Effective Homology of the Pushout of Simplicial Sets

Jónathan Heras

Departamento de Matemáticas y Computación
Universidad de La Rioja
Spain

XII Encuentro de Álgebra Computacional y Aplicaciones
EACA 2010, Santiago de Compostela
Motivation

- Kenzo:
  Symbolic Computation system devoted to Algebraic Topology
  Compute homology groups of spaces of not finite type: loop spaces, classifying spaces, . . .

General goal

- Increase the functionality of Kenzo

Pushout:

Usual construction in Topology

Particular cases: wedges, joins, . . .

Concrete goal

- New Kenzo module for constructing the Pushout of simplicial sets
Motivation

Kenzo:
- Symbolic Computation system devoted to Algebraic Topology

General goal
- Increase the functionality of Kenzo

Concrete goal
- New Kenzo module for constructing the Pushout of simplicial sets
Motivation

- Kenzo:
  - Symbolic Computation system devoted to Algebraic Topology
  - Compute homology groups of spaces of not finite type: loop spaces, classifying spaces, . . .
Motivation

Kenzo:
- Symbolic Computation system devoted to Algebraic Topology
- Compute homology groups of spaces of not finite type: loop spaces, classifying spaces, ...

General goal
Increase the functionality of Kenzo
Motivation

- Kenzo:
  - Symbolic Computation system devoted to Algebraic Topology
  - Compute homology groups of spaces of not finite type: loop spaces, classifying spaces,…

General goal

Increase the functionality of Kenzo

- Pushout:
Motivation

- **Kenzo:**
  - Symbolic Computation system devoted to Algebraic Topology
  - Compute homology groups of spaces of not finite type: loop spaces, classifying spaces, . . .

**General goal**

Increase the functionality of Kenzo

- **Pushout:**
  - Usual construction in Topology
Motivation

- **Kenzo:**
  - Symbolic Computation system devoted to Algebraic Topology
  - Compute homology groups of spaces of not finite type: loop spaces, classifying spaces, . . .

General goal

**Increase the functionality of Kenzo**

- **Pushout:**
  - Usual construction in Topology
  - Particular cases: wedges, joins, . . .
Motivation

Kenzo:
- Symbolic Computation system devoted to Algebraic Topology
- Compute homology groups of spaces of not finite type: loop spaces, classifying spaces, ...

General goal
Increase the functionality of Kenzo

Pushout:
- Usual construction in Topology
- Particular cases: wedges, joins, ...

Concrete goal
New Kenzo module for constructing the Pushout of simplicial sets
Table of Contents

1 Previous concepts
   - Effective Homology
   - Pushout

2 Effective Homology of the Pushout

3 Examples

4 Conclusions and Further Work
Table of Contents

1. Previous concepts
   - Effective Homology
   - Pushout

2. Effective Homology of the Pushout

3. Examples

4. Conclusions and Further Work
Effective Homology

- Finite nature objects:
Effective Homology

- Finite nature objects:
  - Adjacency matrix is an integer matrix
Effective Homology

- Finite nature objects:
  - Adjacency matrix is an integer matrix
  - Homology groups: Smith Normal Form
Effective Homology

- Finite nature objects:
  - Adjacency matrix is an integer matrix
  - Homology groups: Smith Normal Form
- Non finite nature objects:
Effective Homology

- Finite nature objects:
  - Adjacency matrix is an integer matrix
  - Homology groups: Smith Normal Form
- Non finite nature objects:
  - Traditional methods can not be applied
Effective Homology

- Finite nature objects:
  - Adjacency matrix is an integer matrix
  - Homology groups: Smith Normal Form

- Non finite nature objects:
  - Traditional methods can not be applied
  - Effective Homology:
Effective Homology

- Finite nature objects:
  - Adjacency matrix is an integer matrix
  - Homology groups: Smith Normal Form

- Non finite nature objects:
  - Traditional methods can not be applied
  - Effective Homology:
    - Sergeraert’s ideas
Effective Homology

- Finite nature objects:
  - Adjacency matrix is an integer matrix
  - Homology groups: Smith Normal Form
- Non finite nature objects:
  - Traditional methods can not be applied
  - Effective Homology:
    - Sergeraert’s ideas
    - Provides real algorithms to compute homology groups
Effective Homology

Finite nature objects:
- Adjacency matrix is an integer matrix
- Homology groups: Smith Normal Form

Non finite nature objects:
- Traditional methods can not be applied
- Effective Homology:
  - Sergeraert’s ideas
  - Provides real algorithms to compute homology groups
  - Implemented in the Kenzo system using Higher-order functional programming
Effective Homology

- **Finite nature objects:**
  - Adjacency matrix is an integer matrix
  - Homology groups: Smith Normal Form

- **Non finite nature objects:**
  - Traditional methods can not be applied
  - Effective Homology:
    - Sergeraert’s ideas
    - Provides real algorithms to compute homology groups
    - Implemented in the Kenzo system using Higher-order functional programming
Effective Homology

- Finite nature objects: Effective Objects
  - Adjacency matrix is an integer matrix
  - Homology groups: Smith Normal Form
- Non finite nature objects: Locally Effective Objects
  - Traditional methods can not be applied
  - Effective Homology:
    - Sergeraert’s ideas
    - Provides real algorithms to compute homology groups
    - Implemented in the Kenzo system using Higher-order functional programming
Intuitive idea

- Effective Objects:
  - Explicit list of generators
  - Local properties: determine if an element belongs to a set
  - Global properties: determine if a set is empty
  - Example: set definition by extension

- Locally Effective Objects:
  - Non available explicit list of generators
  - Infinite numbers of generators
  - Characteristic function
  - Local information is available
  - Example: set definition by intension
Intuitive idea

Effective Objects:
- Explicit list of generators
Intuitive idea

- Effective Objects:
  - Explicit list of generators
  - Local properties: determine if an elements belongs to a set
Intuitive idea

- **Effective Objects:**
  - Explicit list of generators
  - Local properties: determine if an elements belongs to a set
  - Global properties: determine if a set is empty
Intuitive idea

Effective Objects:

- Explicit list of generators
- Local properties: determine if an elements belongs to a set
- Global properties: determine if a set is empty
- Example: set definition by extension
Intuitive idea

- **Effective Objects:**
  - Explicit list of generators
  - Local properties: determine if an element belongs to a set
  - Global properties: determine if a set is empty
  - Example: set definition by extension

- **Locally Effective Objects:**
Intuitive idea

- **Effective Objects:**
  - Explicit list of generators
  - Local properties: determine if an element belongs to a set
  - Global properties: determine if a set is empty
  - Example: set definition by extension

- **Locally Effective Objects:**
  - Non available explicit list of generators
Intuitive idea

- **Effective Objects:**
  - Explicit list of generators
  - Local properties: determine if an element belongs to a set
  - Global properties: determine if a set is empty
  - Example: set definition by extension

- **Locally Effective Objects:**
  - Non available explicit list of generators
    - Infinite numbers of generators
    - Characteristic function
Intuitive idea

**Effective Objects:**
- Explicit list of generators
- Local properties: determine if an elements belongs to a set
- Global properties: determine if a set is empty
- Example: set definition by extension

**Locally Effective Objects:**
- Non available explicit list of generators
  - Infinite numbers of generators
  - Characteristic function
- Local information is available
Intuitive idea

- **Effective Objects:**
  - Explicit list of generators
  - Local properties: determine if an element belongs to a set
  - Global properties: determine if a set is empty
  - Example: set definition by extension

- **Locally Effective Objects:**
  - Non available explicit list of generators
    - Infinite numbers of generators
    - Characteristic function
  - Local information is available
  - Example: set definition by intension
Effective vs Locally Effective Chain Complexes

**Definition**

An effective chain complex is a free chain complex of $\mathbb{Z}$-modules, $C_* = (C_n, d_n)_{n \in \mathbb{N}}$, where each group $C_n$ is finitely generated and

- an algorithm returns a $\mathbb{Z}$-base in each grade $n$
- an algorithm provides the differentials $d_n$

**Definition**

A locally effective chain complex is a free chain complex of $\mathbb{Z}$-modules, $C_* = (C_n, d_n)_{n \in \mathbb{N}}$, where each group $C_n$ is formed by an infinite number of generators.

J. Heras

Effective Homology of the Pushout of Simplicial Sets
Effective vs Locally Effective Chain Complexes

**Definition**

An effective chain complex is a free chain complex of $\mathbb{Z}$-modules, $C_* = (C_n, d_n)_{n \in \mathbb{N}}$, where each group $C_n$ is finitely generated and

- an algorithm returns a $\mathbb{Z}$-base in each grade $n$
- an algorithm provides the differentials $d_n$

- differentials $d_n : C_n \rightarrow C_{n-1}$ can be expressed as integer matrices
- possible to compute $\text{Ker } d_n$ y $\text{Im } d_{n+1}$

**Definition**

A locally effective chain complex is a free chain complex of $\mathbb{Z}$-modules, $C_* = (C_n, d_n)_{n \in \mathbb{N}}$, where each group $C_n$ is formed by an infinite number of generators

- impossible to compute $\text{Ker } d_n$ y $\text{Im } d_{n+1}$
Effective vs Locally Effective Chain Complexes

**Definition**

An effective chain complex is a free chain complex of \(\mathbb{Z}\)-modules, \(C_\ast = (C_n, d_n)_{n \in \mathbb{N}}\), where each group \(C_n\) is finitely generated and

- an algorithm returns a \(\mathbb{Z}\)-base in each grade \(n\)
- an algorithm provides the differentials \(d_n\)

- differentials \(d_n : C_n \to C_{n-1}\) can be expressed as integer matrices
- possible to compute \(\text{Ker} \ d_n\) y \(\text{Im} \ d_{n+1}\)
- possible to compute the homology groups

**Definition**

A locally effective chain complex is a free chain complex of \(\mathbb{Z}\)-modules, \(C_\ast = (C_n, d_n)_{n \in \mathbb{N}}\), where each group \(C_n\) is formed by an infinite number of generators

- impossible to compute \(\text{Ker} \ d_n\) y \(\text{Im} \ d_{n+1}\)
- possible to perform local computations, differential of a generator
Effective Homology

Definition

A reduction \( \rho \) between two chain complexes \( C_* \) and \( D_* \) (denoted by \( \rho : C_* \Rightarrow D_* \)) is a triple \( \rho = (f, g, h) \) satisfying the following relations:

1) \( fg = \text{Id}_{D_*} \);
2) \( d_C h + hd_C = \text{Id}_{C_*} - gf \);
3) \( fh = 0 \); \( hg = 0 \); \( hh = 0 \).
Effective Homology

**Definition**

A reduction $\rho$ between two chain complexes $C_*$ and $D_*$ (denoted by $\rho : C_* \Rightarrow D_*$) is a triple $\rho = (f, g, h)$ satisfying the following relations:

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

**Theorem**

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

J. Heras
Effective Homology

**Definition**

A *strong chain equivalence* \( \varepsilon \) between \( C_* \) and \( D_* \), \( \varepsilon : C_* \rightleftharpoons D_* \), is a triple \( \varepsilon = (B_*, \rho, \rho') \) where \( B_* \) is a chain complex, \( \rho : B_* \to C_* \) and \( \rho' : B_* \to D_* \).
Effective Homology

**Definition**

A (strong chain) equivalence \( \varepsilon \) between \( C_\ast \) and \( D_\ast \), \( \varepsilon : C_\ast \leftrightarrow D_\ast \), is a triple \( \varepsilon = (B_\ast, \rho, \rho') \) where \( B_\ast \) is a chain complex, \( \rho : B_\ast \Rightarrow C_\ast \) and \( \rho' : B_\ast \Rightarrow D_\ast \).

**Definition**

An object with effective homology is a quadruple \((X, C_\ast(X), HC_\ast, \varepsilon)\) where:

- \( X \) is a locally effective object
- \( C_\ast(X) \) is a (locally effective) chain complex canonically associated with \( X \), which allows the study of the homological nature of \( X \)
- \( HC_\ast \) is an effective chain complex
- \( \varepsilon \) is a equivalence \( \varepsilon : C_\ast(X) \leftrightarrow HC_\ast \)

Theorem

Let an object with effective homology \((X, C_\ast(X), HC_\ast, \varepsilon)\) then \( H_n(X) \sim = H_n(HC_\ast) \) for all \( n \).
Definition

A (strong chain) equivalence $\varepsilon$ between $C_*$ and $D_*$, $\varepsilon : C_* \Leftrightarrow D_*$, is a triple $\varepsilon = (B_*, \rho, \rho')$ where $B_*$ is a chain complex, $\rho : B_* \Rightarrow C_*$ and $\rho' : B_* \Rightarrow D_*$. 

\[
\begin{array}{c}
C_* \\
\leftarrow \\
\downarrow \\
B_* \\
\leftarrow \\
\downarrow \\
D_* \\
\leftarrow \\
\downarrow \\
\rightarrow \\
\end{array}
\]

Definition

An object with effective homology is a quadruple $(X, C_*(X), HC_*, \varepsilon)$ where:

- $X$ is a locally effective object
- $C_*(X)$ is a (locally effective) chain complex canonically associated with $X$, which allows the study of the homological nature of $X$
- $HC_*$ is an effective chain complex
- $\varepsilon$ is an equivalence $\varepsilon : C_*(X) \Leftrightarrow HC_*$

Theorem

Let an object with effective homology $(X, C_*(X), HC_*, \varepsilon)$ then $H_n(X) \cong H_n(HC_*)$ for all $n$. 

J. Heras
**Definition**

Let $f, g$ morphisms, the pushout of $f, g$

$$
\begin{array}{c}
X \xrightarrow{f} Y \\
\downarrow g \\
Z
\end{array}
$$
Pushout

Definition

Let \( f, g \) morphisms, the pushout of \( f, g \)

\[
\begin{array}{c}
X \xrightarrow{f} Y \\
\downarrow g \quad \downarrow f' \\
Z \xrightarrow{g'} P_{(f,g)}
\end{array}
\]

is an object \( P \) for which the diagram:

- commutes

J. Heras

Effective Homology of the Pushout of Simplicial Sets
Pushout

**Definition**

Let $f, g$ morphisms, the pushout of $f, g$

![Diagram](image)

is an object $P$ for which the diagram:

- commutes
- respects the universal property
Pushout

Standard Construction

\[ P_{(f,g)} \cong (Y \amalg (X \times I) \amalg Z) / \sim \]

where:

- \( I \) is the unit interval
- for every \( x \in X \), \( \sim \):
  - \((x, 0) \sim f(x) \in Y\)
  - \((x, 1) \sim g(x) \in Z\)
## Table of Contents

1. Previous concepts
2. Effective Homology of the Pushout
3. Examples
4. Conclusions and Further Work
Construction of the Effective Homology of the Pushout: Step 1

Given \( f : X \rightarrow Y \) and \( g : X \rightarrow Z \) simplicial morphisms where \( X, Y \) and \( Z \) are simplicial sets:

\[
\begin{array}{ccc}
X & \xrightarrow{f} & Y \\
\downarrow{g} & & \downarrow{g} \\
Z & & Z
\end{array}
\]
Construction of the Effective Homology of the Pushout:

Step 1

Given \( f : X \to Y \) and \( g : X \to Z \) simplicial morphisms where \( X, Y \) and \( Z \) are simplicial sets:

\[
\begin{array}{ccc}
X & \xrightarrow{f} & Y \\
\downarrow g & & \downarrow f' \\
Z & \xrightarrow{g'} & P(f, g)
\end{array}
\]
Construction of the Effective Homology of the Pushout: Step 1

Given \( f : X \rightarrow Y \) and \( g : X \rightarrow Z \) simplicial morphisms where \( X, Y \) and \( Z \) are simplicial sets:

\[
\begin{array}{ccc}
X & \xrightarrow{f} & Y \\
\downarrow{} & & \downarrow{} \\
Z & \xrightarrow{g'} & P(f,g)
\end{array}
\]

**Theorem** (Algorithm: Standard Construction, Implementation: J. Heras)

**Input:** Two simplicial morphisms \( f : X \rightarrow Y \) and \( g : X \rightarrow Z \) where \( X, Y \) and \( Z \) are simplicial sets.

**Output:** The pushout \( P(f,g) \).

J. Heras

Effective Homology of the Pushout of Simplicial Sets
Construction of the Effective Homology of the Pushout: Step 2

Given \( f : X \rightarrow Y \) and \( g : X \rightarrow Z \) simplicial morphisms where \( X, Y \) and \( Z \) are simplicial sets with effective homology:

\[
\begin{array}{ccc}
(X, C_*(X), HX_*, \varepsilon_X) & \xrightarrow{f} & (Y, C_*(Y), HY_*, \varepsilon_Y) \\
g & & g \\
(Z, C_*(Z), HZ_*, \varepsilon_Z) & \xrightarrow{g} & (P(f,g), C_*(P(f,g)), -, -)
\end{array}
\]
Construction of the Effective Homology of the Pushout:

Step 2

Given \( f : X \rightarrow Y \) and \( g : X \rightarrow Z \) simplicial morphisms where \( X, Y \) and \( Z \) are simplicial sets with effective homology:

\[
\begin{align*}
(X, C_\ast(X), HX_\ast, \varepsilon_X) & \xrightarrow{f} (Y, C_\ast(Y), HY_\ast, \varepsilon_Y) \\
\downarrow g & \downarrow g \\
(Z, C_\ast(Z), HZ_\ast, \varepsilon_Z) & \xrightarrow{g \circ f} (P_{f,g}, C_\ast(P_{f,g}), HP_\ast, \varepsilon_P)
\end{align*}
\]
Construction of the Effective Homology of the Pushout: Step 2

Given \( f : X \rightarrow Y \) and \( g : X \rightarrow Z \) simplicial morphisms where \( X, Y \) and \( Z \) are simplicial sets with effective homology:

\[
(X, C_\ast(X), HX_\ast, \varepsilon_X) \xrightarrow{f} (Y, C_\ast(Y), HY_\ast, \varepsilon_Y) \downarrow g \downarrow
(Z, C_\ast(Z), HZ_\ast, \varepsilon_Z) \xrightarrow{g} (P(f,g), C_\ast(P(f,g)), HP_\ast, \varepsilon_P)
\]

**Theorem (Algorithm: F. Sergeraert, Implementation: J. Heras)**

**Input:** two simplicial morphisms \( f : X \rightarrow Y \) and \( g : X \rightarrow Z \) where \( X, Y \) and \( Z \) are simplicial sets with effective homology.

**Output:** the effective homology version of \( P(f,g) \), that is, an equivalence \( C_\ast(P(f,g)) \Leftrightarrow HP_\ast \), where \( HP_\ast \) is an effective chain complex.
Effective Homology of the Pushout

Theorem

Input:

\[ C_*(B) \text{ a chain complex;} \]

\[ (C_*(A), HA_*, \varepsilon_A); \]

\[ (C_*(C), HC_*, \varepsilon_C); \]

\[ 0 \leftarrow 0 \quad C_*(A)_* \xleftarrow{\sigma} C_*(B) \xrightarrow{\rho} C_*(C) \leftarrow 0 \]

Output: \( (C_*(B), HB_*, \varepsilon_B) \)

\[ 0 \leftarrow M \leftarrow CP(f,g) \leftarrow CY \oplus CZ \leftarrow 0 \]

where \( M = X \times I \setminus ((X \times \{0\}) \cup (X \times \{1\})) \)
Effective Homology of the Pushout

\[ 0 \leftrightarrow M \leftrightarrow CP(f,g) \leftrightarrow CY \oplus CZ \leftrightarrow 0 \]

where \( M = X \times I \setminus ((X \times \{0\}) \cup (X \times \{1\})) \)

**Theorem**

*Input*: two simplicial sets \( X \) and \( Y \) with effective homology

*Output*: an equivalence \( C_*(X \oplus Y) \leftrightarrow DD_* \Rightarrow HD_* \), where \( HD_* \) is effective.
Effective Homology of the Pushout

\[
0 \quad \quad M \quad \quad \quad CP(f,g) \quad \quad CY \oplus CZ \quad \quad 0
\]

where

\[M = X \times I \setminus ((X \times \{0\}) \cup (X \times \{1\}))\]

Consider the short exact sequence:

\[
0 \quad \quad M \quad \quad \quad C(X \times I) \quad \quad C(X \times \{0\}) \oplus C(X \times \{1\}) \quad \quad 0
\]
Effective Homology of the Pushout of Simplicial Sets

Theorem

Input:

\[ C_*(A) \text{ a chain complex;} \]
\[ (C_*(B), HB_*, \varepsilon_B); \]
\[ (C_*(C), HC_*, \varepsilon_C); \]

\[ 0 \leftarrow C_*(A) \xleftarrow{\sigma} C_*(B) \xrightarrow{\rho} C_*(C) \leftarrow 0 \]

Output: \((C_*(A), HA_*, \varepsilon_A)\)

where
\[ M = X \times I \setminus ((X \times \{0\}) \cup (X \times \{1\})) \]
Effective Homology of the Pushout of Simplicial Sets

\[ 0 \leftarrow M \leftarrow C(X \times I) \leftarrow C(X \times \{0\}) \oplus C(X \times \{1\}) \leftarrow 0 \]

where \( M = X \times I \setminus ((X \times \{0\}) \cup (X \times \{1\})) \)

**Theorem (Eilenberg-Zilber Theorem)**

Input: two simplicial sets \( X \) and \( Y \) with effective homology
Output: an equivalence \( C_\ast(X \times Y) \leftarrow DC_\ast \Rightarrow C_\ast(X) \otimes C_\ast(Y) \), where \( C_\ast(X) \otimes C_\ast(Y) \) are effective.
Effective Homology of the Pushout

\[
\begin{align*}
0 & \leftarrow M \xrightarrow{\sim} C(X \times I) \xrightarrow{\sim} C(X \times \{0\}) \oplus C(X \times \{1\}) \xrightarrow{\sim} 0 \\
\text{where } M & = X \times I \setminus ((X \times \{0\}) \cup (X \times \{1\}))
\end{align*}
\]

**Theorem**

*Input:* two simplicial sets $X$ and $Y$ with effective homology  
*Output:* an equivalence $C_*(X \oplus Y) \Leftrightarrow DD_* \Rightarrow HD_*$, where $HD_*$ is effective.
Effective Homology of the Pushout of Simplicial Sets

$$\begin{align*}
0 & \hookrightarrow M \hookrightarrow C(X \times I) \hookrightarrow C(X \times \{0\}) \oplus C(X \times \{1\}) \hookrightarrow 0 \\
\text{where } M &= X \times I \setminus ((X \times \{0\}) \cup (X \times \{1\}))
\end{align*}$$

**Theorem**

**Input:**

- $C_*(A)$ a chain complex;
- $(C_*(B), HB_*, \varepsilon_B)$;
- $(C_*(C), HC_*, \varepsilon_C)$;

$$
0 \hookrightarrow \begin{array}{c}
C_*(A) \\
\sigma \\
j
\end{array} \hookrightarrow 
\begin{array}{c}
C_*(B) \\
\rho \\
i
\end{array} 
\hookrightarrow 
\begin{array}{c}
C_*(C) \\
\end{array} \hookrightarrow 0
$$

**Output:** $(C_*(A), HA_*, \varepsilon_A)$
Effective Homology of the Pushout

\[ 0 \leftarrow M \leftarrow \text{CP}(f,g) \leftarrow CY \oplus CZ \leftarrow 0 \]

where \( M = X \times I \setminus ((X \times \{0\}) \cup (X \times \{1\})) \)

**Theorem**

**Input:**

- \( C_*(B) \) a chain complex;
- \((C_*(A), HA_*, \varepsilon_A)\);
- \((C_*(C), HC_*, \varepsilon_C)\);

\[ 0 \leftarrow 0 \leftarrow C_*(A) \leftarrow \text{CP}(f,g) \leftarrow C_*(B) \leftarrow C_*(C) \leftarrow 0 \]

**Output:** \((C_*(B), HB_*, \varepsilon_B)\)
# Table of Contents

1. Previous concepts
2. Effective Homology of the Pushout
3. Examples
4. Conclusions and Further Work
Examples

- New module allows the computation of homology groups of spaces
Examples

- New module allows the computation of homology groups of spaces
- Demo
Table of Contents

1. Previous concepts
2. Effective Homology of the Pushout
3. Examples
4. Conclusions and Further Work
Conclusions:

- New Kenzo module (1600 lines) allows the computation of homology groups of spaces defined as the pushout of simplicial sets.
Conclusions

- Conclusions:
  - New Kenzo module (1600 lines) allows the computation of homology groups of spaces defined as the pushout of simplicial sets

- Further Work:
  - Implementation of new constructions
Effective Homology of the Pushout of Simplicial Sets

Jónathan Heras

Departamento de Matemáticas y Computación
Universidad de La Rioja
Spain

XII Encuentro de Álgebra Computacional y Aplicaciones
EACA 2010, Santiago de Compostela