

Trabalho 4 - Análise semântica e geração de código

Implementação

Este é o ununciado da última parte do trabalho, que trata sobre a geração do código intermediário LLVM. A análise semântica produz uma árvore sintática abstrata anotada (ASTO). A tabela de símbolos produzida por meio desta árvore anotada e a própria estrutura da árvore deve ser utilizada para a construção do código intermediário.

Na fase de geração de código intermediário (RI), a ASTO deverá ser percorrida novamente, classificando assim como a “terceira passada” ou a “segunda passada” do compilador, como foi comentado em sala de aula, para geração da RI. A linguagem alvo irá gerar a RI da LLVM [1][2][3][4] e todas as estratégias para a geração do código são apresentadas em sala de aula, além de materiais de apoio postados no moodle.

Ao criar o código para a chamada de função, o tipo dos parâmetros reais deve ser comparado com o tipo dos parâmetros formais. Essa análise é possível pois os parâmetros formais estão declarados na tabela de símbolos. Um erro/aviso deste tipo deverá ser do tipo semântico.

É esperado no fim desse trabalho um código IR gerado em LLVM *otimizado*. Para a realização de otimizações, deverá somente aplicar algumas configuração no ambiente de geração de código (API) do LLVM. Tais considerações serão apresentadas durante a aula.

Documentação

Durante toda a disciplina o aluno criará uma documentação formal da implementação da linguagem. Neste ponto do trabalho o aluno deverá incluir na documentação:

- Explicar se houve alguma estratégia diferenciada para a geração do código IR LLVM;
- Explicar a(s) otimizações utilizadas na linguagem de programação; e
- Explicar como o compilador deverá ser executado.

Avaliação

Será avaliado se o compilador realiza todas as fases de compilação corretamente para um exemplo simples escrito pelo professor. O essencial da linguagem deve ser mantido (conforme especificado nos trabalhos anteriores) para que o mesmo exemplo funcione para testar todos os trabalhos com o mínimo de alteração.

Entrega e apresentação

O trabalho será **individual** e deverá ser entregue até o dia 06/12/2016 no moodle da disciplina em um pacote compactado e apresentado no mesmo dia. Não haverá extensão de tempo, então se organizem para fazer os trabalhos completamente.

Referências

1. Site oficial do LLVM
2. Site da API escrita Python para LLVM
3. Tutorial Kaleidoscope escrito em C
4. Tutorial não oficial utilizando a API em C para LLVM