

Exercício 2 - Análise Sintática Parte 1

Aluno: Caiubi Oliveira Fonseca

1) Mostre a execução do *parser* mostrado anteriormente para as seguintes entradas

Entrada 1

```
if num=num then
  begin
    print num=num;
    print num=num;
  end
else
  print num = num
```

Execução

```
tok = IF
S()
  case IF: eat(IF) , tok=NUM
    E()
      eat(NUM), tok = EQ
      eat(EQ), tok = NUM
      eat(NUM), tok = THEN
      eat(THEN), tok = BEGIN
    S()
      case BEGIN: eat(BEGIN), tok = print
        S()
          case PRINT: eat(PRINT), tok=NUM
            E()
              eat(NUM), tok = EQ
              eat(EQ), tok = NUM
              eat(NUM), tok = SEMI
            L()
              case SEMI: eat(SEMI), tok = print
                S()
                  case PRINT: eat(PRINT), tok=NUM
                    E()
```

```
eat(NUM), tok = EQ
eat(EQ), tok = NUM
eat(NUM), tok = SEMI
L()
case SEMI: eat(SEMI), tok = END
S() // ERROR
```

Entrada 2

```
if num=num then
    begin
        print num=num;
        print num=num
    end
else
    print num = num
end
```

Execução

```
if num=num then
    begin
        print num=num;
        print num=num
    end
else
    print num = num

tok = IF
S()
case IF: eat(IF) , tok=NUM
E()
eat(NUM), tok = EQ
eat(EQ), tok = NUM
eat(NUM), tok = THEN
eat(THEN), tok = BEGIN
S()
case BEGIN: eat(BEGIN), tok = print
S()
case PRINT: eat(PRINT), tok=NUM
E()
eat(NUM), tok = EQ
eat(EQ), tok = NUM
eat(NUM), tok = SEMI
L()
case SEMI: eat(SEMI), tok = print
```

```

        S()
        case PRINT: eat(PRINT), tok=NUM
        E()
            eat(NUM), tok = EQ
            eat(EQ), tok = NUM
            eat(NUM), tok = END
        L()
            case END: eat(END), tok = ELSE
eat(else), tok = print
S()
    case PRINT: eat(PRINT), tok=NUM
    E()
        eat(NUM), tok = EQ
        eat(EQ), tok = NUM
        eat(NUM), tok = Fim de arquivo
break; // fim da análise

```

2) Mostre a implementação de um parser preditivo

```

final int IF=1, THEN=2, ELSE=3, BEGIN=4; END=5; PRINT=6, SEMI=7,
NUM=8, EQ=9; OPEN_PAR=10; CLOSE_PAR=11; FOR=12; EXPR=13;
OTHER=14;
int tok = getToken(); //lê primeiro token

void advance() {
    tok = getToken(); //lê próximo token
}
void eat(int t){
    if (tok==t) advance();
    else error();
}

optexpr(){
    switch (tok) {
        case EXPR:
            eat(EXPR);
    }
}

stmt() {
    switch(tok){
        case EXPR:
            eat(EXPR);
            eat(SEMI);
            break;
        case IF:

```

```
        eat(IF);
        eat(OPEN_PAR);
        eat(EXPR);
        eat(CLOSE_PAR);
        stmt();
        break;
    case FOR:
        eat(FOR);
        eat(OPEN_PAR);
        optexpr();
        eat(SEMI);
        optexpr();
        eat(SEMI);
        optexpr();
        eat(CLOSE_PAR);
        stmt();
        break;
    case OTHER:
        eat(OTHER);
        break;
    default:
        error();
}
}
```