

Trabalho Prático - Especificação da Etapa 7: Recuperação de Erros e Otimização

Resumo:

Na sétima etapa do trabalho de implementação de um compilador pode-se que sejam implementados duas novas funções, recuperação de erros e otimização de código. Em ambos os casos, a definição permite que você escolha quais mecanismos irá implementar. Opcionalmente, os alunos são incentivados a fazerem uma proposta de implementação de algum outro recurso significativo adicional após todo o seu trabalho estar funcionando de acordo com as especificações até a etapa6, valendo como alternativa para etapa7. Sugere-se neste caso obter aprovação prévia do professor para garantir que a alternativa desenvolvida seja suficiente em termos de conteúdo e complexidade.

Tarefas necessárias:

- a. Recuperação de erros – você deve incluir novas regras na descrição sintática para o *yacc*, usando o *token error*, e ações com mensagens de erro sintático adequadas, de forma a demonstrar que o analisador assim implementado consegue identificar, recuperar-se e informar múltiplos erros de sintaxe em apenas uma chamada. A etapa de verificação semântica deve continuar operacional, mas na presença de qualquer erro sintático, as etapas seguintes, de geração de código, devem ser desabilitadas;
- b. Otimização de código – Nesta parte você deve selecionar ao menos uma técnica de otimização de código, que pode ser sobre código intermediário ou sobre o código assembly e demonstrar seu funcionamento. A escolha da(s) técnica(s) é livre, mas para obter nota máxima nesta etapa, pede-se que você consiga não somente mostrar que o código gerado ficou diferente (melhorado, por exemplo, mais curto), mas verificar experimentalmente que isso gerou otimização no tempo de execução do programa;

Controle e organização do seu código fonte

Você deve seguir as mesmas regras das etapas anteriores para organizar o código, permitir compilação com **make**, permitir que o código seja rodado com **./etapa7**, e esteja disponível como **etapa7.tgz**. Esta etapa, assim como a etapa6, deve ser acompanhada de um pequeno relatório com demonstração do funcionamento e efeitos dos recursos implementados, com *logs* ou *prints* de tela. Os alunos devem estar preparados para demonstrarem o funcionamento do trabalho na máquina que usaram para desenvolvimento, de forma remota, em horários a serem combinados com o professor, individualmente ou em grupo, devido a incompatibilidade entre linguagens assembly e máquinas de diferentes sistemas.