

COMPILADORES

Prova 2
19/10/2021
Prof. Marcus Ramos

Observações:

- A prova será feita no horário da aula (entre 10:00h e 12:00h);
- A prova é individual;
- Qualquer suspeita de cópia ou ajuda indevida implicará a atribuição da nota zero à prova;
- Ao término da prova (12:00h) as respostas deverão ser enviadas por email para `marcus.ramos@univasf.edu.br` com o título "C20212-P2-<nome>", onde <nome> deverá ser substituído pelo seu nome; não usar aspas no título da mensagem;
- Em caso de mais de um envio, apenas o primeiro será considerado;
- Os emails deverão ser recebidos até 12:30h; será emitida uma confirmação de recebimento para cada email enviado;
- Emails não recebidos até 12:30h não serão considerados.

1. A montagem da AST pelo analisador sintático é importante pois:
 - a) Não é possível determinar se a cadeia pertence à linguagem sem ela;
 - b) Não é possível executar as demais fases sem ela;
 - c) Permite que as fases posteriores da compilação sejam executadas mais facilmente;
 - d) Facilita a análise léxica;
2. De que maneira o padrão de projeto Visitor facilita a construção do compilador?
 - a) Através da separação entre estrutura de dados e operação que é executada sobre a mesma;
 - b) O Visitor não facilita em nada a construção do compilador;

- c) Através da combinação entre estrutura de dados e da operação que é executada sobre a mesma;
 - d) Na medida em que facilita a construção do analisador de contexto, apesar das dificuldades introduzidas para a construção do gerador de código;
3. Qual motivo justifica a existência de uma fase de análise de contexto em um compilador?
- a) A necessidade de compilar em um único passo;
 - b) A necessidade de eliminar cadeias de entrada que não obedecem às regras de contexto;
 - c) A necessidade de verificar se a gramática livre de contexto está sendo observada;
 - d) A necessidade de minimizar erros de sintaxe livre de contexto na cadeia de entrada;
4. O que acontece durante a subfase "identificação" da análise de contexto?
- a) Apenas a verificação se cada nome usado foi previamente declarado;
 - b) Apenas a verificação se cada nome é declarado uma única vez em cada bloco;
 - c) A verificação se cada nome é declarado uma única vez em cada bloco e também se cada nome usado foi previamente declarado;
 - d) A verificação do tipo das variáveis;
5. Qual é uma mensagem de erro típica emitida durante a subfase de "identificação"?
- a) Variável de tipo incompatível;
 - b) Símbolo desconhecido;
 - c) Símbolo inserido em local incorreto;
 - d) Identificador não declarado;
6. Qual é o papel desempenhado pela Tabela de Símbolos durante a subfase de "identificação"?
- a) Auxiliar na identificação, garantindo que a mesma possa acontecer em apenas dois passos;
 - b) Facilitar a verificação de tipos;

- c) Viabilizar a análise de contexto;
 - d) Auxiliar na identificação, garantindo que a mesma possa acontecer em um único passo;
7. O que acontece durante a subfase "verificação de tipos" da análise de contexto?
- a) Os tipos dos operandos são modificados para satisfazer as operações em que são usados;
 - b) O valor de todas as expressões é determinado;
 - c) Os tipos dos operandos são confrontados com as assinaturas das operações;
 - d) As operações são avaliadas conforme sua precedência e associatividade, porém de forma independente dos operandos;
8. Qual é uma mensagem de erro típica emitida durante a subfase de "verificação de tipos"?
- a) Tipos incompatíveis;
 - b) Variável já declarada;
 - c) Variável não declarada;
 - d) Símbolo desconhecido;
9. Qual é o papel desempenhado pela precedência e pela associatividade dos operadores na avaliação de uma expressão?
- a) Estabelecer uma ordem para a avaliação da expressão;
 - b) Garantir que todos os parênteses sejam avaliados por último;
 - c) Garantir que toda expressão tenha um tipo;
 - d) Garantir que toda expressão produza um valor;
10. O que é uma AST decorada?
- a) É uma árvore de sintaxe depois da análise léxica;
 - b) É a árvore de sintaxe depois da identificação e da verificação de tipos;
 - c) É uma árvore que representa a cadeia de entrada;
 - d) É uma árvore que representa as regras de contexto da linguagem-fonte;

11. Se um tipo de dados possui 900 valores diferentes, em quantos bits (no mínimo) esse tipo deve ser representado?
- a) 9;
 - b) 8;
 - c) 10;
 - d) 16;
12. Se um tipo de dados é implementado em 20 bits, quantos valores distintos ele é capaz de representar?
- a) $1.024 * 1.024$;
 - b) Um milhão;
 - c) 65.536;
 - d) $256 * 256 * 256$;
13. O que justifica certos tipos serem implementados de forma direta e outros de forma indireta?
- a) A forma direta garante flexibilidade, a forma indireta garante eficiência;
 - b) Aspectos sintáticos da linguagem-fonte;
 - c) A necessidade de se compilar em um único passo;
 - d) A forma direta garante eficiência, a forma indireta garante flexibilidade;
14. Por que os valores das expressões indexadoras devem ser verificados antes de se fazer a indexação propriamente dita?
- a) Para garantir que nenhuma área de memória seja acessada ilegalmente;
 - b) Para garantir que as regras de contexto sejam observadas;
 - c) Para garantir que os tipos das expressões indexadoras sejam corretos;
 - d) Para garantir que os mesmos estejam fora da faixa prevista;
15. Qual a fórmula que representa o endereço do elemento $m[i][j]$, supondo a declaração em C `"int m [10][20]"` e também que cada "int" ocupa 2 bytes na memória do computador?
- a) $\text{address } [m[i][j]] = \text{address } [m] + j * 40 + i * 2$

- b) $\text{address } [m[i][j]] = \text{address } [m] + i * 2 + j * 40$
 - c) $\text{address } [m[i][j]] = \text{address } [m] + i * 40 + j * 2$
 - d) $\text{address } [m[i][j]] = \text{address } [m] + j * 2 + * 40$
16. Por que a geração de código para expressões é mais complexa no modelo "máquina de registradores" do que no modelo "máquina de pilha"?
- a) Porque é necessário gerenciar o uso que é feito de cada registrador;
 - b) Porque os processadores típicos possuem poucos registradores;
 - c) Porque as operações que manipulam os registradores são lentas;
 - d) Porque os algoritmos são antigos;
17. Como é feita a atribuição de endereços para as variáveis em um compilador típico?
- a) Por meio de um endereço físico;
 - b) Por meio de um par "deslocamento + registrador";
 - c) Por meio de um deslocamento;
 - d) Por meio de um ponteiro que contém o endereço da variável;
18. Qual o papel, respectivamente, dos ponteiros SB e do LB no sistema de execução?
- a) Apontar para o início da área que armazena as variáveis locais e o início do frame que armazena as variáveis globais;
 - b) Apontar para o início da área que armazena as variáveis não-locais;
 - c) Apontar para o início da área que armazena as variáveis globais e o início do frame que armazena as variáveis locais;
 - d) Os dois tem a mesma função, que é apontar para a área que armazena as variáveis globais e locais;
19. Qual o papel, respectivamente, dos links dinâmico e estático no sistema de execução?
- a) Apontar para o início do frame anterior e apontar para a última ativação do frame que envolve estaticamente o bloco corrente;
 - b) Apontar para a última ativação do frame que envolve estaticamente o bloco corrente e apontar para o início do frame anterior;
 - c) Permitir o acesso às variáveis não-locais;

- d) Permitir o acesso às variáveis globais e locais;
20. Por que o link dinâmico não pode ser usado no lugar do link estático?
- a) Porque todo bloco que é chamado está fora do bloco chamador;
 - b) Porque o link dinâmico permite acessar apenas as variáveis globais;
 - c) Porque o link estático permite acessar apenas as variáveis locais;
 - d) Porque nem todo bloco que é chamado está contido no bloco chamador;