

Esquema de Tradução – De Expressão Aritmética para Código da Máquina de Pilha

por Antonio Atta
atta@uneb.br

Segundo o “livro do dragão”, Esquemas de Tradução constituem um tipo de notação complementar às Definições Dirigidas pela Sintaxe (ver texto livre “Definições Dirigidas pela Sintaxe – DDS”) que podem ser definidos e usados em Traduções Dirigidas pela Sintaxe. Consiste em “ações semânticas” que são associadas às regras de produção da gramática da linguagem. Essas ações semânticas podem ser usadas para denotar operações procedimentais (comandos de linguagem de programação) a serem incorporadas ao código do tradutor e executadas no momento da construção da árvore sintática, com objetivos diversos (proceder operações de análise semântica, executar comandos para consulta e gestão da tabela de símbolos, emitir comandos para a geração de código intermediário etc.). Neste texto vamos usar a gramática das Expressões Aritméticas trabalhada nos textos “Compreendendo a ASDR” e “Definições Dirigidas pela Sintaxe – DDS” como exemplo, para incorporar sobre ela um esquema de tradução que visa gerar códigos de uma Máquina de Pilha – MP (que executa os comandos traduzidos usando uma pilha de dados como estrutura de dados auxiliar) relativos às expressões aritméticas de entrada; ao final, apresentaremos o ASDR associado com a incorporação do esquema de tradução discutido.

Esquema de Tradução – De Expressão Aritmética para o Código da MP

Planejar um esquema de tradução para ser usado na geração de código em um compilador que aplica a análise sintática descendente recursiva – ASDR está diretamente relacionado com buscar os pontos do processo de montagem da árvore sintática em que os comandos de emissão do código (comandos da MP nesse caso) devem ser feitos para gerar um código intermediário que guarde similaridade semântica (de lógica) com o programa fonte sendo compilado. Se assemelha com o planejamento de uma Definição Dirigida pela Sintaxe – DDS, mas em vez de indicar como o cálculo de atributos de não-terminais da gramática deve ser feito, o esquema de tradução cria novos elementos, ou ações semânticas, do lado direito das regras de produção da gramática que devem ser “executadas” pelo compilador ao se “visitar” tais elementos na construção da árvore sintática; o código fonte indicado por esses elementos deve ser enxertado no código fonte do compilador, no ponto em que ele ocorre na ação semântica associada à regra de produção selecionada para a montagem da árvore sintática naquele ponto da execução da tradução. Uma única regra de produção pode incorporar diversas ações semânticas entremeadas nos símbolos terminais e não-terminais que a compõe.

Na maioria das vezes, um esquema de tradução correto pode ser deduzido a partir da análise e compreensão da lógica de montagem do trecho da árvore sintática que corresponde a cada construção válida (comandos) da linguagem fonte para a qual o compilador está sendo desenvolvido.

Vamos usar a gramáticas de Expressões Aritméticas já conhecida para entender como esse processo de concepção de esquema de tradução ocorre.

A Figura 1 a seguir mostra a gramática adotada no texto anterior a partir da qual o ASDR apresentado (no Texto I) foi construído.

Gramática simples para Expressões Aritméticas sem recursão à esquerda (regras de produção)

```

Expr → Termo Resto
Termo → Fator Sobra
Resto → '+' Termo Resto | '-' Termo Resto | ε
Sobra → '*' Fator Sobra | '/' Fator Sobra | ε
Fator → '(' Expr ')' | num

```

Figura 1 – Gramática para Expressões Aritméticas em modo infix adequada à implementação de ASDRs

Vejamos agora alguns exemplos de expressões aritméticas e o correspondente código da MP que se espera que o compilador de expressões gere, de forma a respeitar o funcionamento da MP e gerar código coerente com o cálculo esperado da expressão aritmética original correspondente.

EXPRESSÃO ARITMÉTICA	CÓDIGO DA MÁQUINA DE PILHA CORRESPONDENTE
2+3=	PUSH 2 PUSH 3 ADD
2+3*5=	PUSH 2 PUSH 3 PUSH 5 MUL ADD
(2+6) / (3+1)=	PUSH 2 PUSH 6 ADD PUSH 3 PUSH 1 ADD DIV
2+3+4*3 / (2+1)=	PUSH 2 PUSH 3 ADD PUSH 4 PUSH 3 MUL PUSH 2 PUSH 1 ADD

	DIV
	ADD

As sequências de tradução da tabela acima são suficientes para ilustrar o que se espera de um esquema de tradução que gere código da MP e que, por sua vez, ao ser executado deixará no topo da pilha de dados o resultado calculado da expressão. Ao gerar as árvores sintáticas de cada uma das expressões do exemplo acima (deixado como exercício) é possível projetar esquema de tradução a seguir, a ser usado na geração do código da MP (apresentado na coluna da direita na tabela anterior). A chamada da função “emit” (grifada em vermelho na coluna das ações semânticas) foi usada nesse esquema de tradução para indicar que, no ponto em que ela ocorre, o seu argumento (ou comando correspondente da MP) será emitido na saída (impressa ou em arquivo), correspondendo, no conjunto das várias execuções de “emit”, ao código da MP traduzido a partir da expressão aritmética fonte.

GRAMÁTICA	AÇÕES SEMÂNTICAS
EXPR → TERMO RESTO	
TERMO → FATOR SOBRA	
RESTO → + TERMO RESTO	RESTO → + TERMO { emit(ADD); } RESTO
RESTO → - TERMO RESTO	RESTO → - TERMO { emit(SUB); } RESTO
RESTO → ε	
FATOR → (EXPR)	
FATOR → num	FATOR → num { emit (PUSH num); }
SOBRA → * FATOR SOBRA	SOBRA → * FATOR { emit(MUL); } SOBRA
SOBRA → / FATOR SOBRA	SOBRA → / FATOR { emit(DIV); } SOBRA
SOBRA → ε	

As ações semânticas (chamadas da função “emit”, em vermelho) correspondem a nós “filhos” que passam a integrar a construção da árvore sintática de cada expressão e que, ao ocorrerem nesse processo de construção, devem ser executados (daí o nome de ação) pelo tradutor. Ou seja, em termos práticos, devem ser inseridos no código já desenvolvido do ASDR, no ponto em que ocorrem na descrição da ação semântica. Se tomarmos como exemplo a regra de produção “**RESTO → + TERMO RESTO**” da gramática, o esquema de tradução apresentado indica que a ação de emissão do comando da MP “**ADD**” deve ser feita no ASDR **exatamente** após o reconhecimento do token “+” e a chamada da função do não-terminal “**TERMO**” e antes da chamada da função correspondente ao não-terminal “**RESTO**”. Caso a emissão desse comando da MP seja feita em outro momento (ou ponto do compilador), o código gerado ainda será o da MP, mas estará semanticamente incorreto em relação à expressão aritmética que o gerou.

Experimente agora montar em uma folha de papel as árvores das três expressões aritméticas apresentadas acima e em cada uma delas desenhe as ações semânticas como se fossem nós filhos regulares das árvores (a exemplo dos símbolos não-terminais e terminais da gramática) – nesse caso, as ações semânticas sempre serão nós do tipo “folha” da árvore, já que não possuem derivação. Use arestas tracejadas na ligação entre um nó pai e um filho do tipo ação semântica para demarcar que este é um filho “especial”. Feito isso, percorra cada árvore como se estivesse construindo-a pela primeira vez (ou seja, no mesmo caminhamento que é feito pelo ASDR na sua execução). À medida que for passando por um nó do tipo ação semântica do esquema de tradução escreva ao lado da árvore, em sequência, os comandos que vão sendo emitidos pela execução de cada ação semântica. No final, você verá que gerou, ao longo do processo, o mesmo código correto da MP apresentado nos exemplos de expressões algébricas acima.

Juntando o Esquema de Tradução e o ASDR

O ASDR do texto “Compreendendo a ASDR” (o mesmo que usamos para incorporar a DDS definida no texto “Definições Dirigidas pela Sintaxe – DDS) volta a ser usado agora para a inserção do esquema de tradução discutido neste texto. A Figura 2 a seguir apresenta esse código com o esquema de tradução incorporado. Notem como fica muito fácil enxertar as ações semânticas, uma vez que o esquema de tradução esteja bem definido e tenha sido previamente testado como funcional e correto. Nessa figura, os comandos do tradutor correspondentes às ações semânticas aparecem grifados em vermelho para facilitar a sua identificação. É apresentado apenas o código das funções do ASDR pois são nelas que o esquema de tradução é incorporado. Analise o código e veja como ele se relaciona diretamente com o esquema de tradução discutido acima.

Considerações Finais

Esquemas de Tradução são ferramentas poderosas na construção correta de ações semânticas que devem ser desencadeadas no processo de compilação, particularmente na geração de código traduzido. Recomenda-se que os exemplos de codificação C presentes neste texto (Figura 2), associados ao ASDR de expressões aritméticas, com a incorporação do esquema de tradução discutido, sejam estudados até o completo entendimento do processo de desenvolvimento e inserção das ações semânticas no analisador sintático que geram código da MP para expressões aritméticas. Eles constituem uma peça importante na construção de compiladores pela técnica da Tradução Dirigida pela Sintaxe. Esse exercício é fundamental para desenvolver a habilidade de aplicação de esquemas de tradução diversos à ASDRs para a concepção de Geradores de Código, conforme visto. Para facilitar esse estudo e permitir a manipulação dos códigos C do ASDR com o Esquema de Tradução apresentado, disponibilizamos a codificação C completa dos procedimentos e o respectivo projeto (IDE Code::Blocks) em:

<https://github.com/antonioatta/ASDR-Expressoes-Aritmeticas-MP>

```

#include <stdlib.h>
#include "analex.h"
#include "anasint.h"

void Expr() { // EXPR
    Termo();
    Resto();
}

void Termo() {
    Fator();
    Sobra();
}

void Resto() { // RESTO
    if ((t.cat==SN && t.cod==SOMA) || (t.cat==SN && t.cod==SUBT)) {
        int sinal = t.cod;
        Analex(stdin);
        Termo();
        printf("%s\n", ((sinal==SOMA) ? "ADD" : "SUB"));
        Resto();
    }
    else ; // saida por vazio
}

void Sobra() { // SOBRA

    if ((t.cat==SN && t.cod==MULT) || (t.cat==SN && t.cod==DIVI)) {
        int sinal = t.cod;
        Analex(stdin);
        Fator();
        printf("%s\n", ((sinal==MULT) ? "MUL" : "DIV"));
        Sobra();
    }
    else ; // saida por vazio
}

void Fator() { // FATOR

    if (t.cat==SN && t.cod==ABRE_P) {
        Analex(stdin);
        Expr();
        if (t.cat!=SN || t.cod!=FECHA_P) {
            Erro(3);
        }
        Analex(stdin);
    }
    else if (t.cat==OP) {
        printf("PUSH %d\n", (int)t.valor);
        Analex(stdin);
    }
    else Erro(4);
}

```

Figura 2 - Implementação dos procedimentos do Analisador Sintático (ASDR) com Esquema de Tradução para gerar código da Máquina de Pilha para as Expressões Aritméticas