ANEXO

.

PROGRAMA DE ASIGNATURA	
ASIGNATURA: Formalización de Matemática y Ciencia de la Computación en Asistentes de Prueba.	**AÑO:** 2023
CARACTER: Optativa	**UBICACIÓN EN LA CARRERA:** 5° año 2° cuatrimestre
CARRERA: Licenciatura en Ciencias de la Computación	
REGIMEN: Cuatrimestral	**CARGA HORARIA:** 120 Horas.

FUNDAMENTACIÓN Y OBJETIVOS

Un artículo reciente de matemáticos y computólogos (Bayer et al, 2022) pone en cuestión la noción de prueba matemática: “Una prueba es uno de los conceptos más importantes de la matemática. Sin embargo, hay una diferencia muy importante entre cómo se define una prueba en teoría y cómo se usa en la práctica. Esto pone en peligro el estatus único de la matemática como una ciencia exacta.”

A medida que la sofisticación de la matemática avanza, es cada vez más difícil tener certeza sobre la corrección de la validez de las pruebas. ¡Aún si han sido revisadas y publicadas en revistas de primer nivel! De allí la afirmación de la cita. Vladimir Voevodsky (2014) se asustó ante el hecho de que haya errores que pasen inadvertidos por un tiempo largo, indicando que esto sucede porque “Difícilmente se verifica en detalle un argumento técnico de un autor confiado, que es difícil de verificar y parece similar a argumentos que se saben correctos”. Más recientemente, Kevin Buzzard (2020) señala dos artículos publicados en Annals of Mathematics, uno en 2004 y el otro en 2006 con resultados contradictorios. Así como hay programas de computación que asisten en el cómputo matemático (Sage, Maple, Mathematica) también hay herramientas, llamadas “asistentes de prueba”, que asisten en la creación de pruebas matemáticas de manera de asegurar la validez de las pruebas. Hasta hace unos quince años el uso de estas herramientas estaba casi confinado a una comunidad pequeña y la mayoría de los proyectos eran liderados por personas formadas en ciencias de la computación; un caso excepcional fue el proyecto de Thomas Hales para formalizar completamente su prueba sobre la conjetura de Kepler. De unos años a esta parte la comunidad matemática ha empezado a usar los asistentes de prueba para formalizar resultados novedosos y para entender mejor sus propias construcciones.

Uno de los impedimentos para la adopción de estas herramientas por parte de la comunidad matemática era la falta de un corpus grande con los requisitos obvios para encarar la formalización de estructuras sofisticadas. Es decir, quien quisiera formalizar algo debía empezar casi desde cero. Hoy en día se cuenta con grandes bibliotecas de matemática formalizada para poder encarar proyectos interesantes.

Otra dificultad es lo que implica en sí formalizar matemática: es transcribir en un lenguaje absolutamente formal y con todo detalle las pruebas de manera que el asistente pueda comprobar que cada paso de razonamiento es válido (a partir de los axiomas de la teoría en cuestión). La validez de las inferencias está dada por la lógica subyacente al asistente; en algunos casos, se trata de una lógica intuicionista que resulta ajena a la práctica habitual. Por otro lado, algunos razonamientos “obvios” en una prueba pueden exigir, al formalizarlos, pasos aparentemente innecesarios (u obvios) para los ojos matemáticos.

Objetivos

Se espera que quien tome este curso logre conocer las teorías subyacentes a diferentes asistentes de prueba y que pueda elegir uno que se adapte a la formalización que le interese hacer.

Quien apruebe el curso tendrá la capacidad para definir estructuras matemáticas (o computacionales), enunciar propiedades sobre ellas y probar teoremas en algún asistente; se fomentará el uso de las bibliotecas disponibles en el asistente en cuestión.

Se hará hincapié en la importancia de conceptos de ingeniería de software (componentes e interfaces entre componentes) para delinear la formalización inicialmente y como guías que permitan el desarrollo modular de la formalización. En este sentido, se espera que quien apruebe el curso comprenda la ventaja de definir tácticas (y el uso de meta-programación en general) para resolver tareas repetitivas.

CONTENIDO

1. Fundamentos

Repaso de lógica de primer orden: términos, fórmulas, reglas de inferencia. Verificación mecánica de pruebas matemáticas: codificación de sistemas de prueba en Isabelle/ZF como ejemplo minimal. La arquitectura de LCF. Sistema simple de tipos. Presentación de Isabelle/HOL.

2. Teoría de tipos dependientes para mecanización de matemática

La correspondencia de Curry-Howard. La igualdad en Teoría de Tipos. Presentación de Agda, Coq y Lean. Tipos inductivos y Familias de tipos. Inducción y recursión. Registros. Definiciones por pattern-matching.

3. Formalización de teorías matemáticas

Tácticas, secciones, variables. Type-classes. Coerciones. Meta-programación y definición de tácticas propias. Bibliotecas de matemática formalizadas existentes.

4. Proyecto individual

Presentación de propuesta inicial con la estructura de módulos. Definición de los componentes de manera abstracta, enunciado de teoremas más importantes y esquema de prueba. Avance con lemas conducentes a la prueba de los teoremas.

BIBLIOGRAFÍA

BIBLIOGRAFÍA BÁSICA

Jeremy Avigad, Leonardo de Moura, y Soonho Kong. Theorem Proving in Lean. Electronic textbook, 2021. https://leanprover.github.io/theorem_proving_in_lean/

Jeremy Avigad, Kevin Buzzard, Robert Y. Lewis, Patrick Massot. Mathematics in Lean. 2020. https://leanprover-community.github.io/mathematics_in_lean

Yves Bertot y Pierre Castéran. Interactive Theorem Proving and Program Development - Coq'Art: The Calculus of Inductive Constructions. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004. doi: 10.1007/978-3-662-07964-5.

Kevin Buzzard. Formalising Undergraduate Mathematics. Presentación en CICM13. 2020.

Assia Mahboubi y Enrico Tassi. Mathematical Components. Zenodo, Sept. 2022. doi: 10.5281/zenodo.7118596.

Tobias Nipkow y Gerwin Klein. Concrete Semantics - With Isabelle/HOL. Springer, 2014. doi: 10.1007/978-3-319-10542-0.

Vladimir Voevodsky. Univalent Foundations. Presentación en el Institute of Advanced Studies. 2014. Disponible en: https://www.math.ias.edu/~vladimir/Site3/Univalent_Foundations_files/2014_IAS.pdf

Philip Wadler, Wen Kokke, and Jeremy G. Siek. Programming Language Foundations in Agda. Aug. 2022.

BIBLIOGRAFÍA COMPLEMENTARIA

John Harrison. Handbook of Practical Logic and Automated Reasoning. Cambridge University Press 2009, ISBN 978-0-521-89957-4

Benjamin C. Pierce et al. Logical Foundations. Ed. by Benjamin C. Pierce. Vol. 1. Software Foundations. Version 6.3, http://softwarefoundations.cis.upenn.edu. Electronic textbook, 2023.

Egbert Rijke. Introduction to Homotopy Type Theory. Pre-print. doi: 10.48550/arXiv.2212.11082. 2023.

The Univalent Foundations Program. Homotopy Type Theory: Univalent Foundations of Mathematics. Institute for Advanced Study, 2013.

EVALUACIÓN

FORMAS DE EVALUACIÓN

La evaluación será a través de presentaciones que demuestren el avance de un proyecto de formalización. Se prevén tres presentaciones:

- Presentación del proyecto de formalización
- Presentación de avance de formalización
- Presentación al final del curso/momento de rendir

Las primeras dos presentaciones constituyen los Trabajos Prácticos previstos durante el cursado. La tercera, tendrá la forma de coloquio, para quienes estén en condiciones de promocionar, o de examen final escrito y oral, para quienes hayan regularizado, en cuyo caso deberá contemplar las observaciones recibidas en las dos instancias de evaluación anteriores.

El desarrollo de un trabajo de formalización es fundamental para la aprobación del curso. Por ello, quienes deseen aprobarlo en condición de estudiantes libres deberán necesariamente presentar, tres meses antes de rendir, una propuesta de proyecto de formalización que incluya un cronograma de trabajo y presentaciones de avance. La presentación final será la evaluación final escrita y oral.

REGULARIDAD

Aprobar los dos Trabajos Prácticos.

PROMOCIÓN

Aprobar los dos Trabajos Prácticos con una nota no menor a 6 (seis) y aprobar un coloquio.

CORRELATIVIDADES

- para cursar: Introducción a la Lógica y la Computación (aprobada) y Lenguajes Formales y Computabilidad (regularizada)
- para rendir: Introducción a la Lógica y la Computación (aprobada) y Lenguajes Formales y Computabilidad (aprobada)