Symbolic Manipulation and Biomedical Images*

Jónathan Heras, Gadea Mata, María Poza, and Julio Rubio

Department of Mathematics and Computer Science
University of La Rioja
Spain

June 13, 2012

*Partially supported by Ministerio de Educación y Ciencia, project MTM2009-13842-C02-01, and by European Commission FP7, STREP project ForMath, n. 243847

Table of Contents

1 Motivation

Table of Contents

1. Motivation
2. Mathematical preliminaries

Table of Contents

1. Motivation
2. Mathematical preliminaries
3. Admissible discrete vector fields

Table of Contents

1. Motivation
2. Mathematical preliminaries
3. Admissible discrete vector fields
4. From computation to verification through deduction

Table of Contents

1. Motivation
2. Mathematical preliminaries
3. Admissible discrete vector fields
4. From computation to verification through deduction
5. Application

Table of Contents

1. Motivation
2. Mathematical preliminaries
3. Admissible discrete vector fields
4. From computation to verification through deduction
5. Application
6. Conclusions and further work

Table of Contents

1. Motivation
2. Mathematical preliminaries
3. Admissible discrete vector fields
4. From computation to verification through deduction
5. Application
6. Conclusions and further work

Analysis of Biomedical Images

Problems

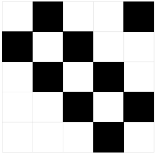
- Reliability
- Size of the images
- Huge time investment
- Many processes are repetitive tasks

Goal

Automatic, efficient and reliable methods

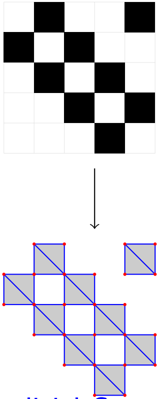
Digital Algebraic Topology

Digital Image



Digital Algebraic Topology

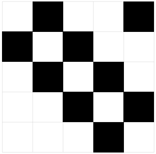
Digital Image

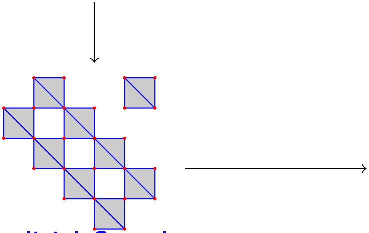


Simplicial Complex

Digital Algebraic Topology

Digital Image





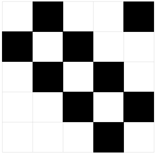
Simplicial Complex

C_0 = vertices
C_1 = edges
C_2 = triangles

Chain Complex

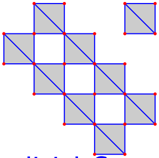
Digital Algebraic Topology

Digital Image



Homology Groups

$$H_0 = \mathbb{Z} \oplus \mathbb{Z}$$
$$H_1 = \mathbb{Z} \oplus \mathbb{Z} \oplus \mathbb{Z}$$



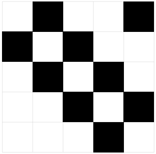
Simplicial Complex

$$C_0 = \text{vertices}$$
$$C_1 = \text{edges}$$
$$C_2 = \text{triangles}$$

Chain Complex

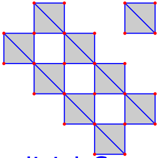
Digital Algebraic Topology

Digital Image ← **Homology Groups**



$$H_0 = \mathbb{Z} \oplus \mathbb{Z}$$
$$H_1 = \mathbb{Z} \oplus \mathbb{Z} \oplus \mathbb{Z}$$

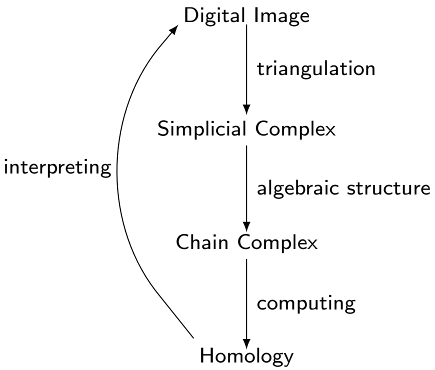
↓ ↑



Simplicial Complex → **Chain Complex**

C_0 = vertices
C_1 = edges
C_2 = triangles

The method


Digital Image
triangulation
Simplicial Complex
algebraic structure
Chain Complex
computing
Homology
interpreting

The method

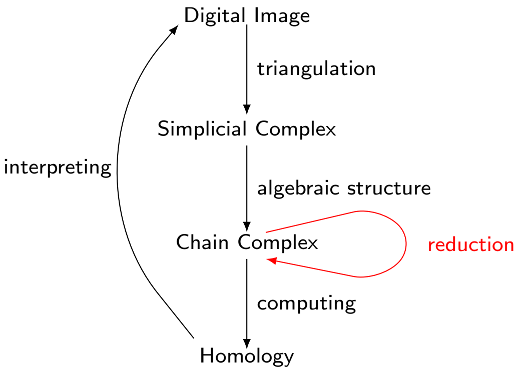

Digital Image
triangulation
Simplicial Complex
algebraic structure
Chain Complex
reduction
computing
Homology
interpreting

Table of Contents

1 Motivation

2 Mathematical preliminaries

3 Admissible discrete vector fields

4 From computation to verification through deduction

5 Application

6 Conclusions and further work

Simplicial Complex

Definition

Let V be a set, called the vertex set, a *simplex* over V is any finite subset of V

Simplicial Complex

Definition

Let V be a set, called the vertex set, a *simplex* over V is any finite subset of V

Definition

An *(abstract) simplicial complex* over V is a set of simplices C over V satisfying the property:

$$\forall \alpha \in C, \ si \ \beta \subseteq \alpha \Rightarrow \beta \in C$$

Simplicial Complex

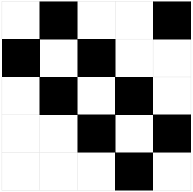
Definition

Let V be a set, called the vertex set, a *simplex* over V is any finite subset of V

Definition

An *(abstract) simplicial complex* over V is a set of simplices C over V satisfying the property:

$$\forall \alpha \in C, \ si \ \beta \subseteq \alpha \Rightarrow \beta \in C$$



Simplicial Complex

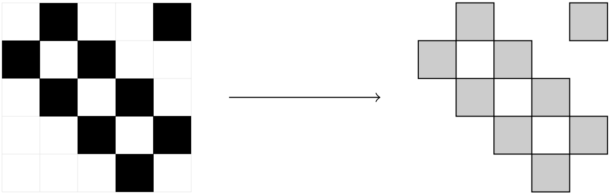
Definition

Let V be a set, called the vertex set, a *simplex* over V is any finite subset of V

Definition

An *(abstract) simplicial complex* over V is a set of simplices C over V satisfying the property:

$$\forall \alpha \in C, \; si \; \beta \subseteq \alpha \Rightarrow \beta \in C$$



Simplicial Complex

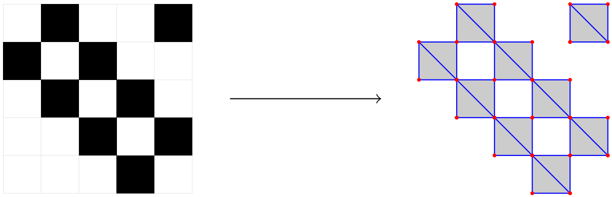
Definition

Let V be a set, called the vertex set, a *simplex* over V is any finite subset of V

Definition

An *(abstract) simplicial complex* over V is a set of simplices C over V satisfying the property:

$$\forall \alpha \in C,\ si\ \beta \subseteq \alpha \Rightarrow \beta \in C$$



Chain Complex

Definition

A chain complex C_* is a pair of sequences $C_* = (C_q, d_q)_{q \in \mathbb{Z}}$ where:

- For every $q \in \mathbb{Z}$, the component C_q is a R-module, the chain group in degree q
- For every $q \in \mathbb{Z}$, the component d_q is a morphism $d_q : C_q \to C_{q-1}$, the differential function
- For every $q \in \mathbb{Z}$, the composition $d_q d_{q+1}$ is null: $d_q d_{q+1} = 0$

Example

$$0 \leftarrow C_0 \xleftarrow{d_1} C_1 \xleftarrow{d_2} C_2 \leftarrow 0$$

Example

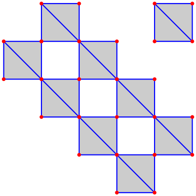
$$0 \leftarrow C_0 \xleftarrow{d_1} C_1 \xleftarrow{d_2} C_2 \leftarrow 0$$

$$\begin{array}{lcllcl} C_0 & = & \mathbb{Z}\,[\text{vertices}] & d_0(v) & = & 0 \\ C_1 & = & \mathbb{Z}\,[\text{edges}] & d_1(v_1 v_2) & = & v_2 - v_1 \\ C_2 & = & \mathbb{Z}\,[\text{triangles}] & d_2(v_1 v_2 v_3) & = & v_2 v_3 - v_1 v_3 + v_1 v_2 \end{array}$$

Example

$$0 \leftarrow C_0 \overset{d_1}{\longleftarrow} C_1 \overset{d_2}{\longleftarrow} C_2 \leftarrow 0$$

$$\begin{array}{lcl lcl} C_0 & = & \mathbb{Z}[\text{vertices}] & d_0(v) & = & 0 \\ C_1 & = & \mathbb{Z}[\text{edges}] & d_1(v_1 v_2) & = & v_2 - v_1 \\ C_2 & = & \mathbb{Z}[\text{triangles}] & d_2(v_1 v_2 v_3) & = & v_2 v_3 - v_1 v_3 + v_1 v_2 \end{array}$$



$$\longrightarrow \qquad 0 \leftarrow \mathbb{Z}^{26} \overset{d_1}{\longleftarrow} \mathbb{Z}^{36} \overset{d_2}{\longleftarrow} \mathbb{Z}^{18} \leftarrow 0$$

Homology

Definition

Let $C_* = (C_q, d_q)_{q \in \mathbb{Z}}$ be a chain complex:

- The image $B_q = im\ d_{q+1} \subseteq C_q$ is the (sub)-module of q-boundaries
- The kernel $Z_q = ker\ d_q \subseteq C_q$ is the (sub)-module de q-cycles

Definition

Let $C_* = (C_q, d_q)_{q \in \mathbb{Z}}$ be a chain complex. For every degree $n \in \mathbb{Z}$, the n-th homology group of C_* is defined as the quotient:

$$H_n(C_*) = Z_n / B_n$$

Homology

Definition

Let $C_* = (C_q, d_q)_{q \in \mathbb{Z}}$ be a chain complex:

- The image $B_q = im\ d_{q+1} \subseteq C_q$ is the (sub)-module of q-boundaries
- The kernel $Z_q = ker\ d_q \subseteq C_q$ is the (sub)-module de q-cycles

Definition

Let $C_* = (C_q, d_q)_{q \in \mathbb{Z}}$ be a chain complex. For every degree $n \in \mathbb{Z}$, the n-th homology group of C_* is defined as the quotient:

$$H_n(C_*) = Z_n / B_n$$

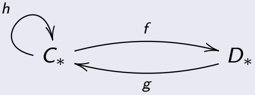
Geometrically:

- H_0 measures the number of connected components
- H_1 measures the number of holes

Reduction

Definition

A *reduction* ρ between two chain complexes C_* y D_* (denoted by $\rho : C_* \Rightarrow D_*$) is a tern $\rho = (f, g, h)$



satisfying the following relations:

1) $fg = id_{D_*}$;
2) $d_C h + h d_C = id_{C_*} - gf$;
3) $fh = 0; \quad hg = 0; \quad hh = 0.$

Theorem

If $C_* \Rightarrow D_*$, then $C_* \cong D_* \oplus A_*$, with A_* acyclic, what implies that $H_n(C_*) \cong H_n(D_*)$ for all n.

Table of Contents

1 Motivation

2 Mathematical preliminaries

3 Admissible discrete vector fields

4 From computation to verification through deduction

5 Application

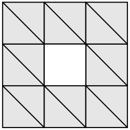
6 Conclusions and further work

Intuitive idea

- Reduce information but keeping the homological properties
- Discrete Morse Theory
  - Vector fields are a tool to cancel “useless” information

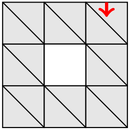
Intuitive idea

- Reduce information but keeping the homological properties
- Discrete Morse Theory
  - Vector fields are a tool to cancel “useless” information



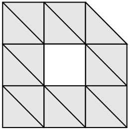
Intuitive idea

- Reduce information but keeping the homological properties
- Discrete Morse Theory
  - Vector fields are a tool to cancel "useless" information



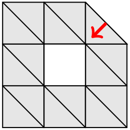
Intuitive idea

- Reduce information but keeping the homological properties
- Discrete Morse Theory
  - Vector fields are a tool to cancel "useless" information



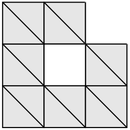
Intuitive idea

- Reduce information but keeping the homological properties
- Discrete Morse Theory
  - Vector fields are a tool to cancel "useless" information



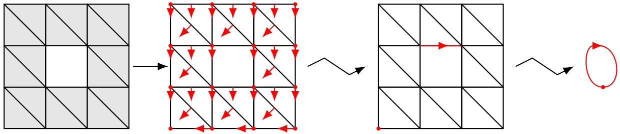
Intuitive idea

- Reduce information but keeping the homological properties
- Discrete Morse Theory
  - Vector fields are a tool to cancel "useless" information



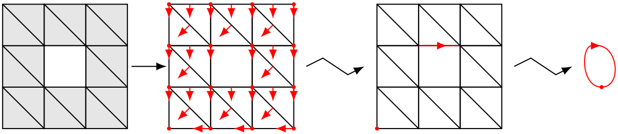
Intuitive idea

- Reduce information but keeping the homological properties
- Discrete Morse Theory
  - Vector fields are a tool to cancel "useless" information



Intuitive idea

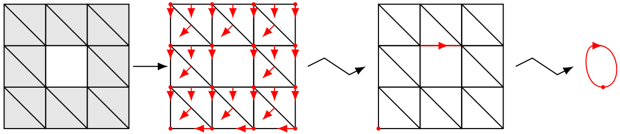
- Reduce information but keeping the homological properties
- Discrete Morse Theory
  - Vector fields are a tool to cancel “useless” information



- Given a chain complex C_* and a *dvf*, V over C_*
  - $C_* \Rightarrow C_*^c$
  - generators of C_*^c are critical cells of C_*

Intuitive idea

- Reduce information but keeping the homological properties
- Discrete Morse Theory
  - Vector fields are a tool to cancel “useless” information



- Given a chain complex C_* and a *dvf*, V over C_*
  - $C_* \Rightarrow C_*^c$
  - generators of C_*^c are critical cells of C_*

$$
\begin{array}{c}
0 \leftarrow \mathbb{Z}^{16} \xleftarrow{d_1} \mathbb{Z}^{32} \xleftarrow{d_2} \mathbb{Z}^{16} \leftarrow 0 \\
\downarrow \\
0 \leftarrow \mathbb{Z} \xleftarrow{\widehat{d}_1} \mathbb{Z} \xleftarrow{\widehat{d}_2} 0 \leftarrow 0
\end{array}
$$

Discrete Morse Theory

Definition

Let $C_* = (C_p, d_p)_{p \in \mathbb{Z}}$ be a free chain complex with distinguished $\mathbb{Z}-$basis $\beta_p \subset C_p$. A $(p-1)$-cell σ is a *face* of a p-cell τ if the coefficient of σ in $d\tau$ is non-null. It is a *regular face* if this coefficient is $+1$ or -1

Definition

A *discrete vector field* on C_* is a collection of pairs $V = \{(\sigma_i, \tau_i)\}_{i \in \beta}$ satisfying the conditions:

1. Every σ_i is some element of β_p, in which case the other corresponding component $\tau_i \in \beta_{p+1}$. The degree p depends on i and in general is not constant
2. Every component σ_i is a *regular face* of the corresponding component τ_i
3. A generator of C_* appears at most one time in V

Discrete Morse Theory

Definition

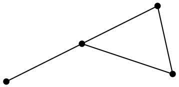
A *V-path of degree p* is a sequence $\pi = ((\sigma_{i_k}, \tau_{i_k}))_{0 \leq k < m}$ satisfying:

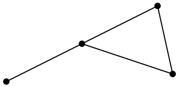
1. Every pair $((\sigma_{i_k}, \tau_{i_k}))$ is a component of V and the cell τ_{i_k} is a p-cell
2. For every $0 < k < m$, the component σ_{i_k} is a face of $\tau_{i_{k-1}}$, non necessarily regular, but different from $\sigma_{i_{k-1}}$

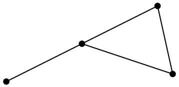
Definition

A discrete vector field V is *admissible* if for every $p \in \mathbb{Z}$, a function $\lambda_p : \beta_p \rightarrow \mathbb{Z}$ is provided satisfying the property: every V-path starting from $\sigma \in \beta_p$ has a length bounded by $\lambda_p(\sigma)$

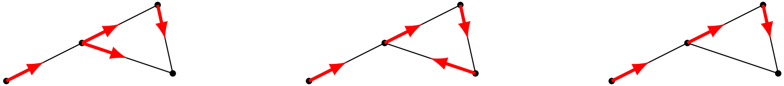
Example: an admissible discrete vector field



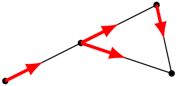




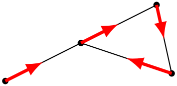
Example: an admissible discrete vector field

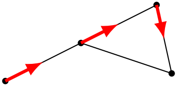


Example: an admissible discrete vector field

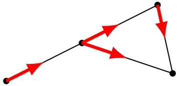


Dvf x

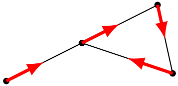




Example: an admissible discrete vector field

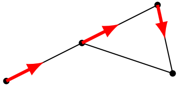


Dvf x

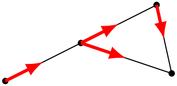


Dvf ✓

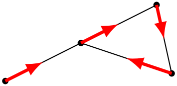
Admissible x



Example: an admissible discrete vector field

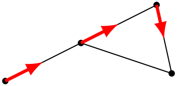


Dvf x



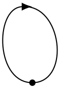
Dvf ✓

Admissible x



Dvf ✓

Admissible ✓



Discrete Morse Theory

Definition

A cell χ which does not appear in a discrete vector field $V = \{(\sigma_i, \tau_i)\}_{i \in \beta}$ is called a *critical cell*

Vector-Field Reduction Theorem

Let $C_* = (C_p, d_p\beta_p)_p$ be a free chain complex and $V = \{(\sigma_i, \beta_i)\}_{i \in \beta}$ be an admissible discrete vector field on C_*. Then the vector field V defines a canonical reduction $\rho = (f, g, h) : (C_p, d_p) \Longrightarrow (C_p^c, d'_p)$ where $C_p^c = \mathbb{Z}\left[\beta_p^c\right]$ is the free $\mathbb{Z}-$module generated by the critical p-cells

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

Table of Contents

1 Motivation

2 Mathematical preliminaries

3 Admissible discrete vector fields

4 From computation to verification through deduction

5 Application

6 Conclusions and further work

Methodology

- *Haskell* as programming language
- *QuickCheck* to test the programs
- *Coq/SSReflect* to verify the correctness of the programs

Haskell

Algorithm (*gen_adm_dvf*)

Input: A matrix M
Output: An admissible discrete vector field for M

Algorithm (*reduced_cc*)

Input: A chain complex C_*
Output: A reduced chain complex $\hat{C}_*$

```
gen_adm_dvf [[1,0,1,1],[0,0,1,0],[1,1,0,1]]
[(0,0),(1,2),(2,1)]
```

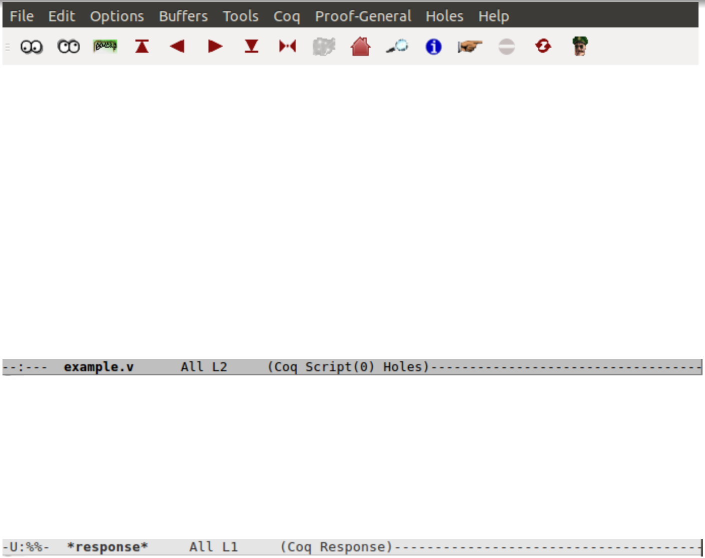
QuickCheck

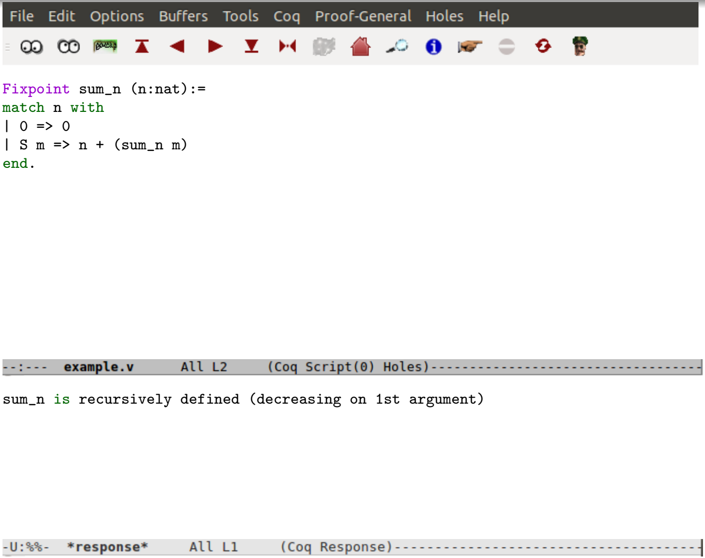
- A specification of the properties which our program must verify
- Testing them
  - Towards verification
  - Detect bugs

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

Coq/SSReflect

- Coq
  - Theorem Prover tool
  - Higher-order logic
- SSReflect
  - Extension of Coq
  - Introduce new tactics and libraries
  - Used to formalize the Four Colour Theorem


File Edit Options Buffers Tools Coq Proof-General Holes Help
--:--- example.v All L2 (Coq Script(0) Holes)
-U:%%- *response* All L1 (Coq Response)


File Edit Options Buffers Tools Coq Proof-General Holes Help
Fixpoint sum_n (n:nat):=
match n with
| 0 => 0
| S m => n + (sum_n m)
end.
--:--- example.v All L2 (Coq Script(0) Holes)
sum_n is recursively defined (decreasing on 1st argument)
-U:%%- *response* All L1 (Coq Response)

File Edit Options Buffers Tools Coq Proof-General Holes Help

```
Fixpoint sum_n (n:nat):=
match n with
| 0 => 0
| S m => n + (sum_n m)
end.

Lemma sum_n_p (n:nat): 2 * (sum_n n) = n*(n+1).
Proof.
```

```
--:---  example.v      All L2     (Coq Script(0) Holes)----------------------------------
```

```
1 subgoal (ID 6)

  n : nat
  ============================
   2 * sum_n n = n * (n + 1)
```

```
-U:%%-  *response*     All L1     (Coq Response)---------------------------------------
```

File Edit Options Buffers Tools Coq Proof-General Holes Help

```
Fixpoint sum_n (n:nat):=
match n with
| 0 => 0
| S m => n + (sum_n m)
end.

Lemma sum_n_p (n:nat): 2 * (sum_n n) = n*(n+1).
Proof.
elim: n.
```

```
--:---  example.v      All L2     (Coq Script(0) Holes)----------------------------------
```

```
2 subgoals, subgoal 1 (ID 14)

  ============================
   2 * sum_n 0 = 0 * (0 + 1)

subgoal 2 (ID 15) is:
 forall n : nat,
 2 * sum_n n = n * (n + 1) -> 2 * sum_n n.+1 = n.+1 * (n.+1 + 1)
-U:%%-  *response*     All L1     (Coq Response)-------------------------------------
```

File Edit Options Buffers Tools Coq Proof-General Holes Help

```
Fixpoint sum_n (n:nat):=
match n with
| 0 => 0
| S m => n + (sum_n m)
end.

Lemma sum_n_p (n:nat): 2 * (sum_n n) = n*(n+1).
Proof.
elim: n.
  rewrite /sum_n //=.
```

```
--:---  example.v       All L2     (Coq Script(0) Holes)----------------------------------
```

```
1 subgoal (ID 15)

  ============================
   forall n : nat,
   2 * sum_n n = n * (n + 1) -> 2 * sum_n n.+1 = n.+1 * (n.+1 + 1)
```

```
-U:%%-  *response*      All L1     (Coq Response)------------------------------------
```

File Edit Options Buffers Tools Coq Proof-General Holes Help

```
Fixpoint sum_n (n:nat):=
match n with
| 0 => 0
| S m => n + (sum_n m)
end.

Lemma sum_n_p (n:nat): 2 * (sum_n n) = n*(n+1).
Proof.
elim: n.
  rewrite /sum_n //=.
move=> n H.
rewrite /sum_n -/sum_n.
```

```
--:---  example.v      All L2      (Coq Script(0) Holes)----------------------------------
```

```
1 subgoal (ID 21)

  n : nat
  H : 2 * sum_n n = n * (n + 1)
  ============================
   2 * (n.+1 + sum_n n) = n.+1 * (n.+1 + 1)
```

```
-U:%%-  *response*     All L1      (Coq Response)--------------------------------------
```

File Edit Options Buffers Tools Coq Proof-General Holes Help

```
Fixpoint sum_n (n:nat):=
match n with
| 0 => 0
| S m => n + (sum_n m)
end.

Lemma sum_n_p (n:nat): 2 * (sum_n n) = n*(n+1).
Proof.
elim: n.
  rewrite /sum_n //=.
move=> n H.
rewrite /sum_n -/sum_n.
rewrite muln_addr.
```

```
--:---  example.v      All L2     (Coq Script(0) Holes)----------------------------------
```

```
1 subgoal (ID 21)

  n : nat
  H : 2 * sum_n n = n * (n + 1)
  ============================
   2 * n.+1 + 2 * sum_n n = n.+1 * (n.+1 + 1)
```

```
-U:%%-  *response*     All L1     (Coq Response)--------------------------------------
```

File Edit Options Buffers Tools Coq Proof-General Holes Help

```
Fixpoint sum_n (n:nat):=
match n with
| 0 => 0
| S m => n + (sum_n m)
end.

Lemma sum_n_p (n:nat): 2 * (sum_n n) = n*(n+1).
Proof.
elim: n.
  rewrite /sum_n //=.
move=> n H.
rewrite /sum_n -/sum_n.
rewrite muln_addr.
rewrite H; ring.
```

--:--- **example.v** All L2 (Coq Script(0) Holes)----------------------------------

```
No more subgoals.
```

-U:%%- ***response*** All L1 (Coq Response)------------------------------------

File Edit Options Buffers Tools Coq Proof-General Holes Help

```
Fixpoint sum_n (n:nat):=
match n with
| 0 => 0
| S m => n + (sum_n m)
end.

Lemma sum_n_p (n:nat): 2 * (sum_n n) = n*(n+1).
Proof.
elim: n.
  rewrite /sum_n //=.
move=> n H.
rewrite /sum_n -/sum_n.
rewrite muln_addr.
rewrite H; ring.
Qed.
```

```
--:---  example.v      All L2     (Coq Script(0) Holes)-----------------------------------
```

```
sum_n_p is defined
```

```
-U:%%-  *response*     All L1     (Coq Response)-------------------------------------------
```

SSReflect

Lema SSReflect:

Lemma *gen_adm_dvf_is_admissible* (M:seq(seq Z2_ring)) :
admissible (gen_adm_dvf M).

Lema SSReflect:

Lemma *is_reduction* (C:chaincomplex) : reduction C (reduced_cc C).

Lema SSReflect:

Lemma *reduction_preserves_betti* (C D : chaincomplex) (rho : reduction C D):
Betti C = Betti D.

Table of Contents

1 Motivation

2 Mathematical preliminaries

3 Admissible discrete vector fields

4 From computation to verification through deduction

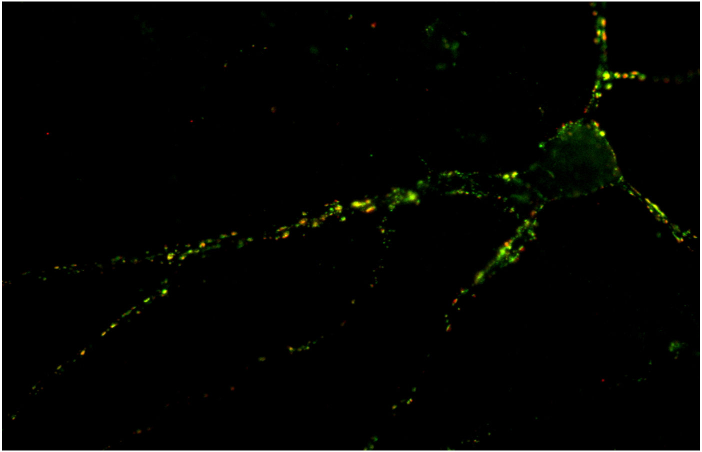
5 Application

6 Conclusions and further work

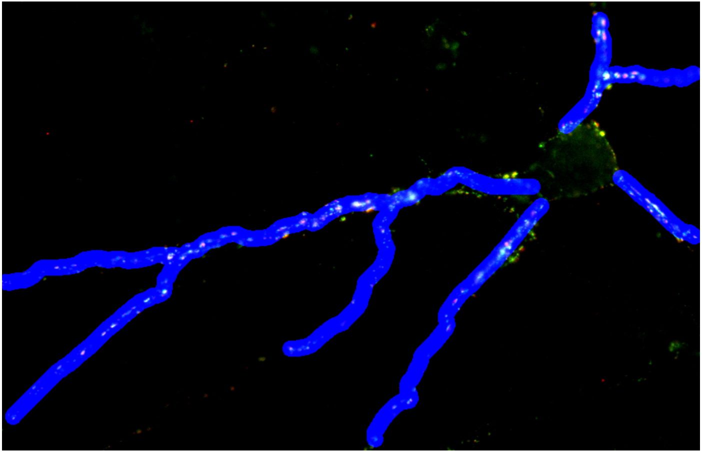
Counting Synapses

- *Synapses* are the points of connection between neurons
- *Relevance*: Computational capabilities of the brain
- Procedures to modify the synaptic density may be an important asset in the treatment of neurological diseases
- An automated and reliable method is necessary

Counting Synapses



Counting Synapses



Counting Synapses

Table of Contents

1. Motivation
2. Mathematical preliminaries
3. Admissible discrete vector fields
4. From computation to verification through deduction
5. Application
6. Conclusions and further work

Conclusions

- Towards an Algebraic Topology Formal library

Conclusions

- Towards an Algebraic Topology Formal library
- Methodology to study Biomedical images using Homological tools

Conclusions

- Towards an Algebraic Topology Formal library
- Methodology to study Biomedical images using Homological tools
- Discrete Vector Fields to deal with big images

Conclusions

- Towards an Algebraic Topology Formal library
- Methodology to study Biomedical images using Homological tools
- Discrete Vector Fields to deal with big images
- Certified computation of Homology from digital images

Conclusions

- Towards an Algebraic Topology Formal library
- Methodology to study Biomedical images using Homological tools
- Discrete Vector Fields to deal with big images
- Certified computation of Homology from digital images
- Application to a biomedical problem: counting synapses

Further work

- Verification
  - Another reductions: collapses
  - Cubical complex
  - Homology over $\mathbb{Z}$
- Apply homological tools in other Biomedical imaging contexts
  - tools:
    - Persistent Homology
    - Homology in higher dimensions
  - context:
    - Count and classify spines
    - Detect neurological structure

Symbolic Manipulation and Biomedical Images*

Jónathan Heras, Gadea Mata, María Poza, and Julio Rubio

Department of Mathematics and Computer Science
University of La Rioja
Spain

June 13, 2012

* Partially supported by Ministerio de Educación y Ciencia, project MTM2009-13842-C02-01, and by European Commission FP7, STREP project ForMath, n. 243847