SYMBOLIC MANIPULATION AND BIOMEDICAL IMAGES

JÓNATHAN HERAS, GADEA MATA, MARÍA POZA, AND JULIO RUBIO

ABSTRACT. Discrete Morse theory is an efficient method which allows one to study the topology of discrete objects. The instrumental tool in the algebraic setting of this theory is the notion of *admissible discrete vector field*. In this paper, we present a *formally verified* implementation of an algorithm in charge of building an admissible discrete vector field from a *digital image*. Such a program will play a key role to analyze *biomedical images*.

INTRODUCTION

Symbolic manipulation is a field of Computer Science which consists of two branches: *Computer Algebra* and *Mechanized Reasoning* based on computational logic. The *Calculemus* initiative [2] set out the challenge of merging these two lines of symbolic computation through the integration of Computer Algebra (computation) and Mechanized Reasoning (deduction) systems.

In this paper, we present a particularization of this integration to the scope of Algebraic Digital Topology. In this discipline the use of homological methods to analyze digital images has been broadly applied, see, for instance, [12]. In our case, we introduce the implementation of an algorithm removing some elements from a digital image but keeping the same homological information. The integration of computation and deduction stems from the fact that the correctness of our implementation has been *formally verified* by means of an *interactive* Proof Assistant.

The feasibility of applying our programs in real situations has been tested through the study of some *biomedical images*. In those examples, and thanks to our programs, the computational times are dramatically reduced.

1. THE REDUCTION ALGORITHM

The task of extracting homological information from a digital image can be accomplished through matrix operations (rank, diagonalization, ...) of some matrices, called *boundary* (or *incidence*) matrices, associated with the image. A detailed description of the procedure to obtain boundary matrices from an image can be seen in [9].

However, in some cases those matrices are so huge that makes direct computations impractical. Therefore, a method to reduce the size of the matrices but preserving homology is appealing. Our reduction process is based on *Discrete Morse Theory* [5]; namely, we work in the algebraic setting presented in [13]. There, an algorithm (from now on, called RS's algorithm; RS stands for Romero–Sergeraert) to construct an *admissible discrete vector field* is explained. Roughly speaking, an admissible discrete vector field, associated with an image,

Partially supported by Ministerio de Educación y Ciencia, project MTM2009-13842-C02-01, and by the European Union's 7th Framework Programme under grant agreement nr. 243847 (ForMath).

is a *recipe* which indicates both the "useless" elements of the image (in the sense, that they can be removed without changing its homology) and the *critical* ones (those whose removal modifies the homology).

The input of the RS algorithm is a boundary matrix and its output is an admissible discrete vector field. From the boundary matrix and the vector field, a new matrix (smaller than the original one) can be obtained. The importance of this new matrix is twofold: on the one hand, we can obtain the same homological information from the two matrices (this was proved in [13, Theorem 19]); and, on the other hand, the new matrix is much smaller than the original one (in addition, in most of the cases it is null or empty). Therefore, we can use the matrix produced by the RS algorithm to obtain information about the digital image in a much faster way than using the boundary matrix.

It is not the aim of this paper to provide a detailed description of the RS algorithm; on the contrary, we provide some details about our implementation of that algorithm, its corresponding verification and its use to handle huge images.

2. From Computation to Deduction through Testing

The development of a formally certified implementation of the RS algorithm has followed the methodology presented in [11]. Firstly, we implement a version of our programs in *Haskell* [10], a *lazy* functional programming language. Subsequently, we intensively test our implementation using *QuickCheck* [3], a tool which allows one to automatically test properties about programs implemented in Haskell. Finally, we verify the correctness of our programs using the Coq *interactive* Proof Assistant [4] and its SSReflect library [6].

The choice of Haskell to implement our programs was because both the code and the way of working is similar to the ones of the Coq formal proof management system. In this programming language, we have defined the functions `gen_adm_dvf` and `matrixReduced` which implement, respectively, the RS algorithm and the reduction process (the latter function invokes the former one to generate an admissible discrete vector field from a matrix and, subsequently, obtains a reduced matrix).

The usage of QuickCheck can be considered as a good starting point towards the formal verification of our programs. On the one hand, a specification of the properties which must be satisfied by our programs is given (a necessary step in the formalization process). On the other hand, before trying a formal verification of our programs (a quite difficult task) we are testing them, a process which can be useful in order to detect bugs. Let us present an example: using a Haskell function called `admissible` we have specified the admissibility property; now, we can test whether `gen_adm_dvf` satisfies such property.

```
> quickCheck M -> admissible (gen_adm_dvf M)
+ + + OK, passed 100 tests.
```

The above display must be read as follows. In the first line we state that given a matrix `M`, the output of `gen_adm_dvf` fulfills the specification of the property called `admissible`. The second line, which is the result produced by QuickCheck when evaluating the statement of the first line, means that QuickCheck has generated 100 random values for `M`, checking that the property was true for all these cases.

After testing our programs, and as final step to confirm their reliability, we can undertake the challenge of formally certify their correctness. To this aim, we must translate both the

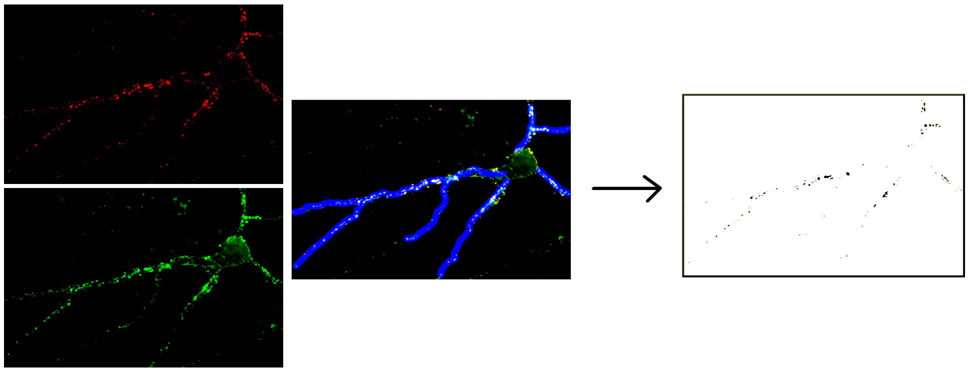

FIGURE 1. Synapses extraction from three images of a neuron

programs and the properties from Haskell to COQ, a task which is quite direct since these two systems are close. Then, we have to prove lemmas like the following one.

```
Lemma admissibility_gen_adm_dvf: forall M, admissible (gen_adm_dvf M)
```

In particular, the above lemma states the admissibility property for `gen_adm_dvf`. This lemma and the one stating that `gen_adm_dvf` constructs a discrete vector field have been proved in COQ; thereby, our implementation of the RS algorithm is correct.

3. APPLICATION TO BIOMEDICAL IMAGES

Biomedical images are a suitable benchmark for testing our programs. On the one hand, the amount of information included in this kind of images is usually huge; then, a process able to reduce those images but keeping the homological properties can be really useful. On the other hand, software systems dealing with biomedical images must be trustworthy; this is our case since we have formally verified the correctness of our programs.

As an example, we can consider the problem of counting the number of *synapses* in a neuron. Synapses [1] are the points of connection between neurons and are related to the computational capabilities of the brain. Therefore, the treatment of neurological diseases, such as Alzheimer, may take advantage of procedures modifying the number of synapses.

Up to now, the study of the *synaptic density evolution* of neurons was a time-consuming task since it was performed, mainly, manually. To overcome this issue, an automatic method was presented in [8]. Briefly speaking, such process can be split into two parts. Firstly, from three images of a neuron (the neuron with two antibody markers and the structure of the neuron), a monochromatic image is obtained, see Figure 1[1]. In such an image, each connected component represents a synapse. So, the problem of measuring the number of synapses is translated into a question of counting the connected components of a monochromatic image.

In the context of Algebraic Digital Topology, this issue can be tackled by means of the computation of the homology group in degree 0 of the monochromatic image. This task can be performed in COQ through the formally verified programs which were presented in [7]. Nevertheless, such programs are not able to handle images like the one of the right side of

[1] The same images with higher resolution can be seen in http://www.unirioja.es/cu/joheras/synapses/

Figure 1 due to its huge size (it is worth remarking that Coq is a Proof Assistant tool and not a Computer Algebra system). In order to overcome this drawback, we can integrate our reduction programs with the ones presented in [7]. Using this approach, we can successfully compute the homology in just 25 seconds, an impressive time for an execution inside Coq.

4. Conclusions and Further work

A formally verified implementation of the RS algorithm has been presented. Thanks to the integration of this tool with the certified programs devoted to obtain homological properties from digital images, we are able to deal with some biomedical images inside Coq.

Some formalization issues remain as further work; namely, we have to verify the correctness of the function `matrixReduced`; that is to say, given a matrix `M`, the result produced by `(matrixReduced M)` is a matrix homologically equivalent to `M`. Another open research line is the use of our *certified* programs in the study of more complex biomedical problems. For instance, in the detection of the structure of a neuron, a problem which seems to involve the computation of homology groups in higher degrees.

References

[1] M. Bear, B. Connors, and M. Paradiso. *Neuroscience: Exploring the Brain.* Lippincott Williams & Wilkins, 2006.

[2] The Calculemus Project. `http://www.calculemus.net/`.

[3] K. Claessen and J. Hughes. QuickCheck: A Lightweight Tool for Random Testing of Haskell Programs. In *ACM SIGPLAN Notices*, pages 268–279. ACM Press, 2000.

[4] Coq development team. The Coq Proof Assistant, version 8.3. Technical report, 2010.

[5] R. Forman. Morse theory for cell complexes. *Advances in Mathematics*, 134:90–145, 1998.

[6] G. Gonthier and A. Mahboubi. An introduction to small scale reflection in Coq. *Journal of Formal Reasoning*, 3(2):95–152, 2010.

[7] J. Heras, M. Dénès, G. Mata, A. Mörtberg, M. Poza, and V. Siles. Towards a certified computation of homology groups for digital images. In *Proceedings 4th International Workshoph on Computational Topology in Image Context*, To appear in Lectures Notes in Computer Science, 2012.

[8] J. Heras, G. Mata, M. Poza, and J. Rubio. Homological processing of biomedical digital images: automation and certification. In *17th International Conferences on Applications of Computer Algebra. Computer Algebra in Algebraic Topology and its applications session*, 2011.

[9] J. Heras, M. Poza, M. Dénès, and L. Rideau. Incidence simplicial matrices formalized in Coq/SSReflect. In *Proceedings 18th Symposium on the Integration of Symbolic Computation and Mechanised Reasoning (Calculemus'2011)*, volume 6824 of *Lectures Notes in Computer Science*, pages 30–44, 2011.

[10] S. P. Jones et al. The Haskell 98 language and libraries: The revised report. *Journal of Functional Programming*, 13(1):0–255, 2003. `http://www.haskell.org`.

[11] A. Mörtberg. Constructive algebra in functional programming and type theory. Mathematics, Algorithms and Proofs 2010, 2010. `http://www.unirioja.es/dptos/dmc/MAP2010/program.shtml`.

[12] M. Mrozek et al. Homological methods for extraction and analysis of linear features in multidimensional images. *Pattern Recognition*, 45(1):285–298, 2012.

[13] A. Romero and F. Sergeraert. Discrete Vector Fields and Fundamental Algebraic Topology, 2010. `http://arxiv.org/abs/1005.5685v1`.

Departamento Matemáticas y Computación. Universidad de La Rioja. Logroño (La Rioja, Spain).
E-mail address: `{jonathan.heras,maria.poza,julio.rubio}@unirioja.es`

Laboratorio de Plasticidad Sináptica Estructural. Centro de Investigación Biomédica de La Rioja. Logroño (La Rioja, Spain).
E-mail address: `gmata.ext@riojasalud.es`