

SCC0605 Teoria da Computação e Compiladores

Prof. Thiago A. S. Pardo

Especificação do Trabalho 1: análise léxica

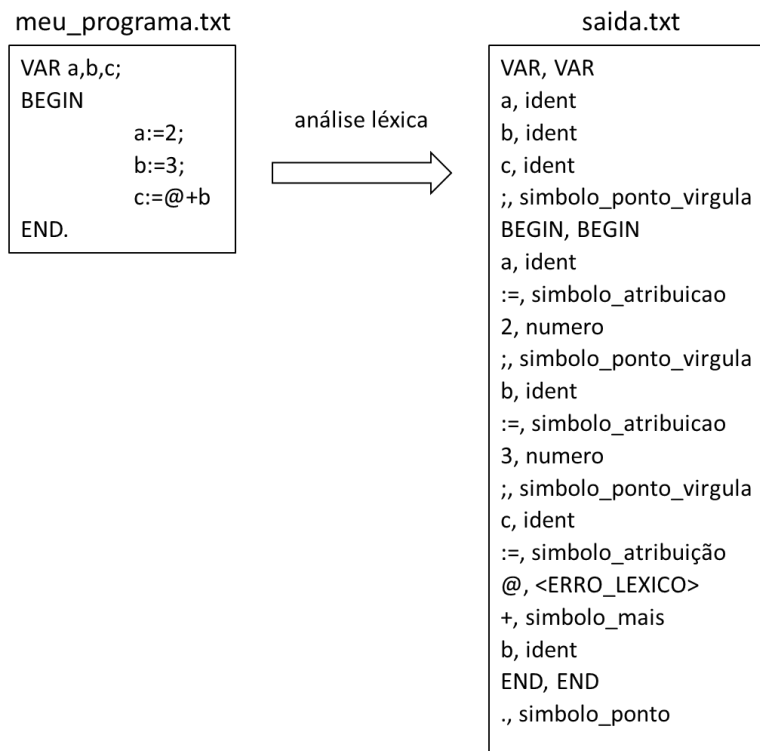
Com base na gramática da linguagem PL/0, disponível no e-Disciplinas, desenvolver o analisador léxico para esta linguagem.

O trabalho pode ser feito em grupos de até 5 pessoas. Vocês deverão produzir:

1. os autômatos (se desejarem, podem usar a ferramenta JFlap ou outra que preferirem para desenhar os autômatos);
2. o código-fonte correspondente aos autômatos projetados (na linguagem de programação C).

Ao fim, deverá ser entregue um arquivo compactado com: (i) os códigos-fonte de seu analisador léxico; e (ii) um relatório (sugere-se até 10 páginas), que contenha a identificação dos membros do grupo, um breve relato das decisões de projeto, os autômatos projetados (pode ser cópia e cola das imagens dos autômatos), instruções passo a passo para compilar/rodar seu código-fonte (que sistema usar, versão, parâmetros esperados, etc.) e pelo menos um exemplo de execução. O trabalho também deverá ser apresentado brevemente ao docente.

Espera-se que seu analisador léxico aceite um arquivo txt com o programa escrito em PL/0 e produza um outro arquivo txt com a saída, com um par token-classe por linha (indicando os erros léxicos, se houver), conforme ilustração abaixo.



Sugere-se que seja implementada uma função principal cujo único propósito seja chamar o procedimento do analisador léxico tantas vezes quanto necessário para processar o programa inteiro do usuário. Assim, a cada chamada, o analisador léxico devolve um par token-classe, que é impresso pela função principal. O propósito dessa função principal é simular o funcionamento do analisador sintático (que, no próximo trabalho prático, será incorporado ao seu programa e substituirá essa função principal).

Será necessário tomar várias decisões de projeto, por exemplo, como implementar os autômatos, como controlar os símbolos já lidos (veja no material de aula a estratégia de “retroceder” nos autômatos e de usar um símbolo “lookahead”), como tratar erros, quais são as classes associados aos tokens (sugere-se usar strings que indiquem claramente as classes), etc. Nesse ponto, sugere-se que se use a experiência adquirida em sala de aula, empregue seu bom senso e faça o exercício de se colocar no lugar do possível usuário de seu sistema, respondendo a perguntas como “esse sistema está simples e fácil de usar?” e “a saída dele é legível?”.

O trabalho será principalmente avaliado em função da correção da execução (ou seja, o programa deve fazer o que se espera que ele faça em cada caso), da usabilidade da interface com o usuário (mesmo que seja via linha de comando), da qualidade da implementação (código-fonte, modularidade, documentação interna, etc.) e da documentação externa (relatório apresentado).

A entrega do arquivo compactado deve ser realizada no e-Disciplinas até o dia 17 de maio (sexta-feira). Basta que um dos membros do grupo faça a submissão. Em seguida, o grupo deve contatar o docente para agendar um horário para apresentar o trabalho.

Atenção:

- **a cada dia de atraso, será descontado 1 ponto do trabalho;**
- **se plágio for detectado, todos os grupos envolvidos (quem plagiou e quem compartilhou seu trabalho) terão suas notas zeradas e o caso será encaminhado para a coordenação do curso para as devidas providências (como avaliação da possibilidade de reprovação automática na disciplina e encaminhamento para abertura de sindicância interna). Recomenda-se a leitura do documento a seguir: <http://www.leginf.usp.br/?resolucao=resolucao-no-4871-de-22-de-outubro-de-2001#t5>**