

Primeira Prova de MAB 471 2015.1 — Compiladores I

Fabio Mascarenhas

12 de maio de 2015

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	Total
Pontos:	4	4	2	10
Nota:				

- Muitas linguagens possuem regras bem intrincadas sobre quais caracteres e códigos de escape são permitidos dentro de cadeias de caracteres literais. Suponha que cadeias de caracteres literais começam e terminam com aspas duplas, e podem conter qualquer caractere exceto aspas duplas e a barra invertida, a não ser que a barra invertida seja o início de um código de escape. Os códigos de escape podem ser `\\`, `\n`, `\"`, `\ddd` (barra invertida seguida por exatamente três dígitos de 0 a 9), e `\u{hexa}` (barra invertida seguida de u e um ou mais dígitos hexadecimais entre chaves).
 - (2 pontos) Escreva uma expressão regular para cadeias de caracteres literais.
 - (2 pontos) Escreva um autômato finito determinístico para cadeias de caracteres literais.
- A gramática a seguir é um fragmento da gramática da linguagem MiniJava, no formato EBNF:

```
CMD -> '{' {CMD} '}'  
      | if '(' EXP ')' CMD [else CMD]  
      | id = EXP ;  
      | id '[' EXP ']' = EXP ;
```

 - (2 pontos) Escreva um analisador recursivo *sem retrocesso* para esse fragmento. Não é necessário construir a árvore sintática. Assuma que o token de lookahead está no campo `la`, e que o método `terminal(tipo)` avança para o próximo token se o look-ahead bate com `tipo`, ou dá erro.
 - (2 pontos) Mostre que essa gramática é ambígua.
- (2 pontos) Escreva um analisador sintático recursivo *com retrocesso local* para a gramática $A \rightarrow aAa \mid \varepsilon$. Não é necessário construir a árvore sintática. Assuma que a posição atual na entrada fica guardada no campo `atual`, e que o método `terminal(tipo)` avança para o próximo token se o atual bate com `tipo`, ou lança uma exceção `Falha`.

BOA SORTE!