Semantics of Programming Languages

Vesal Vojdani
Technische Universität München

Programming Language Research

- Goals is to improve how programs are developed!
- For example, how can we assist engineers to take advantage of multi-core platforms?

The three layers ...

- Change the Paradigm
  Develop novel programming languages with concurrency in mind.
- Automate the parallelization
  Compiler automagically parallelizes your legacy code.
- Quality ensurance tools
  Assist programmers avoid common concurrency pitfalls.

1) Language Design

- Tends to be very academic, because
- Designing languages requires rigorous understanding of concepts
  - What does concurrent execution mean?
- Interesting examples for concurrency
  - STM Monad in Haskell
  - Polyphonic C# (now Cω)
  - …

2) Compiler Research

- Moore: 2x computing power each 18 months
- Proebstring: 2x compiler gains per 18 years

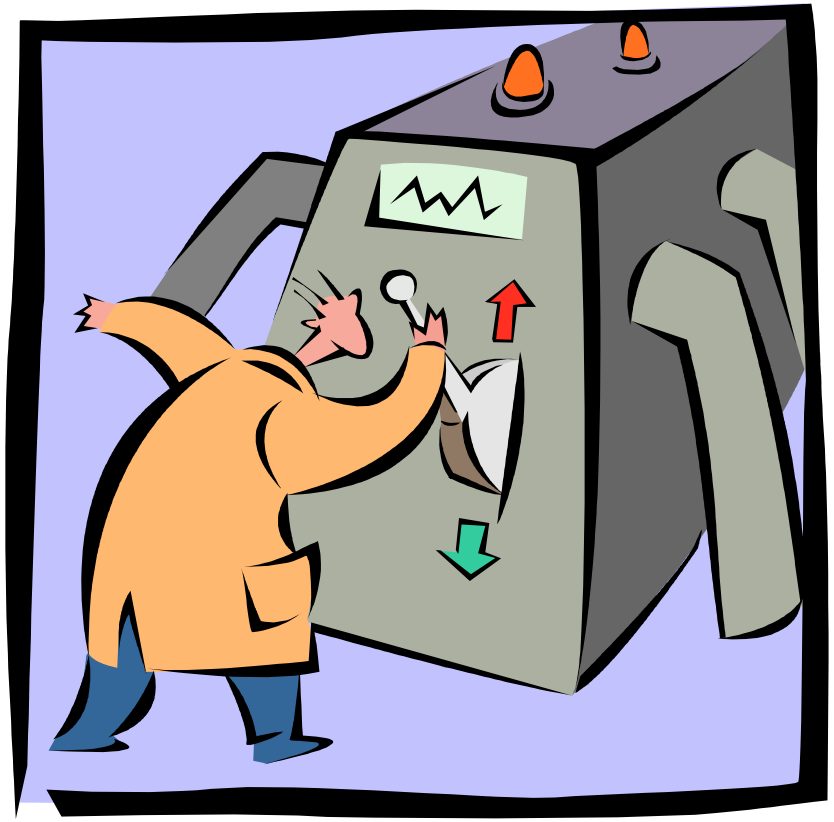
- Is compiler research a waste of time?

The Real Benefit ...

- Getting decent performance for Higher-Level abstractions
- Once exotic features of functional languages:
  - Garbage collection
  - Generics (universal polymorphism)
  - Map/Reduce
  - Monads (LINQ)

3) Static Analysis

- Goal:
  Build a machine to ensure software quality
- Example:
  Static Data Race Detection
- This is what I do!



A data race for `int loser`

```
void *t1 (void *arg) {
    loser = 200;
}
```

```
void *t2 (void *arg) {
    loser = 1;
}
```

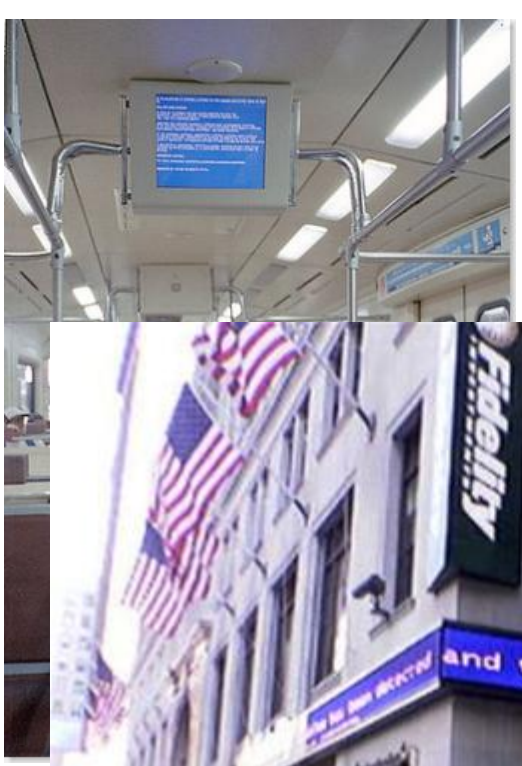
WHY

Analysis?

Murphy's Law

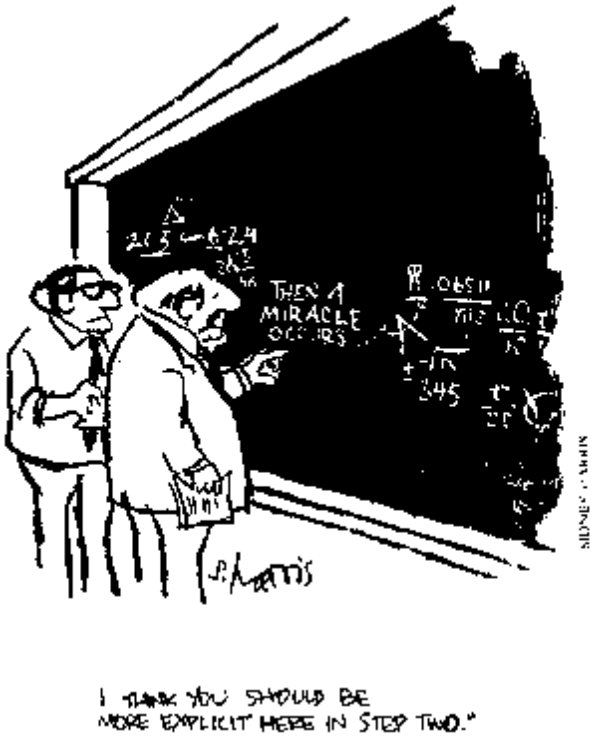
fails

Device Drivers run in Kernel mode

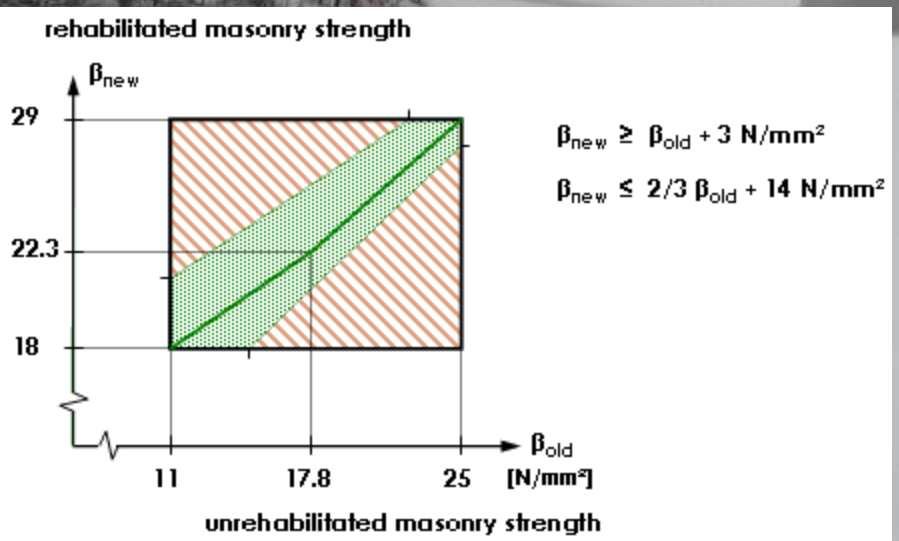


The Right Thing™

Prove the absence of bugs



Like structural engineers

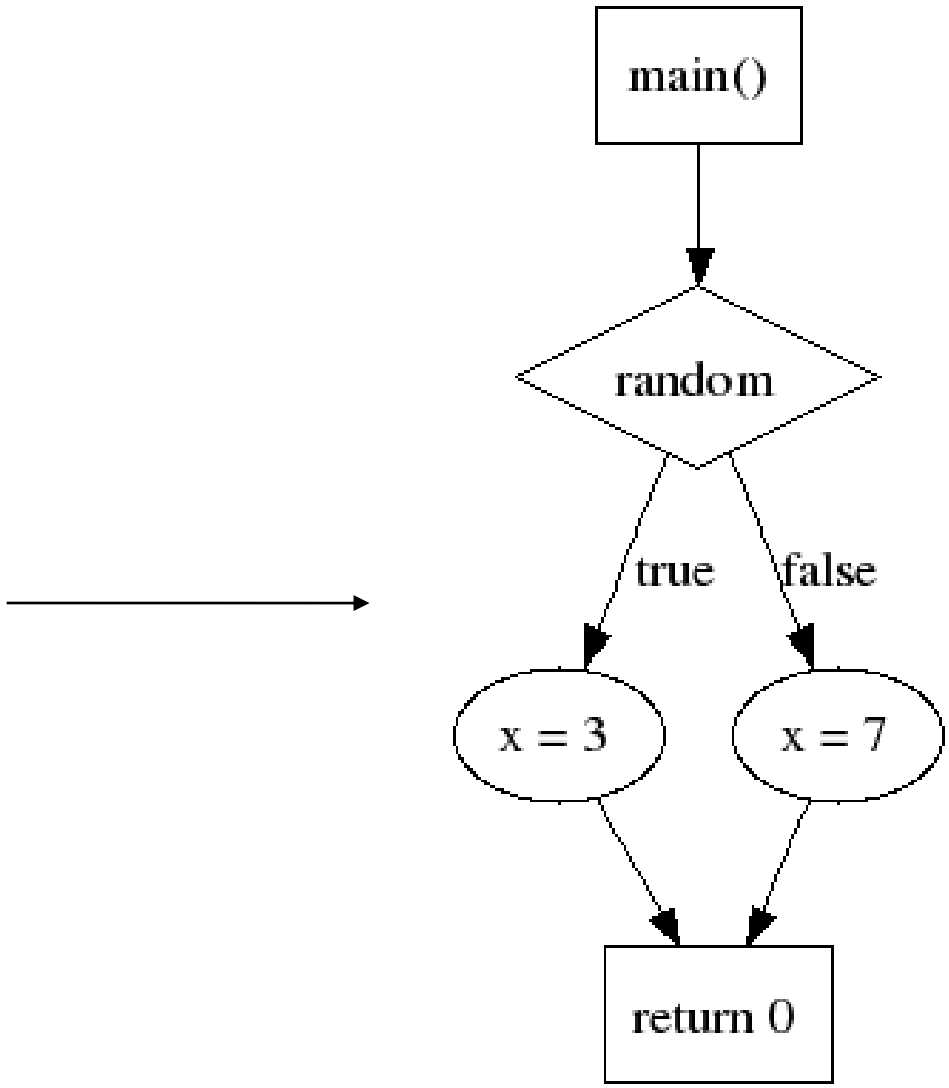


$\beta_{new} \geq \beta_{old} + 3$ N/mm²

$\beta_{new} \leq 2/3\ \beta_{old} + 14$ N/mm²

Data Flow

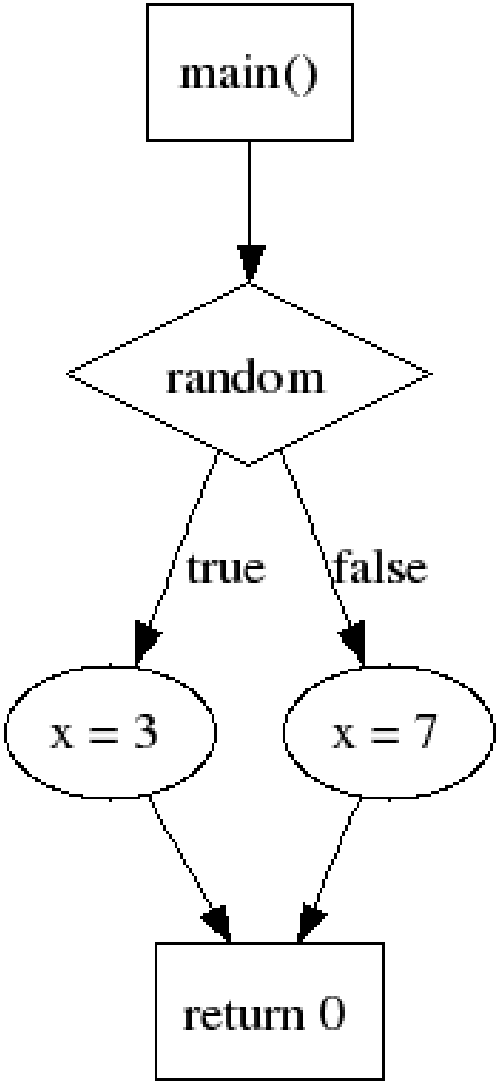
```
int main () {
   int random;
   int x;
   if (random)
      x = 3;
   else
      x = 7;
   return 0;
}
```



And we solve flow-equations!

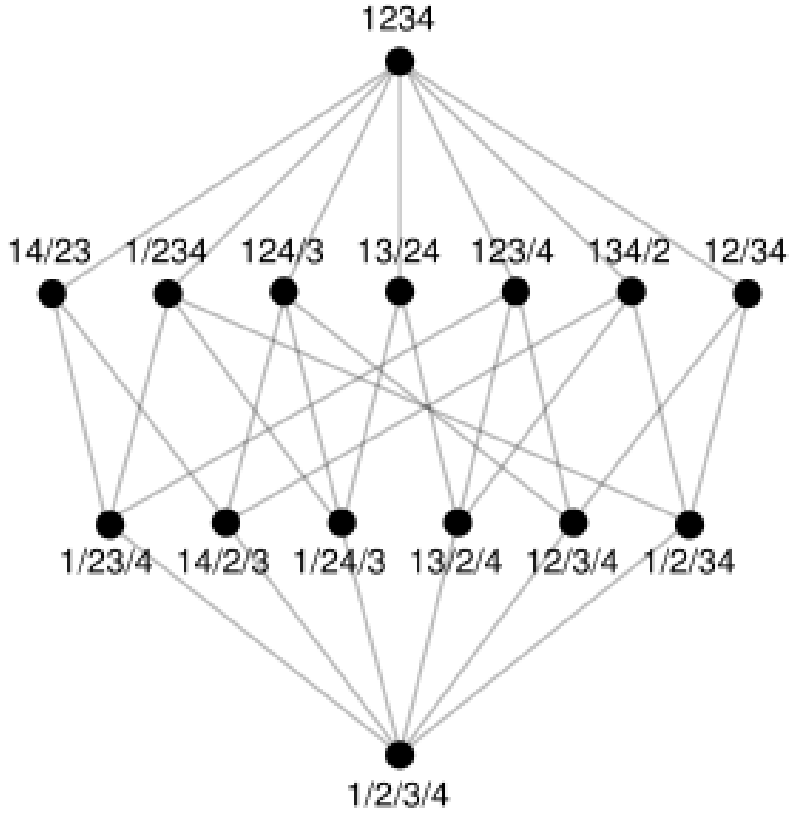
- S[1] = initial
- S[2] = “x=3”(S[1])
  S[3] = “x=7”(S[1])
- S[4] = S[2] + S[3]

- This is pseudo-code!



There is elegant theory behind this

- Lattice theory neatly captures joining of flow information!
- Extremely useful frameworks (Kam, Ullman, Kildall, 70s)
- Used in every compiler!



Why I like Semantics

- Balance between theory and practice
  Collaborations with Airbus, Daimler, etc.
- The mathematics is cute
  Varied: logic, categories, linear algebra …
- I like my workplace …
  Everything, except the language, is really great in Germany; highly recommended …

YOU and Semantics

- The Fundamental Attribution Error
  - Therefore, always blame the environment …
- Being paid for Ph.D. work is important!
  - I love my job, but ...
  - debugging a prototype is not all that fun.
- Supervision is also important
  - Not all of us were born with the ability to write papers that get accepted ☺

YOU and Semantics

- Goal: do your Ph.D. in semantics!
  - Varmo Vene in Tartu
  - Tarmo Uustalu in Tallinn
  - Helmut Seidl in Munich
- For this, you should first
  - Take courses on functional programming very seriously