

Definições Dirigidas pela Sintaxe - DDS

por Antonio Atta
atta@uneb.br

A Análise Sintática Descendente Recursiva (ASDR) é uma das formas mais usuais de implementar-se analisadores sintáticos de linguagens de programação pela codificação manual dos mesmos. Nessa proposta, o modelo de compilação adotado é a compilação em um passo, com o processo de tradução sendo dirigido pela execução da análise sintática – as etapas seguintes do projeto do compilador, análise semântica, geração de código intermediário, etc. são executadas de forma concomitante com a análise sintática. Isto não quer dizer que o desenvolvimento do compilador não ocorra de forma modular. É possível escrever e testar todo o código do analisador sintático e somente depois ir inserindo no mesmo o código referente às outras etapas do processo de compilação. No final, o código do analisador sintático finda por conter também, embutido nele, o código de todas as etapas restantes do compilador, sendo possível, no entanto, identificar claramente todas as etapas do processo de compilação, daí a noção de Tradução Dirigida pela Sintaxe.

DDS para o ASDR de Expressões Aritméticas

No texto livre “Compreendendo a Análise Sintática Descendente Recursiva – ASDR” apresentamos, usando um enfoque prático, como implementar um ASDR para Expressões Aritméticas. Neste texto vamos ilustrar como implementar uma Tradução Dirigida pela Sintaxe a partir daquele ASDR de Expressões Aritméticas. No código apresentado no texto anterior, a análise findava apenas indicando se a expressão aritmética digitada pelo usuário na entrada padrão (teclado) estava correta ou não, do ponto de vista das análises léxica e sintática. Neste texto desenvolvemos uma Definição Dirigida pela Sintaxe – DDS que incorpora àquele código uma tradução dirigida pela sintaxe que calcula e emite o resultado da expressão aritmética de entrada. Ou seja, traduz a expressão do seu formato de entrada (linguagem de expressões aritméticas) para o resultado correspondente (código alvo).

Toda DDS é constituída pelo conjunto de regras de produção da gramática em uso pelo Analisador Sintático, adicionadas, em uma relação de um para um, pelo conjunto de *regras semânticas* que indicam como atributos devem ser associados à cada nó não-terminal da árvore de derivação de uma determinada cadeia válida da linguagem. A execução das regras semânticas resulta na *árvore anotada* associada a essa cadeia. A árvore anotada nada mais é do que a árvore sintática com uma série de atributos (ou variáveis) associados (“pendurados”) a cada símbolo não-terminal, cujos conteúdos são calculados a partir das respectivas regras semânticas da DDS em uso. No final, a composição dos valores dos atributos, que vão sendo calculados à medida que a árvore sintática vai sendo construída (justificando a denominação de “tradução dirigida pela sintaxe”), indicará o código traduzido para a linguagem alvo associado a cada cadeia válida da linguagem fonte.

Como visto, tudo começa com a gramática a partir da qual o ASDR foi construído. Recordemos então a gramática final para Expressões Aritméticas adotada no texto anterior.

A Gramática de Expressões Aritméticas

A Figura 1 abaixo mostra a gramática adotada no texto anterior a partir da qual o ASDR apresentado 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

O processo de desenvolvimento da DDS consiste em associar um ou mais atributos para cada um dos símbolos não-terminais em cada uma de suas regras de produção na gramática e indicar, por meio de regras semânticas, como esses atributos devem ser calculados. Como o processo de tradução acompanha a construção (e destruição) da árvore sintática, o cálculo dos valores dos atributos na DDS também é um processo dinâmico, com os atributos em um determinado nó da árvore sintática (símbolo não-terminal) fazendo uso dos valores dos atributos de nós de sua sub-árvores ou passando valores para que esses atributos da sub-árvore possam ser corretamente calculados.

Tipos de Atributos em uma DDS

Existem dois tipos de atributos possíveis em uma DDS:

1. **Atributos Sintetizados** – são aqueles onde a regra semântica associada (que indica o seu cálculo) usa valores de atributos de seus nodos filhos (que já foram calculados quando da construção da sub-árvore); portanto, um atributo sintetizado em um nó X da árvore anotada é calculado a partir dos valores dos atributos dos nós filhos de X . Se $S \rightarrow ABC$ é uma regra de produção, então um atributo t **sintetizado** de S , denotado $S.t$, usa valores de quaisquer atributos dos não-terminais A , B ou C ;
2. **Atributos Herdados** – são aqueles que são usados nas regras semânticas de elementos da sub-árvore mas que são passados para os nodos filhos a partir dos nodos pais ou são oriundos de nós *irmãos*; portanto, um atributo herdado em um nó Y da árvore anotada é calculado a partir dos valores dos atributos calculados em nós pai ou irmãos de Y . Se $S \rightarrow$

ABC é uma regra de produção, então um atributo u herdado de A (ou B , ou C), denotado por $A.u$, usa valores de quaisquer atributos dos não-terminais S , B ou C .

Para compreender melhor o conceito e utilização dos dois tipos de atributos de DDS relacionados acima, sugerimos fortemente a leitura da seção destinada a DDS no capítulo de Análise Sintática do livro do dragão (AHO, 2008).

Especificação de uma DDS

DDSs são especificadas de forma tabular com o lado esquerdo contendo as regras de produção e o lado direito as regras semânticas sob a forma de equações que indicam como calcular os valores dos atributos usando operações primitivas (soma, subtração, multiplicação, divisão, concatenação de cadeias, operações lógicas, etc), ou seja, nenhuma operação que exija sequencia e controle de fluxo de comandos (rotina).

Exemplo: DDS que valida números decimais (pelo uso da gramática) e traduz uma cadeia contendo um número decimal para o correspondente à soma de seus dígitos. Em cada linha da tabela x é um atributo (nesse caso sintetizado) que é calculado pelo uso da regra semântica correspondente. Ao final do cálculo de x em cada nó da árvore sintática teremos no atributo x do nó raiz da árvore o valor da soma dos dígitos do número.

GRAMÁTICA	REGRAS SEMÂNTICAS
$NUM \rightarrow DIGITO\ NUM_1$	$NUM.x = NUM_1.x + DIGITO.x$
$NUM \rightarrow DIGITO$	$NUM.x = DIGITO.x$
$DIGITO \rightarrow 0$	$DIGITO.x = 0$
$DIGITO \rightarrow 1$	$DIGITO.x = 1$
\dots	\dots
$DIGITO \rightarrow 9$	$DIGITO.x = 9$

Na DDS acima o índice subscrito em NUM_1 serve apenas para diferenciar o NUM do lado direito do NUM do lado esquerdo na notação da regra semântica; na gramática eles são essencialmente o mesmo não terminal NUM . Fica como exercício validar a DDS acima e comprovar que para uma cadeia que corresponde a um número decimal válido a DDS produz como resultado da tradução a soma de dígitos desse número (armazenado no atributo x do nó raiz da árvore anotada).

Desenvolvendo uma DDS para um ASDR de Expressões Aritméticas

Retomemos agora a nossa gramática de Expressões Aritméticas cujo ASDR foi desenvolvido no texto anterior. Qual seria uma possível DDS para traduzir uma expressão aritmética validada pela gramática para o código alvo resultado calculado da expressão?

A DDS a seguir é uma possível solução para uma DDS desenvolvida para a gramática apresentada no início deste texto.

GRAMÁTICA	REGRAS SEMÂNTICAS
EXPR \rightarrow TERMO RESTO	EXPR.x = TERMO.y EXPR.y = RESTO.y
TERMO \rightarrow FATOR SOBRA	TERMO.x = FATOR.y TERMO.y = SOBRA.y
RESTO \rightarrow + TERMO RESTO ₁	RESTO.x = PAI \uparrow .x + TERMO.y (onde PAI = EXPR ou RESTO) RESTO.y = RESTO ₁ .y
RESTO \rightarrow - TERMO RESTO ₁	RESTO.x = PAI \uparrow .x - TERMO.y (onde PAI = EXPR ou RESTO) RESTO.y = RESTO ₁ .y
RESTO \rightarrow ϵ	RESTO.y = PAI \uparrow .x (onde PAI = EXPR ou RESTO)
FATOR \rightarrow (EXPR)	FATOR.y = EXPR.y
FATOR \rightarrow num	FATOR.y = valor(num)
SOBRA \rightarrow * FATOR SOBRA ₁	SOBRA.x = PAI \uparrow .x \times FATOR.y (onde PAI = TERMO ou SOBRA) SOBRA.y = SOBRA ₁ .y
SOBRA \rightarrow / FATOR SOBRA ₁	SOBRA.x = PAI \uparrow .x \div FATOR.y (onde PAI = TERMO ou SOBRA) SOBRA.y = SOBRA ₁ .y
SOBRA \rightarrow ϵ	SOBRA.y = PAI \uparrow .x (onde PAI = TERMO ou SOBRA)

Nessa DDS são aplicados tanto atributos sintetizados quanto atributos herdados. O atributo “x” sempre armazena valor a ser passado para a operação na sub-árvore abaixo do não-terminal (atributo herdado) e o atributo “y” sempre armazena o resultado parcial de uma operação que será passado ao não-terminal do nó pai superior (atributo sintetizado). A notação com a seta para cima (\uparrow) na DDS indica o uso de um atributo herdado.

Exercite o entendimento da DDS acima escrevendo expressões aritméticas válidas, construindo sua árvore sintática correspondente e aplicando as regras semânticas da DDS para obter a árvore anotada e, a partir dela, o resultado da expressão no atributo sintetizado “y” de EXPR na raiz da árvore.

Implementando a DDS no ASDR desenvolvido anteriormente.

Observe agora a partir do extrato de código abaixo nas Figuras 2A e 2B (correspondente apenas ao Analisador Sintático apresentado no texto anterior – onde foram incluídos os cálculos referentes às Regras Semânticas da DDS), como a DDS definida acima pode ser incorporada ao ASDR já desenvolvido que, até então, só fazia a análise sintática da expressão aritmética.

Nesses extratos, a variável “`primOp`” corresponde ao atributo “`x`” na DDS, que será herdado pelos símbolos não-terminais filhos usando o mecanismo de passagem de parâmetros em funções na linguagem C; já a variável “`result`” faz o papel do atributo “`y`” da DDS, que retorna das chamadas de função e é sintetizado no cálculo dos atributos do não-terminal pai associado.

Considerações Finais

Recomenda-se que os exemplos de codificação C associados ao ASDR de expressões aritméticas, com a incorporação da DDS apresentada, presentes neste texto sejam estudados até o completo entendimento do processo de desenvolvimento e inserção de DDS a analisadores sintáticos pela técnica da Tradução Dirigida pela Sintaxe. Esse exercício é fundamental para desenvolver a habilidade de aplicação de DDS a ASDR para a construção de Analisadores Semânticos e Geradores de Código, conforme visto. Para facilitar esse estudo e permitir a manipulação dos códigos C do ASDR com a DDS apresentados, disponibilizamos a codificação C completa dos procedimentos e o respectivo projeto (IDE Code::Blocks) em:

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

Referências

Aho, Alfred V. [et al.]; **Compiladores: Princípios , Técnicas e Ferramentas.** 2ª Edição. São Paulo: Pearson Addison-Wesley. 2008. ISBN 978-8588639249

```

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

extern TOKEN t;

float Expr();
float Termo();
float Resto(float);
float Sobra(float);
float Fator();
void Erro(int);

float Expr() {
    float primOp, result;

    primOp=Termo();
    result=Resto(primOp);

    return result;
}

float Termo() {
    float primOp, result;

    primOp=Fator();
    result=Sobra(primOp);

    return result;
}

float Resto(float primOp) {
    float segunOp, tmpResult, result;

    if ((t.cat==SN && t.cod==SOMA) || (t.cat==SN && t.cod==SUBT)) {
        TCOD opSalva=t.cod;
        Analex(stdin);
        segunOp=Termo();
        tmpResult=(opSalva==SOMA ? primOp+segunOp : primOp-segunOp);
        result=Resto(tmpResult);
        return result;
    }
    else {          // saida por vazio
        return primOp;
    }
}

```

Figura 2A - Implementação dos procedimentos do Analisador Sintático (ASDR)

```

float Sobre(float primOp) {

    float segunOp, tmpResult, result;

    if ((t.cat==SN && t.cod==MULT) || (t.cat==SN && t.cod==DIVI)) {
        TCOD opSalva=t.cod;
        Analex(stdin);
        segunOp=Fator();
        if (opSalva==DIVI && segunOp==0.0) Erro(2);
        tmpResult=(opSalva==MULT ? primOp*segunOp : primOp/segunOp);
        result=Sobre(tmpResult);
        return result;
    }
    else {          // saida por vazio
        return primOp;
    }
}

float Fator() {

    float result;

    if (t.cat==SN && t.cod==ABRE_P) {
        Analex(stdin);
        result=Expr();
        if (t.cat!=SN || t.cod!=FECHA_P) {
            Erro(3);
        }
    }
    else if (t.cat==SN && (t.cod==SOMA || t.cod==SUBT)) {
        int sinal = (t.cod==SOMA ? 1 : -1);
        Analex(stdin);
        if (t.cat==OP) result=sinal*t.valor;
        else Erro(4);
    }
    else if (t.cat==OP) {
        result=t.valor;
    }
    else Erro(1);
    Analex(stdin);
    return result;
}

```

Figura 2B - Implementação dos procedimentos do Analisador Sintático (ASDR)