

UMA FERRAMENTA PARA GERAÇÃO DE AVALIADORES DE ATRIBUTOS

Edmar Wienskoski Junior ¹José Lucas Mourão Rangel Netto ²

I - Introdução

Bochmann apresentou uma forma de avaliação de atributos, em que são usados atributos herdados, sintetizados e a seguinte regra de avaliação: em cada regra sintática, a avaliação inicia pelos atributos herdados dos filhos, da esquerda para a direita, e então são avaliados os atributos sintetizados do pai. Este método possibilita manter a gramática sem alterações (como criar reduções artificiais de modo a só utilizar atributos sintetizados). Em BOCHMANN (1976), o autor admite a possibilidade de que a avaliação dos atributos seja feita em várias passagens da esquerda para a direita, mas essa possibilidade não é considerada aqui. Para o caso de avaliação em uma única passagem, também não se faz necessária a construção da árvore de derivação, pois os atributos herdados e sintetizados podem ter seus valores armazenados em uma pilha. Esta classe de gramática de atributos é às vezes chamada de L-AG (DERANSART, 1988).

A idéia central aqui apresentada visa combinar de forma simples para o usuário a facilidade de construção de um analisador ascendente com a facilidade de definição de regras semânticas através do uso de L-AGs. Isto é feito distinguindo a sintaxe concreta da sintaxe abstrata para a linguagem de programação considerada. Assim, para a construção de um analisador sintático usa-se uma gramática (que descreve a sintaxe concreta), enquanto a gramática da sintaxe abstrata é usada para a definição das regras semânticas da L-AG. Apoiada nessa idéia, a Linguagem de Descrição de Atributos LINDA, oferece mecanismos para:

- a definição da gramática da sintaxe concreta
- a definição de uma L-AG (baseada na gramática da sintaxe abstrata)
- a definição de correspondência entre as regras das duas gramáticas
- a inclusão de variáveis globais para maior eficiência do avaliador

II - O compilador LINDA

Pode-se ver que a avaliação dos atributos em uma L-AG deve ser feita usando um analisador descendente. O avaliador construído pelo compilador LINDA simula este analisador da seguinte forma: um analisador ascendente R*S realiza a análise sintática do texto fonte, e constrói uma árvore de sintaxe abstrata. Para a simulação, os nós dessa árvore são visitados na ordem adequada para a avaliação de atributos em uma L-AG.

Três pontos devem ser acrescentados:

- O método de análise sintática R*S pré-calcula o efeito das regras simples (não-terminal → não-terminal), de forma que essas regras não são incluídas na árvore.

¹Racimec Informática Brasileira S.A. - Estrada dos bandeirantes 10710 - Rio de Janeiro - CEP 22700 - (021) 342-8484

²Pontifícia Universidade Católica do Rio de Janeiro - R. Marques de São Vicente 225 - RDC sala 415 - Rio de Janeiro - CEP 22451 - (021) 274-4449

- A árvore de sintaxe abstrata é gerada da forma usual, exceto pelo fato de que as regras não-simples da gramática da sintaxe concreta são substituídas pelas correspondentes da gramática da sintaxe abstrata.

- Para maior eficiência, a análise léxica não é repetida durante a simulação de análise sintática mencionada acima. Isto é evitado guardando os nomes dos itens léxicos (*tokens*) em um arquivo, para posterior processamento.

A escolha das gramáticas que definem as sintaxe concreta e abstrata da linguagem de trabalho obedece às regras usuais: Na sintaxe concreta são incluídos aqueles aspectos necessários para evitar ambiguidade, garantir a precedência de operadores, e em geral, usar a pontuação ou sinalização para melhorar a legibilidade da linguagem. Por exemplo, delimitadores das diversas seções de um programa, possivelmente usados de forma redundante, podem facilitar a legibilidade pelo lado do programador, e a recuperação de erros, no caso do compilador. Na sintaxe abstrata, por outro lado, retira-se os elementos redundantes, e não há necessidade de preocupação com aspectos como ambiguidade e precedência, uma vez que a estrutura de árvore usada pela sintaxe abstrata elimina esses problemas.

Em resumo, esta ferramenta toma a forma de um compilador, sendo as descrições das gramáticas e dos seus vínculos realizadas através de uma linguagem especialmente projetada chamada LINDA. O compilador LINDA fará uma verificação dos tipos dos atributos, verificará se a gramática de atributos é uma L-AG, e irá gerar um programa avaliador de atributos incluindo as respectivas ações semânticas. Detalhes da implementação do compilador, e de funcionamento do avaliador de atributos se encontram em WIENSKOSKI (1990).

Foi combinada à técnica citada o analisador R*S. Este elimina reduções em produções simples, permitindo a economia do espaço de memória que seria alocado para os atributos, e a otimização do processamento do avaliador de atributos por evitar "cópias" dos atributos.

A linguagem LINDA permite ainda transformar regras não simples em regras semanticamente simples, o que significa estender as vantagens do analisador R*S para, virtualmente, qualquer produção da linguagem, e permite através do uso de elementos de programa Pascal o tratamento de variáveis globais e efeitos colaterais, necessário para atingir a eficiência desejada.

Todos os resultados obtidos da aplicação desta ferramenta são apresentados em código fonte Pascal, facilitando, caso isso seja conveniente, a interferência direta do projetista.

III - Referências Bibliográficas

- BOCHMANN, G. V., (1976) "Semantic evaluation from left to right", Communications of the ACM, feb. 1976, vol 19, n^o 2, pp. 55-62.
- DERANSART, T. P., JOURDAN, M., LORHO, B., "Attribute Grammars, Definitions, Systems and Bibliography", Springer-Verlag, Berlin, Lectures Notes in Computer Science, vol 323, (1988).
- RANGEL, J. L., (1988) "Manual de operação do sistema de geração de analisadores sintáticos R*S simples", Departamento de informática PUC-RJ, (1988).
- WIENSKOSKI Jr., E., (1990) "Método Prático de Geração de Avaliadores de Atributos em Um Passo", Tese de mestrado COPPE / UFRJ, (1990).