

# Verification of Functional Programs

## Introduction

Andrés Sicard-Ramírez

Universidad EAFIT

Semester 2026-1

# Pácto Pedagógico

# Pacto Pedagógico

---

Como miembros de la Universidad EAFIT, nos comprometemos a actuar de manera íntegra siguiendo los más altos estándares éticos y morales.

- Respeto
- Tolerancia
- Honradez
- Compromiso

# Pacto Pedagógico

---

Página web del curso

<https://asr.github.io/courses/verification-of-functional-programs/2026-1>

# Pacto Pedagógico

---

Página web del curso

<https://asr.github.io/courses/verification-of-functional-programs/2026-1>

Conducto regular, fechas y porcentajes de las evaluaciones

La información está en la página web del curso.

# Pacto Pedagógico

---

## Página web del curso

<https://asr.github.io/courses/verification-of-functional-programs/2026-1>

## Conducto regular, fechas y porcentajes de las evaluaciones

La información está en la página web del curso.

## Responsabilidad compartida

- Profesor
- Estudiantes

# Pacto Pedagógico

---

## Asistencia a clase

Reglamento académico de los programas de posgrado, Capítulo VI, Artículo 62, Parágrafo 2.

*“El estudiante de posgrado cuyas faltas de asistencia lleguen al treinta por ciento (30%) del total de las horas de clase programadas para el curso o para una parte de éste, cuando se desarrolle con más de un profesor, en secciones temáticas denominadas ‘módulo’, pierde con calificación de cero punto cero (0.0) del seminario o curso correspondiente y esta nota afecta el promedio crédito acumulado.”*

# Pacto Pedagógico

---

## Orientaciones para el curso

- Se recomienda cuatro horas de trabajo por semana (dos horas por cada hora de clase).
- Las clases son presenciales.
- La evaluación a la docencia es obligatoria.
- Se recomienda revisar periódicamente los canales de comunicación institucionales (EAFIT Interactiva y el correo institucional).
- El estudiante podrá entrar a clase a más tardar 20 minutos después de la hora asignada para su inicio.

## Course Outline

# Preliminaries

---

## Textbook

Tobias Nipkow ([2025] 2026). Functional Data Structures and Algorithms.

## Convention

The numbers and page numbers assigned to chapters, examples, exercises, figures, quotes, sections and theorems on these slides correspond to the numbers assigned in the textbook.

# Course Outline

---

- Introduction
- Isabelle/HOL
- Analysis of algorithms
- Algorithms for sorting and selection
- Algorithms on trees
- Algorithm design techniques and analysis techniques

# Interactive Proof Assistants

# Interactive Proof Assistants

---

## Description

*“Proof assistants are computer systems that allow a user to do mathematics on a computer, but not so much the computing (numerical or symbolical) aspect of mathematics but the aspects of **proving** and **defining**. So a user can **set up** a mathematical theory, define properties and do logical reasoning with them.”*  
(Geuvers 2009, p. 3.)

# Interactive Proof Assistants

---

## Description

*“Proof assistants are computer systems that allow a user to do mathematics on a computer, but not so much the computing (numerical or symbolical) aspect of mathematics but the aspects of **proving** and **defining**. So a user can **set up** a mathematical theory, define properties and do logical reasoning with them.”*  
(Geuvers 2009, p. 3.)

## Examples

- Based on set theory: Isabelle/ZFC, Metamath and Mizar
- Based on higher-order logic: HOL4, HOL Light and Isabelle/HOL
- Based on type theories: Agda, Rocq and Lean.

# Isabelle

---

- University of Cambridge (England) and Technical University of Munich (German)
- Based on higher-order logic
- Tactic-based
- Extraction of programs to Haskell, OCaml, Scala and SML
- Written in SML
- Integration with ATPs and SMT solvers

## References

## References

---

- H. Geuvers (2009). Proof Assistants: History, Ideas and Future. *Sadhana* 34.1, pp. 3–25. DOI: [10.1007/s12046-009-0001-5](https://doi.org/10.1007/s12046-009-0001-5) (cit. on pp. 13, 14).
- Tobias Nipkow, ed. (2025). *Functional Data Structures and Algorithms. A Proof Assistant Approach*. Association for Computing Machinery (ACM). DOI: [10.1145/3731369](https://doi.org/10.1145/3731369) (cit. on p. 17).
- Tobias Nipkow, ed. [2025] (22nd Jan. 2026). *Functional Data Structures and Algorithms. A Proof Assistant Approach*. Updated version of (Nipkow 2025). URL: <https://fdsa-book.net/> (cit. on p. 10).