

Primeira Prova de MAB 471 — Compiladores I

Fabio Mascarenhas

25 de Abril de 2012

A prova é individual e sem consulta. Responda as questões na folha de respostas, a lápis ou a caneta. Se tiver qualquer dúvida consulte o professor.

Nome: _____

DRE: _____

Questão:	1	2	3	4	Total
Pontos:	1/2	2	2	5 1/2	10
Nota:					

- (1/2 ponto) Qual o problema causado por aninhamento de comentários em um analisador léxico gerado a partir de expressões regulares?
- (2 pontos) O seguinte trecho de código Java é responsável por ler e ignorar comentários no analisador léxico de TINY escrito à mão:

```
case '{':
    nextChar();
    while((char)lookAhead != '}' &&
          lookAhead != -1)
        nextChar();
    if((char)lookAhead != '}')
        throw new Error("comentário não terminado");
    nextChar();
    continue;
```

Como você reescreveria esse trecho para permitir comentários aninhados? Você pode definir métodos auxiliares. *Dica: pense em um analisador sintático recursivo.*

- Considere a seguinte gramática:

$$\begin{aligned} S &\rightarrow aS \mid Ab \\ A &\rightarrow XYZ \mid \varepsilon \\ X &\rightarrow cS \mid \varepsilon \\ Y &\rightarrow dS \mid \varepsilon \\ Z &\rightarrow eS \end{aligned}$$

- (a) ($\frac{1}{2}$ ponto) Dê a derivação mais à esquerda para a cadeia `aebb`.
- (b) ($1\frac{1}{2}$ pontos) Explique por que ao adicionar a produção $X \rightarrow bS$ a gramática deixa de ser LL(1).
4. Na gramática a seguir os não-terminais estão em maiúsculas e os terminais em minúsculas, todos separados por espaços (estilo BNF):

```
PROGRAMA -> begin LISTACMD end
LISTACMD -> LISTACMD CMD
          | CMD
CMD       -> do id := num to num begin LISTACMD end
          | read id
          | write EXP
          | id := EXP
EXP       -> EXP + EXP | EXP - EXP | num | id | ( EXP )
```

- (a) ($\frac{1}{2}$ ponto) Mostre a árvore sintática para o programa a seguir (pode abreviar os não-terminais assim: P, L, C, E, T):

```
begin
  read x
  do i := 1 to 100 begin
    x := x + i
  end
  write x
end
```

- (b) ($\frac{1}{2}$ ponto) Qual a principal razão para essa gramática não ser LL(1)?
- (c) (1 ponto) Mostre que essa gramática é ambígua.
- (d) (1 ponto) Remova a ambiguidade dessa gramática.
- (e) ($2\frac{1}{2}$ pontos) Reescreva a gramática para ser LL(1). Dê os conjuntos de lookahead (FIRST+) para cada regra que tenha alternativas.