ToolsForHomalg

Tools for the homalg project

Version 2014.12.08

December 2014

Mohamed Barakat

Sebastian Gutsche

Markus Lange-Hegermann

(this manual is still under construction)

This manual is best viewed as an HTML document. The latest version is available ONLINE at:

http://wwwb.math.rwth-aachen.de/~gutsche/gap_packages/ToolsForHomalg/chap0.html

An OFFLINE version should be included in the documentation subfolder of the package. This package is part of the homalg-project:

http://homalg.math.rwth-aachen.de/index.php/core-packages/toolsforhomalg
Mohamed Barakat
Email: barakat@mathematik.uni-kl.de
Homepage: http://www.mathematik.uni-kl.de/~barakat/
Address: Department of Mathematics,
University of Kaiserslautern,
67653 Kaiserslautern,
Germany

Sebastian Gutsche
Email: sebastian.gutsche@rwth-aachen.de
Homepage: http://wwwb.math.rwth-aachen.de/~gutsche/
Address: Lehrstuhl B für Mathematik, RWTH Aachen, Templer-
graben 64, 52056 Aachen, Germany

Markus Lange-Hegermann
Email: markus.lange.hegermann@rwth-aachen.de
Homepage: http://wwwb.math.rwth-aachen.de/~markus/
Address: Lehrstuhl B für Mathematik, RWTH Aachen, Templer-
graben 64, 52056 Aachen, Germany
Copyright

© 2007-2014 by Mohamed Barakat, Sebastian Gutsche, and Markus Lange-Hegermann

This package may be distributed under the terms and conditions of the GNU Public License Version 2.
## Contents

1 Introduction 4
  1.1 What is the role of the ToolsForHomalg package in the homalg project? 4

2 Installation of the ToolsForHomalg Package 5

3 ToDo-list 6
  3.1 Blueprints 6
  3.2 ToDo-list entries 8
  3.3 Category 10
  3.4 Constructor 10
  3.5 This is the magic 10
  3.6 Methods for all objects 11
  3.7 Proof tracking 11
  3.8 Maintainance 11

4 ToolsForHomalg automatic generated documentation 13
  4.1 ToolsForHomalg automatic generated documentation of global variables 13

5 Basics 14
  5.1 Categories 14

6 Pointers 16
  6.1 Weak pointer objects 16
  6.2 Pointer objects 17

7 Tools 18
  7.1 Functions 18
  7.2 Examplefunctions 23

8 Trees 24
  8.1 Trees 24

References 26

Index 27
Chapter 1

Introduction

1.1 What is the role of the ToolsForHomalg package in the homalg project?

1.1.1 ToolsForHomalg provides ...

The package ToolsForHomalg provides for the homalg project [hpa10]:

4
Chapter 2

Installation of the **ToolsForHomalg** Package

To install this package just extract the package’s archive file to the GAP pkg directory.

By default the **ToolsForHomalg** package is not automatically loaded by GAP when it is installed.

You must load the package with

```
LoadPackage( "ToolsForHomalg" );
```

before its functions become available.

Please, send me an e-mail if you have any questions, remarks, suggestions, etc. concerning this package. Also, I would be pleased to hear about applications of this package.

Mohamed Barakat
Chapter 3

ToDo-list

3.1 Blueprints

3.1.1 ToDoListEntryToMaintainEqualAttributes (for IsList, IsList, IsList)

\[\text{ToDoListEntryToMaintainEqualAttributes(indicator, objects, attributes)}\]

**Returns:** a todo list entry

The first argument is the `indicator`. It is a list of sources like in ToDoListEntry. Each entry \(SP\) has to be a threetuple. First entry of \(SP\) has to an object, for which the second entry of \(SP\), which has to be the name of an attribute, must become known. Once the attribute is known to the object, it will be compared to the third entry of the list. This can be a value, which is compared directly a function, which is launched and its return value is compared, or a list, consisting of a function and arguments, so the return value of the function with given arguments is compared. If there is no third entry in \(SP\), it is only looked up if the value is known. Once all entries in `indicator` are processed like this, and all returned true in the comparasion, a list of ToDoListEntryForEqualAttributes is installed. They are installed for the two entries of the list `objects` which can either be the objects itself or a list containing a function and arguments, which return value is used. For each entry in `attributes` such an entry is installed. Such an entry can be the name of an attribute, if both objects in `objects` should share the value between attributes with the same name, or a list of two names, if the attributes do not have the same name.

3.1.2 ToDoListEntryToMaintainFollowingAttributes (for IsList, IsList, IsList)

\[\text{ToDoListEntryToMaintainFollowingAttributes(indicator, objects, attributes)}\]

**Returns:** a todo list entry

This function creates a ToDoListEntry which can install several ToDoListEntries. The first two arguments, `indicator` and `objects` except that there will be only ToDoListEntries installed between the two objects in `objects`. Each entry in `attributes` can either be a string which means that the attribute with the given name will be set from the first to the second object in `objects` once it is known. The third argument `attributes` is a list of attributes that will be propagated by ToDoListEntries. Each entry `TP` can either be a list consisting of a DescriptionOfImplication string and one of the following or just one of the following lists: It can be a string, which means that the Attribute with the given name will be propagated from the first to the second object. It can be a list, consisting of
two entries, where the first entry is a list of sources like in `ToDoListEntry` and the second might be a function which will be launched once the first part is fulfilled. It can also be a threetuple which will serve as second to fourth argument of `ToDoListEntry`. Or it can be a string, which will set the attribute named like this of the first object to the one named in the second object.

### 3.1.3 `ToDoListEntry (for IsList, IsList)`

- **Returns**: a todo list entry

  This function allows to create more than one `ToDoListEntry` with identical list of sources at one time. First argument is a list of sources like in the other `ToDoListEntry` functions. Second argument is a list of threetuples, which serve as second to fourth argument of `ToDoListEntry` or a function, which serves as second argument for `ToDoListEntry` or a tuple with a description string and one of the above.

### 3.1.4 `ToDoList_this_object`

- **Returns**: nothing

  Represents the objects for which the blueprint is created in the arguments.

### 3.1.5 `ToDoListEntryToMaintainEqualAttributesBlueprint (for IsObject, IsList, IsList)`

- **Returns**: nothing

  This function installs an immediate method which can install `ToDoListEntryToMaintainEqualAttributesBlueprint` is installed with the second to fourth argument as first to third. In those attributes, at any point, the variable `ToDoList_this_object` can be used. When the entry is installed, this will be replaced with the object to which the filters became known, i.e. the one which triggered the immediate method.

### 3.1.6 `ToDoListEntryToMaintainFollowingAttributesBlueprint (for IsObject, IsList, IsList)`

- **Returns**: nothing

  The same as `ToDoListEntryToMaintainEqualAttributesBlueprint` for `ToDoListEntryToMaintainFollowingAttributes`.

### 3.1.7 `ToDoListEntryBlueprint (for IsObject, IsList, IsList)`

- **Returns**: nothing

  The same as `ToDoListEntryToMaintainEqualAttributesBlueprint` for `ToDoListEntry`
3.2  ToDo-list entries

3.2.1  AddToToDoList (for IsToDoListEntry)

▷ AddToToDoList$(E)$  (operation)

Returns:

 Adds the ToDo-list entry $E$ to the ToDo-lists of its source objects and creates a new one, if this is needed. This function might be called with lists of entries

3.2.2  SourcePart (for IsToDoListEntry)

▷ SourcePart$(entry)$  (operation)

Returns: a list

Returns the a list of source parts of the ToDo-list entry $entry$. This is a triple of an object, a name of a filter/attribute, and a value to which the attribute has to be set to activate the entry

3.2.3  TargetPart (for IsToDoListEntry)

▷ TargetPart$(entry)$  (operation)

Returns: a list

Returns the target part of the ToDo-list entry $entry$. This is a triple of an object, a name of a filter/attribute, and a value to which the specific filter/attribute should be set. The third entry of the list might also be a function to which return value the attribute is set.

3.2.4  ProcessAToDoListEntry (for IsToDoListEntry)

▷ ProcessAToDoListEntry$(arg)$  (operation)

Returns: a boolean

Processes a ToDo-list entry, i.e. sets the information given in TargetPart if the definitions in SourcePart are fulfilled. Returns a function if the entry could be processed, false if not, and fail if SourcePart or TargetPart weren't available anymore.

3.2.5  ToDoListEntry (for IsList, IsObject, IsString, IsObject)

▷ ToDoListEntry$(arg1, arg2, arg3, arg4)$  (operation)

Returns: a ToDoListEntry

The first argument must be a list consisting of two, three or four-tuples where the first entry must be the object to which the attribute given as a string in the second entry must be known to process this entry. The second entry can also be a list of strings, in that case all the attributes given as names must be known. Also, in this case, only two entries in this tuple are allowed. The third part can be a value or a list, consisting of a function followed by arguments which will be computed by the time the attribute given as second entry becomes known to the first entry. If the second part is only a string, and there is a third entry in the tuple the attribute is compared to the third entry. One can set a comparating function as fourth entry, which must take two entries and return false or true. If the value of the attribute matches the (computed) value in the third entry for all members of the list in the first argument the attribute given as third argument, also by name, of the second argument is set to the value of the fourth argument. This can also be a list which has to be computed, or a function, which return value is used in this case.
3.2.6  ToDoListEntry (for IsList, IsFunction)

ToDoListEntry(arg1, arg2)

Returns: a ToDoListEntry
The first argument is a list of three-tuples like above. Once all preconditions become fulfilled the
function given as second argument is launched.

3.2.7  SetTargetValueObject (for IsToDoListEntry, IsObject)

SetTargetValueObject(entry, value)

Returns: nothing
If the given value of the target part is the return value of a function this command sets the target
value of the entry to a function. This is done to keep proof tracking available.

3.2.8  SetTargetObject (for IsToDoListEntry, IsObject)

SetTargetObject(entry, obj)

Returns: nothing
If the target object, i.e. the first entry of the target part, was given as a function, this method can
set this entry to the return value computed in ProcessToDoListEntry. This happens atomatically, do
not worry about it.

3.2.9  ToDoListEntryWithContraposition (for IsObject, IsString, IsBool, IsObject, Is-
String, IsBool)

ToDoListEntryWithContraposition(sobj, source_prop, sval, tobj, target, tval)

Returns: a ToDoListEntry
Creates a ToDoListEntry which also installs a contraposition. The arguments source_prop and
target need to be strings which name a property, and sval and tval need to be boolean values, i.e.
true or false.

3.2.10 DescriptionOfImplication (for IsToDoListEntry)

DescriptionOfImplication(arg)

Returns: a list
Has to be set to a string, which describes the reason for the conclusion. If the ToDo-list entry is
displayed, the given string will be displayed with a because before it.

3.2.11 ToDoListEntryForEqualAttributes (for IsObject, IsString, IsObject, IsString)

ToDoListEntryForEqualAttributes(arg1, arg2, arg3, arg4)

Returns: a ToDoListEntry
Creates a ToDoListEntry for two equal attributes, which means that both values of the two at-
tributes will be propagated in both directions.
3.2.12 **ToDoListEntryForEquivalentAttributes** (for IsObject, IsString, IsObject, IsObject, IsString, IsObject)

\[\text{ToDoListEntryForEquivalentAttributes}(\text{arg1}, \text{arg2}, \text{arg3}, \text{arg4}, \text{arg5}, \text{arg6})\]

**Returns:** a ToDoListEntry

Creates a ToDoListEntry for two equivalent attributes, which means that both values of the two attributes will be propagated in both directions. Please note that this one does NOT implement contrapositions.

### 3.3 Category

3.3.1 **IsToDoList** (for IsObject)

\[\text{IsToDoList}(\text{arg})\]

**Returns:** true or false

This is the category of ToDo-lists. Every ToDo-list is an object of this category, which basically contains the ToDo-lists.

### 3.4 Constructor

3.4.1 **NewToDoList** (for )

\[\text{NewToDoList}(\text{arg})\]

**Returns:** nothing

Creates a new empty ToDo-list.

### 3.5 This is the magic

3.5.1 **Process_A_ToDo_List_Entry**

\[\text{Process_A_ToDo_List_Entry}(\text{arg})\]

**Returns:** a boolean

Gets a ToDo-list entry, which is a pair of a list of strings and a weak pointer object, and processes it. If the action was done, it returns true, if not, it returns false, and it returns fail if the action is not possible anymore due to deleted objects.

3.5.2 **ProcessToDoList** (for IsObject)

\[\text{ProcessToDoList}(\text{A})\]

**Returns:** nothing

This is the magic! This attribute is never set. Creating an ToDo-list entry installs an ImmediateMethod for this attribute for the specific category of the object to which ToDo-list is added, and the filter the entry contains. It is then triggeert if the filters become applicable, so the ToDo-list is processed.
3.6 Methods for all objects

3.6.1 ToDoList (for IsObject)

\[
\text{ToDoList}(\text{arg}) \quad \text{(attribute)}
\]

Returns: A ToDo-list

Returns the ToDo-list of an object, or creates a new one.

3.7 Proof tracking

This is a way to track proofs from ToDoLists. Not only for debugging, but also for knowing how things work together.

3.7.1 TraceProof (for IsObject, IsString, IsObject)

\[
\text{TraceProof}(\text{obj}, \text{name}, \text{val}) \quad \text{(operation)}
\]

Returns: a tree

If the object \text{obj} has the attribute \text{name}, and its value is \text{val}, and the knowledge has been obtained trough ToDoList-entries, this method traces the way the property was set, and returns a tree which describes the full way of how the attribute became known.

3.8 Maintainance

3.8.1 ActivateToDoList (for IsObject)

\[
\text{ActivateToDoList}(\text{arg}) \quad \text{(operation)}
\]

Returns: nothing

This operation activates ToDoLists for the argument.

3.8.2 ActivateToDoList (for)

\[
\text{ActivateToDoList}(\text{arg}) \quad \text{(operation)}
\]

Returns: nothing

This operation activates ToDoLists for all objects.

3.8.3 DeactivateToDoList (for IsObject)

\[
\text{DeactivateToDoList}(\text{arg}) \quad \text{(operation)}
\]

Returns: nothing

This operation deactivates ToDoLists for the argument.

3.8.4 DeactivateToDoList (for)

\[
\text{DeactivateToDoList}(\text{arg}) \quad \text{(operation)}
\]

Returns: nothing

This operation deactivates ToDoLists for all objects. Note that it is not possible to activate ToDoList for a single object while they are not activated. ToDoListEntries will yet be stored for
all objects that can have ToDoLists. All objects created while ToDoLists are deactivated have by
default no ToDoList.

3.8.5 ActivateWhereInfosInEntries

\textbf{ActivateWhereInfosInEntries}(arg) \hspace{1cm} \text{(function)}

\textbf{Returns:} nothing

Stores the result of Where(100) in an entry if the entry is triggered. This is not activated by
default, since it might slow down the system.

3.8.6 DeactivateWhereInfosInEntries

\textbf{DeactivateWhereInfosInEntries}(arg) \hspace{1cm} \text{(function)}

\textbf{Returns:} nothing

Deactives the storage of the result of Where(100) if an entry is triggered. This is the default.
Chapter 4

ToolsForHomalg automatic generated documentation

4.1 ToolsForHomalg automatic generated documentation of global variables

4.1.1 HOMALG_TOOLS

A central place for configurations. ChapterInfo Basics, Variables
Chapter 5

Basics

5.1 Categories

5.1.1 IsStructureObjectOrObjectOrMorphism (for IsAttributeStoringRep)

\[ \text{IsStructureObjectOrObjectOrMorphism}(\text{arg}) \]

\textbf{Returns: true or false}

This is the super super GAP-category which will include the GAP-categories IsStructureObjectOrObject and IsHomalgObjectOrMorphism

5.1.2 IsStructureObjectOrObject (for IsStructureObjectOrObjectOrMorphism)

\[ \text{IsStructureObjectOrObject}(\text{arg}) \]

\textbf{Returns: true or false}

This is the super GAP-category which will include the GAP-categories IsHomalgRing, IsHomalgModule, IsHomalgRingOrModule and IsHomalgComplex

5.1.3 IsStructureObject (for IsStructureObjectOrObject)

\[ \text{IsStructureObject}(\text{arg}) \]

\textbf{Returns: true or false}

This is the super GAP-category which will include the GAP-categories IsHomalgRing we need this GAP-category to define things like Hom(M,R) as easy as Hom(M,N) without distinguishing between structure objects (e.g. rings) and objects (e.g. modules)

5.1.4 IsStructureObjectMorphism (for IsAttributeStoringRep)

\[ \text{IsStructureObjectMorphism}(\text{arg}) \]

\textbf{Returns: true or false}

This is the super GAP-category which will include the GAP-categories IsHomalgRingMap, etc.

5.1.5 IsHomalgRingOrModule (for IsStructureObjectOrObject)

\[ \text{IsHomalgRingOrModule}(\text{arg}) \]

\textbf{Returns: true or false}
this is the super GAP-category which will include the GAP-categories IsHomalgRing, IsHomalgModule:
Chapter 6

Pointers

6.1 Weak pointer objects

6.1.1 IsContainerForWeakPointers (for IsComponentObjectRep)

▷ IsContainerForWeakPointers(arg) (filter)
   Returns: true or false
   The category for weak pointer objects

6.1.2 ContainerForWeakPointers

▷ ContainerForWeakPointers(arg) (function)
   Returns: a list which can store weak pointers
   The constructor for lists of weak pointers.

6.1.3 UpdateContainerOfWeakPointers (for IsContainerForWeakPointers)

▷ UpdateContainerOfWeakPointers(arg) (operation)
   Returns:
   Updates the weak pointers in a container and deletes the empty ones

6.1.4 _AddElmWPObj_ForHomalg

▷ _AddElmWPObj_ForHomalg(arg) (function)
   Returns:
   Adds a weak pointer of an objects to a weak pointer list.

6.1.5 _AddTwoElmWPObj_ForHomalg

▷ _AddTwoElmWPObj_ForHomalg(arg) (function)
   Returns:
   Adds a weak pointer which depends on two objects to a list of weak pointers
6.1.6  _ElmWPObj_ForHomalg (for IsContainerForWeakPointers, IsObject, IsObject)

▷ _ElmWPObj_ForHomalg(arg1, arg2, arg3)  (operation)

    Returns:
    Creates a weak pointer depending on two objects and adds it to the container.

6.2  Pointer objects

6.2.1  IsContainerForPointers (for IsComponentObjectRep)

▷ IsContainerForPointers(arg)  (filter)

    Returns: true or false
    The category for pointer objects

6.2.2  ContainerForPointers

▷ ContainerForPointers(arg)  (function)

    Returns: a container for pointers
    Creates a container for pointers.

6.2.3  UpdateContainerOfPointers (for IsContainerForPointers)

▷ UpdateContainerOfPointers(arg)  (operation)

    Returns:
    Updates the container of pointers, removes old.

6.2.4  _AddElmPObj_ForHomalg

▷ _AddElmPObj_ForHomalg(arg)  (function)

    Returns:
    Adds a pointer to an object to a container for pointers.

6.2.5  _AddTwoElmPObj_ForHomalg

▷ _AddTwoElmPObj_ForHomalg(arg)  (function)

    Returns:
    Adds a pointer to two objects to a container for pointers

6.2.6  _ElmPObj_ForHomalg (for IsContainerForPointers, IsObject, IsObject)

▷ _ElmPObj_ForHomalg(arg1, arg2, arg3)  (operation)

    Returns: an object
    Returns an object which a pointer refers to.
Chapter 7

Tools

7.1 Functions

7.1.1 homalgTotalRuntimes

\[ \text{homalgTotalRuntimes}(\text{arg}) \] (function)

**Returns:**

an integer

A tool to compute the runtime of several methods.

7.1.2 AddLeftRightLogicalImplicationsForHomalg

\[ \text{AddLeftRightLogicalImplicationsForHomalg}(\text{arg}) \] (function)

**Returns:**

A tool to install equivalence between filters.

7.1.3 LogicalImplicationsForOneHomalgObject

\[ \text{LogicalImplicationsForOneHomalgObject}(\text{arg}) \] (function)

**Returns:**

Installs a logical implication for one type with all it’s contrapositions.

7.1.4 LogicalImplicationsForTwoHomalgBasicObjects

\[ \text{LogicalImplicationsForTwoHomalgBasicObjects}(\text{arg}) \] (function)

**Returns:**

7.1.5 InstallLogicalImplicationsForHomalgBasicObjects

\[ \text{InstallLogicalImplicationsForHomalgBasicObjects}(\text{arg}) \] (function)

**Returns:**

7.1.6 LeftRightAttributesForHomalg

\[ \text{LeftRightAttributesForHomalg}(\text{arg}) \] (function)

**Returns:**
7.1.7 InstallLeftRightAttributesForHomalg

InstallLeftRightAttributesForHomalg(arg) (function)

Returns:

7.1.8 MatchPropertiesAndAttributes

MatchPropertiesAndAttributes(arg) (function)

Returns:
A method to match the properties and attributes of two objects.

7.1.9 InstallImmediateMethodToPullPropertyOrAttribute

InstallImmediateMethodToPullPropertyOrAttribute(arg) (function)

Returns:
Installs methods to pull new known properties and attributes from one object to another.

7.1.10 InstallImmediateMethodToConditionallyPullPropertyOrAttribute

InstallImmediateMethodToConditionallyPullPropertyOrAttribute(arg) (function)

Returns:
Installs methods to pull new known properties and attributes under certain conditions from one object to another.

7.1.11 InstallImmediateMethodToPullPropertyOrAttributeWithDifferentName

InstallImmediateMethodToPullPropertyOrAttributeWithDifferentName(arg) (function)

Returns:
Installs an immediate method which can pull a property from one object to another with different names.

7.1.12 InstallImmediateMethodToPullPropertiesOrAttributes

InstallImmediateMethodToPullPropertiesOrAttributes(arg) (function)

Returns:
Installs an immediate method to pull several properties or attributes from one object to another.

7.1.13 InstallImmediateMethodToPullTrueProperty

InstallImmediateMethodToPullTrueProperty(arg) (function)

Returns:
Installs an immediate method to pull a property if it is true.

7.1.14 InstallImmediateMethodToConditionallyPullTrueProperty

InstallImmediateMethodToConditionallyPullTrueProperty(arg) (function)

Returns:
Installs an immediate method which conditionally pulls a property if it is true.
7.1.15 **InstallImmediateMethodToPullTruePropertyWithDifferentName**

▷ **InstallImmediateMethodToPullTruePropertyWithDifferentName** (arg) (function)

Returns:
Installs an immediate method which pulls a property with a different name if it is true.

7.1.16 **InstallImmediateMethodToPullTrueProperties**

▷ **InstallImmediateMethodToPullTrueProperties** (arg) (function)

Returns:
Installs an immediate method which pulls several properties if they are true

7.1.17 **InstallImmediateMethodToPullFalseProperty**

▷ **InstallImmediateMethodToPullFalseProperty** (arg) (function)

Returns:
Installs an immediate method to pull a property if it is false.

7.1.18 **InstallImmediateMethodToConditionallyPullFalseProperty**

▷ **InstallImmediateMethodToConditionallyPullFalseProperty** (arg) (function)

Returns:
Installs an immediate method which conditionally pulls a property if it is false.

7.1.19 **InstallImmediateMethodToPullFalsePropertyWithDifferentName**

▷ **InstallImmediateMethodToPullFalsePropertyWithDifferentName** (arg) (function)

Returns:
Installs an immediate method which pulls a property with a different name if it is false.

7.1.20 **InstallImmediateMethodToPullFalseProperties**

▷ **InstallImmediateMethodToPullFalseProperties** (arg) (function)

Returns:
Installs an immediate method which pulls several properties if they are false.

7.1.21 **InstallImmediateMethodToPushPropertyOrAttribute**

▷ **InstallImmediateMethodToPushPropertyOrAttribute** (arg) (function)

Returns:
Installs an immediate method to push a property from one object to another.

7.1.22 **InstallImmediateMethodToConditionallyPushPropertyOrAttribute**

▷ **InstallImmediateMethodToConditionallyPushPropertyOrAttribute** (arg) (function)

Returns:
Installs an immediate method to conditionally push a property from one object to another.
7.1.23  InstallImmediateMethodToPushPropertyOrAttributeWithDifferentName

- InstallImmediateMethodToPushPropertyOrAttributeWithDifferentName(arg)  (function)
  Returns:
  Installs an immediate method which can push a property from one object to another with different names.

7.1.24  InstallImmediateMethodToPushPropertiesOrAttributes

- InstallImmediateMethodToPushPropertiesOrAttributes(arg)  (function)
  Returns:
  Installs an immediate method to push several properties or attributes from one object to another.

7.1.25  InstallImmediateMethodToPushTrueProperty

- InstallImmediateMethodToPushTrueProperty(arg)  (function)
  Returns:
  Installs an immediate method to push a property if it is true.

7.1.26  InstallImmediateMethodToPushTruePropertyWithDifferentName

- InstallImmediateMethodToPushTruePropertyWithDifferentName(arg)  (function)
  Returns:
  Installs an immediate method which pushes a property with a different name if it is true.

7.1.27  InstallImmediateMethodToPushTrueProperties

- InstallImmediateMethodToPushTrueProperties(arg)  (function)
  Returns:
  Installs an immediate method which pushes several properties if they are true.

7.1.28  InstallImmediateMethodToPushFalseProperty

- InstallImmediateMethodToPushFalseProperty(arg)  (function)
  Returns:
  Installs an immediate method to push a property if it is false.

7.1.29  InstallImmediateMethodToPushFalsePropertyWithDifferentName

- InstallImmediateMethodToPushFalsePropertyWithDifferentName(arg)  (function)
  Returns:
  Installs an immediate method which pushes a property with a different name if it is false.

7.1.30  InstallImmediateMethodToPushFalseProperties

- InstallImmediateMethodToPushFalseProperties(arg)  (function)
  Returns:
  Installs an immediate method which push several properties if they are false.
7.1.31 DeclareAttributeWithCustomGetter

- DeclareAttributeWithCustomGetter(arg) (function)
  
  Returns: Installs an attribute with a custom getter function.

7.1.32 AppendToAhomalgTable

- AppendToAhomalgTable(arg) (function)
  
  Returns: Appends an entry to a homalg table.

7.1.33 homalgNamesOfComponentsToIntLists

- homalgNamesOfComponentsToIntLists(arg) (function)
  
  Returns: a list of integers
  Creates a list of integers out of the names of components.

7.1.34 IncreaseExistingCounterInObject

- IncreaseExistingCounterInObject(arg) (function)
  
  Returns: Increases an existing counter in an object.

7.1.35 IncreaseExistingCounterInObjectWithTiming

- IncreaseExistingCounterInObjectWithTiming(arg) (function)
  
  Returns: Increases an existing counter on an object with timing.

7.1.36 IncreaseCounterInObject

- IncreaseCounterInObject(arg) (function)
  
  Returns: Increases a counter in an object and creates one if it not exists

7.1.37 MemoryToString

- MemoryToString(arg) (function)
  
  Returns: Converts the current memory state to a string

7.1.38 PrimePowerExponent

- PrimePowerExponent(n, p) (function)
  
  Returns: A nonnegative integer
  Returns the $p$-exponent of the integer $n$, where $p$ is a rational prime.
7.1.39 ViewList (for IsList)

- ViewList(L) (operation)
  - Returns: nothing
  - Apply ViewObj to the list L.

7.1.40 homalgLaTeX (for IsObject)

- homalgLaTeX(arg) (operation)
  - Returns:

7.2 Examplefunctions

7.2.1 ExamplesForHomalg (for )

- ExamplesForHomalg(arg) (operation)
  - Returns: true or false
  - Runs the examples for homalg if the package is loadable.

7.2.2 ExamplesForHomalg (for IsInt)

- ExamplesForHomalg(arg) (operation)
  - Returns: true or false
  - Runs the named example for homalg
Chapter 8

Trees

The trees are used in ToDoLists. They are a technical feature, and fairly general, so they also can be used somewhere else.

8.1 Trees

8.1.1 IsTree (for IsObject)

<table>
<thead>
<tr>
<th>IsTree(arg)</th>
<th>returns: true or false</th>
</tr>
</thead>
<tbody>
<tr>
<td>The category of trees. A tree may have a content, a list of successors, a predecessor and it knows if it is a leave of a tree or not.</td>
<td></td>
</tr>
</tbody>
</table>

8.1.2 Content (for IsTree)

<table>
<thead>
<tr>
<th>Content(arg)</th>
<th>returns: object</th>
</tr>
</thead>
<tbody>
<tr>
<td>The content of the tree. May be any object.</td>
<td></td>
</tr>
</tbody>
</table>

8.1.3 ListOfSuccessors (for IsTree)

<table>
<thead>
<tr>
<th>ListOfSuccessors(arg)</th>
<th>returns: a list of trees</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns the list of successors of a tree.</td>
<td></td>
</tr>
</tbody>
</table>

8.1.4 Predecessor (for IsTree)

<table>
<thead>
<tr>
<th>Predecessor(arg)</th>
<th>returns: a tree or fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns the predecessor of a tree, or fail if there is none.</td>
<td></td>
</tr>
</tbody>
</table>

8.1.5 ListOfSentinels (for IsTree)

<table>
<thead>
<tr>
<th>ListOfSentinels(arg)</th>
<th>returns: a list</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns a list of leaves of the tree.</td>
<td></td>
</tr>
</tbody>
</table>
8.1.6 RemoveHead (for IsTree)

- **RemoveHead**(arg)  
  **Returns:** a tree  
  Returns the first successor of the tree, and adds all other successors of the tree to the tree that is returned. If the tree is a leave, it returns an empty tree. If the tree is empty, it returns the tree itself.

8.1.7 Tree (for )

- **Tree**(arg)  
  **Returns:** a tree  
  Returns an empty tree.

8.1.8 Tree (for IsObject)

- **Tree**(obj)  
  **Returns:** a tree  
  Returns a tree with argument obj.

8.1.9 Add (for IsTree, IsTree)

- **Add**(tree, new_tree)  
  **Returns:** nothing  
  Adds the [list of] tree[s] new_tree as successor to the tree tree.

8.1.10 ContentListFromSentinelToHead (for IsTree)

- **ContentListFromSentinelToHead**(sent)  
  **Returns:** a list  
  Returns a list of the contents of the trees from the leave sent up to the content of the head of the tree.

8.1.11 PostOrder (for IsTree)

- **PostOrder**(arg)  
  **Returns:** a list  
  Returns the contents of the nodes of the tree in post-order.
References

Index

ToolsForHomalg, 4
ActivateToDoList
  for , 11
    for IsObject, 11
ActivateWhereInfosInEntries, 12
Add
  for IsTree, IsTree, 25
  _AddElmPObj_ForHomalg, 17
  _AddElmWPObj_ForHomalg, 16
AddLeftRightLogicalImplicationsForHomalg, 18
AddToDoList
  for IsToDoListEntry, 8
  _AddTwoElmPObj_ForHomalg, 17
  _AddTwoElmWPObj_ForHomalg, 16
AddToAhomalgTable, 22
ContainerForPointers, 17
ContainerForWeakPointers, 16
Content
  for IsTree, 24
  ContentListFromSentinelToHead
    for IsTree, 25
DeactivateToDoList
  for , 11
    for IsObject, 11
DeactivateWhereInfosInEntries, 12
DeclareAttributeWithCustomGetter, 22
DescriptionOfImplication
  for IsToDoListEntry, 9
_ElmPObj_ForHomalg
  for IsContainerForPointers, IsObject, IsObject, 17
_ElmWPObj_ForHomalg
  for IsContainerForWeakPointers, IsObject, IsObject, 17
ExamplesForHomalg
  for , 23
    for IsInt, 23
homalgLaTeX
  for IsObject, 23
homalgNamesOfComponentsToIntLists, 22
HOMALG_TOOLS, 13
homalgTotalRuntimes, 18
IncreaseCounterInObject, 22
IncreaseExistingCounterInObject, 22
IncreaseExistingCounterInObjectWithTiming, 22
InstallImmediateMethodToConditionallyPullFalseProperty, 20
InstallImmediateMethodToConditionallyPullPropertyOrAttribute, 19
InstallImmediateMethodToConditionallyPullTrueProperty, 19
InstallImmediateMethodToConditionallyPullPropertyOrAttribute, 20
InstallImmediateMethodToPullFalseProperties, 20
InstallImmediateMethodToPullFalseProperty, 20
InstallImmediateMethodToPullFalsePropertyWithDifferentName, 20
InstallImmediateMethodToPullPropertiesOrAttributes, 19
InstallImmediateMethodToPullPropertyOrAttribute, 19
InstallImmediateMethodToPullPropertyOrAttributeWithDifferentName, 19
InstallImmediateMethodToPullTrueProperties, 20
InstallImmediateMethodToPullTrueProperty, 19
InstallImmediateMethodToPullTruePropertyWithDifferentName, 20
InstallImmediateMethodToPushFalse-Properties, 21
InstallImmediateMethodToPushFalse-Property, 21
InstallImmediateMethodToPushFalse-PropertyWithDifferentName, 21
InstallImmediateMethodToPush-PropertiesOrAttributes, 21
InstallImmediateMethodToPushProperty-OrAttribute, 20
InstallImmediateMethodToPushProperty-OrAttributeWithDifferentName, 21
InstallImmediateMethodToPushTrue-Properties, 21
InstallImmediateMethodToPushTrue-Property, 21
InstallImmediateMethodToPushTrue-PropertyWithDifferentName, 21
InstallLeftRightAttributesForHomalg, 19
InstallLogicalImplicationsForHomalg-BasicObjects, 18
IsContainerForPointers for IsComponentObjectRep, 17
IsContainerForWeakPointers for IsComponentObjectRep, 16
IsHomalgRingOrModule for IsStructureObjectOrObject, 14
IsStructureObject for IsStructureObjectOrObject, 14
IsStructureObjectMorphism for IsAttributeStoringRep, 14
IsStructureObjectOrObject for IsStructureObjectOrObjectOrMorphism, 14
IsStructureObjectOrObjectOrMorphism for IsAttributeStoringRep, 14
IsToDoList for IsObject, 11
IsToDoListEntry for IsToDoListEntry, IsObject, 9
IsToDoListEntryBlueprint for IsObject, IsList, IsList, 7
IsToDoListEntryForEqualAttributes for IsObject, IsString, IsObject, IsString, IsObject, 10
IsToDoListEntryToMaintainEqualAttributes for IsObject, IsString, IsObject, IsObject, IsString, IsObject, 10
InstallLogicalImplicationsForOneHomalgObject, 18
LogicalImplicationsForTwoHomalgBasic-Objects, 18
MatchPropertiesAndAttributes, 19
MemoryToString, 22
NewToDoList for IsObject, 10
PostOrder for IsTree, 25
Predecessor for IsTree, 24
PrimePowerExponent, 22
ProcessAToDoListEntry for IsToDoListEntry, 8
ProcessA_Todo_List_Entry, 10
ProcessToDoList for IsObject, 10
RemoveHead for IsTree, 25
SetTargetObject for IsToDoListEntry, IsObject, 9
SetTargetValueObject for IsToDoListEntry, IsObject, 9
SourcePart for IsToDoListEntry, 8
TargetPart for IsToDoListEntry, 8
ToDoList for IsObject, 11
ToDoListEntry for IsList, IsFunction, 9
for IsList, IsList, 7
for IsList, IsObject, IsString, IsObject, 8
ToDoListEntryBlueprint for IsObject, IsList, IsList, 7
ToDoListEntryForEqualAttributes for IsObject, IsString, IsObject, IsString, 9
ToDoListEntryForEquivalentAttributes for IsObject, IsString, IsObject, IsObject, IsString, IsObject, 10
ToDoListEntryToMaintainEqualAttributes for IsObject, IsString, IsObject, IsObject, IsString, IsObject, 10
ToolsForHomalg

for IsList, IsList, IsList, 6
ToDoListEntryToMaintainEqualAttributesBlueprint
for IsObject, IsList, IsList, IsList, 7
ToDoListEntryToMaintainFollowingAttributes
for IsList, IsList, IsList, 6
ToDoListEntryToMaintainFollowingAttributesBlueprint
for IsObject, IsList, IsList, IsList, 7
ToDoListEntryWithContraposition
ToDoList_this_object, 7
TraceProof
for IsObject, IsString, IsObject, 11
Tree
for , 25
for IsObject, 25
UpdateContainerOfPointers
for IsContainerForPointers, 17
UpdateContainerOfWeakPointers
for IsContainerForWeakPointers, 16
ViewList
for IsList, 23