GERAÇÃO AUTOMÁTICA DE ANALISADORES SINTÁTICOS PARA O SPD - EVOLUÇÃO E ESTADO DA ARTE

João José Neto
EPUSP - Lab. de Sist. Digitais
Caixa Postal 11.455 - CEP 05508 - SP

O SPD - sistema de apoio ao projeto e desenvolvimento de sistemas digitais |1| foi desenvolvido inicialmente com a finalidade de auxiliar o projetista de sistemas digitais a obter rápida e eficientemente um apoio de "cross-software" para a elaboração de novas máquinas |7|. Com este enfoque, foi desenvolvido um compilador através de cuja linguagem de entrada (LD - linguagem de definição) são definidos o hardware e o conjunto de instruções de uma máquina |2|. O compilador converte esta descrição em um programa executável cujo financiamento simula a máquina descrita. Este programa é executado através do acionamento de seus módulos, por intermédio de uma linguagem de comandos, a LE - linguagem de execução, que permite operar o simulador automaticamente gerado pelo SPD. Posteriormente |3| as capacidades do SPD foram ampliadas, permitindo além da definição de simuladores, a descrição por parte do usuário do funcionamento de um montador para a máquina simulada. Esta descrição também é realizada por intermédio da LD, através do uso de um conjunto de extensões da linguagem original, dando ao final do sistema a oportunidade de programar a máquina simulada através de uma linguagem de montagem ("assembler") de sua preferência.

Dispõe-se atualmente, no SPD, destas duas ferramentas, que estão em operação. Com o intuito de dar ao projetista de sistemas de programação a oportunidade de obter, em prazo curto, uma linguagem de alto nível para a nova máquina, procurou-se, paralelamente aos desenvolvimentos do SPD acima descritos, criar uma infraestrutura para o desenvolvimento semi-automático de compiladores e

outros processadores de linguagem. Para isto, com base em experiências bem sucedidas realizadas com o ensino e com a construção de compiladores no Laboratório de Sistemas Digitais, foi primeiramente feito um estudo teórico dos mecanismos através dos quais os processadores de linguagem poderiam ser representados, resultando deste estudo a formalização de um modelo que permite a descrição e a operação de reconhecedores sintáticos para linguagens livres de contexto gerais. A publicação [4] descreve detalhadamente tal formalização, e mostra que reconhecedores por ela descritos podem ser obtidos para qualquer linguagem livre de contexto. Com a garantia deste lastro teórico, foi em seguida elaborado um algoritmo para a geração de um reconhecedor sintático a partir de uma gramática BNF da linguagem livre de contexto cujo compilador se deseja construir [5]. Este algoritmo efetua transformações gramaticais, através das quais é possível obter uma gramática equivalente à fornecida mas que exibe algumas características que auxiliam na obtenção do reconhecedor sintático desejado.

Obtém-se desta maneira uma gramática que, manipulada por um outro algoritmo de transdução, é mapeada em um conjunto de regras que obedecem ao modelo formal anteriormente mencionado. Através deste algoritmo, obtém-se um reconhecedor sintático para a linguagem, que opera em tempo linear com o comprimento do texto de entrada em todos os casos de interesse, o que o torna aplicável em casos onde não seja indispensável a preservação da estrutura exibida pela gramática inicialmente fornecida. Para solucionar este problema, foram estendidos os algoritmos de transformação da gramática e de geração de reconhecedor, criando-se descritores de relação da estrutura da gramática transformada com a da gramática original. Por meio de consulta a tais descritores da estrutura e à gramática transformada, um algoritmo de tradução, descrito em [6] é capaz de adicionar, às produções geradas pelo algoritmo que gera o reconhecedor sintáti

co, regras de mapeamento a serem escolhidas. Na publicação mencionada, é mostrado um exemplo de aplicação, extremamente geral, que fornece as regras de mapeamento adequadas à transformação de uma gramática livre de contexto dada em BNF em um transdutor sintático cuja saída consiste na árvore de reconhecimento do texto, fornecido na linguagem de entrada. Naturalmente, esta árvore pode ser utilizada como linguagem de entrada de um novo passo de compilação, no caso de ser adotado um esquema de compilação em mais de um passo. O desenvolvimento de compiladores torna-se através desta ferramenta, bastante facilitado, já que os primeiros problemas, correspondentes à obtenção de passos de análise, já foram totalmente automatizados. Em todas as publicações acima citadas não é mencionada a tarefa de construção de filtros léxicos necessários ao funcionamento do conjunto. Foram desenvolvidos nesta linha alguns programas voltadas à construção de tais rotinas, programas estes encarregados de criar e manter tabelas de símbolos, de efetuar a extração e a classificação de itens léxicos de um programa. Foram também implementados programas dedicados à redução dos reconhecedores obtidos pelos algoritmos automáticos de geração, e um programa que efetua a conversão das regras do reconhecedor em programa executável. Através do conjunto assim obtido, é possível partir de uma gramática BNF, e obter um analisador - filtro léxico na forma de regras ou de programa executável, compatível com o analisador sintático gerado automaticamente pelo sistema. Pretende-se, em um futuro próximo, integrar todos estes programas, obtendo assim uma ferramenta poderosa para o auxílio à construção de compiladores. Está em andamento um estudo cuja finalidade é a de criar automaticamente mecanismos para a deteção e recuperação de erros de sintaxe nos reconhecedores sintáticos. Como meta seguinte está-se planejando a incorporação de mecanismos para a definição e processamento da sintaxe dinâmica da linguagem de entrada, visando preparar o reconhecedor para receber a noção de atributos, através dos

quais a geração de código e o tratamento da semântica, indispensável à construção final dos compiladores, possa ser introduzida.

A presente publicação tem por objetivo coletar os diversos resultados parciais já obtidos nesta linha de pesquisa, apresentando-os no contexto do SPD, onde deverão ser integrados, como etapa final dos trabalhos, permitindo desta maneira a utilização plena e cômoda do sistema em sua forma mais versátil, através do uso direto de linguagem de alto nível como ferramenta para a confecção de programas para as máquinas simuladas pelo SPD, em adição aos métodos usuais já em utilização no Laboratório de Sistemas Digitais.

[1] - José Neto, J.
"SPD - Um sistema automático de apoio ao projeto e desenvolvimento de Sistemas Digitais"
Tese de Doutorado, EPUSP, 1980

[2] - José Neto, J. e Marangoni, C.
"Simulação Digital. Uso do SPD na simulação de Microprocessadores"
2º Simpósio sobre Desenvolvimento de Software Básico para Micros - S.Paulo, 1982

[3] - José Neto, J.
"Cross-Assemblers" para Microprocessadores - Geração automática através do SPD
1º CONAI - São Paulo, 1983

[4] - José Neto, J. e Magalhães, M.E.S.
"Reconhecedores Sintáticos - Uma alternativa didática para o uso em cursos de engenharia"
XIV CNI - SUCESU - São Paulo, 1981

|5| - José Neto, J. e Magalhães, M.E.S.
"Um Gerador Automático de Reconhecedores Sintáticos para o SPD"
VIII SEMISH - Florianópolis, 1981

|6| - Magalhães, M.E.S. e José Neto, J.
"Um gerador automático de núcleos eficientes para compiladores dirigidos por sintaxe"
X SEMISH - Campinas, 1983

|7| - José Neto, J.
"A automação do projeto e desenvolvimento de Sistemas Digitais: o SPD como ferramenta para a obtenção automática de "Cross-Software" para sistemas de computação"
4º Congresso Brasileiro de Automática - Campinas, 1982