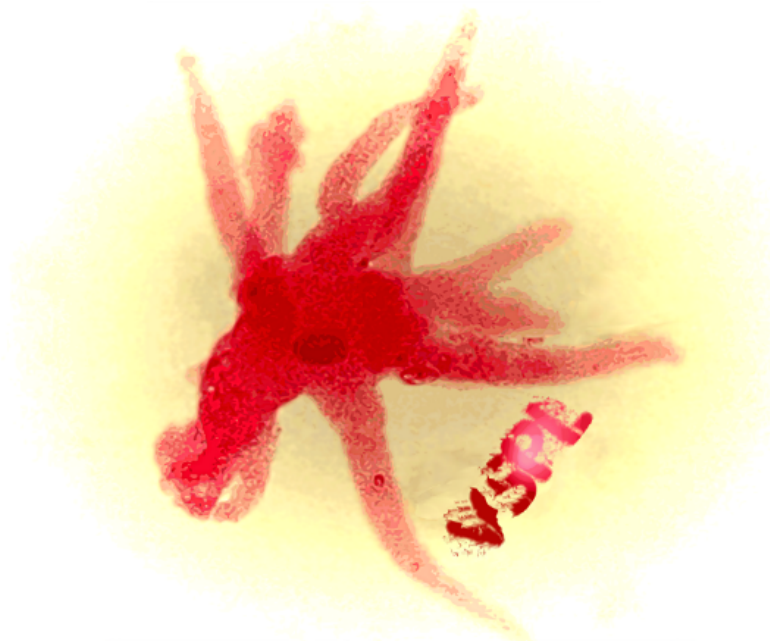




UNIVERSIDADE DE ÉVORA

DISCIPLINA DE COMPILADORES

Compilador de VSPL



Autores:

Ricardo FUSCO, 29263

Marcus SANTOS, 29764

Professores:

Salvador ABREU

Pedro PATINHO

July 7, 2014

Índice

| | | |
|----------|------------------------------|----------|
| 1 | Introdução | 3 |
| 2 | Desenvolvimento | 4 |
| 2.1 | Elementos Lexicais | 4 |
| 2.2 | Análise lexical | 4 |
| 2.3 | Análise sintáctica | 4 |
| 2.4 | Análise semântica | 6 |
| 3 | Conclusão | 7 |

Lista de Figuras

| | | |
|---|---|---|
| 1 | Parte da Árvore de sintaxe abstrata | 5 |
|---|---|---|

1 Introdução

No âmbito da unidade curricular de Compiladores pretende-se, através das ferramentas e métodos abordados ao longo das aulas de Compiladores implementar um compilador de VSPL composto pelo analisador lexical(Flex), sintáctico (Bison) e semântico, e um gerador de código SiM. Este programa irá receber como input um ficheiro com código VSPL e gerar o código correspondente em SiM.

2 Desenvolvimento

A linguagem VSPL foi a linguagem escolhida pelo professor para a implementação deste trabalho prático por ser uma linguagem simples de perceber e com uma gramática relativamente simples.

2.1 Elementos Lexicais

- Identificadores - São os identificadores habituais compostos por letras maiúsculas e minúsculas visto que o VSPL é sensível às diferenças entre estas (case sensitive).
- Palavras reservadas - Estas palavras reservadas são as palavras cuja definição é a própria palavra como, por exemplo, as operações AND, OR, NOT, o WHILE, IF, ELSE, entre outras.
- Comentários - O comentário começa com o caracter '#' e termina no fim da linha onde ocorre.

- Literais - São 3 os tipos de constantes (literals) reconhecidos:
Constantes Float (virgula flutuante) - REAL_LIT

Constantes Int (inteiras) - INT_LIT

constantes Bool (booleanas) - BOOL_LIT

Também podem ser considerados como literais:

literals de tipos compostos como tuplos, arrays
ou literais funcionais

2.2 Análise lexical

O analisador lexical encontra-se completamente feito identificando desta maneira todos os elementos lexicais próprios da linguagem.

2.3 Análise sintáctica

O analisador sintáctico foi praticamente todo construído tendo por base os fundamentos teóricos presentes nas notas acerca da sintaxe da linguagem VSPL providenciadas pelo professor da disciplina. De todos os tipos presentes nesta linguagem apenas não foram implementados devidamente as classes, os arrays e primary. Tirando estes tipos que faltam o analisador garante que todas as regras de sintaxe sejam cumpridas devidamente e que todos os tipos sejam propriamente identificados. Na próxima figura pode ser observada parte da árvore de sintaxe abstracta para que se tenha uma ideia da maneira como estão feitas as produções.

2.4 Análise semântica

Na parte da análise semântica foram feitas todas as structs de cada produção e as funções necessárias à inicialização de cada nó da árvore estando o programa preparado para fazer a análise de nomes e a inferência de tipos. Apenas faltam as funções necessárias para percorrer os nós da árvore e a tabela de símbolos para que possa ser feita a inferência de tipos.

3 Conclusão

Acabado este trabalho podemos concluir que conseguimos cumprir a maior parte dos objectivos chave deste trabalho que foram conseguir ter um compilador de VSPL minimamente funcional que fizesse a análise lexical e sintáctica correctamente reconhecendo quase na totalidade os elementos lexicais e sintácticos característicos desta linguagem de programação. No que diz respeito à análise semântica conseguimos ter todas as estruturas e funções necessárias para criar e inicializar os nós da árvore, apenas falta a criação de uma tabela de símbolos e as funções necessária para percorrer a árvore e realizar a análise de nomes e inferência de tipos correctamente de maneira a que o programa esteja preparado para gerar código. Não conseguimos também realizar o gerador de código SiM. Mas além destes percalços achamos que conseguimos fazer e compreender os básicos deste trabalho e aplicámos os conhecimentos que conseguimos adquirir ao longo nesta disciplina.