Languages, Compilers and Interpreters (Lab)

2020/2021

Presentation

IMT SCUOLA ALTI STUDI LUCCA

Instructor: Dr. Letterio Galletta

- Assistant Professor @ IMT Lucca
- Instructor @ UniFI for the course “Distributed Programming for IoT” (MSc. Degree in Computer Science)
- Coach of C3T team @ cyberchallenge.it (born2scan)

Research interest: PL, Language-based security, Program analysis and verification, Blockchain, IoT

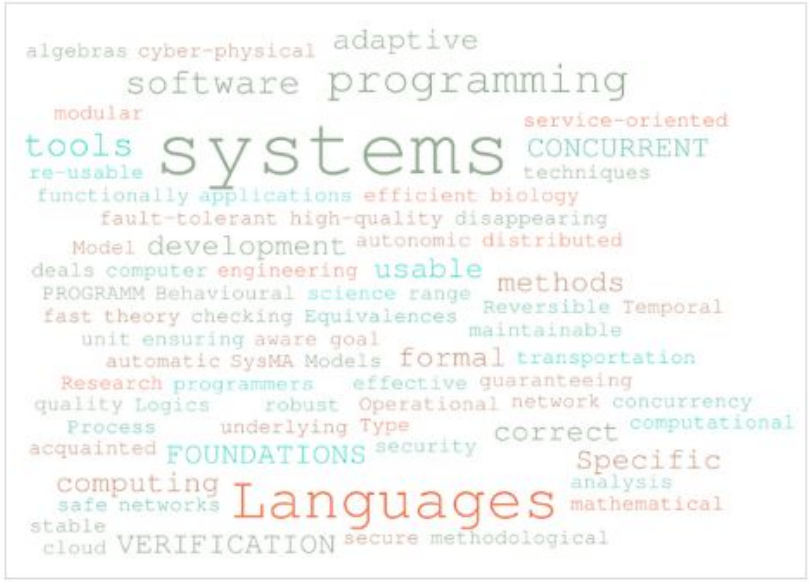
- letterio.galletta@imtlucca.it
- http://sysma.imtlucca.it, http://spuma.di.unipi.it/

Presentation

Research in computer science, carried out within the SysMA unit, deals with the development of languages and techniques for the analysis and verification of concurrent and distributed systems.

Our goal is to push the use of formal methods as methodological and automatic tools for the development of high-quality, correct software and systems that are fast, usable, re-usable, maintainable, and modular.

To make our tools usable by programmers not acquainted with the underlying mathematical



Course Logistics

Class Schedule:

- Friday: 11:00 - 12:45 classroom: MS Teams

Office hours: e-mail me to set up a virtual meeting

Course page: https://github.com/lillo/compiler-course-unipi

There will be programming exercises: bring your laptop

Course Readings

Programming tutorials, papers & further material

References will be given during the classes and will posted on the course page

A note about slides

- Slides are made to be used **by the teacher** during the lectures, **not** to be studied **by students**
- Slides will be available on the page course
- Students must study the corresponding chapters of the books and the other referenced material

This Lab Will Teach You

How to implement a small compiler and interpreter

Side effects:

1. The basics of OCaml
2. How to implement parsers using parsers generators
3. How to implement semantic analyses
4. How to use the LLVM toolchain for generating and optimize code

Our main goal: the MicroC compiler

A small subset of the C language:

- datatypes: only int and char variables, arrays, and pointers
- no structs, unions, doubles, function pointers, ...
- no initializers in variable declarations
- functions can return only int, char, void, bool
- no pointer arithmetic
- pointers and arrays are not interchangeable
- no dynamic allocation of memory

```
void insertionSort(int arr[], int n)
{
  int i;
  int key;
  int j;

  for (i = 1; i < n; i = i + 1) {
    key = arr[i];
    j = i - 1;

    while (j >= 0 && arr[j] > key) {
      arr[j + 1] = arr[j];
      j = j - 1;
    }
    arr[j + 1] = key;
  }
}
```

Our main goal: the MicroC compiler

At the end of each block of lectures there will be an assignment concerning the implementation of a piece of the compiler

Program of the Course

1. Introduction to OCaml
2. Lexing and parsing using ocamllex and menhir
3. Semantic analysis implementation: type checking, scope management, control-flow analysis
4. Introduction to LLVM infrastructure and LLVM intermediate language
5. Code generation

Questions?