

Compiladores

CIÊNCIA DA COMPUTAÇÃO

Prof. Dr. João Paulo Aramuni

Sumário

* **Atividade Autoinstrucional**

Atividade Autoinstrucional

- * **Elaboração de um Analisador Léxico Simples**

- * **Objetivo:**

- * Ao final deste trabalho os alunos terão tido a oportunidade de elaborar um analisador léxico simples, mas que tenha sido capaz de apresentar as particularidades inerentes ao processo de compilação.

Atividade Autoinstrucional

- * **Elaboração de um Analisador Léxico Simples**

- * **Proposta:**

- * Este trabalho tem como meta a prática em análise léxica.
- * O trabalho deverá ser feito em grupo de até 4 alunos.
- * Os grupos deverão considerar os caracteres usados na linguagem de programação C.

Atividade Autoinstrucional

* Elaboração de um Analisador Léxico Simples

* Especificação:

- * Faça uma **gramática regular** para uma linguagem considerando como base as mesmas condições estabelecidas para o Scanner4 e que atenda as especificações dos itens abaixo:
 - * 1) Deverão ser aceitos comentários da forma “/* . . . */” que poderão ser incluídos onde existir um branco, ou seja, entre quaisquer itens léxicos da expressão.
 - * 2) A expressão poderá começar e terminar por espaços (brancos) ou por um comentário da forma “/* . . . */”.
 - * 3) Portanto, a expressão poderá:
 - * Começar por letras, por espaços (brancos) ou por um comentário da forma /* . . . */;
 - * Terminar por letras, por algarismos, por espaços (brancos) ou por um comentário da forma /* . . . */.

Atividade Autoinstrucional

* Elaboração de um Analisador Léxico Simples

* Especificação:

- * Faça uma **gramática regular** para uma linguagem considerando como base as mesmas condições estabelecidas para o Scanner4 e que atenda as especificações dos itens abaixo:
 - * 4) Considerar comentários // em qualquer lugar da expressão.
 - * 5) Verificar se as variáveis têm nome de palavras reservadas. Exemplo: for, while, if, else, int, float, string, etc. Caso tiverem, exiba um erro na linha.
 - * 6) Exibir o número do último estado visitado no autômato antes do erro ocorrer. Veja o exemplo no arquivo Scanner5.cpp.
 - * 7) Todas as sentenças devem terminar com ; (ponto e vírgula).
 - * 8) Os espaços em branco não podem ser removidos do arquivo de entrada. Eles devem ser lidos como qualquer outro caractere.

Atividade Autoinstrucional

- * **Elaboração de um Analisador Léxico Simples**

- * **Especificação:**

- * Faça uma **gramática regular** para uma linguagem considerando como base as mesmas condições estabelecidas para o Scanner4 e que atenda as especificações dos itens abaixo:
 - * 9) Cada registro do arquivo de entrada, “Testes_ScannerAAI.txt”, (cada linha) será considerado uma expressão e uma expressão será colocada totalmente em uma única linha: a expressão não poderá continuar na linha seguinte.
 - * 10) O programa deverá gravar um arquivo de saída, “saida.txt”, contendo todas as expressões, identificando as que estão corretas e as incorretas.
- * Consulte no material didático a Gramatica_Scanner4.txt e use-a como material de apoio.

Atividade Autoinstrucional

- * **Elaboração de um Analisador Léxico Simples**

- * **Especificação:**

- * Desenhe o **autômato finito** correspondente à gramática regular.
 - * Consulte no material didático os autômatos: Automato_Scanner3.png e Automato_Scanner5.png e use-os como material de apoio.
 - * Preferencialmente, o **número de estados** do autômato deve ser **igual ao número de não-terminais** na gramática.
 - * Exemplo: A gramática Gramatica_Scanner3.txt tem 12 não-terminais (de 0 a 11), e o Automato_Scanner3.png possui 12 estados (de 0 a 11).
 - * Outro exemplo: A gramática Gramatica_Scanner5.txt tem 6 não-terminais (de 0 a 5), e o Automato_Scanner5.png possui 6 estados (de 0 a 5).

Atividade Autoinstrucional

- * **Elaboração de um Analisador Léxico Simples**

- * **Especificação:**

- * Escreva o **programa** do analisador léxico correspondente, codificado preferencialmente em C, com o nome de ScannerAAI.
- * Consulte no material didático o programa Scanner4.cpp e use-o como material de apoio.
- * Preferencialmente, o **número de ‘cases’** do comando `switch` deve ser **igual** ao **número de não-terminais** na gramática (e consequentemente igual ao número de estados do autômato).
 - * Exemplo: O programa Scanner3.cpp tem 12 cases (de 0 a 11), e a Gramatica_Scanner4.txt possui 12 não-terminais (de 0 a 11).
 - * Outro exemplo: O programa Scanner5.cpp tem 6 cases (de 0 a 5), e a Gramatica_Scanner5.txt possui 6 não-terminais (de 0 a 5).

Atividade Autoinstrucional

- * **Elaboração de um Analisador Léxico Simples**
- * **O que deve ser entregue:**
 - * Arquivo zip entregue via SINEF contendo:
 - * A **gramática** regular (Arquivo com extensão .txt);
 - * O **autômato** finito (Arquivo com extensão .png);
 - * O **código** do “ScannerAAI” (Arquivo com extensão .c, .cpp, .java, etc). Envie também o executável para testes.
- * **Todos do grupo devem entregar!**

Atividade Autoinstrucional

- * **Elaboração de um Analisador Léxico Simples**
- * **O que deve ser entregue:**
 - * Entregue também:
 - * A massa de testes (arquivo de entrada) utilizada.
 - * Exemplo: Testes_Scanner5.txt.
 - * O arquivo executável do programa fonte.
 - * Exemplo: Scanner5.exe.

Atividade Autoinstrucional

- * **Elaboração de um Analisador Léxico Simples**

- * **Distribuição de pontos:**

- * Gramática Regular: 2,5 pts
- * Autômato Finito: 2,5 pts
- * Código do “ScannerAAI”: 2,5 pts
- * Apresentação do trabalho: 2,5 pts

- * Total: 10 pts.

- * Não serão aceitas cópias de trabalho.



Atividade Autoinstrucional

* **Elaboração de um Analisador Léxico Simples**

* **Observações:**

- * Coloque no início do programa fonte um comentário com os nomes dos componentes do grupo.
- * A imagem do autômato finito, bem como o arquivo de texto da gramática regular, também deverão conter os nomes dos autores do trabalho.
- * A implementação poderá ser feita nas linguagens C, C++ ou Java e deverá ser baseada no diagrama do autômato finito correspondente. Deve ficar claro a relação entre código, autômato e gramática.
- * A apresentação do scanner será feita através de defesa em laboratório, na data previamente marcada, em horário de aula. Todos do grupo devem participar, de forma ativa, da apresentação.

Atividade Autoinstrucional

* **Elaboração de um Analisador Léxico Simples**

* **Observações:**

- * A defesa do trabalho consistirá em apresentação, explicação, execução do programa (ao vivo, sem gravações), e questionamentos do scanner por parte do professor e da turma (após a apresentação).
- * Critérios de avaliação: Os trabalhos serão avaliados de acordo com os seguintes parâmetros:
 - * Funcionamento: O programa deverá estar isento de erros e executar corretamente as massas de testes. Além disso, deverá corresponder integralmente às condições definidas no autômato finito.
 - * Organização do código: O código fonte deverá estar estruturado de maneira adequada e devidamente comentado. (Boas práticas de programação).
 - * Interface: O scanner deverá oferecer boa usabilidade e produzir mensagens de tratamento de erros de fácil entendimento pelo usuário.

Atividade Autoinstrucional

- * **Elaboração de um Analisador Léxico Simples**

- * **Apresentação:**

- * O grupo deverá se preparar para a apresentação:
 - * Explicar o código fonte, a gramática regular e o autômato finito;
 - * Relacionar o código fonte com a gramática e com o autômato;
 - * Apontar os problemas encontrados durante o trabalho;
 - * Executar o ScannerAAI, inserindo a massa de testes e tendo como resposta o arquivo de saída contendo todas as expressões de entrada, identificando as que estão corretas e as incorretas;
 - * Tentar trazer algo inovador, que enriqueça o trabalho.

Obrigado.

joapauloaramuni@gmail.com
joapauloaramuni@fumec.br