

Formal Systems and their Applications

Course nr: H04H8A

Frank Piessens
(Frank.Piessens@cs.kuleuven.be)

Acknowledgment: these slides are based in part on slides from Benjamin Pierce

Course Overview

- Introduction
- Formal models of programming languages
- Simple type systems
- Subtyping
- Conclusion

Summary

- We have studied:
 - Formal models of programming languages
 - Syntax and evaluation
 - Formal models of type checking
 - Typing rules, and safety guarantees offered by typing
- For programming languages with:
 - (higher-order) functions, references and assignment, objects, classes, records, exceptions, subtyping, ...

Lessons Learned

- What I hope you will take with you from this class:
 - A deeper understanding of various programming language features and type checking
 - Familiarity with the techniques used by programming language researchers
 - You should be able to read and understand (some) research papers now
 - Capable to design your own (small) languages in a rigorous way

The road ahead

- Topics in the book we did not cover
 - Mainly “making the type system more expressive”
 - Recursive types (but we saw something in FJ)
 - Polymorphism
 - i.e. The foundations of Java and C# Generics
 - Existential types
 - The foundations of Abstract Data Types
 - Some more advanced type systems that are not yet in mainstream languages
- There is a follow-up book if you want more 😊 !

Examination

- Evaluation by means of a classical exam:
 - Prepare in writing
 - Defend orally
 - Including a defense of your solution to the project assignment
- Emphasis is on checking your understanding