Formal Semantics of Prog. Lang. – Introduction

Hanru Jiang (蒋瀚如)

BIMSA

Acknowledgments: some slides are taken from Zhong Shao, Xinyu Feng, and Hongjin Liang's slides for their courses.

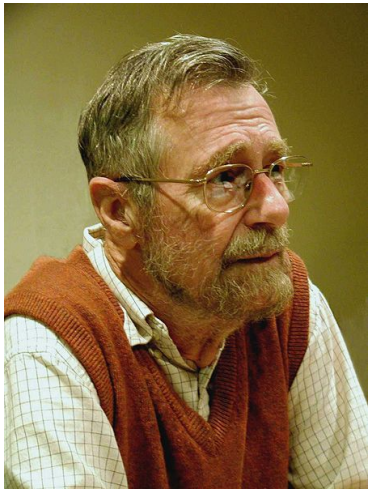
Programming Languages

- A fundamental field in computer science
  - As classic as OS and Architectures
    - SOSP: 1967；ISCA: 1973；POPL: 1973
- Old, but still very active
  - Many important research problems
  - New languages keep showing up
    - Rust, Go, Scala, F#, R, Matlab, Python, Kotlin, Julia, Swift, Typescript, Libra, …

Programming Languages

Computer Science is no more about computers than astronomy is about telescopes.

Edsger W. Dijkstra



How about PL then?

Programming Languages

PL research is broader than designing and implementing new languages. To me, a PL researcher is someone who views the programming language as having a central place in solving computing problems. From this vantage point, PL researchers tend to focus on **developing *general abstractions***, or *building blocks*, for solving problems, or classes of problems. PL research also **considers software *behavior* in a *rigorous and general way***, e.g., **to prove that (classes of) programs enjoy properties we want, and/or eschew properties we don't**. This approach has proven to be very valuable for solving a wide ranging set of problems.

Blog post by Mike Hicks:

What is PL research and how is it useful? - The PL Enthusiast

www.pl-enthusiast.net/2015/05/27/what-is-pl-research-and-how-is-it-useful/

Formal Semantics

- To assign *mathematical* meanings to language contructs & programs
- A *scientific* way to study PL and programming
  - "*developing general abstractions*"

Abstraction

Formal Semantics

- To assign *mathematical* meanings to language contructs & programs

- A *scientific* way to study PL and programming
  - "*developing general abstractions*"
  - "*considers software behavior in a rigorous and general way*"
    - More than testing

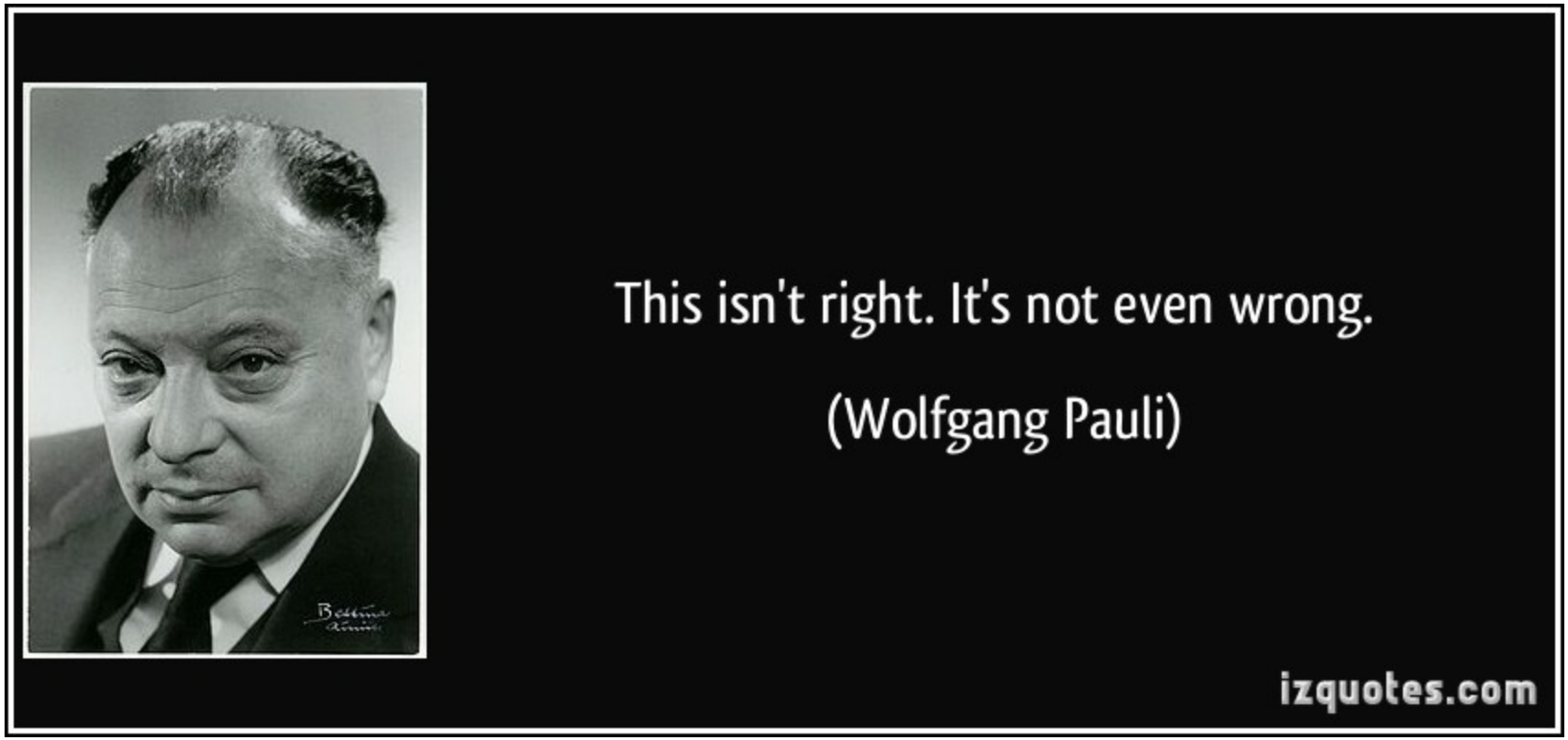
```
factorial( n )
{
  c = n;
  result = 1;

  while (c>1) {
    result = result * c;
    c = c-1;
  }
  return result;
}
```

Is it correct?

What do we mean by correctness?

```
factorial( n )
{
```



What do we mean by correctness?

{ n ≥ 0 }

```
factorial( n )
{
   c = n;
   result = 1;

   while (c>1) {
      result = result * c;
      c = c-1;
   }
   { result = n!/c! ∧ c=1}
   return result;
}
```

{ result = n! }

{ result = n!/c!}

We'll try to answer question like:

- How to describe meanings of programs?
- How to describe properties of programs?
- How to reason about programs?
- How to tell if two programs have the same behaviors or not?
- How to design a new language?

Why take this course?

- *Software reliability and security are the biggest problems faced by the IT industry today!*

You are likely to worry about them in your future job!

暴风影音-BaoFeng Player

版本 5.2.6，87.1 MB

2017年8月24日

更新

支持几乎所有本地视频格式，聚合全网在线视频！

修复了闪退的bug，还杀了一个程序员祭天

1、长短视频详情页进行了改版，显示为信息流

2、优化了搜索页面，

3、手势拖动快进功能优化

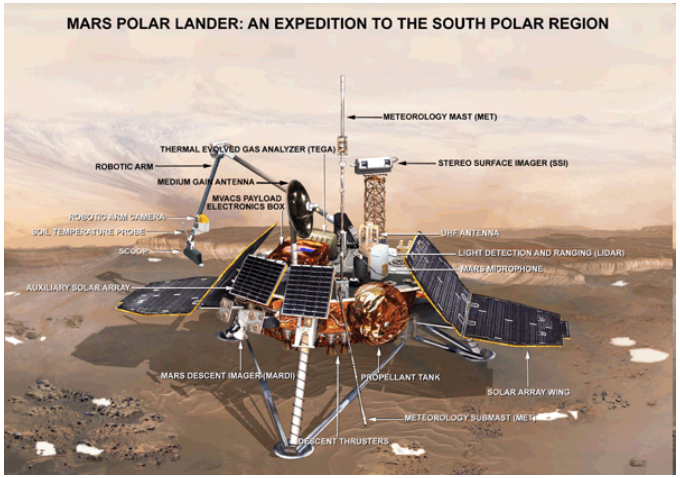
Arianne 5

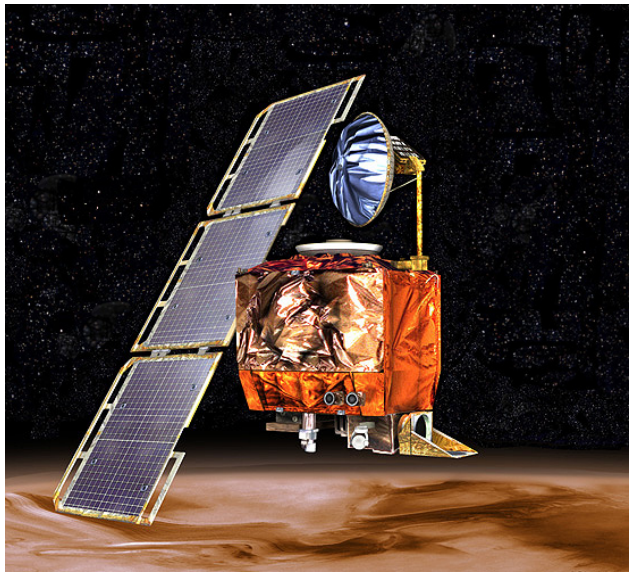
- On June 4, 1996, the Arianne 5 took off on its maiden flight.

- 40 seconds into its flight it veered off course and exploded.
- “Conversion of a 64bit integer into a 16bit signed integer leads to an overflow.”

- This picture become quite popular in talks on software reliability

“Better, Faster, Cheaper”

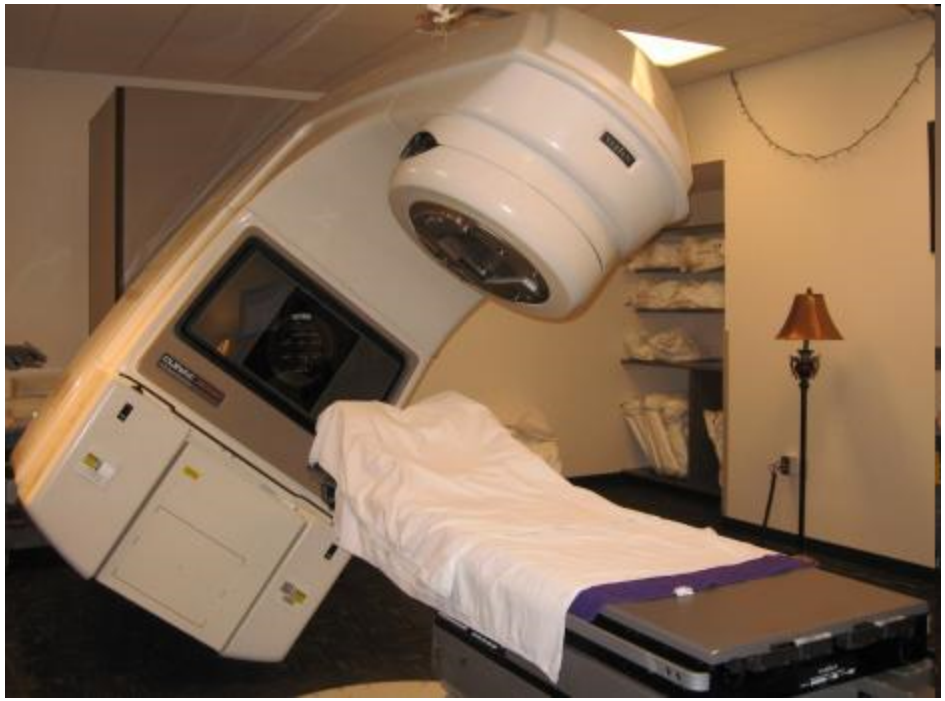




- In 1999, NASA lost both the Mars Polar Lander and the Climate Orbiter.
- Later investigations determined software errors were to blame.
  - Orbiter: Component reuse error.
  - Lander: Precondition violation.

Therac-25

From 1985-1987, 6 patients were killed or seriously injured as a result of overdosed radiation (100 times of the intended dose) by Therac-25, a radiation treatment facility.



The problem was due to a subtle race condition between concurrent processes.

Northeast blackout, 2003

A widespread power outage that occurred throughout parts of the Northeastern and Midwestern United States and Ontario, Canada on Thursday, August 14, 2003

Race conditions in GE Energy's Unix-based XA/21 energy management system caused alarm system failure.

Now think of viruses and Trojan Horses

Stuxnet is used to attack the nuclear power station in Iran in 2010.

The virus took advantage of 4 undeclared bugs in windows to take over the system.

SCIENTIFIC AMERICAN™

COMPUTING

Software Error Doomed Japanese Hitomi Spacecraft

Space agency declares the astronomy satellite a loss

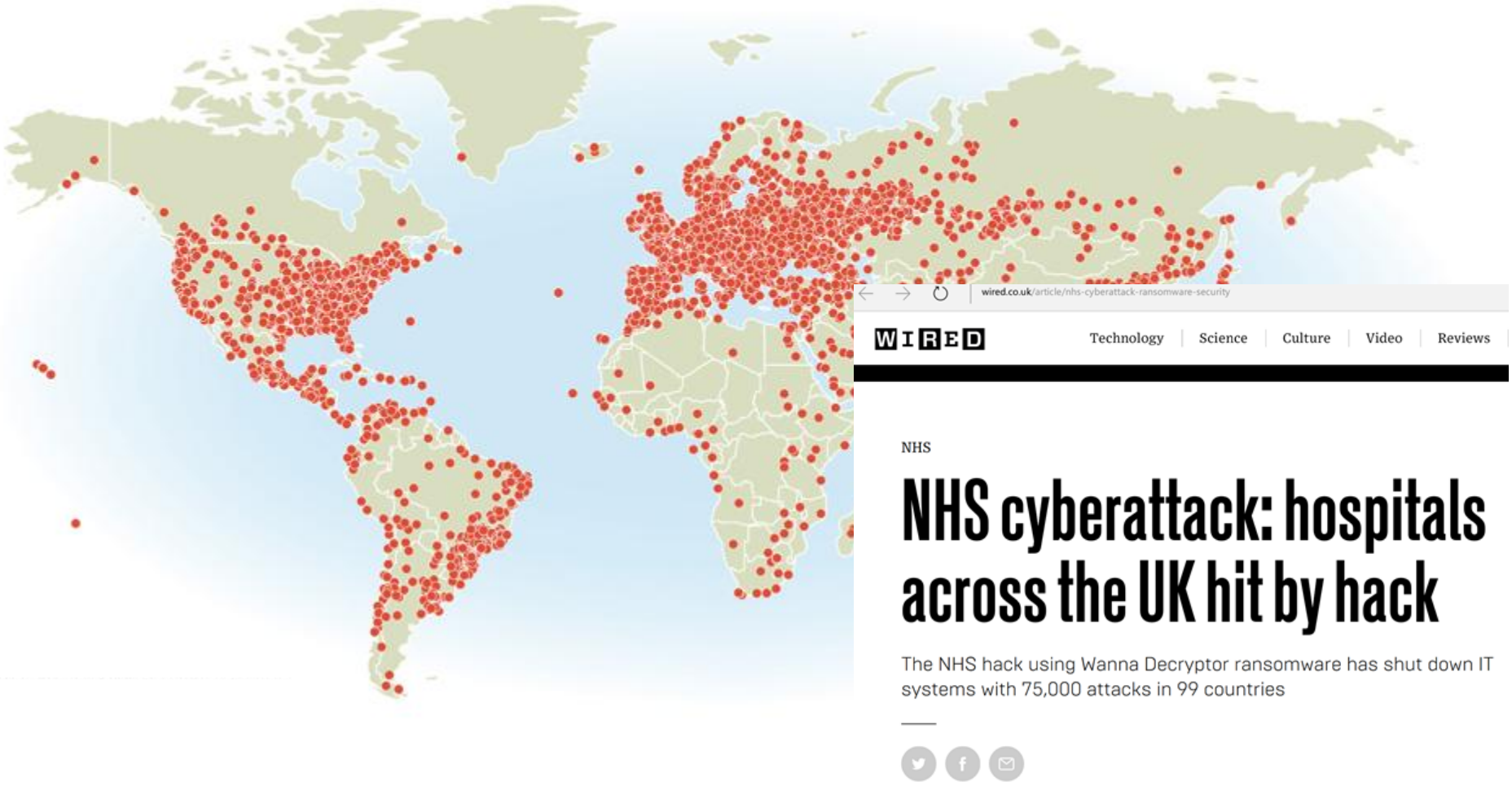
By Alexandra Witze,

底层软件BUG葬送了日本这台价值18亿的天文卫星

2016年4月30日　来源： 美果橘子

2017年5月WannaCry勒索软件攻击

利用Windows操作系统的缺陷



Uber无人车车祸原因：硬件“看”到人 软件撞死人

IT业界　腾讯科技　2018-05-08 07:22

“无人车本身发现了这名正在横穿马路的行人，但是自动驾驶软件系统没有采取避让的措施。”

腾讯科技讯 今年三月份，美国网约车公司Uber发生了[illegible]第一宗完全自动驾驶汽车致人死亡的事故，事故引发了全球舆论震惊，以及对无人车安全性的讨论。据外媒最新消息，Uber公司已经查出了这次车祸的原因，即无人车本身发现了这名正在横穿马路的行人，但是自动驾驶软件系统没有采取避让的措施。

KLINT FINLEY BUSINESS 06.18.16 04:30 AM

A $50 MILLION HACK JUST SHOWED THAT THE DAO WAS ALL TOO HUMAN

安全已成挑战 前4月区块链因安全损失达19亿美元

2018-05-08 16:35:00 作者：HAFOM 出处：比特网

区块链技术作为一种新兴技术，安全性威胁已是其面临的最重要的问题之一。5月8日，“安全.赢未来——2018区块链安全高峰论坛”在北京召开，会上白帽汇安全研究院发布《区块链产业安全分析报告》，并同时推出区块链安全网站。

安全 | https://segmentfault.com/a/1190000014531999

一行代码蒸发了¥6,447,277,680 人民币！

原 区块链 kimg1234 4月22日发布

uint256 amount = uint256(cnt) * _value

```
function batchTransfer(address[] _receivers, uint256 _value) public whenNotPaused returns (bool) {
    uint cnt = _receivers.length;
    uint256 amount = uint256(cnt) * _value;
    require(cnt > 0 && cnt <= 20);
    require(_value > 0 && balances[msg.sender] >= amount);

    balances[msg.sender] = balances[msg.sender].sub(amount);
    for (uint i = 0; i < cnt; i++) {
        balances[_receivers[i]] = balances[_receivers[i]].add(_value);
        Transfer(msg.sender, _receivers[i], _value);
    }
    return true;
  }
}
```

CNN BUSINESS Markets Tech Media Success Perspectives Video

Facebook软件漏洞导致1400万用户私密帖子被公开

Facebook bug set 14 million users' sharing settings to public

by Heather Kelly @heatherkelly

June 7, 2018: 3:00 PM ET

CCTV.com 央视网 新闻 国内 国际 图片

波音CEO就两起坠机事故致歉 承认飞行系统有问题

新闻频道 来源：央视网 2019年04月05日 10:19 A- A+ 我要

鳳凰網 科技

拼多多一晚上被薅200亿，一个bug引发的惨案

2019年01月

sina 新浪财经 产经 > 正文

行情 简称/代码/拼音

携程发致歉声明：“二次支付显示无票”为程序BUG

2019年03月11日 11:28 新浪财经 新浪财经APP

Bug-Free Software?

- **A grand challenge for computer scientists**
  - **Posed since 1960's**
  - **Significant progress, but still challenging**

- **Great practical implication**
  - **Software bugs cause the loss of 59.6 billion US dollars each year (0.6% GDP)**
    - **2002 report from NIST**
  - **"Null References: The Billion Dollar Mistake"**
    - Tony Hoare

Observations

- Failure often due to simple problems “in the details”.
  - **Small theorems about large programs would be useful.**
- Need clearly specified interfaces and checking of interface compliance.
  - **Better languages would help!**

New Challenges

Software is becoming more complex nowadays:

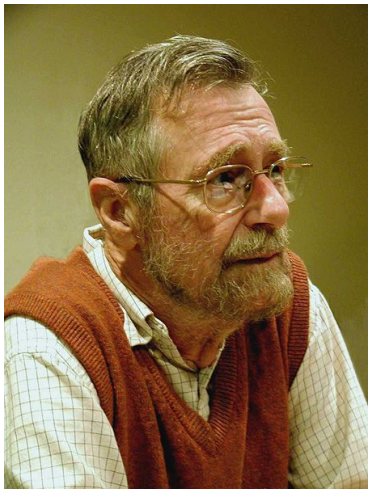
- Multi-core software
  - Concurrency
- Embedded software
  - Limited resources
- Distributed and cloud computing
  - Network environment
- Ubiquitous computing and Internet of Things
- Quantum computing
- …

测试的局限性

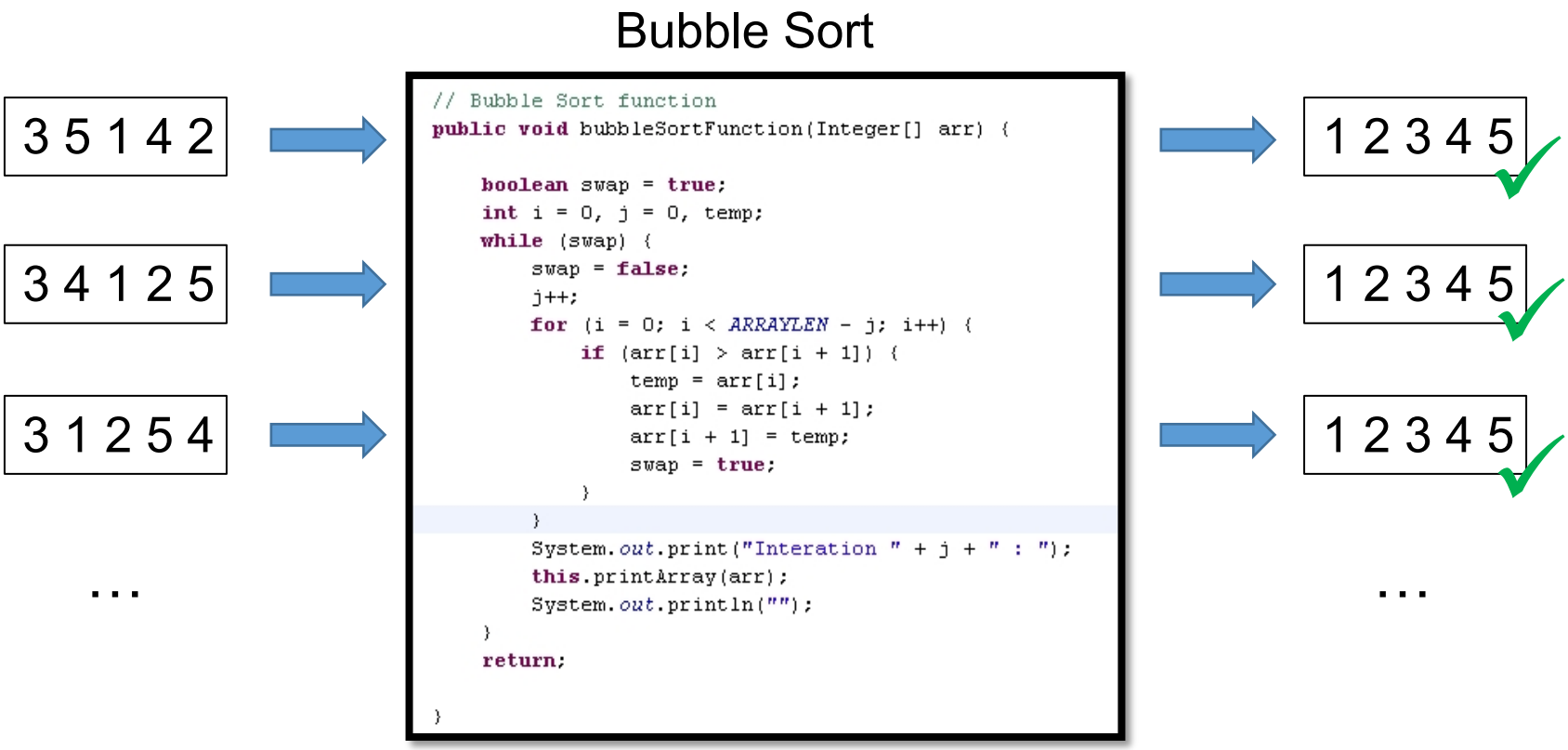
- 工程中广泛应用，并且有效
  - ☺ 简单、容易自动化
  - ☹ 无法保证没有bug
  - ☹ 在并发（多核/网络）下面作用有限

Testing shows the presence, not the absence of bugs.

Edsger W. Dijkstra



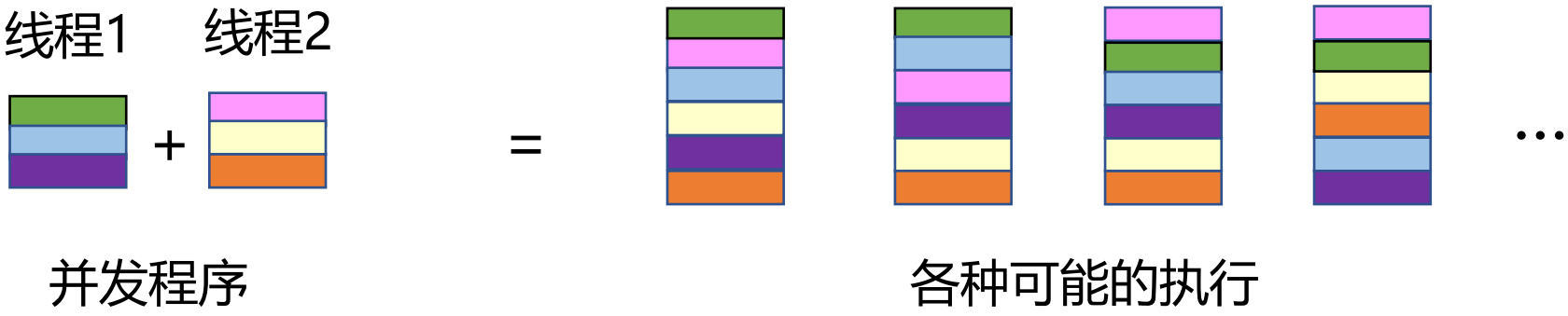
测试的局限性



无法遍历所有可能，因此不能肯定永远正确

测试的另一局限

测试并发程序：发现的bug难以重现



每一次运行同一个并发程序，都可能会产生不同的结果

Limitation of testing – Northeast blackout, 2003

We test exhaustively, we test with third parties, and we had in excess of three million online operational hours in which nothing had ever exercised that bug. I'm not sure that more testing would have revealed that. Unfortunately, that's kind of the nature of software… you may never find the problem. I don't think that's unique to control systems or any particular vendor software.

-- Mike Unum, manager at GE Energy

Race conditions in GE Energy's Unix-based XA/21 energy management system caused alarm system failure.

Opportunities

High assurance / reliability depends fundamentally on our ability to reason about programs.

The opportunities for new languages as well as formal semantics, type theory, computational logic, and so on, are great.

2011世界十大新兴技术

技术将会改变你的行为方式：你将用身体姿势来操控电视、
术可以促进你的健康，例如医生们将对不同肿瘤的相关基
出更有效的癌症疗法。不管技术属于哪一个类别，它们的
更加美好。

40 社交索引
42 智能变压器
44 手势识别接口
45 癌症基因组学
46 固态电池
48 同态加密
50 云流媒体
51 防崩溃代码
52 染色体分离
54 合成细胞

防崩溃代码

2011世界十大新兴技术

Social Indexing

Facebook remaps the Web to personalize online services

Homomorphic Encryption

Making cloud computing more secure

Smart Transformers

Controlling the flow of

Cloud Streaming

Bringing high-performance software to mobile devices

Crash-Proof Code

Controlling computers with our bodies

Crash-Proof Code

Making critical software safer

Cancer Genomics

Deciphering the genetics behind the disease

Separating Chromosomes

A more precise way to read DNA will change how we treat disease

40 社
42 智
44 手
4
4

英文版：**Technology Review, 2011(6), MIT Press**
http://www.technologyreview.com/tr10/

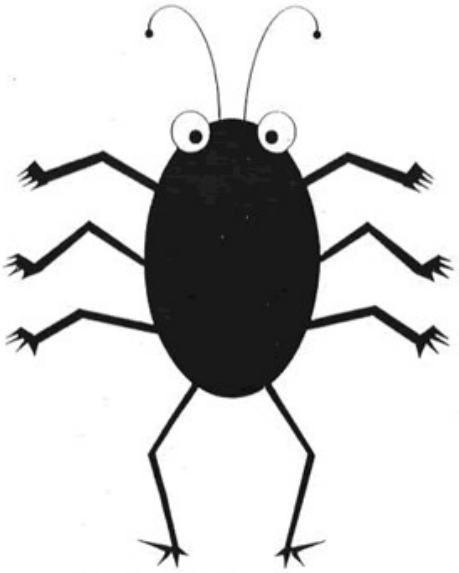
hetic Cells

ning new genomes
speed the creation of vaccines and biofuel-producing bacteria

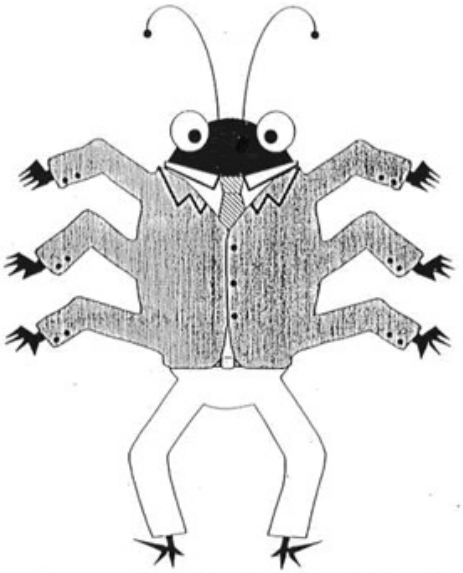
Report on Crash-Proof Code

- Verification of the seL4 OS kernel
  - Done by the Australian group at NICTA
  - Give mathematical proof showing the kernel would never crash
    - How to do this mathematically?
    - How to define “crash”?
    - How to prove the system is “crash-proof”?
    - We will answer the questions in this course

How to define “crash”?



BUG



FEATURE

央视新闻

8-9 15:00 来自 微博 weibo... 已编辑

146万
阅读

#华为正式发布鸿蒙#快讯！华为正式发布全新分布式操作系统：鸿蒙！为中国品牌，转！点赞！

鸿蒙发布会现场，余承东介绍鸿蒙的技术特性

Why take this course

- Software reliability and security are the biggest problems faced by the IT industry today! You are likely to worry about them in your future job!
- It will give you an edge over your competitors: **industry and most other schools don't teach this**.
- It will improve your programming skills – because you will have a better appreciation of what your programs actually *mean*.
- You will be better able to compare and contrast programming languages, or even design your own.
- It's intellectually deep: there're many challenging and hot research problems.

20位图灵奖得主

$$\{\ f \in [1..N] \longrightarrow [1..N] \ \mid$$
$$\quad \forall\ y \in [1..N].\ \ \exists\ x \in [1..N].\ \ f(x)=y\ \}$$

Course Overview

Goals of the Course

- Survey existing language features
  - What they mean? What they do? How they compare?
- Methods to define behaviors of programs
  - Operational/Denotational/Axiomatic Semantics
- Methods to reason about properties of programs
  - Define and prove “correctness” of programs
  - Building “crash-proof” or “bug-free” software

Preliminary Syllabus

- Introduction, Coq, Mathematical foundations
- Lambda calculus
  - Untyped, simply-typed
- Imperative languages
  - Operational semantics, denotational, Hoare logic
- C-like pointer programs and separation logic
- Advanced Topics:
  - Compiler verification
  - Quantum programming languages

Course Webpage:

https://hrjiang.github.io/semantics/

Lecture notes, homework and reading materials will all be posted on the course webpage.

You can email me your questions.

Please pay attention to the updates.