File Edit Tabs Help

```
~/compiler/class/3 $ n badsyntax_2.cc
[/home/justin/compiler/class/3] (master)
~/compiler/class/3 $ n badsyntax_1.cc
[/home/justin/compiler/class/3] (master)
~/compiler/class/3 $ make
happy -gca ParCPP.y
unused terminals: 1
alex -g LexCPP.x
ghc --make TestCPP.hs -o TestCPP
[1 of 7] Compiling AbsCPP           ( AbsCPP.hs, AbsCPP.o )
[2 of 7] Compiling ErrM             ( ErrM.hs, ErrM.o )
[3 of 7] Compiling LexCPP           ( LexCPP.hs, LexCPP.o )
[4 of 7] Compiling ParCPP           ( ParCPP.hs, ParCPP.o )
[5 of 7] Compiling PrintCPP         ( PrintCPP.hs, PrintCPP.o )
[6 of 7] Compiling SkelCPP          ( SkelCPP.hs, SkelCPP.o )
[7 of 7] Compiling Main             ( TestCPP.hs, TestCPP.o )
Linking TestCPP ...
[/home/justin/compiler/class/3] (master)
~/compiler/class/3 $ ./TestCPP good.cc
good.cc

Parse Successful!

[Abstract Syntax]

PDefs [DFun Type_int (Id "factr") [ADecl Type_int (Id "n")] [SIfElse (ELt (EId (Id "n")) (EInt 2)) (SReturn (EInt 1)) (
SReturn (ETimes (EId (Id "n")) (EApp (Id "factr") [EMinus (EId (Id "n")) (EInt 1)])))],DFun Type_int (Id "main") [] [SI
nit Type_int (Id "i") (EApp (Id "factr") [EInt 7]),SReturn (EInt 0)]]

[Linearized tree]

int factr (int n){
  if (n < 2)return 1 ;
  else return n * factr (n - 1);
  }
int main () {
  int i = factr (7);
  return 0 ;
  }

[/home/justin/compiler/class/3] (master)
~/compiler/class/3 $ ./TestCPP bad.cc
bad.cc

Parse Successful!

[Abstract Syntax]

PDefs [DFun Type_int (Id "f") [ADecl Type_double (Id "c")] [SInit Type_int (Id "n") (EInt 1),SWhile (EId (Id "c")) (SEx
p (EIncr (EId (Id "n"))))]]

[Linearized tree]

int f (double c){
  int n = 1 ;
  while (c)++ n ;
  }

[/home/justin/compiler/class/3] (master)
~/compiler/class/3 $
```

```
[/home/justin/compiler/class/3] (master)
~/compiler/class/3 $ scrot -u
```

File Edit Tabs Help

```
int main () {
  int i = factr (7);
  return 0 ;
  }

[/home/justin/compiler/class/3] (master)
~/compiler/class/3 $ ./TestCPP bad.cc
bad.cc

Parse Successful!

[Abstract Syntax]

PDefs [DFun Type_int (Id "f") [ADecl Type_double (Id "c")] [SInit Type_int (Id "n") (EInt 1),SWhile (EId (Id "c")) (SEx
p (EIncr (EId (Id "n"))))]]

[Linearized tree]

int f (double c){
  int n = 1 ;
  while (c)++ n ;
  }

[/home/justin/compiler/class/3] (master)
~/compiler/class/3 $ ./TestCPP bad
bad.cc          badsyntax_1.cc  badsyntax_2.cc
[/home/justin/compiler/class/3] (master)
~/compiler/class/3 $ ./TestCPP badsyntax_1.cc
badsyntax_1.cc

Parse Successful!

[Abstract Syntax]

PDefs [DFun Type_int (Id "f") [ADecl Type_double (Id "c")] [SInit Type_int (Id "n") (EInt 1),SWhile (EId (Id "c")) (SEx
p (EIncr (EId (Id "n"))))]]

[Linearized tree]

int f (double c){
  int n = 1 ;
  while (c)++ n ;
  }

[/home/justin/compiler/class/3] (master)
~/compiler/class/3 $ ./TestCPP badsyntax_2.cc
badsyntax_2.cc

Parse              Failed...

Tokens:
[PT (Pn 150 4 1) (TS "int" 25),PT (Pn 154 4 5) (T_Id "f"),PT (Pn 156 4 7) (TS "(" 3),PT (Pn 157 4 8) (TS "double" 21),P
T (Pn 164 4 15) (T_Id "c"),PT (Pn 165 4 16) (TS "," 8),PT (Pn 166 4 17) (TS "," 8),PT (Pn 167 4 18) (TS ")" 4),PT (Pn 1
69 5 1) (TS "{" 31),PT (Pn 172 6 2) (TS "int" 25),PT (Pn 176 6 6) (T_Id "n"),PT (Pn 178 6 8) (TS "=" 16),PT (Pn 180 6 1
0) (TI "1"),PT (Pn 182 6 12) (TS ";" 13),PT (Pn 185 7 2) (TS "while" 30),PT (Pn 190 7 7) (TS "(" 3),PT (Pn 191 7 8) (T_
Id "c"),PT (Pn 192 7 9) (TS ")" 4),PT (Pn 194 7 11) (TS "++" 7),PT (Pn 196 7 13) (T_Id "n"),PT (Pn 198 7 15) (TS ";" 13
),PT (Pn 200 8 1) (TS "}" 33)]
syntax error at line 4 before , ) { int
[/home/justin/compiler/class/3] (master)
~/compiler/class/3 $
```

```
[/home/justin/compiler/class/3] (master)
~/compiler/class/3 $ scrot -u
[/home/justin/compiler/class/3] (master)
~/compiler/class/3 $ scrot -u
```

```
0 '' '' 1 bash ''                                   '|' 12:00 '|' 27 Feb '' justin '' raspberrypi
```

File Edit Tabs Help

```
~/compiler/class/3 $ n explanation.txt
[/home/justin/compiler/class/3] (master)
~/compiler/class/3 $ clear
[/home/justin/compiler/class/3] (master)
~/compiler/class/3 $ scrot -u
[/home/justin/compiler/class/3] (master)
~/compiler/class/3 $ scrot -u
[/home/justin/compiler/class/3] (master)
~/compiler/class/3 $ n
2020-02-27-120016_1920x1016_scrot.png  ErrM.hs                                 ParCPP.y
2020-02-27-120046_1920x1016_scrot.png  ErrM.o                                  PrintCPP.hi
AbsCPP.hi                              good.cc                                 PrintCPP.hs
AbsCPP.hs                              LexCPP.hi                               PrintCPP.o
AbsCPP.o                               LexCPP.hs                               SkelCPP.hi
bad.cc                                 LexCPP.o                                SkelCPP.hs
badsyntax_1.cc                         LexCPP.x                                SkelCPP.o
badsyntax_2.cc                         Makefile                                TestCPP
CPP.cf                                 ParCPP.hi                               TestCPP.hi
DocCPP.txt                             ParCPP.hs                               TestCPP.hs
ErrM.hi                                ParCPP.o                                TestCPP.o
[/home/justin/compiler/class/3] (master)
~/compiler/class/3 $ n explanation.txt
[/home/justin/compiler/class/3] (master)
~/compiler/class/3 $ cat explanation.txt
good.cc was correct syntax.
bad.cc has a type error, and thus cannot be checked to be incorrect at parsing.
badsyntax_1.cc has to do with a bug in LBNF's terminators and separators. It seems to be explained as "can be considered as a bug"
badsyntax_2.cc has much to do with the same as badsyntax_1.cc, but it fails the parser. It has a double comma, and fails.

This bug has to do with "accepting a list terminating with a " comma, for multiple arguments in the case of this grammar.

[/home/justin/compiler/class/3] (master)
~/compiler/class/3 $ scrot -u
```