

UMA SOLUÇÃO ADAPTATIVA PARA RECONHECEDORES SINTÁTICOS

João José Neto, Assistente-Doutor, EPUSP

ANAIS EPUSP.

UNITERMOS: Compiladores dirigidos por sintaxe. Transdutores. Autômatos.
Linguagens formais. Dependências de contexto. Extensibilidade

UMA SOLUÇÃO ADAPTATIVA PARA RECONHECEDORES SINTÁTICOS

João José Neto
Escola Politécnica da USP
Prof. Assistente Doutor
Departamento de Eng. de Eletricidade

RESUMO

Apresenta-se uma solução nova para a elaboração de núcleos reconhecedores eficientes para compiladores dirigidos por sintaxe. Esta solução se baseia no conceito, aqui introduzido, de transdutor adaptativo, uma classe de máquinas de estados finitos com memória organizada em pilha, e que exhibe recursos de aprendizado baseados na alteração dinâmica de sua configuração, com base nas transições efetuadas pelo transdutor.

ABSTRACT

The present paper presents a new solution for the problem of building efficient recognizers for use as kernels in syntax-driven compilers. Adaptive transducers, which are introduced here as the basis for that solution, are a class of finite-state devices with stack memory, which feature learning capability through transition-driven self-modification.

PALAVRAS-CHAVE

Compiladores dirigidos por sintaxe, transdutores, autómatos, linguagens formais, dependências de contexto, extensibilidade.

1. APRESENTAÇÃO

Tradicionalmente, compiladores de linguagens de programação têm utilizado as soluções clássicas, baseadas em autômatos finitos e de pilha, para a elaboração dos mecanismos básicos de reconhecimento para as linguagens sobre as quais devem operar.

Não são usualmente considerados, a nível sintático, quaisquer aspectos associados às dependências de contexto, às estruturas de blocos e correspondentes problemas ligados ao escopo das variáveis, aos tratamentos de macros, aos testes de consistência de utilização para os tipos das variáveis, a toda a problemática ligada à extensibilidade sintática das linguagens e a tantos outros aspectos puramente sintáticos.

O tratamento destes problemas é, geralmente, excluído da análise sintática, migrando para secções impróprias do compilador, de cujo originalmente semântico.

No presente texto, apresenta-se, de forma puramente intuitiva e sem nenhum rigor formal, o conceito de transdutor adaptativo, uma modalidade de transdutor sintático baseada no modelo formal descrito em [1].

2. HISTÓRICO

Originalmente, esta pesquisa girou em torno da busca de um modelo matemático adequado para a representação de autômatos estruturados, com potencial para o reconhecimento de qualquer linguagem livre de contexto, e passível de manipulação no sentido de produzir reconhecedores de alto desempenho.

Este modelo mostrou-se extremamente adequado aos propósitos que estavam sendo perseguidos, uma vez que permite a utilização de submáquinas mutuamente recursivas para o reconhecimento da linguagem, cada uma das quais operando como autômato finito enquanto executa transições internas.

Uma memória, organizada na forma de pilha, complementa o autômato, de modo que transições entre submáquinas possam ser executadas, permitindo desta forma a chamada de submáquinas como se fossem sub-rotinas umas das outras. A pilha é utilizada, no caso, para memorizar, na ocasião da chamada, o estado para onde deverá ser efetuada a transição de retorno, informando, na ocasião do retorno à submáquina chamadora, o estado para onde a transição de retorno deve ser executada.

Prova-se [1] que autômatos organizados desta forma têm a mesma potência dos autômatos de pilha clássicos, sendo portanto capazes de representar qualquer linguagem livre de contexto.

Através de algoritmos relativamente simples, é possível construir automaticamente, para uma linguagem definida através de uma gramática livre de contexto, autômatos estruturados de acordo com esse modelo, por meio de manipulações das gramáticas que as definem [2].

Uma evolução do trabalho resultou na elaboração de um método para a construção automática de árvores de reconhecimento sintático completas como resultado da aplicação de um transdutor sintático derivado do modelo em questão [3].

Inúmeras aplicações didáticas são exibidas por esse modelo, devido à sua aplicabilidade direta, fácil assimilação por não-especialistas, baixa complexidade, etc. [1,4].

O trabalho relatado neste artigo corresponde à conclusão de mais uma importante etapa desta pesquisa, através de um substancial aumento na capacidade de reconhecimento do modelo: por meio de recursos de aprendizado, o transdutor passa a ter a possibilidade de se auto-modificar, em função do texto cuja análise realiza.

As implicações desta nova capacidade do modelo refletem-se diretamente na classe de linguagens que ele passa a ter capacidade de tratar, permitindo explorá-lo para aplicações em que os modelos convencionais se mostram sintaticamente ineficazes, como ocorre especialmente no caso do tratamento das dependências de contexto e dos aspectos da extensibilidade das linguagens.

3. CONCEITOS

Antes de passar ao estudo dos transdutores adaptativos, convém recordar os elementos básicos de um transdutor sintático.

Essencialmente, um transdutor sintático finito tradicional é um dispositivo de tradução com poder de reconhecimento de linguagens regulares, ou seja, idêntico ao dos autômatos finitos, e consta dos seguintes componentes fundamentais:

- um conjunto Q de estados
- um alfabeto de entrada W
- um alfabeto de saída S
- uma função de transdução D , que mapeia pares (q,w) em pares (q',s') , com q, q' estados, w símbolo de entrada e s cadeia de símbolos de saída
- um estado inicial q_0
- um conjunto de estados finais F , subconjunto de Q .

A operação básica do transdutor finito consiste em, sendo w_0 uma cadeia de símbolos de entrada a ser reconhecida, partindo-se de uma situação inicial (q_0, w_0) , atingir uma situação final $(q_f, \#)$, onde q_f é estado final e $\#$ denota a cadeia vazia.

Esta operação é efetuada atingindo-se, sucessivamente, situações intermediárias (q_i, w_i) pela aplicação repetida da função de transdução D aos pares formados pelos estados q_j e pelos símbolos z_j de entrada que iniciam as cadeias w_j que define cada situação intermediária (q_j, w_j) .

Concatenando-se sucessivamente as diversas cadeias de saída s_j determinadas pela função de transdução, obtém-se uma cadeia s_0 de símbolos do alfabeto de saída, a tradução da cadeia de entrada w_0 , efetuada conforme as regras ditas pelo

transdutor.

Pode-se dizer que o transdutor reconheceu a cadeia w_0 se e apenas se o percurso acima descrito for concluído com sucesso. Neste caso, diz-se que a cadeia w_0 é uma sentença da linguagem de entrada definida pelo transdutor, e que este traduziu w_0 para uma sentença s_0 da linguagem de saída definida pelo mesmo transdutor.

Define-se de modo análogo o funcionamento de um transdutor de pilha. Neste caso, o dispositivo utiliza-se de uma pilha como memória auxiliar para guardar a história do reconhecimento sintático efetuado.

Um transdutor de pilha apresenta, portanto, algumas diferenças em relação aos transdutores finitos:

- inclui um alfabeto de pilha G
- inclui um marcador de pilha vazia g_0
- exibe uma função de transdução D , mapeamento que leva em consideração o estado da pilha para executar as transições: mapeia triplas (q, g, w) em triplas (q', g', s) , ou seja, produz uma cadeia s de saída sempre que, encontrando-se no estado q , com g no topo da pilha, o transdutor receber um símbolo de entrada w , consumindo-o e transitando para o estado q' . O conteúdo do topo da pilha é alterado para g' neste caso.

Para o caso de transdutores de pilha estruturados, pode-se encontrar diferenças adicionais, devido à necessidade de mecanismos que garantam uma disciplina estruturada na organização dos transdutores:

- um conjunto A de submáquinas transdutoras
- uma submáquina transdutora inicial a_0 pertencente a A

Cada sub-máquina transdutora apresenta-se como um transdutor finito em seu funcionamento e em sua estrutura, sendo que as alterações eventuais da pilha são efetuadas exclusivamente através de movimentos especiais do transdutor, na forma de transições em vazio:

- transições de chamada de sub-máquina: através deste tipo de movimento, o transdutor inicia transições em uma sub-máquina, eventualmente diversa daquela que executou tal transição. Este tipo de movimento é feito sem consumo de símbolos de entrada, e encarrega-se de memorizar, na pilha, informações acerca do estado para onde deverá ser efetuada uma transição de retorno ao final da operação da sub-máquina chamada. Opcionalmente, ao ser executada uma transição deste tipo, é possível emitir uma cadeia de símbolos do alfabeto de saída.

- transições de retorno à sub-máquina chamadora: através desta classe de transições em vazio, abandona-se a sub-máquina correntemente ativa, retornando-se à operação da sub-máquina que a ativou. Para tanto, é efetuado um desempilhamento das informações previamente armazenadas na pilha, com base nas quais é determinada a sub-máquina chamadora e o estado correto para onde será efetuada a transição em vazio de retorno. Como no caso

anterior, uma cadeia de símbolos do alfabeto de saída pode ser emitida nesta ocasião.

Os transdutores adaptativos, aqui introduzidos, baseiam-se fortemente na estrutura dos autômatos de pilha estruturados descritos acima. Exibem todos os seus elementos básicos, incorporando as seguintes características adicionais:

- uma função de modificação M , destinada a efetuar alterações estruturais sobre o transdutor. Esta função é aplicada sobre o transdutor a cada transição executada, e sua ação consiste em alterar, se necessário, os conjuntos Q , F e A de estados, de estados finais e de submáquinas, os alfabetos W , S , L e G , a função de transição D e a própria função de modificação M , transformando-as respectivamente em conjuntos Q' , F' e A' , W' , S' , L' e G' , e em funções D' e M' . Estes símbolos denotam o resultado da modificação operada pela função M sobre os elementos correspondentes, definidos anteriormente.

- uma função de modificação inicial M_0 , utilizada pelo transdutor adaptativo para dar início às suas operações.

- a inclusão da característica de inserção automática, na ocasião dos retornos de submáquina, na cadeia de entrada, de átomos indicadores do estado final através do qual foi efetuado o retorno.

A operação de um transdutor adaptativo pode ser analisada em duas etapas: na primeira, uma transição convencional tem lugar como se se tratasse de um simples transdutor de pilha, e na segunda, a função de modificação é aplicada sobre o transdutor, provocando eventuais alterações na estrutura do mesmo, através da inclusão ou exclusão de estados ou de transições, de modo que, após esta operação, o transdutor esteja configurado adequadamente para prosseguir seu trabalho, porém agora munido de informações adicionais, aprendidas durante o processamento já realizado.

Diz-se que um transdutor adaptativo que define uma linguagem de entrada L_e e sua tradução para uma linguagem de saída L_s aceita uma cadeia w de símbolos do alfabeto de entrada e a traduz para uma cadeia s de símbolos do alfabeto de saída quando, partindo de sua situação inicial, o transdutor efetua sucessivos movimentos até atingir uma situação final.

Caracteriza-se uma situação inicial do transdutor adaptativo como sendo aquela em que o transdutor se apresenta com os seus conjuntos iniciais de estados ($Q=Q_0$), de submáquinas ($A=A_0$), de símbolos de entrada ($W=W_0$), de pilha ($G=G_0$), de estados finais ($F=F_0$). O seu estado corrente deve ser o estado inicial q_0 , sua pilha deve estar em estado de repouso, ou seja, com g_0 em seu topo, e a cadeia de entrada w deve estar intacta, totalmente por ser analisada.

Uma situação final é atingida ao se esgotar a cadeia de entrada, ou seja, quando a cadeia a ser analisada ficar reduzida à cadeia vazia ($\#$), estando o transdutor em algum dos estados do conjunto corrente F de estados finais, com a pilha em estado de repouso.

Os movimentos do transdutor adaptativo são

caracterizados pela passagem do transdutor de uma situação para a situação seguinte, ou seja:

- com base no próximo elemento a ser analisado da cadeia de entrada, no estado corrente do transdutor e no conteúdo do topo da pilha, promove-se transições de estado e atualizações na pilha do transdutor.

- pela aplicação da função corrente de modificação M, é efetuada uma eventual alteração do transdutor, correspondente à ação de aprendizado, característica do transdutor adaptativo, que não é exibida pelos outros modelos anteriormente apresentados.

4. ANALISE

Através da observação das propriedades exibidas pelos modelos descritos, é possível notar uma série de potencialidades extremamente convenientes, resultantes da maneira como esses dispositivos estão concebidos, bem como de disciplinas que podem ser aplicadas quando da construção de compiladores neles baseados.

Não se pretende neste texto provar ou teorizar esta dimensão dos transdutores adaptativos, mas sim relatar os principais resultados que podem ser encontrados ao se efetuar sobre os modelos em questão estudos formais mais rigorosos [5]:

- os transdutores adaptativos têm potencial para tratar as dependências de contexto usualmente exibidas pelas linguagens de programação.

- os transdutores adaptativos permitem tratar com muita facilidade alguns aspectos importantes de linguagens extensíveis, tais como a definição de macros e sua expansão.

- é muito confortável o uso de transdutores adaptativos para o tratamento puramente sintático de dependências de contexto como aquelas exibidas pelas linguagens de alto nível com estrutura de blocos, em relação à declaração de variáveis e o controle de seu escopo.

- torna-se trivial o teste de consistência de tipos ("type-checking") com o uso de transdutores adaptativos, uma vez que é possível, com tais dispositivos, efetuar tais testes através de mecanismos exclusivamente sintáticos, de forma totalmente transparente ao projetista.

- a explicitação de cadeias de saída, associadas às transições, torna os transdutores adaptativos mais confortáveis que os autômatos usualmente empregados na construção de núcleos de compiladores dirigidos por sintaxe.

- como os transdutores adaptativos são essencialmente autômatos de pilha estruturados, enriquecidos com alguns recursos adicionais, preservam daqueles as propriedades de reconhecimento de linguagens livres de contexto, através da utilização de submáquinas que operam como transdutores finitos. Assim sendo, torna-se possível obter transdutores adaptativos de eficiência muito grande, através do uso de conjuntos adequados de submáquinas otimizadas.

- pela mesma razão é possível utilizar métodos relativamente simples para a construção, manual ou mesmo automática, dos

transdutores adaptativos, com base nos algoritmos já conhecidos de construção de autômatos de pilha estruturados.

Estas e outras muitas propriedades exibidas pelo modelo fazem com que ele se torne não apenas uma alternativa muito atraente como também uma ferramenta extremamente poderosa aplicável à elaboração de núcleos eficientes de compiladores dirigidos por sintaxe.

5. EXEMPLO

O exemplo seguinte foi incluído neste texto com a finalidade única de dar uma idéia de como pode ser utilizada, na prática, a idéia encerrada no conceito de transdutor adaptativo, para aplicação direta à construção de um núcleo de compilador dirigido por sintaxe, com tratamento puramente sintático das dependências de contexto ligadas aos escopos (estáticos) e aos testes de consistência de tipos para identificadores de variáveis em uma linguagem estruturada em blocos.

A linguagem em questão é formada de textos em que são inicialmente declaradas variáveis, de tipos inteiro ou booleano, através de declarações específicas. A seguir, entre as palavras-chave BEGIN e END, uma seqüência de comandos de atribuição compõem o corpo do programa.

Não são permitidas atribuições de expressões que envolvam tipos diferentes do da variável-destino, nem operações inadequadas sobre as variáveis.

Os diagramas abaixo representam as submáquinas de um transdutor de pilha estruturado adaptativo, em seu aspecto livre de contexto. Os complementos, responsáveis pelo tratamento das dependências de contexto, são realçados, intuitivamente apenas, através do texto que segue os diagramas.

São considerados átomos da linguagem as palavras reservadas (BEGIN, END, INTEGER, BOOLEAN) e os caracteres especiais. Letras e dígitos são tratados ao nível das submáquinas que extraem os identificadores e números inteiros. Compõe-se, desta forma, a especificação de um analisador léxico para esta linguagem. Note-se que, por conveniência, não serão feitas quaisquer distinções, entre as submáquinas encarregadas das análises léxica e sintática.

identificador:



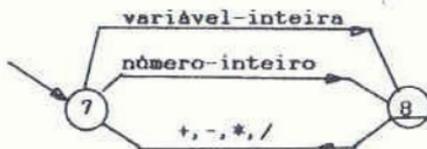
número-inteiro:



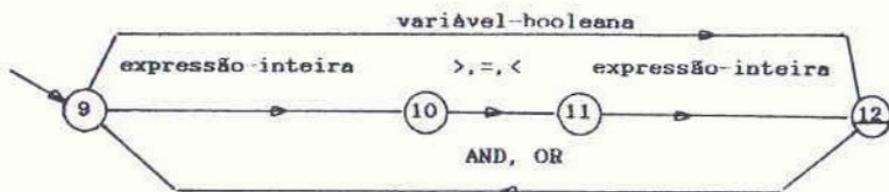
variável: (utilizada para variável-inteira e variável-booleana)



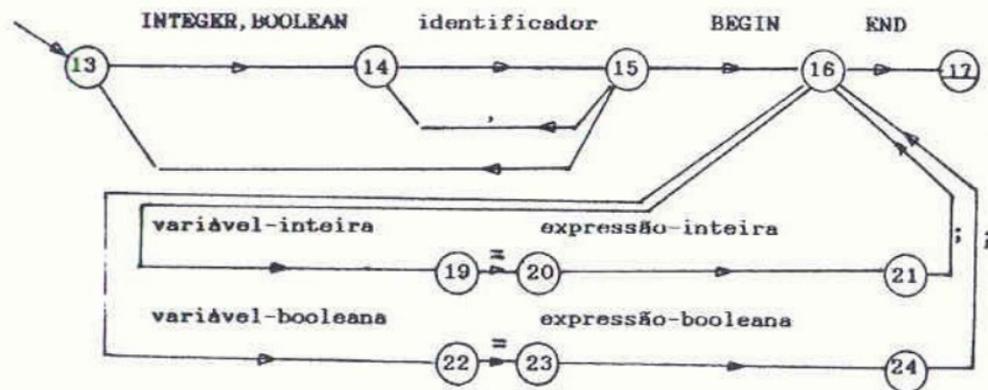
expressão-inteira:



expressão-booleana:



programa:



Os diagramas acima representam a situação inicial do transdutor. Para um programa como o seguinte:

```

[1]      INTEGER x,y,z;
[2]      BOOLEAN i,j;
[3]      INTEGER m,n;
[4]      BEGIN
[5]          x=1+y-z;
[6]          i=j AND x < y-1;
[7]          i=m;
[8]          b=4;
[9]      END

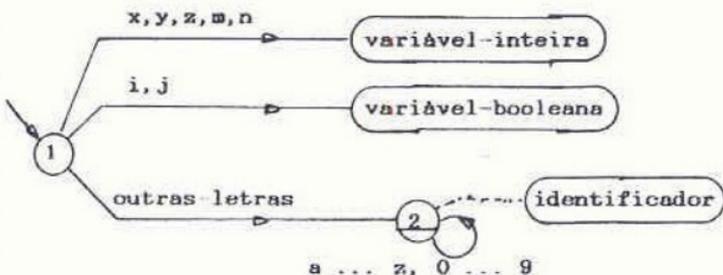
```

as transições entre os estados 14 e 15 da submáquina correspondente ao não-terminal programa encarregam-se de ativar a submáquina identificador a toda ocorrência dos identificadores nas declarações das linhas 1, 2 e 3 do programa. A função de modificação do transdutor deve encarregar-se de criar, para cada um desses identificadores, as transições que se fizerem necessárias ao seu reconhecimento na submáquina variável, a qual deverá emitir, como saída, os átomos variável-inteira ou variável-booleana, conforme os identificadores considerados.

Note-se que a operação da submáquina identificador deve ser alterada ligeiramente: para que seu funcionamento seja determinístico, torna-se necessário fundi-la com a submáquina variável, fazendo-se ainda com que a submáquina resultante emita como saídas os átomos identificador, variável-inteira e variável-booleana, de acordo com o estado final por onde ela tenha concluído sua operação.

Do ponto de vista prático, o reconhecimento das palavras reservadas pode ser efetuado também nesta submáquina, para dar uniformidade ao tratamento de todos esses elementos.

Após o tratamento de todas as declarações, a submáquina em questão torna-se (sem considerar as palavras reservadas):



Note-se a total prescindibilidade de quaisquer mecanismos adicionais de testes de consistência de tipos, uma vez que o próprio transdutor se encarrega de executá-los, por meios próprios e totalmente transparentes ao programador do sistema. Observe-se o comportamento do transdutor nos comandos incorretos das linhas 7 e 8 do programa-exemplo: no primeiro caso, encontrada a variável booleana i, o mecanismo de análise sintática força a ativação da submáquina expressão-booleana, que

exigiria uma comparação entre expressões-inteiras, o que não é satisfeito, provocando a detecção de um erro de sintaxe; no segundo caso, o identificador b, não tendo sido declarado anteriormente, será interpretado pela análise léxica como um simples identificador, provocando assim a detecção de um outro erro de sintaxe ao tentar transitar a partir do estado 16.

6. CONCLUSÃO

Embora muito simples e incompleto, o exemplo acima serve-se para mostrar, sem qualquer rigor ou formalismo, uma parte da potencialidade do transdutor adaptativo. Note-se que, mesmo neste exemplo trivial, uma série de ganhos podem ser identificados: a análise puramente sintática do programa, a eliminação de programas distintos para o tratamento das tabelas de símbolos, tabelas de atributos, tabelas de palavras reservadas, testes de consistência da utilização dos identificadores, verificação de escopo dos identificadores, etc..

Explorando este conceito de transdutor adaptativo, em conjunto com a aplicação das técnicas já desenvolvidas de obtenção automática de reconhecedores eficientes baseados em autômatos estruturados, torna-se disponível mais uma poderosa ferramenta para o projetista de sistemas de programação interessado na aplicação de técnicas eficientes para a produção de núcleos confiáveis e de baixa complexidade para compiladores dirigidos por sintaxe.

Espera-se, contando novamente com a versatilidade do modelo, tornar mais atraente e popular a manipulação de linguagens com recursos de extensibilidade, já que se torna baixo o custo de inclusão de tais recursos na implementação de uma linguagem que apresente tais características.

7. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] JOSÉ NETO, J.; MAGALHÃES, M.E.S. Reconhedores sintáticos: uma alternativa didática para uso em cursos de engenharia. In: CONGRESSO NACIONAL DE PROCESSAMENTO DE DADOS, 14, São Paulo, 1981. Anais. São Paulo, SUCESU, 1981. p.171-81.
- [2] _____; _____. Um gerador automático de reconhedores sintáticos para o SPD. In: SEMINÁRIO INTEGRADO DE SOFTWARE E HARDWARE, 8, Florianópolis, 1981. Anais. Florianópolis, SBC/UFSC, 1981. p.213-38.
- [3] MAGALHÃES, M.E.S.; JOSÉ NETO, J. Um gerador automático de núcleos eficientes para compiladores dirigidos por sintaxe. In: SEMINÁRIO INTEGRADO DE SOFTWARE E HARDWARE, 10, Campinas, 1983. Anais. Campinas, SBC/UNICAMP, 1983. v.1, p.105-21.
- [4] JOSÉ NETO, J. Introdução a compilação. Rio de Janeiro, LTC, 1987. 222p. (Engenharia de Computação)
- [5] JOSÉ NETO, J. Tese de Livre-docência a ser apresentada ao Departamento de Engenharia de Eletricidade da Escola Politécnica. Em preparação.

Data de entrega: dezembro/83