How Do You Test Your Compiler? Here's How I Test Mine

LangDev — 9 March 2018

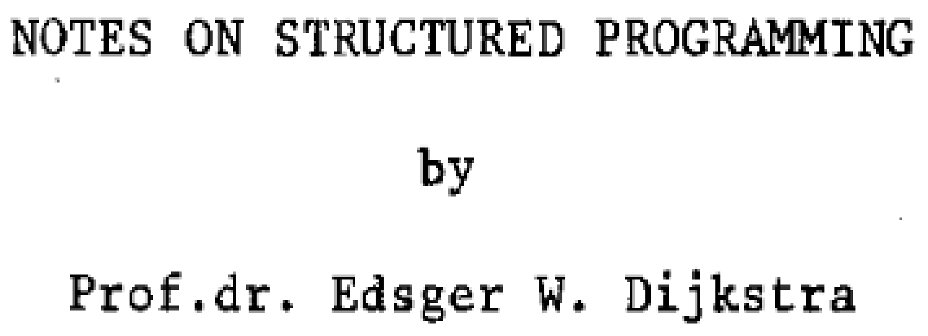
Dr. Vadim Zaytsev aka @grammarware

raincode

LABS

compiler experts

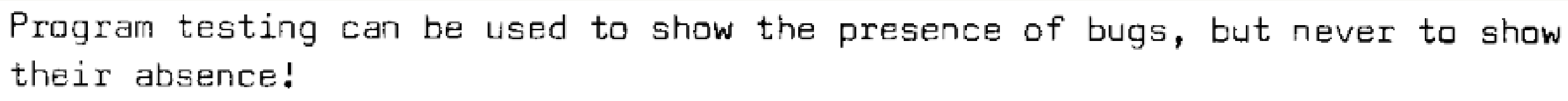
Dijkstra vs Goodenough



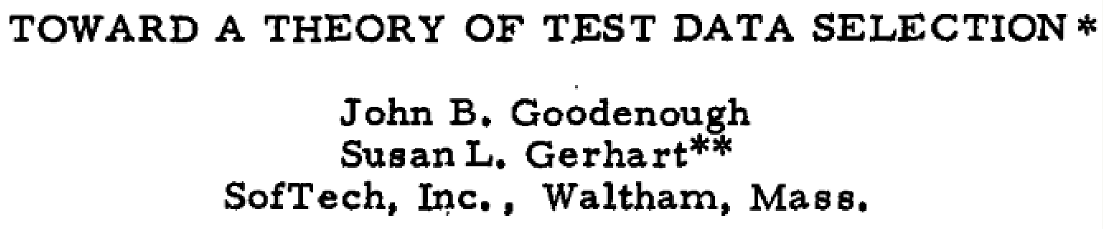
NOTES ON STRUCTURED PROGRAMMING

by

Prof.dr. Edsger W. Dijkstra

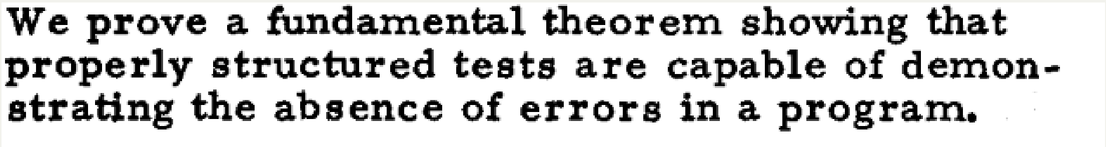


Program testing can be used to show the presence of bugs, but never to show their absence!

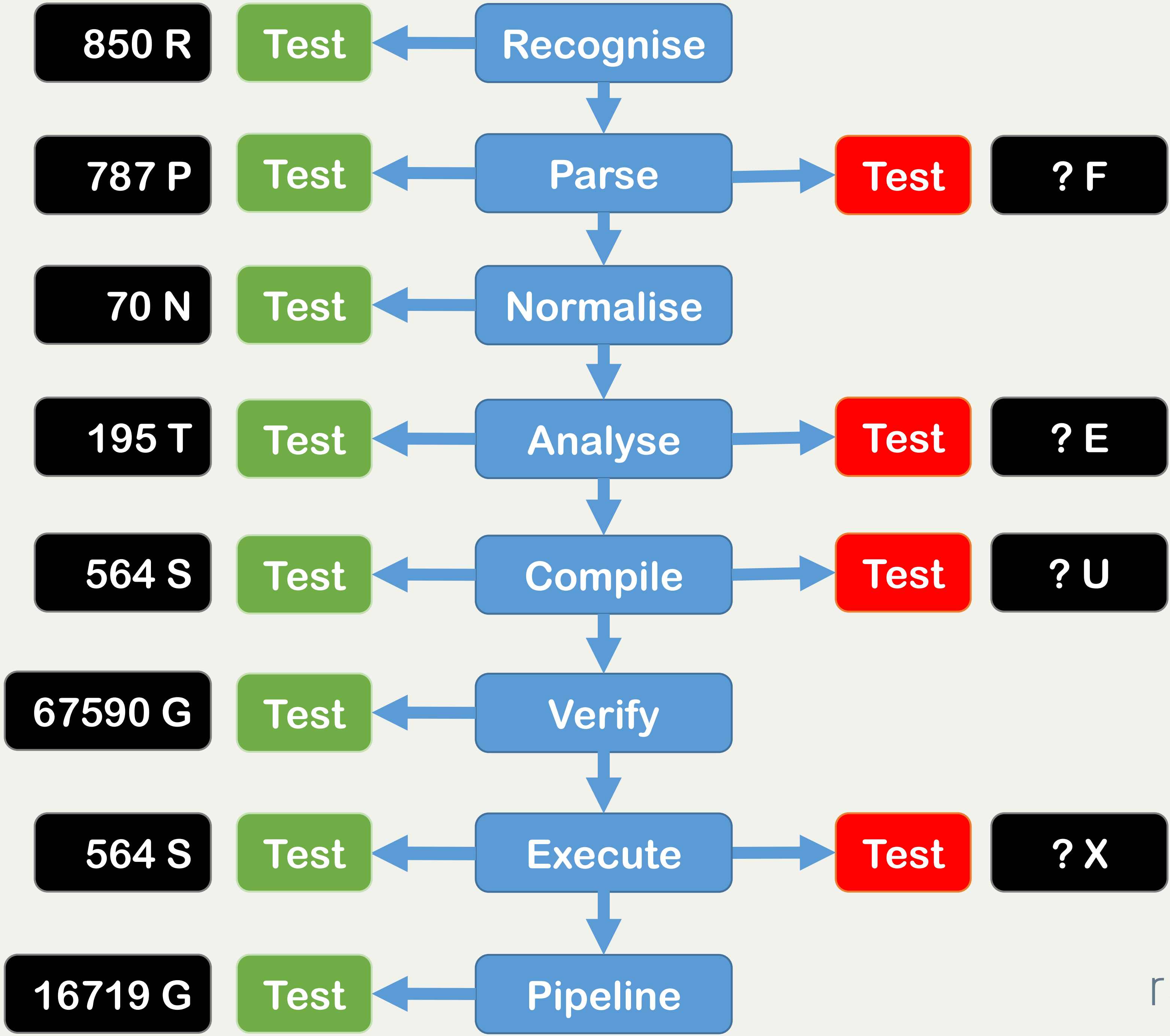


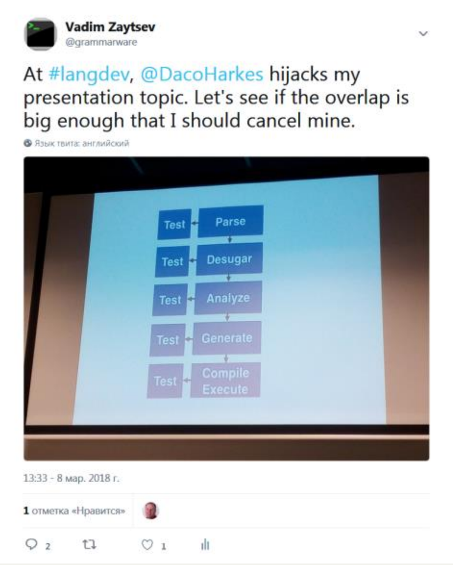
TOWARD A THEORY OF TEST DATA SELECTION *

John B. Goodenough
Susan L. Gerhart**
SofTech, Inc., Waltham, Mass.



We prove a fundamental theorem showing that properly structured tests are capable of demonstrating the absence of errors in a program.





Vadim Zaytsev
@grammarware

At #langdev, @DacoHarkes hijacks my presentation topic. Let's see if the overlap is big enough that I should cancel mine.

Язык твита: английский

13:33 - 8 мар. 2018 г.

1 отметка «Нравится»

2 1

Testing in Castle

- G-tests: can the compiler handle the customer's codebase?
- R-tests: can the parser recognise this input?
- F-tests: can the parser rightfully reject this input?
- P-tests: can the parser construct a good tree from this input?
- N-tests: can the normaliser rewrite this tree well?
- E-tests: can this input error be fixed automatically?
- T-tests: can this program be typed correctly?
- A-tests: can this program be rejected by static semantic analysis?
- C-tests: can this program be successfully compiled to produce a DLL?
- V-tests: can this program be compiled to a verified DLL?
- U-tests: can this problem be rightfully rejected during compilation?
- S-tests: can this program successfully execute to produce output?
- X-tests: can this program throw the right exception?
- D-tests: does this runtime library function work?

Conclusion

- Testing a compiler is a lot of work
- No out of the box solution
- No out of the box comprehensive methodology
- Existing papers are scarce and focused
- Follow @grammarware and attend SLEBoK at SPLASH’18