A system for computing and reasoning in Algebraic Topology*

Jónathan Heras, Vico Pascual, and Julio Rubio

Department of Mathematics and Computer Science of University of La Rioja
{jonathan.heras, vico.pascual, julio.rubio}@unirioja.es

Abstract. In this paper we present the *fKenzo* system, an *integral* assistant for teaching and research in (a subset of) Algebraic Topology. The *fKenzo* system provides a friendly graphical user interface which allows the user to interact with both the *Kenzo* and *GAP* Computer Algebra systems and, also, with the *ACL2* Theorem Prover by means of an intermediary layer based on XML technology.

System Description

Algebraic Topology is a mathematical subject which studies topological spaces using algebraic means, in particular through algebraic invariants (groups or rings, usually). This allows one to study interesting properties about topological spaces using statements about groups which are often easier to prove.

The *fKenzo* system [6] has been developed with the aim of being an *integral* assistant for research and teaching in (a subset of) Algebraic Topology. The "integral" adjective means that this assistant not only provides a graphical interface for using computational kernels, but also guides the user in his interaction with the system, and as far as possible, produces certificates about the correctness of the computations performed.

The *fKenzo* system provides a friendly front-end allowing the interoperability among different sources for computation and deduction by means of an intermediary layer based on XML technology.

The main computational kernel of our system is *Kenzo* [2], a Common Lisp program devoted to Symbolic Computation in Algebraic Topology which was developed by F. Sergeraert, which allows an *fKenzo* user to compute homology and homotopy groups of spaces. In addition, *GAP* [1], a Computer Algebra system well-known for its contributions in the area of Computational Group Theory, and its *HAP* package [3], an homological algebra library developed by G. Ellis, have been integrated in *fKenzo* allowing computations related to group homology. From the theorem proving side, *ACL2* [7], a first order logic theorem prover tool, is the core for verifying the correctness of statements.

In addition, we can say that the final aim of *fKenzo* has consisted not only in integrating several Computer Algebra systems and Theorem Prover tools, and

* Partially supported by Ministerio de Educación y Ciencia, project MTM2009-13842-C02-01

use them individually by means of a common GUI, but also in making them work in a coordinate and collaborative way to obtain new tools and results not reachable if we use individually each system.

As an example of this interoperability among systems, inspired by the work presented in [8], *Kenzo* and *GAP* have been combined in *fKenzo* to construct some spaces, namely Eilenberg MacLane spaces of type $K(G,1)$ where G is a cyclic group, which are instrumental in the computation of homotopy groups. The methodology presented in [8] to compute the homology groups of those Eilenberg MacLane spaces can be summed up as follows:

1. Load the necessary packages and files in GAP and Kenzo,
2. build the cyclic group G in GAP,
3. build a resolution of the cyclic group G using the HAP package,
4. export from GAP the resolution into a file using the OpenMath format,
5. import the resolution to Kenzo,
6. build the cyclic group G in Kenzo (thanks to a new Kenzo module developed in [8]),
7. assign the resolution to the corresponding cyclic group G in Kenzo,
8. build the space $K(G,1)$ where G is the cyclic group in Kenzo,
9. compute the homology groups of $K(G,1)$.

This approach has some drawbacks. First of all, the user must install several programs and packages: GAP, its HAP package, the OpenMath package for GAP [9], an extension for this OpenMath package developed in [8], the Kenzo system and the new module developed in [8]. In addition, of course, the user must know how to mix all the ingredients in order to obtain the desired result. Moreover, some of the steps could be performed automatically by a computer program; for instance, the importation/exportation of the resolution from GAP to Kenzo.

On the contrary, the procedure that the user must follow using *fKenzo* is:

1. Load the GAP *fKenzo* module,
2. build the cyclic group G,
3. build the space $K(G,1)$,
4. compute the homology groups of $K(G,1)$.

As can be seen, this is a much simpler approach than the one presented in [8] from the user point of view. To deal with the importation/exportation of the resolution from GAP to Kenzo, the SCSCP protocol [4] has been used.

Moreover, the reliability of such construction is increased by means of the *ACL2* Theorem Prover. Namely, ACL2 is invoked from *fKenzo* to generate a certificate of the correctness of the implementation of the cyclic group G which is used as input to construct the Eilenberg MacLane space $K(G,1)$. Therefore, we can claim that *Kenzo*, *GAP* and *ACL2* work together to provide a powerful and reliable tool thanks to the *fKenzo* system.

We urge the interested reader to consult [5] where he can find several demos, related papers and a complete system description of *fKenzo*.

References

1. GAP - Groups, Algorithms, Programming - System for Computational Discrete algebra. `http://www.gap-system.org`.
2. X. Dousson, J. Rubio, F. Sergeraert, and Y. Siret. The Kenzo program. Institut Fourier, Grenoble, 1998. `http://www-fourier.ujf-grenoble.fr/~sergerar/Kenzo/`.
3. G. Ellis. HAP package for GAP, 2009. `http://www.gap-system.org/Packages/hap.html`.
4. S. Freundt, P. Horn, A. Konovalov, S. Lindon, and D. Roozemond. Symbolic Computation Software Composability Protocol (SCSCP) specification, version 1.3, 2009. `http://www.symbolic-computation.org/scscp`.
5. J. Heras. The *fKenzo* program. University of La Rioja, 2010. `http://www.unirioja.es/cu/joheras/fKenzo/`.
6. J. Heras, V. Pascual, J. Rubio, and F. Sergeraert. *fKenzo*: A user interface for computations in Algebraic Topology. *Journal of Symbolic Computation*, 46(6):685–698, 2011.
7. M. Kaufmann and J. S. Moore. ACL2. `http://www.cs.utexas.edu/users/moore/acl2/`.
8. A. Romero, G. Ellis, and J. Rubio. Interoperating between computer algebra systems: computing homology of groups with Kenzo and GAP. In *Proceedings 34th International Symposium on Symbolic and Algebraic Computation (ISSAC'2009)*, pages 303–310, 2009.
9. A. Solomon and M. Costantini. GAP package OpenMath, 2009. `http://www.gap-system.org/Packages/openmath.html`.