

Processamento de Linguagens (3º ano de MiEI)

Trabalho Prático 2

Relatório de Desenvolvimento de uma *LPIS*

Gustavo da Costa Gomes-Aluno
(72223)

José Carlos da Silva Brandão Gonçalves-Aluno
(71223)

Tiago João Lopes Carvalhais-Aluno
(70443)

28 de Maio de 2016

Resumo

Isto é o resumo do relatório da unidade curricular Processamento de Linguagens relativamente ao Trabalho Prático 2.

Este visa a produção de uma linguagem de programação imperativa simples, *LPIS*, e para tal será necessário produzir uma gramática independente de contexto *GIC* com condição LR(0), bem como a produção do respetivo compilador, com auxílio do Gerador *Yacc/Flex* e da realização de um conjunto de testes, escritos na linguagem especificada na *GIC*, por forma a verificar o seu correto funcionamento.

E esse compilador, no final, produzirá pseudo-código assembly, que será interpretado pela máquina virtual e que gerará o resultado esperado.

Conteúdo

1	Introdução	2
1.1	Estrutura do Relatório	2
2	Análise e Especificação	3
2.1	Descrição informal do problema	3
2.2	Especificação do Requisitos	4
3	Concepção/desenho da Resolução	5
3.1	Estruturas de Dados	5
3.2	Algoritmos Implementados	5
4	Codificação	6
4.1	Alternativas, Decisões e Problemas de Implementação	6
4.2	Testes realizados e Resultados	6
5	Conclusão	7
A	Código do Programa	8

Capítulo 1

Introdução

Neste trabalho todos os parâmetros exigidos foram realizados, bem como a realização de um conjunto de testes que foram para além dos seis testes que foram estabelecidos como requisitos mínimos.

A realização desses parâmetros passou, inicialmente, por produzir uma gramática independente de contexto *GIC* e, após a sua validação pelo docente Pedro Rangel Henriques, por criar um compilador que reconhecesse um ficheiro de entrada, escrito nessa linguagem, e que produzisse o ficheiro *.vm* correspondente, recorrendo ao Gerador *Yacc/Flex*. Ficheiro esse que contém o conjunto das instruções assembly para a máquina virtual.

Enquadramento Utilização do Gerador *Yacc/Flex* e Desenvolvimento de uma *GIC* com o objetivo de produzir pseudo-código para a máquina virtual a partir dos padrões da linguagem produzida existentes num outro ficheiro.

Estrutura do documento Este documento possui a respectiva solução elaborada e uma conclusão final que une o exercício.

Resultados Os resultados serão apreciados no capítulo vindouro e cuja solução estará em Anexo.

Conteúdo do documento Contém a explicação do problema em si, bem como a apresentação da solução produzida para produzir o resultado esperado, auxiliada com documentação e código produzido.

1.1 Estrutura do Relatório

Este relatório possui três capítulos, uma conclusão, um capítulo extra dedicado a Anexos e a respectiva Bibliografia utilizada durante a realização deste projecto.

Os capítulos são Análise e Especificação do problema, Arquitectura da solução para o problema em questão, que envolverá a explicação das estruturas de dados utilizadas e por fim o capítulo designado Codificação, que incluirá alguns aspectos relevantes de todos os testes realizados para a verificação do seu correcto funcionamento.

Capítulo 2

Análise e Especificação

2.1 Descrição informal do problema

Este trabalho prático pretende que o compilador produzido seja capaz de construir ficheiros com o pseudo-código assembly para a máquina de stack virtual(VM) fornecido pelos docentes.

Esse compilador para realizar a sua função em pleno necessita de receber como argumento ficheiros escritos com a linguagem que também terá de ser desenvolvida e validada por um dos docentes. A linguagem a produzir deverá permitir:

- Declaração e manuseamento de variáveis atómicas do tipo inteiro, com os quais se podem realizar as habituais operações aritméticas, relacionais e lógicas.
- Declaração e manuseamento de variáveis estruturadas do tipo array (a 1 ou 2 dimensões, matrizes) de inteiros, em relação aos quais é apenas permitida a operação de indexação (índice inteiro).
- Realização de instruções algorítmicas básicas como o caso da atribuição de expressões a variáveis.
- Leitura do standard input e escrita no standard output.
- Realização de instruções para controlo do fluxo de execução, condicional e cíclica, que possam ser aninhadas.
- Definição e invocação de sub programas sem parâmetros, mas que possam retornar um resultado atómico, sendo este tópico opcional. Contudo também será produzido, e testado, por exemplo, a função "potencia(Base,Exp)".

Esta linguagem deverá ainda ter em conta que as variáveis deverão ser declaradas no início do programa e que não se poderá efetuar declarações nem utilizações sem declaração prévia das mesmas. Se nada for dito em relação ao valor que a variável toma, essa toma, por defeito, o valor zero, após a sua declaração.

Após se produzir essa gramática, a mesma deverá ser validada antes de se prosseguir. E por fim, após a validação dessa é necessário realizar as análises léxica e sintática acompanhada pelas respectivas ações a tomar que resultarão na criação de um ficheiro com as instruções assembly correspondentes, ordenadas corretamente por forma a produzirem o resultado final devido pela máquina de stack virtual.

2.2 Especificação do Requisitos

Capítulo 3

Concepção/desenho da Resolução

3.1 Estruturas de Dados

3.2 Algoritmos Implementados

Capítulo 4

Codificação

4.1 Alternativas, Decisões e Problemas de Implementação

4.2 Testes realizados e Resultados

Capítulo 5

Conclusão

O grupo concluiu que o resultado final obtido é bom e cujos problemas encontrados foram todos ultrapassados. No que diz respeito á gramática desenvolvida pode-se concluir que cumpre todos os requisitos, incluindo o requisito opcional e ainda que a sua expansão será fácil em muitos aspectos, como por exemplo, a evolução para outros tipos, que não inteiros(float,doubles,etc).

Em relação ao compilador produzido e com base nos testes mínimos que foram desenvolvidos, este constrói efetivamente os ficheiros com as instruções assembly na ordem correta que permite ser interpretado pela máquina de stack virtual e ainda obter o resultado esperado na mesma.

Optou-se ainda por desenvolver outros testes que demonstram outras situações comuns em que o compilador também funciona.

Apêndice A

Código do Programa

Bibliografia

- [1] V. Aho, Alfred , S. Lam, Monica , Sethi, Ravi and D. Ullman, Jeffrey Second Edition, *Compilers Principles Techniques and Tools*.