**2ª Avaliação de Compiladores**

Ciência da Computação 7MA 2023.1 – Antenor Parnaíba

Angélica Marina Martiniano da Silva - 01344715

**Questão 2: Descreva quais as vantagens da Forma Backus Naur BNF para compiladores.**

**R:** A Forma Backus Naur é uma notação utilizada para descrever a sintaxe de linguagens de programação e outras gramáticas formais. Essa notação tem diversas vantagens para o desenvolvimento de compiladores. Aqui estão algumas delas:

1. Simplicidade e clareza: A BNF é uma notação simples e fácil de entender. Ela utiliza regras gramaticais básicas, como produções, símbolos terminais e não-terminais, para descrever a estrutura sintática da linguagem. Isso torna a BNF uma ferramenta clara e concisa para especificar a sintaxe de uma linguagem de programação.
2. Legibilidade e manutenção: A BNF permite que os desenvolvedores leiam e compreendam facilmente a estrutura da linguagem através das regras gramaticais definidas. Isso facilita a manutenção do compilador, pois qualquer alteração na sintaxe da linguagem pode ser feita diretamente na notação BNF, tornando a tarefa de atualização da especificação de linguagem mais fácil e menos propensa a erros.
3. Geração automática de analisadores sintáticos: Com a especificação em BNF, é possível gerar automaticamente analisadores sintáticos, como parsers, a partir das regras gramaticais definidas. Essa capacidade permite que os desenvolvedores economizem tempo e esforço, evitando a implementação manual desses analisadores.
4. Validação da sintaxe: A BNF pode ser usada para validar a sintaxe de programas escritos em uma determinada linguagem. Com a especificação em BNF, é possível verificar se um programa está escrito corretamente, de acordo com as regras gramaticais da linguagem. Isso é útil durante a compilação, pois permite identificar erros de sintaxe antes da geração de código objeto.
5. Facilita a documentação: A BNF é amplamente utilizada para documentar a sintaxe de linguagens de programação. Ao fornecer uma descrição precisa e formal da sintaxe, a BNF ajuda os desenvolvedores, usuários e estudantes a entenderem como a linguagem funciona e como escrever programas corretos nessa linguagem. A documentação em BNF também pode ser usada como referência durante o desenvolvimento de compiladores e ferramentas de análise.

**Questão 4: Descreva quais as funções de um analisador semântico em compiladores.**

**R:** Um analisador semântico, também conhecido como fase de análise semântica, é uma etapa importante no processo de compilação de um programa. Ele verifica se o código-fonte está de acordo com as regras semânticas da linguagem de programação e identifica erros ou incoerências que não podem ser detectados apenas pela análise sintática. Aqui estão algumas das principais funções de um analisador semântico em compiladores:

1. Verificação de tipos: O analisador semântico verifica se as operações e expressões são aplicadas a tipos de dados compatíveis. Ele analisa as expressões presentes no código e verifica se os tipos dos operandos e operadores são consistentes. Por exemplo, se uma operação de adição está sendo aplicada a dois operandos do tipo inteiro ou se uma variável está sendo atribuída a um tipo compatível.
2. Declaração de variáveis: O analisador semântico valida as declarações de variáveis, verificando se os identificadores são únicos, se os tipos de dados estão corretamente especificados e se as variáveis estão sendo usadas de acordo com as regras de escopo da linguagem. Ele garante que as variáveis sejam declaradas antes de serem usadas e que não ocorram conflitos de escopo.
3. Verificação de regras específicas da linguagem: Cada linguagem de programação possui regras semânticas específicas que precisam ser verificadas. O analisador semântico se encarrega de validar essas regras, como restrições de acesso a membros de classes, uso correto de palavras-chave reservadas, conformidade com padrões de nomenclatura, entre outros.
4. Resolução de referências: O analisador semântico resolve as referências a identificadores, como variáveis, funções e classes, garantindo que essas referências sejam válidas e correspondam a elementos previamente declarados. Ele verifica se as chamadas de função possuem o número e tipo correto de argumentos e se as variáveis estão sendo usadas dentro de seu escopo adequado.
5. Detecção de erros semânticos: O analisador semântico identifica erros e inconsistências que não são detectados pela análise sintática, como a atribuição de valores incompatíveis, uso de variáveis não declaradas, conversões de tipo inválidas, entre outros. Ele emite mensagens de erro indicando a natureza e a localização desses problemas para auxiliar os programadores na correção.
6. O papel do analisador semântico é fundamental para garantir que um programa esteja semanticamente correto antes de prosseguir para as etapas subsequentes do processo de compilação. Ele desempenha um papel crucial na detecção de erros e na validação da semântica do código-fonte, contribuindo para a produção de um código objeto correto e executável.