MC911 - Projeto em Compiladores

1o. Semestre de 2014

Turmas A e B

Avisos

24/03
. Link de Submissão do Projeto 1: SubP1 . Encerra-se dia 26/03, às 14h00. Leiam detalhes da submissão no enunciado.

19/03
. Início da entrega do Projeto 1 no SuSy. Encerra-se dia 26/03, às 14h00. Leiam detalhes da submissão no enunciado.
. Pacote v01 para Projeto 2 disponível aqui . Sempre que este pacote receber uma atualização será comunicado via grupo.

17/03
. Entrega do Projeto 1 prorrogada para o dia 26/03. Terá um bonus de 10% para quem entregar dia 19/03.

13/03
. Trabalho de Laboratório 01 disponível no SuSy.
. Código fonte do trabalho SQL SELECT: parser-didatico-sql-select .

09/03
. Formulário de auxílio para a formação de duplas: Formulario para alunos sem dupla
. Tabela com os alunos que preencheram o formulário: Alunos sem dupla definida .

06/03
. Formulário para cadastro das duplas: Formulario cadastro duplas .
. Tabela com as duplas cadastradas: Duplas sadastradas .

26/02
. Página da disciplina no ar. Confira critérios de avaliação e calendário.
. Códigos FLEX e BISON feitos hoje em sala: teste1 e parser-didatico-sql .

Aulas

Turmas A e B
Qua: 14-18h, salas IC 302 e 303

Professores

Contato

· Professor: Sandro Rigo (sandro AT ic dot unicamp dot br)
· Monitor: Maxiwell Garcia (maxiwell AT ic dot unicamp dot br)
· OBS.: Quando enviar um e-mail favor colocar no subject [MC911], caso contrário você corre sério risco de seu email ser filtrado como spam.

Avaliação

Avaliação

Teremos três projetos a serem elaborados (descrição, data de entrega e pesos abaixo) ao longo do semestre. Cada entrega será composta de uma avaliação presencial e da entrega do pacote com o código fonte desenvolvido. A nota de cada etapa será composta por Pi = 0,3*(AP)+0,7*(I). Onde AP é a avaliação presencial e I é a implementação. A avaliação presencial é obrigatória a todos os alunos. A não participação nesta avaliação implica em nota zero na etapa correspondente. Além disso, existirão exercícios de desenvolvimento aplicados em aula. Esses exercícios deverão ser realizados e entregues durante o período da aula. A média do desempenho será calculada por:
M = 0,25 * P1 + 0,35*P2 + 0,25*P3 + 0,15 EX
Se M >= 5,0 o aluno aprovou-se. Caso contrário o aluno estará reprovado.
A entrega de todos os projetos é obrigatória. Nota menor do que 2,5 em qualquer um dos projetos acarreta automaticamente a reprovação com média final MF = Min (4,0; M).

Fraudes

Qualquer tentativa de fraude implicará em nota ZERO no projeto correspondente, reprovando automaticamente todos os envolvidos.

Calendário

- 19/03: Entrega e Avaliação presencial do Projeto 1
- 30/04: Entrega e avaliação presencial do Projeto 2
- 11/06: Entrega e avaliação presencial do Projeto 3

Projetos

== Projeto 1 ==

Especificação dos tokens, da gramática livre de contexto e implementação de um parser para uma linguagem de alto nível chamada "News Publication Language (NPL)". Esta linguagem é usada para geração de código HTML para um site de notícias.
O trabalho deve ser realizado em duplas usando a ferramenta Flex e Bison.

O arquivo sempre_online.npl (em HTML com ***syntax highlight***: sempre_online.npl.html) apresenta a especificação da linguagem NPL através de um exemplo. Nele aparecem as estruturas válidas da linguagem e as palavras reservadas. Aos usuários de VIM, aqui encontra-se um arquivo de

syntax highlight da linguagem NPL.
O objetivo final, é usar uma descrição em NPL para gerar um site de notícias como este jornal. Note que a formatação usada neste site é apenas ilustrativa. Cada grupo deve escolher o estilo de formatação que lhe pareça mais adequado, inclusive aplicando o uso de CSS.
Cada grupo pode propor extensões para a linguagem NPL, visando torná-la mais poderosa ou flexível em termos de definição e apresentação do conteúdo do jornal. A nota final do trabalho será composta de 70% para a implementação correta do parser e o cumprimento total da especificação, 30% para a qualidade visual e de organização do site produzido, e as extensões como bônus extra na nota. A especificação da linguagem NPL pode ser encontrada em sempre_online.npl.
Atenção especial às possibilidades de formatação do texto inserido nos campos dos objetos. Essa formatação deve usar o padrão de códigos do wiki da Wikipedia (link). O programa sempre_online.npl usa alguns desses códigos de formatação no campo "text".

Submissão
Clique aqui para submissão. Detalhes do envio aqui .

== Projeto 2 ==

Este segundo projeto abordará o tópico de geração de código.

Teoria
A parte teórica da necessária para a implementação do laboratório foi vista no curso de MC910, e pode ser revista nos slides da disciplina ou capítulo 9 do Appel (2a edição). É importante ressaltar que conhecimentos sobre Árvores de representações intermediária, também vistos na disciplina MC910 e contidos no capítulo 7 do Appel (2a edição), são necessários.

Descrição
Você deverá implementar uma classe na linguagem Java que recebe uma lista de árvores e sobre elas executa algoritmo de seleção de instruções (maximal munch) e devolve uma lista com as instruções referentes a essas árvores. Para mais detalhes sobre os nós que podem estar nessas árvores, você pode verificar o livro do Appel (2a edição) na seção 7.1. As instruções deverão ser instruções assembly para a representação intermediária do compilador LLVM. Note que a arquitetura é diferente da arquitetura utilizada pelo livro do Appel, portanto a parte da implementação do livro (seção 9.3) não necessariamente diz respeito ao que deve ser implementado no projeto (porém pode ajudar a entender melhor o que fazer).

Material
Pacote de apoio: minijava-llvm-v01 . Sempre que este pacote receber uma atualização será comunicado via grupo. É possível conferir a versão do pacote na primeira linha do Makefile.

Submissão
Em breve maiores detalhes.

== Projeto 3 ==

O objetivo desta parte é implementar a otimização Dead Code Elimination (DCE), descrito no livro do Appel no capítulo 17, pág 360, como um passo do compilador LLVM . EM breve discutiremos em aula e colocaremos instruções de como um passo para LLVM deve ser implementado.

Submissão
Em breve maiores detalhes.

Links

Referências Principais

Notas de Aula

Modern Compiler Implementation in Java
Andrew Appel, 2a Edicão

Compiladores : Princípios, Técnicas e Ferramentas
Aho, Sethi & Ullman

Implementação de Linguagens de Programação
Kowaltowski, Editora Guanabara Dois, 1983.