

Curso avanzado sobre tipos dependientes y demostradores interactivos
Pablo Barenbaum (Profesor DC, FCEN, UBA)

Programa:

El curso se propone dar los fundamentos teóricos que subyacen a los asistentes de demostración interactiva de teoremas, y en particular a aquellos basados en teorías de tipos dependientes. Desde el punto de vista práctico, el curso se propone introducir a los estudiantes al uso de demostradores interactivos para la verificación de programas y la formalización de teoremas matemáticos.

Se espera que, después de haber aprobado el curso, los estudiantes:

- Entiendan los fundamentos de la correspondencia entre proposiciones y tipos, y en particular la importancia de los teoremas de preservación de tipos y normalización, tanto desde el punto de vista lógico-filosófico como desde el punto de vista lógico-matemático.
- Conozcan las principales teorías en las que se basan los asistentes de demostración y los principios lógicos que usan: lógica clásica vs. intuicionista, teoría de conjuntos vs. teoría de tipos, igualdad extensional vs. intensional, predicatividad vs. impredicatividad, relevancia vs. irrelevancia de demostraciones, axioma K.
- Se familiaricen con los conceptos teóricos necesarios para usar demostradores interactivos, y especialmente con las nociones de producto y suma dependiente, el tipo identidad, la noción de equivalencia de tipos y algunas nociones provenientes de la teoría homotópica de tipos (n-types y principio de univalencia).
- Adquieran experiencia práctica a través de la formalización de problemas de pequeña o mediana escala.

Temario:

1. Cálculo-lambda tipado. Alfa, beta y eta-equivalencia. Correspondencia entre proposiciones y tipos. Confluencia, preservación de tipos y normalización del cálculo-lambda. Polimorfismo y sistema F. Normalización del sistema F a través de la técnica de candidatos de reducibilidad.
2. Teoría de tipos de Martin-Löf y cálculo de construcciones. Producto y suma dependiente. Tipos inductivos. Tipo identidad. Inducción en caminos. Jerarquía de universos, predicatividad y acumulatividad.
3. Nociones de transporte y homotopía. Isomorfismo y equivalencia de tipos. Nociones de equivalencia de tipos: quasi-inversas, semi-adjunciones, transformaciones bi-inversibles, y fibras contráctiles. Equivalencia entre diversas nociones de equivalencia de tipos. Teorema de Eckmann-Hilton. Axioma de univalencia. Univalencia implica extensionalidad. Técnica code/decode.
4. Noción de n-types: proposiciones, truncamientos y tipos contráctiles. Tipos decidibles y teorema de Hedberg. Higher-inductive types.
5. Verificación de programas y formalización de teoremas usando asistentes de demostración interactivos basados en teoría de tipos dependientes (p. ej. Coq, Agda o Lean):
 - Inducción sobre tipos de datos inductivos, familias de tipos y proposiciones definidas inductivamente.
 - Tipo de la igualdad como una familia de tipos y razonamiento ecuacional.
 - Manejo de los conectivos lógicos: conjunción, disyunción, implicación, negación, cuantificadores.
 - Tácticas y técnicas de meta-programación.

- Procedimientos de decisión y técnicas de reflexión.

Bibliografía:

- Benjamin C. Pierce y otros. *Software Foundations series, volume 1: Logical Foundations*. Versión 6.7, disponible en <https://softwarefoundations.cis.upenn.edu/>. Electronic textbook, 2024.
- Egbert Rijke. *Introduction to Homotopy Type Theory*. Cambridge University Press, por aparecer en 2025. Version preliminar disponible en <https://arxiv.org/abs/2212.11082>
- The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Disponible en <https://homotopytypetheory.org/book/>. Institute for Advanced Study, 2013.
- Philip Wadler, Wen Kokke y Jeremy G. Siek. *Programming Language Foundations in Agda*. Disponible en <https://plfa.inf.ed.ac.uk/22.08/>. 2022.
- Morte Heine Sørensen y Paweł Urzyczyn. *Lectures on the Curry-Howard isomorphism*. Elsevier, 2006.
- Yves Bertot, Pierre Castéran. *Coq'Art: The Calculus of Inductive Constructions*. Springer, 2004.
- Leonardo de Moura y otros. The Lean Theorem Prover. *Automated Deduction - CADE-25, 25th International Conference on Automated Deduction, Berlin, Germany, August 1-7, 2015, Proceedings*. 2015.