

PEP Scala (5)

Email: christian.urban at kcl.ac.uk
Office: N7.07 (North Wing, Bush House)
Slides & Code: KEATS
Office Hours: Mondays 12:00 – 14:00

Marks for CW7 (Part 1 + 2)

Raw marks (234 submissions):

- 6%: 192 students
- 5%: 16
- 4%: 7
- 3%: 2
- 2%: 6
- 1%: 1
- 0%: 9

```
def get_csv_url(url: String) : List[String] = {  
    val csv = Try(Source.fromURL(url)).getOrElse(null)  
    if (csv == null){  
        List()  
    }  
    else {  
        ....  
    }  
}
```

```
def get_csv_url(url: String) : List[String] = {  
  val csv = Try(Source.fromURL(url)).getOrElse(null)  
  if (csv == null){  
    List()  
  }  
  else {  
    ....  
  }  
}
```

```
def get_csv_url(url: String) : List[String] = {  
  Try(Source.fromURL(url)....).getOrElse(Nil)  
}
```

```
def get_csv_url(url: String) : List[String] = {  
  try {  
    val csvFile = Source.fromURL(url)  
    ....  
  } catch {  
    case unknown : Throwable => List()  
  }  
}
```

```
def get_csv_url(url: String) : List[String] = {  
  Try(Source.fromURL(url)....).getOrElse Nil  
}
```

Dijkstra on Testing

“Program testing can be a very effective way to show the presence of bugs, but it is hopelessly inadequate for showing their absence.”

Proving Programs to be Correct

Theorem: There are infinitely many prime numbers.

Proof ...

similarly

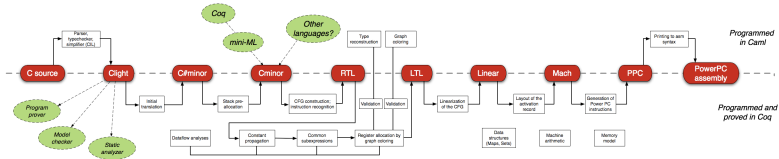
Theorem: The program is doing what it is supposed to be doing.

Long, long proof ...

This can be a gigantic proof. The only hope is to have help from the computer. 'Program' is here to be understood to be quite general (compiler, OS, ...).

Can This Be Done?

- in 2011, verification of a small C-compiler (CompCert)
 - “if my input program has a certain behaviour, then the compiled machine code has the same behaviour”
 - is as good as gcc -O1, but much, much less buggy



Fuzzy Testing C-Compilers

- tested GCC, LLVM and others by randomly generating C-programs
- found more than 300 bugs in GCC and also many in LLVM (some of them highest-level critical)
- about CompCert:

“The striking thing about our CompCert results is that the middle-end bugs we found in all other compilers are absent. As of early 2011, the under-development version of CompCert is the only compiler we have tested for which Csmith cannot find wrong-code errors. This is not for lack of trying: we have devoted about six CPU-years to the task.”

seL4 / Isabelle

- verified a microkernel operating system (≈ 8000 lines of C code)
- US DoD has competitions to hack into drones; they found that the isolation guarantees of seL4 hold up
- CompCert and seL4 sell their code

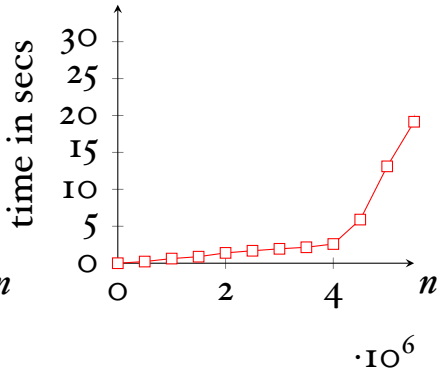
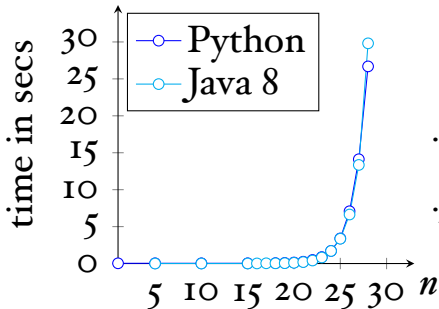
seL4 / Isabelle

- verified a microkernel
lines of C code
- US DoD has contracted
they found that
hold up
- CompCert and seL4 sell their code



CW9 : Regexes

Graphs: $(a^*)^*b$ and strings $\underbrace{a \dots a}_n$

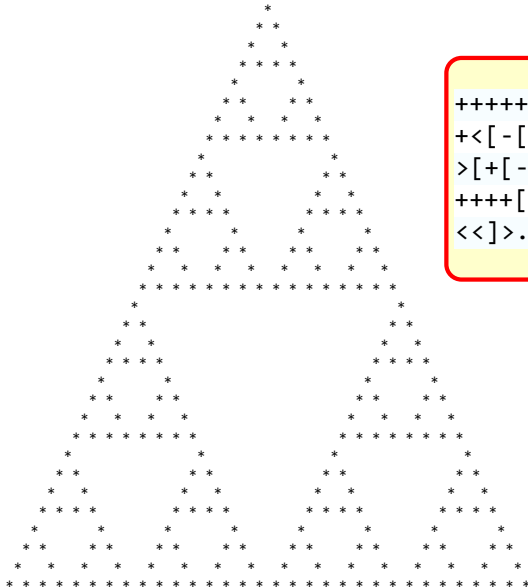


Kuklewicz: most POSIX matchers are buggy
http://www.haskell.org/haskellwiki/Regex_Posix

Where to go on from here?

- Martin Odersky (EPFL)...he is currently throwing out everything and starts again with the dotty compiler for Scala
- Elm (<http://elm-lang.org>)...web applications with style
- Haskell, Ocaml, Standard ML, Scheme, ...

Questions?



```
+++++++ [ >+>++++< <- ]>+>>  
+< [ - [ >>+< <- ]+>> ]>+ [ -<<< [ -  
> [ + [ - ]+>+>>>> -<< ]< [ < ]>>+>  
++++ [ <<++++>> - ]+<<+> . [ - ]  
<< ]> . >+ [ >> ]>+ ]
```

Marks for CW6 (Part 1 + 2)

Raw marks:

- 6%: 154 students
- 5%: 66
- 4%: 18
- 3%: 13
- 2%: 2
- 1%: 1
- 0%: 21