

## Prova de Compilação – 30/09/2017

Questão 1 – B

Questão 2 – D

Questão 3 – E

Questão 4 – C

Questão 5 – A

Questão 6 – B

Questão 7 – D

Questão 8 – A

Questão 9 – C

Questão 10 – B

Questão 11 – E

Questão 12 – D

### Questão 13.

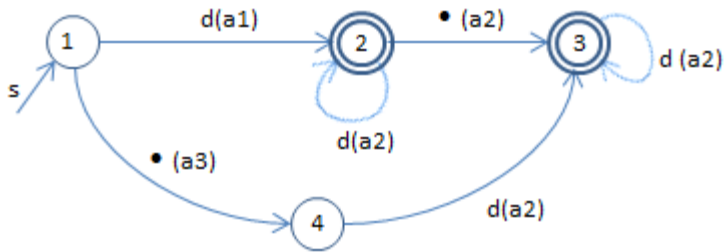
Conjunto de símbolos terminais = { d, • }. O símbolo d indica qualquer algarismo de zero a nove.

Conjunto não terminais = { S, A, B, C } com S sendo o símbolo inicial da Gramática. As produções da gramática são as seguintes:

$$S \rightarrow dA \mid \bullet C$$
$$A \rightarrow dA \mid \bullet B \mid \lambda$$
$$B \rightarrow dB \mid \lambda$$
$$C \rightarrow dB$$

Questão 14.

AEF que reconhece cadeias de números inteiros e com ponto decimal podendo iniciar ou terminar com o ponto decimal.

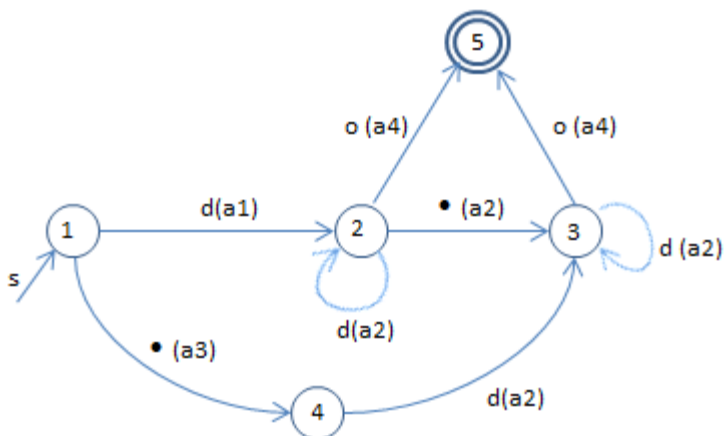


a1: Inicializa o lexema.

a2: Concatena o símbolo corrente ao lexema.

a3: Inicializa o lexema com o símbolo zero e o ponto.

AEF que reconhece cadeias de números inteiros e com ponto decimal podendo iniciar ou terminar com o ponto decimal e com transições "Others".



a1: Inicializa o lexema.

a2: Concatena o símbolo corrente ao lexema.

a3: Inicializa o lexema com o símbolo zero e o ponto.

a4: Finaliza o lexema, inserindo o fim de cadeia '\0'.

Questão 15.

```
q ← q0;  
c ← le_simbolo();  
enquanto (não acabou a cadeia de entrada) faça  
  se (M[q,c] não está definida) então  
    rejeita();  
  senão  
    q ← M [q][c];  
    c ← le_simbolo();  
  
se (Q[q] é FINAL)  
  aceita();  
senão  
  rejeita();
```

```
estado ← 0;  
simbolo ← next_char();  
enquanto ((simbolo != EOF) &&  
  (existe_transicao(estado,simbolo) >= 0)) {  
  estado ← M [estado][simbolo];  
  simbolo ← next_char();  
}  
if ((Q[estado]) && (EOF)) {  
  // Realiza ação semântica  
  reconhece();  
}  
else {  
  if (!EOF) {  
    sobrou_cadeia();  
  }  
  else /** acabou a cadeia mais não é um estado final */  
    não_e_estado_final();  
}
```