Propuesta de Enseñanza de la Formalización de la Matemática utilizando un Asistente de Pruebas en Estudiantes de la Licenciatura en Ciencias de la Computación[⋆]

Daniel Severín[1,4][0000−0003−2391−3489] and Alejandro Hernández[2,3]

[1] Depto. de Matemática, FCEIA, Universidad Nacional de Rosario, Argentina
[2] Depto. de Cs. de la Computación, FCEIA, Universidad Nacional de Rosario, Argentina
[3] CAETI, Universidad Abierta Interamericana, Argentina
[4] CONICET, Argentina

Abstract. En este trabajo proponemos realizar una actividad con estudiantes de Cs. de la Computación en donde se plantea la importancia de la formalización de la Matemática, y su abordaje a través del asistente de pruebas *Coq*. El objetivo es lograr que los estudiantes conozcan la existencia de este tipo de herramientas, adquieran destreza en formalizar las definiciones y enunciados, y puedan realizar pruebas formales de enunciados sencillos.

Keywords: Propuesta de Enseñanza · Formalización de la Matemática · Coq · Funciones Recursivas Primitivas.

1 Introducción

Durante la década del '70, comenzó un ambicioso proyecto denominado *Mizar Project* con el propósito de reescribir el actual estado de la Matemática en un lenguaje formal que pueda ser entendible para un matemático y, a la vez, procesable por una computadora. Algunas falencias de Mizar dieron lugar a que, en décadas posteriores, otros asistentes de pruebas se consolidaran [1]. Uno de ellos es *Coq* [2], el cual es una implementación de un tipo de lógica intuicionista de alto orden con tipos dependientes e inductivos. Aquí, el usuario introduce definiciones y hace demostraciones en un estilo de deducción natural, que luego son chequeadas mecánicamente por el sistema.

Una posible aplicación de los asistentes de pruebas surge cuando es inevitable demostrar un resultado a través de una separación en una gran cantidad de casos. Un ejemplo clásico es el Teorema de los Cuatro Colores que afirma que un grafo planar es 4-coloreable, y lo reduce a la verificación de miles de casos relativamente mecánicos [3]. Los autores resolvieron este inconveniente escribiendo un programa informático que verifica los casos exhaustiva y automáticamente.

[⋆] Este trabajo es subsidiado con el proyecto PID-UNR 80020180100091UR.

La desventaja es que un lector o revisor que quiera *convencerse* de la correctitud del teorema debería *confiar* en la correctitud de la implementación del programa y del compilador utilizado (y, posiblemente, del sistema operativo y equipo físico donde el programa se ejecuta). Esto dio lugar a que parte de la comunidad de matemáticos no consideraran en aquel momento este resultado como un "teorema matemático" aduciendo que no es posible chequearlo *a mano* [4].

Otro enfoque que reduce esta brecha de confianza consiste en generar la prueba de los casos en un lenguaje formal, la cual puede ser certificada independientemente por un asistente de pruebas. Este nuevo enfoque permite que el lector solo deba reducir su confianza al asistente de pruebas. La formalización del Teorema de los Cuatro Colores en Coq es justamente un ejemplo de ello [5].

La *Formalización de la Matemática* es una rama (actualmente de las Ciencias de la Computación) que aun no se ha desarrollado lo suficiente como para que los matemáticos la utilicen con frecuencia. El principal inconveniente es la rigidez de los lenguajes de los asistentes de pruebas para expresar conceptos, ideas y deducciones matemáticas. Sin embargo, como ha pasado con otro tipo de tecnologías, eventualmente los matemáticos las incorporan, ya sea procesadores de texto (e.g. LaTeX), lenguajes matemáticos de cálculo numérico (e.g. Matlab, Scilab) o simbólico (e.g. Mathematica, Maple, SageMath).

Asimismo, existen factores que permiten predecir que en pocas décadas se podrían imponer los asistentes de pruebas [6]: un factor está relacionado a la *digitalización* del actual conocimiento matemático, ya que así es posible aplicar herramientas de Búsqueda y Data Mining a esta base de conocimiento; el otro factor, a la *creación* del nuevo conocimiento matemático, por ejemplo modificando los protocolos de revistas para tener en cuenta una estandarización en la formalización de pruebas durante el proceso de *peer-review*.

Hay ejemplos concretos de la necesidad de contar con la formalización de pruebas extensas como lo es la clasificación de grupos finitos simples: un triunfo del siglo XX que involucra 10000 páginas de prueba a lo largo de 500 artículos de revistas realizado por más de 100 autores diferentes. Esto se volvió un problema ya que el entendimiento completo de la prueba sólo recae en muy pocas personas. Aunque hay un consenso entre matemáticos que consideran "completa" a esta clasificación de grupos finitos, los mismos matemáticos no están del todo seguros dada la complejidad y el tamaño de la misma [7].

Durante un congreso internacional sobre asistentes de pruebas celebrado en 2019, uno de los autores de este trabajo (D. Severín) asistió a la charla de un matemático reconocido, el Dr. Kevin Buzzard, del Imperial College London [8], quien lleva adelante un proyecto [9], donde plantea la importancia de enseñar asistentes de pruebas a estudiantes de matemática que cursan su ciclo básico debido al potencial uso que podrían tener en un futuro no lejano. También plantea la necesidad de que los Departamentos de Matemática cuenten con investigadores formados en Ciencias de la Computación para que colaboren junto a los matemáticos en la formalización de resultados.

2 Propuesta

Se propone realizar una actividad con estudiantes de Ciencias de la Computación de la Fac. de Cs. Exactas, Ingeniería y Agrimensura de la Universidad Nacional de Rosario que cursan la materia *Lenguajes Formales y Computabilidad*. Cabe destacar que actualmente, en la currícula de la carrera de Cs. de la Computación, se aborda el uso de asistentes de pruebas (Unidad III: Especificaciones formales, de la materia *Ingeniería de Software I*). No obstante, su uso se limita a otro contexto donde la meta es especificar y verificar un sistema informático mediante Métodos Formales, diferente del propuesto aquí que es el uso de asistentes de pruebas para la formalización de conceptos y enunciados matemáticos.

El objetivo de la actividad es lograr que los estudiantes conozcan la existencia de herramientas para realizar pruebas formales y que son de libre distribución, adquieran destreza en formalizar las definiciones y enunciados, y puedan realizar pruebas formales de enunciados sencillos.

La evaluación de esta actividad permitirá adquirir experiencia con el fin de analizar la factibilidad de enseñar estos conocimientos, y de ser así, plantear la posibilidad de extenderla a otras materias de la Lic. en Cs. de la Computación, y eventualmente a la Lic. en Matemática.

Durante todo el proceso, los estudiantes recibirán el apoyo de, entre otros, el Dr. Hernández quien actualmente dicta la práctica de dicha materia, y estará supervisada por el Dr. Severín. La actividad consiste en:

- Realizar una exposición que explique de qué se trata la Formalización de la Matemática, resaltando su importancia y el rol de los asistentes de pruebas en ella. Esta exposición continúa con los conocimientos mínimos necesarios para que los estudiantes puedan instalar y utilizar Coq.
- Desarrollar una clase teórico/práctica en un laboratorio, donde se les proporcionará un archivo Coq incompleto y ellos deberán rellenarlo. En esta clase se explicará brevemente la sintaxis de Coq, algunos comandos (`Check`, `Definition`, `Eval`, `Proposition`) y tácticas básicas (`intros`, `unfold`, `fold`, `reflexivity`, `apply`, `induction`, `rewrite`). Luego se les propondrá a los estudiantes formalizar los enunciados de algunos ejercicios seleccionados de la práctica de *Funciones Recursivas Primitivas* y realizar pequeñas pruebas, con la asistencia de los docentes involucrados.
- Prestar especial atención al momento en que los estudiantes deban abordar la formalización de pruebas por inducción matemática, ya que se prevee que éste será el mayor obstáculo.
- El material que se utilizará consiste en un archivo Coq (con codificación UTF-8) que se puede encontrar en el siguiente link:

  https://www.fceia.unr.edu.ar/~daniel/stuff/Coq/frp.v

  Este material está organizado de forma que el estudiante puede aprender casi de forma autónoma varios conceptos mediante la observación de ejemplos y propuesta de ejercicios con dificultad creciente. También está redactado de forma que funcione como guía para desarrollar la clase.

– La modalidad de evaluación es a través de la presentación de un trabajo práctico donde los estudiantes resuelven algunos ejercicios previamente seleccionados del material entregado.

La práctica de *Funciones Recursivas Primitivas* fue escogida debido a que el tema que aborda es autocontenido, no necesitándose formalizar nociones previas. Dado que las funciones son sólo a valores naturales, los únicos *tipos* que aparecen son el de los número naturales y funciones sobre los mismos (para funciones de 1 argumento es $\mathbb{N} \to \mathbb{N}$, para 2 argumentos es $\mathbb{N} \to \mathbb{N} \to \mathbb{N}$ y así sucesivamente), por lo que se evita lidiar con conceptos más avanzados de Teoría de Tipos. Además, las únicas técnicas utilizadas para probar resultados son la de inducción matemática y reescritura de términos, no siendo necesario realizar otro tipo de manipulación simbólica ni deducciones lógicas avanzadas. En particular, es natural definir la recursión primitiva en Coq [10].

Finalmente, las definiciones en el archivo Coq fueron realizadas de forma que se puede escribir una función recursiva primitiva muy similar a la mostrada en la clase de teoría, y *gratuitamente* puede chequearse su sintaxis y ser evaluada. Por ejemplo, la suma, que se define con la expresión simbólica $\Sigma^{(2)} := R(p_1^{(1)}, \varphi(s^{(1)}, p_3^{(3)}))$, puede ser definida y evaluada numéricamente:

```
Definition Σ := R2 p1 1 (φ1 3 s p3 3).
Eval compute in Σ 5 7.
          = 12 : N
```

pero si uno ingresa una fórmula inválida, por ejemplo si las aridades de las funciones no coinciden, Coq la va a rechazar. Esto permite al estudiante contar con una herramienta tecnológica para revisar las funciones que defina.

References

1. F. Wiedijk. *Formal Proof - Getting Started*. Notices of the AMS **55** (2008) 1408–1414.
2. *The Coq Proof Assistant*. https://coq.inria.fr/
3. K. Appel, W. Haken. *Every Planar Map is Four Colorable*. Illinois Journal of Mathematics **21** (1977) 429–567 (1977).
4. E. R. Swart. *The philosophical implications of the four-color problem*. American Mathematical Monthly **87** (1980) 697–702.
5. G. Gonthier. *Formal Proof - The Four-Color Theorem*. Notices of the AMS **55** (2008) 1382–1393.
6. J. Harrison. *Formal Proof - Theory and Practice*. Notices of the AMS **55** (2008) 1395–1406.
7. M. Freiberger. *Pure maths in crisis?* (Interview to K. Buzzard) https://plus.maths.org/content/pure-maths-crisis
8. K. Buzzard. *What makes a mathematician tick?* Plenary session. International Theorem Proving (ITP 2019), Portland OR, USA. https://itp19.cecs.pdx.edu/wp-content/uploads/2019/09/buzzard-slides.pdf
9. K. Buzzard. *Xena Project*. https://xenaproject.wordpress.com/
10. C. D. Luna. *Enseñando Métodos Formales con Coq*. Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología **1** (2006) 55–64.