TRABALHO DE LFT – PORTUGOL.GRAMMAR

Alunos:

>> **Anderson dos Santos Nunes [201010006940]**

>> **Fábio Nascimento Santos [201110006090]**

Professores Responsáveis:

>> **Breno Piva Ribeiro**

>> **Kalil Araújo Bispo**

Ferramentas utilizadas:

>> **IDE: Eclipse**

>> **SableCC, versão 3.7**

1- UMA BREVE VISÃO DO QUE FOI O PROJETO:

Nosso projeto obteve um resultado muito bom, não foi totalmente o esperado pois gostaríamos de ter tido mais tempo para a parte Semântica da linguagem, mas de uma maneira geral, fizemos a parte Léxica, a Sintática e a Sintática Abstrata e isso nos ajudou a aumentar e assimilar o conceito visto em aula ao nosso conhecimento sobre compiladores, e os passos para a compilação.

Ao começarmos a Léxica construímos todos os Tokens com base na especificação versão 4, e chegamos a necessidade de criar os Helpers para ajudar a concluir esta fase, e além disso percebemos que para que comentários aninhados era necessário aplicar um novo conhecimento chamado State (Estado), que faz como um autômato com pilha e consegue contar e saber se está balanceado!

Quando fomos fazer a Sintática, percebemos a dificuldade que o SableCC tem em construir a linguagem, mesmo a gramática estando em LALR(1) (ao qual ele reconhece), geralmente dando erro Shift/Reduz, onde ele deveria preferir por Shift e não dar conflito. Então foi-se necessário criar uma lógica que não alterasse a gramática, e tão pouco as precedências dentre os operadores, para que se conseguisse concluir esta fase!

Ao fazer a Árvore Sintática Abstrata, novamente tivemos que alterar a Sintática Concreta, para criar uma árvore onde abstraísse as transformações intermediárias, ou pode-se

dizer auxiliares, que serão realmente utilizada na semântica aplicada a esta árvore abstrata, e assim, sendo um tipo de otimização da gramática.

O trabalho foi bem gratificante, principalmente quando conseguíamos colocar em prática todo o conhecimento obtido através das leituras do livro do Dragão e o Livro de Teoria da Computação! Apesar do SableCC dar alguns erros por besteira, ele ajuda muito, pois nos mostra em que linha está dando o erro. Assim, mesmo que o erro não seja propriamente naquela linha, é possível identificá-lo de maneira mais rápida e corrigi-lo de maneira precisa.

2 - ORGANIZAÇÃO DOS ARQUIVOS (READ ME)

A organização foi baseada na própria construção de pacotes e classes que o SableCC constrói e prontamente com a criação de alguns pacotes e classes auxiliares que se faz necessário para que o teste funcione.

Sendo assim, os pacotes (package) que o SableCC gera são: ***analysis***, ***lexer***, ***node***, ***parser***. E todos, inclusive os criados por nós para teste, estão no pacote geral chamado ***portugol***, onde definimos na escrita da gramática.

Além destes foi-se criado o pacote ***doc_e_aux*** que possui esta documentação e um roteiro que o site http://sablecc.sourceforge.net/thesis/thesis.html específica para poder usar os States (estados) criado na parte léxica da gramática, para poder ter um comentário aninhado; e neste pacote há também uma classe chamada ***New_Lexer.java*** que estende a classe ***Lexer.java*** para poder testar a parte léxica da gramática.

Foi criado também o pacote ***main_test*** que possui as classes para poder-se testar as nossas fases, ***Compilador.java***, ***GerarGramatica.java***, ***Test_lexer.java*** e ***Test_parser.java***, além de um arquivo em texto (Teste_programa.txt) que é utilizado como um programa exemplo.

3 – GERAR GRAMÁTICA

Além do comando base na prompt de comandos, "java –jar sablecc.jar portugol.grammar", fizemos um teste que já gera a gramática e após gerar pode-se fazer os testes e o caso em que a gramática já exista, pode-se ir diretamente para o caso de testa-la.

4 – CONCLUSÕES

O trabalho foi uma boa forma de didática para assimilar os conceitos vistos em sala pelos professores citados. Apesar da especificação possuir alguns erros, e por isso alterada até a versão 4, e com isso também tivemos que alterar por quatro vezes alguns códigos, achamos muito bom o processo de compilação, algumas dúvidas ainda continuam, pois não obtivemos

tempo necessário para a conclusão do projeto, porém pensamos em continuar futuramente, pois foi interessante ver cada fase funcionando da forma que esperávamos.

Desde já agradecemos a honra de poder ter um trabalho que nos foi dado como ajuda para assim entender como se dá a Compilação e como a Teoria da Computação surgiu, e foi adquirindo a forma que está atualmente.