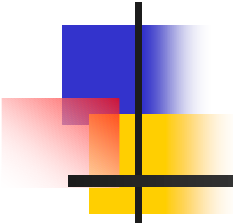


Especificações Gerais do Compilador e Definição do Pascal Simplificado

- 
-
- 1. Especificações Gerais do Compilador
(Decisões de projeto)**
 - 2. Especificações da Linguagem Fonte**
 - Definição Informal**
 - Considerações Léxicas*
 - Considerações Sintáticas*
 - Considerações Semânticas*
 - Definição Formal**
 - 3. Especificações da Linguagem Objeto**



Sintaxe e Semântica

- Um dos objetivos do curso de compiladores é a implementação de uma linguagem de programação.
- Uma LP tem que ter uma definição precisa antes de ser implementada. Esta definição consiste de 2 partes:
 - 1) especificação das seqüências de símbolos que constituem **programas válidos** na linguagem (porque somente estes devem ser traduzidos pelo compilador).

Esta parte é chamada da especificação da **sintaxe**

- 2) especificação do **significado** associado a cada construção da linguagem.

Esta parte é chamada de **semântica** da linguagem e o código objeto gerado pelo compilador está baseado nesta especificação



Métodos Formais

- Existem métodos formais para a especificação completa tanto da sintaxe quanto da semântica de LPs, mas estes últimos são complexos.
- Na prática utilizam-se **métodos formais**, geralmente uma GLC, para **especificar a maior parte dos aspectos sintáticos** da LP.
- **Análise Contextual**, por exemplo:
 - declarações de variável antes de seu uso como usa Pascal,
 - compatibilidade entre declaração formal e uso real de número de parâmetros de procedimentos e funções

bem como **a checagem de tipos estática** são especificadas informalmente.

- O que se perde??? Tendo métodos formais de especificação pode-se automatizar várias partes de um compilador. Por exemplo como fazem os geradores **Lex e YACC**.



Especificações Gerais do Compilador

1) Finalidade a que se destina o compilador

O compilador será desenvolvido com a finalidade didática, para introduzir os conceitos da área de compiladores e ensinar técnicas para cada uma das fases de um compilador.

2) Requisito de velocidade do compilador

Justamente pelo fato de não ter uma finalidade comercial, sendo puramente didático, não há requisito de velocidade. Será um compilador simples, construído com ferramentas geradoras de compiladores, de tempo de desenvolvimento pequeno e transportável.

→ **Porque é transportável? Que características tem nosso projeto que facilita o transporte para outras máquinas?**



Especificações Gerais do Compilador

3) Requisito de velocidade do código-objeto

Não há requisito de velocidade pelo mesmo motivo já descrito. Sendo a finalidade didática é conveniente que a geração de código seja para uma linguagem de montagem real ou linguagem de máquina fictícia, utilizando-se após um montador ou um interpretador para uma máquina hipotética, respectivamente.

→ Qual das possibilidades acima usamos no nosso projeto?

4) Máquina hospedeira do compilador

(Decidam). Compilador cruzado ou auto-residente? Compilador auto-compilável? Como ficam esses conceitos se a máquina objeto é hipotética?

5) Máquina objeto a ser utilizada

Será utilizada a Máquina de Execução para PAscal: MEPA

-> Descrevam as características Gerais da MEPA (pág. 94 do Livro do Kowaltowski)



Especificações Gerais do Compilador

6. Sistemas Operacionais disponíveis em ambas as máquinas

-> (Decidam). Leiam sobre JavaCC para ver onde ele roda (qual a plataforma Java necessária). Leiam sobre o código gerado pelo compiler para também responder esta pergunta (<https://javacc.dev.java.net/doc/features.html>)

7. Ambiente de execução oferecido pela máquina objeto

-> Leiam o Capítulo 9.6 (págs. 157-160 do Livro do Kowaltowski) para saberem como poderíamos implementar um macromontador para as instruções da MEPA para que ao invés de gerar código para uma máquina hipotética gerarmos macros para cada instrução da MEPA que seriam responsáveis por gerar código de máquina de um computador real ou de montagem.

8. Grau de complexidade da linguagem fonte.

Iremos desenvolver um compilador para uma linguagem Pascal Simplificada, que não apresenta um grau de complexidade elevado.

JUSTIFIQUEM, dizendo o que o PS tem que o torna simples (e o que não tem)

Especificações Gerais do Compilador

9. Número de passos do compilador desejado.

Compilador de um passo. Definam.

10. Técnicas de implementação adotadas

→ Qual o *compiler compiler* escolhido? Quais fases ele especifica? E as outras fases, com que linguagem especificaremos?

-> Modelaremos o Analisador Léxico como um AF e com ER estendidas do LEX antes de o especificarmos com JavaCC

O AF ajuda analisar os casos de ambigüidade e devolver o código correto. Por exemplo:

2..3 2.3 2.

> >=

: ::
: :=

-> O Analisador Sintático será Descendente Recursivo pela clareza e simplicidade de o construir a partir da gramática em EBNF.

→ As ações semânticas e as de geração de código serão sub-rotinas inseridas no arquivo de especificação do JavaCC.

→ Não faremos otimizações.

Especificações da Linguagem

Fonte - Definição Informal

Considerações Léxicas

- **Comentários**

Permite Comentário dentro de Comentário?

→ Grande questão!!!! Qual o tipo de gramática que dá conta de checar duplo balanceamento: $a^n b^n$? Isto tem implicações na hora de especificar os comentários, pois uma ER não suportaria.

São sensíveis à linha ou não?

Qual o marcador de comentário? (* e *) { e } ou os dois?

- **Identificadores e palavras reservadas**

Os identificadores e as palavras-reservadas serão case-sensitive?

Ou seja, haverá diferenciação entre letras minúsculas e maiúsculas?

- **Tamanho máximo de identificadores**

Os identificadores terão tamanho máximo de caracteres?

Se não, há número de caracteres para diferenciação?



Considerações Léxicas

- **Conjunto de palavras reservadas**
Insiram o conjunto
- **Constantes permitidas**
inteiro: indiquem tamanho máximo
Para os grupos que terão que estender os tipos:
real: indiquem tamanho máximo e notação (ponto fixo ou flutuante, ou os dois)
cadeias de caracteres e caracteres: tamanho, e formação (", ')
- **Caracteres não imprimíveis e símbolos especiais**
Não imprimíveis serão permitidos?
Quais são os símbolos especiais?



Considerações Sintáticas

Tipos

primitivos: ?

definidos pelo usuário: ?

- **Operações sobre tipos elementares**

Quais são?

Tipos resultantes?

Tabela de Prioridade de operadores?

- tipos abstratos de dados?

Não serão permitidos.

- **E/S é tratada como comando ou procedimento pré-definido?**

A entrada e saída será tratada como procedimento pré-definido

(Discutam os prós e contras da escolha ser como procedimento pré-definido versus comando)



Considerações Sintáticas

- **Que tipo pode lido?**
(Digam)
 - **Funções pré-definidas**
(Existem ou não?)
 - **Estruturas de controle ao nível de cláusulas**
(Quais?)
 - **Estruturas de controle ao nível de unidades de programa**
(Quais?)
 - **Exemplos de programas gerados pelo PS com as extensões.**
- > **Construam vários programas, desde os sem procedimentos até com procedimentos sem passagem de parâmetros, com passagem por valor e com passagem por referência.**



Considerações Semânticas

- **Visibilidade e escopo para identificadores**

As variáveis terão tratamento de variáveis locais dentro de procedimentos, ou seja, terão seu escopo limitado ao próprio procedimento em que foram criadas.

Isto implica no caso de termos uma variável global com o mesmo nome da variável local, não haverá meios de acessarmos a variável global

- **Procedimentos aninhados**

Serão permitidos procedimentos aninhados

-> Onde na gramática está especificada esta propriedade?



Considerações Semânticas

- **Passagem de parâmetros de que tipo? Variáveis globais?**

As passagens de parâmetros poderão ser feitas por referencia ou por valor.

-> Como a linguagem as diferencia?? Digam.

Quanto às variáveis globais, elas serão permitidas, mas lembrando da restrição de escopo descrita acima, no caso de termos variáveis locais com o mesmo nome de variáveis globais.

- **Compilação independente, compilação em separado ou o programa está em um único arquivo?**

A compilação será efetuada em um único arquivo.

- **Especificação semântica e exemplos das extensões do grupo.**

-> Tragam as checagens que devem ser feitas num FOR, REPEAT (e outras extensões) para funcionarem corretamente



Especificações da Linguagem

Fonte - Definição Formal

- A gramática da linguagem PS é definida como uma quadrupla: $G = (V_n, V_t, P, S)$
- **V_n** : ??
- **V_t** : ??
- **P** : dado em classe, adicionado das produções do grupo: ??
- **S** : ??



Especificações da Linguagem Objeto

- Será definida uma máquina, MEPA (Máquina de Execução para PAscal).
- Não será exigido um interpretador para o conjunto de instruções que possui o mesmo nome MEPA.
- Façam uma cópia das Instruções da MEPA (págs 180 – 182) do Livro do Kowaltowski)