PROJETO MORFOL
UMA FERRAMENTA PARA ANÁLISE LÓGICA DE CENAS

Aluno: Marco Antônio Barbosa Teixeira
Orientador(es): Edward Hermann Haeusler e Geiza Maria Hamazaki da Silva

Introdução

Este projeto é uma continuidade do Projeto MORFOL[1], cujo objetivo foi o desenvolvimento de uma ferramenta que possibilita a análise morfológica de uma linguagem, através da descrição de um modelo, com base em uma descrição lingüística em linguagem lógica gerando código na linguagem C. Na ferramenta MORFOL, dado um modelo descrito na Linguagem de Descrição é gerado código em Linguagem C para o mesmo modelo, o qual utiliza primitivas de segmentação do Juiz Virtual como primeiro estudo de caso. A linguagem de descrição tem como base uma descrição em linguagem lógica semelhante ao Prolog.

Objetivos

Este projeto objetiva dar continuidade no desenvolvimento da ferramenta MORFOL, e concluir o protótipo inicial da ferramenta, onde será realizada a validação, através de testes sobre os códigos gerados, além da integração deste código com o JV. Serão realizadas comparações entre o método utilizado pelo JV, e o método proposto em MORFOL. É esperado obter subsídios para uma análise mais conclusiva do uso de princípios morfológicos em detrimento de reconhecimento estatístico de cenas no domínio do Juiz Virtual. A linguagem utilizada será amplificada com adição do operador de corte (!), que deverá tornar o processamento mais eficiente.

Metodologia

Na primeira etapa foi finalizada a ferramenta de geração de código, sendo aplicados testes, onde o código gerado foi executado para a verificação dos resultados de suas interpretações. Inicialmente, o código foi testado utilizando valores definidos manualmente. Após esse passo o mesmo código foi integrado ao sistema do JV, utilizando segmentos “reais” reconhecidos pelo programa. Com isto, o código foi testado em situações reais, e foi verificado os ajustes à ferramenta para uma fácil integração do código com outros programas.

Após os testes de integração para a validação do código gerado, foram estudadas e acrescentadas novas “funcionalidades” à MORFOL, como a inclusão do símbolo de corte à linguagem, para tratamento e otimização do algoritmo.

Para o desenvolvimento da ferramenta MORFOL, foi utilizada a linguagem de transformação TXL, que é usada para a Transformação/tradução entre o código de especificação de MORFOL e C. E para a integração do código gerado com o do JV, foi utilizado o compilador GCC, o ambiente de programação Visual Studio 2005/2008, com utilização de ferramentas gráficas, como Glut/ openGL, iup, im, e outras.

Desenvolvimento

Após a implementação da ferramenta de tradução, foi inicializada a fase de teste do código gerado. Esta foi realizada em dois momentos: no primeiro, testamos o código em busca de falhas no processo de interpretação realizado pelo código e no seguinte foi realizado a integração do código ao programa do JV, visando simular o código em situações reais, e verificar os ajustes à ferramenta necessários para uma rápida integração, de preferência automática, de código com programas em contextos diferentes do aplicado. Nesta etapa, foi realizada uma verificação de todos os itens essenciais para a execução da interpretação, como a disponibilização do conjunto de segmentos para a função de interpretação gerada e a forma de retorno das soluções encontradas(é desejado encontrar apenas uma solução correta).

Na verificação de resultados foi implementado um conjunto de funções para ser utilizado temporariamente na interligação entre JV e a função de interpretação.

Não menos importante, é a inclusão do operador de corte (!), comum à linguagens como Prolog, evitando o processamento de certos ramos da árvore de busca, tornando este mais eficiente, tanto em relação a espaço de memória como tempo de processamento.

Resultados

Exemplo de Especificação em MORFOL(aplicando corte)	*Saída obtida pela transformação*

Base:
b = B
c = C
d = C
d = D
e = E
f = F
g = G
 	
Def:
 	
A(xa,xb,xc,xd,xf,xi,xj)=b(xa,xb,xc),c(xb,xa,xf) , ! , d(xf,xd,xi) , \
e(xi,xj,xb) , ! , f(xg,xa,xd) , g(xa,xc).
 	
Cod:
A(xa,xb,xc,xd,xf,xi,xj)
	A(xa,xb,xc,xd,xf,xi,xj){
	for(xa=0; xa < MAXLINHAS; xa ++){
	for(xb=0; xb < MAXLINHAS; xb ++){
	for(xc=0; xc < MAXLINHAS; xc ++){
	for(xf=0; xf < MAXLINHAS; xf ++){
	if(! B(xa,xb,xc)) continue ;
	if(! C(xb,xa,xf)) continue ;
	for(xd=0; xd < MAXLINHAS; xd ++){
	for(xi=0; xi < MAXLINHAS; xi ++){
	for(xj=0; xj< MAXLINHAS; xj++){
	if(!D(xf,xd,xi)); if(!E(xi,xj,xb));
	for(xg=0; xg< MAXLINHAS;xg++){
	if(F(xg,xa,xd)&&G(xa,xc)){
	return OK; }}
	return ERRO;
	}}}
	return ERRO;
	} }}}return ERRO;}

Conclusões

A utilização de uma linguagem de descrição para reconhecer padrões estruturais da imagem que estejam formalmente relacionados apresentou resultados interessantes como a facilidade na descrição do protótipo ou modelo (não se limitando somente à área de PDI), a rápida implementação do mesmo, além da geração de um código legível. A integração do código gerado com outra aplicação levantou novos problemas que muitas vezes não são visualizados na etapa de projeto da aplicação. É interessante citar a inclusão de comandos de descrição para auxilio na geração automática de funções intermediárias, e para estabelecer critérios de seleção de soluções “preferenciais”, mesmo que o desenvolvimento e o projeto desta linguagem sejam parciais e incompletos. E o desenvolvimento de uma linguagem de fácil descrição para determinação dos critérios válidos para a seleção da solução. A inclusão do operador de corte à linguagem tornou o processo de reconhecimento mais eficiente, além de auxiliar na eliminação de soluções incorretas

Como trabalho futuro podemos citar o aperfeiçoamento da ferramenta MORFOL, através de alterações na ferramenta para otimização tanto no processo de geração de código como no código gerado. Podemos citar também o estudo sobre a viabilidade da utilização da ferramenta em outras aplicações.

Referências

[1] TEIXEIRA, M.A.B. PROJETO MORFOL: Uma Ferramenta para Análise Lógica de Cenas – Projeto de Iniciação Científica - Departamento de Informática - PUC-Rio. 2008.
[2] FELIX, M. F. LET: Uma Linguagem para Especificar Traduções e seu Compilador. Dissertação de Mestrado - Departamento de Informática - PUC-Rio. 1998.
[3] Szenberg, F. Acompanhamento de Cenas com Calibração Automática de Câmeras. Tese de Doutorado - Departamento de Informática - PUC-Rio. 2001.
[4] Palazzo,L. A. M. Introdução à Programação Prolog. EDUCAT, Pelotas, 1997.
[5] http://www.w3.org/Style/XSL/