Detectar e informar todos os tipos de erros!
 - Apresentar mensagens precisas sobre o erro ocorrido (caso ocorram);
 - Indicar linhas de erro do código.
- Construir e manipular a Tabela de Símbolos

Análise Semântica

- O sistema de tipos da linguagem precisa ser bem estudado
 - Essenciais para análise semântica e geração de código!
- Os contextos precisam ser conhecidos
 - Programas, variáveis, etc.;
 - Abstrações de funções e procedimentos, comandos, etc.;
 - Essenciais para análise semântica e geração de código!
- As restrições da linguagem fonte precisam ser conhecidas e respeitadas

Análise Semântica – Escopo comum

- Todos os projetos deverão:
 - Realizar checagem de tipos e contextos:
 - Tipos existentes;
 - Abstrações (nome, quantidade e tipos de parâmetros de entrada e tipo de retorno);
 - Declaração e uso de Variáveis;
 - Comandos de atribuição;
 - Expressões aritméticas;
 - literais (inteiros, string, booleanos).

Síntese - Escopo comum

- Todos os projetos deverão:
 - Gerar código para:
 - Abstrações (nome, quantidade e tipos de parâmetros de entrada e tipo de retorno);
 - Declaração e uso de Variáveis;
 - Comandos de atribuição;
 - Expressões aritméticas;
 - literais (inteiros, string, booleanos).

Análise Semântica e Síntese

- Porém, Escopos para análise sintática, semântica e síntese:
 - Escopo A
 - Registros
 - Expressões relacionais
 - Comandos condicionais: *if*
 - Escopo B
 - Sets
 - Expressões booleanas
 - Comandos condicionais: *if*
 - Escopo C
 - Arrays
 - Expressões booleanas
 - Comandos condicionais: *case*

Análise Semântica e Síntese

- Escopo D
 - Registros
 - Expressões relacionais
 - Comandos condicionais: *case*
- Escopo E
 - Sets
 - Expressões relacionais
 - Comandos iterativos: *for*
- Escopo F
 - Arrays
 - Expressões booleanas
 - Comandos iterativos: *for*
- Escopo G
 - Registros
 - Expressões booleanas
 - Comandos iterativos: *while*

Análise Semântica e Síntese

- Escopo H
 - Sets
 - Expressões relacionais
 - Comandos iterativos: *while*
- Escopo I
 - Arrays
 - Expressões relacionais
 - Comandos iterativos: *repeat*
- Escopo J
 - Registros
 - Expressões booleanas
 - Comandos iterativos: *repeat*

Análise Semântica

- Se alguma análise complementar precisar ser feita (para viabilizar as demais análises), ela deverá ser feita, mesmo que não esteja contemplada aqui na especificação.

Síntese

- A etapa de geração de código também terá escopo reduzido semelhante às etapas de análise sintática e semântica (mesmo escopo para as 03 etapas).
 - Maiores detalhes serão dados após a entrega da etapa de análise semântica.

Síntese

- Se alguma geração complementar precisar ser feita (para viabilizar as demais gerações), ela deverá ser feita, mesmo que não esteja contemplada aqui na especificação.

Análise Semântica e Síntese

- As etapas de análise semântica e geração de código devem ser feitas através de **ações semânticas**, incorporadas na própria gramática da linguagem fonte!
 - JFlex + CUP
 - Xtext
- Se o compilador gerar código para exemplos com erros (léxicos, sintáticos ou semânticos), a etapa de geração de código será penalizada também.
 - Análise bem feita é essencial!

Prazos e entregas

- Análise léxica/Sintática: (30%): 25/10
- Análise Semântica (35%): 13/11
- Gerador de Código (35%): 29/11
- As etapas de análise são obrigatórias. Sem elas, a nota final do projeto será ZERO.
 - Ou seja, quem não entregar os analisadores, não poderá entregar o gerador de código!
 - Até porque a geração de código deverá ser feita com ações semânticas!

Projetos

- Equipe: (04)
- Nota: Individual
- Compõe 50% da terceira nota da disciplina
- Projetos poderão ser cobrados em mini-provas!
- Serão solicitadas mudanças nos projetos no momento da defesa!

Projeto

- Mais que viabilizar uma boa nota, o projeto permite ao aluno exercer suas habilidades dentro de sua profissão
- O aprendizado com projetos é enorme!
- Desbravar linguagens e ferramentas faz parte do seu dia-a-dia
- Você estará praticando sua profissão!

