Foundations of Programming Languages

Hongfei Fu

John Hopcroft Center for Computer Science
Shanghai Jiao Tong University

Feb. 25th, 2021

Lecturer

Hongfei Fu

Lecturer:	Hongfei Fu
Research:	Model Checking, Program Verification
Website:	`http://jhc.sjtu.edu.cn/~hongfeifu/`
Email:	`fuhf@cs.sjtu.edu.cn`
Office:	Room 1408-1, Software Expert Building
Office-Hour:	Wednesday afternoon, every week

Qinxiang Cao

Lecturer:	Qinxiang Cao
Research:	Program Verification, Theorem Proving

Course Materials

The Textbook

Glynn Winskel

The Formal Semantics of Programming Languages: An Introduction

The MIT Press, 1993

What is the course about?

Programming Languages

- imperative languages: C, C++, Java, Python, ...
- functional languages: Haskell, ML, OCaml, ...
- assembly languages: MASM, NASM, ...

Programming Languages

- imperative languages: C, C++, Java, Python, ...
- functional languages: Haskell, ML, OCaml, ...
- assembly languages: MASM, NASM, ...

But: Warning!

- This course is not about how to write programs!!!

Course Focus

Program Semantics

Course Focus

Program Semantics

- programs as mathematical objects

Course Focus

Program Semantics

- programs as mathematical objects
- logical characterizations of programs

Course Focus

Program Semantics

- programs as mathematical objects
- logical characterizations of programs
- mathematical properties for programs

Course Focus

Why study programs mathematically?

Course Focus

Why study programs mathematically?

- fundamental components

Course Focus

Why study programs mathematically?

- fundamental components
- rising complexity

Course Focus

Why study programs mathematically?

- fundamental components
- rising complexity
- main factor in system performance

Course Focus

Why study programs mathematically?

- fundamental components
- rising complexity
- main factor in system performance
- main reason for system failure

Course Focus

Why study programs mathematically?

- fundamental components
- rising complexity
- main factor in system performance
- main reason for system failure

Potential Hazard of Program Error

- malfunction of codes
- crash of systems
- attack from hackers
- ...

Course Focus

Program Verification

Formal approaches for analyzing programs:

- rigorous proof for absence of bugs
- rigorous proof for absence of vulnerabilities
- full enumeration of race situations
- timing analysis of programs
- ...

Course Focus

Testing Approaches

Course Focus

Testing Approaches

- easy to conduct
- can detect normal bugs
- less coverage over codes
- more tendency to neglect critical bugs
- inapplicable in certain situations (e.g., concurrency)

Stuxnet

Stuxnet Computer Worm:

- targets programmable logic controllers;
- hinders automation of electromechanical processes;
- damaged Iran's nuclear plants.

WannaCry

WannaCry Ransomware:

- utilizes a flaw in the Microsoft SMB protocol;
- enforces encryption of data and demands ransom;
- affected computers worldwide.

Timsort

Timsort Implementation Bug:

- is introduced by optimization on merge sort;
- is widely used;
- causes software crash;
- happens rarely.

Retrospect

Retrospect

- Subtle critical bugs are hard to detect through testing.

Retrospect

- Subtle critical bugs are hard to detect through testing.
- Subtle critical bugs can be devastating if they are triggered.

Retrospect

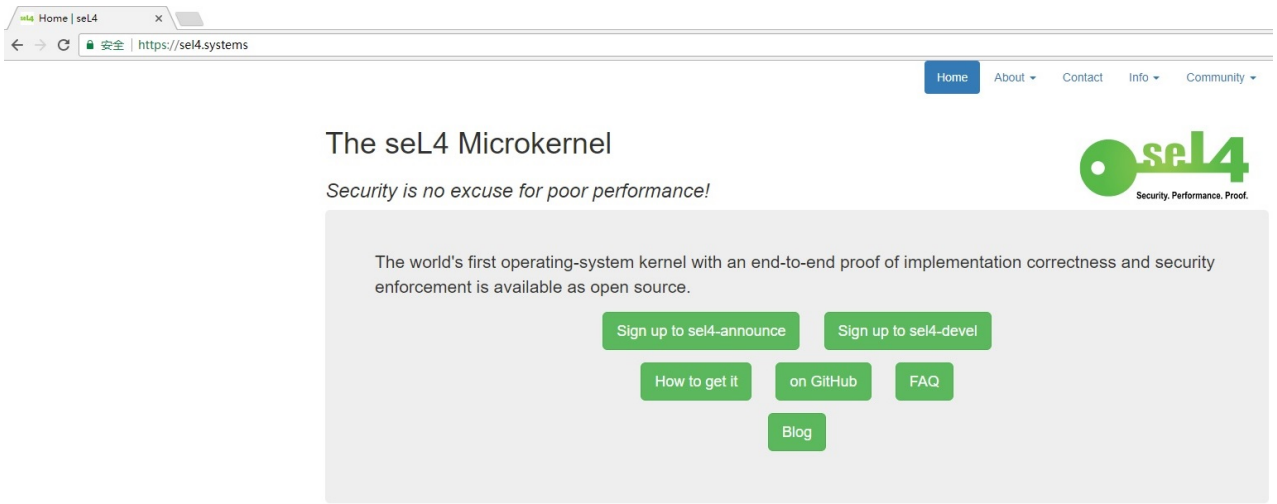
- Subtle critical bugs are hard to detect through testing.
- Subtle critical bugs can be devastating if they are triggered.
- Subtle critical bugs are vulnerable against adversaries.

Successful Stories

- functionally-correct operating system: SEL4
- hacker-free operating system: CertiKOS
- error-free compiler: CompCert, L2C
- race-free concurrency: Astrée
- ...

Successful Stories

SEL4 Operating System



source: from internet

Successful Stories

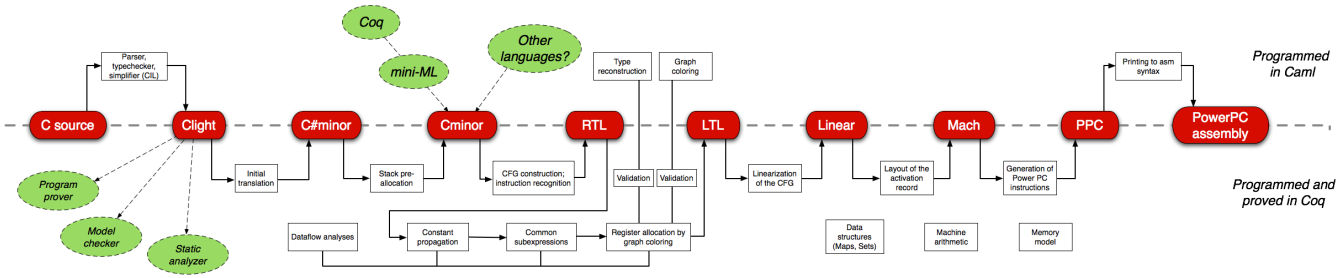
CertiKOS Operating System



source: from internet

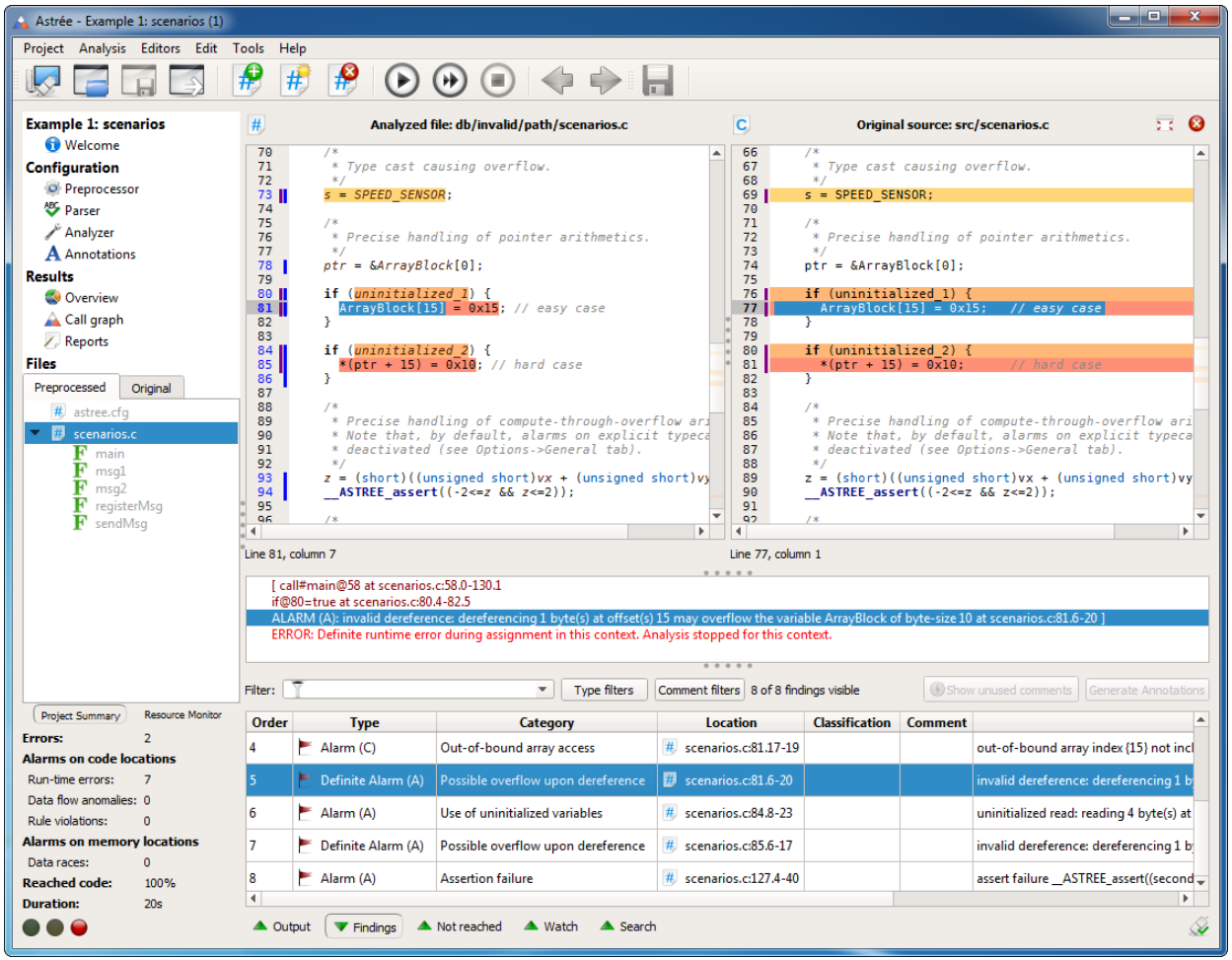
Successful Stories

CompCert Compiler



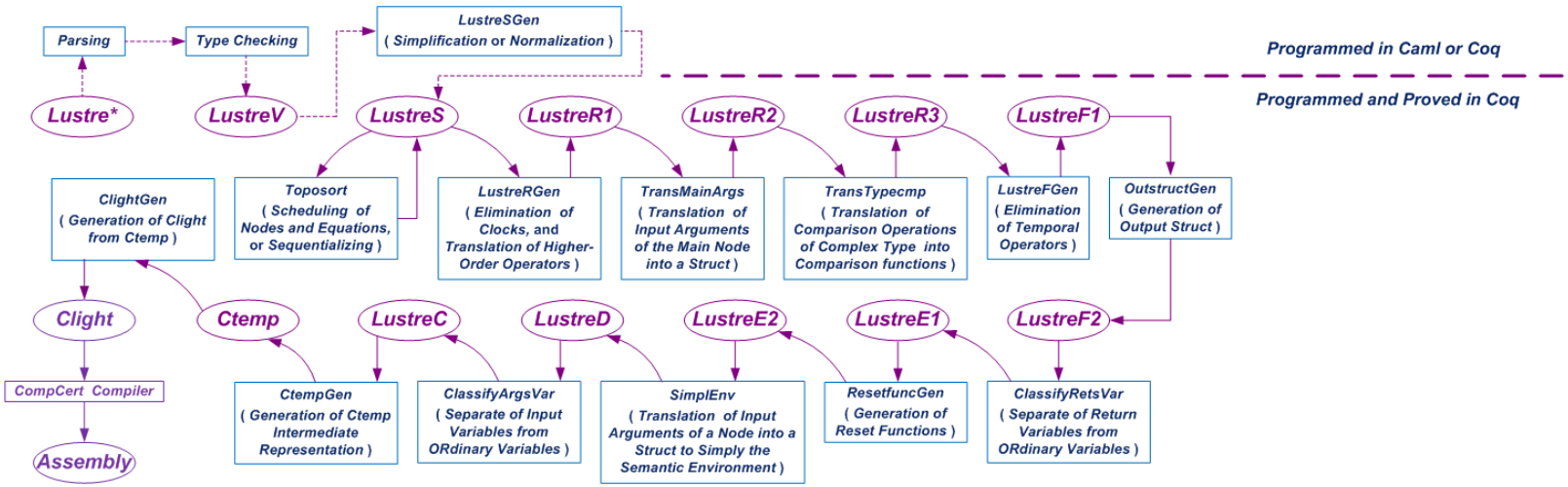
source: from internet

Astrée Static Analyzer



source: from internet

L2C: A Formally Verified Compiler From Lustre To C



source: from internet

Successful Stories

Amazon Web Services



source: from internet

Successful Stories

Huawei Harmony OS



source: from internet

Perspective

Program Verification

Perspective

Program Verification

- is difficult, but not infeasible.

Perspective

Program Verification

- is difficult, but not infeasible.
- has much better guarantee than testing.

Perspective

Program Verification

- is difficult, but not infeasible.
- has much better guarantee than testing.
- is necessary in critical systems.

Perspective

Program Verification

- is difficult, but not infeasible.
- has much better guarantee than testing.
- is necessary in critical systems.

In Our Course:

Program semantics provides a solid theoretical foundation for program verification.

What will the course cover?

Course Content

Program Semantics

- operational semantics: how do programs execute ?
- denotational semantics: what do programs output ?
- axiomatic semantics: which requirements do programs meet ?

Course Content

Types of Programs

- (mostly) imperative programs
- (some) functional programs

Course Content

Types of Programs

- (mostly) imperative programs
- (some) functional programs

Technical Content

- logical definitions for programs
- mathematical background behind program semantics

Course Content

Chapter 1:	set theory
Chapter 2,3,4:	operational semantics
Chapter 5:	denotational semantics
Chapter 6,7:	axiomatic semantics
Chapter 8:	domain theory (if we have time)
by Prof. Qinxiang Cao:	typed languages

An Example

$$\textbf{while}\ \ (X \leq 100)\ \ \textbf{do}\ \ X := X + 2$$

Another Example

$$
\begin{array}{l}
\textbf{while}\ \ (X \leq 100)\ \ \textbf{do} \\
\quad \textbf{if}\ \ (X \leq 0) \\
\qquad \textbf{then}\ \ X := X - 1 \\
\qquad \textbf{else}\ \ X := X + 2
\end{array}
$$

What can we gain from this course?

Course Benefits

- a rigorous thinking of programs
- a comprehensive start to program analysis and verification

Course Evaluation

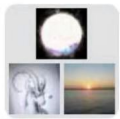
Course Evaluation

final exam	50%
course participation	50%

Course Participation (at my side)

homework	30%
class participation	10%
attendance	10%

Wechat Group Code for the Course



2021程序语言理论课程群

该二维码7天内(3月1日前)有效，重新进入将更新

Questions?

Happy Lantern's Festival!