Linguagem i–

Hugo Frade*, Miguel Costa†, and Milton Nunes‡

Análise e Transformação de Software,
UCE30 Análise e Concepção de Software,
Mestrado em Engenharia Informatica,
Universidade do Minho

18 de Fevereiro de 2013

Resumo

Este documento apresenta a resolução do Trabalho Prático de Análise e Transformação de Software em que se definiu a linguagem i–, usando o AnTLR criou-se um compilador de forma a gerar código MSP.

*Email: hugoecfrade@gmail.com
†Email: miguelpintodacosta@gmail.com
‡Email: milton.nunes52@gmail.com

Conteúdo

1 Introdução **3**

2 Ambiente de Trabalho **3**

3 Descrição do problema **3**

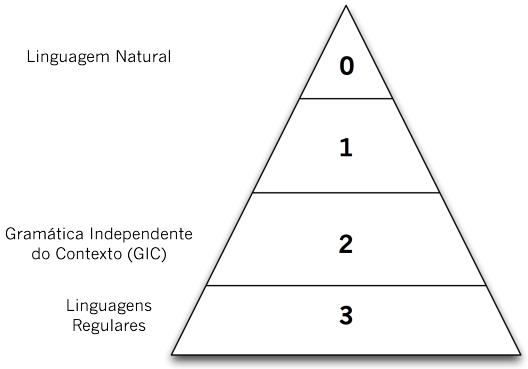
4 MSP **4**
4.1 Implementação 4

5 Conclusões **4**

6 Anexos **5**

1 Introdução

Tal como em maior parte das coisas no nosso dia à dia, as linguagens possuem uma hierarquia. No topo (0), encontra-se a linguagem natural, a mais difícil de decifrar devido à maior diversidade de termos e expressões que pudemos usar. Na base (3) encontram-se as linguagem regulares, que possuem um número muito limitado de termos, e por isso são bastante fáceis de descodificar e perceber.



Formalmente uma gramática independente do contexto é definida como uma gramática formal[1] por regras de produção da formalmente definidas como:`X -> x`, onde X é um símbolo não terminal e x é uma sequência de não terminais, ou até mesmo o vazio.

Depois de definida a gramática precisamos de um parser para o identificar. Como tal foi utilizado o AnTLR para criar esse parser. O AnTLR é uma ferramenta de reconhecimento de linguagem. Este aceita como input uma gramática que especifica a linguagem e gera o código fonte para o reconhecimento da linguagem. O AnTLR utiliza o algoritmo LL(*), algoritmo classificado como top-down.

Quando se escrever um programa, o objetivo é que ele faça alguma tarefa automaticamente, para tentar simular essa execução, nesta fase usamos uma máquina virutal com o nome MSP (Mais Simples Possível)

2 Ambiente de Trabalho

Foi necessário usar um Gerador de Compiladores para gerar o nosso próprio compilador, por isso usamos o AnTLR que é também usado nas aulas. Para facilitar o processo de debugging durante a resolução do problema, usamos a ferramenta AnTLRWorks, que tem uma interface bastante agradável e simpática para ajudar a resolver problemas desta natureza.

Visto que era necessário gerar código para ser executado pelo MSP, foi utilizada a máquina virtual que o professor disponibilizou para testes.

3 Descrição do problema

O que é pretendido é usar a gramática criada nos trabalhos anteriores da disciplina e adaptar para gerar código para ser lido pela MSP.

Das várias formas possíveis de fazer isso, a utilizada foi adicionar instruções às produções da gramática para dar como output o código pretendido.

[1] Objecto matemático que permite criação de linguagens através de um conjunto de regras de formação.

4 MSP

Explicar o que é a MSP

4.1 Implementação

Incluir a geração de código MSP na nossa linguagem foi apenas fazer instruções nas produções para imprimir no output.

5 Conclusões

A resolução deste exercício permitiu perceber melhor a forma como as linguagens podem ser úteis para gerar um programa, que dependendo do input que irá receber, o resultado final seja o esperado sem ter de estar a alterar o código do programa que é automaticamente gerado. Apesar de não termos qualquer tipo de output, as árvores geradas permitiram chegar a estas conclusões.

Umas das dificuldades foi perceber como o AnTLR fazia o parser das frases de forma a não haver ambiguidade nas produções.

6 Anexos