

Grupos revisem os problemas da análise léxica e sintática ANTES de passarem para as fases seguintes. Os grupos que não fizeram revisão hoje (13/5), podem fazer na semana dia 24/5

• **Grupo 1: 6,0**

- **Parte Léxica (2,0): 1,6**

- definição de número junto com símbolos especiais (-0,1)
- define sinais e operadores relacionais em um mesmo token (<SIGN> e <RELOP>) (-0,2)
- não define “:”, “;”, ... como símbolos especiais (-0,1)
- no primeiro trabalho definiu que os identificadores e palavras reservadas não seriam case sensitive, mas neste trabalho são (-0,1)

- **Parte Sintática (5,0): 1,9**

LEMBREM de fazer o tratamento dos procedimentos pré-definidos READ e WRITE no analisador semântico

- alterou a gramática em <bloco> (parte de declaração de sub-rotina dever ser opcional) (-0,2)
- alterou a gramática de <Record> (incluiu “;” no fim) (-0,2)
- inclui símbolos especiais diretamente no código quando deveria estar na parte léxica (-0,2)
- alterou a gramática de <seção de parâmetros formais> (incluiu function e procedure) (-0,2)
- alterou a gramática de <comando composto> (falta “;”) (-0,2)
- alterou a gramática de <comando> (define “;” como obrigatório no final de <comando composto>) (-0,2)
- não fatorou corretamente (erro em <atribuição>) (-0,2)
- alterou a gramática de <atribuição> (não há “;” no fim) (-0,2)
- alterou a gramática do “IF” (-0,2)
- alterou a gramática de <expressão> (não colocou <relação>) (-0,2)
- **Tratamento de erros + Apresentação (3,0): 2,5**
- apresentação (-0,5)

- **Grupo 3: 8,5 (nota revisada)**

- **Parte Léxica (2,0):** 1,9

- Na parte I especificou tamanho de identificador com 512 caracteres, mas não tratou (-0,1)

- **Parte Sintática (5,0):** 4,6

- Utilizou lookahead(2) sem utilidade pois havia fatorado (-0,2)

- Alteração da gramática em <comando composto> (linha 312) – incluiu “;” como opcional ao final de comando (-0,2)

- **Tratamento de Erros + Apresentação (3,0):** 1,8

- Para todos os casos de testes a mensagem de erro está incompleta. Exemplo:

- “teste1e1.pas” não espera somente “end”, pode ser “:=”. (-0,2)

- “teste2e4.pas” – não há erro no local especificado (-0,2)

- O AL foi feito separado, mas não há explicação (-0,1)

- Apresentação (-0,5)

- **Grupo 4: 5,9**

- **Parte Léxica (2,0): 1,7**

- não define “_” (underline) como possível caractere do identificador (-0,2)
 - definiu símbolos especiais junto com identificadores e com números (-0,1)

- **Parte Sintática (5,0): 3,6**

- alterou a gramática em <bloco> (<parte de declaração de variáveis> e <parte de declaração de procedimentos> são opcionais) (-0,2)
 - alterou a gramática de <decl_vars> (o correto seria: lista_identificadores():tipo()) (-0,2)
 - alterou a gramática de <parte_decl_proc> (falta “;”) (-0,2)
 - alterou a gramática em comando composto (inclui condicional) (-0,2)
 - alterou a gramática do case (-0,2)
 - alterou a gramática de declaração de variáveis (-0,2)
 - não especificou <tipo> (-0,2)

- **Tratamento de erros + Apresentação (3,0): 0,5**

- não apresenta saída (-2,0)
 - apresentação (-0,5)
 - editou os arquivos TokenMgrError.java e ParseException.java (+0,5)

- **Grupo 20: 5,1**

- **Parte Léxica (2,0): 1,2**

- No trabalho 1 define comentários como {e} porém não trata neste trabalho (-0,2)
- Define modulo sem que seja atribuição do grupo (-0,1)
- read, write, boolean, char, integer, true e false são identificadores pré-definidos (-0,2) MAS é importante especificar o READ e WRITE, pois criaremos regras para eles no Analisador semântico
- não há modularização na definição de palavras reservadas e símbolos especiais (-0,1)
- não define “_” (underline) como possível parte de um identificador (-0,2)

- **Parte Sintática (5,0): 3,9**

LEMBREM de fazer o tratamento dos procedimentos pré-definidos READ e WRITE no analisador semântico

- alterou a gramática em <parametros> (var é opcional) (-0,2)
- alterou a gramática em <tipo> (incluindo <integer> e <char> sendo que o correto seria <id>) (-0,2)
- alterou a gramática em <expsimples> (incluiu <char>) (-0,2)
- incluiu divisão, mas a gramática não deve aceitar (-0,1)
- não trata corretamente “records” em <fator> (falta .<campo>) (-0,2)
- alterou a gramática do “IF” (não trata “else”) (-0,2)
- **Tratamento de Erros + Apresentação (3,0): 0**
- não apresenta resultados (-2,0)
- apresentação (-0,5)
- não editou tratamento de erros (-0,5)

- **Grupo 10: 8,2**

- **Parte Léxica (2,0): 1,3**

- define as palavras reservadas "for" e "function" mas estas não foram especificadas ao grupo (-0,2)
- "real" é um identificador pré-definido (-0,2)
- os sinais + e - são definidos em um mesmo token <SIGN> (-0,2)
- no trabalho 1 definiu que as palavras reservadas não seriam case sensitive, mas não tratou (-0,1). É só usar a opção:

```
options {
```

```
  IGNORE_CASE = true;
```

```
}
```

- **Parte Sintática (5,0): 4,2**

LEMBREM de fazer o tratamento dos procedimentos pré-definidos READ e WRITE no analisador semântico

- alterou a gramática em <bloco> (<comando composto> não é opcional) (-0,2)
- alterou a gramática em <parâmetros formais> (";" é opcional) (-0,2)
- alterou a gramática em <comando composto> (";" é opcional) (-0,2)
- falta divisão "/" para o tipo "real" (-0,2)
- Usou lookahead (sign) e não precisava
- Usou lookahead (else) e não precisava, pois o parser dá a interpretação correta para este comando naturalmente ambíguo, que é casar o else com o If mais próximo
- **Tratamento de Erros + Apresentação (3,0): 2,7**
- apresentação (-0,3)

- **Grupo 8: 9,6**

- **Parte Léxica (2,0):** 1,8

- “true” e “false” são identificadores pré-definidos (-0,2)

- Parte Sintática (5,0):** 4,8

LEMBREM de fazer o tratamento dos procedimentos pré-definidos READ e WRITE no analisador semântico

- alterou a gramática do <tipo> (depois do “of” aceita <ident>) (-0,2)

- Sugestão: deixar arrays com 1 dimensão, por simplicidade.

- **Tratamento de erros + Apresentação (3,0):** 3,0

- bons casos de teste e muito boa apresentação do trabalho!

- **Grupo 12: 6,6**

- **Parte Léxica (2,0): 0,8**

- define a palavra reservada "function" sem esta ser atribuída ao grupo (-0,1)
- read, write, integer e boolean são identificadores pré-definidos (-0,2)
- define "constante" sem esta ser atribuída ao grupo (-0,1)
- define "string" sem esta ser atribuída ao grupo (alterou a gramática ao colocar no write) (-0,2)
- definem palavra reservadas mas não utilizam (-0,2)
- definem símbolos especiais mas não utilizam (-0,2)
- no trabalho 1 definem identificadores com limite de 32 caracteres, mas não tratam (-0,1)
- no trabalho 1 definem palavras reservadas e identificadores como não sendo case sensitive, mas neste trabalho são (-0,1)

- **Parte Sintática (5,0): 3,8**

LEMBREM de fazer o tratamento dos procedimentos pré-definidos READ e WRITE no analisador semântico

- apresentam palavras reservadas diretamente no código quando estas deveriam estar somente no léxico (-0,2)
- apresentam símbolos especiais diretamente no código quando deveriam estar somente no léxico (-0,2)
- alterou a gramática de <parâmetros formais> (falta parênteses) (-0,2)
- utiliza lookahead(2) (deveria fatorar!) (-0,1)
- chamada de read()/write() => linha 235 e 236 OK
- "/" e "mod" não fazem parte da gramática (-0,2)
- alterou a gramática em <variável> (deve aceitar <expressão> (-0,2)

- **Tratamento de erros + Apresentação (3,0): 2,0**

- não alterou o arquivo de mensagens de erros; mensagens saem em inglês, diminuindo características de usabilidade (-0,5)
- apresentação (-0,5)

- **Grupo 7: 9,4**

- **Parte Léxica (2,0): 1,6**

- define tipos e constantes pré-definidos, mas estes são IDs em Pascal (-0,2)
- na especificação léxica diz que os ids tem no máximo 236, mas não checa (-0,1)
- não testa comentário que se abre e não fecha (-0,1)

- **Parte Sintática (5,0): 4,8**

LEMBREM de fazer o tratamento dos procedimentos pré-definidos READ e WRITE no analisador semântico

- colocou ; antes de END, em comando composto, alterando a especificação dada (-0,2)
- coloca lookahead (ELSE), mas não precisava, pois mesmo com o warning a interpretação dada pelo parser é a correta (associa o else com o IF mais próximo); deveria ter lido a documentação sobre lookaheads

- **Tratamento de erros + Apresentação (3,0): 3,0**

- 3 programas, com 2 versões cada (correta e incorreta)
- tradução dos arquivos de erro
- boa apresentação do trabalho, com comentários dos testes feitos, separa os tipos de tokens, comenta o código (0,5)

- **Grupo 6: 8.8**

- **Parte Léxica (2,0): 1.3**

- disse que só aceitaria no programa quebra de linhas, mas limpa tabs, Cr (-0,1)
- sensível a caixa baixa OK
- trata comentário de linha e de bloco OK
- na especificação léxica, o ID teria até 255, mas não checa (-0,1)
- erro conceitual do que seria palavra reservada, mas pelo menos as coloca antes de IDs (-0,1)
- erro de código de símbolos; juntas vários com um mesmo código (-0,2)
- problema de modularização: alguns símbolos são tratados no léxico outros ficam no sintático; deveria colocar todo o VT na parte léxica com um código igual ao token (-0,2)

- **Parte Sintática (5,0): 4.5**

LEMBREM de fazer o tratamento dos procedimentos pré-definidos READ e WRITE no analisador semântico

- - uso de lookahead(2) quando deveria ter fatorado a gramática. Isto leva a uma degradação da eficiência do parser além de indicar que não leu a parte de documentação do aplicativo. Pedi explicitamente para fatorar (-0,1)
 - alterou a especificação da gramática no comando composto “;” (-0,2)
 - case e repeat possuem ; antes do end quando não devia (-0,1)
 - não precisava colocar Lookahead no IF como 1 desde que este é o default. Se tivesse lido a especificação saberia que o parser dá a interpretação esperada para este comando ambíguo
 - faltam sinais em constante (-0,1)

- **Tratamento de erros + Apresentação (3,0): 3.0**

- 6 testes; 1 bem completo e correto; outros com erros indicando os símbolos esperados. Teste de erro em comentário
- Boa apresentação do trabalho (0,5)

- **Grupo 2: 8.3**

- **Parte Léxica (2,0): 1.2**

- elimina tab no parser, mas na especificação disse que só trataria (-0,1)
- especifica 2 tipos de comentários, mas só trata 1 (-0,1)
- trata read e write, integer e boolean com palavras reservadas, mas são identificadores (-0,2)
- COLOQUEM and, or, div junto com as outras palavras reservadas (código comentado)
- deixou código comentado de alguns símbolos que saíram da especificação léxica e foram parar no parser. Mas nem todos estão comentados. Os símbolos de operações serão muito importantes no tratamento de tipos e geração de código, assim é muito conveniente o seu tratamento na especificação léxica, devolvendo código para eles
- cada token deve ter um único código (-0,2)
- não incluem “_” na especificação de identificador (-0,1)
- prometem ID com até 50 caracteres, mas não tratam (-0,1)

- **Parte Sintática (5,0): 4.6**

- coloca lookahead (ELSE), mas não precisava, pois mesmo com o warning a interpretação dada pelo parser é a correta (associa o else com o IF mais próximo); deveria ter lido a documentação sobre lookaheads
- Mudou a gramática em definição dos tipos dos parâmetros, mas ela estava correta, pois da forma como especificada força que os parâmetros usem compatibilidade de tipos por NOME e não estrutural (-0,1)
- Já implementou READ e WRITE, MAS Write usa uma lista de EXPRESSÕES e não lista de IDs (-0,1)
- ótimo: fatorou a gramática nos pontos nos quais não era LL(1)
- case, definição de tipo e função estão OK
- Limpem os códigos comentados
- a definição de tipo deve chamar a antiga rotina <tipo> ::= <identificador> senão a vossa definição de tipos não seria usada como está previsto no PASCAL (-0,2)

LEMBREM de fazer o tratamento dos procedimentos pré-definidos READ e WRITE no analisador semântico: o grupo já fez

- **Tratamento de erros + Apresentação (3,0): 2.5**

- testa com programas com erros e corretos
- tradução dos arquivos de erros
- apresentação do trabalho muito precária, pois imprimiram arquivo de tratamento de erros em vez do de especificação do parser; deixam partes com comentários de decisões antigas dentro do código; não fazem uma boa separação dos símbolos que recebem código na parte de especificação léxica dos que são tratados diretamente no parser: devem decidir por uma ou outra forma, embora trazer todo o tratamento no léxico (com definição de token e código) seja mais modular e ajuda em futuras manutenções (-0,5)

- **Grupo 14: 8.6**

- **Parte Léxica (2,0): 1.4**

- erro em comentário: não permite * dentro de { ... } (-0,1)
- erro em definir VT com { e } (-0,1)
- erro em definir { e }, pois vocês não tem arrays (-0,1)
- erro em definir string e type (-0,1)
- não trata char; deve reconhecer o pacote "a" no Léxico (-0,2)

-OK em explicitar READ e WRITE, pois iremos fazer a regra deles no Analisador Semântico

- **Parte Sintática (5,0): 4.3**

LEMBREM de fazer o tratamento dos procedimentos pré-definidos READ e WRITE no analisador semântico

-erro de modularização: colocou mensagem de sucesso de compilação na gramática (-0,1)

- erro em definição de tipo (-0,1)

- Erro: só permite 1 procedimento de nível 0 (-0,2)

- Erro: não devia tratar string (-0,1)

- Coloca lookahead (2) quando pedi para deixar a gramática LL(1) (-0,2)

- **Tratamento de erros + Apresentação (3,0): 2,9**

Boa apresentação do trabalho (0,5)

8 testes: ótima testagem.

Editou arquivo de erros léxicos e sintáticos: ótimo

Imprimiu info desnecessária nas mensagens; tracing de tokens é importante para depuração de erros e não para usuários (-0,1)

- **Grupo 18: 9,1**

- **Parte Léxica (2,0): 1.8**

- juntou real e inteiro na parte léxica: deixe na parte sintática em fator.

- colocou true e false como reservadas, deixe como IDs (-0,1)

- especificou IDs com caixa baixa mas permitiu alta em programa (-0,1)

- **Parte Sintática (5,0): 4.3**

LEMBREM de fazer o tratamento dos procedimentos pré-definidos READ e WRITE no analisador semântico

- usa lookahead(2) quando não devia. Pedi explicitamente para fatorar pontos na gramática que não eram LL(1). (-0,2)
- problema de modularização: joga mensagem de aceitação do parsing na gramática quando deveria deixar na parte de declaração da classe (-0,1)
- alterou a especificação da gramática em comando composto (-0,2)
- alterou a especificação de repeat (-0,1)
- colocar divisão real em termo (-0,1)
- **Tratamento de erros + Apresentação (3,0): 3.0**
3 programas corretos e 3 com erros. Traduziu os arquivos de erros
Apresentação (0,5)

• **Grupo 5 (vale 8.0 – entrega atrasada): 8.9 – 2.0 = 6,9**

- **Parte Léxica (2,0): 1.7**
 - Read e Write foram explicitados no Léxico. OK, pois vai precisar fazer a regra destes procedimentos no analisador semântico.
 - define sinal de + e – e os operadores + e - , mas não devia, pois não há distinção léxica entre eles (-0,2)
 - ID não tem underscore, mas devia (-0,1)
- **Parte Sintática (5,0): 4.5**

LEMBREM de fazer o tratamento dos procedimentos pré-definidos READ e WRITE no analisador semântico (Já fizeram)

-cria não terminal bloco para tratar procedimentos e não devia, pois alterou a gramática especificada (-0,2)

-cria outra regra para <tipo> e não devia, deixando <tipo> sem ser usado (-0,1)

- usa lookahead (2) em 2 partes e não devia, pois pedi para tornar a gramática LL(1) (-0,2)

- **Tratamento de erros + Apresentação (3,0): 2.7**

Editaram o arquivo de Erros léxicos e sintáticos (OK).

Testaram problema em comentário (OK). Criaram 7 programas, mas não o separaram em certos e errados, não comenta qual erro estão testando. Apresentação (0,3)

- **Grupo 9: 9.8**

- **Parte Léxica (2,0): 1.9**

- trata " e não devia, pois não possuem char/string (-0,1)

- **Parte Sintática (5,0): 5.0**

LEMBREM de fazer o tratamento dos procedimentos pré-definidos READ e WRITE no analisador semântico

- colocou lookahead(1) no IF mas este é o default. Não precisa fazer nada neste comando, pois o parser consegue dar a interpretação correta (casar o else com o IF mais próximo) para esta ambiguidade

- **Tratamento de erros + Apresentação (3,0): 2.9**

11 programas, com erros variados. Parabéns pela vasta testagem.
Apresentação do trabalho (0,4).

- **Grupo 17: 8.5**

- **Parte Léxica (2,0): 1.4**

- trata comentário como /* ... *//, não havia especificado nas considerações léxicas
- Erro: trata palavras reservadas que não são do grupo: function, case, const, file, type (-0,2)
- Erro: trata integer, true, false e boolean como palavras reservadas e não como IDs (-0,1)
- trata real, caractere e string e não devia, pois não é tarefa do grupo (-0,2)
- ID em Pascal tem underscore e não colocou (-0,1)

- **Parte Sintática (5,0): 4.5**

LEMBREM de fazer o tratamento dos procedimentos pré-definidos READ e WRITE no analisador semântico

- inclui regra de definição de tipo (construtor de tipo) sem ser tarefa do grupo (-0,1)
- não inclui record e array como tipo, como estava especificado na gramática estendida (-0,1)
- não permite Record em declaração de variáveis, como estava na gramática estendida (na verdade, deveria ser colocada em tipo, pois assim a declaração de variáveis se beneficiaria) (-0,1)
- não permite o uso de variáveis de Record, nem variáveis indexadas em FATOR (-0,2), nem na atribuição ($x[1] := 1$, $x.y := 1$)

- **Tratamento de erros + Apresentação (3,0): 2.6**

- Boa testagem, usou 7 programas sem erros e 7 com erros, separados e com explicação
- formatação da parte léxica atrapalha leitura (-0,1)
- Não editou os arquivos de erros (léxicos e sintáticos), assim as mensagens saem em inglês, sendo que o foco seria português. Isto tornaria a saída mais amigável (-0,1)
- Não apresenta a parte de definição da classe do javacc. Ela é importante, pois lá também tem mensagens de erros (-0,2)