Construção de Compiladores
Daniel Lucrédio
Especificação e Critérios de notas do Trabalho 2
(Última revisão: dez/2021)

O trabalho 2 (T2) da disciplina consiste em implementar um analisador sintático para a linguagem LA (Linguagem Algorítmica) desenvolvida pelo prof. Jander, no âmbito do DC/UFSCar. O analisador sintático deve ler um programa-fonte e apontar onde existe erro sintático, indicando a linha e o lexema que causou a detecção do erro. Exemplo:

Entrada:

```
{ leitura de nome e idade com escrita de mensagem usando estes dados }


        declare
                nome: literal
        declare
                idade: inteiro

        { leitura de nome e idade do teclado }
        leia(nome)
        leia(idade)

        { saída da mensagem na tela }
        escreva(nome, " tem ", idade, " anos.")

fim_algoritmo
```

Saída produzida:

```
Linha 10: erro sintatico proximo a leia
Fim da compilacao
```

Os erros léxicos detectados no T1 devem continuar sendo detectados corretamente.

A gramática da linguagem LA será fornecida, portanto a princípio basta digitá-la em um gerador de parsers (recomenda-se o ANTLR) e deve funcionar corretamente. Cuidado ao digitar as regras, para que não sejam inseridos erros que irão prejudicar trabalhos futuros. Aproveite o trabalho para compreender a estrutura sintática da linguagem.

O analisador deve poder ser executado em linha de comando (windows, mac ou linux), com DOIS ARGUMENTOS OBRIGATORIAMENTE:

Argumento 1: arquivo de entrada (caminho completo)
Argumento 2: arquivo de saída (caminho completo)

Exemplo de como seu analisador deve rodar:

```
c:\java -jar c:\compilador\meu-compilador.jar c:\casos-de-teste\arquivo1.txt
c:\temp\saida.txt
```

Como resultado, seu compilador deve ler a entrada de `c:\casos-de-teste\arquivo1.txt` e salvar a saída no arquivo `c:\temp\saida.txt`

NÃO SERÃO ACEITOS programas que imprimem a saída no terminal. É obrigatório salvar no arquivo.

Critérios de avaliação do Trabalho 2 e descontos nas notas

O trabalho 2 deve ser desenvolvido em **grupos de até 3 estudantes** (máximo). A nota do trabalho 2 será composta de 3 parcelas, cada uma valendo de 0 a 10, e com os pesos assim distribuídos:

DE - Documentação externa: 10%
DI - Documentação interna: 10%
CT2 - Casos de teste sintático: 80%

DE - Documentação externa: deve ser fornecido um arquivo de ajuda, para possibilitar que qualquer pessoa consiga compilar e executar seu trabalho. Deve incluir programas que precisam ser instalados, suas respectivas versões, configurações necessárias, e os passos de execução.

Exemplos de erros comuns na documentação externa e que causarão desconto na nota:

- Só diz como executar, mas não como compilar o programa, ou vice-versa
- Nada foi dito sobre como compilar/interpretar o programa e executá-lo
- Ausência da documentação externa

DI - Documentação interna: o código-fonte (gramática + demais arquivos) devem ser documentados a ponto de possibilitar seu entendimento por parte de outros programadores olhando seu código. Insira comentários explicativos em todos os pontos relevantes do seu código. Nomes de variáveis e funções também fazem parte da documentação interna.

Exemplos de erros comuns na documentação interna e que causarão desconto na nota:

- Pouco documentado (sem explicação do propósito das regras léxicas, sintáticas, semânticas e geração de código, entrada, saída, descrição das variáveis, etc.)
- Descuido nos comentários ou nomes de funções/variáveis pouco indicativos
- Ausência de comentários sobre o processo de compilação
- Nenhuma linha de documentação relevante

CT2 - Casos de teste com erros léxicos + sintáticos: o compilador deve apontar corretamente todos os erros léxicos e sintáticos, da forma exata como nos casos de teste. São **62 casos de teste** nesta categoria.