Universidade de São Paulo

Instituto de Ciências Matemáticas e de Computação

SCC0206 - Introdução à Compilação
Professora - Sandra Maria Aluísio

FRANKIE

Implementação de uma linguagem de programação híbrida entre Pascal e C

Parte III

Implementação do Analisador Semântico

*Projeto de Curso Desenvolvido pelos Alunos (**GRUPO 7**)*

Ubiratan F. Soares (5634292)
Humberto Yagi (5634420)
Ulisses F. Soares (5377365)

São Carlos, 06 de junho de 2011

Sobre a correção da Gramática da FRANKIE

Conforme foi depreendido da submissão do último trabalho, a sintaxe da FRANKIE possuía uma falha de implementação relacionada à extensão do núcleo da linguagem proposta para o grupo : no caso, a implementação da atribuição para uma variável do tipo Union não estava correto, pois essa não poderia aceitar uma lista de campos do lado esquerdo da atribuição.

<COMMAND-TYPE> := ["**.**" **<**IDENTIFIER>] "**=**" <EXPRESSION> | ["**(**" <EXPRESSION-LIST> "**)**"**]**

Antes da correção, tínhamos a seguinte regra na gramática da FRANKIE
a qual levava a um erro de compilação da seguinte construção de linguagem, por exemplo :

x.y.z.w := 10;

<COMMAND-TYPE> := ("**.**" **<**IDENTIFIER>)+ "**=**" <EXPRESSION> | ["**(**" <EXPRESSION-LIST> "**)**"**]**

permitindo, por exemplo, a construção de linguagem acima mencionada, ou seja, uma variável a receber uma atribuição pode ser um campo de uma estrutura Union aninhada.

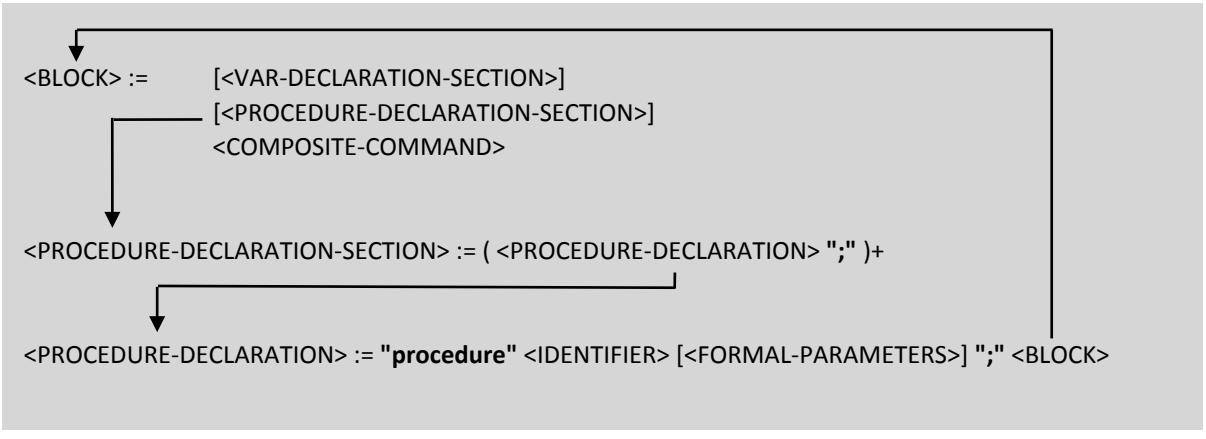
Sobre as Especificações Semânticas

Sobre a Visibilidade e Escopo para Identificadores

Segundo a proposta minimalista da FRANKIE, as variáveis têm escopo limitado ao procedimento aonde foram criadas. No caso de uma variável local possuir o mesmo nome de uma variável global, essa ficará inacessível ao procedimento.

Sobre Procedimentos Aninhados

FRANKIE oferece suporte à procedimentos aninhados. Esta propriedade está colocada pelas seguintes regras na gramática :



Sobre a Passagem de Parâmetros e Variáveis Globais

A passagem de parâmetros pode ser realizada por valor ou por referência. A distinção entre os tipos de passagem é realizada pelas seguintes regras da gramática

<PROCEDURE-DECLARATION> := **"procedure"** <IDENTIFIER> [<FORMAL-PARAMETERS>] **";"** <BLOCK>

<FORMAL-PARAMETERS> := **"("** <FORMAL-PARAMETERS-SECTION> (**";"** <FORMAL-PARAMETERS-SECTION>)* **")"**

<FORMAL-PARAMETERS-SECTION> := [**"var"**] <DECLARATION-LIST> **":"** (<TYPE-SPECIFIER> | **"union"**)

ou seja, a distinção entre uma lista de parâmetros passadas por valor ou por referência se dá através da palavra reservada **var** precedendo as declarações.

Adicionalmente, como já discutido nos aspectos de escopo de identificadores, variáveis globais são acessíveis pelos procedimentos, desde que não existam variáveis locais de mesmo nome.

Sobre o Método de Compilação

O compilador FRANKIE proposto não oferece suporte à compilação separada nem à compilação independente, ou seja, todo o programa-fonte deve estar contido em um único arquivo. A compilação não é interrompida se erros semânticos são detectados.

Sobre a Decisão de Implementação da Tabela de Símbolos

Para a implementação da tabela de símbolos utilizada na análise semântica, foi utilizada a estrutura de dados lista ligada encadeada, suportada pela coleção **LinkedList** da API da linguagem Java. Essa decisão foi tomada de acordo com o critério custo - benefício, com base nas propriedades dessa estrutura de dados em detrimento das operações mais frequentemente realizadas na tabela de símbolos, conforme discussão em sala de aula.

Foi criada uma classe **Campos.java** para representar cada entrada da tabela. Para remover entradas, por exemplo quando da checagem de tipos de parâmetros em procedimentos, uma variável dessa classe mantém uma segunda lista ligada de inteiros, que correspondem às linhas nas quais os parâmetros dos procedimentos em análise se encontram. Nesse caso, essas linhas tem seu *id* marcado como **null**, representando a remoção.

Sobre a Implementação da Checagem de Tipos

Para a checagem de tipos, foi utilizada uma amarração estática. Foram implementadas as verificações :

- declaração de identificadores repetidos,
- compatibilidade de tipos em comandos,
- concordância entre parâmetros formais (quantidade e tipos)

Nossa verificação é comandada pela análise sintática, procedimento que permite a checagem de tipo em um processo de compilação de uma única passagem.