

Compiladores

Introdução

José Luis Seixas Junior

Índice

- Introdução.
 - Interpretadores.
 - Compilador.
- Passos de Compilação.
- Analisadores:
 - Léxico;
 - Sintático;
 - Semântico;
- Nosso compilador.

Introdução

- Interpretador:
 - Interpretam unidades básicas do código fonte para executá-los imediatamente;
 - Levam de uma linguagem para uma linguagem simplificada.
 - Código intermediário à ser interpretado por uma máquina acima da linguagem de máquina;
- Linguagens de muito alto nível possuem muitos atributos que não podem ser determinados em tempo de compilação.

Interpretadores

- Vantagens:
 - Operações de entrada e saída podem não ter correspondência compilável;
- Desvantagens:
 - Acarreta a codificação repetida de várias partes sendo ineficiente;
 - Métodos *inline*;

Introdução

- Compiladores:
 - Levam o código-fonte para uma linguagem objeto, geralmente já direcionadas à linguagem de máquina;
 - Percurso direto do fonte completo;
- A geração do código próximo à linguagem de máquina, tende a gerar fonte executável ou mais eficiente;

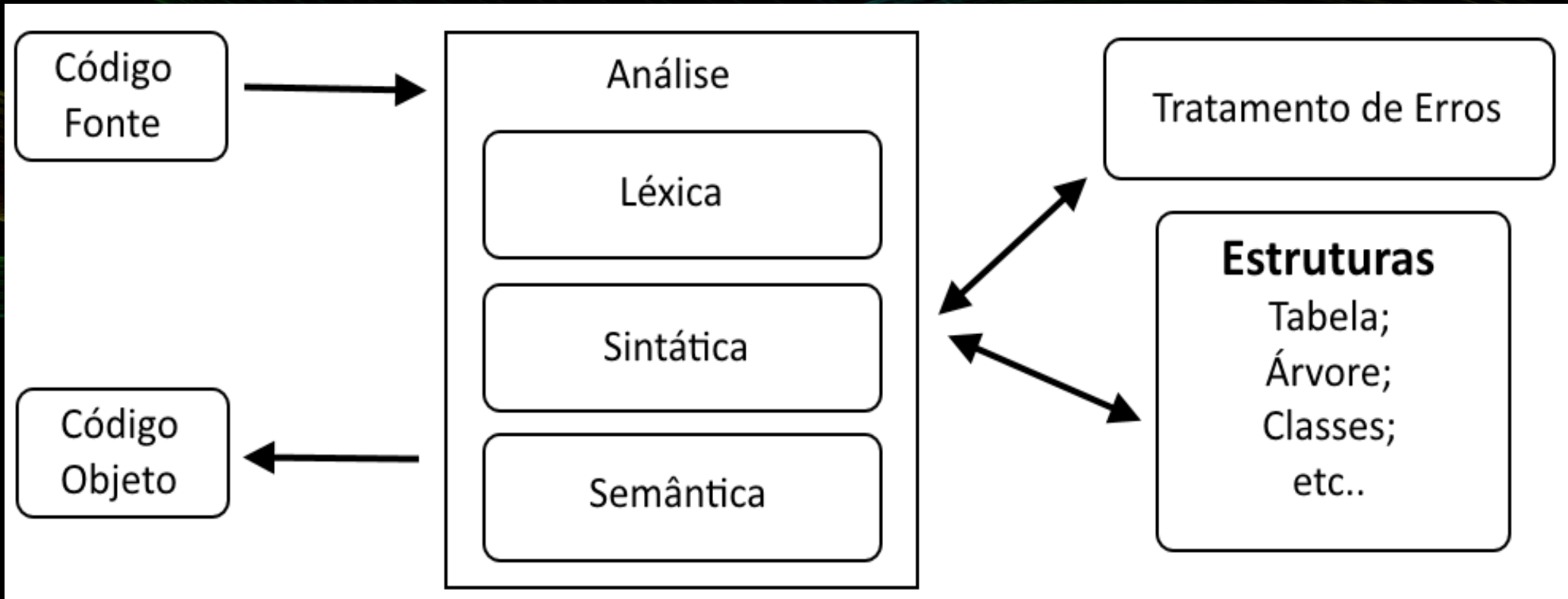
Compiladores

- Único passo:
 - Executa todas as suas atividades com uma única leitura do texto;
- Vários passos:
 - Executa as atividades separadas por funções gerando linguagens intermediárias;

Compiladores (vários passos)

- Vantagens:
 - Menor utilização de memória;
 - Mais fácil otimização;
 - Independência facilita a expressão de erros;
- Desvantagens:
 - Maior volume de entradas e saídas para compilação;
 - Normalmente aumenta o tempo de compilação;
 - Aumenta o tamanho do projeto com gerações intermediárias;

Analísadores



Analisadores

- Léxico:
 - Autômato Finito Determinístico:
 - Expressão regular;
 - Tokenização;
 - Manuseio de palavras complexas;
 - Manuseio de grupos básicos;
 - Manuseio de sintagmas complexos;
 - Fusão de estruturas;

Analísadores

- Sintático:
 - Processo de analisar uma cadeia de palavras para descobrir a sua estrutura frasal [Russel & Norvig 2010].

Analísadores

- Semântico:
 - Analisar a cadeia de palavras por sua estrutura de significado e contexto;
 - Escopo de variáveis, declarações, tipos;

Observações

- As fases do compilador típico não são necessariamente sequenciais;
- Algumas fases podem não existir;
- Algumas estruturas podem ser apenas conceituais, mas não explícitas;
- Geração de código pode ou não ser executável em linguagem de máquina (interpretador);

Nosso Compilador (Léxico)

- and;
- array;
- case;
- const;
- div;
- do;
- else;
- end;
- file;
- for;
- goto;
- in;
- label;
- mod;
- nil;
- not;
- of;

Nosso Compilador (Léxico)

- or;
- repeat;
- set;
- then;
- to;
- type;
- until;
- with;
- var;
- procedure;
- function;
- begin;
- if;
- while;

Nosso Compilador (Léxico)

- Program;
- read;
- write;
- integer;
- boolean;
- double;
- char;
- function;

Nosso Compilador (Léxico)

- Identificadores;
 - Iniciados com não numéricos;
 - Sem caracteres especiais;
- Tipos:
 - Boolean;
 - Integer;
 - Double;
- Comentários:
 - { ... }
 - (* ... *)
- Espaços;
- Quebras de linhas;
- Tokenização de qualquer estrutura escrita;

Nosso Compilador (Sintático)

- Árvore sintática descendente com retrocesso.
 - Sem retrocesso;
 - Ascendente;
- Bloco;
- Comando (palavra reservada);
- Numérico;
- Identificador;
- Comentário;
- Expressão regular;
- Operador.

Nosso Compilador (Semântico)

- Análise de sentido:
 - Depende da expressão de saída do sintático;
 - Conjunto de filhos de mesmo tipo:
 - Casting;
 - Truncamento;
 - Conversão;
- Retorno de função;
- Atribuição;
- Operação;
- Funções em valor ou referência;

Referências

- AHO, A. V.; SETHI, R.; ULLMAN, J. D.; *Compiladores: Princípios, técnicas e ferramentas*, LTC - Livros Técnicos e Científicos Editora, 1995.
- KOWALTOSKI, T.; *Implementação de Linguagens de Programação*, Editora Guanabara Dois, 1983.