Extensions

EXTENDED GRAMMAR

```
MAIN> ::= "{" <INSTRCTLST>
<INSTRCTLST> ::= <INSTRUCTION><INSTRCTLST> |
      "}"
<INSTRUCTION> ::= <FD> |
      <LT> |
      <RT> |
      <DO> |
      <SET>
<FD> ::= "FD" <VARNUM>
<LT> ::= "LT" <VARNUM>
<RT> ::= "RT" <VARNUM>
<DO> ::= "DO" <VAR> "FROM" <VARNUM> "TO"
      <VARNUM> "{" <INSTRCTLST>
<VAR> ::= [A-Z]
<VARNUM> ::= number |
      <VAR>
<SET> ::= "SET" <VAR> ":=" <POLISH>
<POLISH> ::= <OP> <POLISH> |
      <VARNUM> <POLISH> |
      ";"
<OP> ::= "+" |
      "-" |
      "*" |
      "/"
<SET_DRAW> ::= “SET_DRAW” <VARNUM> <VARNUM>
<IF> ::= “(“ <VARNUM> <COMPARATOR> <VARNUM> “)”
      “{“
<COMPARATOR> “<” || “>” || “==”
```

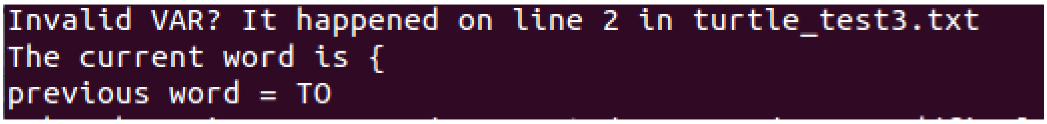
IF statements

The grammar has been extended to include basic IF statements. It can evaluate a statement in the form 'A comparator B' and only carry out a section of code if the statement is true. It is not capable of handling nested if statements however.

SET_DRAW

Set draw allows the draw point to be reset to allow drawing pictures that do not have connecting lines.

Detailed Error Reporting



```
Invalid VAR? It happened on line 2 in turtle_test3.txt
The current word is {
previous word = TO
```

Error reporting is line and file specific within the turtle file being run. This is done using a unique line_id for each word in the program.