



UNIVERSIDADE DE SÃO PAULO - ICMC

Departamento de Ciências de Computação

SCC 605 - Teoria da Computação e Compiladores - 1º Sem /2010

PROFa: Sandra Aluisio

WIKI: <http://wiki.icmc.usp.br/index.php/SCC-605>

Trabalho Prático 2 -- Entrega: 24/5 (PARTE 1) na sala de aula e 8/6 (PARTE 2)

Parte 1:

a) Estudar o compilador JavaCC, disponível para download no site <http://javacc.dev.java.net> para fazer o trabalho abaixo. Devem responder a questão de como o JavaCC trata a ambigüidade natural da análise léxica que deve distinguir IDs de palavras reservadas e reais de inteiros, além de símbolos da linguagem que possuem o mesmo prefixo como ">" e ">=" e ":" e "!=":

Entregar o arquivo de especificação **javacc** para o Analisador Léxico do Pascal Simplificado com as extensões do grupo; os arquivos de teste e as saídas.

b) Especificar:

- a. Como são os comentários
- b. Se as palavras reservadas e os identificadores são sensíveis à caixa como na linguagem C ou insensíveis como em Pascal
- c. O formato das constantes, para quem tem extensão relacionada a constantes char, real e strings

ROTEIRO:

- 1) Fazer um arquivo de especificação para o compilador Javacc (por exemplo, **exparser.jj**)
- 2) Compilar o tal arquivo para termos o analisador léxico executável (ALE)

Para compilar este arquivo use "**javacc**" e depois "**javac**":

```
$ javacc exparser.jj
$ javac *.java
```

Para executar o programa:

```
$ java ExampleParser file
```

- 3) Testar o ALE com programas exemplos de programas válidos e não válidos, com as extensões do grupo.

Um Analisador Léxico faz a interface entre o arquivo que contem o programa e o resto do compilador e tem como funcionalidades:

- separar cada elemento do VT da linguagem (token) e devolver um código apropriado;
- eliminar comentários, brancos, \n e tabs;

- reconhecer erros léxicos (ex. símbolo não pertencente ao V_t , constantes mal formadas, comentário aberto e não fechado).

Entregar um relatório IMPRESSO:

- **dizendo como usou o Javacc, isto é, com quais opções e para que.**
- **um arquivo de especificação para o compiler compiler Javacc e seus arquivos de testes;**
- **a saída do javacc para os programas de teste acima.**

Parte 2: Especificação da Análise Léxica JUNTO com a análise Sintática com as extensões do grupo, usando o gerador de compilador JavaCC (num único arquivo).

Façam testes com vários programas corretos e com problemas sintáticos. Façam um relatório explicativo das decisões de projeto tomadas acima, na parte 1, quanto à especificação léxica e como decidiram a impressão dos erros.

ROTEIRO:

1. Colocar a especificação para o Analisador Léxico do Pascal Simplificado com as extensões do grupo na **parte do arquivo de especificação** usado pelo JavaCC dedicada ao reconhecimento dos tokens.
2. Especificar **a parte da análise sintática no arquivo de entrada** para ser usado pelo JavaCC. JavaCC admite uma descrição da gramática em EBNF, assim a gramática está pronta para uso. Porém esta deve ser LL(1) para ser eficiente, embora o JavaCC trabalhe com uma gramática LL(k). Vejam se há pontos que precisam de reescrita.
3. Customizem as mensagens de erros léxicos e sintáticos. JavaCC gera dois arquivos para gerenciar os erros léxicos e sintáticos:
 - TokenMgrError.java
 - ParseException.java

e testem o parser gerado com vários programas corretos e com erros. Estes devem ter as extensões do grupo. No nível sintático, quando algum **erro** for encontrado, uma mensagem de erro será retornada ao usuário. A análise sintática deverá parar. No final da Análise se houve sucesso imprimir mensagem.

Forma de Entrega:

- **Arquivo de especificação do JavaCC, arquivos de teste e arquivo de saída da análise sintática: impressos e também no CD.**
- **Código executável e instruções de como rodar: CD.**