COMP LabBook ERE 2021 1

Lucas M. Schnorr

March 12, 2022

Contents

1 E2

1.1 Logs de testes, tabelas de decisão e entradas de teste

Estes são os arquivos fornecidos no ZIP.

```
unzip -vl e2/e2_CSV_LOG.zip

Archive:  e2/e2_CSV_LOG.zip
 Length   Method    Size  Cmpr    Date    Time   CRC-32   Name
--------  ------  ------- ---- ---------- ----- --------  ----
  173461  Defl:N    16020  91% 2022-03-12 14:23 6313f52c  e2.log
  143025  Defl:N    14166  90% 2022-03-12 14:23 62e38769  e2.csv
    1743  Defl:N      398  77% 2022-03-12 14:23 583c5782  e2_esperado.csv
     213  Defl:N      100  53% 2022-03-12 14:23 17e02690  e2_output_objetivo.csv
    1706  Defl:N      398  77% 2022-03-12 14:23 2b2f0eac  e2_output_quais_os_testes_mais_errado
   20607  Defl:N     2131  90% 2022-03-12 14:23 1838f709  e2_output_quem_errou_o_que.csv
     278  Defl:N      121  57% 2022-03-12 14:22 27b765f7  e2_output_subjetivo.csv
--------          ------- ---                            -------
  341033            33334  90%                            7 files
```

- `e2.log`: o arquivo contém o log cru dos testes
- `e2.csv`: após parsing para obter apenas a saída, as colunas são as seguintes:
  - Grupo: identificador do grupo
  - Teste: identificador do teste
  - Result: resultado da solução do grupo
    - 0, aceitou a entrada
    - 1, refugou a entrada
  - Razao
    - Vazio se aceitou a entrada
    - Erro sintático (possivelmente) informando o erro
- `e2_esperado.csv`: informada se a entrada é correta ou incorreta sintaticamente de acordo com a especificação da E2
  - Teste: identificador do teste
  - Esperado
    - 0, entrada sintaticamente correta de acordo com a E2
    - 1, entrada sintaticamente **incorreta** de acordo com a E2
- `e2_output_objetivo.csv`
  - E2.O: nota final do teste objetivo (sobre 10)
- `e2_output_quais_os_testes_mais_errados.csv`
  - Estatística que ilustra os testes que mais deram errado
- `e2_output_quem_errou_o_que.csv`
  - Indica apenas os testes que falharam, para cada grupo
  - Grupo, Test, Result, Esperado, Correto, Razao
    - Result: resultado da solução do grupo
    - Esperado: resultado esperado para aquele teste
  - Por exemplo:
    - Temos a linha `GrupoA,asl031,1,0,FALSE,line 4:  syntax error`
      - Esse teste foi considerado errado pois o GrupoA informou erro sintático na entrada asl031 (1 na coluna Result) quando na realidade sintaticamente o teste está correto (0 na coluna Esperado).
    - Para fazer essas verificações, consulte o TGZ (listagem abaixo).

Estes são os arquivos fornecidos no TGZ.

- Cada arquivo tem um comentário //CORRECT ou //INCORRECT para facilitar a identificação se o teste está correto ou incorreto do ponto de vista da E2. Estes comentários são utilizados para definir o conteúdo de `e2_esperado.csv`.

```
tar vftz e2/E2.tgz

-rw-r--r-- schnorr/schnorr  18 2021-09-14 12:18 asl001
-rw-r--r-- schnorr/schnorr  25 2021-09-14 12:18 asl002
-rw-r--r-- schnorr/schnorr  21 2021-09-14 12:18 asl003
-rw-r--r-- schnorr/schnorr  28 2021-09-14 12:18 asl004
-rw-r--r-- schnorr/schnorr  19 2021-09-14 12:18 asl005
-rw-r--r-- schnorr/schnorr  18 2021-09-14 12:18 asl006
-rw-r--r-- schnorr/schnorr  18 2021-09-14 12:18 asl007
-rw-r--r-- schnorr/schnorr  20 2021-09-14 12:18 asl008
-rw-r--r-- schnorr/schnorr  51 2021-09-14 12:18 asl009
-rw-r--r-- schnorr/schnorr  21 2021-09-14 12:18 asl010
-rw-r--r-- schnorr/schnorr  18 2021-09-14 12:18 asl011
-rw-r--r-- schnorr/schnorr  24 2021-09-14 12:18 asl012
-rw-r--r-- schnorr/schnorr  42 2021-09-14 12:18 asl013
-rw-r--r-- schnorr/schnorr  57 2021-09-14 12:18 asl014
-rw-r--r-- schnorr/schnorr  29 2021-09-14 12:18 asl015
-rw-r--r-- schnorr/schnorr  57 2021-09-14 12:18 asl016
-rw-r--r-- schnorr/schnorr  57 2021-09-14 12:18 asl017
-rw-r--r-- schnorr/schnorr  35 2021-09-14 12:18 asl018
-rw-r--r-- schnorr/schnorr  43 2021-09-14 12:18 asl019
-rw-r--r-- schnorr/schnorr  45 2021-09-14 12:18 asl020
```

```
-rw-r--r-- schnorr/schnorr  49 2021-09-14 12:18 asl021
-rw-r--r-- schnorr/schnorr  41 2021-09-14 12:18 asl022
-rw-r--r-- schnorr/schnorr  78 2021-09-14 12:18 asl023
-rw-r--r-- schnorr/schnorr  66 2021-09-14 12:18 asl024
-rw-r--r-- schnorr/schnorr  23 2021-09-14 12:18 asl025
-rw-r--r-- schnorr/schnorr  26 2021-09-14 12:18 asl026
-rw-r--r-- schnorr/schnorr  30 2021-09-14 12:18 asl027
-rw-r--r-- schnorr/schnorr  35 2021-09-14 12:18 asl028
-rw-r--r-- schnorr/schnorr  34 2021-09-14 12:18 asl029
-rw-r--r-- schnorr/schnorr  34 2021-09-14 12:18 asl030
-rw-r--r-- schnorr/schnorr  29 2021-09-14 12:18 asl031
-rw-r--r-- schnorr/schnorr  55 2021-09-14 12:18 asl032
-rw-r--r-- schnorr/schnorr  31 2021-09-14 12:18 asl033
-rw-r--r-- schnorr/schnorr  41 2021-09-14 12:18 asl034
-rw-r--r-- schnorr/schnorr  38 2021-09-14 12:18 asl035
-rw-r--r-- schnorr/schnorr  33 2021-09-14 12:18 asl036
-rw-r--r-- schnorr/schnorr  31 2021-09-14 12:18 asl037
-rw-r--r-- schnorr/schnorr  31 2021-09-14 12:18 asl038
-rw-r--r-- schnorr/schnorr  33 2021-09-14 12:18 asl039
-rw-r--r-- schnorr/schnorr  39 2021-09-14 12:18 asl040
-rw-r--r-- schnorr/schnorr  38 2021-09-14 12:18 asl041
-rw-r--r-- schnorr/schnorr  45 2021-09-14 12:18 asl042
-rw-r--r-- schnorr/schnorr  36 2021-09-14 12:18 asl043
-rw-r--r-- schnorr/schnorr  51 2021-09-14 12:18 asl044
-rw-r--r-- schnorr/schnorr  71 2021-09-14 12:18 asl045
-rw-r--r-- schnorr/schnorr  40 2021-09-14 12:18 asl046
-rw-r--r-- schnorr/schnorr  45 2021-09-14 12:18 asl047
-rw-r--r-- schnorr/schnorr  65 2021-09-14 12:18 asl048
-rw-r--r-- schnorr/schnorr  80 2021-09-14 12:18 asl049
-rw-r--r-- schnorr/schnorr  42 2021-09-14 12:18 asl050
-rw-r--r-- schnorr/schnorr  61 2021-09-14 12:18 asl051
-rw-r--r-- schnorr/schnorr  67 2021-09-14 12:18 asl052
-rw-r--r-- schnorr/schnorr  29 2021-09-14 12:18 asl053
-rw-r--r-- schnorr/schnorr  63 2021-09-14 12:18 asl054
-rw-r--r-- schnorr/schnorr  56 2021-09-14 12:18 asl055
-rw-r--r-- schnorr/schnorr  41 2021-09-14 12:18 asl056
-rw-r--r-- schnorr/schnorr  41 2021-09-14 12:18 asl057
-rw-r--r-- schnorr/schnorr  43 2021-09-14 12:18 asl058
-rw-r--r-- schnorr/schnorr  31 2021-09-14 12:18 asl059
-rw-r--r-- schnorr/schnorr  33 2021-09-14 12:18 asl060
-rw-r--r-- schnorr/schnorr  36 2021-09-14 12:18 asl061
-rw-r--r-- schnorr/schnorr  45 2021-09-14 12:18 asl062
-rw-r--r-- schnorr/schnorr  49 2022-03-10 10:36 asl063
-rw-r--r-- schnorr/schnorr  72 2021-09-14 12:18 asl064
-rw-r--r-- schnorr/schnorr  81 2021-09-14 12:18 asl065
-rw-r--r-- schnorr/schnorr  69 2021-09-14 12:18 asl066
-rw-r--r-- schnorr/schnorr  53 2021-09-14 12:18 asl067
-rw-r--r-- schnorr/schnorr  63 2021-09-14 12:18 asl068
-rw-r--r-- schnorr/schnorr  61 2021-09-14 12:18 asl069
-rw-r--r-- schnorr/schnorr  62 2021-09-14 12:18 asl070
-rw-r--r-- schnorr/schnorr  61 2021-09-14 12:18 asl071
-rw-r--r-- schnorr/schnorr  52 2021-09-14 12:18 asl072
-rw-r--r-- schnorr/schnorr  52 2021-09-14 12:18 asl073
-rw-r--r-- schnorr/schnorr  61 2021-09-14 12:18 asl074
-rw-r--r-- schnorr/schnorr  65 2021-09-14 12:18 asl075
-rw-r--r-- schnorr/schnorr  64 2021-09-14 12:18 asl076
-rw-r--r-- schnorr/schnorr  41 2021-09-14 12:18 asl077
-rw-r--r-- schnorr/schnorr  50 2021-09-14 12:18 asl078
-rw-r--r-- schnorr/schnorr  46 2021-09-14 12:18 asl079
-rw-r--r-- schnorr/schnorr  65 2021-09-14 12:18 asl080
-rw-r--r-- schnorr/schnorr  80 2021-09-14 12:18 asl081
-rw-r--r-- schnorr/schnorr  16 2021-09-14 12:18 asl082
-rw-r--r-- schnorr/schnorr  34 2021-09-14 12:18 asl083
-rw-r--r-- schnorr/schnorr  50 2021-09-14 12:18 asl084
-rw-r--r-- schnorr/schnorr  40 2021-09-14 12:18 asl085
-rw-r--r-- schnorr/schnorr  44 2021-09-14 12:18 asl086
-rw-r--r-- schnorr/schnorr  51 2021-09-14 12:18 asl087
```

```
-rw-r--r-- schnorr/schnorr  63 2021-09-14 12:18 asl088
-rw-r--r-- schnorr/schnorr  34 2021-09-14 12:18 asl089
-rw-r--r-- schnorr/schnorr  66 2021-09-14 12:18 asl090
-rw-r--r-- schnorr/schnorr  67 2021-09-14 12:18 asl091
-rw-r--r-- schnorr/schnorr  40 2021-09-14 12:18 asl092
-rw-r--r-- schnorr/schnorr  37 2021-09-14 12:18 asl093
-rw-r--r-- schnorr/schnorr  62 2021-09-14 12:18 asl094
-rw-r--r-- schnorr/schnorr  80 2021-09-14 12:18 asl095
-rw-r--r-- schnorr/schnorr  80 2021-09-14 12:18 asl096
-rw-r--r-- schnorr/schnorr 100 2021-09-14 12:18 asl097
-rw-r--r-- schnorr/schnorr  31 2021-09-14 12:18 asl098
-rw-r--r-- schnorr/schnorr  32 2021-09-14 12:18 asl099
-rw-r--r-- schnorr/schnorr  38 2021-09-14 12:18 asl100
-rw-r--r-- schnorr/schnorr  50 2021-09-14 12:18 asl101
-rw-r--r-- schnorr/schnorr  38 2021-09-14 12:18 asl102
-rw-r--r-- schnorr/schnorr  49 2021-09-14 13:56 asl103
-rw-r--r-- schnorr/schnorr  45 2021-09-14 12:18 asl104
-rw-r--r-- schnorr/schnorr  35 2021-09-14 12:18 asl105
-rw-r--r-- schnorr/schnorr  49 2021-09-14 12:18 asl106
-rw-r--r-- schnorr/schnorr  41 2021-09-14 12:18 asl107
-rw-r--r-- schnorr/schnorr  37 2021-09-14 12:18 asl108
-rw-r--r-- schnorr/schnorr 163 2021-09-14 12:18 asl109
-rw-r--r-- schnorr/schnorr  62 2021-09-14 12:18 asl110
-rw-r--r-- schnorr/schnorr  63 2021-09-14 12:18 asl111
-rw-r--r-- schnorr/schnorr  63 2021-09-14 12:18 asl112
-rw-r--r-- schnorr/schnorr  62 2021-09-14 12:18 asl113
-rw-r--r-- schnorr/schnorr  10 2021-09-14 12:49 asl114
-rw-r--r-- schnorr/schnorr  60 2021-09-14 12:18 asl115
-rw-r--r-- schnorr/schnorr  65 2021-09-14 12:18 asl116
-rw-r--r-- schnorr/schnorr  78 2021-09-14 12:18 asl117
-rw-r--r-- schnorr/schnorr  88 2021-09-14 12:18 asl118
-rw-r--r-- schnorr/schnorr  36 2021-09-14 12:18 asl119
-rw-r--r-- schnorr/schnorr  38 2021-09-14 12:18 asl120
-rw-r--r-- schnorr/schnorr  42 2021-09-14 12:18 asl121
-rw-r--r-- schnorr/schnorr  58 2021-09-14 12:18 asl122
-rw-r--r-- schnorr/schnorr  56 2021-09-14 12:18 asl123
-rw-r--r-- schnorr/schnorr  59 2021-09-14 12:18 asl124
-rw-r--r-- schnorr/schnorr  36 2021-09-14 12:18 asl125
-rw-r--r-- schnorr/schnorr  84 2021-09-14 12:18 asl126
-rw-r--r-- schnorr/schnorr  73 2021-09-14 12:18 asl127
-rw-r--r-- schnorr/schnorr  56 2021-09-14 12:18 asl128
-rw-r--r-- schnorr/schnorr  61 2021-09-14 12:18 asl129
-rw-r--r-- schnorr/schnorr  47 2021-09-14 12:18 asl130
-rw-r--r-- schnorr/schnorr  46 2021-09-14 12:18 asl131
-rw-r--r-- schnorr/schnorr  49 2021-09-14 12:18 asl132
-rw-r--r-- schnorr/schnorr  55 2021-09-14 12:18 asl133
-rw-r--r-- schnorr/schnorr  35 2021-09-14 12:18 asl134
-rw-r--r-- schnorr/schnorr  42 2021-09-14 12:18 asl135
-rw-r--r-- schnorr/schnorr  40 2021-09-14 12:18 asl136
-rw-r--r-- schnorr/schnorr  44 2021-09-14 12:18 asl137
-rw-r--r-- schnorr/schnorr  43 2021-09-14 12:18 asl138
-rw-r--r-- schnorr/schnorr  43 2021-09-14 12:18 asl139
-rw-r--r-- schnorr/schnorr  43 2021-09-14 12:18 asl140
-rw-r--r-- schnorr/schnorr  43 2021-09-14 12:18 asl141
-rw-r--r-- schnorr/schnorr  43 2021-09-14 12:18 asl142
-rw-r--r-- schnorr/schnorr  43 2021-09-14 12:18 asl143
-rw-r--r-- schnorr/schnorr  43 2021-09-14 12:18 asl144
-rw-r--r-- schnorr/schnorr  44 2021-09-14 12:18 asl145
-rw-r--r-- schnorr/schnorr  44 2021-09-14 12:18 asl146
-rw-r--r-- schnorr/schnorr  44 2021-09-14 12:18 asl147
-rw-r--r-- schnorr/schnorr  44 2021-09-14 12:18 asl148
-rw-r--r-- schnorr/schnorr  44 2021-09-14 12:18 asl149
-rw-r--r-- schnorr/schnorr  52 2021-09-14 12:18 asl150
-rw-r--r-- schnorr/schnorr  40 2021-09-14 12:18 asl151
-rw-r--r-- schnorr/schnorr  40 2021-09-14 12:18 asl152
-rw-r--r-- schnorr/schnorr  45 2021-09-14 12:18 asl153
-rw-r--r-- schnorr/schnorr  54 2021-09-14 12:18 asl154
```

```
-rw-r--r-- schnorr/schnorr  55 2021-09-14 12:18 asl155
-rw-r--r-- schnorr/schnorr  54 2021-09-14 12:18 asl156
-rw-r--r-- schnorr/schnorr  52 2021-09-14 12:18 asl157
-rw-r--r-- schnorr/schnorr  52 2021-09-14 12:18 asl158
-rw-r--r-- schnorr/schnorr  28 2021-09-14 12:30 asl159
-rw-r--r-- schnorr/schnorr 126 2022-03-10 10:11 asl160
-rw-r--r-- schnorr/schnorr  69 2022-03-10 10:41 asl161
-rw-r--r-- schnorr/schnorr  62 2021-09-14 12:33 asl162
-rw-r--r-- schnorr/schnorr  53 2022-03-10 10:01 asl163
-rw-r--r-- schnorr/schnorr  92 2022-03-10 10:02 asl164
-rw-r--r-- schnorr/schnorr  67 2021-09-14 12:39 asl165
-rw-r--r-- schnorr/schnorr  65 2021-09-14 12:46 asl166
-rw-r--r-- schnorr/schnorr  69 2022-03-10 10:03 asl167
-rw-r--r-- schnorr/schnorr  93 2021-09-14 13:45 asl168
-rw-r--r-- schnorr/schnorr  53 2021-09-14 17:31 asl169
-rw-r--r-- schnorr/schnorr  66 2022-02-19 10:20 asl170
-rw-r--r-- schnorr/schnorr  43 2022-02-19 10:24 asl171
-rw-r--r-- schnorr/schnorr  57 2022-03-10 09:25 asl172
-rw-r--r-- schnorr/schnorr  43 2022-03-10 09:25 asl173
-rw-r--r-- schnorr/schnorr  46 2022-03-10 09:26 asl174
-rw-r--r-- schnorr/schnorr  47 2022-03-10 09:26 asl175
-rw-r--r-- schnorr/schnorr  71 2022-03-10 09:27 asl176
-rw-r--r-- schnorr/schnorr 106 2022-03-10 09:28 asl177
-rw-r--r-- schnorr/schnorr 104 2022-03-10 09:28 asl178
-rw-r--r-- schnorr/schnorr  66 2022-03-10 09:41 asl179
-rw-r--r-- schnorr/schnorr  30 2022-03-10 09:43 asl180
-rw-r--r-- schnorr/schnorr  56 2022-03-10 09:43 asl181
-rw-r--r-- schnorr/schnorr  61 2022-03-10 10:05 asl182
-rw-r--r-- schnorr/schnorr 130 2022-03-10 10:11 asl183
-rw-r--r-- schnorr/schnorr  62 2022-03-10 10:05 asl184
-rw-r--r-- schnorr/schnorr  60 2022-03-10 10:20 asl185
-rw-r--r-- schnorr/schnorr  64 2022-03-10 10:23 asl186
-rw-r--r-- schnorr/schnorr  70 2022-03-10 10:28 asl187
-rw-r--r-- schnorr/schnorr  80 2022-03-10 10:31 asl188
-rw-r--r-- schnorr/schnorr  73 2022-03-10 10:33 asl189
-rw-r--r-- schnorr/schnorr  56 2022-03-10 10:35 asl190
-rw-r--r-- schnorr/schnorr  42 2022-03-10 10:37 asl191
-rw-r--r-- schnorr/schnorr  53 2022-03-10 10:38 asl192
```

1.2 Comentários

1.2.1 Gerais

- Em todas as regras com recursão gramatical (listas de comandos, listas de funções, etc), dar preferência para a recursão à esquerda pois isso facilita as regras de associatividade (em expressões, por exemplo).
- Colocar comentários na gramática auxilia o professor na avaliação! ;-)

1.2.2 GrupoA

- O arquivo main.c não pode ser renomeado para main.h: ter a função principal em um cabeçalho é uma prática ruim. O que aconteceria se esse arquivo cabeçalho é incluso em mais de um .c compilado separadamente?
- Seria bom ter comentários na especificação gramatical
- Achei particularmente estranho o fato de uma `expressao` ser somente `logica` e a `aritmetica` ser um `operando_logico`. Qual seria a razão para organizar assim?
- Como é garantida a precedência de operações aritméticas? Veja que a regra `operador_aritmetico_binario` faz com que todas as operações ali listadas estejam no mesmo nível de precedência, o que contradiz a especificação pois a soma binária deve ter menos precedência que a multiplicação, por exemplo. Aqui, isso é inócuo (os testes passam pois gramaticalmente está correto), mas nas próximas etapas os testes falharão. Sugiro revisar tomando-se por base os exemplos dados nas primeiras aulas teóricas de análise sintática. O exemplo clássico de expr. aritméticas utilizado extensivamente pode ser utilizado como inspiração.
- As entradas abaixo estão corretas (de acordo com a E2). Verifique porque sua gramática considera eles falhos e faça as alterações para aceitá-los como sintaticamente corretos.
  - Alguns testes são considerados errados de maneira surpreendente, como o asl058, onde temos uma simples declaração de variável reguida de um comando de return.

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoA") %>%
    filter(Result == 1, Esperado == 0) %>%
    pull(Teste)
```

```
[1] "asl031" "asl032" "asl041" "asl063" "asl077" "asl080" "asl157" "asl158"
[9] "asl162" "asl189" "asl192"
```

1.2.3 GrupoB

- Na E2, o binário executável deve se chamar `etapa2`, conforme regras gerais.
- As entradas abaixo estão corretas (de acordo com a E2). Verifique porque sua gramática considera eles falhos e faça as alterações para aceitá-los como sintaticamente corretos.
  - Neste caso basta forçar que blocos de comandos são comandos simples.

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoB") %>%
    filter(Result == 1, Esperado == 0) %>%
    pull(Teste)
```

```
[1] "asl031" "asl032"
```

- As entradas abaixo estão incorretas (de acordo com a E2). Verifique porque sua gramática considera eles corretos e faça as alterações para não aceitá-los como sintaticamente corretos.
  - Literais booleanos não fazem parte de expressões
  - Estranhamente há uma dualidade entre asl180 e asl031/asl032
    - Se asl031/032 são considerados errados (mas deveriam ser considerados corretos), porque asl180 não é considerado certo (embora devesse ser considerado errado)?

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoB") %>%
    filter(!(Result == 1 & Esperado == 0)) %>%
    pull(Teste)
```

```
[1] "asl160" "asl161" "asl167" "asl180" "asl183" "asl184"
```

1.2.4 GrupoC

- As entradas abaixo estão corretas (de acordo com a E2). Verifique porque sua gramática considera eles falhos e faça as alterações para aceitá-los como sintaticamente corretos.

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoC") %>%
    filter(Result == 1, Esperado == 0) %>%
    pull(Teste)
```

```
[1] "asl063" "asl149" "asl150" "asl157" "asl158" "asl162" "asl166" "asl189"
[9] "asl192"
```

- As entradas abaixo estão incorretas (de acordo com a E2). Verifique porque sua gramática considera eles corretos e faça as alterações para não aceitá-los como sintaticamente corretos.
  - Valor do vetor na declaração não é positivo

- Literais booleanos não fazem parte de expressões

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE
    filter(Grupo == "GrupoC") %>%
    filter(!(Result == 1 & Esperado == 0)) %>%
    pull(Teste)
```

```
[1] "asl159" "asl163" "asl164" "asl167" "asl171" "asl183" "asl184"
```

1.2.5 GrupoD

- Submetido em atraso. Política de atraso ativa.
- Como é garantida a precedência de operações aritméticas? Veja que a regra `operador_binario` faz com que todas os operadores ali listadas estejam no mesmo nível de precedência, o que contradiz a especificação pois a soma binária deve ter menos precedência que a multiplicação, por exemplo. Aqui, isso é inócuo (os testes passam pois gramaticalmente está correto), mas nas próximas etapas os testes falharão. Sugiro revisar tomando-se por base os exemplos dados nas primeiras aulas teóricas de análise sintática. O exemplo clássico de expr. aritméticas utilizado extensivamente pode ser utilizado como inspiração.
- As entradas abaixo estão corretas (de acordo com a E2). Verifique porque sua gramática considera eles falhos e faça as alterações para aceitá-los como sintaticamente corretos.
  - Operador '?' não implementado? Exige alteração no scanner.l!
  - Operador ternário?
  - Lista de declaração de variáveis com inicialização.
  - Operador asterisco como operador associativo à direita

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoD") %>%
    filter(Result == 1, Esperado == 0) %>%
    pull(Teste)
```

```
[1] "asl143" "asl150" "asl153" "asl154" "asl177" "asl188"
```

- As entradas abaixo estão incorretas (de acordo com a E2). Verifique porque sua gramática considera eles corretos e faça as alterações para não aceitá-los como sintaticamente corretos.
  - Muito estranho o asl116 pois ali temos uma inicialização e não uma atribuição, como que é aceito? O problema é que `expressao` foi considerada como um `comando_simples`, contradizendo o que está escrito na E2.
  - Sugiro gentilmente de fazer todas as correções sugeridas sob o risco das próximas etapas serem bastante problemáticas.

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE
    filter(Grupo == "GrupoD") %>%
    filter(!(Result == 1 & Esperado == 0)) %>%
    pull(Teste)
```

```
[1] "asl116" "asl134" "asl159" "asl160" "asl161" "asl163" "asl164" "asl167"
[9] "asl171" "asl183" "asl184"
```

1.2.6 GrupoE

- As entradas abaixo estão corretas (de acordo com a E2). Verifique porque sua gramática considera eles falhos e faça as alterações para aceitá-los como sintaticamente corretos.
  - static pode ser empregado nas variáveis globais também (asl176), como o que parece ter sido bem implementado na regra `declaracaoVariavelGlobal`, mas porque esse teste é considerado errado então?

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALS
    filter(Grupo == "GrupoE") %>%
    filter(Result == 1, Esperado == 0) %>%
    pull(Teste)
```

```
[1] "asl176"
```

- As entradas abaixo estão incorretas (de acordo com a E2). Verifique porque sua gramática considera eles corretos e faça as alterações para não aceitá-los como sintaticamente corretos.
  - Literais booleanos não fazem parte de expressões

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoE") %>%
    filter(!(Result == 1 & Esperado == 0)) %>%
    pull(Teste)
```

```
[1] "asl160" "asl161" "asl163" "asl164" "asl167" "asl183" "asl184"
```

1.2.7 GrupoF

- As entradas abaixo estão corretas (de acordo com a E2). Verifique porque sua gramática considera eles falhos e faça as alterações para aceitá-los como sintaticamente corretos.
  - O static pode aparecer também nas variáveis locais (asl035, asl042, ...)
  - Todos os comandos simples são terminados por ponto e vírgula, inclusive o if, if/else, for, etc.
  - Alguns operadores binários não implementados (barra vertical)
  - Operador ternário?

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoF") %>%
    filter(Result == 1, Esperado == 0) %>%
    pull(Teste)
```

```
[1] "asl035" "asl042" "asl062" "asl063" "asl065" "asl066" "asl080" "asl129"
[9] "asl147" "asl150" "asl168" "asl177" "asl192"
```

- As entradas abaixo estão incorretas (de acordo com a E2). Verifique porque sua gramática considera eles corretos e faça as alterações para não aceitá-los como sintaticamente corretos.
  - Literais booleanos não fazem parte de expressões
  - Muito estranho o asl116 pois ali temos uma inicialização e não uma atribuição, como que é aceito? O problema é que `expressao` foi considerada como um `comando_simples`, contradizendo o que está escrito na E2.

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoF") %>%
    filter(!(Result == 1 & Esperado == 0)) %>%
    pull(Teste)
```

```
[1] "asl116" "asl159" "asl160" "asl161" "asl171" "asl183" "asl184"
```

1.2.8 GrupoG

- Colocar nome nos arquivos.
- Todas as entradas são aceitas, embora muitas estejam erradas.
  - ≈40% das entradas fornecidas estão corretas.
- Isso provavelmente está acontecendo pois vocês adicionaram uma função main no parser.y com este conteúdo
  - O fato de vocês sempre retornarem 0 implica que do ponto de vista dos testes automáticos qualquer entrada é aceita. Vocês deviam usar o main.c fornecido na especificação.

```
int main(int argc, char **argv){
  int ret = yyparse();
  if (ret != 0){
      fprintf(stderr, "%d error found.\n", ret);
  }
  return 0;
}
```

- Então, o professor executou os seguinte comandos para fazer uma avaliação obj/sub que reflita melhor o que foi feito do ponto de vista da E2. Peço somente que nas próximas etapas vocês tenham uma atenção maior.

```
head -n-8 parser.y > parser.y.n
mv parser.y.n parser.y
sed -i 's/\(parser.tab.c\)$/\1 main.c/' makefile
sed -i 's/\(parser.tab.o\)/\1 main.o/' makefile
```

- Mesmo após este esforço, a solução ainda encontra-se descolada da especificação. Vejamos detalhes abaixo.
- Todas as entradas corretas são reconhecidas como corretas.

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoG") %>%
    filter(Result == 1, Esperado == 0) %>%
    pull(Teste)
```

```
character(0)
```

- No entanto, as entradas abaixo estão incorretas (de acordo com a E2). Verifique porque sua gramática considera eles corretos e faça as alterações para não aceitá-los como sintaticamente corretos.
  - Corrigir todos.

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoG") %>%
    filter(!(Result == 1 & Esperado == 0)) %>%
    pull(Teste)
```

```
  [1] "asl009" "asl010" "asl011" "asl012" "asl013" "asl014" "asl015" "asl016"
  [9] "asl017" "asl018" "asl019" "asl020" "asl021" "asl022" "asl023" "asl024"
 [17] "asl026" "asl029" "asl034" "asl040" "asl045" "asl049" "asl052" "asl055"
 [25] "asl061" "asl064" "asl067" "asl068" "asl069" "asl070" "asl071" "asl072"
 [33] "asl073" "asl074" "asl075" "asl082" "asl083" "asl084" "asl085" "asl086"
 [41] "asl087" "asl088" "asl089" "asl090" "asl091" "asl092" "asl093" "asl094"
 [49] "asl095" "asl096" "asl097" "asl098" "asl099" "asl100" "asl101" "asl102"
 [57] "asl104" "asl105" "asl106" "asl107" "asl108" "asl109" "asl110" "asl111"
 [65] "asl112" "asl113" "asl115" "asl116" "asl117" "asl118" "asl119" "asl121"
 [73] "asl122" "asl123" "asl124" "asl125" "asl126" "asl127" "asl128" "asl130"
 [81] "asl131" "asl132" "asl133" "asl134" "asl135" "asl136" "asl137" "asl152"
 [89] "asl155" "asl156" "asl159" "asl160" "asl161" "asl163" "asl164" "asl165"
 [97] "asl167" "asl169" "asl170" "asl171" "asl174" "asl175" "asl178" "asl179"
[105] "asl180" "asl181" "asl182" "asl183" "asl184" "asl185" "asl186" "asl191"
```

1.2.9 GrupoH

- As entradas abaixo estão corretas (de acordo com a E2). Verifique porque sua gramática considera eles falhos e faça as alterações para aceitá-los como sintaticamente corretos.
  - Ponto e vírgula depois de bloco de comando, bloco de comando é comando simples.
  - O operador unário de "ponteiro" pode ser recursivo (expressões são recursivas, uma expressão pode ter outra expressão). O `operand` da regra `op` deveria poder ser uma expressão unária não?

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoH") %>%
    filter(Result == 1, Esperado == 0) %>%
    pull(Teste)
```

```
[1] "asl031" "asl032" "asl188"
```

- As entradas abaixo estão incorretas (de acordo com a E2). Verifique porque sua gramática considera eles corretos e faça as alterações para não aceitá-los como sintaticamente corretos.
  - Literais booleanos não fazem parte de expressões
  - Literal char e literal string não fazem parte de expressões

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoH") %>%
    filter(!(Result == 1 & Esperado == 0)) %>%
    pull(Teste)
```

```
[1] "asl160" "asl161" "asl167" "asl170" "asl179" "asl182" "asl183" "asl184"
[9] "asl185" "asl186"
```

1.2.10 GrupoI

- Na E2, o binário executável deve se chamar `etapa2`, conforme regras gerais.
- As entradas abaixo estão corretas (de acordo com a E2). Verifique porque sua gramática considera eles falhos e faça as alterações para aceitá-los como sintaticamente corretos.
  - Ponto e vírgula depois de bloco de comando, bloco de comando é comando simples.

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoI") %>%
    filter(Result == 1, Esperado == 0) %>%
    pull(Teste)
```

```
[1] "asl031" "asl032"
```

- As entradas abaixo estão incorretas (de acordo com a E2). Verifique porque sua gramática considera eles corretos e faça as alterações para não aceitá-los como sintaticamente corretos.
  - Literais booleanos não fazem parte de expressões

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoI") %>%
    filter(!(Result == 1 & Esperado == 0)) %>%
    pull(Teste)
```

```
[1] "asl160" "asl161" "asl167" "asl180" "asl183" "asl184"
```

1.2.11 GrupoJ

- As entradas abaixo estão corretas (de acordo com a E2). Verifique porque sua gramática considera eles falhos e faça as alterações para aceitá-los como sintaticamente corretos.
  - Ponto e vírgula depois de bloco de comando, bloco de comando é comando simples.
  - Todos os comandos simples são terminados por ponto e vírgula, inclusive o if, if/else, for, etc.
  - No acesso ao vetor, podemos ter uma expressão (asl077)
  - O operador unário de "ponteiro" pode ser recursivo (expressões são recursivas, uma expressão pode ter outra expressão).

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoJ") %>%
    filter(Result == 1, Esperado == 0) %>%
    pull(Teste)
```

```
[1] "asl031" "asl032" "asl062" "asl063" "asl065" "asl066" "asl077" "asl080"
[9] "asl129" "asl168" "asl188" "asl192"
```

- As entradas abaixo estão incorretas (de acordo com a E2). Verifique porque sua gramática considera eles corretos e faça as alterações para não aceitá-los como sintaticamente corretos.
  - Literais booleanos não fazem parte de expressões

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoJ") %>%
    filter(!(Result == 1 & Esperado == 0)) %>%
    pull(Teste)
```

```
[1] "asl160" "asl161" "asl180" "asl183" "asl184"
```

1.2.12 GrupoK

- O programa está imprimindo 0 ou 1 na saída padrão, sugiro remover.
  - Removendo o printf da função main.c
- As entradas abaixo estão corretas (de acordo com a E2). Verifique porque sua gramática considera eles falhos e faça as alterações para aceitá-los como sintaticamente corretos.
  - Todos os comandos simples são terminados por ponto e vírgula, inclusive o if, if/else, for, etc.
  - Ponto e vírgula depois de bloco de comando, bloco de comando é comando simples.
  - Por que o asl187 não é reconhecido como correto?
  - Expressões log/arit podem estar mescladas (asl166)
  - Revisar os testes abaixo individualmente e fazer as correções gramaticais.

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoK") %>%
    filter(Result == 1, Esperado == 0) %>%
    pull(Teste)
```

```
 [1] "asl031" "asl032" "asl048" "asl062" "asl063" "asl065" "asl066" "asl077"
 [9] "asl078" "asl080" "asl129" "asl157" "asl158" "asl162" "asl166" "asl168"
[17] "asl187" "asl189" "asl192"
```

- As entradas abaixo estão incorretas (de acordo com a E2). Verifique porque sua gramática considera eles corretos e faça as alterações para não aceitá-los como sintaticamente corretos.

- Valor do vetor na declaração não é positivo (asl159)
- Não deveria aceitar o asl086 pois há uma vírgula sem parâmetro em seguida. Sugestão: `Parameter_List` não deveria poder derivar para vazio.

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoK") %>%
    filter(!(Result == 1 & Esperado == 0)) %>%
    pull(Teste)
```

```
[1] "asl086" "asl159" "asl180"
```

1.2.13 GrupoL

- As entradas abaixo estão corretas (de acordo com a E2). Verifique porque sua gramática considera eles falhos e faça as alterações para aceitá-los como sintaticamente corretos.
  - Todos os comandos simples são terminados por ponto e vírgula, inclusive o if, if/else, for, etc.
  - Operador '?' não implementado? Exige alteração no scanner.l! (asl143)
  - Operador '%' não implementado? Exige alteração no scanner.l! (asl146)
  - Operador ternário...

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoL") %>%
    filter(Result == 1, Esperado == 0) %>%
    pull(Teste)
```

```
[1] "asl062" "asl063" "asl065" "asl066" "asl080" "asl120" "asl129" "asl143"
[9] "asl146" "asl150" "asl168" "asl192"
```

- As entradas abaixo estão incorretas (de acordo com a E2). Verifique porque sua gramática considera eles corretos e faça as alterações para não aceitá-los como sintaticamente corretos.
  - Valor do vetor na declaração não é positivo (asl159)
  - Literais booleanos não fazem parte de expressões
  - Literal char e literal string não fazem parte de expressões
  - Aceita shift com número negativo (reveja a E2!)

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoL") %>%
    filter(!(Result == 1 & Esperado == 0)) %>%
    pull(Teste)
```

```
[1] "asl159" "asl160" "asl161" "asl170" "asl171" "asl179" "asl182" "asl183"
[9] "asl184" "asl185" "asl186"
```

1.2.14 GrupoM

- As entradas abaixo estão corretas (de acordo com a E2). Verifique porque sua gramática considera eles falhos e faça as alterações para aceitá-los como sintaticamente corretos.
  - Operador '?' não implementado? Exige alteração no scanner.l! (asl143)
  - Operador circunflexo (potência) não implementado? Exige alteração no scanner.l! (asl149)
  - Não aceita o operador ternário? Uma expressão aritmética pode ser uma expressão lógica.

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoM") %>%
    filter(Result == 1, Esperado == 0) %>%
    pull(Teste)
```

```
[1] "asl143" "asl149" "asl150"
```

- As entradas abaixo estão incorretas (de acordo com a E2). Verifique porque sua gramática considera eles corretos e faça as alterações para não aceitá-los como sintaticamente corretos.
  - Todos os comandos simples são terminados por ponto e vírgula, inclusive o if, if/else, for, etc.
  - Literal char e literal string não fazem parte de expressões
  - Literais booleanos não fazem parte de expressões
  - Valor do vetor na declaração não é positivo (asl159)
  - Aceita shift com número negativo (reveja a E2!)

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoM") %>%
    filter(!(Result == 1 & Esperado == 0)) %>%
    pull(Teste)
```

```
[1] "asl159" "asl160" "asl161" "asl163" "asl164" "asl167" "asl170" "asl171"
[9] "asl179" "asl182" "asl183" "asl184" "asl185" "asl186"
```

1.2.15 GrupoN

- As entradas abaixo estão corretas (de acordo com a E2). Verifique porque sua gramática considera eles falhos e faça as alterações para aceitá-los como sintaticamente corretos.
  - Todos os comandos simples são terminados por ponto e vírgula, inclusive o if, if/else, for, etc.
  - Ponto e vírgula depois de bloco de comando, bloco de comando é comando simples.
  - O programa vazio é permitido!
  - Não aceita o operador ternário? Uma expressão aritmética pode ser uma expressão lógica.
  - O operador unário de "ponteiro" pode ser recursivo (expressões são recursivas, uma expressão pode ter outra expressão).

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoN") %>%
    filter(Result == 1, Esperado == 0) %>%
    pull(Teste)
```

```
[1] "asl031" "asl032" "asl062" "asl063" "asl065" "asl066" "asl080" "asl114"
[9] "asl129" "asl150" "asl168" "asl188" "asl192"
```

- As entradas abaixo estão incorretas (de acordo com a E2). Verifique porque sua gramática considera eles corretos e faça as alterações para não aceitá-los como sintaticamente corretos.
  - Literais booleanos não fazem parte de expressões

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoN") %>%
    filter(!(Result == 1 & Esperado == 0)) %>%
    pull(Teste)
```

```
[1] "asl159" "asl160" "asl161" "asl171" "asl183" "asl184"
```

1.2.16 GrupoO

- As entradas abaixo estão corretas (de acordo com a E2). Verifique porque sua gramática considera eles falhos e faça as alterações para aceitá-los como sintaticamente corretos.

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoO") %>%
    filter(Result == 1, Esperado == 0) %>%
    pull(Teste)
```

```
[1] "asl062" "asl063" "asl065" "asl066" "asl129" "asl140" "asl150" "asl157"
[9] "asl158" "asl162" "asl166" "asl189" "asl192"
```

- As entradas abaixo estão incorretas (de acordo com a E2). Verifique porque sua gramática considera eles corretos e faça as alterações para não aceitá-los como sintaticamente corretos.
  - Não deveria aceitar o asl086 pois há uma vírgula sem parâmetro em seguida. Sugestão: `parametros` não deveria poder derivar para vazio.
  - Valor do vetor na declaração não é positivo (asl159)
  - Literais booleanos não fazem parte de expressões
  - Aceita shift com número negativo (reveja a E2!)

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoO") %>%
    filter(!(Result == 1 & Esperado == 0)) %>%
    pull(Teste)
```

```
[1] "asl086" "asl159" "asl163" "asl164" "asl167" "asl171" "asl183" "asl184"
```

1.2.17 GrupoP

- As entradas abaixo estão corretas (de acordo com a E2). Verifique porque sua gramática considera eles falhos e faça as alterações para aceitá-los como sintaticamente corretos.
  - Todos os comandos simples são terminados por ponto e vírgula, inclusive o if, if/else, for, etc.

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoP") %>%
    filter(Result == 1, Esperado == 0) %>%
    pull(Teste)
```

```
[1] "asl062" "asl063" "asl065" "asl066" "asl080" "asl129" "asl168" "asl192"
```

- As entradas abaixo estão incorretas (de acordo com a E2). Verifique porque sua gramática considera eles corretos e faça as alterações para não aceitá-los como sintaticamente corretos.
  - Não deveria aceitar o asl086 pois há uma vírgula sem parâmetro em seguida. Sugestão: `parameter_list` não deveria poder derivar para vazio.
  - Literais booleanos não fazem parte de expressões
  - O static não pode estar em um parâmetro!

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoP") %>%
    filter(!(Result == 1 & Esperado == 0)) %>%
    pull(Teste)
```

```
[1] "asl086" "asl101" "asl160" "asl161" "asl183" "asl184"
```

1.2.18 GrupoQ

- As entradas abaixo estão corretas (de acordo com a E2). Verifique porque sua gramática considera eles falhos e faça as alterações para aceitá-los como sintaticamente corretos.
  - Todos os comandos simples são terminados por ponto e vírgula, inclusive o if, if/else, for, etc.
  - Operador '?' não implementado? Exige alteração no scanner.l! (asl143)
  - Parece que houve um equívoco de interpretação da E2, como podemos ver com as entradas simples tais como asl046, asl047.... Por favor, peço gentilmente que releiam a especificação da E2 e façam as correções necessárias.
  - Revisar os testes abaixo individualmente e fazer as correções gramaticais.

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoQ") %>%
    filter(Result == 1, Esperado == 0) %>%
    pull(Teste)
```

```
 [1] "asl046" "asl047" "asl048" "asl054" "asl058" "asl062" "asl063" "asl065"
 [9] "asl066" "asl077" "asl078" "asl080" "asl103" "asl129" "asl143" "asl150"
[17] "asl162" "asl168" "asl187" "asl188" "asl189" "asl190" "asl192"
```

- As entradas abaixo estão incorretas (de acordo com a E2). Verifique porque sua gramática considera eles corretos e faça as alterações para não aceitá-los como sintaticamente corretos.

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %
    filter(Grupo == "GrupoQ") %>%
    filter(!(Result == 1 & Esperado == 0)) %>%
    pull(Teste)
```

```
character(0)
```

1.2.19 GrupoR

- Nada foi submetido no prazo, nem no prazo estendido.

1.2.20 GrupoS

- Nada foi submetido no prazo, nem no prazo estendido.

1.3 Objetivo

1.3.1 Visão Geral

```
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_objetivo.csv") %>% arrange(Grupo) %>% print -> e2.objetivo
```

Grupo	E2.O
GrupoA	9.43
GrupoB	9.58
GrupoC	9.17
GrupoD	9.11
GrupoE	9.58
GrupoF	8.96
GrupoG	4.17
GrupoH	9.32
GrupoI	9.58
GrupoJ	9.11
GrupoK	8.85
GrupoL	8.8
GrupoM	9.11
GrupoN	9.01
GrupoO	8.91
GrupoP	9.27
GrupoQ	8.8

1.3.2 Quem errou o que

```
options(crayon.enabled=FALSE)
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_quem_errou_o_que.csv", show_col_types = FALSE, progress=FALSE, ) %>%
    sample_n(20)
```

```
# A tibble: 20 × 6
   Grupo  Teste  Result Esperado Correto Razao
   <chr>  <chr>   <dbl>    <dbl> <lgl>   <chr>
 1 GrupoG asl155      0        1 FALSE   1 error found. Line 3: syntax error
 2 GrupoA asl192      1        0 FALSE   line 5: syntax error
 3 GrupoA asl158      1        0 FALSE   line 3: syntax error
 4 GrupoG asl113      0        1 FALSE   1 error found. Line 2: syntax error
 5 GrupoG asl068      0        1 FALSE   1 error found. Line 4: syntax error
 6 GrupoG asl109      0        1 FALSE   1 error found. Line 2: syntax error
 7 GrupoG asl133      0        1 FALSE   1 error found. Line 3: syntax error
 8 GrupoO asl189      1        0 FALSE   Erro de sintaxe na linha 5 : [syntax e...
 9 GrupoG asl084      0        1 FALSE   1 error found. Line 3: syntax error
10 GrupoG asl156      0        1 FALSE   1 error found. Line 3: syntax error
11 GrupoJ asl066      1        0 FALSE   ERROR: line: 7 message: syntax error
12 GrupoA asl031      1        0 FALSE   line 4: syntax error
13 GrupoK asl065      1        0 FALSE   syntax error, unexpected ';', expectin...
14 GrupoG asl022      0        1 FALSE   1 error found. Line 2: syntax error
15 GrupoL asl066      1        0 FALSE   An error was found at line 7: syntax e...
16 GrupoG asl070      0        1 FALSE   1 error found. Line 6: syntax error
17 GrupoJ asl183      0        1 FALSE   <NA>
18 GrupoO asl164      0        1 FALSE   <NA>
19 GrupoG asl009      0        1 FALSE   1 error found. Line 2: syntax error
20 GrupoG asl024      0        1 FALSE   1 error found. Line 2: syntax error
```

1.4 Subjetivo

1.4.1 Tabela com valores dos critérios por grupo

```
suppressMessages(library(tidyverse))
read_csv("e2/e2_output_objetivo.csv", show_col_types = FALSE, progress=FALSE) %>% arrange(Grupo
e2.objetivo %>%
    mutate(Critério = "Gramática",
           Nota = trunc(E2.O)) %>%
    select(-E2.O) %>%
    bind_rows(dados) %>%
    bind_rows(df.e2.opiniao) %>%
    print -> e2.tab.subjetiva
```

Grupo	Critério	Nota
GrupoA	Gramática	9
GrupoB	Gramática	9
GrupoC	Gramática	9
GrupoD	Gramática	9
GrupoE	Gramática	9
GrupoF	Gramática	8
GrupoG	Gramática	4
GrupoH	Gramática	9
GrupoI	Gramática	9
GrupoJ	Gramática	9
GrupoK	Gramática	8
GrupoL	Gramática	8
GrupoM	Gramática	9
GrupoN	Gramática	9
GrupoO	Gramática	8
GrupoP	Gramática	9
GrupoQ	Gramática	8
GrupoA	Relatório de Erro	5
GrupoB	Relatório de Erro	10
GrupoC	Relatório de Erro	10

Continued on next page

Continued from previous page

Grupo	Critério	Nota
GrupoD	Relatório de Erro	5
GrupoE	Relatório de Erro	10
GrupoF	Relatório de Erro	10
GrupoG	Relatório de Erro	nil
GrupoH	Relatório de Erro	10
GrupoI	Relatório de Erro	5
GrupoJ	Relatório de Erro	5
GrupoK	Relatório de Erro	10
GrupoL	Relatório de Erro	10
GrupoM	Relatório de Erro	5
GrupoN	Relatório de Erro	5
GrupoO	Relatório de Erro	5
GrupoP	Relatório de Erro	10
GrupoQ	Relatório de Erro	5
GrupoA	Remoção de Conflitos	10
GrupoB	Remoção de Conflitos	10
GrupoC	Remoção de Conflitos	10
GrupoD	Remoção de Conflitos	10
GrupoE	Remoção de Conflitos	10
GrupoF	Remoção de Conflitos	10
GrupoG	Remoção de Conflitos	10
GrupoH	Remoção de Conflitos	10
GrupoI	Remoção de Conflitos	10
GrupoJ	Remoção de Conflitos	10
GrupoK	Remoção de Conflitos	10
GrupoL	Remoção de Conflitos	10
GrupoM	Remoção de Conflitos	10
GrupoN	Remoção de Conflitos	10
GrupoO	Remoção de Conflitos	10
GrupoP	Remoção de Conflitos	10
GrupoQ	Remoção de Conflitos	10
GrupoA	Opinião	6.6
GrupoB	Opinião	8.5
GrupoC	Opinião	6.4
GrupoD	Opinião	7.1
GrupoE	Opinião	9.7
GrupoF	Opinião	6.8
GrupoG	Opinião	nil
GrupoH	Opinião	9.1
GrupoI	Opinião	8.5
GrupoJ	Opinião	7.8
GrupoK	Opinião	5.8
GrupoL	Opinião	7.1
GrupoM	Opinião	7.8
GrupoN	Opinião	7.4
GrupoO	Opinião	6.9
GrupoP	Opinião	8.7
GrupoQ	Opinião	5.1

1.4.2 Justificativas da tabela

1. Gramática (Sec 2.1)

   Alinha-se com os resultados objetivos, com a função `trunc` (obtém o valor inteiro do ponto-flutuante).

2. Relatório de Erro (Sec 2.2)

   - Nota 10: grupo apresentou solução com "mensagem de erro informando a linha do código da entrada que gerou o erro sintático e informações adicionais que auxiliem o programador que está utilizando o compilamensagem de erro informando a linha do código da"
   - Nota 5: grupo apresentou solução que apenas identifica o número da linha onde aconteceu o erro sintática, sem informar qual foi o erro especificamente
   - Nota 0: grupo apresentou solução que apenas identifica que houve um erro, sem informar o número da linha ou qual foi o erro

3. Remoção de Conflitos (Sec 2.3)

- Nota 10: sem conflitos shift/reduce ou reduce/reduce
- Nota 0: com pelo menos um conflito acima ou `expect` não justificado

4. Opinião

   Baseada na combinação da análise objetiva, com relatório de erro, remoção de conflitos e gramática, além da revisão das regras gramaticais, adequação às regras gerais, e demais comentários ressaltados acima.

1.4.3 Código/Exporte do subjetivo

```
options(crayon.enabled=FALSE)
library(tidyverse)
nota.maxima = peso %>% pull(Peso) %>% sum * 10;
e2.tab.subjetiva %>%
    as_tibble() %>%
    left_join(peso, by = c("Critério")) %>%
    mutate(Nota = ifelse(is.na(Nota), 0, Nota)) %>%
    mutate(Valor.F = Nota * Peso) %>%
    group_by(Grupo) %>%
    summarize(Soma = sum(Valor.F)/nota.maxima*10, .groups="drop") %>%
    arrange(-Soma) %>%
    rename(E2.S = Soma) %>%
    write_csv("e2/e2_output_subjetivo.csv") %>%
    print(n=100) -> e2.subjetivo
```

Grupo	E2.S
GrupoE	9.58
GrupoH	9.34
GrupoP	9.18
GrupoB	9.1
GrupoC	8.26
GrupoL	8.24
GrupoF	8.12
GrupoI	8.1
GrupoJ	7.82
GrupoM	7.82
GrupoK	7.72
GrupoN	7.66
GrupoD	7.54
GrupoA	7.34
GrupoO	7.16
GrupoQ	6.44
GrupoG	2.2

1.5 PESOS

```
e2.objetivo %>%
    select(Grupo) %>%
    mutate(E2.P = case_when(Grupo == "GrupoD" ~ 0.8,
                            TRUE ~ 1.0)) %>%
    print -> e2.peso
```

```
# A tibble: 17 × 2
   Grupo   E2.P
   <chr>  <dbl>
 1 GrupoA   1
 2 GrupoB   1
 3 GrupoC   1
 4 GrupoD   0.8
 5 GrupoE   1
 6 GrupoF   1
 7 GrupoG   1
 8 GrupoH   1
 9 GrupoI   1
10 GrupoJ   1
11 GrupoK   1
12 GrupoL   1
13 GrupoM   1
```

```
14 GrupoN   1
15 GrupoO   1
16 GrupoP   1
17 GrupoQ   1
```

1.6 Final

```
e2.objetivo %>%
    mutate(Etapa = "E2") %>%
    left_join(e2.subjetivo, by="Grupo") %>%
    left_join(e2.peso, by="Grupo") %>%
    select(Grupo, Etapa, everything()) %>%
    mutate(E2 = round((E2.O + E2.S)/2, 2)) %>%
    print -> fin.group.e2
```

Grupo	Etapa	E2.O	E2.S	E2.P	E2
GrupoA	E2	9.43	7.34	1	8.38
GrupoB	E2	9.58	9.1	1	9.34
GrupoC	E2	9.17	8.26	1	8.71
GrupoD	E2	9.11	7.54	0.8	8.32
GrupoE	E2	9.58	9.58	1	9.58
GrupoF	E2	8.96	8.12	1	8.54
GrupoG	E2	4.17	2.2	1	3.18
GrupoH	E2	9.32	9.34	1	9.33
GrupoI	E2	9.58	8.1	1	8.84
GrupoJ	E2	9.11	7.82	1	8.46
GrupoK	E2	8.85	7.72	1	8.29
GrupoL	E2	8.8	8.24	1	8.52
GrupoM	E2	9.11	7.82	1	8.46
GrupoN	E2	9.01	7.66	1	8.33
GrupoO	E2	8.91	7.16	1	8.04
GrupoP	E2	9.27	9.18	1	9.22
GrupoQ	E2	8.8	6.44	1	7.62

Grupos em recuperação da E2:

- Em Moodle já configurado para tal, use o novo link "Recuperação E2"
- Política de recuperação ativada (vejam regras gerais)

```
[1] "GrupoG"
```