

Grupo 1

Correção:

- Para alguns erros léxicos, não é alertado o tipo do erro. O parser imprime somente a cadeia que ocasionou o erro
- Em alguns casos de teste, o parser para quando encontra um erro, pois não encontra dois erros
- Quando o programa está correto não é impresso nada na tela
- Alguns erros sintáticos foram apontados na linha errada
- Em alguns casos de teste é apontado dois erros sintáticos, quando há apenas um
- Em alguns casos de teste, foram apontados erros sintáticos diferentes
- Erro sintático no while, mesmo quando ele está correto
- O comando while não precisa possuir sua condição entre parênteses

Dicas:

- Alguns erros léxicos são impressos duas vezes pelo parser, chame as mensagens de erro apenas do sintático
- Verifique corretamente os tokens de sincronização, para que não seja apontado dois erros sintáticos quando há apenas 1
-

Grupo 2

Correção:

- A recuperação de erro em alguns caso, ocasionou outros erros

Dicas:

- Padronizem o idioma do trabalho
- Adicione o \r em caracteres de espaço, pois mesmo usando Linux, alguns arquivos podem usar o padrão de quebra de linha do windows
- Algumas mensagens de erro podem ser melhoradas, a respeito de qual token era esperado
- O compilador de vocês aceitam char, string, etc.

Grupo 3

Correção:

- Quando há erros léxicos, novos erros sintáticos são disparados
- Era esperando um) quando tinha um)
- Expressões podem ter parênteses
- Quando deu erro na declaração de program, o compilador apontou que faltava um '!' após o end de um procedimento

Dicas:

- Melhore as mensagens de erro, deixe-as mais informativas (erro de atribuição por exemplo)

Grupo 4

Correção

- Quando há erro léxico, o compilador esperava um end.
 - Comentário não fechado
- Dicas:
- Quando o código está correto, não é impresso nenhuma mensagem
 - Verifique alguns tokens de sincronização para diminuir a quantidade de erros
 - Melhore as mensagens de erro para erros com expressões lógicas

Grupo 5

Correção

- Quando há um erro léxico, o compilador lança outros erros
- Dicas:
- Verifique os tokens de sincronização, vários erros estão sendo lançados após um erro ocorrer
 - Quando o programa está correto, não é apresentado nenhuma mensagem
 - Melhore a mensagem para falta de operadores lógicos

Grupo 6

Correção

- Quando há erro léxico, não é verificado corretamente o token de sincronização
- Dicas:
- Makefile não funciona , cuidado
 - Quando há um erro também é impresso sucesso, acho que não deveria ter sucesso né?
 - Melhore mensagens de erro quando há erro de atribuição
 - O que isso significa: Encontrado o Símbolo ":= " mas estava esperado

Grupo 7

Correção

- Loop infinito quando há erro léxico
- Loop infinito quando há erro de atribuição
- Loop infinito quando não há then
- Loop infinito quando não há do

- Erro de comentário não fechado
- Dicas
- Quando há um erro, a compilação foi um sucesso?
- Deixar as mensagens de erro mais informativas

Grupo 8

Correção

- Decisões de projeto e tratamento de erro no relatório
- Vários erros sintáticos apontados quando havia somente um erro léxico
- Recuperação de erros sintáticos está lançando muitos outros erros
- Erro de repeat sem until, mas havia esses comando
- Em erros de expressões lógicas, o compilador aponta que “era esperado operadores de soma, subtração...”
- Dicas
- Quando há um erro, é impresso Compilado com sucesso. Isso está certo?
- Aponta o mesmo erro várias vezes

Grupo 9

Correção

- Loop infinito quando teve erro léxico
- Encontrou alguns erros que foram informados errado
- Loop quando há while em alguns programas corretos, apontando erros inexistentes (falta do do)
- Erro sintático onde não tem, sinal de < é booleano
- Dicas
- Cuidado com os tokens de sincronização

Grupo 10

Correção

- Vários erros sintáticos e léxicos são apontados quando há apenas um erro léxico
- Loop infinito em alguns erros sintáticos
- Erro léxico e sintático em programas corretos
- Não aponta erro de falta de 'then'

- Aponta um erro sintático diferente quando não há o comando do Dicas
- Quando fizerem relatórios, coloquem o texto justificado!

Grupo 11

Correção

- Aponta que era inesperado um número quando era possível utilizar um número Dicas
- Melhore alguns mensagens de erro, é apresentado cod_var

Grupo 12

Correção

- Vários erros sintáticos lançados quando havia um erro léxico
- Erro no comando while correto
- Erro de memória em alguns casos de teste
- Quando há um while errado, não apontou o erro corretamente Dicas
- O gráfico sintático ficou ruim para ser lido
- Verifique os tokens de sincronização que estão usando, estão sendo lançadas muitos erros sintáticos

Grupo 13

Correção

- Apontados alguns erros sintáticos errados Dicas
- Padronizem o idioma das mensagens de erro

Grupo 14

Correção

- Quando há erros léxico, é disparados vários erros sintáticos
- Apontado erro sintático em um programa correto
- Erros para comando while
- Erros apontados na linha errada
- Na maioria dos testes, o compilador sempre diz que era esperado um .

Dicas

- Melhorar mensagens de erro
- Quando for um erro léxico, aponte-o

Grupo 15

Correção

- Loop infinito em um programa correto
- Muitos erros sintáticos quando há apenas um erro
- Loop infinito em programa com apenas um erro sintático
-

Dicas

- Quando o programa tem erros, é impresso compilado com sucesso

Grupo 16

Correção

- O comando while, em pascal, não precisa necessariamente de (e)
- Encontrou erros inexistentes: Expected semicolon, but found

Dicas

- Melhorar as mensagens de erro

Grupo 17

Correção:

- o comando while, em pascal, não precisa necessariamente no (e)
- Vários erros sintáticos apontados, quando havia um na declaração de variáveis
- Vários erros sintáticos levantados quando havia apenas um
- Erro de ELSE quando havia ELSE
-

Dicas:

- O compilador sempre precisa de um arquivo de saída
- Imprima alguma mensagem quando a compilação ocorrer com sucesso

Grupo 18

Correção

- Vários erros sintáticos quando haviam apenas um erro em programa correto

Dicas

- Quando fizerem relatório, justifiquem o texto
- Melhorar as mensagens do compilador

Grupo 19

Correção:

- Não foi apontado erro léxico
- Loop infinito em programa com erro sintático
- Aponta erro sintático em programa correto
-

Dicas:

- Melhorar mensagens de erro
- É impresso várias vezes a mesma mensagem de erro

Grupo 20

Correção

- Erro sintático em programa correto (while com e sem parêntese)
- Loop infinito em programa com erro sintático

Dicas

- Melhorar mensagens de erro, imprima o número da linha que ocasionou o erro

Grupo 21

Correção

- Erro de comentário não fechado

Dicas

- & é um erro léxico, não sintático
- Quando o programa está correto, não é impresso nenhuma mensagem

Grupo 22

Correção

- Vários erros sintáticos lançados, quando haviam apenas um Dicas
- Alguns erros são impressos em linhas erradas
- Decisões de projeto no relatório (tratamento de erro, etc.)