CS 292C Computer-Aided Reasoning for Software

Lecture 1: Introduction

Yu Feng

Fall 2019

Outline of this lecture

- Introducing the cast
- Motivation and goals
- Course structure

Introducing the cast

Y’all are playing the lead, not audience!

Introducing the cast

Instructor: Yu Feng

yufeng@cs.ucsb.edu

Fri 11:00 am

Office: HFH 2157

Introducing the cast

PLSE lab@UCSB

Yu Feng

Yanju Chen

You?

Introducing the cast

Research@PLSE lab

Programming languages

- Algorithms for program analysis
- Algorithms for program synthesis

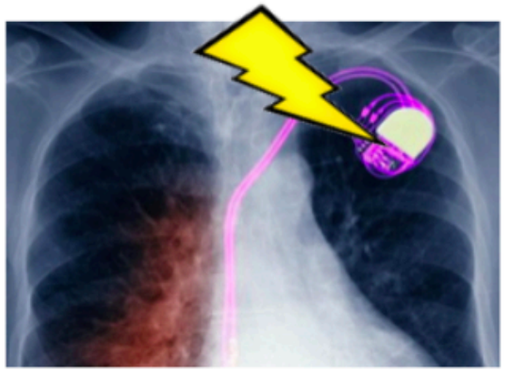
Security

- Smart contracts (with MSR)
- Malware detection (with Google Play Protect)

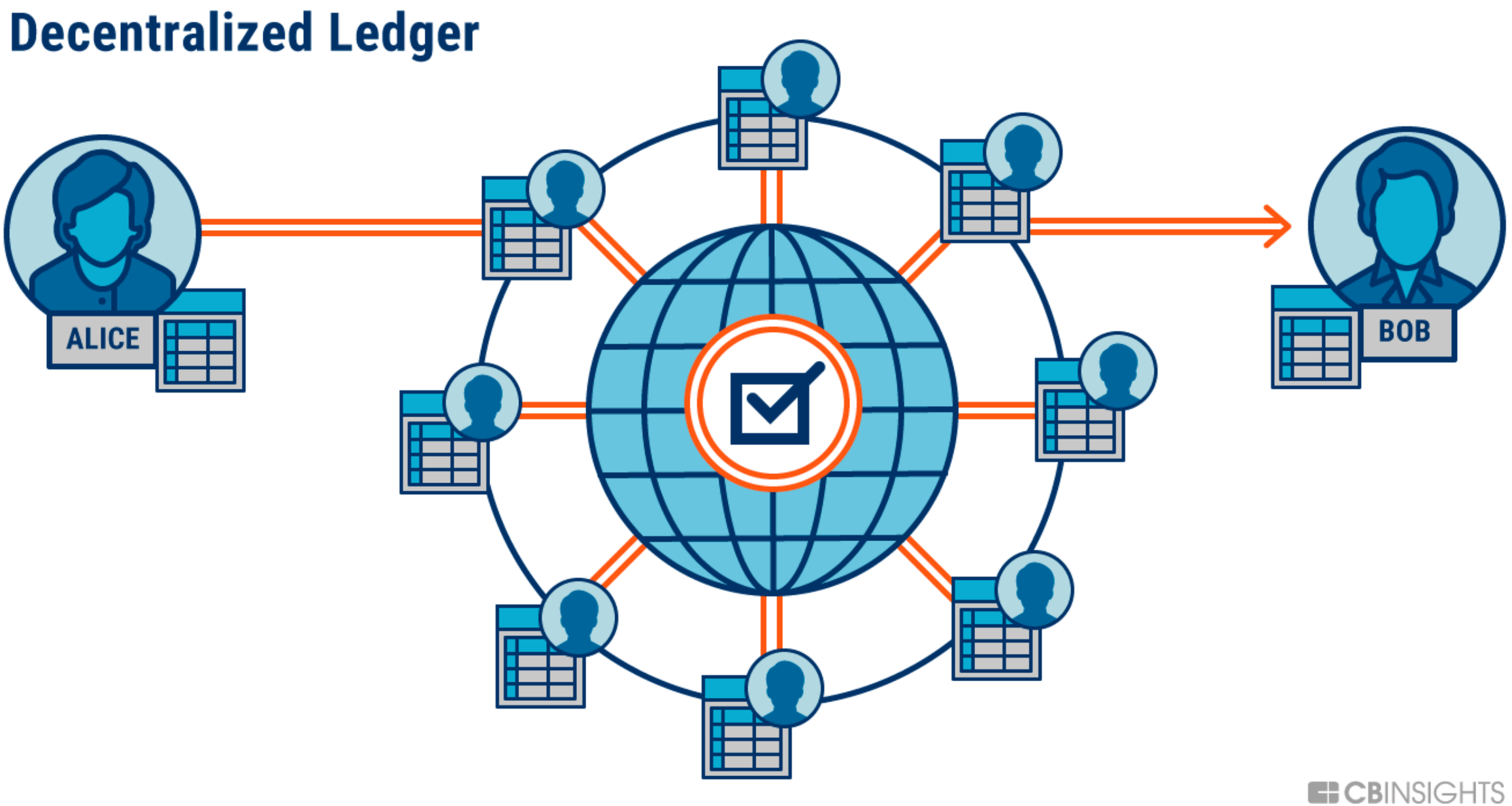
Software engineering & HCI

- Data wrangling
- Data visualization
- …

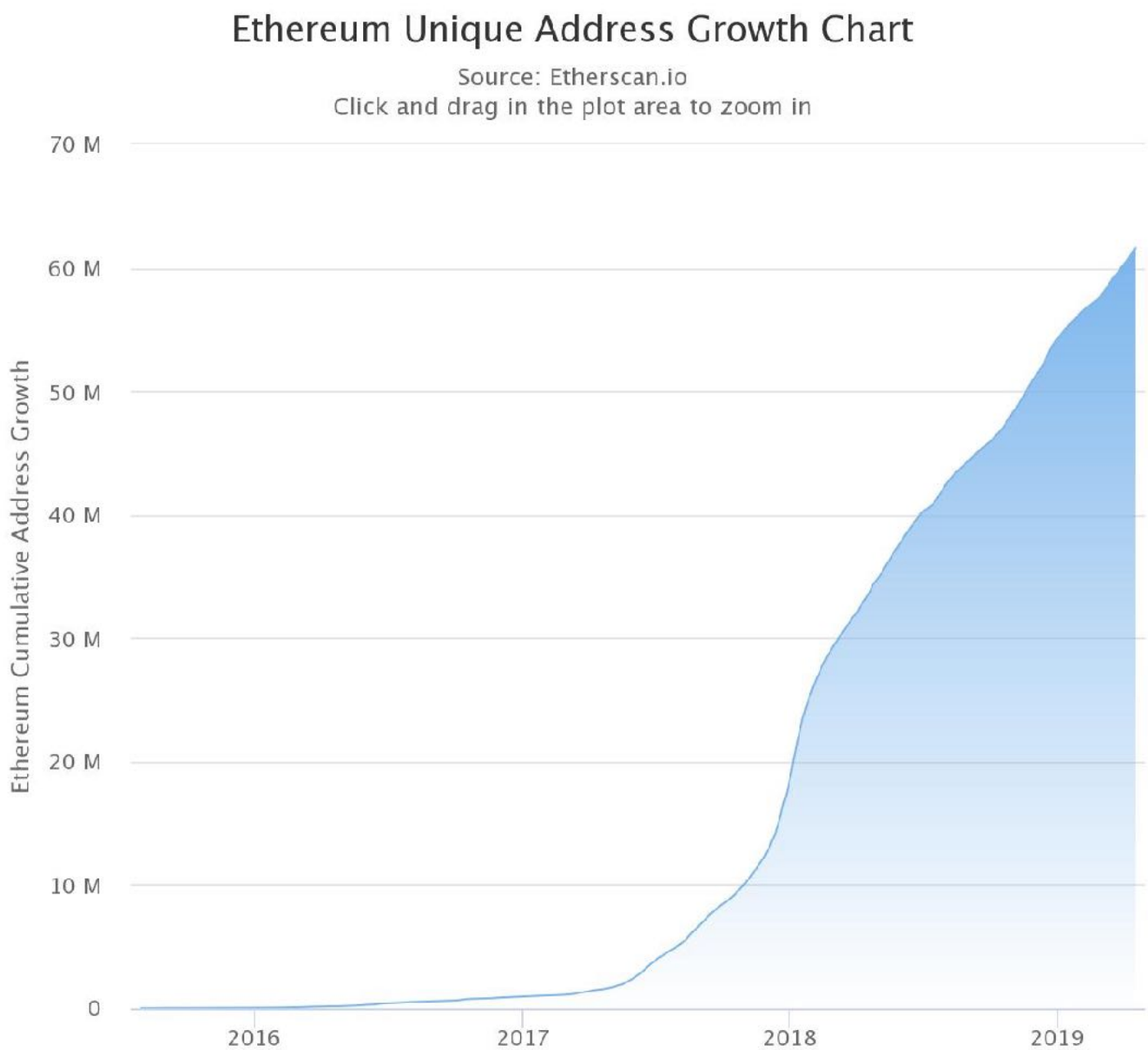
Motivation and goals



Motivation and goals



Motivation and goals



https://etherscan.io/chart/address

Motivation and goals

```
1  contract PausableToken {
2  bool flag = false;
3
4  function makeFlag(bool fg) {
5   flag = fg;
```

ETHEREUM

BatchOverflow Exploit Creates Trillions of Ethereum Tokens, Major Exchanges Halt ERC20 Deposits

Sam Town · Apr 25, 2018 · 3 min read

```
13
14      balances[msg.sender] =
15        balances[msg.sender].sub(amount);
16      for (uint i = 0; i < cnt; i++) {
17        address recv = _receivers[i];
18        balances[recv] =
19          balances[recv].add(_value);
20        Transfer(msg.sender, recv, _value);
21      }
22      return true;
23    }
24  }
```

How to ensure software robustness?

Testing, screening, …

Why is so difficult to ensure software robustness?

The problem is undecidable…

Motivation and goals

```
1  contract PausableToken {
2  bool flag = false;
3
4  function makeFlag(bool fg) {
5   flag = fg;
```

ETHEREUM

BatchOverflow Exploit Creates Trillions of Ethereum Tokens, Major Exchanges Halt ERC20 Deposits

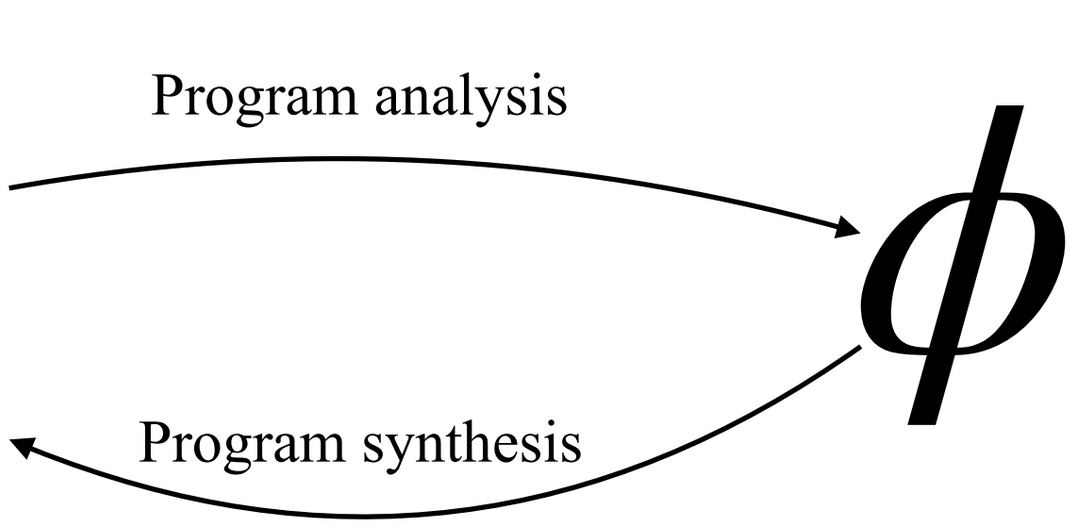
Sam Town · Apr 25, 2018 · 3 min read

```
13
14      balances[msg.sender] =
15        balances[msg.sender].sub(amount);
16      for (uint i = 0; i < cnt; i++) {
17        address recv = _receivers[i];
18        balances[recv] =
19          balances[recv].add(_value);
20        Transfer(msg.sender, recv, _value);
21      }
22      return true;
23    }
24  }
```

Motivation and goals

Goal: Ensure robustness of software via symbolic reasoning

```
1  contract PausableToken {
2  bool flag = false;
3
4  function makeFlag(bool fg) {
5   flag = fg;
6  }
7
8  function batchTransfer(address[] _receivers,
       uint256 _value) {
9      uint cnt = _receivers.length;
10     uint256 amount = uint256(cnt) * _value;
11     require(flag);
12     require(balances[msg.sender] >= amount);
13
14     balances[msg.sender] =
15       balances[msg.sender].sub(amount);
16     for (uint i = 0; i < cnt; i++) {
17       address recv = _receivers[i];
18       balances[recv] =
19         balances[recv].add(_value);
20       Transfer(msg.sender, recv, _value);
21     }
22     return true;
23   }
24 }
```



Course structure: prerequisites

Discrete mathematics

Compilers

Programming languages

Course structure: logistics

Website: https://github.com/fredfeng/CS292C

Q&A: https://github.com/fredfeng/CS292C/issues

You need a Git account to post questions!

Workload: medium

- 3 programming assignments
- Paper reviews
- Final project

Course structure: syllabus

- Solver-aided reasoning
- Program analysis
    - Symbolic execution
    - Bounded model checking
- Program synthesis
    - Synthesis from formal specifications
    - Synthesis from informal specification
    - Applications of program synthesis

Course structure: grading

- Programming assignments: 15%
  - 3 programming assignments, 5% each
- Paper reviews: 30%
  - 6 papers, 5% each
- Final Project: 50%
  - Team formed by deadline: 5%
  - 1-page project proposal: 15%
  - Project presentation: 15%
  - Final report: 15%
- Class Participation: 5%

Course structure: project

- Types of final projects
  - Re-implement a technique from a paper
  - Apply existing synthesis framework to a new domain (Robotics, systems, security)
  - Extend/improve existing synthesis algorithm or tool
- Criteria
  - Quality of execution
  - Originality
  - Scope
  - Related to program analysis, verification, or synthesis

TODOs by Friday

- Create your Git account
- Install Rosette and Neo
  - Install Rosette: https://docs.racket-lang.org/rosette-guide/ch_getting-started.html
  - Install Neo: https://github.com/fredfeng/Trinity
- Start to look for partners for your final project!