

COMPILADORES

Segunda Chamada - 10/12/2010 - Prof. Marcus Ramos

1. (0.9 ponto) Explique em que consiste, como funcionam e quais as principais diferenças entre os métodos de análise sintática descendente e ascendente.
2. (0.9 ponto) Defina gramática LL(1).
3. (0.8 ponto) Por que é importante que a gramática da linguagem-fonte seja modificada para se tornar LL(1) antes de se começar a construir um reconhecedor sintático top-down para a mesma?
4. (0.8 ponto) Defina escopo e tempo de vida de uma variável numa linguagem de programação.
5. (0.8 ponto) De que forma as regras de escopo e de tipos determinam as ações que devem ser executadas pela fase de análise de contexto num compilador típico?
6. (0.8 ponto) Seja A uma linguagem com tipos estáticos e B uma linguagem com tipos dinâmicos. Descreva como pode (ou deve) ser feita a verificação de tipos nos processadores para cada uma dessas linguagens.
7. (0.8 ponto) Descreva a tabela de símbolos como um tipo abstrato de dados, explicando o seu uso dentro do compilador e as operações que são executadas sobre a mesma.
8. (0.8 ponto) Descreva como é feita a alocação de variáveis estáticas, automáticas e dinâmicas numa linguagem de programação típica. Como são determinados os endereços de variáveis alocadas em cada uma dessas três formas?
9. (0.8 ponto) Descreva a estrutura de um "frame", a sua função num ambiente típico de execução e para que servem cada um das informações nele armazenadas.
10. (0.8 ponto) Explique como é feita a geração de código para acessar os elementos de uma matriz n-dimensional em tempo de execução, incluindo a verificação dos índices correspondentes.
11. (0.9 ponto) Descreva como são representados e acessado os parâmetros de subprogramas em tempo de execução, incluindo passagem por valor e passagem por endereço.
12. (0.9 ponto) Descreva como é feito o retorno do valor de uma função para o ambiente do bloco chamador usando a estrutura de frames do ambiente de execução.