# Integrating multiple sources to answer questions in Algebraic Topology

Jónathan Heras    Vico Pascual    Ana Romero    Julio Rubio

*Departamento de Matemáticas y Computación*
Universidad de La Rioja
Spain

9th International Conference on Mathematical Knowledge Management
MKM 2010, Paris

# Introduction

- Different sources of information:

# Introduction

- Different sources of information:
  - Consult papers or textbooks

# Introduction

- Different sources of information:
  - Consult papers or textbooks
  - Computations with CAS

# Introduction

- Different sources of information:
  - Consult papers or textbooks
  - Computations with CAS
  - Check results against tables

# Introduction

- Different sources of information:
    - Consult papers or textbooks
    - Computations with CAS
    - Check results against tables
    - Verify conjectures with a proof assistant tool

# Introduction

- Different sources of information:
  - Consult papers or textbooks
  - Computations with CAS
  - Check results against tables
  - Verify conjectures with a proof assistant tool
- Aim:

# Introduction

- Different sources of information:
  - Consult papers or textbooks
  - Computations with CAS
  - Check results against tables
  - Verify conjectures with a proof assistant tool
- Aim:
  - Mechanize the management of these sources

# Introduction

- Different sources of information:
  - Consult papers or textbooks
  - Computations with CAS
  - Check results against tables
  - Verify conjectures with a proof assistant tool
- Aim:
  - Mechanize the management of these sources
  - Focus to a subset of Algebraic Topology:

# Introduction

- Different sources of information:
  - Consult papers or textbooks
  - Computations with CAS
  - Check results against tables
  - Verify conjectures with a proof assistant tool
- Aim:
  - Mechanize the management of these sources
  - Focus to a subset of Algebraic Topology:
    - Two Computer Algebra Systems
    - A Theorem Prover
    - A Rule Based System

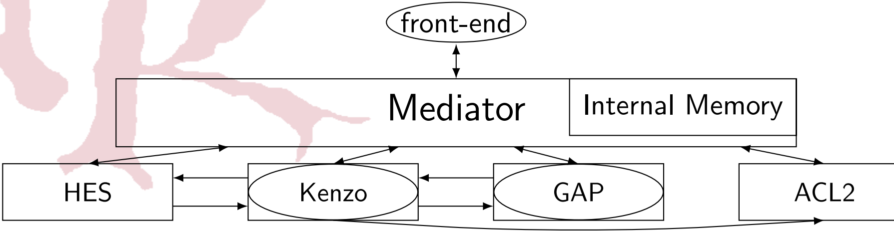
# Table of Contents


1. General View of the system
2. Interoperability among modules
3. Putting all together
4. Conclusions and Further Work

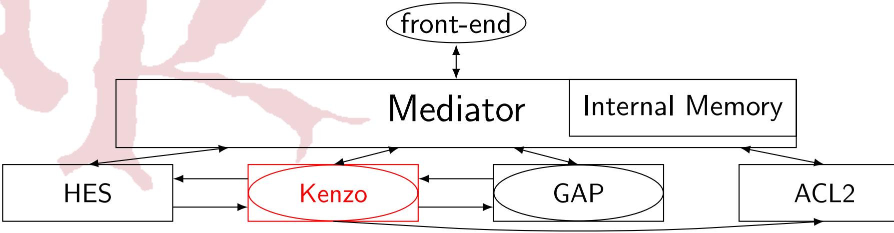
# Table of Contents


1. General View of the system
2. Interoperability among modules
3. Putting all together
4. Conclusions and Further Work

# Architecture of the system

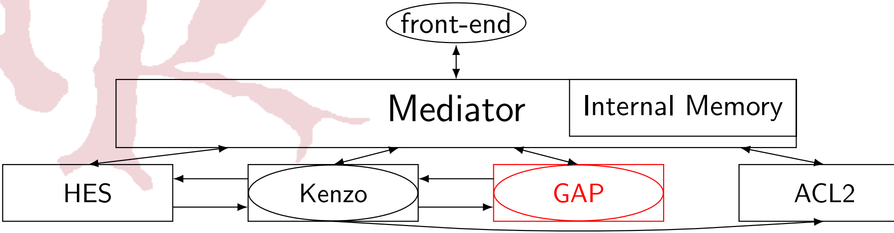


# Architecture of the system



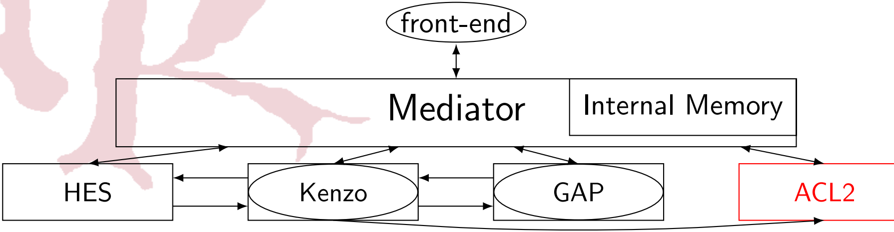

- Kenzo:
  - Symbolic Computation System devoted to Algebraic Topology
  - Homology groups unreachable by any other means
  - Core for computations related to homology groups of spaces

# Architecture of the system



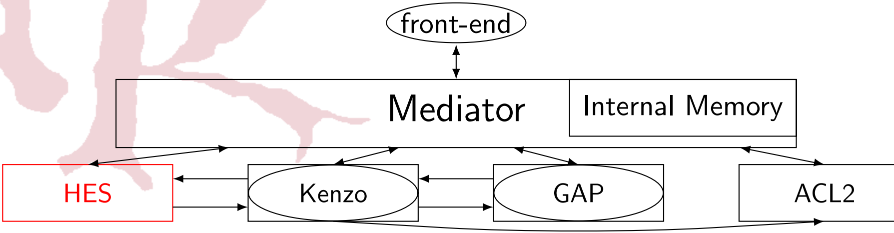

- GAP/HAP:
  - GAP: Computer Algebra System devoted to Computational Group Theory
  - HAP: Homological Algebra library for using with GAP
  - Core for computations related to group homology

# Architecture of the system



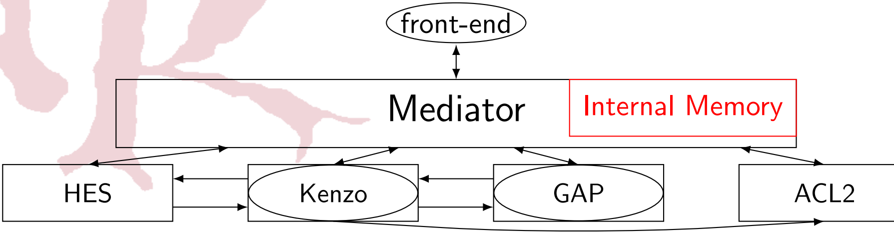

- ACL2:
  - A Computational Logic for an Applicative Common Lisp
  - Programming language, logic and theorem prover
  - Kernel for verifying the correctness of statements

# Architecture of the system




- Homotopy Expert System:
  - A new rule-based expert system
  - Computation of homotopy groups of spaces

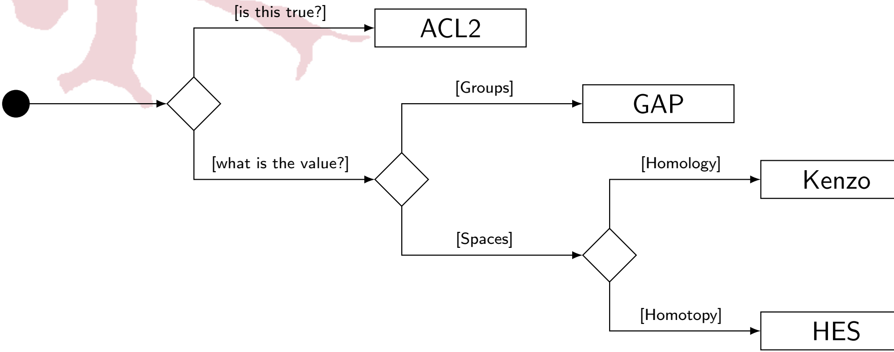
# Architecture of the system




- Internal Memory:
  - Memoization
  - Objects decorated with several annotations:
    - type of the object
    - reduction degree
    - ...

# Decision procedure

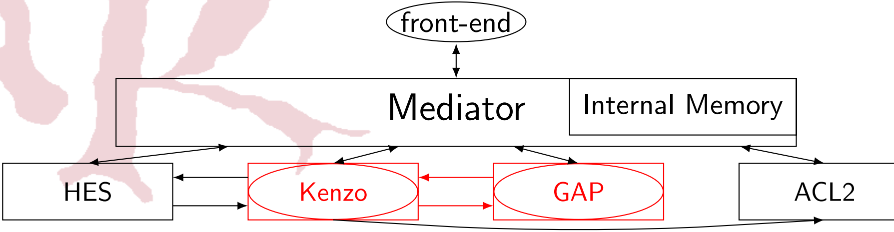
Information guides the mediator to decide which component to use.



# Table of Contents

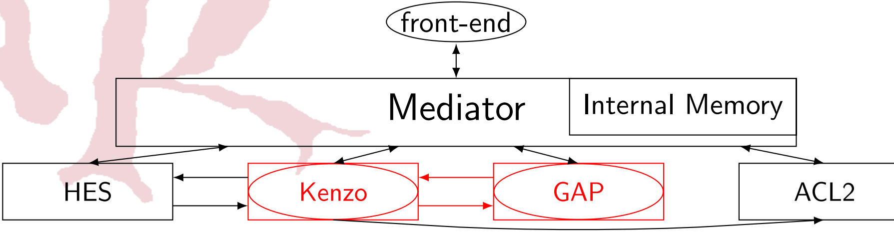

1. General View of the system
2. Interoperability among modules
   - Kenzo $\leftrightarrow$ GAP
   - Kenzo $\longrightarrow$ ACL2
   - Kenzo $\leftrightarrow$ HES
3. Putting all together
4. Conclusions and Further Work

# Kenzo $\leftrightarrow$ GAP



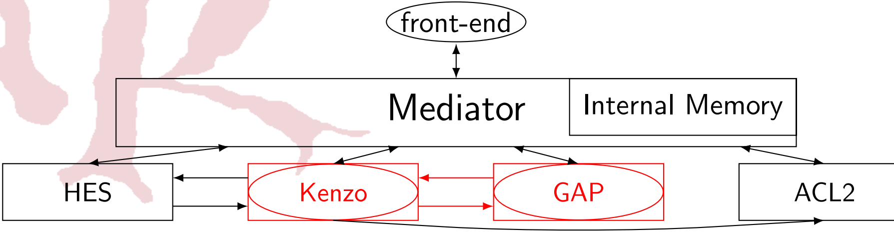

- Aim:
  - Compute homology groups of Eilenberg-MacLane spaces $K(\pi, n)$

# Kenzo ↔ GAP



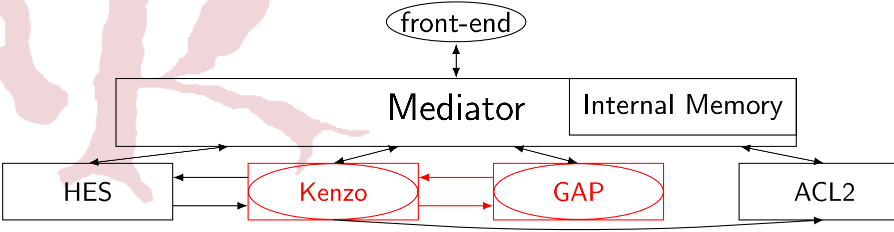

- Aim:
  - Compute homology groups of Eilenberg-MacLane spaces $K(\pi, n)$
- Kenzo → GAP
  - Requests intermediary computations

# Kenzo ↔ GAP




- Aim:
  - Compute homology groups of Eilenberg-MacLane spaces $K(\pi, n)$
- Kenzo $\longrightarrow$ GAP
  - Requests intermediary computations
- Kenzo $\longleftarrow$ GAP
  - Sends results

# Kenzo ↔ GAP




- Aim:
  - Compute homology groups of Eilenberg-MacLane spaces $K(\pi, n)$
- Kenzo → GAP
  - Requests intermediary computations
- Kenzo ← GAP
  - Sends results

A. Romero, G. Ellis and J. Rubio. *Interoperating between Computer Algebra Systems: computing homology groups with Kenzo and GAP*. Proceedings of International Symposium on Symbolic and Algebraic Computation, pages 303-310, 2009.

# Kenzo $\leftrightarrow$ GAP

Previous approach:

1. load the necessary packages and files in GAP and Kenzo,
2. build the cyclic group $G$ in GAP,
3. build a resolution for $G$ using the HAP package,
4. export from GAP the resolution in a file using an OpenMath format,
5. import the resolution to Kenzo,
6. build the cyclic group in Kenzo,
7. assign the resolution to the corresponding cyclic group in Kenzo,
8. build the space $K(G, 1)$ in Kenzo, using the imported resolution and finally
9. compute $H_n(K(G, 1))$ in Kenzo

# Kenzo $\leftrightarrow$ GAP

Previous approach:

1. load the necessary packages and files in GAP and Kenzo,
2. build the cyclic group $G$ in GAP,
3. build a resolution for $G$ using the HAP package,
4. export from GAP the resolution in a file using an OpenMath format,
5. import the resolution to Kenzo,
6. build the cyclic group in Kenzo,
7. assign the resolution to the corresponding cyclic group in Kenzo,
8. build the space $K(G,1)$ in Kenzo, using the imported resolution and finally
9. compute $H_n(K(G,1))$ in Kenzo

Our approach:

1. load the GAP and Computing package,
2. build the cyclic group $G$,
3. build the space $K(G,1)$ and
4. compute $H_n(K(G,1))$

# Kenzo ↔ GAP

Previous approach:

1. load the necessary packages and files in GAP and Kenzo,
2. build the cyclic group $G$ in GAP,
3. build a resolution for $G$ using the HAP package,
4. export from GAP the resolution in a file using an OpenMath format,
5. import the resolution to Kenzo,
6. build the cyclic group in Kenzo,
7. assign the resolution to the corresponding cyclic group in Kenzo,
8. build the space $K(G,1)$ in Kenzo, using the imported resolution and finally
9. compute $H_n(K(G,1))$ in Kenzo

Our approach:

1. load the GAP and Computing package,
2. build the cyclic group $G$,
3. build the space $K(G,1)$ and
4. compute $H_n(K(G,1))$

- Automatization of intermediary steps

# Kenzo ↔ GAP

Previous approach:

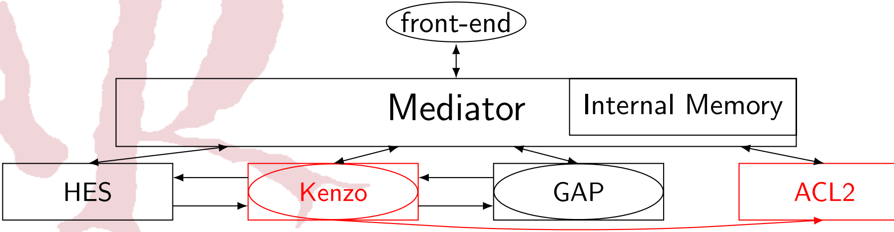
1. load the necessary packages and files in GAP and Kenzo,
2. build the cyclic group $G$ in GAP,
3. build a resolution for $G$ using the HAP package,
4. export from GAP the resolution in a file using an OpenMath format,
5. import the resolution to Kenzo,
6. build the cyclic group in Kenzo,
7. assign the resolution to the corresponding cyclic group in Kenzo,
8. build the space $K(G,1)$ in Kenzo, using the imported resolution and finally
9. compute $H_n(K(G,1))$ in Kenzo

Our approach:

1. load the GAP and Computing package,
2. build the cyclic group $G$,
3. build the space $K(G,1)$ and
4. compute $H_n(K(G,1))$

- Automatization of intermediary steps
- Communication Kenzo ↔ GAP
  - SCSCP protocol

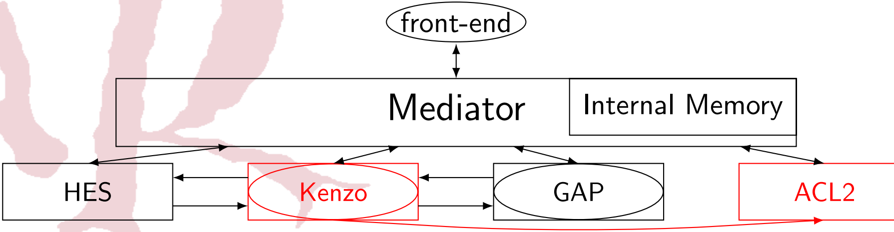
# Kenzo $\longrightarrow$ ACL2




- Aim: Prove truthfulness of Kenzo statements

```
> (cyclicgroup 5)
[K1 Abelian-Group]
```

# Kenzo → ACL2



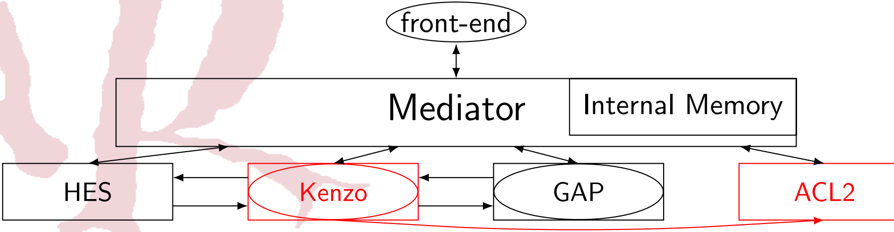

- Aim: Prove truthfulness of Kenzo statements

```
> (cyclicgroup 5)
[K1 Abelian-Group]
```

- Importance:

```
(DEFMETHOD K-G-1 ((group AB-GROUP))
...)
```

# Kenzo → ACL2




- Aim: Prove truthfulness of Kenzo statements

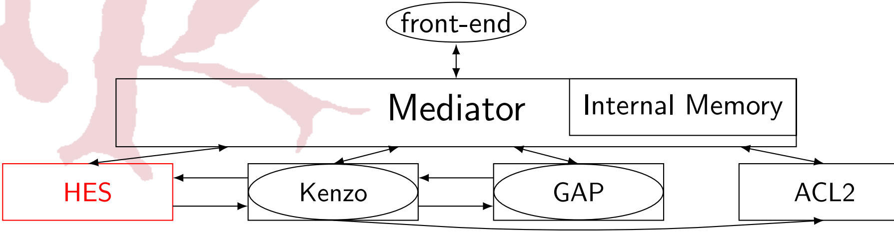
```
> (cyclicgroup 5)
[K1 Abelian-Group]
```

- Importance:

```
(DEFMETHOD K-G-1 ((group AB-GROUP))
...)
```

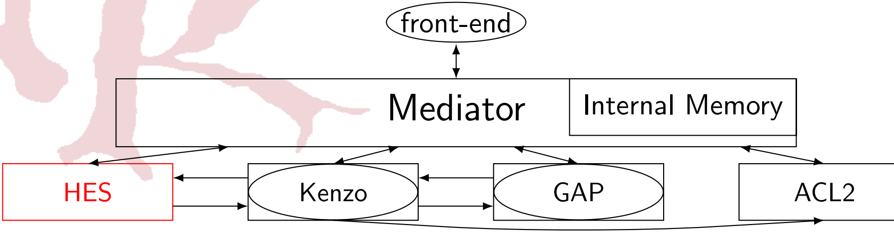
- Communication by means of OMDoc documents

# Homotopy Expert System



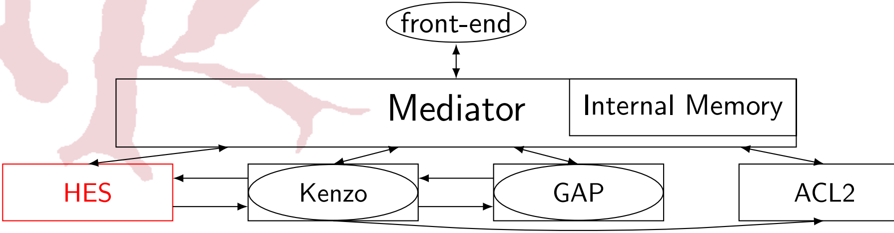

- Homotopy Expert System:

# Homotopy Expert System



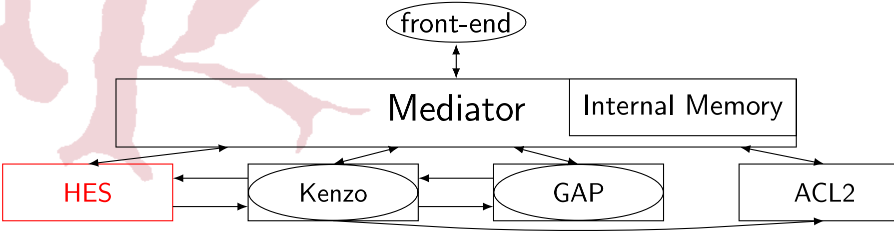

- Homotopy Expert System:
  - Rule based expert system

# Homotopy Expert System



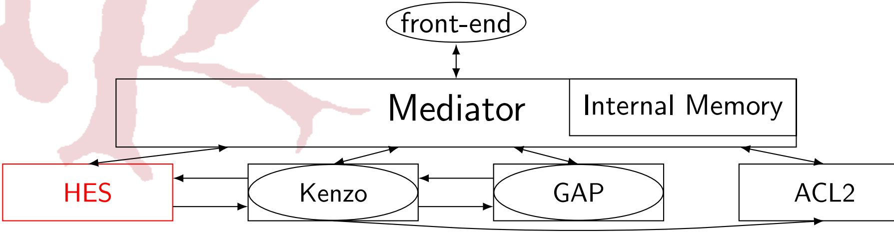

- Homotopy Expert System:
  - Rule based expert system
  - Compute homotopy groups

# Homotopy Expert System



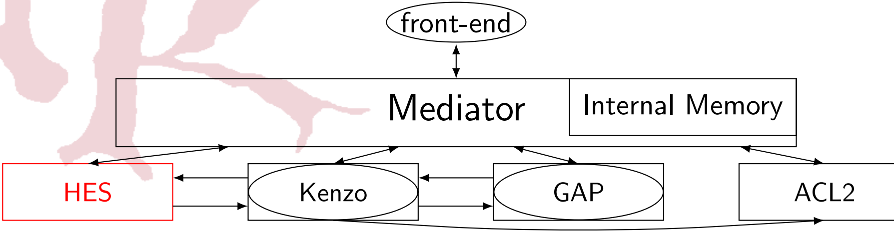

- Homotopy Expert System:
  - Rule based expert system
  - Compute homotopy groups
  - Two kinds of facts: static and dynamic

# Homotopy Expert System



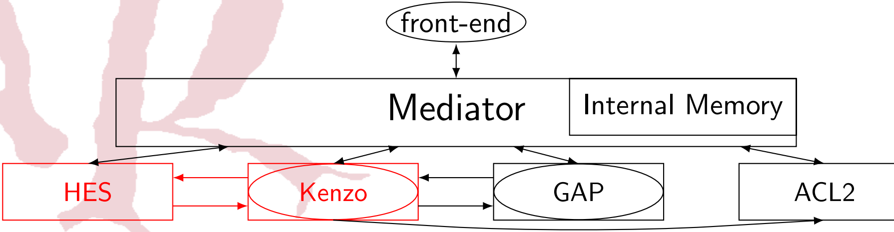

- Homotopy Expert System:
  - Rule based expert system
  - Compute homotopy groups
  - Two kinds of facts: static and dynamic
  - Rules: RuleML and OMDoc files

# Homotopy Expert System



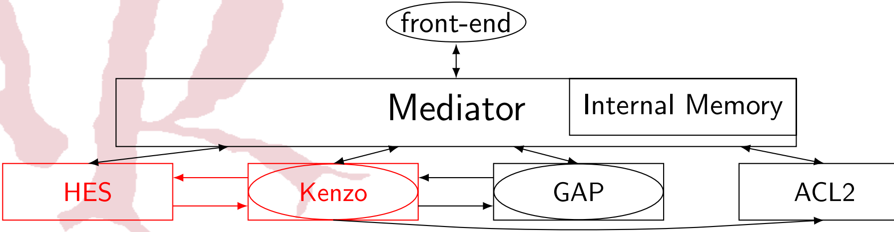

- Homotopy Expert System:
  - Rule based expert system
  - Compute homotopy groups
  - Two kinds of facts: static and dynamic
  - Rules: RuleML and OMDoc files
  - Explanation facility module

# Kenzo ↔ HES



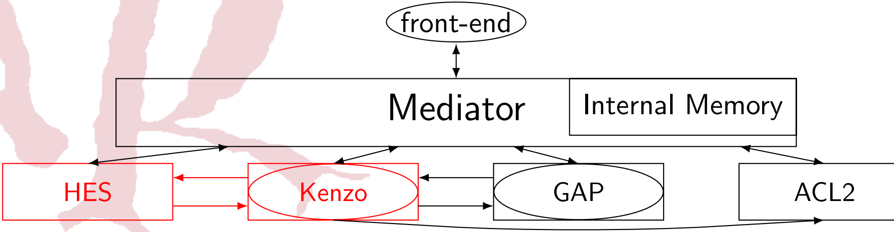

- Aim:
  - Compute homotopy groups of spaces

# Kenzo ↔ HES



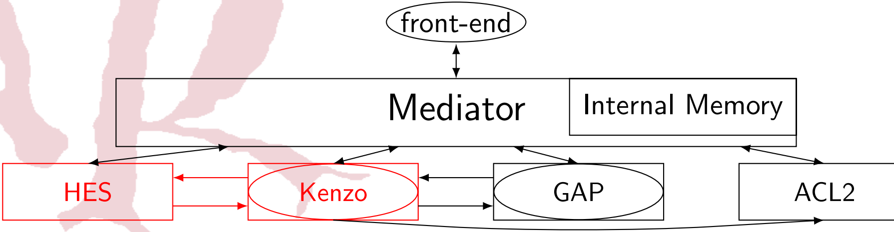

- Aim:
    - Compute homotopy groups of spaces
- Kenzo ← HES
    - Requests homology groups

# Kenzo ↔ HES




- Aim:
  - Compute homotopy groups of spaces
- Kenzo ← HES
  - Requests homology groups
- Kenzo → HES
  - Sends results

# Kenzo ↔ HES




- Aim:
  - Compute homotopy groups of spaces
- Kenzo ← HES
  - Requests homology groups
- Kenzo → HES
  - Sends results
- Example:
  - Compute $\pi_n(S^3)$ for $n = 1, \ldots, 6$

# Kenzo $\leftrightarrow$ HES

1. $S^3$ is 1-connected

# Kenzo $\leftrightarrow$ HES

1. $S^3$ is 1-connected
2. HES asks to Kenzo:
   - $\exists H_n(S^3) \neq 0$ with $1 \leq n \leq 6$ ?

# Kenzo $\leftrightarrow$ HES

1. $S^3$ is 1-connected
2. HES asks to Kenzo:
   - $\exists H_n(S^3) \neq 0$ with $1 \leq n \leq 6$ ?
3. Kenzo computes these homology groups
   - $H_3(S^3) \neq 0$

# Kenzo $\leftrightarrow$ HES

1. $S^3$ is 1-connected
2. HES asks to Kenzo:
    - $\exists H_n(S^3) \neq 0$ with $1 \leq n \leq 6$ ?
3. Kenzo computes these homology groups
    - $H_3(S^3) \neq 0$
4. Kenzo sends the results to HES

# Kenzo $\leftrightarrow$ HES

1. $S^3$ is 1-connected
2. HES asks to Kenzo:
    - $\exists H_n(S^3) \neq 0$ with $1 \leq n \leq 6$ ?
3. Kenzo computes these homology groups
    - $H_3(S^3) \neq 0$
4. Kenzo sends the results to HES
5. HES applies:
    - **IF** $X$ is a 1-connected space
      **AND** $H_r(X) = 0$ for all $1 \leq r < n$
      **AND** $H_s(X)$ is computed for all $1 \leq s \leq n$
      **THEN** $\pi_s(X) = H_s(X)$

# Kenzo $\leftrightarrow$ HES

1. $S^3$ is 1-connected
2. HES asks to Kenzo:
   - $\exists H_n(S^3) \neq 0$ with $1 \leq n \leq 6$ ?
3. Kenzo computes these homology groups
   - $H_3(S^3) \neq 0$
4. Kenzo sends the results to HES
5. HES applies:
   - **IF** $X$ is a 1-connected space
     **AND** $H_r(X) = 0$ for all $1 \leq r < n$
     **AND** $H_s(X)$ is computed for all $1 \leq s \leq n$
     **THEN** $\pi_s(X) = H_s(X)$
   - $\pi_i(S^3) = H_i(S^3) = 0$ for $i = 1, 2$

# Kenzo $\leftrightarrow$ HES

1. $S^3$ is 1-connected
2. HES asks to Kenzo:
   - $\exists H_n(S^3) \neq 0$ with $1 \leq n \leq 6$ ?
3. Kenzo computes these homology groups
   - $H_3(S^3) \neq 0$
4. Kenzo sends the results to HES
5. HES applies:
   - **IF** $X$ is a 1-connected space
     **AND** $H_r(X) = 0$ for all $1 \leq r < n$
     **AND** $H_s(X)$ is computed for all $1 \leq s \leq n$
     **THEN** $\pi_s(X) = H_s(X)$
   - $\pi_i(S^3) = H_i(S^3) = 0$ for $i = 1, 2$
   - $\pi_3(S^3) = H_3(S^3) = \mathbb{Z}$

# Kenzo $\leftrightarrow$ HES

- $\pi_i(S^3)$ for $i = 4, 5, 6$ ?

# Kenzo $\leftrightarrow$ HES

- $\pi_i(S^3)$ for $i = 4, 5, 6$ ?
- HES:

# Kenzo $\leftrightarrow$ HES

- $\pi_i(S^3)$ for $i = 4, 5, 6$ ?
- HES:
  - Algorithm instead of rules

# Kenzo $\leftrightarrow$ HES

- $\pi_i(S^3)$ for $i = 4, 5, 6$ ?
- HES:
  - Algorithm instead of rules
  - Chains several tools available in Kenzo

# Kenzo ↔ HES

- $\pi_i(S^3)$ for $i = 4, 5, 6$ ?
- HES:
  - Algorithm instead of rules
  - Chains several tools available in Kenzo
  - "last appeal" tool

# Kenzo $\leftrightarrow$ HES

- $\pi_i(S^3)$ for $i = 4, 5, 6$ ?
- HES:
  - Algorithm instead of rules
  - Chains several tools available in Kenzo
  - "last appeal" tool
  - high complexity

# Table of Contents

1 General View of the system

2 Interoperability among modules

3 Putting all together

4 Conclusions and Further Work

# Putting all together

- Demo:
    - Computations with Kenzo
    - Computations with GAP
    - Computations with HES
    - Computations with Kenzo + GAP
    - Computations with Kenzo + HES
    - Certifications with ACL2

# Table of Contents

1 General View of the system

2 Interoperability among modules

3 Putting all together

4 Conclusions and Further Work

# Conclusions and Further Work

- Conclusions

# Conclusions and Further Work

- Conclusions
  - Integration of different tools for computing and reasoning

# Conclusions and Further Work

- Conclusions
  - Integration of different tools for computing and reasoning
  - OpenMath provides interoperability

# Conclusions and Further Work

- Conclusions
  - Integration of different tools for computing and reasoning
  - OpenMath provides interoperability
  - HES for computing homotopy groups

# Conclusions and Further Work

- Conclusions
  - Integration of different tools for computing and reasoning
  - OpenMath provides interoperability
  - HES for computing homotopy groups
- Further Work

# Conclusions and Further Work

- Conclusions
  - Integration of different tools for computing and reasoning
  - OpenMath provides interoperability
  - HES for computing homotopy groups
- Further Work
  - Give more resources

# Conclusions and Further Work

- Conclusions
  - Integration of different tools for computing and reasoning
  - OpenMath provides interoperability
  - HES for computing homotopy groups
- Further Work
  - Give more resources
  - Improve interaction with ACL2

# Integrating multiple sources to answer questions in Algebraic Topology

Jónathan Heras Vico Pascual Ana Romero Julio Rubio

*Departamento de Matemáticas y Computación*
Universidad de La Rioja
Spain

9th International Conference on Mathematical Knowledge Management
MKM 2010, Paris