Biddy

1.7.1

Generated by Doxygen 1.8.11

Contents

Chapter 1

USER MANUAL

TL;DR

Biddy is a multi-platform academic Binary Decision Diagrams package. It supports ROBDDs with complemented edges, 0-sup-BDDs with complemented edges, and TZBDDs.

Biddy is capable of all the typical operations regarding Boolean functions and BDDs.

Biddy is a library to be included in your C and C++ projects:

```
1 #define UNIX
2 #define USE_BIDDY
3 #include "/path/to/biddy.h"
```

To compile Biddy library:

```
1 biddy> make dynamic
2 biddy> make clean
```

Dependencies:

- on GNU/Linux, you need libgmp (https://gmplib.org/).
- on MS Windows, you need MPIR library (http://mpir.org/).

When using Biddy on GNU/Linux, you may have to tell bash about the library:

```
1 export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/absolute/path/to/biddy/library
```

There are two aditional packages included into Biddy distribution:

- bddview is a pure Tcl/Tk script for visualization of BDDs,
- BDD Scout is a demo application demonstrating the capability of Biddy and bddview.

Biddy is free software maintained by Robert Meolic (robert.meolic@um.si) at University of Maribor, Slovenia.

Homepage: http://biddy.meolic.com/

1. AN OVERVIEW

Biddy supports ROBDDs with complemented edges as described in "K.S. Brace, R.L. Rudell, R.E. Bryant. Efficient Implementation of a BDD Package. 27th ACM/IEEE DAC, pages 40-45, 1990".

Biddy supports 0-sup-BDDs with complemented edges as described in "S. Minato. Zero-Suppressed BDDs for Set Manipulation in Combinatorial Problems, 30th ACM/IEEE DAC, pages 272-277, 1993".

Biddy supports TZBDDs (tagged zero-supressed binary decison diagrams) as introduced by Robert Meolic in 2016 (to be published).

Biddy includes:

- automatic garbage collection with a system age (a variant of a mark-and-sweep approach),
- node management through formulae protecting,
- variable swapping and sifting algorithm (ROBDDs, only).

Biddy is optimized for efficiency, but it is mainly oriented towards readable and comprehensible source code in C.

Biddy is currently used in the following projects:

- BDD Scout, demo project which allows visualization of BDDs and also includes some benchmarks
- Efficient Symbolic Tools (EST), model checking and other algorithms for formal verification of systems (http://est.meolic.com/)

2. SOURCE CODE

Biddy is free software released under GPL.

The short name of Biddy package is 'biddy'. This name is placed in front of all filenames and external identifiers. It may appear in all lowercase, or with its first letter capitalized, written as 'biddy' and 'Biddy', respectively.

There are three categories of C functions.

- Exported functions are visible outside the package.
- Internal functions are visible to all files within the package.
- Static functions are visible to the file only.

There are two types of C functions.

- General functions which operates on a particular BDD and considering only graph properties (i.e. changing edge's mark, selecting nodes, counting nodes etc.). These functions are the same for different type of decision diagrams (BDD, ZDD, etc.). Functions, which add or delete nodes or those which needs info about variables (e.g. name) are not general functions. Exported general functions have prefix Biddy_.
- Managed functions, which operates on a global properties of a BDD system (e.g. node table, variable table, formula table, various caches, etc.) or consider a BDD as a Boolean function (e.g. Boolean operations, counting minterms, etc.). These functions need info stored in a manager. Exported managed functions have prefix Biddy_Managed_.

Biddy consists of the following files:

- README.md (this file)
- CHANGES (history of changes)
- COPYING (license file)
- VERSION (project's version)
- Makefile (used to produce binary code)
- Makefile.Linux (Makefile definitions for GNU/Linux)
- Makefile.MINGW (Makefile definitions for MS Windows)
- Makefile.Darwin (Makefile definitions for MacOS)
- biddy.h (header)
- biddyInt.h (header)
- biddyMain.c (main functions, ROBDDs only, deprecated)
- biddyMainGDD.c (main functions)
- biddyStat.c (functions for statistic)
- biddyInOut.c (parsers and generators for Boolean functions)
- package-source (script used to build distribution)
- package-bin (script used to build distribution)
- package-bin.bat (script used to build distribution)
- package-tgz (script used to build distribution)
- package-deb (script used to build distribution)
- package-rpm (script used to build distribution)
- debian/∗ (files used when creating deb package)
- rpm/∗ (files used when creating rpm package)
- biddy-example-8queens.c (8 Queens example)
- biddy-example-independence.c (8 Queens example)
- biddy-example-independence-usa.c (Independence example)
- biddy-example-independence-europe.c (Independence example)
- biddy-example-random.c (Random formulae example)
- biddy-example-hanoi.c (Tower of Hanoi example)

There are two C headers, external and internal. The external header file, named biddy.h, defines features visible from outside the package. The internal header file, named biddyInt.h defines features used in multiple files inside the package, but not outside.

There are two aditional packages included into Biddy distribution:

- bddview is a pure Tcl/Tk script for visualization of BDDs.
- BDD Scout is a demo application demonstrating the capability of Biddy and bddview.

Details about building are given in Section 4.

3. USING BIDDY LIBRARY

Precompiled packages include dynamically linked library (i.e. ∗.so on GNU/Linux, ∗.dll on MS Windows, ∗.dylib on Mac OS X), and the appropriate C header biddy.h. Currently, there are no interfaces for other programming languages.

For linking with Biddy library you have to use:

```
1 -lbiddy -lgmp
```

Biddy is capable of all the typical operations regarding Boolean functions and BDDs.

The following code is an example of usage. Please note, that functions for node management are not shown. Moreover, Biddy has a manager but its usage is optional and it is also not shown in the given example.

IMPORTANT: You should define UNIX, MACOSX, or MINGW. You should define USE_BIDDY iff you are using Biddy via dynamic library.

```
1 #define UNIX
2 #define USE_BIDDY
3 #include "/path/to/biddy.h"
4
5 #define Str2Var(x) (Biddy_GetVariable(Biddy_String)x)
6
7 int main() {
8   Biddy_Edge f,g,h,r;
9
10   Biddy_Init();
11
12   f = Biddy_Eval1((Biddy_String)"(OR H E L L O)"); /* PREFIX INPUT */
13   g = Biddy_Eval1((Biddy_String)"(AND W O R L D)"); /* PREFIX INPUT */
14   h = Biddy_Eval2((Biddy_String)"~(H * E * L * L)"); /* INFIX INPUT */
15
16   /* BASIC OPERATION */
17   r = Biddy_Xor(f,g);
18
19   /* REPLACE SOME VARIABLES */
20   Biddy_ResetVariablesValue();
21   Biddy_SetVariableValue(Str2Var("H"),Biddy_FoaVariable((Biddy_String)"L"));
22   Biddy_SetVariableValue(Str2Var("K"),Biddy_FoaVariable((Biddy_String)"L"));
23   Biddy_SetVariableValue(Str2Var("W"),Biddy_FoaVariable((Biddy_String)"L"));
24   r = Biddy_Replace(r);
25
26   /* SIMPLE RESTRICTION */
27   r = Biddy_Restrict(r,Str2Var("E"),Biddy_GetConstantZero());
28
29   /* COUDERT AND MADRE'S RESTRICT FUNCTION */
30   r = Biddy_Simplify(r,h);
31
32   /* SOME STATS */
33   printf("Function r depends on %u variables.\n",Biddy_VariableNumber(r));
34   printf("Function r has %.0f minterms.\n",Biddy_CountMinterm(r,0));
35   printf("BDD for function r has %u nodes.\n",Biddy_NodeNumber(r));
36
37   /* TRUTH TABLE */
38   printf("Here is a truth table for function r\n");
39   Biddy_WriteTable(r);
40
41   /* GRAPHVIZ/DOT OUTPUT */
42   Biddy_WriteDot("biddy.dot",r,"Function r");
43   printf("USE 'dot -y -Tpng -O biddy.dot' to visualize BDD for function r.\n");
44
45   Biddy_Exit();
46 }
```

3.1 NODE MANAGEMENT THROUGH FORMULAE PROTECTING

Biddy includes powerful node management based on formulae tagging. There are six user functions to maintain nodes.

Biddy_AddFormula(name,bdd,c)

Nodes of the given BDD will be preserved for the given number of cleanings. If (name != NULL) then formula is accessible by its name. If formula with a given name already exists it is overwritten. If (c == 0) then formula is persistently preserved and you have to use Biddy_DeleteFormula to remove its nodes. There are two macros defined to simplify formulae management. Macro Biddy_AddTmpFormula(bdd,c) is defined as Biddy_AddFormula(NUL↩L,bdd,c) and macro Biddy_AddPersistentFormula(name,bdd) is defined as Biddy_AddFormula(name,bdd,0).

Biddy_DeleteFormula(name)

Nodes of the given formula are tagged as not needed. Formula is not accessible by its name anymore. Regular cleaning with Biddy_Clean is not considered this tag.

Biddy_Clean()

Discard all nodes which were not preserved or which are not preserved anymore. Obsolete nodes are not immediately removed, they will be removed during the first garbage collection. Use Biddy_Purge or Biddy_PurgeAnd↩Reorder to immediately remove all non-preserved nodes.

Biddy_Purge()

Immediately remove all nodes which were not preserved or which are not preserved anymore. Nodes from deleted prolonged formulae and nodes from deleted fortified formulae are removed if they are not needed by other formulae. Call to Biddy_Purge does not count as cleaning and thus all preserved formulae remains preserved for the same number of cleanings.

Biddy_PurgeAndReorder(bdd)

The same as Biddy_Purge but also trigger reordering on function (if BDD is given) or global reordering (if NULL is given).

Biddy_Refresh(bdd)

All obsolete nodes become fresh nodes. This is an external variant of internal function BiddyRefresh. It is needed to implement user caches.

3.2. EXAMPLES OF NODE MANAGEMENT WITH BIDDY

The first example is a straightforward calculation.

```
1 f1 = op(...);
2 f2 = op(...);
3 g1 = op(f1,f2,...);
4 Biddy_AddTmpFormula(g1,1); /* g1 is preserved for next cleaning */
5 f1 = op(...);
6 f2 = op(...);
7 g2 = op(f1,f2,...);
8 Biddy_AddTmpFormula(g2,1); /* g2 is preserved for next cleaning */
9 Biddy_Clean(); /* g1 and g2 are still usable, f1 and f2 are obsolete */
10 result = op(g1,g2,...);
11 Biddy_AddPersistentFormula("result",result); /* final result is permanently preserved */
12 Biddy_Clean(); /* g1 and g2 are not needed, anymore */
```

If additional garbage collection is needed also after the calculation of g1, you can use the following code after the calculation of g1:

```
1 Biddy_AddTmpFormula(g1,2); /* g1 is preserved for next two cleanings */
2 Biddy_Clean(); /* g1 remains preserved for next cleaning */
```

In this approach, f1 and f2 become obsolete after Biddy_Clean, but their nodes are not immediately removed (automatic garbage collection is only started when there are no free nodes in the system). If garbage collection should be started immediately, you must use the following code after the calculation of g1:

```
1 Biddy_AddTmpFormula(g1,2); /* g1 is preserved for next two cleanings */
2 Biddy_Clean(); /* g1 remains preserved for next cleaning */
3 Biddy_Purge(); /* keep only preserved (g1) formulae */
```

Alternatively, you can use the following approach:

```
1 Biddy_AddTmpFormula(g1,1); /* g1 is preserved for next cleaning */
2 Biddy_Purge(); /* this will not make g1 obsolete */
```

To trigger reordering in this example, you should use Biddy_PurgeAndReorder to get the following code:

```
1 f1 = op(...);
2 f2 = op(...);
3 g1 = op(f1,f2,...);
4 Biddy_AddTmpFormula(g1,2); /* g1 is preserved for next two cleanings */
5 Biddy_PurgeAndReorder(NULL); /* keep only preserved formulae (g1), perform reordering */
6 Biddy_Clean(); /* g1 remains preserved for one additional cleaning */
7 f1 = op(...);
8 f2 = op(...);
9 g2 = op(f1,f2,...);
10 Biddy_AddTmpFormula(g2,1); /* g2 is preserved for next cleaning */
11 Biddy_PurgeAndReorder(NULL); /* keep only preserved formulae (g1, g2), perform reordering */
12 Biddy_Clean(); /* g1 and g2 are still usable but not preserved */
13 result = op(g1,g2,...);
14 Biddy_AddPersistentFormula("result",result); /* result is permanently preserved */
15 Biddy_PurgeAndReorder(NULL); /* keep only preserved formulae (result), perform reordering */
```

The second example is an iterative calculation:

```
1 f = op(...);
2 result = op(f,...);
3 while (!finish) {
4   Biddy_AddTmpFormula(result,1); /* result is preserved for next cleaning */
5   Biddy_Clean(); /* result is still usable but not preserved */
6   f = op(...);
7   g = op(f,...);
8   result = op(result,g,...);
9 }
10 Biddy_AddPersistentFormula("result",result); /* final result is permanently preserved */
11 Biddy_Clean(); /* temp results are not needed, anymore */
```

If garbage collection is needed also after the calculation of g, you must use the following code:

```
1 f = op(...);
2 result = op(f,...);
3 while (!finish) {
4   Biddy_AddTmpFormula(result,2); /* result is preserved for next two cleanings */
5   Biddy_Clean(); /* result remains preserved for one additional cleaning */
6   f = op(...);
7   g = op(f,...);
8   Biddy_AddTmpFormula(g,1); /* g is preserved for next cleaning */
9   Biddy_Clean(); /* result and g are still usable but not preserved */
10   result = op(result,g,...);
11 }
12 Biddy_AddPersistentFormula("result",result); /* final result is permanently preserved */
13 Biddy_Clean(); /* temp results are not needed, anymore */
```

To trigger reordering in the second example, you should change code in the following way:

```
1 f = op(...);
2 result = op(f,...);
3 while (!finish) {
4   Biddy_AddTmpFormula(result,2); /* result is preserved for next two cleanings */
5   Biddy_PurgeAndReorder(NULL); /* keep only preserved formulae (result), perform reordering */
6   Biddy_Clean(); /* result remains preserved for one additional cleaning */
7   f = op(...);
8   g = op(f,...);
9   Biddy_AddTmpFormula(g,1); /* g is preserved for next cleaning */
10   Biddy_Clean(); /* result and g are still usable but not preserved */
11   result = op(result,g,...);
12 }
13 Biddy_AddPersistentFormula("result",result); /* final result is permanently preserved */
14 Biddy_PurgeAndReorder(NULL); /* keep only preserved formulae (result), perform reordering */
```

The third example is an outline of an implementation of model checking where we are trying to benefit from regularly reordering:

```
1 sup = Prepare(...);
2 Biddy_AddPersistentFormula("sup",sup) /* sup is permanently preserved */
3 Z = 0;
4 last = 1;
5 while (Z!=last) {
6   Biddy_AddTmpFormula(Z,1); /* Z is preserved for next cleaning */
7   Biddy_PurgeAndReorder(NULL); /* keep only preserved formulae (sup,Z), perform reordering */
8   Biddy_Clean(); /* Z remains usable but not preserved */
9   last = Z;
10   Z = NextSet(Z,sup,...);
11 }
12 result = Z;
13 Biddy_AddPersistentFormula("result",result); /* final result is permanently preserved */
14 Biddy_DeleteFormula("sup"); /* optional, if you really need to remove sup */
15 Biddy_Purge(); /* optional, immediately remove non-preserved nodes */
```

The fourth example is an outline of an implementation of bisimulation where we are trying to benefit from regularly reordering:

```
1 init = AND(init_p,init_q)
2 Biddy_AddPersistentFormula("init",init) /* init is permanently preserved */
3 eq = InitialEq(init_p,tr_p,init_q,tr_q,...);
4 do {
5   Biddy_AddTmpFormula(eq,1); /* eq is preserved for next cleaning */
6   Biddy_PurgeAndReorder(NULL); /* keep only preserved formulae (init, eq), perform reordering */
7   Biddy_Clean(); /* eq remains usable but not preserved */
8   last = eq;
9   eq1 = NextEqPart1(eq,tr_p,tr_q,...);
10   eq2 = NextEqPart2(eq,tr_p,tr_q,...);
11   eq = AND(eq1,eq2);
12 } while (AND(init,eq)!=0 && eq!=last)
13 if (AND(init,eq)!=0) return false; else return true;
14 Biddy_DeleteFormula("init"); /* optional, if you really need to remove init */
15 Biddy_Purge(); /* optional, immediately remove non-preserved nodes */
```

The fifth example is an outline of an implementation of parallel composition where we are trying to benefit from intensive GC:

```
1 sacc = snew = AND(init_1,init_2,...,init_N);
2 for (i=1;i<=N;i++) di[i] = 0;
3 for (i=1;i<=N;i++) for (j=1;i<=N;j++) dij[i,j] = 0;
4 do {
5   Biddy_AddTmpFormula(snew,N*(N+1)); /* snew is preserved just long enough */
6   Biddy_AddTmpFormula(sacc,N*(N+1)); /* sacc is preserved just long enough */
7   new1 = 0;
8   for (i=1;i<=N;i++) {
9     sup = OneStep(snew,tr_i,...);
10     di[i] = OR(d[i],sup);
11     new1 = OR(new1,NextState(sup,...));
12     Biddy_AddTmpFormula(d[i],N*(N+1)); /* di[i] is preserved just long enough */
13     Biddy_AddTmpFormula(new1,1); /* new1 is preserved for next cleaning */
14     Biddy_Clean(); /* new1 remains usable but not preserved */
15   }
16   Biddy_AddTmpFormula(new1,N*N); /* new1 is preserved just long enough */
17   new2 = 0;
18   for (i=1;i<=N;i++) for (j=1;j<=N;j++) {
19     sup = OneStep(snew,tr_i,tr_j,...);
20     dij[i,j] = OR(d[i,j],sup);
21     new2 = OR(new2,NextState(sup,...);
22     Biddy_AddTmpFormula(dij[i,j],N*(N+1)); /* dij[i,j] is preserved just long enough */
23     Biddy_AddTmpFormula(new2,1); /* new2 is preserved for next cleaning */
24     Biddy_Clean(); /* new2 remains usable but not preserved */
25   }
26   snew = AND(OR(new1,new2),NOT(sacc));
27   sacc = OR(sacc,snew);
28 } while (snew!=0)
29 tr1 = 0;
30 for (i=1;i<=N;i++) {
31   sup = AddStab(di[i],...);
32   tr1 = OR(tr1,sup);
33   Biddy_AddTmpFormula(tr1,1); /* tr1 is preserved for next cleaning */
34   Biddy_Clean(); /* tr1 remains usable but not preserved */
35 }
36 Biddy_AddTmpFormula(tr1,N*N); /* tr1 is preserved just long enough */
37 tr2 = 0;
38 for (i=1;i<=N;i++) for (j=1;j<=N;j++) {
39   sup = AddStab(dij[i,j],...);
40   tr2 = OR(tr2,sup);
41   Biddy_AddTmpFormula(tr2,1); /* tr2 is preserved for next cleaning */
42   Biddy_Clean(); /* tr2 remains usable but not preserved */
43 }
44 result = OR(tr1,tr2);
45 Biddy_AddPersistentFormula("result",result); /* final result is permanently preserved */
46 Biddy_Clean(); /* temp results are not needed, anymore */
```

3.3 GARBAGE COLLECTION WITH A SYSTEM AGE

Garbage collection is automatically triggered if nodes from all reserved blocks of nodes are used. Garbage collection will remove as many obsolete nodes as possible.

Biddy does not use reference counter. We call the implemented algorithm "GC with a system age". It is a variant of a mark-and-sweep approach. Please note, that it relies on additional structure (Formula table).

Using system age instead of reference counter has some advantages. It allows GC to be started in any time without breaking the ongoing calculation. Thus, there is no need to taking care of repeating broken calculations. Moreover, the usage of system age instead of reference counter removes all the hassle of referencing and dereferencing nodes and thus it is favorable in an academic package oriented towards simple and readable source code.

There are four classes of nodes. Every node belongs to one of these classes:

- **fortified** node (expiry value = 0);
- **fresh** node (expiry value = biddySystemAge);
- **prolonged** node (expiry value $>$ biddySystemAge);
- **obsolete** node (0 $<$ expiry value $<$ biddySystemAge).

Before GC, nodes must be refreshed in such a way that no successor of a non-obsolete node is obsolete. This can be achieved with relative simple loop which will check each node at least once.

There are three internal functions to maintain nodes.

- **BiddyProlongOne** (one node is prolonged)
- **BiddyProlongRecursively** (one node and its successors are prolonged recursively until a non-obsolete node is reached)
- **BiddyIncSystemAge** (all fresh nodes become obsolete nodes)

There are four functions which can be used by an expert user.

- **Biddy_Garbage** (explicite garbage collection call, removes all obsolete nodes)
- **Biddy_SwapWithLower** (explicite swap of two variables, removes all obsolete and fresh nodes)
- **Biddy_SwapWithHigher** (explicite swap of two variables, removes all obsolete and fresh nodes)
- **Biddy_Sifting** (explicite sifting, removes all obsolete and fresh nodes)

These four functions will keep all nodes preserved by Biddy_AddFormula and they will not change the class of any node. Please note, that Biddy_Garbage can be started in any time whilst Biddy_SwapWithLower, Biddy_Swap↩ WithHigher, and Biddy_Sifting will break an ongoing calculation (because they remove fresh nodes).

3.4 MORE DETAILS OF MEMORY MANAGEMENT (NODE CHAINING)

Biddy relies on a single hash table for all variables. However, it supports chaining of nodes to form different lists (using an extra pointer in each node). This facility is used to improve efficiency of garbage collection and sifting.

Please node, that node chaining is not determined or limited by using formulae protecting schema or a system age approach, it is an independent mechanism.

4. BUILDING PACKAGES

Compiling Biddy library

On GNU/Linux, we are using gcc.

```
1 biddy> make dynamic
2 biddy> make clean
```

Alternatively, you can use:

```
1 biddy> make static
2 biddy> make debug
3 biddy> make profile
```

You can use specify the target folder:

```
1 biddy> make dynamic "BINDIR = ./bin"
2 biddy> make clean "BINDIR = ./bin"
```

On MS Windows, we are using MSYS2. We use pacman to prepare the environment:

```
1 MSYS shell> pacman -Syuu
2 MSYS shell> pacman -S mingw-w64-i686-gcc
3 MSYS shell> pacman -S mingw-w64-x86_64-gcc
4 MSYS shell> pacman -S make
5 MSYS shell> pacman -S bison
6 MSYS shell> pacman -S gdb
7 MSYS shell> pacman -S nano
8 MSYS shell> pacman -S tar
9 MSYS shell> pacman -S subversion
```

Alternatively, you can use Visual Studio for building. There is a prepared solution consisting of many projects. You need to adapt include and lib folders.

```
1 ./VS/Biddy.sln
```

To produce nice setup files, we use Advanced Installer (http://www.advancedinstaller.com/). We have been granted a free licence. MANY THANKS!

Dependencies

On GNU/Linux, we are using libgmp (https://gmplib.org/).

On MS Windows, we are using MPIR library (http://mpir.org/).

Creating Biddy library as a zip package

```
1 biddy> ./package-bin
```

You need a zip program.

On MS Windows, you need 7-Zip (http://7-zip.org/) - and it has a strange use of -x! You also need file 7zsd_All_x64.sfx that you should download as part of "7z SFX Tools" from http://7zsfx.info/en/ and put in the directory containing 7z.exe.

You install the resulting package by extracting libraries to the appropriate directory (may be local, e.g. user's home directory).

When using this package on GNU/Linux, you have to tell bash about the library:

```
1 export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/absolute/path/to/biddy/library
```

Creating zip file with source code of a complete Biddy project

```
1 biddy> ./package-source
```

If available, source code of bddview and BDD Scout will be included, too.

Creating packages for GNU/Linux

```
1 biddy> ./package-tgz
2 biddy> ./package-deb
3 biddy> ./package-rpm
```

These scripts are intedend to be used on Ubuntu. These scripts needs release number as an argument. Script package-tgz must be invoked before running package-deb. Debian packages must be created before RPM packages.

./package-tgz should create orig.tar.gz file and prepare directories for creating debian and RPM packages. You can run ./package-tgz only if version changes.

./package-deb should create debian packages (libbidy and libbiddy-dev). They are tested on Ubuntu system.

./package-rpm should create RPM packages (libbidy and libbiddy-devel). They are tested on openSUSE system.

Creating demo aplication bddscout

You need complete sources for biddy, bddview, and bddscout, Scripts are located in biddy/bddscout.

```
1 bddscout> ./package-bin
2 bddscout> ./package-tgz
3 bddscout> ./package-deb
4 bddscout> ./package-rpm
```

package-bin should create BDD Scout (staticaly linked with Biddy library). The script will produce a zip file. You install BDD Scout by simply unzip to the target directory.

./package-tgz should create orig.tar.gz file and prepare directories for creating debian and RPM packages. You can run ./package-tgz only if version changes.

./package-deb should create debian packages (bddscout, bddscout-bra, bddscout-ifip, bddscout-bddtrace, bddscout-ifip-data, and bddscout-bddtrace-data). They are tested on Ubuntu system.

./package-rpm should create RPM packages (bddscout, bddscout-bra, bddscout-ifip, bddscout-bddtrace, bddscout-ifip-data, and bddscout-bddtrace-data). They are tested on openSUSE system.

5. HISTORY

Biddy is based on a BDD package written in Pascal in 1992 as a student project. At that time, it was a great work and the paper about it won a second place at IEEE Region Student Paper Contest (Paris-Evry, 1993). The paper was published by IEEE as "A. Casar, R. Meolic. Representation of Boolean functions with ROBDDs. IEEE Student paper contest : regional contest winners 1990-1997 : prize-winning papers demonstrating student excellence worldwide, Piscataway, 2000" and can be obtained from http://research.meolic.com/papers/robdd.pdf

In 1995, this BDD package was rewritten in C. Later, this BDD package become an integral part of EST package, a prototype tool for formal verification of concurrent systems. The homepage for EST project is http://est.↩meolic.com/

In 2003, BDD package from EST was included in the report presented at 16th Symposium on Integrated Circuits and Systems Design (SBCCI'03). The report is available as a paper "G. Janssen. A Consumer Report on BDD Packages. IBM T.J. Watson Research Center. 2003". Get it from http://doi.ieeecomputersociety.↩org/10.1109/SBCCI.2003.1232832

In 2006, BDD package in EST got the name Biddy.

In 2007, a main part of Biddy package was extracted from EST forming a separate package called Biddy. The code has been reorganized in such a way, that EST is not using internal structures (e.g. node table) directly but using the provided API only.

In 2007, we created local svn repository for maintaining the source code (not live, anymore).

On May 15, 2008, Biddy v1.0 was released containing also bddview v0.95 (Tcl/Tk BDD viewer) and Bdd Scout v0.90 (demo application).

In 2009, 2010, and 2012 an updated version of Biddy v1.0 was released which added support for debian packaging, support for RPM packaging, fix error in Biddy_E and Biddy_A, and fix and improve details of documentation, packaging, and Tcl/Tk GUI.

In 2012, a research paper about Biddy library appears in Journal of Software (doi:10.4304/jsw.7.6.1358-1366) http://ojs.academypublisher.com/index.php/jsw/article/view/jsw070613581366/

In 2013, Biddy v1.1 was released. Biddy_Edge became a pointer instead of structure and all other structures were optimized.

In 2014, Biddy v1.2 was released. Variable swapping and sifting algorithm were the most significant additions.

In 2014, svn repositories for biddy, bddview and bddscout are moved to Savannah. http://svn.savannah.nongnu.org/viewvc/?root=biddy

In 2015, Biddy v1.3, v.1.4 and v1.5 was released. Various input/output methods have been added. Support for 64-bit arhitectures and support for Visual Studio projects were improved. Biddy got a manager. Many CUDD-like functions have been added. Comment's style changed to support doxygen. HTML and PDF documentation were produced.

Also in 2015, Personal Package Archive ppa:meolic/biddy has been created https://launchpad.net/~meolic/+archive/ubuntu/biddy

Also in 2015, sources became available on GitHub https://github.com/meolic/biddy

In 2016, Biddy v1.6 was released. Formulae are not recursively refreshed all the time. The size of Node table became resizable.

In 2017, Biddy v1.7 was released. Terminology is changed a lot, e.g. "formulae counter" is now "system age". Added support for 0-sup-BDDs and TZBDDs. Implemented creation and manipulaton of non-anonymous managers. Added manipulation of combination sets. Sifting does not require removing fresh nodes. Many new CUDD-like functions have been added. bddview and BDD Scout have been significantly improved.

6. PUBLICATIONS

If you find our work useful, please, cite us.

- Robert Meolic. **Biddy - a multi-platform academic BDD package.** Journal of Software, 7(6), pp. 1358-1366, 2012. http://ojs.academypublisher.com/index.php/jsw/article/view/jsw070613581366
- Robert Meolic. **Biddy: ???** We are preparing a paper for SCP.

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 4

Data Structure Documentation

4.1 Biddy_XY Struct Reference

`#include <biddy.h>`

4.1.1 Detailed Description

Biddy_XY is used in Biddy_WriteBddview to pass node coordinates

Definition at line 236 of file biddy.h.

The documentation for this struct was generated from the following file:

- biddy.h

Chapter 5

File Documentation

5.1 biddy.h File Reference

File biddy.h contains declaration of all external data structures.

```
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <stdarg.h>
#include <time.h>
```

Data Structures

- struct Biddy_XY

Macros

- #define Biddy_IsNull(f) (f == NULL)
- #define Biddy_IsConstant(f) ((((void**)((uintptr_t) f & ~((uintptr_t) 1)))[2] == NULL) && (((void**)((uintptr_t) f & ~((uintptr_t) 1)))[3] == NULL))
- #define Biddy_IsTaggedConstant(f) (((uintptr_t) f >> 48 == 0) && (((void**)((uintptr_t) f & ~((uintptr_t) 1)))[2] == NULL) && (((void**)((uintptr_t) f & ~((uintptr_t) 1)))[3] == NULL))
- #define Biddy_IsEqvPointer(f, g) (((uintptr_t) f & ~((uintptr_t) 1)) == ((uintptr_t) g & ~((uintptr_t) 1)))
- #define Biddy_GetMark(f) (((uintptr_t) f & (uintptr_t) 1) != 0)
- #define Biddy_SetMark(f) (f = (Biddy_Edge) ((uintptr_t) f | (uintptr_t) 1))
- #define Biddy_ClearMark(f) (f = (Biddy_Edge) ((uintptr_t) f & ~((uintptr_t) 1)))
- #define Biddy_InvertMark(f) (f = (Biddy_Edge) ((uintptr_t) f ^ (uintptr_t) 1))
- #define Biddy_Inv(f) ((Biddy_Edge) ((uintptr_t) f ^ (uintptr_t) 1))
- #define Biddy_InvCond(f, c) (c ? ((Biddy_Edge) ((uintptr_t) f ^ (uintptr_t) 1)) : f)
- #define Biddy_Regular(f) ((Biddy_Edge) ((uintptr_t) f & ~((uintptr_t) 1)))
- #define Biddy_Complement(f) ((Biddy_Edge) ((uintptr_t) f | (uintptr_t) 1))
- #define Biddy_GetTag(f) ((Biddy_Variable) ((uintptr_t) f >> 48))
- #define Biddy_SetTag(f, t) (f = (Biddy_Edge) (((uintptr_t) f & 0x0000ffffffffffff) | ((uintptr_t) t << 48)))
- #define Biddy_ClearTag(f) (f = (Biddy_Edge) ((uintptr_t) f & 0x0000ffffffffffff))
- #define Biddy_Init() Biddy_InitMNG(NULL,BIDDYTYPEOBDD)
- #define Biddy_Exit() Biddy_ExitMNG(NULL)

- #define Biddy_GetManagerType() Biddy_Managed_GetManagerType(NULL)
- #define Biddy_SetManagerParameters(gcr, gcrF, gcrX, rr, rrF, rrX, st, fst, cst, fcst) Biddy_Managed_Set↩ ManagerParameters(NULL,gcr,gcrF,gcrX,rr,rrF,rrX,st,fst,cst,fcst)
- #define Biddy_Managed_GetThen(MNG, f) Biddy_GetThen(f)
- #define Biddy_Managed_GetElse(MNG, f) Biddy_GetElse(f)
- #define Biddy_Managed_GetTopVariable(MNG, f) Biddy_GetTopVariable(f)
- #define Biddy_IsEqv(f1, MNG2, f2) Biddy_Managed_IsEqv(NULL,f1,MNG2,f2)
- #define Biddy_SelectNode(f) Biddy_Managed_SelectNode(NULL,f)
- #define Biddy_DeselectNode(f) Biddy_Managed_DeselectNode(NULL,f)
- #define Biddy_IsSelected(f) Biddy_Managed_IsSelected(NULL,f)
- #define Biddy_SelectFunction(f) Biddy_Managed_SelectFunction(NULL,f)
- #define Biddy_DeselectAll() Biddy_Managed_DeselectAll(NULL)
- #define Biddy_GetTerminal() Biddy_Managed_GetTerminal(NULL)
- #define Biddy_GetConstantZero() Biddy_Managed_GetConstantZero(NULL)
- #define Biddy_GetConstantOne() Biddy_Managed_GetConstantOne(NULL)
- #define Biddy_GetBaseSet() Biddy_Managed_GetBaseSet(NULL)
- #define Biddy_GetVariable(x) Biddy_Managed_GetVariable(NULL,x)
- #define Biddy_GetPrevVariable(v) Biddy_Managed_GetPrevVariable(NULL,v)
- #define Biddy_GetNextVariable(v) Biddy_Managed_GetNextVariable(NULL,v)
- #define Biddy_GetVariableEdge(v) Biddy_Managed_GetVariableEdge(NULL,v)
- #define Biddy_GetElementEdge(v) Biddy_Managed_GetElementEdge(NULL,v)
- #define Biddy_GetVariableName(v) Biddy_Managed_GetVariableName(NULL,v)
- #define Biddy_GetTopVariableEdge(f) Biddy_Managed_GetTopVariableEdge(NULL,f)
- #define Biddy_GetTopVariableName(f) Biddy_Managed_GetTopVariableName(NULL,f)
- #define Biddy_GetTopVariableChar(f) Biddy_Managed_GetTopVariableChar(NULL,f)
- #define Biddy_ResetVariablesValue() Biddy_Managed_ResetVariablesValue(NULL)
- #define Biddy_SetVariableValue(v, f) Biddy_Managed_SetVariableValue(NULL,v,f)
- #define Biddy_IsSmaller(fv, gv) Biddy_Managed_IsSmaller(NULL,fv,gv)
- #define Biddy_FoaVariable(x, varelem) Biddy_Managed_FoaVariable(NULL,x,varelem)
- #define Biddy_AddVariableByName(x) Biddy_Managed_AddVariableByName(NULL,x)
- #define Biddy_AddElementByName(x) Biddy_Managed_AddElementByName(NULL,x)
- #define Biddy_AddVariableBelow(v) Biddy_Managed_AddVariableBelow(NULL,v)
- #define Biddy_AddVariableAbove(v) Biddy_Managed_AddVariableAbove(NULL,v)
- #define Biddy_TransferMark(f, mark, leftright) Biddy_Managed_TransferMark(NULL,f,mark,leftright)
- #define Biddy_IncTag(f) Biddy_Managed_IncTag(NULL,f)
- #define Biddy_TaggedFoaNode(v, pf, pt, ptag, garbageAllowed) Biddy_Managed_TaggedFoaNode(NU↩ LL,v,pf,pt,ptag,garbageAllowed)
- #define Biddy_Not(f) Biddy_Managed_Not(NULL,f)
- #define Biddy_ITE(f, g, h) Biddy_Managed_ITE(NULL,f,g,h)
- #define Biddy_And(f, g) Biddy_Managed_And(NULL,f,g)
- #define Biddy_Or(f, g) Biddy_Managed_Or(NULL,f,g)
- #define Biddy_Nand(f, g) Biddy_Managed_Nand(NULL,f,g)
- #define Biddy_Nor(f, g) Biddy_Managed_Nor(NULL,f,g)
- #define Biddy_Xor(f, g) Biddy_Managed_Xor(NULL,f,g)
- #define Biddy_Xnor(f, g) Biddy_Managed_Xnor(NULL,f,g)
- #define Biddy_Leq(f, g) Biddy_Managed_Leq(NULL,f,g)
- #define Biddy_Gt(f, g) Biddy_Managed_Gt(NULL,f,g)
- #define Biddy_IsLeq(f, g) Biddy_Managed_IsLeq(NULL,f,g)
- #define Biddy_Restrict(f, v, value) Biddy_Managed_Restrict(NULL,f,v,value)
- #define Biddy_Compose(f, g, v) Biddy_Managed_Compose(NULL,f,g,v)
- #define Biddy_E(f, v) Biddy_Managed_E(NULL,f,v)
- #define Biddy_A(f, v) Biddy_Managed_A(NULL,f,v)
- #define Biddy_IsVariableDependent(f, v) Biddy_Managed_IsVariableDependent(NULL,f,v)
- #define Biddy_ExistAbstract(f, cube) Biddy_Managed_ExistAbstract(NULL,f,cube)
- #define Biddy_UnivAbstract(f, cube) Biddy_Managed_UnivAbstract(NULL,f,cube)

- #define Biddy_AndAbstract(f, g, cube) Biddy_Managed_AndAbstract(NULL,f,g,cube)
- #define Biddy_Constrain(f, c) Biddy_Managed_Constrain(NULL,f,c)
- #define Biddy_Simplify(f, c) Biddy_Managed_Simplify(NULL,f,c)
- #define Biddy_Support(f) Biddy_Managed_Support(NULL,f)
- #define Biddy_Replace(f) Biddy_Managed_Replace(NULL,f)
- #define Biddy_Change(f, v) Biddy_Managed_Change(NULL,f,v)
- #define Biddy_Subset(f, v, value) Biddy_Managed_Subset(NULL,f,v,value)
- #define Biddy_IsOK(f) Biddy_Managed_IsOK(NULL,f)
- #define Biddy_GC(target, purge, total) Biddy_Managed_GC(NULL,target,purge,total)
- #define Biddy_Clean() Biddy_Managed_Clean(NULL)
- #define Biddy_Purge() Biddy_Managed_Purge(NULL)
- #define Biddy_PurgeAndReorder(f, c) Biddy_Managed_PurgeAndReorder(NULL,f,c)
- #define Biddy_Refresh(f) Biddy_Managed_Refresh(NULL,f)
- #define Biddy_AddCache(gc) Biddy_Managed_AddCache(NULL,gc)
- #define Biddy_AddFormula(x, f, c) Biddy_Managed_AddFormula(NULL,x,f,c)
- #define Biddy_FindFormula(x, f) Biddy_Managed_FindFormula(NULL,x,f)
- #define Biddy_DeleteFormula(x) Biddy_Managed_DeleteFormula(NULL,x)
- #define Biddy_DeleteIthFormula(x) Biddy_Managed_DeleteIthFormula(NULL,x)
- #define Biddy_GetIthFormula(i) Biddy_Managed_GetIthFormula(NULL,i)
- #define Biddy_GetIthFormulaName(i) Biddy_Managed_GetIthFormulaName(NULL,i)
- #define Biddy_SwapWithHigher(v) Biddy_Managed_SwapWithHigher(NULL,v)
- #define Biddy_SwapWithLower(v) Biddy_Managed_SwapWithLower(NULL,v)
- #define Biddy_Sifting(f, c) Biddy_Managed_Sifting(NULL,f,c)
- #define Biddy_Copy(MNG2, f) Biddy_Managed_Copy(NULL,MNG2,f)
- #define Biddy_CopyFormula(MNG2, x) Biddy_Managed_CopyFormula(NULL,MNG2,x)
- #define Biddy_Eval(f) Biddy_Managed_Eval(NULL,f)
- #define Biddy_Random(support, r) Biddy_Managed_Random(NULL,support,r)
- #define Biddy_RandomSet(unit, r) Biddy_Managed_RandomSet(NULL,unit,r)
- #define Biddy_NodeNumber(f) Biddy_Managed_NodeNumber(NULL,f)
- #define Biddy_Managed_NodeMaxLevel(MNG, f) Biddy_NodeMaxLevel(f)
- #define Biddy_Managed_NodeAvgLevel(MNG, f) Biddy_NodeAvgLevel(f)
- #define Biddy_VariableTableNum() Biddy_Managed_VariableTableNum(NULL)
- #define Biddy_NodeTableSize() Biddy_Managed_NodeTableSize(NULL)
- #define Biddy_NodeTableBlockNumber() Biddy_Managed_NodeTableBlockNumber(NULL)
- #define Biddy_NodeTableGenerated() Biddy_Managed_NodeTableGenerated(NULL)
- #define Biddy_NodeTableMax() Biddy_Managed_NodeTableMax(NULL)
- #define Biddy_NodeTableNum() Biddy_Managed_NodeTableNum(NULL)
- #define Biddy_NodeTableNumVar(v) Biddy_Managed_NodeTableNumVar(NULL,v)
- #define Biddy_NodeTableGCNumber() Biddy_Managed_NodeTableGCNumber(NULL)
- #define Biddy_NodeTableSwapNumber() Biddy_Managed_NodeTableSwapNumber(NULL)
- #define Biddy_NodeTableSiftingNumber() Biddy_Managed_NodeTableSiftingNumber(NULL)
- #define Biddy_NodeTableResizeNumber() Biddy_Managed_NodeTableResizeNumber(NULL)
- #define Biddy_NodeTableITENumber() Biddy_Managed_NodeTableITENumber(NULL)
- #define Biddy_NodeTableITERecursiveNumber() Biddy_Managed_NodeTableITERecursiveNumber(NULL)
- #define Biddy_NodeTableANDORNumber() Biddy_Managed_NodeTableANDORNumber(NULL)
- #define Biddy_NodeTableANDORRecursiveNumber() Biddy_Managed_NodeTableANDORRecursive↩ Number(NULL)
- #define Biddy_NodeTableXORNumber() Biddy_Managed_NodeTableXORNumber(NULL)
- #define Biddy_NodeTableXORRecursiveNumber() Biddy_Managed_NodeTableXORRecursiveNumber(NU↩ LL)
- #define Biddy_NodeTableGCTime() Biddy_Managed_NodeTableGCTime(NULL)
- #define Biddy_NodeTableGCObsoleteNumber() Biddy_Managed_NodeTableGCObsoleteNumber(NULL)
- #define Biddy_NodeTableDRTime() Biddy_Managed_NodeTableDRTime(NULL)
- #define Biddy_FormulaTableNum() Biddy_Managed_FormulaTableNum(NULL)
- #define Biddy_ListUsed() Biddy_Managed_ListUsed(NULL)

- #define Biddy_ListMaxLength() Biddy_Managed_ListMaxLength(NULL)
- #define Biddy_ListAvgLength() Biddy_Managed_ListAvgLength(NULL)
- #define Biddy_OPCacheSearch() Biddy_Managed_OPCacheSearch(NULL)
- #define Biddy_OPCacheFind() Biddy_Managed_OPCacheFind(NULL)
- #define Biddy_OPCacheOverwrite() Biddy_Managed_OPCacheOverwrite(NULL)
- #define Biddy_NodeNumberPlain(f) Biddy_Managed_NodeNumberPlain(NULL,f)
- #define Biddy_DependentVariableNumber(f) Biddy_Managed_DependentVariableNumber(NULL,f)
- #define Biddy_NodeVarNumber(f, n, v) Biddy_Managed_NodeVarNumber(NULL,f,n,v)
- #define Biddy_CountPaths(f) Biddy_Managed_CountPaths(NULL,f)
- #define Biddy_CountMinterm(f, nvars) Biddy_Managed_CountMinterm(NULL,f,nvars)
- #define Biddy_DensityFunction(f, nvars) Biddy_Managed_DensityFunction(NULL,f,nvars)
- #define Biddy_DensityBDD(f, nvars) Biddy_Managed_DensityBDD(NULL,f,nvars)
- #define Biddy_ReadMemoryInUse() Biddy_Managed_ReadMemoryInUse(NULL)
- #define Biddy_PrintInfo(f) Biddy_Managed_PrintInfo(NULL,f)
- #define Biddy_Eval0(s) Biddy_Managed_Eval0(NULL,s)
- #define Biddy_Eval1x(s, lf) Biddy_Managed_Eval1x(NULL,s,lf)
- #define Biddy_Eval2(boolFunc) Biddy_Managed_Eval2(NULL,boolFunc)
- #define Biddy_ReadVerilogFile(filename, prefix) Biddy_Managed_ReadVerilogFile(NULL,filename,prefix)
- #define Biddy_PrintfBDD(f) Biddy_Managed_PrintfBDD(NULL,f)
- #define Biddy_WriteBDD(filename, f, label) Biddy_Managed_WriteBDD(NULL,filename,f,label)
- #define Biddy_PrintfTable(f) Biddy_Managed_PrintfTable(NULL,f)
- #define Biddy_WriteTable(filename, f) Biddy_Managed_WriteTable(NULL,filename,f)
- #define Biddy_PrintfSOP(f) Biddy_Managed_PrintfSOP(NULL,f)
- #define Biddy_WriteSOP(filename, f) Biddy_Managed_WriteSOP(NULL,filename,f)
- #define Biddy_WriteDot(filename, f, label, id, cudd) Biddy_Managed_WriteDot(NULL,filename,f,label,id,cudd);
- #define Biddy_WriteBddview(filename, f, label, table) Biddy_Managed_WriteBddview(NULL,filename,f,label,table);

Typedefs

- typedef char Biddy_Boolean
- typedef char * Biddy_String
- typedef void ** Biddy_Manager
- typedef void * Biddy_Cache
- typedef unsigned short int Biddy_Variable
- typedef void * Biddy_Edge
- typedef void(* Biddy_GCFunction) (Biddy_Manager)
- typedef Biddy_Boolean(* Biddy_LookupFunction) (Biddy_String, Biddy_Edge *)

5.1.1 Detailed Description

File biddy.h contains declaration of all external data structures.

Description

```
PackageName [Biddy]
Synopsis    [Biddy provides data structures and algorithms for the
             representation and manipulation of Boolean functions with
             ROBDDs. A hash table is used for quick search of nodes.
             Complement edges decreases the number of nodes. An automatic
             garbage collection with a system age is implemented.
             Variable swapping and sifting are implemented.]

FileName    [biddy.h]
Revision    [$Revision: 252 $]
Date        [$Date: 2017-03-17 23:30:03 +0100 (pet, 17 mar 2017) $]
Authors     [Robert Meolic (robert.meolic@um.si),
             Ales Casar (ales@homemade.net)]
```

Copyright

More info

See also: biddyInt.h

5.1.2 Macro Definition Documentation

5.1.2.1 #define Biddy_IsNull(f) (f == NULL)

Biddy_IsNull returns TRUE iff given BDD is a null edge.

Definition at line 121 of file biddy.h.

5.1.2.2 #define Biddy_IsConstant(f) ((((void**)((uintptr_t) f & ~((uintptr_t) 1)))[2] == NULL) && (((void**)((uintptr_t) f & ~((uintptr_t) 1)))[3] == NULL))

Biddy_IsConstant returns TRUE iff given edge points to the terminal node.

Definition at line 130 of file biddy.h.

5.1.2.3 #define Biddy_IsTaggedConstant(f) (((uintptr_t) f >> 48 == 0) && (((void**)((uintptr_t) f & ~((uintptr_t) 1)))[2] == NULL) && (((void**)((uintptr_t) f & ~((uintptr_t) 1)))[3] == NULL))

Biddy_IsTaggedConstant returns TRUE iff given edge points to the terminal node and it represents constant 0 or 1, since Biddy v1.7.

Definition at line 139 of file biddy.h.

5.1.2.4 #define Biddy_IsEqvPointer(f, g) (((uintptr_t) f & ~((uintptr_t) 1)) == ((uintptr_t) g & ~((uintptr_t) 1)))

Biddy_IsEqvPointer returns TRUE iff given edges points to the same node.

Definition at line 144 of file biddy.h.

5.1.2.5 #define Biddy_GetMark(f) (((uintptr_t) f & (uintptr_t) 1) != 0)

Biddy_GetMark returns TRUE iff given edge is complemented.

Definition at line 147 of file biddy.h.

5.1.2.6 #define Biddy_SetMark(f) (f = (Biddy_Edge) ((uintptr_t) f | (uintptr_t) 1))

Biddy_SetMark makes given edge complemented.

Definition at line 150 of file biddy.h.

5.1.2.7 #define Biddy_ClearMark(f) (f = (Biddy_Edge) ((uintptr_t) f & ~((uintptr_t) 1)))

Biddy_ClearMark makes given edge not-complemented.

Definition at line 153 of file biddy.h.

5.1.2.8 #define Biddy_InvertMark(f) (f = (Biddy_Edge) ((uintptr_t) f ^ (uintptr_t) 1))

Biddy_InvertMark changes complement bit of the given edge.

Definition at line 156 of file biddy.h.

5.1.2.9 #define Biddy_Inv(f) ((Biddy_Edge) ((uintptr_t) f ^ (uintptr_t) 1))

Biddy_Inv returns edge with changed complement bit.

Definition at line 160 of file biddy.h.

5.1.2.10 #define Biddy_InvCond(f, c) (c ? ((Biddy_Edge) ((uintptr_t) f ^ (uintptr_t) 1)) : f)

Biddy_InvCond returns edge with conditinonaly changed complement bit.

Definition at line 163 of file biddy.h.

5.1.2.11 #define Biddy_Regular(f) ((Biddy_Edge) ((uintptr_t) f & ~((uintptr_t) 1)))

Biddy_Regular returns not-complemented version of edge, since Biddy v1.4.

Definition at line 166 of file biddy.h.

5.1.2.12 #define Biddy_Complement(f) ((Biddy_Edge) ((uintptr_t) f | (uintptr_t) 1))

Biddy_Complement returns complemented version of edge, since Biddy v1.4.

Definition at line 169 of file biddy.h.

5.1.2.13 #define Biddy_GetTag(f) ((Biddy_Variable) ((uintptr_t) f >> 48))

Biddy_GetTag returns tag used for the given edge, since Biddy v1.7.

Definition at line 174 of file biddy.h.

5.1.2.14 #define Biddy_SetTag(f, t) (f = (Biddy_Edge) (((uintptr_t) f & 0x0000ffffffffffff) | ((uintptr_t) t << 48)))

Biddy_SetTag adds tag to the given edge, since Biddy v1.7.

Definition at line 180 of file biddy.h.

5.1.2.15 #define Biddy_ClearTag(f) (f = (Biddy_Edge) ((uintptr_t) f & 0x0000ffffffffffff))

Biddy_ClearTag removes tag from the given edge, since Biddy v1.7.

Definition at line 184 of file biddy.h.

5.1.2.16 #define Biddy_Init( ) Biddy_InitMNG(NULL,BIDDYTYPEOBDD)

Macros Biddy_Init and Biddy_InitAnonymous will initialize anonymous manager.

Definition at line 266 of file biddy.h.

5.1.2.17 #define Biddy_Exit( ) Biddy_ExitMNG(NULL)

Macro Biddy_Exit will delete anonymous manager.

Definition at line 272 of file biddy.h.

5.1.2.18 #define Biddy_GetManagerType( ) Biddy_Managed_GetManagerType(NULL)

Macro Biddy_GetManagerType is defined for use with anonymous manager.

Definition at line 280 of file biddy.h.

5.1.2.19 #define Biddy_SetManagerParameters(gcr, gcrF, gcrX, rr, rrF, rrX, st, fst, cst, fcst) Biddy_Managed_SetManagerParameters(NULL,gcr,gcrF,gcrX,rr,rrF,rrX,st,fst,cst,fcst)

Macro Biddy_SetManagerParameters is defined for use with anonymous manager.

Definition at line 285 of file biddy.h.

5.1.2.20 #define Biddy_Managed_GetThen(*MNG, f*) Biddy_GetThen(f)

Macro Biddy_Managed_GetThen is defined for your convenience.

Definition at line 290 of file biddy.h.

5.1.2.21 #define Biddy_Managed_GetElse(*MNG, f*) Biddy_GetElse(f)

Macro Biddy_Managed_GetElse is defined for your convenience.

Definition at line 295 of file biddy.h.

5.1.2.22 #define Biddy_Managed_GetTopVariable(*MNG, f*) Biddy_GetTopVariable(f)

Macro Biddy_Managed_GetTopVariable is defined for your convenience.

Definition at line 300 of file biddy.h.

5.1.2.23 #define Biddy_IsEqv(*f1, MNG2, f2*) Biddy_Managed_IsEqv(NULL,f1,MNG2,f2)

Macro Biddy_IsEqv is defined for use with anonymous manager.

Definition at line 305 of file biddy.h.

5.1.2.24 #define Biddy_SelectNode(*f*) Biddy_Managed_SelectNode(NULL,f)

Macro Biddy_SelectNode is defined for use with anonymous manager.

Definition at line 310 of file biddy.h.

5.1.2.25 #define Biddy_DeselectNode(*f*) Biddy_Managed_DeselectNode(NULL,f)

Macro Biddy_DeselectNode is defined for use with anonymous manager.

Definition at line 315 of file biddy.h.

5.1.2.26 #define Biddy_IsSelected(*f*) Biddy_Managed_IsSelected(NULL,f)

Macro Biddy_IsSelected is defined for use with anonymous manager.

Definition at line 320 of file biddy.h.

5.1.2.27 #define Biddy_SelectFunction(*f*) Biddy_Managed_SelectFunction(NULL,f)

Macro Biddy_SelectFunction is defined for use with anonymous manager.

Definition at line 325 of file biddy.h.

5.1.2.28 #define Biddy_DeselectAll( ) Biddy_Managed_DeselectAll(NULL)

Macro Biddy_DeselectAll is defined for use with anonymous manager.

Definition at line 330 of file biddy.h.

5.1.2.29 #define Biddy_GetTerminal( ) Biddy_Managed_GetTerminal(NULL)

Macro Biddy_GetTerminal is defined for use with anonymous manager.

Definition at line 335 of file biddy.h.

5.1.2.30 #define Biddy_GetConstantZero( ) Biddy_Managed_GetConstantZero(NULL)

Macro Biddy_GetConstantZero is defined for use with anonymous manager.

Definition at line 340 of file biddy.h.

5.1.2.31 #define Biddy_GetConstantOne( ) Biddy_Managed_GetConstantOne(NULL)

Macro Biddy_GetConstantOne is defined for use with anonymous manager.

Definition at line 347 of file biddy.h.

5.1.2.32 #define Biddy_GetBaseSet( ) Biddy_Managed_GetBaseSet(NULL)

Macro Biddy_GetBaseSet is defined for use with anonymous manager.

Definition at line 354 of file biddy.h.

5.1.2.33 #define Biddy_GetVariable(*x*) Biddy_Managed_GetVariable(NULL,x)

Macro Biddy_GetVariable is defined for use with anonymous manager.

Definition at line 359 of file biddy.h.

5.1.2.34 #define Biddy_GetPrevVariable(*v*) Biddy_Managed_GetPrevVariable(NULL,v)

Macro Biddy_GetPrevVariable is defined for use with anonymous manager.

Definition at line 364 of file biddy.h.

5.1.2.35 #define Biddy_GetNextVariable(*v*) Biddy_Managed_GetNextVariable(NULL,v)

Macro Biddy_GetNextVariable is defined for use with anonymous manager.

Definition at line 369 of file biddy.h.

5.1.2.36 #define Biddy_GetVariableEdge(v) Biddy_Managed_GetVariableEdge(NULL,v)

Macro Biddy_GetVariableEdge is defined for use with anonymous manager.

Definition at line 374 of file biddy.h.

5.1.2.37 #define Biddy_GetElementEdge(v) Biddy_Managed_GetElementEdge(NULL,v)

Macro Biddy_GetElementEdge is defined for use with anonymous manager.

Definition at line 379 of file biddy.h.

5.1.2.38 #define Biddy_GetVariableName(v) Biddy_Managed_GetVariableName(NULL,v)

Macro Biddy_GetVariableName is defined for use with anonymous manager.

Definition at line 384 of file biddy.h.

5.1.2.39 #define Biddy_GetTopVariableEdge(f) Biddy_Managed_GetTopVariableEdge(NULL,f)

Macro Biddy_GetTopVariableEdge is defined for use with anonymous manager.

Definition at line 389 of file biddy.h.

5.1.2.40 #define Biddy_GetTopVariableName(f) Biddy_Managed_GetTopVariableName(NULL,f)

Macro Biddy_GetTopVariableName is defined for use with anonymous manager.

Definition at line 394 of file biddy.h.

5.1.2.41 #define Biddy_GetTopVariableChar(f) Biddy_Managed_GetTopVariableChar(NULL,f)

Macro Biddy_GetTopVariableChar is defined for use with anonymous manager.

Definition at line 399 of file biddy.h.

5.1.2.42 #define Biddy_ResetVariablesValue( ) Biddy_Managed_ResetVariablesValue(NULL)

Macro Biddy_ResetVariablesValue is defined for use with anonymous manager.

Definition at line 404 of file biddy.h.

5.1.2.43 #define Biddy_SetVariableValue(v, f) Biddy_Managed_SetVariableValue(NULL,v,f)

Macro Biddy_SetVariableValue is defined for use with anonymous manager.

Definition at line 409 of file biddy.h.

5.1.2.44 #define Biddy_IsSmaller(*fv, gv*) Biddy_Managed_IsSmaller(NULL,fv,gv)

Macro Biddy_IsSmaller is defined for use with anonymous manager.

Definition at line 414 of file biddy.h.

5.1.2.45 #define Biddy_FoaVariable(*x, varelem*) Biddy_Managed_FoaVariable(NULL,x,varelem)

Macro Biddy_FoaVariable is defined for use with anonymous manager.

Definition at line 419 of file biddy.h.

5.1.2.46 #define Biddy_AddVariableByName(*x*) Biddy_Managed_AddVariableByName(NULL,x)

Macro Biddy_AddVariableByName is defined for use with anonymous manager.

Definition at line 424 of file biddy.h.

5.1.2.47 #define Biddy_AddElementByName(*x*) Biddy_Managed_AddElementByName(NULL,x)

Macro Biddy_AddElementByName is defined for use with anonymous manager.

Definition at line 431 of file biddy.h.

5.1.2.48 #define Biddy_AddVariableBelow(*v*) Biddy_Managed_AddVariableBelow(NULL,v)

Macro Biddy_AddVariableBelow is defined for use with anonymous manager.

Definition at line 438 of file biddy.h.

5.1.2.49 #define Biddy_AddVariableAbove(*v*) Biddy_Managed_AddVariableAbove(NULL,v)

Macro Biddy_AddVariableAbove is defined for use with anonymous manager.

Definition at line 443 of file biddy.h.

5.1.2.50 #define Biddy_TransferMark(*f, mark, leftright*) Biddy_Managed_TransferMark(NULL,f,mark,leftright)

Macro Biddy_TransferMark is defined for use with anonymous manager.

For OBDD, use macro Biddy_InvCond.

Definition at line 449 of file biddy.h.

5.1.2.51 #define Biddy_IncTag(*f*) Biddy_Managed_IncTag(NULL,f)

Macro Biddy_IncTag is defined for use with anonymous manager.

Definition at line 454 of file biddy.h.

5.1.2.52 #define Biddy_TaggedFoaNode(*v, pf, pt, ptag, garbageAllowed*) Biddy_Managed_TaggedFoaNode(NULL,v,pf,pt,ptag,garbageAllowed)

Macro Biddy_TaggedFoaNode is defined for use with anonymous manager.

Definition at line 459 of file biddy.h.

5.1.2.53 #define Biddy_Not(*f*) Biddy_Managed_Not(NULL,f)

Macro Biddy_Not is defined for use with anonymous manager.

For OBDD and OFDD, use macro Biddy_Inv.

Definition at line 467 of file biddy.h.

5.1.2.54 #define Biddy_ITE(*f, g, h*) Biddy_Managed_ITE(NULL,f,g,h)

Macro Biddy_ITE is defined for use with anonymous manager.

Definition at line 472 of file biddy.h.

5.1.2.55 #define Biddy_And(*f, g*) Biddy_Managed_And(NULL,f,g)

Macro Biddy_And is defined for use with anonymous manager.

Macros Biddy_Managed_Intersect and Biddy_Intersect are defined for set manipulation.

Definition at line 478 of file biddy.h.

5.1.2.56 #define Biddy_Or(*f, g*) Biddy_Managed_Or(NULL,f,g)

Macro Biddy_Or is defined for use with anonymous manager.

Macros Biddy_Managed_Union and Biddy_Union are defined for set manipulation.

Definition at line 486 of file biddy.h.

5.1.2.57 #define Biddy_Nand(*f, g*) Biddy_Managed_Nand(NULL,f,g)

Macro Biddy_Nand is defined for use with anonymous manager.

Definition at line 493 of file biddy.h.

5.1.2.58 #define Biddy_Nor(f, g) Biddy_Managed_Nor(NULL,f,g)

Macro Biddy_Nor is defined for use with anonymous manager.

Definition at line 498 of file biddy.h.

5.1.2.59 #define Biddy_Xor(f, g) Biddy_Managed_Xor(NULL,f,g)

Macro Biddy_Xor is defined for use with anonymous manager.

Definition at line 503 of file biddy.h.

5.1.2.60 #define Biddy_Xnor(f, g) Biddy_Managed_Xnor(NULL,f,g)

Macro Biddy_Xnor is defined for use with anonymous manager.

Definition at line 508 of file biddy.h.

5.1.2.61 #define Biddy_Leq(f, g) Biddy_Managed_Leq(NULL,f,g)

Macro Biddy_Leq is defined for use with anonymous manager.

Definition at line 513 of file biddy.h.

5.1.2.62 #define Biddy_Gt(f, g) Biddy_Managed_Gt(NULL,f,g)

Macro Biddy_Gt is defined for use with anonymous manager.

Definition at line 518 of file biddy.h.

5.1.2.63 #define Biddy_IsLeq(f, g) Biddy_Managed_IsLeq(NULL,f,g)

Macro Biddy_IsLeq is defined for use with anonymous manager.

Definition at line 525 of file biddy.h.

5.1.2.64 #define Biddy_Restrict(f, v, value) Biddy_Managed_Restrict(NULL,f,v,value)

Macro Biddy_Restrict is defined for use with anonymous manager.

Definition at line 531 of file biddy.h.

5.1.2.65 #define Biddy_Compose(f, g, v) Biddy_Managed_Compose(NULL,f,g,v)

Macro Biddy_Compose is defined for use with anonymous manager.

Definition at line 536 of file biddy.h.

5.1.2.66 #define Biddy_E(f, v) Biddy_Managed_E(NULL,f,v)

Macro Biddy_E is defined for use with anonymous manager.

Definition at line 541 of file biddy.h.

5.1.2.67 #define Biddy_A(f, v) Biddy_Managed_A(NULL,f,v)

Macro Biddy_A is defined for use with anonymous manager.

Definition at line 546 of file biddy.h.

5.1.2.68 #define Biddy_IsVariableDependent(f, v) Biddy_Managed_IsVariableDependent(NULL,f,v)

Macro Biddy_IsVariableDependent is defined for use with anonymous manager.

Definition at line 551 of file biddy.h.

5.1.2.69 #define Biddy_ExistAbstract(f, cube) Biddy_Managed_ExistAbstract(NULL,f,cube)

Macro Biddy_ExistAbstract is defined for use with anonymous manager.

Definition at line 556 of file biddy.h.

5.1.2.70 #define Biddy_UnivAbstract(f, cube) Biddy_Managed_UnivAbstract(NULL,f,cube)

Macro Biddy_UnivAbstract is defined for use with anonymous manager.

Definition at line 561 of file biddy.h.

5.1.2.71 #define Biddy_AndAbstract(f, g, cube) Biddy_Managed_AndAbstract(NULL,f,g,cube)

Macro Biddy_AndAbstract is defined for use with anonymous manager.

Definition at line 566 of file biddy.h.

5.1.2.72 #define Biddy_Constrain(f, c) Biddy_Managed_Constrain(NULL,f,c)

Macro Biddy_Constrain is defined for use with anonymous manager.

Definition at line 571 of file biddy.h.

5.1.2.73 #define Biddy_Simplify(f, c) Biddy_Managed_Simplify(NULL,f,c)

Macro Biddy_Simplify is defined for use with anonymous manager.

Definition at line 577 of file biddy.h.

5.1.2.74 #define Biddy_Support(*f*) Biddy_Managed_Support(NULL,f)

Macro Biddy_Support is defined for use with anonymous manager.

Definition at line 582 of file biddy.h.

5.1.2.75 #define Biddy_Replace(*f*) Biddy_Managed_Replace(NULL,f)

Macro Biddy_Replace is defined for use with anonymous manager.

Definition at line 587 of file biddy.h.

5.1.2.76 #define Biddy_Change(*f, v*) Biddy_Managed_Change(NULL,f,v)

Macro Biddy_Change is defined for use with anonymous manager.

Definition at line 592 of file biddy.h.

5.1.2.77 #define Biddy_Subset(*f, v, value*) Biddy_Managed_Subset(NULL,f,v,value)

Macro Biddy_Subset is defined for use with anonymous manager.

Definition at line 598 of file biddy.h.

5.1.2.78 #define Biddy_IsOK(*f*) Biddy_Managed_IsOK(NULL,f)

Macro Biddy_IsOK is defined for use with anonymous manager.

Definition at line 607 of file biddy.h.

5.1.2.79 #define Biddy_GC(*target, purge, total*) Biddy_Managed_GC(NULL,target,purge,total)

Macro Biddy_GC is defined for use with anonymous manager.

Macros Biddy_Managed_FullGC and Biddy_FullGC are useful variants.

Definition at line 613 of file biddy.h.

5.1.2.80 #define Biddy_Clean( ) Biddy_Managed_Clean(NULL)

Macro Biddy_Clean is defined for use with anonymous manager.

Definition at line 620 of file biddy.h.

5.1.2.81 #define Biddy_Purge( ) Biddy_Managed_Purge(NULL)

Macro Biddy_Purge is defined for use with anonymous manager.

Definition at line 625 of file biddy.h.

5.1.2.82 #define Biddy_PurgeAndReorder(*f, c*) Biddy_Managed_PurgeAndReorder(NULL,f,c)

Macro Biddy_PurgeAndReorder is defined for use with anonymous manager.

Definition at line 630 of file biddy.h.

5.1.2.83 #define Biddy_Refresh(*f*) Biddy_Managed_Refresh(NULL,f)

Macro Biddy_Refresh is defined for use with anonymous manager.

Definition at line 635 of file biddy.h.

5.1.2.84 #define Biddy_AddCache(*gc*) Biddy_Managed_AddCache(NULL,gc)

Macro Biddy_AddCache is defined for use with anonymous manager.

Definition at line 640 of file biddy.h.

5.1.2.85 #define Biddy_AddFormula(*x, f, c*) Biddy_Managed_AddFormula(NULL,x,f,c)

Macro Biddy_AddFormula is defined for use with anonymous manager.

Definition at line 645 of file biddy.h.

5.1.2.86 #define Biddy_FindFormula(*x, f*) Biddy_Managed_FindFormula(NULL,x,f)

Macro Biddy_FindFormula is defined for use with anonymous manager.

Definition at line 654 of file biddy.h.

5.1.2.87 #define Biddy_DeleteFormula(*x*) Biddy_Managed_DeleteFormula(NULL,x)

Macro Biddy_DeleteFormula is defined for use with anonymous manager.

Definition at line 659 of file biddy.h.

5.1.2.88 #define Biddy_DeleteIthFormula(*x*) Biddy_Managed_DeleteIthFormula(NULL,x)

Macro Biddy_DeleteIthFormula is defined for use with anonymous manager.

Definition at line 664 of file biddy.h.

5.1.2.89 #define Biddy_GetIthFormula(*i*) Biddy_Managed_GetIthFormula(NULL,i)

Macro Biddy_GetIthFormula is defined for use with anonymous manager.

Definition at line 669 of file biddy.h.

5.1.2.90 #define Biddy_GetIthFormulaName(*i*) Biddy_Managed_GetIthFormulaName(NULL,i)

Macro Biddy_GetIthFormulaName is defined for use with anonymous manager.

Definition at line 674 of file biddy.h.

5.1.2.91 #define Biddy_SwapWithHigher(*v*) Biddy_Managed_SwapWithHigher(NULL,v)

Macro Biddy_SwapWithHigher is defined for use with anonymous manager.

Definition at line 679 of file biddy.h.

5.1.2.92 #define Biddy_SwapWithLower(*v*) Biddy_Managed_SwapWithLower(NULL,v)

Macro Biddy_SwapWithLower is defined for use with anonymous manager.

Definition at line 684 of file biddy.h.

5.1.2.93 #define Biddy_Sifting(*f, c*) Biddy_Managed_Sifting(NULL,f,c)

Macro Biddy_Sifting is defined for use with anonymous manager.

Definition at line 689 of file biddy.h.

5.1.2.94 #define Biddy_Copy(*MNG2, f*) Biddy_Managed_Copy(NULL,MNG2,f)

Macro Biddy_Copy is defined for use with anonymous manager.

Definition at line 694 of file biddy.h.

5.1.2.95 #define Biddy_CopyFormula(*MNG2, x*) Biddy_Managed_CopyFormula(NULL,MNG2,x)

Macro Biddy_CopyFormula is defined for use with anonymous manager.

Definition at line 699 of file biddy.h.

5.1.2.96 #define Biddy_Eval(*f*) Biddy_Managed_Eval(NULL,f)

Macro Biddy_Eval is defined for use with anonymous manager.

Definition at line 704 of file biddy.h.

5.1.2.97 #define Biddy_Random(*support, r*) Biddy_Managed_Random(NULL,support,r)

Macro Biddy_Random is defined for use with anonymous manager.

Definition at line 709 of file biddy.h.

5.1.2.98 #define Biddy_RandomSet(*unit, r*) Biddy_Managed_RandomSet(NULL,unit,r)

Macro Biddy_RandomSet is defined for use with anonymous manager.

Definition at line 714 of file biddy.h.

5.1.2.99 #define Biddy_NodeNumber(*f*) Biddy_Managed_NodeNumber(NULL,f)

Macro Biddy_NodeNumber(f) is defined for use with anonymous manager.

Definition at line 735 of file biddy.h.

5.1.2.100 #define Biddy_Managed_NodeMaxLevel(*MNG, f*) Biddy_NodeMaxLevel(f)

Macro Biddy_Managed_NodeMaxLevel(MNG,f) is defined for your convenience.

Definition at line 740 of file biddy.h.

5.1.2.101 #define Biddy_Managed_NodeAvgLevel(*MNG, f*) Biddy_NodeAvgLevel(f)

Macro Biddy_Managed_NodeAvgLevel(MNG,f) is defined for your convenience.

Definition at line 745 of file biddy.h.

5.1.2.102 #define Biddy_VariableTableNum( ) Biddy_Managed_VariableTableNum(NULL)

Macro Biddy_VariableTableNum is defined for use with anonymous manager.

Definition at line 750 of file biddy.h.

5.1.2.103 #define Biddy_NodeTableSize( ) Biddy_Managed_NodeTableSize(NULL)

Macro Biddy_NodeTableSize is defined for use with anonymous manager.

Definition at line 755 of file biddy.h.

5.1.2.104 #define Biddy_NodeTableBlockNumber( ) Biddy_Managed_NodeTableBlockNumber(NULL)

Macro Biddy_NodeTableBlockNumber is defined for use with anonymous manager.

Definition at line 760 of file biddy.h.

5.1.2.105 #define Biddy_NodeTableGenerated( ) Biddy_Managed_NodeTableGenerated(NULL)

Macro Biddy_NodeTableGenerated is defined for use with anonymous manager.

Definition at line 765 of file biddy.h.

5.1.2.106 #define Biddy_NodeTableMax( ) Biddy_Managed_NodeTableMax(NULL)

Macro Biddy_NodeTableMax is defined for use with anonymous manager.

Definition at line 770 of file biddy.h.

5.1.2.107 #define Biddy_NodeTableNum( ) Biddy_Managed_NodeTableNum(NULL)

Macro Biddy_NodeTableNum is defined for use with anonymous manager.

Definition at line 775 of file biddy.h.

5.1.2.108 #define Biddy_NodeTableNumVar(*v*) Biddy_Managed_NodeTableNumVar(NULL,v)

Macro Biddy_NodeTableNumVar is defined for use with anonymous manager.

Definition at line 780 of file biddy.h.

5.1.2.109 #define Biddy_NodeTableGCNumber( ) Biddy_Managed_NodeTableGCNumber(NULL)

Macro Biddy_NodeTableGCNumber is defined for use with anonymous manager.

Definition at line 785 of file biddy.h.

5.1.2.110 #define Biddy_NodeTableSwapNumber( ) Biddy_Managed_NodeTableSwapNumber(NULL)

Macro Biddy_NodeTableSwapNumber is defined for use with anonymous manager.

Definition at line 790 of file biddy.h.

5.1.2.111 #define Biddy_NodeTableSiftingNumber( ) Biddy_Managed_NodeTableSiftingNumber(NULL)

Macro Biddy_NodeTableSiftingNumber is defined for use with anonymous manager.

Definition at line 795 of file biddy.h.

5.1.2.112 #define Biddy_NodeTableResizeNumber( ) Biddy_Managed_NodeTableResizeNumber(NULL)

Macro Biddy_NodeTableResizeNumber is defined for use with anonymous manager.

Definition at line 800 of file biddy.h.

5.1.2.113 #define Biddy_NodeTableITENumber( ) Biddy_Managed_NodeTableITENumber(NULL)

Macro Biddy_NodeTableITENumber is defined for use with anonymous manager.

Definition at line 805 of file biddy.h.

5.1.2.114 #define Biddy_NodeTableITERecursiveNumber( ) Biddy_Managed_NodeTableITERecursiveNumber(NULL)

Macro Biddy_NodeTableITERecursiveNumber is defined for use with anonymous manager.

Definition at line 810 of file biddy.h.

5.1.2.115 #define Biddy_NodeTableANDORNumber( ) Biddy_Managed_NodeTableANDORNumber(NULL)

Macro Biddy_NodeTableANDORNumber is defined for use with anonymous manager.

Definition at line 815 of file biddy.h.

5.1.2.116 #define Biddy_NodeTableANDORRecursiveNumber( ) Biddy_Managed_NodeTableANDORRecursive↩ Number(NULL)

Macro Biddy_NodeTableANDORRecursiveNumber is defined for use with anonymous manager.

Definition at line 820 of file biddy.h.

5.1.2.117 #define Biddy_NodeTableXORNumber( ) Biddy_Managed_NodeTableXORNumber(NULL)

Macro Biddy_NodeTableXORNumber is defined for use with anonymous manager.

Definition at line 825 of file biddy.h.

5.1.2.118 #define Biddy_NodeTableXORRecursiveNumber( ) Biddy_Managed_NodeTableXORRecursiveNumber(N↩ ULL)

Macro Biddy_NodeTableXORRecursiveNumber is defined for use with anonymous manager.

Definition at line 830 of file biddy.h.

5.1.2.119 #define Biddy_NodeTableGCTime( ) Biddy_Managed_NodeTableGCTime(NULL)

Macro Biddy_NodeTableGCTime is defined for use with anonymous manager.

Definition at line 835 of file biddy.h.

5.1.2.120 #define Biddy_NodeTableGCObsoleteNumber( ) Biddy_Managed_NodeTableGCObsoleteNumber(NULL)

Macro Biddy_NodeTableGCObsoleteNumber is defined for use with anonymous manager.

Definition at line 840 of file biddy.h.

5.1.2.121 #define Biddy_NodeTableDRTime( ) Biddy_Managed_NodeTableDRTime(NULL)

Macro Biddy_NodeTableDRTime is defined for use with anonymous manager.

Definition at line 845 of file biddy.h.

5.1.2.122 #define Biddy_FormulaTableNum( ) Biddy_Managed_FormulaTableNum(NULL)

Macro Biddy_FormulaTableNum is defined for use with anonymous manager.

Definition at line 850 of file biddy.h.

5.1.2.123 #define Biddy_ListUsed( ) Biddy_Managed_ListUsed(NULL)

Macro Biddy_ListUsed is defined for use with anonymous manager.

Definition at line 855 of file biddy.h.

5.1.2.124 #define Biddy_ListMaxLength( ) Biddy_Managed_ListMaxLength(NULL)

Macro Biddy_ListMaxLength is defined for use with anonymous manager.

Definition at line 860 of file biddy.h.

5.1.2.125 #define Biddy_ListAvgLength( ) Biddy_Managed_ListAvgLength(NULL)

Macro Biddy_ListAvgLength is defined for use with anonymous manager.

Definition at line 865 of file biddy.h.

5.1.2.126 #define Biddy_OPCacheSearch( ) Biddy_Managed_OPCacheSearch(NULL)

Macro Biddy_OPCacheSearch is defined for use with anonymous manager.

Definition at line 870 of file biddy.h.

5.1.2.127 #define Biddy_OPCacheFind( ) Biddy_Managed_OPCacheFind(NULL)

Macro Biddy_OPCacheFind is defined for use with anonymous manager.

Definition at line 875 of file biddy.h.

5.1.2.128 #define Biddy_OPCacheOverwrite( ) Biddy_Managed_OPCacheOverwrite(NULL)

Macro Biddy_OPCacheOverwrite is defined for use with anonymous manager.

Definition at line 880 of file biddy.h.

5.1.2.129 #define Biddy_NodeNumberPlain(*f*) Biddy_Managed_NodeNumberPlain(NULL,f)

Macro Biddy_NodeNumberPlain is defined for use with anonymous manager.

Definition at line 885 of file biddy.h.

5.1.2.130 #define Biddy_DependentVariableNumber(*f*) Biddy_Managed_DependentVariableNumber(NULL,f)

Macro Biddy_DependentVariableNumber is defined for use with anonymous manager.

Definition at line 890 of file biddy.h.

5.1.2.131 #define Biddy_NodeVarNumber(*f, n, v*) Biddy_Managed_NodeVarNumber(NULL,f,n,v)

Macro Biddy_NodeVarNumber is defined for use with anonymous manager.

Definition at line 895 of file biddy.h.

5.1.2.132 #define Biddy_CountPaths(*f*) Biddy_Managed_CountPaths(NULL,f)

Macro Biddy_CountPaths is defined for use with anonymous manager.

Definition at line 900 of file biddy.h.

5.1.2.133 #define Biddy_CountMinterm(*f, nvars*) Biddy_Managed_CountMinterm(NULL,f,nvars)

Macro Biddy_CountMinterm is defined for use with anonymous manager.

Definition at line 905 of file biddy.h.

5.1.2.134 #define Biddy_DensityFunction(*f, nvars*) Biddy_Managed_DensityFunction(NULL,f,nvars)

Macro Biddy_DensityFunction is defined for use with anonymous manager.

Definition at line 912 of file biddy.h.

5.1.2.135 #define Biddy_DensityBDD(*f, nvars*) Biddy_Managed_DensityBDD(NULL,f,nvars)

Macro Biddy_DensityBDD is defined for use with anonymous manager.

Definition at line 917 of file biddy.h.

5.1.2.136 #define Biddy_ReadMemoryInUse( ) Biddy_Managed_ReadMemoryInUse(NULL)

Macro Biddy_ReadMemoryInUse is defined for use with anonymous manager.

Definition at line 922 of file biddy.h.

5.1.2.137 #define Biddy_PrintInfo(*f*) Biddy_Managed_PrintInfo(NULL,f)

Macro Biddy_PrintInfo is defined for use with anonymous manager.

Definition at line 927 of file biddy.h.

5.1.2.138 #define Biddy_Eval0(*s*) Biddy_Managed_Eval0(NULL,s)

Macro Biddy_Eval0 is defined for use with anonymous manager.

Definition at line 944 of file biddy.h.

5.1.2.139 #define Biddy_Eval1x(*s, lf*) Biddy_Managed_Eval1x(NULL,s,lf)

Macro Biddy_Eval1x is defined for use with anonymous manager.

Definition at line 949 of file biddy.h.

5.1.2.140 #define Biddy_Eval2(*boolFunc*) Biddy_Managed_Eval2(NULL,boolFunc)

Macro Biddy_Eval2 is defined for use with anonymous manager.

Definition at line 956 of file biddy.h.

5.1.2.141 #define Biddy_ReadVerilogFile(*filename, prefix*) Biddy_Managed_ReadVerilogFile(NULL,filename,prefix)

Macro Biddy_ReadVerilogFile is defined for use with anonymous manager.

Definition at line 961 of file biddy.h.

5.1.2.142 #define Biddy_PrintfBDD(*f*) Biddy_Managed_PrintfBDD(NULL,f)

Macro Biddy_PrintfBDD is defined for use with anonymous manager.

Definition at line 966 of file biddy.h.

5.1.2.143 #define Biddy_WriteBDD(*filename, f, label*) Biddy_Managed_WriteBDD(NULL,filename,f,label)

Macro Biddy_WriteBDD is defined for use with anonymous manager.

Definition at line 971 of file biddy.h.

5.1.2.144 #define Biddy_PrintfTable(*f*) **Biddy_Managed_PrintfTable**(NULL,f)

Macro Biddy_PrintfTable is defined for use with anonymous manager.

Definition at line 976 of file biddy.h.

5.1.2.145 #define Biddy_WriteTable(*filename, f*) **Biddy_Managed_WriteTable**(NULL,filename,f)

Macro Biddy_WriteTable is defined for use with anonymous manager.

Definition at line 981 of file biddy.h.

5.1.2.146 #define Biddy_PrintfSOP(*f*) **Biddy_Managed_PrintfSOP**(NULL,f)

Macro Biddy_PrintfSOP is defined for use with anonymous manager.

Definition at line 986 of file biddy.h.

5.1.2.147 #define Biddy_WriteSOP(*filename, f*) **Biddy_Managed_WriteSOP**(NULL,filename,f)

Macro Biddy_WriteSOP is defined for use with anonymous manager.

Definition at line 991 of file biddy.h.

5.1.2.148 #define Biddy_WriteDot(*filename, f, label, id, cudd*) **Biddy_Managed_WriteDot**(NU↩ LL,filename,f,label,id,cudd);

Macro Biddy_WriteDot is defined for use with anonymous manager.

Definition at line 996 of file biddy.h.

5.1.2.149 #define Biddy_WriteBddview(*filename, f, label, table*) **Biddy_Managed_WriteBddview**(NU↩ LL,filename,f,label,table);

Macro Biddy_WriteBddview is defined for use with anonymous manager.

Definition at line 1001 of file biddy.h.

5.1.3 Typedef Documentation

5.1.3.1 typedef char **Biddy_Boolean**

Biddy_Boolean is used for boolean values.

Definition at line 191 of file biddy.h.

5.1.3.2 typedef char* Biddy_String

Biddy_String is used for strings.

Definition at line 194 of file biddy.h.

5.1.3.3 typedef void** Biddy_Manager

Biddy_Manager is used to specify manager. Manager is a pointer to BiddyManager. A manager includes Node Table, Variable Table, Formulae Table, Ordering Table, three basic caches (ITE Cache, EA Cache and RC Cache), list of user's caches, system age and some other structures needed for memory management. Internal structure of BiddyManager is not exported but must be imitated to create user's managers

Definition at line 203 of file biddy.h.

5.1.3.4 typedef void* Biddy_Cache

Biddy_Cache is used to specify user's cache table. Caches for different operations are different and the user is responsible for the correct internal structure.

Definition at line 208 of file biddy.h.

5.1.3.5 typedef unsigned short int Biddy_Variable

Biddy_Variable is used for indices in variable table.

Definition at line 212 of file biddy.h.

5.1.3.6 typedef void* Biddy_Edge

Biddy_Edge is a marked edge (i.e. a marked pointer to BiddyNode). Mark is encoded as the value of the last significant bit. For TZBDDs and TZFDDs, edges are tagged. Tag is a 16 bit number (unsigned short int) whish is stored in the highest part of the pointer (this is safe because only 48 bits are used). TZBDDs and TZFDDs are supported only on 64-bits architectures. Internal structure of BiddyNode is not visible to the user.

Definition at line 221 of file biddy.h.

5.1.3.7 typedef void(* Biddy_GCFunction) (Biddy_Manager)

Biddy_GCFunction is used in Biddy_AddCache to specify user's function which will performs garbage collection.

Definition at line 225 of file biddy.h.

5.1.3.8 typedef Biddy_Boolean(* Biddy_LookupFunction) (Biddy_String, Biddy_Edge *)

Biddy_LookupFunction is used in Biddy_Eval1x to specify user's function which will lookups in a user's formula table.

Definition at line 229 of file biddy.h.

5.2 biddyInOut.c File Reference

File biddyInOut.c contains various parsers and generators.

```
#include "biddyInt.h"
```

Functions

- Biddy_String Biddy_Managed_Eval0 (Biddy_Manager MNG, Biddy_String s)

  Function Biddy_Managed_Eval0 evaluates raw format.
- Biddy_Edge Biddy_Managed_Eval1x (Biddy_Manager MNG, Biddy_String s, Biddy_LookupFunction lf)

  Function Biddy_Managed_Eval1x evaluates prefix AND-OR-EXOR-NOT format.
- Biddy_Edge Biddy_Managed_Eval2 (Biddy_Manager MNG, Biddy_String boolFunc)

  Function Biddy_Managed_Eval2 evaluates infix &|^∼>< format.
- void Biddy_Managed_ReadVerilogFile (Biddy_Manager MNG, const char filename[ ], Biddy_String prefix)

  Function Biddy_Managed_ReadVerilogFile reads Verilog file and creates variables for all primary inputs and Boolean functions for all primary outputs.
- void Biddy_Managed_PrintfBDD (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_PrintfBDD writes raw format using printf.
- void Biddy_Managed_WriteBDD (Biddy_Manager MNG, const char filename[ ], Biddy_Edge f, Biddy_String label)

  Function Biddy_Managed_WriteBDD writes raw format using fprintf.
- void Biddy_Managed_PrintfTable (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_PrintfTable writes truth table using printf.
- void Biddy_Managed_WriteTable (Biddy_Manager MNG, const char filename[ ], Biddy_Edge f)

  Function Biddy_Managed_WriteTable writes truth table using fprintf.
- void Biddy_Managed_PrintfSOP (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_PrintfSOP writes SOP using printf.
- void Biddy_Managed_WriteSOP (Biddy_Manager MNG, const char filename[ ], Biddy_Edge f)

  Function Biddy_Managed_WriteSOP writes SOP using fprintf.
- unsigned int Biddy_Managed_WriteDot (Biddy_Manager MNG, const char filename[ ], Biddy_Edge f, const char label[ ], int id, Biddy_Boolean cudd)

  Function Biddy_Managed_WriteDot writes dot/graphviz format using fprintf.
- unsigned int Biddy_Managed_WriteBddview (Biddy_Manager MNG, const char filename[ ], Biddy_Edge f, const char label[ ], Biddy_XY ∗table)

  Function Biddy_Managed_WriteBDDView writes bddview format using fprintf.

5.2.1 Detailed Description

File biddyInOut.c contains various parsers and generators.

Description

```
PackageName [Biddy]
Synopsis    [Biddy provides data structures and algorithms for the
             representation and manipulation of Boolean functions with
             ROBDDs. A hash table is used for quick search of nodes.
             Complement edges decreases the number of nodes. An automatic
             garbage collection with a system age is implemented.
             Variable swapping and sifting are implemented.]

FileName    [biddyInOut.c]
Revision    [$Revision: 251 $]
Date        [$Date: 2017-03-01 21:34:41 +0100 (sre, 01 mar 2017) $]
Authors     [Robert Meolic (robert.meolic@um.si),
             Ales Casar (ales@homemade.net),
             Jan Kraner (jankristian.kraner@student.um.si),
             Ziga Kobale (ziga.kobale@student.um.si),
             Volodymyr Mihav (mihaw.wolodymyr@gmail.com),
             David Kebo Houngninou (dhoungninou@smu.edu)]
```

Copyright

More info

See also: biddy.h, biddyInt.h

5.2.2 Function Documentation

5.2.2.1 Biddy_String Biddy_Managed_Eval0 (Biddy_Manager *MNG,* Biddy_String *s*)

Function Biddy_Managed_Eval0 evaluates raw format.

Description

First word is a name. It is followed by raw format. Function return name of the formula.

Side effects

All variables should already exists in the correct ordering! Not reentrant.

More info

Macro Biddy_Eval0(s) is defined for use with anonymous manager.

Definition at line 238 of file biddyInOut.c.

5.2.2.2 Biddy_Edge Biddy_Managed_Eval1x (Biddy_Manager *MNG*, Biddy_String *s*, Biddy_LookupFunction *lf*)

Function Biddy_Managed_Eval1x evaluates prefix AND-OR-EXOR-NOT format.

Description

Parameter lf is a lookup function in the user-defined cache table.

Side effects

Not reentrant.

More info

Macro Biddy_Eval1x(s,lf) is defined for use with anonymous manager. Macros Biddy_Managed_Eval1(s) and Biddy_Eval1(s) are defined for use without searching in the user-defined cache.

Definition at line 304 of file biddyInOut.c.

5.2.2.3 Biddy_Edge Biddy_Managed_Eval2 (Biddy_Manager *MNG*, Biddy_String *boolFunc*)

Function Biddy_Managed_Eval2 evaluates infix &|^~>< format.

Description

Boolean constants are '0' and '1'. Parenthesis are implemented. Operators' priority is implemented. Formula Tree is suported (global table, only). Operators '*' and '+' are also allowed for conjunction/disjunction.

Side effects

Variable names must be one letter a-zA-Z optionally followed by int number.

More info

Original author: Volodymyr Mihav (mihaw.wolodymyr@gmail.com) Original implementation of this function is on https://github.com/sungmaster/liBDD. Macro Biddy_Eval2(boolFunc) is defined for use with anonymous manager.

Definition at line 360 of file biddyInOut.c.

5.2.2.4 void Biddy_Managed_ReadVerilogFile (Biddy_Manager *MNG*, const char *filename[]*, Biddy_String *prefix*)

Function Biddy_Managed_ReadVerilogFile reads Verilog file and creates variables for all primary inputs and Boolean functions for all primary outputs.

Description

If (prefix != NULL) then the created BDD variables and formulae will get it.

Side effects

More info

Original author: David Kebo Houngninou, Southern Methodist University Original implementation of this function is on https://github.com/davidkebo/verilog-parser Macro Biddy_ReadVerilogFile(filename,prefix) is defined for use with anonymous manager.

Definition at line 546 of file biddyInOut.c.

5.2.2.5 void Biddy_Managed_PrintfBDD (Biddy_Manager *MNG*, Biddy_Edge *f*)

Function Biddy_Managed_PrintfBDD writes raw format using printf.

Description

This function is intended for writing to an output channel via macro which overrides the meaning of standard printf calls. For writing raw format into the file, use Biddy_Managed_WriteBDD.

Side effects

More info

Macro Biddy_PrintfBDD(f) is defined for use with anonymous manager.

Definition at line 647 of file biddyInOut.c.

5.2.2.6 void Biddy_Managed_WriteBDD (Biddy_Manager *MNG*, const char *filename[]*, Biddy_Edge *f*, Biddy_String *label*)

Function Biddy_Managed_WriteBDD writes raw format using fprintf.

Description

Side effects

More info

Macro Biddy_WriteBDD(f) is defined for use with anonymous manager.

Definition at line 696 of file biddyInOut.c.

5.2.2.7 void Biddy_Managed_PrintfTable (Biddy_Manager *MNG*, Biddy_Edge *f*)

Function Biddy_Managed_PrintfTable writes truth table using printf.

Description

This function is intended for writing to an output channel via macro which overrides the meaning of standard printf calls. For writing truth table into the file, use Biddy_Managed_WriteTable.

Side effects

More info

Thanks to Jan Kraner (jankristian.kraner@student.um.si) and Ziga Kobale (ziga.kobale@student.um.si). Macro Biddy_PrintfTable(f) is defined for use with anonymous manager.

Definition at line 743 of file biddyInOut.c.

5.2.2.8 void Biddy_Managed_WriteTable (Biddy_Manager *MNG*, const char *filename[ ]*, Biddy_Edge *f*)

Function Biddy_Managed_WriteTable writes truth table using fprintf.

Description

Side effects

More info

Thanks to Jan Kraner (jankristian.kraner@student.um.si) and Ziga Kobale (ziga.kobale@student.um.si). Macro Biddy_WriteTable(f) is defined for use with anonymous manager.

Definition at line 834 of file biddyInOut.c.

5.2.2.9 void Biddy_Managed_PrintfSOP (Biddy_Manager *MNG*, Biddy_Edge *f*)

Function Biddy_Managed_PrintfSOP writes SOP using printf.

Description

Side effects

More info

Definition at line 856 of file biddyInOut.c.

5.2.2.10 void Biddy_Managed_WriteSOP (Biddy_Manager *MNG*, const char *filename[ ]*, Biddy_Edge *f*)

Function Biddy_Managed_WriteSOP writes SOP using fprintf.

Description

Side effects

More info

Definition at line 885 of file biddyInOut.c.

5.2.2.11 unsigned int Biddy_Managed_WriteDot (Biddy_Manager *MNG*, const char *filename[ ]*, Biddy_Edge *f*, const char *label[ ]*, int *id*, Biddy_Boolean *cudd*)

Function Biddy_Managed_WriteDot writes dot/graphviz format using fprintf.

Description

Output dot format. Two approaches are implemented. The CUDD-like implementation is copyied from CUDD 3.0.

Side effects

If (id != -1) then id is used instead of <...> for variable names. Function resets all variables value.

More info

Macro Biddy_WriteDot(filename,f,label) is defined for use with anonymous manager.

Definition at line 914 of file biddyInOut.c.

5.2.2.12 unsigned int Biddy_Managed_WriteBddview (Biddy_Manager *MNG*, const char *filename[ ]*, Biddy_Edge *f*, const char *label[ ]*, Biddy_XY * *table*)

Function Biddy_Managed_WriteBDDView writes bddview format using fprintf.

Description

Output bddview format.

Side effects

Parameter table is optional, if not NULL then it must contain node names and coordinates.

More info

Macro Biddy_WriteBddview(filename,f,label) is defined for use with anonymous manager.

Definition at line 1115 of file biddyInOut.c.

5.3 biddyInt.h File Reference

File biddyInt.h contains declaration of internal data structures.

```
#include "biddy.h"
#include <assert.h>
#include <string.h>
#include <ctype.h>
#include <gmp.h>
```

5.3.1 Detailed Description

File biddyInt.h contains declaration of internal data structures.

Description

```
PackageName [Biddy]
Synopsis    [Biddy provides data structures and algorithms for the
             representation and manipulation of Boolean functions with
             ROBDDs. A hash table is used for quick search of nodes.
             Complement edges decreases the number of nodes. An automatic
             garbage collection with a system age is implemented.
             Variable swapping and sifting are implemented.]

FileName    [biddyInt.h]
Revision    [$Revision: 255 $]
Date        [$Date: 2017-03-20 19:48:30 +0100 (pon, 20 mar 2017) $]
Authors     [Robert Meolic (robert.meolic@um.si),
             Ales Casar (ales@homemade.net)]
```

Copyright

More info

See also: biddy.h

5.4 biddyMain.c File Reference

File biddyMain.c contains main functions for representation and manipulation of boolean functions with ROBDDs.

```
#include "biddyInt.h"
```

Functions

- void Biddy_InitMNG (Biddy_Manager *mng, int gddtype)

  Function Biddy_InitMNG initialize a manager.
- void Biddy_ExitMNG (Biddy_Manager *mng)

  Function Biddy_ExitMNG deletes a manager.
- Biddy_String Biddy_About ()

  Function Biddy_About reports version of Biddy package.
- int Biddy_Managed_GetManagerType (Biddy_Manager MNG)

  Function Biddy_Managed_GetManagerType reports BDD type used in the manager.
- void Biddy_Managed_SetManagerParameters (Biddy_Manager MNG, float gcr, float gcrF, float gcrX, float rr, float rrF, float rrX, float st, float fst, float cst, float fcst)

  Function Biddy_Managed_SetManagerParameters set modifiable parameters.
- Biddy_Edge Biddy_GetThen (Biddy_Edge fun)

  Function Biddy_GetThen returns THEN successor.
- Biddy_Edge Biddy_GetElse (Biddy_Edge fun)

  Function Biddy_GetElse returns ELSE successor.
- Biddy_Variable Biddy_GetTopVariable (Biddy_Edge fun)

  Function Biddy_GetTopVariable returns the top variable.
- Biddy_Boolean Biddy_Managed_IsEqv (Biddy_Manager MNG1, Biddy_Edge f1, Biddy_Manager MNG2, Biddy_Edge f2)

  Function Biddy_Managed_IsEqv returns TRUE iff two BDDs are equal.
- void Biddy_Managed_SelectNode (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_SelectNode selects the top node of the given function.
- void Biddy_Managed_DeselectNode (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_DeselectNode deselects the top node of the given function.
- Biddy_Boolean Biddy_Managed_IsSelected (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_IsSelected returns TRUE iff the top node of the given function is selected.
- void Biddy_Managed_SelectFunction (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_SelectFunction recursively selects all nodes of a given function.
- void Biddy_Managed_DeselectAll (Biddy_Manager MNG)

  Function Biddy_Managed_DeselectAll deselects all nodes.
- Biddy_Edge Biddy_Managed_GetTerminal (Biddy_Manager MNG)

  Function Biddy_Managed_GetTerminal returns unmarked edge pointing to the constant node 1.
- Biddy_Edge Biddy_Managed_GetConstantZero (Biddy_Manager MNG)

  Function Biddy_Managed_GetConstantZero returns constant 0.
- Biddy_Edge Biddy_Managed_GetConstantOne (Biddy_Manager MNG)

  Function Biddy_Managed_GetConstantOne returns constant 1.
- Biddy_Edge Biddy_Managed_GetBaseSet (Biddy_Manager MNG)

  Function Biddy_Managed_GetBaseSet returns set containing only a null combination, i.e. it returns {{}}.
- Biddy_Variable Biddy_Managed_GetVariable (Biddy_Manager MNG, Biddy_String x)

  Function Biddy_Managed_GetVariable returns variable with the given name.
- Biddy_Variable Biddy_Managed_GetPrevVariable (Biddy_Manager MNG, Biddy_Variable v)

  Function Biddy_Managed_GetPrevVariable returns previous variable in the global ordering (lower, topmore).
- Biddy_Variable Biddy_Managed_GetNextVariable (Biddy_Manager MNG, Biddy_Variable v)

  Function Biddy_Managed_GetNextVariable returns next variable in the global ordering (higher, bottommore).
- Biddy_Edge Biddy_Managed_GetVariableEdge (Biddy_Manager MNG, Biddy_Variable v)

  Function Biddy_Managed_GetVariableEdge returns variable's edge.
- Biddy_Edge Biddy_Managed_GetElementEdge (Biddy_Manager MNG, Biddy_Variable v)

  Function Biddy_Managed_GetElementEdge returns element's edge.
- Biddy_String Biddy_Managed_GetVariableName (Biddy_Manager MNG, Biddy_Variable v)

Function Biddy_Managed_GetVariableName returns the name of a variable.

- Biddy_Edge Biddy_Managed_GetTopVariableEdge (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_GetTopVariableEdge returns variable's edge of top variable.
- Biddy_String Biddy_Managed_GetTopVariableName (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_GetTopVariableName returns the name of top variable.
- char Biddy_Managed_GetTopVariableChar (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_GetTopVariableChar returns the first character in the name of top variable.
- void Biddy_Managed_ResetVariablesValue (Biddy_Manager MNG)

  Function Biddy_Managed_ResetVariablesValue sets all variable's value to biddyZero.
- void Biddy_Managed_SetVariableValue (Biddy_Manager MNG, Biddy_Variable v, Biddy_Edge f)

  Function Biddy_Managed_SetVariableValue sets variable's value.
- Biddy_Boolean Biddy_Managed_IsSmaller (Biddy_Manager MNG, Biddy_Variable fv, Biddy_Variable gv)

  Function Biddy_Managed_IsSmaller returns TRUE if the first variable is smaller (= lower = previous = above = top-more).
- Biddy_Variable Biddy_Managed_FoaVariable (Biddy_Manager MNG, Biddy_String x, Biddy_Boolean varelem)

  Function Biddy_Managed_FoaVariable finds variable/element or adds new variable (i.e. Boolean function f = x) and new element (i.e. it creates set {{x}}).
- Biddy_Edge Biddy_Managed_AddVariableByName (Biddy_Manager MNG, Biddy_String x)

  Function Biddy_Managed_AddVariableByName adds variable.
- Biddy_Edge Biddy_Managed_AddElementByName (Biddy_Manager MNG, Biddy_String x)

  Function Biddy_Managed_AddElementByName adds element.
- Biddy_Edge Biddy_Managed_AddVariableBelow (Biddy_Manager MNG, Biddy_Variable v)

  Function Biddy_Managed_AddVariableBelow adds a numbered variable.
- Biddy_Edge Biddy_Managed_AddVariableAbove (Biddy_Manager MNG, Biddy_Variable v)

  Function Biddy_Managed_AddVariableAbove adds a numbered variable.
- Biddy_Edge Biddy_Managed_TransferMark (Biddy_Manager MNG, Biddy_Edge f, Biddy_Boolean mark, Biddy_Boolean leftright)

  Function Biddy_Managed_TransferMark returns edge with inverted complement bit iff the second parameter is TRUE and normalization rules require this.
- Biddy_Edge Biddy_Managed_IncTag (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_IncTag returns edge with an incremented tag.
- Biddy_Edge Biddy_Managed_TaggedFoaNode (Biddy_Manager MNG, Biddy_Variable v, Biddy_Edge pf, Biddy_Edge pt, Biddy_Variable ptag, Biddy_Boolean garbageAllowed)

  Function Biddy_Managed_TaggedFoaNode finds or adds new node with the given variable and successors.
- Biddy_Edge Biddy_Managed_Not (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_Not calculates Boolean function NOT.
- Biddy_Edge Biddy_Managed_ITE (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge g, Biddy_Edge h)

  Function Biddy_Managed_ITE calculates ITE operation of three Boolean functions.
- Biddy_Edge Biddy_Managed_And (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge g)

  Function Biddy_Managed_And calculates Boolean function AND (conjunction).
- Biddy_Edge Biddy_Managed_Or (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge g)

  Function Biddy_Managed_Or calculates Boolean function OR (disjunction).
- Biddy_Edge Biddy_Managed_Nand (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge g)

  Function Biddy_Managed_Nand calculates Boolean function NAND (Sheffer).
- Biddy_Edge Biddy_Managed_Nor (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge g)

  Function Biddy_Managed_Nor calculates Boolean function NOR (Peirce).
- Biddy_Edge Biddy_Managed_Xor (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge g)

  Function Biddy_Managed_Xor calculates Boolean function XOR.
- Biddy_Edge Biddy_Managed_Xnor (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge g)

  Function Biddy_Managed_Xnor calculates Boolean function XNOR.
- Biddy_Edge Biddy_Managed_Leq (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge g)

Function Biddy_Managed_Leq calculates Boolean implication.

- Biddy_Edge Biddy_Managed_Gt (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge g)

  Function Biddy_Managed_Gt calculates the negation of Boolean implication.
- Biddy_Boolean Biddy_Managed_IsLeq (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge g)

  Function Biddy_Managed_IsLeq returns TRUE iff function f is included in function g.
- Biddy_Edge Biddy_Managed_SubIntersect (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge g)

  Biddy_Managed_SubIntersect calculates a function included in the intersection of f and g.
- Biddy_Edge Biddy_Managed_Restrict (Biddy_Manager MNG, Biddy_Edge f, Biddy_Variable v, Biddy_Boolean value)

  Function Biddy_Managed_Restrict calculates a restriction of Boolean function.
- Biddy_Edge Biddy_Managed_Compose (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge g, Biddy_Variable v)

  Function Biddy_Managed_Compose calculates a composition of two Boolean functions.
- Biddy_Edge Biddy_Managed_E (Biddy_Manager MNG, Biddy_Edge f, Biddy_Variable v)

  Function Biddy_Managed_E calculates an existential quantification of Boolean function.
- Biddy_Edge Biddy_Managed_A (Biddy_Manager MNG, Biddy_Edge f, Biddy_Variable v)

  Function Biddy_Managed_A calculates an universal quantification of Boolean function.
- Biddy_Boolean Biddy_Managed_IsVariableDependent (Biddy_Manager MNG, Biddy_Edge f, Biddy_Variable v)

  Function Biddy_Managed_IsVariableDependent returns TRUE iff variable is dependent on others in a function.
- Biddy_Edge Biddy_Managed_ExistAbstract (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge cube)

  Function Biddy_Managed_ExistAbstract existentially abstracts all the variables in cube from f.
- Biddy_Edge Biddy_Managed_UnivAbstract (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge cube)

  Function Biddy_Managed_UnivAbstract universally abstracts all the variables in cube from f.
- Biddy_Edge Biddy_Managed_AndAbstract (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge g, Biddy_Edge cube)

  Function Biddy_Managed_AndAbstract calculates the AND of two BDDs and simultaneously (existentially) abstracts the variables in cube.
- Biddy_Edge Biddy_Managed_Constrain (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge c)

  Function Biddy_Managed_Constrain calculates Coudert and Madre's constrain function.
- Biddy_Edge Biddy_Managed_Simplify (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge c)

  Function Biddy_Managed_Simplify calculates Coudert and Madre's restrict function.
- Biddy_Edge Biddy_Managed_Support (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_Support calculates a product of all dependent variables.
- Biddy_Edge Biddy_Managed_Replace (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_Replace calculates BDD with one or more variables replaced.
- Biddy_Edge Biddy_Managed_Change (Biddy_Manager MNG, Biddy_Edge f, Biddy_Variable v)

  Function Biddy_Managed_Change change the form of the given variable (positive literal becomes negative and vice versa).
- Biddy_Edge Biddy_Managed_Subset (Biddy_Manager MNG, Biddy_Edge f, Biddy_Variable v, Biddy_Boolean value)

  Function Biddy_Managed_Subset calculates a division of Boolean function with a literal.
- Biddy_Boolean Biddy_Managed_IsOK (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_IsOK returns TRUE iff given node is not obsolete.
- void Biddy_Managed_GC (Biddy_Manager MNG, Biddy_Variable target, Biddy_Boolean purge, Biddy_Boolean total)

  Function Biddy_Managed_GC performs garbage collection.
- void Biddy_Managed_Clean (Biddy_Manager MNG)

  Function Biddy_Managed_Clean performs cleaning.
- void Biddy_Managed_Purge (Biddy_Manager MNG)

  Function Biddy_Managed_Purge immediately removes all nodes which were not preserved or which are not preserved anymore.

- void Biddy_Managed_PurgeAndReorder (Biddy_Manager MNG, Biddy_Edge f, Biddy_Boolean converge)

  Function Biddy_Managed_PurgeAndReorder immediately removes non-preserved nodes and triggers reordering on function.
- void Biddy_Managed_Refresh (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_Refresh refreshes top node in a given function.
- void Biddy_Managed_AddCache (Biddy_Manager MNG, Biddy_GCFunction gc)

  Function Biddy_Managed_AddCache adds cache to the end of Cache list.
- unsigned int Biddy_Managed_AddFormula (Biddy_Manager MNG, Biddy_String x, Biddy_Edge f, int c)

  Function Biddy_Managed_AddFormula adds formula to Formula table.
- Biddy_Boolean Biddy_Managed_FindFormula (Biddy_Manager MNG, Biddy_String x, Biddy_Edge *f)

  Function Biddy_Managed_FindFormula find formula in Formula table.
- Biddy_Boolean Biddy_Managed_DeleteFormula (Biddy_Manager MNG, Biddy_String x)

  Function Biddy_Managed_DeleteFormula delete formula from Formula table.
- Biddy_Boolean Biddy_Managed_DeleteIthFormula (Biddy_Manager MNG, unsigned int i)

  Function Biddy_Managed_DeleteIthFormula deletes formula from the table.
- Biddy_Edge Biddy_Managed_GetIthFormula (Biddy_Manager MNG, unsigned int i)

  Function Biddy_Managed_GetIthFormula returns ith formula in a Formula table.
- Biddy_String Biddy_Managed_GetIthFormulaName (Biddy_Manager MNG, unsigned int i)

  Function Biddy_Managed_GetIthFormulaName returns name of the ith formula in a Formula table.
- Biddy_Variable Biddy_Managed_SwapWithHigher (Biddy_Manager MNG, Biddy_Variable v)

  Function Biddy_Managed_SwapWithHigher swaps two adjacent variables.
- Biddy_Variable Biddy_Managed_SwapWithLower (Biddy_Manager MNG, Biddy_Variable v)

  Function Biddy_Managed_SwapWithLower swaps two adjacent variables.
- Biddy_Boolean Biddy_Managed_Sifting (Biddy_Manager MNG, Biddy_Edge f, Biddy_Boolean converge)

  Function Biddy_Managed_Sifting reorders variables to minimize node number for the whole system (if f = NULL) or for the given function (if f != NULL) using Rudell's sifting algorithm.
- Biddy_Edge Biddy_Managed_Copy (Biddy_Manager MNG1, Biddy_Manager MNG2, Biddy_Edge f)

  Function Biddy_Managed_Copy copies a graph from one manager to another manager.

  Description

  The function takes a graph from one manager and creates the same graph in another manager. The resulting graph will represent the same Boolean function assuming the domain from the target manager. If f = biddyZero then only the domain is copied.

  Side effects

  If source and target manager are the same then function does nothing. The variable ordering of created BDD is adapted to the target manager.

  More info

  Macro Biddy_Copy(MNG2,f) is defined for use with anonymous manager.
- void Biddy_Managed_CopyFormula (Biddy_Manager MNG1, Biddy_Manager MNG2, Biddy_String x)

  Function Biddy_Managed_CopyFormula uses Biddy_Managed_Copy to copy a graph from one manager to another manager.

  Description

  See Biddy_Managed_Copy.

  Side effects

  If source and target manager are the same then function does nothing. The variable ordering of created BDD is adapted to the target manager. The created formula is not preserved.

  More info

  Macro Biddy_CopyFormula(MNG2,x) is defined for use with anonymous manager.
- Biddy_Boolean Biddy_Managed_Eval (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_Eval returns the value of a Boolean function for a given variable assignment.

Description

Side effects

More info

Macro Biddy_Eval(f) is defined for use with anonymous manager.

- Biddy_Edge Biddy_Managed_Random (Biddy_Manager MNG, Biddy_Edge support, double r)

  Function Biddy_Managed_Random generates a random BDD.
- Biddy_Edge Biddy_Managed_RandomSet (Biddy_Manager MNG, Biddy_Edge unit, double r)

  Function Biddy_Managed_RandomSet generates a random BDD.

5.4.1 Detailed Description

File biddyMain.c contains main functions for representation and manipulation of boolean functions with ROBDDs.

Description

```
PackageName [Biddy]
Synopsis    [Biddy provides data structures and algorithms for the
             representation and manipulation of Boolean functions with
             ROBDDs. A hash table is used for quick search of nodes.
             Complement edges decreases the number of nodes. An automatic
             garbage collection with a system age is implemented.
             Variable swapping and sifting are implemented.]

FileName    [biddyMain.c]
Revision    [$Revision: 253 $]
Date        [$Date: 2017-03-20 09:03:47 +0100 (pon, 20 mar 2017) $]
Authors     [Robert Meolic (robert.meolic@um.si),
             Ales Casar (ales@homemade.net)]
```

Copyright

More info

See also: biddy.h, biddyInt.h

5.4.2 Function Documentation

5.4.2.1 void Biddy_InitMNG (Biddy_Manager * *mng*, int *gddtype*)

Function Biddy_InitMNG initialize a manager.

Description

Biddy_InitMNG creates and initializes a manager. Initialization consists of creating manager structure (MNG), node table (biddyNodeTable), variable table (biddyVariableTable), formula table (biddyFormulaTable), three basic caches (biddyOPCache, biddyEACache and biddyRCCache), and cache list (biddyCacheList). Biddy_InitMNG also initializes constant edges (biddyOne, biddyZero), memory management and automatic garbage collection.

Side effects

Allocates a lot of memory. Parameter gddtype is ignored.

More info

Macro Biddy_Init() will initialize anonymous manager.

Definition at line 210 of file biddyMain.c.

5.4.2.2 void Biddy_ExitMNG (Biddy_Manager * *mng*)

Function Biddy_ExitMNG deletes a manager.

Description

Deallocates all memory allocated by Biddy_InitMNG, Biddy_FoaVariable, Biddy_FoaNode etc.

Side effects

More info

Macro Biddy_Exit() will delete anonymous manager.

Definition at line 691 of file biddyMain.c.

5.4.2.3 Biddy_String Biddy_About ( )

Function Biddy_About reports version of Biddy package.

Description

Side effects

More info

Definition at line 886 of file biddyMain.c.

5.4.2.4 int Biddy_Managed_GetManagerType (Biddy_Manager *MNG*)

Function Biddy_Managed_GetManagerType reports BDD type used in the manager.

Description

Side effects

More info

Macro Biddy_GetManagerType() is defined for use with anonymous manager.

Definition at line 908 of file biddyMain.c.

5.4.2.5 void Biddy_Managed_SetManagerParameters (Biddy_Manager *MNG*, float *gcr*, float *gcrF*, float *gcrX*, float *rr*, float *rrF*, float *rrX*, float *st*, float *fst*, float *cst*, float *fcst*)

Function Biddy_Managed_SetManagerParameters set modifiable parameters.

Description

Function expect 6 float values. If the value is < 0 then the parameter is not modified. The parameters are: biddy↩NodeTable.gcratio (do not delete nodes if the effect is to small), biddyNodeTable.gcratioF (do not delete nodes if the effect is to small), biddyNodeTable.gcratioX (do not delete nodes if the effect is to small), biddyNodeTable.resizeratio (resize Node table if there are to many nodes), biddyNodeTable.resizeratioF (resize Node table if there are to many nodes), biddyNodeTable.resizeratioX (resize Node table if there are to many nodes), biddyNodeTable.siftingtreshold (stop sifting if the size of the system grows to much), biddyNodeTable.fsiftingtreshold (stop sifting if the size of the function grows to much), biddyNodeTable.convergesiftingtreshold (stop one step of converging sifting if the size of the system grows to much), biddyNodeTable.fconvergesiftingtreshold (stop one step of converging sifting if the size of the function grows to much).

Side effects

Initial values are given in Biddy_InitMNG.

More info

Macro Biddy_SetManagerParameters() is defined for use with anonymous manager.

Definition at line 951 of file biddyMain.c.

5.4.2.6 Biddy_Edge Biddy_GetThen (Biddy_Edge *fun*)

Function Biddy_GetThen returns THEN successor.

Description

Input mark is not transfered! External use, only.

Side effects

More info

Macro BiddyT(fun) is defined for internal use.

Definition at line 988 of file biddyMain.c.

5.4.2.7 Biddy_Edge Biddy_GetElse (Biddy_Edge *fun*)

Function Biddy_GetElse returns ELSE successor.

Description

Input mark is not transfered! External use, only.

Side effects

More info

Macro BiddyE(fun) is defined for internal use.

Definition at line 1024 of file biddyMain.c.

5.4.2.8 Biddy_Variable Biddy_GetTopVariable (Biddy_Edge *fun*)

Function Biddy_GetTopVariable returns the top variable.

Description

External use, only.

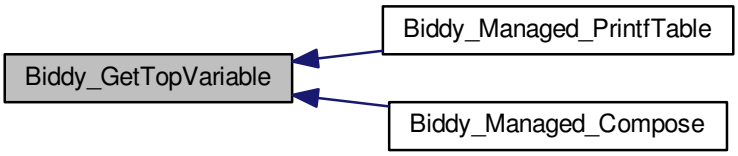
Side effects

More info

Macro BiddyV(fun) is defined for internal use.

Definition at line 1060 of file biddyMain.c.

Here is the caller graph for this function:



5.4.2.9 Biddy_Boolean Biddy_Managed_IsEqv (Biddy_Manager *MNG1,* Biddy_Edge *f1,* Biddy_Manager *MNG2,* Biddy_Edge *f2*)

Function Biddy_Managed_IsEqv returns TRUE iff two BDDs are equal.

Description

It is assumed that f1 and f2 have the same ordering.

Side effects

More info

Macro Biddy_IsEqv(f1,MNG2,f2) is defined for use with anonymous manager.

Definition at line 1084 of file biddyMain.c.

5.4.2.10 void Biddy_Managed_SelectNode (Biddy_Manager *MNG,* Biddy_Edge *f*)

Function Biddy_Managed_SelectNode selects the top node of the given function.

Description

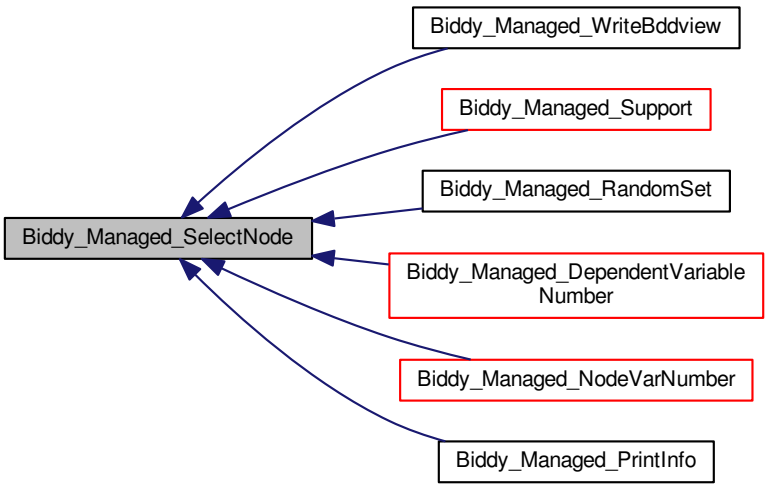
Side effects

More info

Macro Biddy_SelectNode(f) is defined for use with anonymous manager.

Definition at line 1119 of file biddyMain.c.

Here is the caller graph for this function:



5.4.2.11 void Biddy_Managed_DeselectNode (Biddy_Manager *MNG,* Biddy_Edge *f*)

Function Biddy_Managed_DeselectNode deselects the top node of the given function.

Description

Side effects

More info

Macro Biddy_DeselectNode(f) is defined for use with anonymous manager.

Definition at line 1145 of file biddyMain.c.

Here is the caller graph for this function:

5.4.2.12 Biddy_Boolean Biddy_Managed_IsSelected (Biddy_Manager *MNG,* Biddy_Edge *f*)

Function Biddy_Managed_IsSelected returns TRUE iff the top node of the given function is selected.

Description

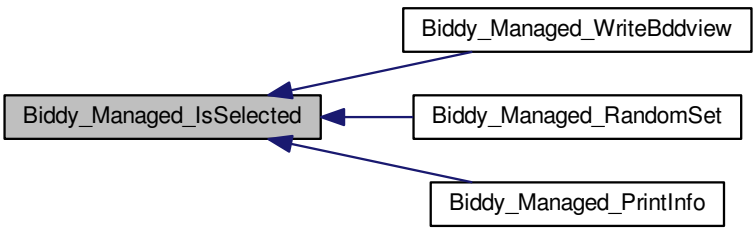
Side effects

More info

Macro Biddy_IsSelected(f) is defined for use with anonymous manager.

Definition at line 1171 of file biddyMain.c.

Here is the caller graph for this function:



5.4.2.13 void Biddy_Managed_SelectFunction (Biddy_Manager *MNG,* Biddy_Edge *f*)

Function Biddy_Managed_SelectFunction recursively selects all nodes of a given function.

Description

Side effects

Constant node must be selected before starting this function!

More info

Macro Biddy_SelectFunction(f) is defined for use with anonymous manager.

Definition at line 1198 of file biddyMain.c.

5.4.2.14 void Biddy_Managed_DeselectAll (Biddy_Manager *MNG*)

Function Biddy_Managed_DeselectAll deselects all nodes.

Description

Deselect all nodes.

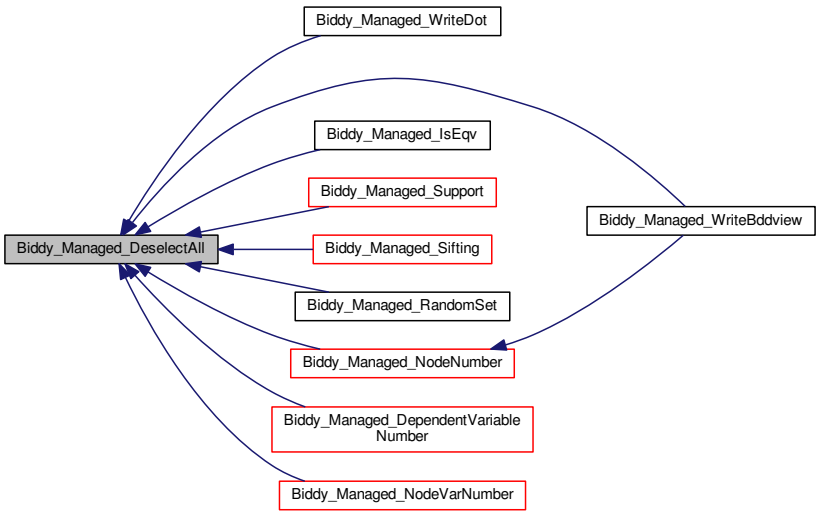
Side effects

More info

Macro Biddy_DeselectAll() is defined for use with anonymous manager.

Definition at line 1255 of file biddyMain.c.

Here is the caller graph for this function:



5.4.2.15 Biddy_Edge Biddy_Managed_GetTerminal (Biddy_Manager *MNG*)

Function Biddy_Managed_GetTerminal returns unmarked edge pointing to the constant node 1.

Description

Terminal node depends on a manager.

Side effects

More info

Internally, use macro biddyTerminal. Macro Biddy_GetTerminal() is defined for use with anonymous manager.

Definition at line 1289 of file biddyMain.c.

5.4.2.16 Biddy_Edge Biddy_Managed_GetConstantZero (Biddy_Manager *MNG*)

Function Biddy_Managed_GetConstantZero returns constant 0.

Description

Constants 0 and 1 depend on a manager. For combination sets, constant 0 coincides with empty set.

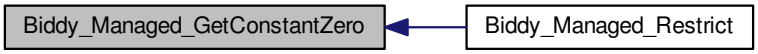
Side effects

More info

Internally, use macro biddyZero. Macro Biddy_GetConstantZero() is defined for use with anonymous manager. Macros Biddy_Managed_GetEmptySet(MNG) and Biddy_GetEmptySet() are defined for manipulation of combination sets.

Definition at line 1319 of file biddyMain.c.

Here is the caller graph for this function:



5.4.2.17 Biddy_Edge Biddy_Managed_GetConstantOne (Biddy_Manager *MNG*)

Function Biddy_Managed_GetConstantOne returns constant 1.

Description

Constants 0 and 1 depend on a manager. For combination sets, constant 1 coincides with universal set.

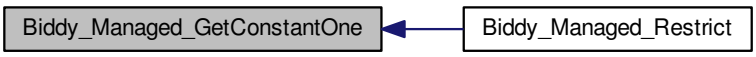
Side effects

More info

Internally, use macro biddyOne. Macro Biddy_GetConstantOne() is defined for use with anonymous manager. Macros Biddy_Managed_GetUniversalSet(MNG) and Biddy_GetUniversalSet() are defined for manipulation of combination sets.

Definition at line 1349 of file biddyMain.c.

Here is the caller graph for this function:



5.4.2.18 Biddy_Edge Biddy_Managed_GetBaseSet (Biddy_Manager *MNG*)

Function Biddy_Managed_GetBaseSet returns set containing only a null combination, i.e. it returns {{}}.

Description

Side effects

More info

Macro Biddy_GetBaseSet() is defined for use with anonymous manager.

Definition at line 1375 of file biddyMain.c.

Here is the caller graph for this function:

5.4.2.19 Biddy_Variable Biddy_Managed_GetVariable (Biddy_Manager *MNG,* Biddy_String *x*)

Function Biddy_Managed_GetVariable returns variable with the given name.

Description

Side effects

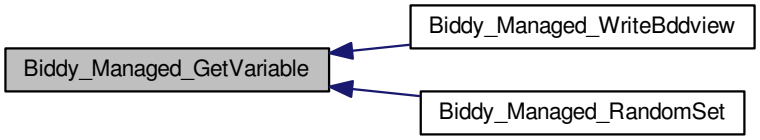
If variable is not found function returns 0!

More info

Macro Biddy_GetVariable(x) is defined for use with anonymous manager.

Definition at line 1410 of file biddyMain.c.

Here is the caller graph for this function:



5.4.2.20 Biddy_Variable Biddy_Managed_GetPrevVariable (Biddy_Manager *MNG,* Biddy_Variable *v*)

Function Biddy_Managed_GetPrevVariable returns previous variable in the global ordering (lower, topmore).

Description

Side effects

More info

Macro Biddy_GetPrevVariable(v) is defined for use with anonymous manager.

Definition at line 1455 of file biddyMain.c.

5.4.2.21 Biddy_Variable Biddy_Managed_GetNextVariable (Biddy_Manager *MNG,* Biddy_Variable *v*)

Function Biddy_Managed_GetNextVariable returns next variable in the global ordering (higher, bottommore).

Description

Side effects

More info

Macro Biddy_GetNextVariable(v) is defined for use with anonymous manager.

Definition at line 1485 of file biddyMain.c.

5.4.2.22 Biddy_Edge Biddy_Managed_GetVariableEdge (Biddy_Manager *MNG,* Biddy_Variable *v*)

Function Biddy_Managed_GetVariableEdge returns variable's edge.

Description

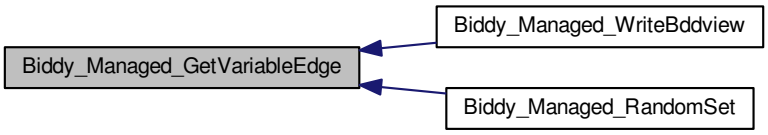
Side effects

More info

Macro Biddy_GetVariableEdge(v) is defined for use with anonymous manager.

Definition at line 1514 of file biddyMain.c.

Here is the caller graph for this function:



5.4.2.23 Biddy_Edge Biddy_Managed_GetElementEdge (Biddy_Manager *MNG,* Biddy_Variable *v*)

Function Biddy_Managed_GetElementEdge returns element's edge.

Description

Side effects

More info

Macro Biddy_GetElementEdge(v) is defined for use with anonymous manager.

Definition at line 1539 of file biddyMain.c.

5.4.2.24 Biddy_String Biddy_Managed_GetVariableName (Biddy_Manager *MNG,* Biddy_Variable *v*)

Function Biddy_Managed_GetVariableName returns the name of a variable.

Description

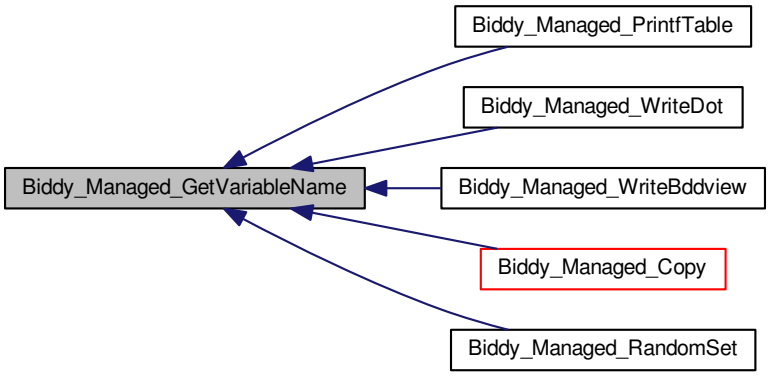
Side effects

More info

Macro Biddy_GetVariableName(v) is defined for use with anonymous manager.

Definition at line 1563 of file biddyMain.c.

Here is the caller graph for this function:



5.4.2.25 Biddy_Edge Biddy_Managed_GetTopVariableEdge (Biddy_Manager *MNG,* Biddy_Edge *f*)

Function Biddy_Managed_GetTopVariableEdge returns variable's edge of top variable.

Description

Side effects

TO DO: For ZBDDs, element edge is sometimes preffered over variable edge.

More info

Macro Biddy_GetTopVariableEdge(f) is defined for use with anonymous manager.

Definition at line 1591 of file biddyMain.c.

5.4.2.26 Biddy_String Biddy_Managed_GetTopVariableName (Biddy_Manager *MNG*, Biddy_Edge *f*)

Function Biddy_Managed_GetTopVariableName returns the name of top variable.

Description

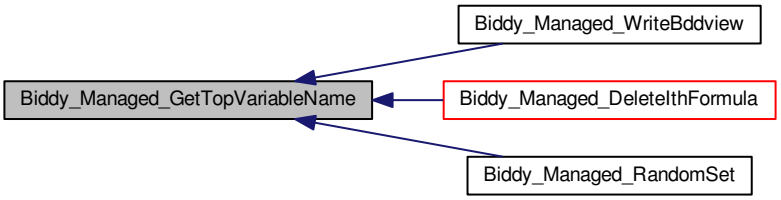
Side effects

More info

Macro Biddy_GetTopVariableName(f) is defined for use with anonymous manager.

Definition at line 1619 of file biddyMain.c.

Here is the caller graph for this function:



5.4.2.27 char Biddy_Managed_GetTopVariableChar (Biddy_Manager *MNG*, Biddy_Edge *f*)

Function Biddy_Managed_GetTopVariableChar returns the first character in the name of top variable.

Description

Side effects

More info

Macro Biddy_GetTopVariableChar(f) is defined for use with anonymous manager.

Definition at line 1647 of file biddyMain.c.

5.4.2.28 void Biddy_Managed_ResetVariablesValue (Biddy_Manager *MNG*)

Function Biddy_Managed_ResetVariablesValue sets all variable's value to biddyZero.

Description

Side effects

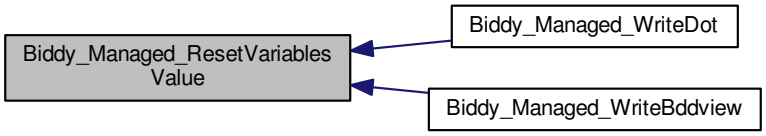
Only active (used) variables are reinitialized.

More info

Macro Biddy_ResetVariablesValue() is defined for use with anonymous manager.

Definition at line 1676 of file biddyMain.c.

Here is the caller graph for this function:



5.4.2.29 void Biddy_Managed_SetVariableValue (Biddy_Manager *MNG,* Biddy_Variable *v,* Biddy_Edge *f*)

Function Biddy_Managed_SetVariableValue sets variable's value.

Description

Side effects

More info

Macro Biddy_SetVariableValue(v,f) is defined for use with anonymous manager.

Definition at line 1705 of file biddyMain.c.

5.4.2.30 Biddy_Boolean Biddy_Managed_IsSmaller (Biddy_Manager *MNG,* Biddy_Variable *fv,* Biddy_Variable *gv*)

Function Biddy_Managed_IsSmaller returns TRUE if the first variable is smaller (= lower = previous = above = topmore).

Description

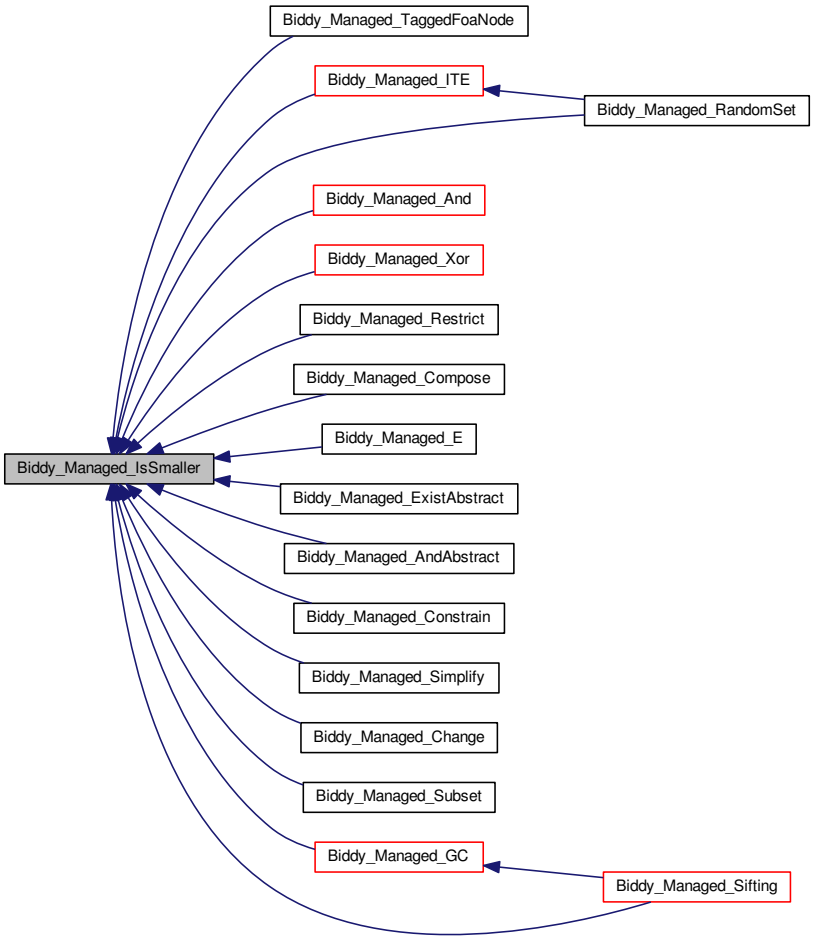
Side effects

More info

Macro Biddy_IsSmaller(fv,gv) is defined for use with anonymous manager.

Definition at line 1731 of file biddyMain.c.

Here is the caller graph for this function:



5.4.2.31 Biddy_Variable Biddy_Managed_FoaVariable (Biddy_Manager *MNG*, Biddy_String *x*, Biddy_Boolean *varelem*)

Function Biddy_Managed_FoaVariable finds variable/element or adds new variable (i.e. Boolean function f = x) and new element (i.e. it creates set {{x}}).

Description

If variable/element already exists, function returns the existing one. If x == NULL then numbered variable/element is added. Numbered variables/elements have only digits in its name. The current number of numbered variables/elements is stored in numnum. If numbered variable/element is requested then function increments numnum and creates a new (non-existing) variable/element. Parameter varelem is used to determine how to adapt the existing BDD base to keep the current formula valid (use varelem = TRUE if formulae represent Boolean functions and varelem = FALSE if they represent combination sets). The ordering of the new variable/element is determined in Biddy_InitMNG. Function always returns variable.

Side effects

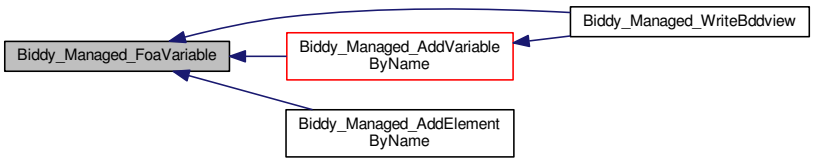
Adding new variable/element may change the meaning of the existing BDDs. Variables and elements are repaired. Moreover, formulae are repaired with regards to the parameter varelem. For OBDDs, it is safe to add new variables/elements if BDDs are used to represent Boolean functions. User should not add numbered variables/elements with some other function. TO DO: Formulae in user's formula tables are not repaired, yet!

More info

Macro Biddy_FoaVariable(x) is defined for use with anonymous manager.

Definition at line 1776 of file biddyMain.c.

Here is the caller graph for this function:



5.4.2.32 Biddy_Edge Biddy_Managed_AddVariableByName (Biddy_Manager *MNG,* Biddy_String *x*)

Function Biddy_Managed_AddVariableByName adds variable.

Description

Biddy_Managed_AddVariableByName uses Biddy_Managed_FoaVariable to find or add variable. Function returns variable edge. If variable already exists, function returns the existing variable edge. For more details see Biddy_↩ Managed_FoaVariable.

Side effects

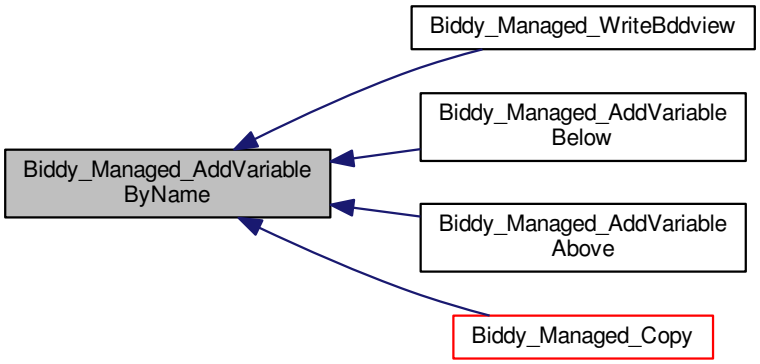
See Biddy_Managed_FoaVariable.

More info

Macro Biddy_AddVariableByName(x) is defined for use with anonymous manager. Macros Biddy_Managed_Add↩ Variable(MNG) and Biddy_AddVariable() are defined for creating numbered variables.

Definition at line 1892 of file biddyMain.c.

Here is the caller graph for this function:



5.4.2.33 Biddy_Edge Biddy_Managed_AddElementByName (Biddy_Manager *MNG*, Biddy_String *x*)

Function Biddy_Managed_AddElementByName adds element.

Description

Biddy_Managed_AddElementByName uses Biddy_Managed_FoaVariable to find or add element. Function returns element edge. If element already exists, function returns the existing element edge. For more details see Biddy_↩ Managed_FoaVariable.

Side effects

See Biddy_Managed_FoaVariable.

More info

Macro Biddy_AddElementByName(x) is defined for use with anonymous manager. Macros Biddy_Managed_Add↩ Element(MNG) and Biddy_AddElement() are defined for creating numbered elements.

Definition at line 1927 of file biddyMain.c.

5.4.2.34 Biddy_Edge Biddy_Managed_AddVariableBelow (Biddy_Manager *MNG*, Biddy_Variable *v*)

Function Biddy_Managed_AddVariableBelow adds a numbered variable.

Description

Biddy_Managed_AddVariableBelow uses Biddy_Managed_AddVariableByName to add numbered variable. Then, the order of the new variable is changed to become immediately below the given variable (below = next = bottom-more in BDD) Function returns variable edge.

Side effects

More info

Macro Biddy_AddVariableBelow(v) is defined for use with anonymous manager.

Definition at line 1959 of file biddyMain.c.

5.4.2.35 Biddy_Edge Biddy_Managed_AddVariableAbove (Biddy_Manager *MNG*, Biddy_Variable *v*)

Function Biddy_Managed_AddVariableAbove adds a numbered variable.

Description

Biddy_Managed_AddVariableAbove uses Biddy_Managed_AddVariableByName to add numbered variable. Then, the order of the new variable is changed to become immediately above the given variable (above = previous = topmore in BDD) Function returns variable edge.

Side effects

More info

Macro Biddy_AddVariableAbove(v) is defined for use with anonymous manager.

Definition at line 2026 of file biddyMain.c.

5.4.2.36 Biddy_Edge Biddy_Managed_TransferMark (Biddy_Manager *MNG*, Biddy_Edge *f*, Biddy_Boolean *mark*, Biddy_Boolean *leftright*)

Function Biddy_Managed_TransferMark returns edge with inverted complement bit iff the second parameter is T↩RUE and normalization rules require this.

Description

It is better to use macro Biddy_InvCond. Parameter leftright is ignored.

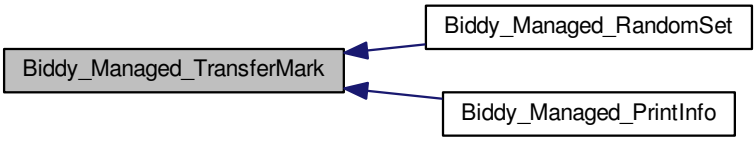
Side effects

More info

Macro Biddy_TransferMark() is defined for use with anonymous manager.

Definition at line 2089 of file biddyMain.c.

Here is the caller graph for this function:



5.4.2.37 Biddy_Edge Biddy_Managed_IncTag (Biddy_Manager *MNG,* Biddy_Edge *f*)

Function Biddy_Managed_IncTag returns edge with an incremented tag.

Description

This function is not used for OBDDs.

Side effects

More info

Macro Biddy_IncTag() is defined for use with anonymous manager.

Definition at line 2117 of file biddyMain.c.

5.4.2.38 Biddy_Edge Biddy_Managed_TaggedFoaNode (Biddy_Manager *MNG,* Biddy_Variable *v,* Biddy_Edge *pf,* Biddy_Edge *pt,* Biddy_Variable *ptag,* Biddy_Boolean *garbageAllowed*)

Function Biddy_Managed_TaggedFoaNode finds or adds new node with the given variable and successors.

Description

If such node already exists, function returns it and does not create the new one. If pf = pt = NULL then new variable is created. This function should not be called directly to add new variables, you must use Biddy_Managed_Foa↩ Variable and Biddy_Managed_AddVariableByName.

Side effects

Parameter ptag is ignored. Using Biddy_Managed_FoaNode you can create node with arbitrary ordering. It is much more safe to use Biddy_Managed_ITE. To enable efficient implementation of sifting the function started with the returned node is not refreshed!

More info

Macro Biddy_Managed_FoaNode(MNG,v,pf,pt,garbageAllowed) is defined for use without tags. Macros Biddy_TaggedFoaNode(v,pf,pt,tag,garbageAllowed) and Biddy_FoaNode(v,pf,pt,garbageAllowed) are defined for use with anonymous manager.

Definition at line 2155 of file biddyMain.c.

5.4.2.39 Biddy_Edge Biddy_Managed_Not (Biddy_Manager *MNG*, Biddy_Edge *f*)

Function Biddy_Managed_Not calculates Boolean function NOT.

Description

It is better to use macro Biddy_Inv.

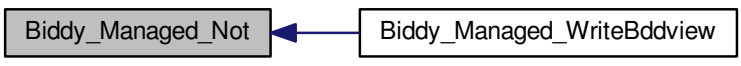
Side effects

More info

Macro Biddy_Not() is defined for use with anonymous manager.

Definition at line 2371 of file biddyMain.c.

Here is the caller graph for this function:



5.4.2.40 Biddy_Edge Biddy_Managed_ITE (Biddy_Manager *MNG*, Biddy_Edge *f*, Biddy_Edge *g*, Biddy_Edge *h*)

Function Biddy_Managed_ITE calculates ITE operation of three Boolean functions.

Description

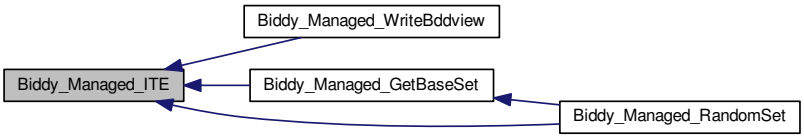
Side Effects

More info

Macro Biddy_ITE(f,g,h) is defined for use with anonymous manager.

Definition at line 2405 of file biddyMain.c.

Here is the caller graph for this function:



5.4.2.41 Biddy_Edge Biddy_Managed_And (Biddy_Manager *MNG*, Biddy_Edge *f*, Biddy_Edge *g*)

Function Biddy_Managed_And calculates Boolean function AND (conjunction).

Description

For combination sets, this function coincides with Intersection.

Side Effects

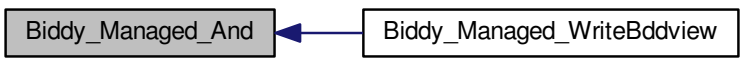
Used by ITE.

More Info

Macro Biddy_And(f,g) is defined for use with anonymous manager. Macros Biddy_Managed_Intersect(MNG,f,g) and Biddy_Intersect(f,g) are defined for manipulation of combination sets.

Definition at line 2699 of file biddyMain.c.

Here is the caller graph for this function:



5.4.2.42 Biddy_Edge Biddy_Managed_Or (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Edge *g*)

Function Biddy_Managed_Or calculates Boolean function OR (disjunction).

Description

For combination sets, this function coincides with Union.

Side Effects

More Info

Macro Biddy_Or(f,g) is defined for use with anonymous manager. Macros Biddy_Managed_Union(MNG,f,g) and Biddy_Union(f,g) are defined for manipulation of combination sets.

Definition at line 2915 of file biddyMain.c.

Here is the caller graph for this function:

5.4.2.43 Biddy_Edge Biddy_Managed_Nand (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Edge *g*)

Function Biddy_Managed_Nand calculates Boolean function NAND (Sheffer).

Description

Side Effects

More Info

Macro Biddy_Nand(f,g) is defined for use with anonymous manager.

Definition at line 2970 of file biddyMain.c.

5.4.2.44 Biddy_Edge Biddy_Managed_Nor (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Edge *g*)

Function Biddy_Managed_Nor calculates Boolean function NOR (Peirce).

Description

Side Effects

More Info

Macro Biddy_Nor(f,g) is defined for use with anonymous manager.

Definition at line 3018 of file biddyMain.c.

5.4.2.45 Biddy_Edge Biddy_Managed_Xor (Biddy_Manager *MNG*, Biddy_Edge *f*, Biddy_Edge *g*)

Function Biddy_Managed_Xor calculates Boolean function XOR.

Description

Side Effects

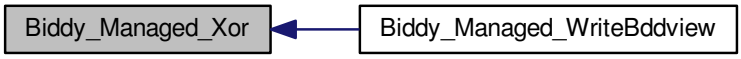
Used by ITE.

More Info

Macro Biddy_Xor(f,g) is defined for use with anonymous manager.

Definition at line 3066 of file biddyMain.c.

Here is the caller graph for this function:



5.4.2.46 Biddy_Edge Biddy_Managed_Xnor (Biddy_Manager *MNG*, Biddy_Edge *f*, Biddy_Edge *g*)

Function Biddy_Managed_Xnor calculates Boolean function XNOR.

Description

Side Effects

More Info

Macro Biddy_Xnor(f,g) is defined for use with anonymous manager.

Definition at line 3292 of file biddyMain.c.

5.4.2.47 Biddy_Edge Biddy_Managed_Leq (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Edge *g*)

Function Biddy_Managed_Leq calculates Boolean implication.

Description

Side Effects

More Info

Macro Biddy_Leq(f,g) is defined for use with anonymous manager.

Definition at line 3339 of file biddyMain.c.

5.4.2.48 Biddy_Edge Biddy_Managed_Gt (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Edge *g*)

Function Biddy_Managed_Gt calculates the negation of Boolean implication.

Description

For combination sets, this function coincides with Diff.

Side Effects

More Info

Macro Biddy_Gt(f,g) is defined for use with anonymous manager. Macros Biddy_Managed_Diff(MNG,f,g) and Biddy_Diff(f,g) are defined for manipulation of combination sets.

Definition at line 3388 of file biddyMain.c.

5.4.2.49 Biddy_Boolean Biddy_Managed_IsLeq (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Edge *g*)

Function Biddy_Managed_IsLeq returns TRUE iff function f is included in function g.

Description

Side Effects

Implemented by calculating full implication which is less efficient as implementation in CUDD.

More Info

Macro Biddy_IsLeq(f,g) is defined for use with anonymous manager.

Definition at line 3437 of file biddyMain.c.

5.4.2.50 Biddy_Edge Biddy_Managed_SubIntersect (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Edge *g*)

Biddy_Managed_SubIntersect calculates a function included in the intersection of f and g.

Description

If the result is not constant 0 then it is a witness that the intersection is not empty. The result should be calculated with as few new nodes as possible, and the result may not be the same as conjunction between functions! If the only result of interest is whether f and g intersect, Biddy_IsLeq should be used (which returns TRUE iff f∗g' == 0).

Side Effects

Implemented by calculating full conjunction which is less efficient as implementation in CUDD.

More Info

Macro Biddy_SubIntersect(f,g) is defined for use with anonymous manager.

Definition at line 3474 of file biddyMain.c.

5.4.2.51 Biddy_Edge Biddy_Managed_Restrict (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Variable *v,* Biddy_Boolean *value*)

Function Biddy_Managed_Restrict calculates a restriction of Boolean function.

Description

Original BDD is not changed. This is not Coudert and Madre's restrict function (use Biddy_Simplify if you need that one).

Side effects

Recursive calls use optimization: F(a=x) == NOT((NOT F)(a=x)).

More info

Macro Biddy_Restrict(f,v,value) is defined for use with anonymous manager.

Definition at line 3508 of file biddyMain.c.

5.4.2.52 Biddy_Edge Biddy_Managed_Compose (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Edge *g,* Biddy_Variable *v*)

Function Biddy_Managed_Compose calculates a composition of two Boolean functions.

Description

Original BDDs are not changed.

Side effects

It uses optimization: F(a=G) == NOT((NOT F)(a=G)).

More info

Macro Biddy_Compose(f,g,v) is defined for use with anonymous manager.

Definition at line 3599 of file biddyMain.c.

5.4.2.53 Biddy_Edge Biddy_Managed_E (Biddy_Manager *MNG*, Biddy_Edge *f*, Biddy_Variable *v*)

Function Biddy_Managed_E calculates an existential quantification of Boolean function.

Description

Original BDD is not changed.

Side effects

Be careful: ExA F != NOT(ExA (NOT F)). Counterexample: Exb (AND (NOT a) b c).

More info

Macro Biddy_E(f,v) is defined for use with anonymous manager.

Definition at line 3692 of file biddyMain.c.

5.4.2.54 Biddy_Edge Biddy_Managed_A (Biddy_Manager *MNG*, Biddy_Edge *f*, Biddy_Variable *v*)

Function Biddy_Managed_A calculates an universal quantification of Boolean function.

Description

Original BDD is not changed.

Side effects

Be careful: AxA F != NOT(AxA (NOT F)). Counterexample: Axb (AND (NOT a) b c).

More info

Macro Biddy_A(f,v) is defined for use with anonymous manager.

Definition at line 3785 of file biddyMain.c.

5.4.2.55 Biddy_Boolean Biddy_Managed_IsVariableDependent (Biddy_Manager *MNG*, Biddy_Edge *f*, Biddy_Variable *v*)

Function Biddy_Managed_IsVariableDependent returns TRUE iff variable is dependent on others in a function.

Description

A variable is dependent on others in a function iff universal quantification of this variable returns constant FALSE.

Side effects

Implemented by calculating full universal quantification which is less efficient as direct implementation in CUDD.

More info

Macro Biddy_IsVariableDependent(f,v) is defined for use with anonymous manager.

Definition at line 3842 of file biddyMain.c.

5.4.2.56 Biddy_Edge Biddy_Managed_ExistAbstract (Biddy_Manager *MNG*, Biddy_Edge *f*, Biddy_Edge *cube*)

Function Biddy_Managed_ExistAbstract existentially abstracts all the variables in cube from f.

Description

Original BDD is not changed.

Side effects

More info

Macro Biddy_ExistAbstract(f,cube) is defined for use with anonymous manager.

Definition at line 3873 of file biddyMain.c.

5.4.2.57 Biddy_Edge Biddy_Managed_UnivAbstract (Biddy_Manager *MNG*, Biddy_Edge *f*, Biddy_Edge *cube*)

Function Biddy_Managed_UnivAbstract universally abstracts all the variables in cube from f.

Description

Original BDD is not changed.

Side effects

More info

Macro Biddy_UnivAbstract(f,cube) is defined for use with anonymous manager.

Definition at line 3978 of file biddyMain.c.

5.4.2.58 Biddy_Edge Biddy_Managed_AndAbstract (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Edge *g,* Biddy_Edge *cube*)

Function Biddy_Managed_AndAbstract calculates the AND of two BDDs and simultaneously (existentially) abstracts the variables in cube.

Description

Side effects

More info

Macro Biddy_AndAbstract(f,g,cube) is defined for use with anonymous manager.

Definition at line 4033 of file biddyMain.c.

5.4.2.59 Biddy_Edge Biddy_Managed_Constrain (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Edge *c*)

Function Biddy_Managed_Constrain calculates Coudert and Madre's constrain function.

Description

Coudert and Madre's constrain function is also called a generalized cofactor of function f with respect to function c.

Side effects

Cache table is not implemented, yet.

More info

Macro Biddy_Constrain(f,c) is defined for use with anonymous manager.

Definition at line 4192 of file biddyMain.c.

5.4.2.60 Biddy_Edge Biddy_Managed_Simplify (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Edge *c*)

Function Biddy_Managed_Simplify calculates Coudert and Madre's restrict function.

Description

Coudert and Madre's restrict function tries to simplify function f by restricting it to the domain covered by function c. No checks are done to see if the result is actually smaller than the input.

Side effects

Cache table is not implemented, yet.

More info

Macro Biddy_Simplify(f,c) is defined for use with anonymous manager.

Definition at line 4285 of file biddyMain.c.

5.4.2.61 Biddy_Edge Biddy_Managed_Support (Biddy_Manager *MNG*, Biddy_Edge *f*)

Function Biddy_Managed_Support calculates a product of all dependent variables.

Description

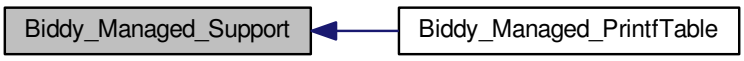
Side effects

More info

Macro Biddy_Support(f) is defined for use with anonymous manager.

Definition at line 4378 of file biddyMain.c.

Here is the caller graph for this function:



5.4.2.62 Biddy_Edge Biddy_Managed_Replace (Biddy_Manager *MNG*, Biddy_Edge *f*)

Function Biddy_Managed_Replace calculates BDD with one or more variables replaced.

Description

Original BDD is not changed. Replacing is controlled by variable's values (which are edges!). Use Biddy_↩ Managed_ResetVariablesValue and Biddy_Managed_SetVariableValue to prepare replacing. Current and new variables should be disjoint sets.

Side effects

Cache table is not implemented, yet.

More info

Macro Biddy_Replace(f) is defined for use with anonymous manager.

Definition at line 4448 of file biddyMain.c.

5.4.2.63 Biddy_Edge Biddy_Managed_Change (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Variable *v*)

Function Biddy_Managed_Change change the form of the given variable (positive literal becomes negative and vice versa).

Description

Side effects

More info

Macro Biddy_Change() is defined for use with anonymous manager.

Definition at line 4516 of file biddyMain.c.

5.4.2.64 Biddy_Edge Biddy_Managed_Subset (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Variable *v,* Biddy_Boolean *value*)

Function Biddy_Managed_Subset calculates a division of Boolean function with a literal.

Description

Original BDD is not changed. For combination sets, this function coincides with Subset0 and Subset1.

Side effects

More info

Macro Biddy_Subset(f,v,value) is defined for use with anonymous manager. Macros Biddy_Managed_Subset0(↩ MNG,f,v), Biddy_Subset0(f,v), Biddy_Managed_Subset1(MNG,f,v), and Biddy_Subset1(f,v) are defined for manipulation of combination sets.

Definition at line 4585 of file biddyMain.c.

5.4.2.65 Biddy_Boolean Biddy_Managed_IsOK (Biddy_Manager *MNG,* Biddy_Edge *f*)

Function Biddy_Managed_IsOK returns TRUE iff given node is not obsolete.

Description

This is needed for implementation of user caches.

Side effects

More info

Macro BiddyIsOK(f) is defined for debugging. It will check more properties and not only the expiry value. Macro Biddy_IsOK(f) is defined for use with anonymous manager.

Definition at line 4691 of file biddyMain.c.

5.4.2.66 void Biddy_Managed_GC (Biddy_Manager *MNG*, Biddy_Variable *target*, Biddy_Boolean *purge*, Biddy_Boolean *total*)

Function Biddy_Managed_GC performs garbage collection.

Description

All obsolete nodes are deleted. Parameter target is used during sifting. If parameter total is true than all obsolete nodes are deleted, otherwise nodes are deleted only if there are enough obsolete nodes. Nodes from deleted non-obsolete formulae are immediately removed only if parameter purge is true (this should not be used during the automatic garbage collection), otherwise these nodes only become fresh.

Side effects

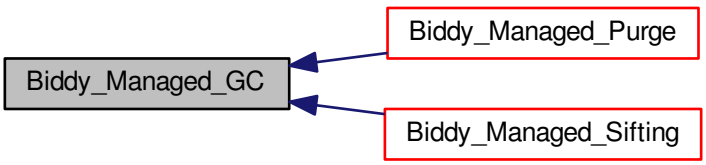
The first element of each chain in a node table should have a special value for its 'prev' element to allow tricky but efficient deleting. Moreover, 'prev' and 'next' should be the first and the second element in the structure BiddyNode, respectively. Garbage collecion is reported by biddyNodeTable.garbage only if some bad nodes are purged!

More info

Macro Biddy_GC(target,purge,total) is defined for use with anonymous manager. Macros Biddy_Managed_Auto↩ GC(MNG) and Biddy_AutoGC() are useful variants with target = 0, purge = FALSE, and total = FALSE.

Definition at line 4731 of file biddyMain.c.

Here is the caller graph for this function:



5.4.2.67 void Biddy_Managed_Clean (Biddy_Manager *MNG*)

Function Biddy_Managed_Clean performs cleaning.

Description

Discard all nodes which are not preserved or which are not preserved anymore. Obsolete nodes are not immediately removed, they will be removed during the first garbage collection.

Side effects

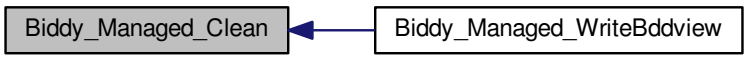
Tag deleted is not considered and thus no fortified node and no prolonged node is discarded. Constants and variables should be fortified! Restoring elements is not implemented, yet.

More info

Macro Biddy_Clean() is defined for use with anonymous manager.

Definition at line 5074 of file biddyMain.c.

Here is the caller graph for this function:



5.4.2.68 void Biddy_Managed_Purge (Biddy_Manager *MNG*)

Function Biddy_Managed_Purge immediately removes all nodes which were not preserved or which are not preserved anymore.

Description

All fresh and obsolete nodes are immediatelly removed. Moreover, nodes from deleted prolonged formulae and nodes from deleted fortified formulae are removed if they are not needed by other formulae. Call to Biddy_Purge does not count as clearing and thus all preserved formulae remains preserved for the same number of clearings.

Side effects

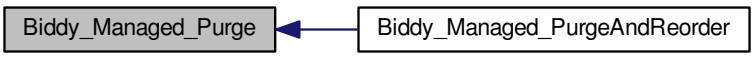
Removes all fresh nodes!

More info

Macro Biddy_Purge(f) is defined for use with anonymous manager.

Definition at line 5108 of file biddyMain.c.

Here is the caller graph for this function:



5.4.2.69 void Biddy_Managed_PurgeAndReorder (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Boolean *converge*)

Function Biddy_Managed_PurgeAndReorder immediately removes non-preserved nodes and triggers reordering on function.

Description

All obsolete nodes are immediatelly removed. Moreover, nodes from deleted prolonged formulae and nodes from deleted fortified formulae are removed if they are not needed by other formulae. If BDD is given (f != NULL), reordering on function is used. Otherwise (f == NULL) global reordering is used. Call to Biddy_PurgeAndReorder does not count as clearing and thus all preserved formulae remains preserved for the same number of clearings.

Side effects

Removes all fresh nodes.

More info

Macro Biddy_PurgeAndReorder(f) is defined for use with anonymous manager.

Definition at line 5144 of file biddyMain.c.

5.4.2.70 void Biddy_Managed_Refresh (Biddy_Manager *MNG,* Biddy_Edge *f*)

Function Biddy_Managed_Refresh refreshes top node in a given function.

Description

This is an external variant of internal macro BiddyRefresh This is needed for implementing user caches.

Side effects

More info

Macro Biddy_Refresh(f) is defined for use with anonymous manager.

Definition at line 5173 of file biddyMain.c.

5.4.2.71 void Biddy_Managed_AddCache (Biddy_Manager *MNG,* Biddy_GCFunction *gc*)

Function Biddy_Managed_AddCache adds cache to the end of Cache list.

Description

If Cache list does not exist, function creates it.

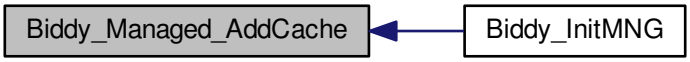
Side effects

More info

Macro Biddy_AddCache(gc) is defined for use with anonymous manager.

Definition at line 5199 of file biddyMain.c.

Here is the caller graph for this function:



5.4.2.72 unsigned int Biddy_Managed_AddFormula (Biddy_Manager *MNG*, Biddy_String *x*, Biddy_Edge *f*, int *c*)

Function Biddy_Managed_AddFormula adds formula to Formula table.

Description

Nodes of the given BDD will be preserved for the given number of clearings. If (x != NULL) then formula is accessible by its name. If (c == -1) then formula is not preserved. If (c == 0) then formula is persistently preserved and you have to use Biddy_DeleteFormula to remove its nodes. There are two macros defined to simplify formulae management. Macro Biddy_Managed_AddTmpFormula(mng,bdd,c) is defined as Biddy_Managed_AddFormula(mng,N↩ULL,bdd,c) and macro Biddy_Managed_AddPersistentFormula(mng,name,bdd) is defined as Biddy_Managed_↩AddFormula(mng,name,bdd,0).

Side effects

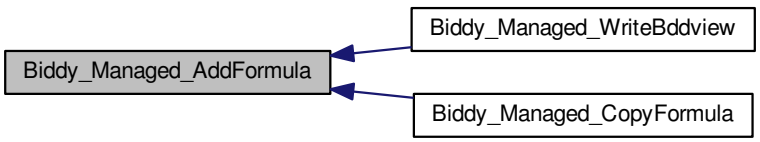
Function is prolonged or fortified. Formulae with name are ordered by name. If formula with the same name already exists, it will be overwritten (preserved and persistently preserved formulae, too)!

More info

Macros Biddy_AddFormula(x,f), Biddy_AddTmpFormula(f,c), and Biddy_AddPersistentFormula(x,f) are defined for use with anonymous manager.

Definition at line 5260 of file biddyMain.c.

Here is the caller graph for this function:



5.4.2.73 Biddy_Boolean Biddy_Managed_FindFormula (Biddy_Manager *MNG*, Biddy_String *x*, Biddy_Edge * *f*)

Function Biddy_Managed_FindFormula find formula in Formula table.

Description

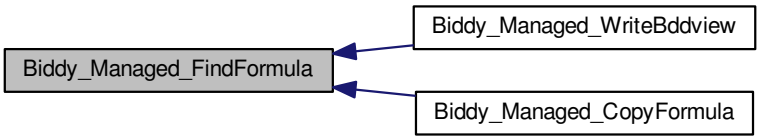
Side effects

More info

Macro Biddy_FindFormula(x,f) is defined for use with anonymous manager.

Definition at line 5453 of file biddyMain.c.

Here is the caller graph for this function:



5.4.2.74 Biddy_Boolean Biddy_Managed_DeleteFormula (Biddy_Manager *MNG*, Biddy_String *x*)

Function Biddy_Managed_DeleteFormula delete formula from Formula table.

Description

Formula is labelled but not immediately removed. Nodes of the given formula are not immediately removed.

Side effects

Formula is not accessible by its name anymore. Formulae representing constants and variables will not be deleted.

More info

Macro Biddy_DeleteFormula(x) is defined for use with anonymous manager.

Definition at line 5554 of file biddyMain.c.

5.4.2.75 Biddy_Boolean Biddy_Managed_DeleteIthFormula (Biddy_Manager *MNG*, unsigned int *i*)

Function Biddy_Managed_DeleteIthFormula deletes formula from the table.

Description

Formula is labelled but not immediately removed. Nodes of the given formula are not immediately removed.

Side effects

Formula is not accessible by its name anymore. The first two formulae ("0" and "1") will not be deleted. Formulae representing variables will not be deleted.

More info

Macro Biddy_DeleteIthFormula(x) is defined for use with anonymous manager.

Definition at line 5617 of file biddyMain.c.

Here is the caller graph for this function:

5.4.2.76 Biddy_Edge Biddy_Managed_GetIthFormula (Biddy_Manager *MNG*, unsigned int *i*)

Function Biddy_Managed_GetIthFormula returns ith formula in a Formula table.

Description

Return biddyNull if ith formulae does not exist.

Side effects

After addding new formula the index of others may change!

More info

Macro Biddy_GetIthFormula(i) is defined for use with anonymous manager.

Definition at line 5677 of file biddyMain.c.

5.4.2.77 Biddy_String Biddy_Managed_GetIthFormulaName (Biddy_Manager *MNG*, unsigned int *i*)

Function Biddy_Managed_GetIthFormulaName returns name of the ith formula in a Formula table.

Description

Return NULL if ith formulae does not exist.

Side effects

After addding new formula the index of others may change!

More info

Macro Biddy_GetIthFormulaName(i) is defined for use with anonymous manager.

Definition at line 5708 of file biddyMain.c.

5.4.2.78 Biddy_Variable Biddy_Managed_SwapWithHigher (Biddy_Manager *MNG,* Biddy_Variable *v*)

Function Biddy_Managed_SwapWithHigher swaps two adjacent variables.

Description

Higher (greater) variable is the bottommore one! The highest element is constant "1". Constant '1' has global ordering numUsedVariables (not smaller than anyone). Global ordering is the number of zeros in corresponding line of orderingTable.

Side effects

All obsolete nodes will be removed.

More info

Macro Biddy_SwapWithHigher(v) is defined for use with anonymous manager.

Definition at line 5749 of file biddyMain.c.

5.4.2.79 Biddy_Variable Biddy_Managed_SwapWithLower (Biddy_Manager *MNG,* Biddy_Variable *v*)

Function Biddy_Managed_SwapWithLower swaps two adjacent variables.

Description

Lower (smaller) variable is the topmore one! The lowest (topmost) element is not fixed. Topmost variable has global ordering 1 (smaller than all except itself). Global ordering is the number of zeros in corresponding line of orderingTable.

Side effects

All obsolete nodes will be removed.

More info

Macro Biddy_SwapWithLower(v) is defined for use with anonymous manager.

Definition at line 5787 of file biddyMain.c.

5.4.2.80 Biddy_Boolean Biddy_Managed_Sifting (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Boolean *converge*)

Function Biddy_Managed_Sifting reorders variables to minimize node number for the whole system (if f = NULL) or for the given function (if f != NULL) using Rudell's sifting algorithm.

Description

Variables are reordered globally. All obsolete nodes will be removed.

Side effects

More info

Macro Biddy_Sifting(f) is defined for use with anonymous manager.

Definition at line 5824 of file biddyMain.c.

Here is the caller graph for this function:

5.4.2.81 Biddy_Edge Biddy_Managed_Random (Biddy_Manager *MNG,* Biddy_Edge *support,* double *r*)

Function Biddy_Managed_Random generates a random BDD.

Description

The represented Boolean function depends on the variables given with parameter support whilst the parameter r determines the ratio between the number of function's minterms and the number of all possible minterms. Parameter support is a product of positive varaiables.

Side effects

Parameter r must be a number from [0,1]. Otherwise, function returns biddyNull.

More info

Macro Biddy_Random(support,r) is defined for use with anonymous manager.

Definition at line 6918 of file biddyMain.c.

5.4.2.82 Biddy_Edge Biddy_Managed_RandomSet (Biddy_Manager *MNG,* Biddy_Edge *unit,* double *r*)

Function Biddy_Managed_RandomSet generates a random BDD.

Description

The represented set is a random combination set determined by the parameter unit whilst the parameter r determines the ratio between the number of set's subsets and the number of all possible subsets. Parameter set is a set containing only one subset which consist of all elements, i.e. it is a set {{x1,x2,...,xn}}.

Side effects

Parameter r must be a number from [0,1]. Otherwise, function returns biddyNull.

More info

Macro Biddy_RandomSet(unit,r) is defined for use with anonymous manager.

Definition at line 7030 of file biddyMain.c.

5.5 biddyMainGDD.c File Reference

File biddyMainGDD.c contains main functions for representation and manipulation of boolean functions with various types of Binary Decision Diagrams (GDD = general decision diagrams).

```
#include "biddyInt.h"
```

Functions

- void Biddy_InitMNG (Biddy_Manager *mng, int gddtype)

  Function Biddy_InitMNG initialize a manager.
- void Biddy_ExitMNG (Biddy_Manager *mng)

  Function Biddy_ExitMNG deletes a manager.
- Biddy_String Biddy_About ()

  Function Biddy_About reports version of Biddy package.
- int Biddy_Managed_GetManagerType (Biddy_Manager MNG)

  Function Biddy_Managed_GetManagerType reports BDD type used in the manager.
- void Biddy_Managed_SetManagerParameters (Biddy_Manager MNG, float gcr, float gcrF, float gcrX, float rr, float rrF, float rrX, float st, float fst, float cst, float fcst)

  Function Biddy_Managed_SetManagerParameters set modifiable parameters.
- Biddy_Edge Biddy_GetThen (Biddy_Edge fun)

Function Biddy_GetThen returns THEN successor.

- Biddy_Edge Biddy_GetElse (Biddy_Edge fun)

  Function Biddy_GetElse returns ELSE successor.
- Biddy_Variable Biddy_GetTopVariable (Biddy_Edge fun)

  Function Biddy_GetTopVariable returns the top variable.
- Biddy_Boolean Biddy_Managed_IsEqv (Biddy_Manager MNG1, Biddy_Edge f1, Biddy_Manager MNG2, Biddy_Edge f2)

  Function Biddy_Managed_IsEqv returns TRUE iff two BDDs are equal.
- void Biddy_Managed_SelectNode (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_SelectNode selects the top node of the given function.
- void Biddy_Managed_DeselectNode (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_DeselectNode deselects the top node of the given function.
- Biddy_Boolean Biddy_Managed_IsSelected (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_IsSelected returns TRUE iff the top node of the given function is selected.
- void Biddy_Managed_SelectFunction (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_SelectFunction recursively selects all nodes of a given function.
- void Biddy_Managed_DeselectAll (Biddy_Manager MNG)

  Function Biddy_Managed_DeselectAll deselects all nodes.
- Biddy_Edge Biddy_Managed_GetTerminal (Biddy_Manager MNG)

  Function Biddy_Managed_GetTerminal returns unmarked and untagged edge pointing to the constant node 1.
- Biddy_Edge Biddy_Managed_GetConstantZero (Biddy_Manager MNG)

  Function Biddy_Managed_GetConstantZero returns constant 0.
- Biddy_Edge Biddy_Managed_GetConstantOne (Biddy_Manager MNG)

  Function Biddy_Managed_GetConstantOne returns constant 1.
- Biddy_Edge Biddy_Managed_GetBaseSet (Biddy_Manager MNG)

  Function Biddy_Managed_GetBaseSet returns set containing only a null combination, i.e. it returns {{}}.
- Biddy_Variable Biddy_Managed_GetVariable (Biddy_Manager MNG, Biddy_String x)

  Function Biddy_Managed_GetVariable returns variable with the given name.
- Biddy_Variable Biddy_Managed_GetPrevVariable (Biddy_Manager MNG, Biddy_Variable v)

  Function Biddy_Managed_GetPrevVariable returns previous variable in the global ordering (lower, topmore).
- Biddy_Variable Biddy_Managed_GetNextVariable (Biddy_Manager MNG, Biddy_Variable v)

  Function Biddy_Managed_GetNextVariable returns next variable in the global ordering (higher, bottommore).
- Biddy_Edge Biddy_Managed_GetVariableEdge (Biddy_Manager MNG, Biddy_Variable v)

  Function Biddy_Managed_GetVariableEdge returns variable's edge.
- Biddy_Edge Biddy_Managed_GetElementEdge (Biddy_Manager MNG, Biddy_Variable v)

  Function Biddy_Managed_GetElementEdge returns element's edge.
- Biddy_String Biddy_Managed_GetVariableName (Biddy_Manager MNG, Biddy_Variable v)

  Function Biddy_Managed_GetVariableName returns the name of a variable.
- Biddy_Edge Biddy_Managed_GetTopVariableEdge (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_GetTopVariableEdge returns variable's edge of top variable.
- Biddy_String Biddy_Managed_GetTopVariableName (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_GetTopVariableName returns the name of top variable.
- char Biddy_Managed_GetTopVariableChar (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_GetTopVariableChar returns the first character in the name of top variable.
- void Biddy_Managed_ResetVariablesValue (Biddy_Manager MNG)

  Function Biddy_Managed_ResetVariablesValue sets all variable's value to biddyZero.
- void Biddy_Managed_SetVariableValue (Biddy_Manager MNG, Biddy_Variable v, Biddy_Edge f)

  Function Biddy_Managed_SetVariableValue sets variable's value.
- Biddy_Boolean Biddy_Managed_IsSmaller (Biddy_Manager MNG, Biddy_Variable fv, Biddy_Variable gv)

  Function Biddy_Managed_IsSmaller returns TRUE if the first variable is smaller (= lower = previous = above = topmore).

- Biddy_Variable Biddy_Managed_FoaVariable (Biddy_Manager MNG, Biddy_String x, Biddy_Boolean varelem)

  Function Biddy_Managed_FoaVariable finds variable/element or adds new variable (i.e. Boolean function f = x) and new element (i.e. it creates set {{x}}).
- Biddy_Edge Biddy_Managed_AddVariableByName (Biddy_Manager MNG, Biddy_String x)

  Function Biddy_Managed_AddVariableByName adds variable.
- Biddy_Edge Biddy_Managed_AddElementByName (Biddy_Manager MNG, Biddy_String x)

  Function Biddy_Managed_AddElementByName adds element.
- Biddy_Edge Biddy_Managed_AddVariableBelow (Biddy_Manager MNG, Biddy_Variable v)

  Function Biddy_Managed_AddVariableBelow adds a numbered variable.
- Biddy_Edge Biddy_Managed_AddVariableAbove (Biddy_Manager MNG, Biddy_Variable v)

  Function Biddy_Managed_AddVariableAbove adds a numbered variable.
- Biddy_Edge Biddy_Managed_TransferMark (Biddy_Manager MNG, Biddy_Edge f, Biddy_Boolean mark, Biddy_Boolean leftright)

  Function Biddy_Managed_TransferMark returns edge with inverted complement bit iff the second parameter is TRUE and normalization rules require this.
- Biddy_Edge Biddy_Managed_IncTag (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_IncTag returns edge with an incremented tag.
- Biddy_Edge Biddy_Managed_TaggedFoaNode (Biddy_Manager MNG, Biddy_Variable v, Biddy_Edge pf, Biddy_Edge pt, Biddy_Variable ptag, Biddy_Boolean garbageAllowed)

  Function Biddy_Managed_TaggedFoaNode finds or adds new node with the given variable and successors.
- Biddy_Edge Biddy_Managed_Not (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_Not calculates Boolean function NOT.
- Biddy_Edge Biddy_Managed_ITE (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge g, Biddy_Edge h)

  Function Biddy_Managed_ITE calculates ITE operation of three Boolean functions.
- Biddy_Edge Biddy_Managed_And (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge g)

  Function Biddy_Managed_And calculates Boolean function AND (conjunction).
- Biddy_Edge Biddy_Managed_Or (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge g)

  Function Biddy_Managed_Or calculates Boolean function OR (disjunction).
- Biddy_Edge Biddy_Managed_Nand (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge g)

  Function Biddy_Managed_Nand calculates Boolean function NAND (Sheffer).
- Biddy_Edge Biddy_Managed_Nor (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge g)

  Function Biddy_Managed_Nor calculates Boolean function NOR (Peirce).
- Biddy_Edge Biddy_Managed_Xor (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge g)

  Function Biddy_Managed_Xor calculates Boolean function XOR.
- Biddy_Edge Biddy_Managed_Xnor (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge g)

  Function Biddy_Managed_Xnor calculates Boolean function XNOR.
- Biddy_Edge Biddy_Managed_Leq (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge g)

  Function Biddy_Managed_Leq calculates Boolean implication.
- Biddy_Edge Biddy_Managed_Gt (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge g)

  Function Biddy_Managed_Gt calculates the negation of Boolean implication.
- Biddy_Boolean Biddy_Managed_IsLeq (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge g)

  Function Biddy_Managed_IsLeq returns TRUE iff function f is included in function g.
- Biddy_Edge Biddy_Managed_Restrict (Biddy_Manager MNG, Biddy_Edge f, Biddy_Variable v, Biddy_Boolean value)

  Function Biddy_Managed_Restrict calculates a restriction of Boolean function.
- Biddy_Edge Biddy_Managed_Compose (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge g, Biddy_Variable v)

  Function Biddy_Managed_Compose calculates a composition of two Boolean functions.
- Biddy_Edge Biddy_Managed_E (Biddy_Manager MNG, Biddy_Edge f, Biddy_Variable v)

  Function Biddy_Managed_E calculates an existential quantification of Boolean function.
- Biddy_Edge Biddy_Managed_A (Biddy_Manager MNG, Biddy_Edge f, Biddy_Variable v)

Function Biddy_Managed_A calculates an universal quantification of Boolean function.

- Biddy_Boolean Biddy_Managed_IsVariableDependent (Biddy_Manager MNG, Biddy_Edge f, Biddy_Variable v)

  Function Biddy_Managed_IsVariableDependent returns TRUE iff variable is dependent on others in a function.
- Biddy_Edge Biddy_Managed_ExistAbstract (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge cube)

  Function Biddy_Managed_ExistAbstract existentially abstracts all the variables in cube from f.
- Biddy_Edge Biddy_Managed_UnivAbstract (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge cube)

  Function Biddy_Managed_UnivAbstract universally abstracts all the variables in cube from f.
- Biddy_Edge Biddy_Managed_AndAbstract (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge g, Biddy_↩ Edge cube)

  Function Biddy_Managed_AndAbstract calculates the AND of two BDDs and simultaneously (existentially) abstracts the variables in cube.
- Biddy_Edge Biddy_Managed_Constrain (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge c)

  Function Biddy_Managed_Constrain calculates Coudert and Madre's constrain function.
- Biddy_Edge Biddy_Managed_Simplify (Biddy_Manager MNG, Biddy_Edge f, Biddy_Edge c)

  Function Biddy_Managed_Simplify calculates Coudert and Madre's restrict function.
- Biddy_Edge Biddy_Managed_Support (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_Support calculates a product of all dependent variables (OBDD and TZBDD) or the combination set containing a subset which includes all dependent variables (ZBDD).
- Biddy_Edge Biddy_Managed_Replace (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_Replace calculates Boolean function with one or more variables replaced.
- Biddy_Edge Biddy_Managed_Change (Biddy_Manager MNG, Biddy_Edge f, Biddy_Variable v)

  Function Biddy_Managed_Change change the form of the given variable (positive literal becomes negative and vice versa).
- Biddy_Edge Biddy_Managed_Subset (Biddy_Manager MNG, Biddy_Edge f, Biddy_Variable v, Biddy_↩ Boolean value)

  Function Biddy_Managed_Subset calculates a division of Boolean function with a literal.
- Biddy_Boolean Biddy_Managed_IsOK (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_IsOK returns TRUE iff given node is not obsolete.
- void Biddy_Managed_GC (Biddy_Manager MNG, Biddy_Variable target, Biddy_Boolean purge, Biddy_↩ Boolean total)

  Function Biddy_Managed_GC performs garbage collection.
- void Biddy_Managed_Clean (Biddy_Manager MNG)

  Function Biddy_Managed_Clean performs cleaning.
- void Biddy_Managed_Purge (Biddy_Manager MNG)

  Function Biddy_Managed_Purge immediately removes all nodes which were not preserved or which are not preserved anymore.
- void Biddy_Managed_PurgeAndReorder (Biddy_Manager MNG, Biddy_Edge f, Biddy_Boolean converge)

  Function Biddy_Managed_PurgeAndReorder immediately removes non-preserved nodes and triggers reordering on function.
- void Biddy_Managed_Refresh (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_Refresh refreshes top node in a given function.
- void Biddy_Managed_AddCache (Biddy_Manager MNG, Biddy_GCFunction gc)

  Function Biddy_Managed_AddCache adds cache to the end of Cache list.
- unsigned int Biddy_Managed_AddFormula (Biddy_Manager MNG, Biddy_String x, Biddy_Edge f, int c)

  Function Biddy_Managed_AddFormula adds formula to Formula table.
- Biddy_Boolean Biddy_Managed_FindFormula (Biddy_Manager MNG, Biddy_String x, Biddy_Edge *f)

  Function Biddy_Managed_FindFormula find formula in Formula table.
- Biddy_Boolean Biddy_Managed_DeleteFormula (Biddy_Manager MNG, Biddy_String x)

  Function Biddy_Managed_DeleteFormula delete formula from Formula table.
- Biddy_Boolean Biddy_Managed_DeleteIthFormula (Biddy_Manager MNG, unsigned int i)

  Function Biddy_Managed_DeleteIthFormula deletes formula from the table.

- Biddy_Edge Biddy_Managed_GetIthFormula (Biddy_Manager MNG, unsigned int i)

  Function Biddy_Managed_GetIthFormula returns ith formula in a Formula table.
- Biddy_String Biddy_Managed_GetIthFormulaName (Biddy_Manager MNG, unsigned int i)

  Function Biddy_Managed_GetIthFormulaName returns name of the ith formula in a Formula table.
- Biddy_Variable Biddy_Managed_SwapWithHigher (Biddy_Manager MNG, Biddy_Variable v)

  Function Biddy_Managed_SwapWithHigher swaps two adjacent variables.
- Biddy_Variable Biddy_Managed_SwapWithLower (Biddy_Manager MNG, Biddy_Variable v)

  Function Biddy_Managed_SwapWithLower swaps two adjacent variables.
- Biddy_Boolean Biddy_Managed_Sifting (Biddy_Manager MNG, Biddy_Edge f, Biddy_Boolean converge)

  Function Biddy_Managed_Sifting reorders variables to minimize node number for the whole system (if f = NULL) or for the given function (if f != NULL) using Rudell's sifting algorithm.
- Biddy_Edge Biddy_Managed_Copy (Biddy_Manager MNG1, Biddy_Manager MNG2, Biddy_Edge f)

  Function Biddy_Managed_Copy copies a graph from one manager to another manager which can use the same or different BDD type.

  Description

  The function takes a graph from one manager and creates the same graph in another manager. If the managers do not use the same BDD type then a graph is converted. The resulting graph will represent the same Boolean function assuming the domain from the target manager. If f = biddyZero then only the domain is copied.

  Side effects

  If source and target manager are the same then function does nothing. The variable ordering of created BDD is adapted to the target manager.

  More info

  Macro Biddy_Copy(MNG2,f) is defined for use with anonymous manager.
- void Biddy_Managed_CopyFormula (Biddy_Manager MNG1, Biddy_Manager MNG2, Biddy_String x)

  Function Biddy_Managed_CopyFormula uses Biddy_Managed_Copy to copy a graph from one manager to another manager which can use the same or different BDD type.

  Description

  See Biddy_Managed_Copy.

  Side effects

  If source and target manager are the same then function does nothing. The variable ordering of created BDD is adapted to the target manager. The created formula is not preserved.

  More info

  Macro Biddy_CopyFormula(MNG2,x) is defined for use with anonymous manager.
- Biddy_Boolean Biddy_Managed_Eval (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_Eval returns the value of a Boolean function for a given variable assignment.

  Description

  Side effects

  More info

  Macro Biddy_Eval(f) is defined for use with anonymous manager.
- Biddy_Edge Biddy_Managed_Random (Biddy_Manager MNG, Biddy_Edge support, double r)

  Function Biddy_Managed_Random generates a random BDD.
- Biddy_Edge Biddy_Managed_RandomSet (Biddy_Manager MNG, Biddy_Edge unit, double r)

  Function Biddy_Managed_RandomSet generates a random BDD.

5.5.1 Detailed Description

File biddyMainGDD.c contains main functions for representation and manipulation of boolean functions with various types of Binary Decision Diagrams (GDD = general decision diagrams).

Description

```
PackageName [Biddy]
Synopsis    [Biddy provides data structures and algorithms for the
             representation and manipulation of Boolean functions with
             ROBDDs, 0-sup-BDDs, and TZBDDs. A hash table is used for quick
     search of nodes. Complement edges decreases the number of
     nodes. An automatic garbage collection with a system age is
             implemented. Variable swapping and sifting are implemented.]

FileName    [biddyMainGDD.c]
Revision    [$Revision: 254 $]
Date        [$Date: 2017-03-20 15:03:19 +0100 (pon, 20 mar 2017) $]
Authors     [Robert Meolic (robert.meolic@um.si)]
```

Copyright

More info

See also: biddy.h, biddyInt.h

5.5.2 Function Documentation

5.5.2.1 void Biddy_InitMNG (Biddy_Manager * *mng,* int *gddtype*)

Function Biddy_InitMNG initialize a manager.

Description

Biddy_InitMNG creates and initializes a manager. Initialization consists of creating manager structure (MNG), node table (biddyNodeTable), variable table (biddyVariableTable), formula table (biddyFormulaTable), three basic caches (biddyOPCache, biddyEACache and biddyRCCache), and cache list (biddyCacheList). Biddy_InitMNG also initializes constant edges (biddyOne, biddyZero), memory management and automatic garbage collection.

Side effects

Allocates a lot of memory.

More info

Macro Biddy_InitAnonymous() will initialize anonymous manager. Macro Biddy_Init() will initialize anonymous manager for ROBDDs.

Definition at line 217 of file biddyMainGDD.c.

Here is the caller graph for this function:

5.5.2.2 void Biddy_ExitMNG (**Biddy_Manager** * *mng*)

Function Biddy_ExitMNG deletes a manager.

Description

Deallocates all memory allocated by Biddy_InitMNG, Biddy_FoaVariable, Biddy_FoaNode etc.

Side effects

More info

Macro Biddy_Exit() will delete anonymous manager.

Definition at line 758 of file biddyMainGDD.c.

Here is the caller graph for this function:

5.5.2.3 Biddy_String Biddy_About ( )

Function Biddy_About reports version of Biddy package.

Description

Side effects

More info

Definition at line 955 of file biddyMainGDD.c.

5.5.2.4 int Biddy_Managed_GetManagerType (Biddy_Manager *MNG*)

Function Biddy_Managed_GetManagerType reports BDD type used in the manager.

Description

Side effects

More info

Macro Biddy_GetManagerType() is defined for use with anonymous manager.

Definition at line 978 of file biddyMainGDD.c.

5.5.2.5 void Biddy_Managed_SetManagerParameters (Biddy_Manager *MNG,* float *gcr,* float *gcrF,* float *gcrX,* float *rr,* float *rrF,* float *rrX,* float *st,* float *fst,* float *cst,* float *fcst*)

Function Biddy_Managed_SetManagerParameters set modifiable parameters.

Description

Function expect 6 float values. If the value is < 0 then the parameter is not modified. The parameters are: biddy↩NodeTable.gcratio (do not delete nodes if the effect is to small), biddyNodeTable.gcratioF (do not delete nodes if the effect is to small), biddyNodeTable.gcratioX (do not delete nodes if the effect is to small), biddyNodeTable.resizeratio (resize Node table if there are to many nodes), biddyNodeTable.resizeratioF (resize Node table if there are to many nodes), biddyNodeTable.resizeratioX (resize Node table if there are to many nodes), biddyNodeTable.siftingtreshold (stop sifting if the size of the system grows to much), biddyNodeTable.fsiftingtreshold (stop sifting if the size of the function grows to much), biddyNodeTable.convergesiftingtreshold (stop one step of converging sifting if the size of the system grows to much), biddyNodeTable.fconvergesiftingtreshold (stop one step of converging sifting if the size of the function grows to much).

Side effects

Initial values are given in Biddy_InitMNG.

More info

Macro Biddy_SetManagerParameters() is defined for use with anonymous manager.

Definition at line 1021 of file biddyMainGDD.c.

5.5.2.6 Biddy_Edge Biddy_GetThen (Biddy_Edge *fun*)

Function Biddy_GetThen returns THEN successor.

Description

Input mark is not transfered! External use, only.

Side effects

More info

Macro BiddyT(fun) is defined for internal use.

Definition at line 1068 of file biddyMainGDD.c.

5.5.2.7 Biddy_Edge Biddy_GetElse (Biddy_Edge *fun*)

Function Biddy_GetElse returns ELSE successor.

Description

Input mark is not transfered! External use, only.

Side effects

More info

Macro BiddyE(fun) is defined for internal use.

Definition at line 1104 of file biddyMainGDD.c.

5.5.2.8 Biddy_Variable Biddy_GetTopVariable (Biddy_Edge *fun*)

Function Biddy_GetTopVariable returns the top variable.

Description

External use, only.

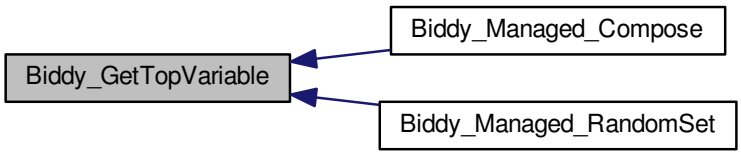
Side effects

More info

Macro BiddyV(fun) is defined for internal use.

Definition at line 1140 of file biddyMainGDD.c.

Here is the caller graph for this function:



5.5.2.9 Biddy_Boolean Biddy_Managed_IsEqv (Biddy_Manager *MNG1,* Biddy_Edge *f1,* Biddy_Manager *MNG2,* Biddy_Edge *f2*)

Function Biddy_Managed_IsEqv returns TRUE iff two BDDs are equal.

Description

It is assumed that f1 and f2 have the same ordering.

Side effects

More info

Macro Biddy_IsEqv(f1,MNG2,f2) is defined for use with anonymous manager.

Definition at line 1164 of file biddyMainGDD.c.

5.5.2.10 void Biddy_Managed_SelectNode (Biddy_Manager *MNG,* Biddy_Edge *f*)

Function Biddy_Managed_SelectNode selects the top node of the given function.

Description

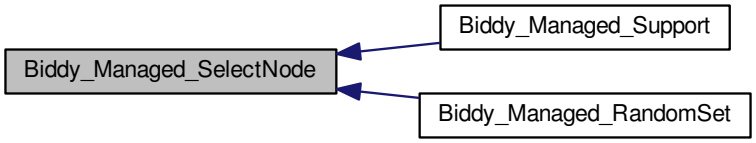
Side effects

More info

Macro Biddy_SelectNode(f) is defined for use with anonymous manager.

Definition at line 1199 of file biddyMainGDD.c.

Here is the caller graph for this function:



5.5.2.11 void Biddy_Managed_DeselectNode (Biddy_Manager *MNG,* Biddy_Edge *f*)

Function Biddy_Managed_DeselectNode deselects the top node of the given function.

Description

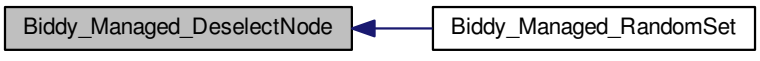
Side effects

More info

Macro Biddy_DeselectNode(f) is defined for use with anonymous manager.

Definition at line 1225 of file biddyMainGDD.c.

Here is the caller graph for this function:



5.5.2.12 Biddy_Boolean Biddy_Managed_IsSelected (Biddy_Manager *MNG,* Biddy_Edge *f*)

Function Biddy_Managed_IsSelected returns TRUE iff the top node of the given function is selected.

Description

Side effects

More info

Macro Biddy_IsSelected(f) is defined for use with anonymous manager.

Definition at line 1251 of file biddyMainGDD.c.

Here is the caller graph for this function:

5.5.2.13 void Biddy_Managed_SelectFunction (Biddy_Manager *MNG*, Biddy_Edge *f*)

Function Biddy_Managed_SelectFunction recursively selects all nodes of a given function.

Description

Side effects

Constant node must be selected before starting this function!

More info

Macro Biddy_SelectFunction(f) is defined for use with anonymous manager.

Definition at line 1278 of file biddyMainGDD.c.

5.5.2.14 void Biddy_Managed_DeselectAll (Biddy_Manager *MNG*)

Function Biddy_Managed_DeselectAll deselects all nodes.

Description

Deselect all nodes.

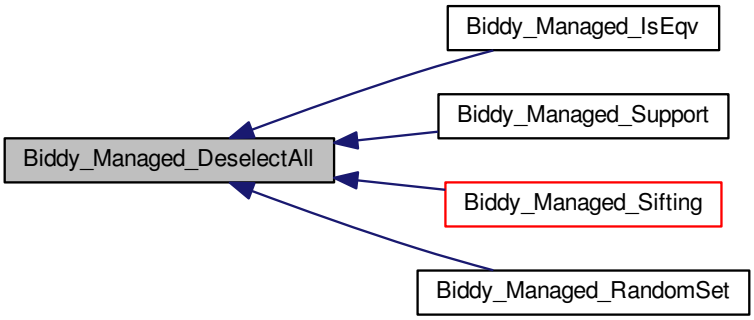
Side effects

More info

Macro Biddy_DeselectAll() is defined for use with anonymous manager.

Definition at line 1335 of file biddyMainGDD.c.

Here is the caller graph for this function:



5.5.2.15 Biddy_Edge Biddy_Managed_GetTerminal (Biddy_Manager *MNG*)

Function Biddy_Managed_GetTerminal returns unmarked and untagged edge pointing to the constant node 1.

Description

Terminal node depends on a manager.

Side effects

More info

Internally, use macro biddyTerminal. Macro Biddy_GetTerminal() is defined for use with anonymous manager.

Definition at line 1369 of file biddyMainGDD.c.

5.5.2.16 Biddy_Edge Biddy_Managed_GetConstantZero (Biddy_Manager *MNG*)

Function Biddy_Managed_GetConstantZero returns constant 0.

Description

Constants 0 and 1 depend on a manager. For combination sets, constant 0 coincides with empty set.

Side effects

More info

Internally, use macro biddyZero. Macro Biddy_GetConstantZero() is defined for use with anonymous manager. Macros Biddy_Managed_GetEmptySet(MNG) and Biddy_GetEmptySet() are defined for manipulation of combination sets.

Definition at line 1399 of file biddyMainGDD.c.

5.5.2.17 Biddy_Edge Biddy_Managed_GetConstantOne (Biddy_Manager *MNG*)

Function Biddy_Managed_GetConstantOne returns constant 1.

Description

Constants 0 and 1 depend on a manager. For combination sets, constant 1 coincides with universal set.

Side effects

For ZBDDs and ZFDDs, you should always obtain constant 1 through the call of this function!

More info

Internally, use macro biddyOne (also for ZBDDs and ZFDDs!). Macro Biddy_GetConstantOne() is defined for use with anonymous manager. Macros Biddy_Managed_GetUniversalSet(MNG) and Biddy_GetUniversalSet() are defined for manipulation of combination sets.

Definition at line 1431 of file biddyMainGDD.c.

5.5.2.18 Biddy_Edge Biddy_Managed_GetBaseSet (Biddy_Manager *MNG*)

Function Biddy_Managed_GetBaseSet returns set containing only a null combination, i.e. it returns {{}}.

Description

Side effects

More info

Macro Biddy_GetBaseSet() is defined for use with anonymous manager.

Definition at line 1457 of file biddyMainGDD.c.

Here is the caller graph for this function:

5.5.2.19 Biddy_Variable Biddy_Managed_GetVariable (Biddy_Manager *MNG,* Biddy_String *x*)

Function Biddy_Managed_GetVariable returns variable with the given name.

Description

Side effects

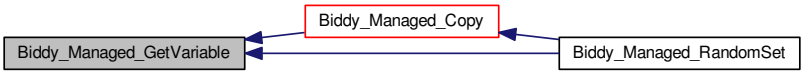
If variable is not found function returns 0!

More info

Macro Biddy_GetVariable(x) is defined for use with anonymous manager.

Definition at line 1515 of file biddyMainGDD.c.

Here is the caller graph for this function:



5.5.2.20 Biddy_Variable Biddy_Managed_GetPrevVariable (Biddy_Manager *MNG,* Biddy_Variable *v*)

Function Biddy_Managed_GetPrevVariable returns previous variable in the global ordering (lower, topmore).

Description

Side effects

More info

Macro Biddy_GetPrevVariable(v) is defined for use with anonymous manager.

Definition at line 1598 of file biddyMainGDD.c.

5.5.2.21 Biddy_Variable Biddy_Managed_GetNextVariable (Biddy_Manager *MNG,* Biddy_Variable *v*)

Function Biddy_Managed_GetNextVariable returns next variable in the global ordering (higher, bottommore).

Description

Side effects

More info

Macro Biddy_GetNextVariable(v) is defined for use with anonymous manager.

Definition at line 1628 of file biddyMainGDD.c.

5.5.2.22 Biddy_Edge Biddy_Managed_GetVariableEdge (Biddy_Manager *MNG*, Biddy_Variable *v*)

Function Biddy_Managed_GetVariableEdge returns variable's edge.

Description

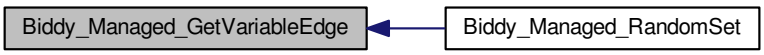
Side effects

More info

Macro Biddy_GetVariableEdge(v) is defined for use with anonymous manager.

Definition at line 1657 of file biddyMainGDD.c.

Here is the caller graph for this function:



5.5.2.23 Biddy_Edge Biddy_Managed_GetElementEdge (Biddy_Manager *MNG*, Biddy_Variable *v*)

Function Biddy_Managed_GetElementEdge returns element's edge.

Description

Side effects

More info

Macro Biddy_GetElementEdge(v) is defined for use with anonymous manager.

Definition at line 1682 of file biddyMainGDD.c.

5.5.2.24 Biddy_String Biddy_Managed_GetVariableName (Biddy_Manager *MNG*, Biddy_Variable *v*)

Function Biddy_Managed_GetVariableName returns the name of a variable.

Description

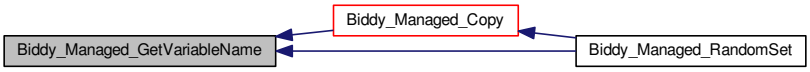
Side effects

More info

Macro Biddy_GetVariableName(v) is defined for use with anonymous manager.

Definition at line 1707 of file biddyMainGDD.c.

Here is the caller graph for this function:



5.5.2.25 Biddy_Edge Biddy_Managed_GetTopVariableEdge (Biddy_Manager *MNG*, Biddy_Edge *f*)

Function Biddy_Managed_GetTopVariableEdge returns variable's edge of top variable.

Description

Side effects

TO DO: For ZBDDs, element edge is sometimes preffered over variable edge.

More info

Macro Biddy_GetTopVariableEdge(f) is defined for use with anonymous manager.

Definition at line 1735 of file biddyMainGDD.c.

5.5.2.26 Biddy_String Biddy_Managed_GetTopVariableName (Biddy_Manager *MNG*, Biddy_Edge *f*)

Function Biddy_Managed_GetTopVariableName returns the name of top variable.

Description

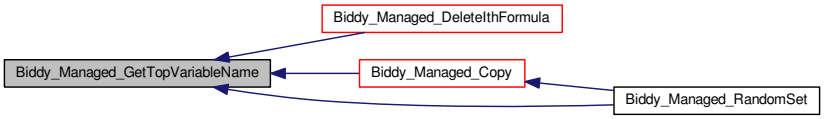
Side effects

More info

Macro Biddy_GetTopVariableName(f) is defined for use with anonymous manager.

Definition at line 1763 of file biddyMainGDD.c.

Here is the caller graph for this function:



5.5.2.27 char Biddy_Managed_GetTopVariableChar (Biddy_Manager *MNG*, Biddy_Edge *f*)

Function Biddy_Managed_GetTopVariableChar returns the first character in the name of top variable.

Description

Side effects

More info

Macro Biddy_GetTopVariableChar(f) is defined for use with anonymous manager.

Definition at line 1791 of file biddyMainGDD.c.

5.5.2.28 void Biddy_Managed_ResetVariablesValue (Biddy_Manager *MNG*)

Function Biddy_Managed_ResetVariablesValue sets all variable's value to biddyZero.

Description

Side effects

Only active (used) variables are reinitialized.

More info

Macro Biddy_ResetVariablesValue() is defined for use with anonymous manager.

Definition at line 1820 of file biddyMainGDD.c.

5.5.2.29 void Biddy_Managed_SetVariableValue (Biddy_Manager *MNG,* Biddy_Variable *v,* Biddy_Edge *f*)

Function Biddy_Managed_SetVariableValue sets variable's value.

Description

Side effects

More info

Macro Biddy_SetVariableValue(v,f) is defined for use with anonymous manager.

Definition at line 1849 of file biddyMainGDD.c.

5.5.2.30 Biddy_Boolean Biddy_Managed_IsSmaller (Biddy_Manager *MNG,* Biddy_Variable *fv,* Biddy_Variable *gv*)

Function Biddy_Managed_IsSmaller returns TRUE if the first variable is smaller (= lower = previous = above = topmore).

Description

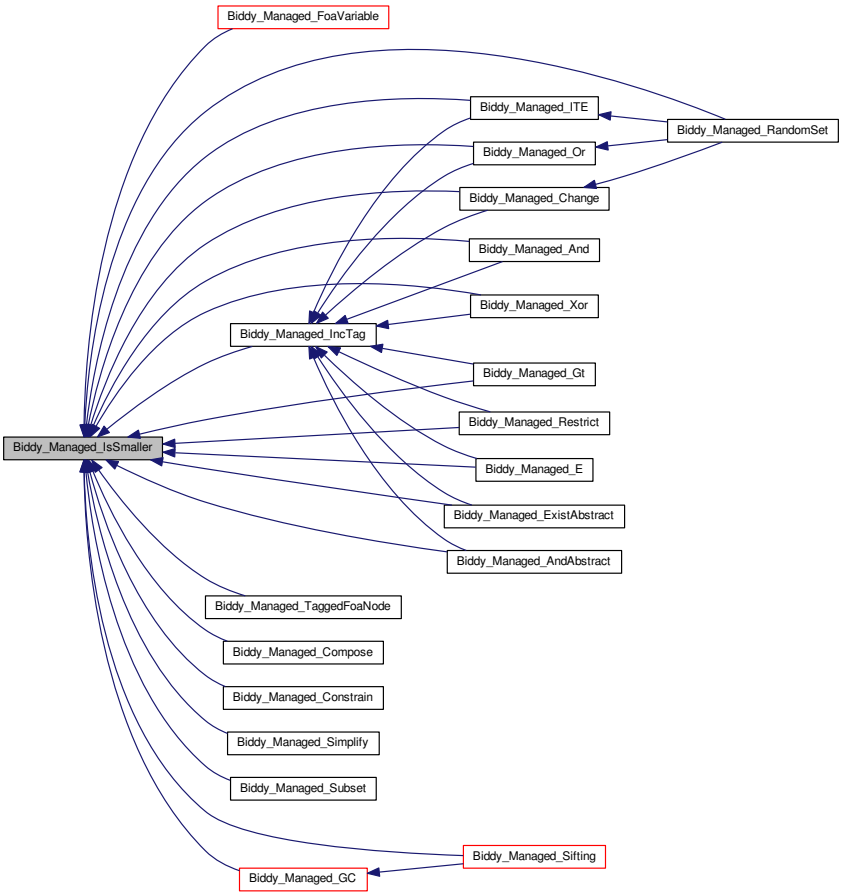
Side effects

More info

Macro Biddy_IsSmaller(fv,gv) is defined for use with anonymous manager.

Definition at line 1875 of file biddyMainGDD.c.

Here is the caller graph for this function:



5.5.2.31 Biddy_Variable Biddy_Managed_FoaVariable (Biddy_Manager *MNG,* Biddy_String *x,* Biddy_Boolean *varelem*)

Function Biddy_Managed_FoaVariable finds variable/element or adds new variable (i.e. Boolean function f = x) and new element (i.e. it creates set {{x}}).

Description

If variable/element already exists, function returns the existing one. If x == NULL then numbered variable/element is added. Numbered variables/elements have only digits in its name. The current number of numbered variables/elements is stored in numnum. If numbered variable/element is requested then function increments numnum and creates a new (non-existing) variable/element. Parameter varelem is used to determine how to adapt the existing BDD base to keep the current formula valid (use varelem = TRUE if formulae represent Boolean functions and varelem = FALSE if they represent combination sets). The ordering of the new variable/element is determined in Biddy_InitMNG. Function always returns variable.

Side effects

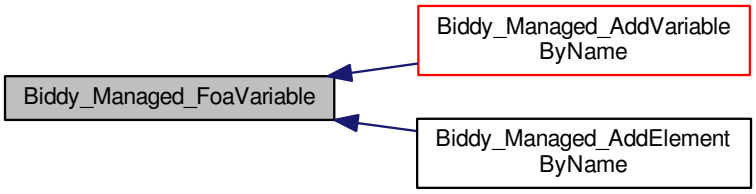
Adding new variable/element may change the meaning of the existing BDDs. Variables and elements are repaired. Moreover, formulae are repaired with regards to the parameter varelem. BDDs without external references are not repaired. For OBDDs, OFDDs, TZBDDS, and TZFDDs, it is safe to add new variables/elements if BDDs are used to represent Boolean functions. For ZBDDs and ZFDDs, it is safe to add new variables/elements if BDDs are used to represent combination sets. User should not add numbered variables/elements with some other function. TO DO: Formulae in user's formula tables are not repaired, yet!

More info

Macro Biddy_FoaVariable(x) is defined for use with anonymous manager.

Definition at line 1923 of file biddyMainGDD.c.

Here is the caller graph for this function:



5.5.2.32 Biddy_Edge Biddy_Managed_AddVariableByName (Biddy_Manager *MNG*, Biddy_String *x*)

Function Biddy_Managed_AddVariableByName adds variable.

Description

Biddy_Managed_AddVariableByName uses Biddy_Managed_FoaVariable to find or add variable. Function returns variable edge. If variable already exists, function returns the existing variable edge. For more details see Biddy_↩ Managed_FoaVariable.

Side effects

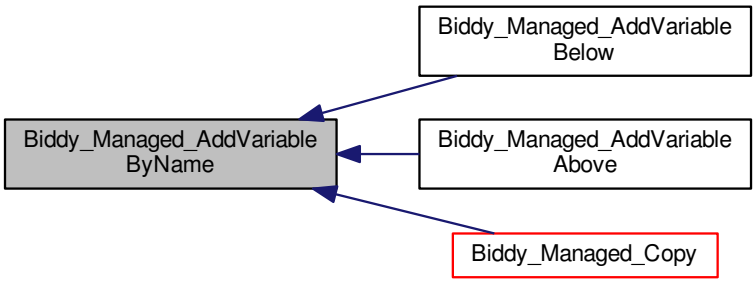
See Biddy_Managed_FoaVariable.

More info

Macro Biddy_AddVariableByName(x) is defined for use with anonymous manager. Macros Biddy_Managed_Add↩ Variable(MNG) and Biddy_AddVariable() are defined for creating numbered variables.

Definition at line 2202 of file biddyMainGDD.c.

Here is the caller graph for this function:



5.5.2.33 Biddy_Edge Biddy_Managed_AddElementByName (Biddy_Manager *MNG*, Biddy_String *x*)

Function Biddy_Managed_AddElementByName adds element.

Description

Biddy_Managed_AddElementByName uses Biddy_Managed_FoaVariable to find or add element. Function returns element edge. If element already exists, function returns the existing element edge. For more details see Biddy_↩ Managed_FoaVariable.

Side effects

See Biddy_Managed_FoaVariable.

More info

Macro Biddy_AddElementByName(x) is defined for use with anonymous manager. Macros Biddy_Managed_Add↩ Element(MNG) and Biddy_AddElement() are defined for creating numbered elements.

Definition at line 2237 of file biddyMainGDD.c.

5.5.2.34 Biddy_Edge Biddy_Managed_AddVariableBelow (Biddy_Manager *MNG*, Biddy_Variable *v*)

Function Biddy_Managed_AddVariableBelow adds a numbered variable.

Description

Biddy_Managed_AddVariableBelow uses Biddy_Managed_AddVariableByName to add numbered variable. Then, the order of the new variable is changed to become immediately below the given variable (below = next = bottom-more in BDD) Function returns variable edge.

Side effects

More info

Macro Biddy_AddVariableBelow(v) is defined for use with anonymous manager.

Definition at line 2269 of file biddyMainGDD.c.

5.5.2.35 Biddy_Edge Biddy_Managed_AddVariableAbove (Biddy_Manager *MNG,* Biddy_Variable *v*)

Function Biddy_Managed_AddVariableAbove adds a numbered variable.

Description

Biddy_Managed_AddVariableAbove uses Biddy_Managed_AddVariableByName to add numbered variable. Then, the order of the new variable is changed to become immediately above the given variable (above = previous = topmore in BDD) Function returns variable edge.

Side effects

More info

Macro Biddy_AddVariableAbove(v) is defined for use with anonymous manager.

Definition at line 2354 of file biddyMainGDD.c.

5.5.2.36 Biddy_Edge Biddy_Managed_TransferMark (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Boolean *mark,* Biddy_Boolean *leftright*)

Function Biddy_Managed_TransferMark returns edge with inverted complement bit iff the second parameter is T↩RUE and normalization rules require this.

Description

Parameter leftright should be TRUE for left and FALSE for right. For OBDD, it is better to use macro Biddy_InvCond. For OBDD, parameter leftrigh is ignored.

Side effects

TO DO: swap the meaning of parameter leftright (left should be FALSE)

More info

Macro Biddy_TransferMark() is defined for use with anonymous manager.

Definition at line 2437 of file biddyMainGDD.c.

Here is the caller graph for this function:

5.5.2.37 Biddy_Edge Biddy_Managed_IncTag (Biddy_Manager *MNG,* Biddy_Edge *f*)

Function Biddy_Managed_IncTag returns edge with an incremented tag.

Description

Used for TZBDDs and TZFDDs, only.

Side effects

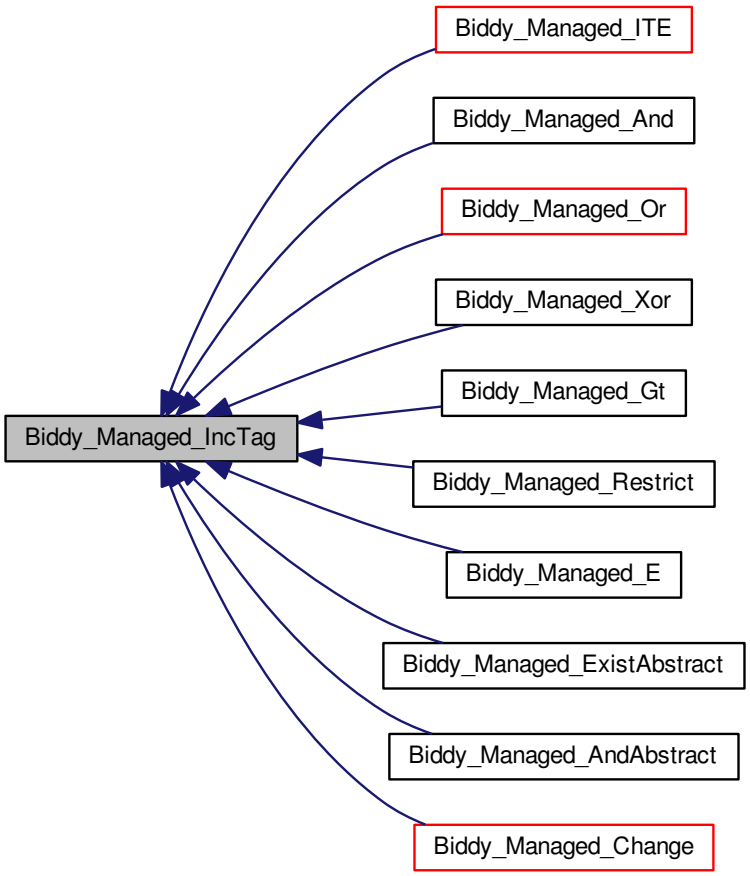
It is not checked, that the resulting tag is not greater than top variable. Function may return non-fresh node even if f is fresh.

More info

Macro Biddy_IncTag() is defined for use with anonymous manager.

Definition at line 2479 of file biddyMainGDD.c.

Here is the caller graph for this function:



5.5.2.38 Biddy_Edge Biddy_Managed_TaggedFoaNode (Biddy_Manager *MNG*, Biddy_Variable *v*, Biddy_Edge *pf*, Biddy_Edge *pt*, Biddy_Variable *ptag*, Biddy_Boolean *garbageAllowed*)

Function Biddy_Managed_TaggedFoaNode finds or adds new node with the given variable and successors.

Description

If such node already exists, function returns it and does not create the new one. If pf = pt = NULL (and ptag = v) then new variable (for OBDD, OFDD, TZBDD, and TZFDD) or new element (for ZBDD and ZFDD) is created. For OBDD, ZBDD, OFDD, and ZFDD, parameter ptag is ignored. For TZBDD and TZFDD, the returned edge is tagged with the given ptag. This function should not be called directly to add new variables and elements, you must use Biddy_Managed_FoaVariable, Biddy_Managed_AddVariableByName, or Biddy_Managed_AddElementByName.

Side effects

Using Biddy_Managed_TaggedFoaNode you can create node with an arbitrary ordering. It is much more safe to use Boolean operators, e.g. Biddy_Managed_ITE.

More info

Macro Biddy_Managed_FoaNode(MNG,v,pf,pt,garbageAllowed) is defined for use with implicit tags or without tags. Macros Biddy_TaggedFoaNode(v,pf,pt,tag,garbageAllowed) and Biddy_FoaNode(v,pf,pt,garbageAllowed) are defined for use with anonymous manager.

Definition at line 2541 of file biddyMainGDD.c.

5.5.2.39 Biddy_Edge Biddy_Managed_Not (Biddy_Manager *MNG,* Biddy_Edge *f*)

Function Biddy_Managed_Not calculates Boolean function NOT.

Description

Side effects

Implemented for OBDD, ZBDD, and TZBDD. For OBDD and OFDD, it is better to use macro Biddy_Inv. For OBDD, cache table is not needed. For ZBDD, recursive calls are via Xor and thus its cache table is used. For TZBDD, results are cached as (biddyOne,f,biddyZero) - see Biddy_Xor.

More info

Macro Biddy_Not() is defined for use with anonymous manager.

Definition at line 2887 of file biddyMainGDD.c.

5.5.2.40 Biddy_Edge Biddy_Managed_ITE (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Edge *g,* Biddy_Edge *h*)

Function Biddy_Managed_ITE calculates ITE operation of three Boolean functions.

Description

Side Effects

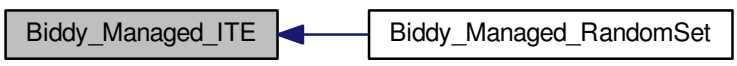
Implemented for OBDDs, ZBDDs, and TZBDDs. For OBDDs, results are cached as parameters to ITE(F,G,H)= F∗G XOR F'∗H. For ZBDDs and TZBDDs, results are cached as (f,g,h) where f,g,h != 0, f != g, f != h, and g != h.

More info

Macro Biddy_ITE(f,g,h) is defined for use with anonymous manager.

Definition at line 3092 of file biddyMainGDD.c.

Here is the caller graph for this function:



5.5.2.41 Biddy_Edge Biddy_Managed_And (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Edge *g*)

Function Biddy_Managed_And calculates Boolean function AND (conjunction).

Description

For combination sets, this function coincides with Intersection.

Side Effects

Used by ITE (for OBDDs). Implemented for OBDDs, ZBDDs, and TZBDDs. For OBDDs, results are cached as parameters to ITE(F,G,H)= F∗G XOR F'∗H. For ZBDDs and TZBDDs, results are cached as (f,biddyZero,g).

More Info

Macro Biddy_And(f,g) is defined for use with anonymous manager. Macros Biddy_Managed_Intersect(MNG,f,g) and Biddy_Intersect(f,g) are defined for manipulation of combination sets.

Definition at line 3658 of file biddyMainGDD.c.

5.5.2.42 Biddy_Edge Biddy_Managed_Or (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Edge *g*)

Function Biddy_Managed_Or calculates Boolean function OR (disjunction).

Description

For combination sets, this function coincides with Union.

Side Effects

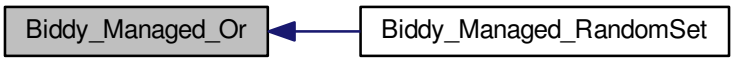
Implemented for OBDDs, ZBDDs, and TZBDDs. For OBDDs, results are cached as parameters to ITE(F,G,H)= F∗G XOR F'∗H. For ZBDDs and TZBDDs, results will be cached as (biddyZero,f,g).

More Info

Macro Biddy_Or(f,g) is defined for use with anonymous manager. Macros Biddy_Managed_Union(MNG,f,g) and Biddy_Union(f,g) are defined for manipulation of combination sets.

Definition at line 4119 of file biddyMainGDD.c.

Here is the caller graph for this function:



5.5.2.43 Biddy_Edge Biddy_Managed_Nand (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Edge *g*)

Function Biddy_Managed_Nand calculates Boolean function NAND (Sheffer).

Description

Side Effects

Implemented for OBDDs. Prototyped for ZBDDs and TZBDDs (via and-not). For OBDDs, results are cached as parameters to ITE(F,G,H)= F∗G XOR F'∗H. For ZBDDs and TZBDDs, results are not cached.

More Info

Macro Biddy_Nand(f,g) is defined for use with anonymous manager.

Definition at line 4544 of file biddyMainGDD.c.

5.5.2.44 Biddy_Edge Biddy_Managed_Nor (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Edge *g*)

Function Biddy_Managed_Nor calculates Boolean function NOR (Peirce).

Description

Side Effects

Implemented for OBDDs. Prototyped for ZBDDs and TZBDDs (via and-not). For OBDDs, results are cached as parameters to ITE(F,G,H)= F∗G XOR F'∗H. For ZBDDs and TZBDDs, results are not cached.

More Info

Macro Biddy_Nor(f,g) is defined for use with anonymous manager.

Definition at line 4624 of file biddyMainGDD.c.

5.5.2.45 Biddy_Edge Biddy_Managed_Xor (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Edge *g*)

Function Biddy_Managed_Xor calculates Boolean function XOR.

Description

Side Effects

Used by ITE (for OBDDs). Implemented for OBDDs, ZBDDs, and TZBDDs. For OBDD, results are cached as parameters to ITE(F,G,H)= F∗G XOR F'∗H. For ZBDDs and TZBDDs, results are cached as (f,g,biddyZero).

More Info

Macro Biddy_Xor(f,g) is defined for use with anonymous manager.

Definition at line 4703 of file biddyMainGDD.c.

5.5.2.46 Biddy_Edge Biddy_Managed_Xnor (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Edge *g*)

Function Biddy_Managed_Xnor calculates Boolean function XNOR.

Description

Side Effects

Implemented for OBDDs. Prototyped for ZBDDs and TZBDDs (via xor-not). For OBDD, results are cached as parameters to ITE(F,G,H)= F∗G XOR F'∗H. For ZBDDs and TZBDDs, results are not cached.

More Info

Macro Biddy_Xnor(f,g) is defined for use with anonymous manager.

Definition at line 5169 of file biddyMainGDD.c.

5.5.2.47 Biddy_Edge Biddy_Managed_Leq (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Edge *g*)

Function Biddy_Managed_Leq calculates Boolean implication.

Description

Boolean function leq(f,g) = or(not(f),g) = not(gt(f,g)). This function coincides with implication f->g.

Side Effects

Implemented for OBDDs. Prototyped for ZBDDs and TZBDDs (via and-not). For OBDD, results are cached as parameters to ITE(F,G,H)= F∗G XOR F'∗H. For ZBDDs and TZBDDs, results are cached as (f,f,g).

More Info

Macro Biddy_Leq(f,g) is defined for use with anonymous manager.

Definition at line 5250 of file biddyMainGDD.c.

5.5.2.48 Biddy_Edge Biddy_Managed_Gt (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Edge *g*)

Function Biddy_Managed_Gt calculates the negation of Boolean implication.

Description

Boolean function gt(f,g) = and(f,not(g)). For combination sets, this function coincides with Diff.

Side Effects

Implemented for OBDDs, ZBDDs, and TZBDDs. For OBDD, results are cached as parameters to ITE(F,G,H)= F∗G XOR F'∗H. For ZBDDs and TZBDDs, results are cached as (f,g,g).

More Info

Macro Biddy_Gt(f,g) is defined for use with anonymous manager. Macros Biddy_Managed_Diff(MNG,f,g) and Biddy_Diff(f,g) are defined for manipulation of combination sets.

Definition at line 5333 of file biddyMainGDD.c.

5.5.2.49 Biddy_Boolean Biddy_Managed_IsLeq (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Edge *g*)

Function Biddy_Managed_IsLeq returns TRUE iff function f is included in function g.

Description

Side Effects

Prototyped for OBDDs, ZBDDs, and TZBDDs (via calculating full implication, this is less efficient as implementation in CUDD).

More Info

Macro Biddy_IsLeq(f,g) is defined for use with anonymous manager.

Definition at line 5640 of file biddyMainGDD.c.

5.5.2.50 Biddy_Edge Biddy_Managed_Restrict (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Variable *v,* Biddy_Boolean *value*)

Function Biddy_Managed_Restrict calculates a restriction of Boolean function.

Description

Original BDD is not changed. This is not Coudert and Madre's restrict function (use Biddy_Simplify if you need that one).

Side effects

For OBDDs, recursive calls use optimization: F(a=x) == NOT((NOT F)(a=x)).

More info

Macro Biddy_Restrict(f,v,value) is defined for use with anonymous manager.

Definition at line 5693 of file biddyMainGDD.c.

5.5.2.51 Biddy_Edge Biddy_Managed_Compose (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Edge *g,* Biddy_Variable *v*)

Function Biddy_Managed_Compose calculates a composition of two Boolean functions.

Description

Original BDD is not changed.

Side effects

For OBDDs, recursive calls use optimization: F(a=G) == NOT((NOT F)(a=G)).

More info

Macro Biddy_Compose(f,g,v) is defined for use with anonymous manager.

Definition at line 5886 of file biddyMainGDD.c.

5.5.2.52 Biddy_Edge Biddy_Managed_E (Biddy_Manager *MNG*, Biddy_Edge *f*, Biddy_Variable *v*)

Function Biddy_Managed_E calculates an existential quantification of Boolean function.

Description

Original BDD is not changed.

Side effects

Be careful: ExA F != NOT(ExA (NOT F)). Counterexample: Exb (AND (NOT a) b c).

More info

Macro Biddy_E(f,v) is defined for use with anonymous manager.

Definition at line 6089 of file biddyMainGDD.c.

5.5.2.53 Biddy_Edge Biddy_Managed_A (Biddy_Manager *MNG*, Biddy_Edge *f*, Biddy_Variable *v*)

Function Biddy_Managed_A calculates an universal quantification of Boolean function.

Description

Original BDD is not changed.

Side effects

Implemented for OBDDs. Prototyped for ZBDDs and TZBDDs. Be careful: AxA F != NOT(AxA (NOT F)). Counterexample: Axb (AND (NOT a) b c).

More info

Macro Biddy_A(f,v) is defined for use with anonymous manager.

Definition at line 6270 of file biddyMainGDD.c.

5.5.2.54 Biddy_Boolean Biddy_Managed_IsVariableDependent (Biddy_Manager *MNG*, Biddy_Edge *f*, Biddy_Variable *v*)

Function Biddy_Managed_IsVariableDependent returns TRUE iff variable is dependent on others in a function.

Description

A variable is dependent on others in a function iff universal quantification of this variable returns constant FALSE.

Side effects

Prototyped for OBDDs (via xA, calculating full universal quantification is less efficient as direct implementation in CUDD).

More info

Macro Biddy_IsVariableDependent(f,v) is defined for use with anonymous manager.

Definition at line 6347 of file biddyMainGDD.c.

5.5.2.55 Biddy_Edge Biddy_Managed_ExistAbstract (Biddy_Manager *MNG*, Biddy_Edge *f*, Biddy_Edge *cube*)

Function Biddy_Managed_ExistAbstract existentially abstracts all the variables in cube from f.

Description

Original BDD is not changed.

Side effects

More info

Macro Biddy_ExistAbstract(f,cube) is defined for use with anonymous manager.

Definition at line 6389 of file biddyMainGDD.c.

5.5.2.56 Biddy_Edge Biddy_Managed_UnivAbstract (Biddy_Manager *MNG*, Biddy_Edge *f*, Biddy_Edge *cube*)

Function Biddy_Managed_UnivAbstract universally abstracts all the variables in cube from f.

Description

Original BDD is not changed.

Side effects

Implemented for OBDDs. Prototyped for ZBDDs and TZBDDs.

More info

Macro Biddy_UnivAbstract(f,cube) is defined for use with anonymous manager.

Definition at line 6619 of file biddyMainGDD.c.

5.5.2.57 Biddy_Edge Biddy_Managed_AndAbstract (Biddy_Manager *MNG*, Biddy_Edge *f*, Biddy_Edge *g*, Biddy_Edge *cube*)

Function Biddy_Managed_AndAbstract calculates the AND of two BDDs and simultaneously (existentially) abstracts the variables in cube.

Description

Side effects

More info

Macro Biddy_AndAbstract(f,g,cube) is defined for use with anonymous manager.

Definition at line 6694 of file biddyMainGDD.c.

5.5.2.58 Biddy_Edge Biddy_Managed_Constrain (Biddy_Manager *MNG*, Biddy_Edge *f*, Biddy_Edge *c*)

Function Biddy_Managed_Constrain calculates Coudert and Madre's constrain function.

Description

Coudert and Madre's constrain function is also called a generalized cofactor of function f with respect to function c.

Side effects

Implemented for OBDDs. Cache table is not implemented, yet.

More info

Macro Biddy_Constrain(f,c) is defined for use with anonymous manager.

Definition at line 7103 of file biddyMainGDD.c.

5.5.2.59 Biddy_Edge Biddy_Managed_Simplify (Biddy_Manager *MNG*, Biddy_Edge *f*, Biddy_Edge *c*)

Function Biddy_Managed_Simplify calculates Coudert and Madre's restrict function.

Description

Coudert and Madre's restrict function tries to simplify function f by restricting it to the domain covered by function c. No checks are done to see if the result is actually smaller than the input.

Side effects

Implemented for OBDDs. Cache table is not implemented, yet.

More info

Macro Biddy_Simplify(f,c) is defined for use with anonymous manager.

Definition at line 7208 of file biddyMainGDD.c.

5.5.2.60 Biddy_Edge Biddy_Managed_Support (Biddy_Manager *MNG,* Biddy_Edge *f*)

Function Biddy_Managed_Support calculates a product of all dependent variables (OBDD and TZBDD) or the combination set containing a subset which includes all dependent variables (ZBDD).

Description

Implemented for OBDDs. Prototyped for ZBDDs and TZBDDs. For OBDD, dependent variables are all variables existing in the graph. For ZBDD and TZBDD, this is not true.

Side effects

More info

Macro Biddy_Support(f) is defined for use with anonymous manager.

Definition at line 7317 of file biddyMainGDD.c.

5.5.2.61 Biddy_Edge Biddy_Managed_Replace (Biddy_Manager *MNG,* Biddy_Edge *f*)

Function Biddy_Managed_Replace calculates Boolean function with one or more variables replaced.

Description

Implemented for OBDDs. Prototyped for ZBDDs and TZBDDs (via And-Xor-Not-Restrict). Original BDD is not changed. The sets of current and new variables should be disjoint. Replacing is controlled by variable's values (which are edges!). Use Biddy_Managed_ResetVariablesValue and Biddy_Managed_SetVariableValue to prepare replacing.

Side effects

Cache table is not implemented, yet.

More info

Macro Biddy_Replace(f) is defined for use with anonymous manager.

Definition at line 7421 of file biddyMainGDD.c.

5.5.2.62 Biddy_Edge Biddy_Managed_Change (Biddy_Manager *MNG*, Biddy_Edge *f*, Biddy_Variable *v*)

Function Biddy_Managed_Change change the form of the given variable (positive literal becomes negative and vice versa).

Description

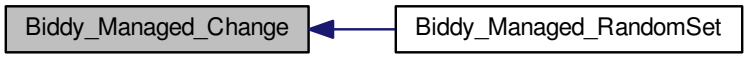
Side effects

More info

Macro Biddy_Change() is defined for use with anonymous manager.

Definition at line 7528 of file biddyMainGDD.c.

Here is the caller graph for this function:



5.5.2.63 Biddy_Edge Biddy_Managed_Subset (Biddy_Manager *MNG*, Biddy_Edge *f*, Biddy_Variable *v*, Biddy_Boolean *value*)

Function Biddy_Managed_Subset calculates a division of Boolean function with a literal.

Description

Original BDD is not changed. For combination sets, this function coincides with Subset0 and Subset1.

Side effects

Cache table for AND is used.

More info

Macro Biddy_Subset(f,v,value) is defined for use with anonymous manager. Macros Biddy_Managed_Subset0(↩ MNG,f,v), Biddy_Subset0(f,v), Biddy_Managed_Subset1(MNG,f,v), and Biddy_Subset1(f,v) are defined for manipulation of combination sets.

Definition at line 7670 of file biddyMainGDD.c.

5.5.2.64 Biddy_Boolean Biddy_Managed_IsOK (Biddy_Manager *MNG*, Biddy_Edge *f*)

Function Biddy_Managed_IsOK returns TRUE iff given node is not obsolete.

Description

This is needed for implementation of user caches.

Side effects

More info

Macro BiddyIsOK(f) is defined for debugging. It will check more properties and not only the expiry value. Macro Biddy_IsOK(f) is defined for use with anonymous manager.

Definition at line 7907 of file biddyMainGDD.c.

5.5.2.65 void Biddy_Managed_GC (Biddy_Manager *MNG*, Biddy_Variable *target*, Biddy_Boolean *purge*, Biddy_Boolean *total*)

Function Biddy_Managed_GC performs garbage collection.

Description

All obsolete nodes are deleted. Parameter target is used during sifting. If parameter total is true than all obsolete nodes are deleted, otherwise nodes are deleted only if there are enough obsolete nodes. Nodes from deleted non-obsolete formulae are immediately removed only if parameter purge is true (this should not be used during the automatic garbage collection), otherwise these nodes only become fresh.

Side effects

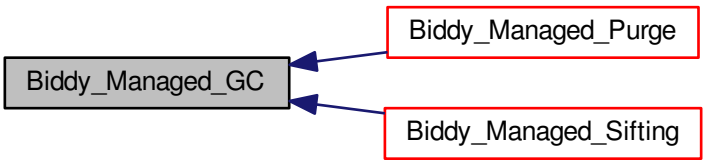
The first element of each chain in a node table should have a special value for its 'prev' element to allow tricky but efficient deleting. Moreover, 'prev' and 'next' should be the first and the second element in the structure BiddyNode, respectively. Garbage collecion is reported by biddyNodeTable.garbage only if some bad nodes are purged!

More info

Macro Biddy_GC(target,purge,total) is defined for use with anonymous manager. Macros Biddy_Managed_Auto↩GC(MNG) and Biddy_AutoGC() are useful variants with target = 0, purge = FALSE, and total = FALSE.

Definition at line 7947 of file biddyMainGDD.c.

Here is the caller graph for this function:



5.5.2.66 void Biddy_Managed_Clean (Biddy_Manager *MNG*)

Function Biddy_Managed_Clean performs cleaning.

Description

Discard all nodes which are not preserved or which are not preserved anymore. Obsolete nodes are not immediately removed, they will be removed during the first garbage collection.

Side effects

Tag deleted is not considered and thus no fortified node and no prolonged node is discarded.

More info

Macro Biddy_Clean() is defined for use with anonymous manager.

Definition at line 8346 of file biddyMainGDD.c.

5.5.2.67 void Biddy_Managed_Purge (Biddy_Manager *MNG*)

Function Biddy_Managed_Purge immediately removes all nodes which were not preserved or which are not preserved anymore.

Description

All fresh and obsolete nodes are immediatelly removed. Moreover, nodes from deleted prolonged formulae and nodes from deleted fortified formulae are removed if they are not needed by other formulae. Call to Biddy_Purge does not count as clearing and thus all preserved formulae remains preserved for the same number of clearings.

Side effects

Removes all fresh nodes!

More info

Macro Biddy_Purge(f) is defined for use with anonymous manager.

Definition at line 8393 of file biddyMainGDD.c.

Here is the caller graph for this function:

5.5.2.68 void Biddy_Managed_PurgeAndReorder (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Boolean *converge*)

Function Biddy_Managed_PurgeAndReorder immediately removes non-preserved nodes and triggers reordering on function.

Description

All obsolete nodes are immediatelly removed. Moreover, nodes from deleted prolonged formulae and nodes from deleted fortified formulae are removed if they are not needed by other formulae. If BDD is given (f != NULL), reordering on function is used. Otherwise (f == NULL) global reordering is used. Call to Biddy_PurgeAndReorder does not count as clearing and thus all preserved formulae remains preserved for the same number of clearings.

Side effects

Removes all fresh nodes.

More info

Macro Biddy_PurgeAndReorder(f) is defined for use with anonymous manager.

Definition at line 8429 of file biddyMainGDD.c.

5.5.2.69 void Biddy_Managed_Refresh (Biddy_Manager *MNG,* Biddy_Edge *f*)

Function Biddy_Managed_Refresh refreshes top node in a given function.

Description

This is an external variant of internal macro BiddyRefresh This is needed for implementing user caches.

Side effects

More info

Macro Biddy_Refresh(f) is defined for use with anonymous manager.

Definition at line 8458 of file biddyMainGDD.c.

5.5.2.70 void Biddy_Managed_AddCache (Biddy_Manager *MNG,* Biddy_GCFunction *gc*)

Function Biddy_Managed_AddCache adds cache to the end of Cache list.

Description

If Cache list does not exist, function creates it.

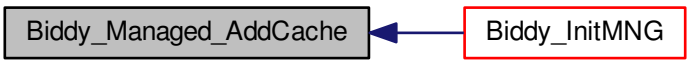
Side effects

More info

Macro Biddy_AddCache(gc) is defined for use with anonymous manager.

Definition at line 8484 of file biddyMainGDD.c.

Here is the caller graph for this function:



5.5.2.71 unsigned int Biddy_Managed_AddFormula (Biddy_Manager *MNG*, Biddy_String *x*, Biddy_Edge *f*, int *c*)

Function Biddy_Managed_AddFormula adds formula to Formula table.

Description

Nodes of the given BDD will be preserved for the given number of clearings. If (x != NULL) then formula is accessible by its name. If (c == -1) then formula is not preserved. If (c == 0) then formula is persistently preserved and you have to use Biddy_DeleteFormula to remove its nodes. There are two macros defined to simplify formulae management. Macro Biddy_Managed_AddTmpFormula(mng,bdd,c) is defined as Biddy_Managed_AddFormula(mng,N↩ULL,bdd,c) and macro Biddy_Managed_AddPersistentFormula(mng,name,bdd) is defined as Biddy_Managed_↩AddFormula(mng,name,bdd,0).

Side effects

Function is prolonged or fortified. Formulae with name are ordered by name. If formula with the same name already exists, it will be overwritten (preserved and persistently preserved formulae, too)!

More info

Macros Biddy_AddFormula(x,f), Biddy_AddTmpFormula(f,c), and Biddy_AddPersistentFormula(x,f) are defined for use with anonymous manager.

Definition at line 8545 of file biddyMainGDD.c.

Here is the caller graph for this function:

5.5.2.72 Biddy_Boolean Biddy_Managed_FindFormula (Biddy_Manager *MNG*, Biddy_String *x*, Biddy_Edge * *f*)

Function Biddy_Managed_FindFormula find formula in Formula table.

Description

Side effects

More info

Macro Biddy_FindFormula(x,f) is defined for use with anonymous manager.

Definition at line 8738 of file biddyMainGDD.c.

Here is the caller graph for this function:

5.5.2.73 Biddy_Boolean Biddy_Managed_DeleteFormula (Biddy_Manager *MNG*, Biddy_String *x*)

Function Biddy_Managed_DeleteFormula delete formula from Formula table.

Description

Formula is labelled but not immediately removed. Nodes of the given formula are not immediately removed.

Side effects

Formula is not accessible by its name anymore. Formulae representing constants and variables will not be deleted.

More info

Macro Biddy_DeleteFormula(x) is defined for use with anonymous manager.

Definition at line 8839 of file biddyMainGDD.c.

5.5.2.74 Biddy_Boolean Biddy_Managed_DeleteIthFormula (Biddy_Manager *MNG*, unsigned int *i*)

Function Biddy_Managed_DeleteIthFormula deletes formula from the table.

Description

Formula is labelled but not immediately removed. Nodes of the given formula are not immediately removed.

Side effects

Formula is not accessible by its name anymore. The first two formulae ("0" and "1") will not be deleted. Formulae representing variables will not be deleted.

More info

Macro Biddy_DeleteIthFormula(x) is defined for use with anonymous manager.

Definition at line 8902 of file biddyMainGDD.c.

Here is the caller graph for this function:

5.5.2.75 Biddy_Edge Biddy_Managed_GetIthFormula (Biddy_Manager *MNG*, unsigned int *i*)

Function Biddy_Managed_GetIthFormula returns ith formula in a Formula table.

Description

Return biddyNull if ith formulae does not exist.

Side effects

After addding new formula the index of others may change!

More info

Macro Biddy_GetIthFormula(i) is defined for use with anonymous manager.

Definition at line 8962 of file biddyMainGDD.c.

5.5.2.76 Biddy_String Biddy_Managed_GetIthFormulaName (Biddy_Manager *MNG*, unsigned int *i*)

Function Biddy_Managed_GetIthFormulaName returns name of the ith formula in a Formula table.

Description

Return NULL if ith formulae does not exist.

Side effects

After addding new formula the index of others may change!

More info

Macro Biddy_GetIthFormulaName(i) is defined for use with anonymous manager.

Definition at line 8993 of file biddyMainGDD.c.

5.5.2.77 Biddy_Variable Biddy_Managed_SwapWithHigher (Biddy_Manager *MNG,* Biddy_Variable *v*)

Function Biddy_Managed_SwapWithHigher swaps two adjacent variables.

Description

Higher (greater) variable is the bottommore one! The highest element is constant "1". Constant '1' has global ordering numUsedVariables (not smaller than anyone). Global ordering is the number of zeros in corresponding line of orderingTable.

Side effects

All obsolete nodes will be removed.

More info

Macro Biddy_SwapWithHigher(v) is defined for use with anonymous manager.

Definition at line 9034 of file biddyMainGDD.c.

5.5.2.78 Biddy_Variable Biddy_Managed_SwapWithLower (Biddy_Manager *MNG,* Biddy_Variable *v*)

Function Biddy_Managed_SwapWithLower swaps two adjacent variables.

Description

Lower (smaller) variable is the topmore one! The lowest (topmost) element is not fixed. Topmost variable has global ordering 1 (smaller than all except itself). Global ordering is the number of zeros in corresponding line of orderingTable.

Side effects

All obsolete nodes will be removed.

More info

Macro Biddy_SwapWithLower(v) is defined for use with anonymous manager.

Definition at line 9079 of file biddyMainGDD.c.

5.5.2.79 Biddy_Boolean Biddy_Managed_Sifting (Biddy_Manager *MNG,* Biddy_Edge *f,* Biddy_Boolean *converge*)

Function Biddy_Managed_Sifting reorders variables to minimize node number for the whole system (if f = NULL) or for the given function (if f != NULL) using Rudell's sifting algorithm.

Description

Variables are reordered globally. All obsolete nodes will be removed.

Side effects

More info

Macro Biddy_Sifting(f) is defined for use with anonymous manager.

Definition at line 9123 of file biddyMainGDD.c.

Here is the caller graph for this function:

5.5.2.80 Biddy_Edge Biddy_Managed_Random (Biddy_Manager *MNG,* Biddy_Edge *support,* double *r*)

Function Biddy_Managed_Random generates a random BDD.

Description

The represented Boolean function depends on the variables given with parameter support whilst the parameter r determines the ratio between the number of function's minterms and the number of all possible minterms. Parameter support is a product of positive varaiables.

Side effects

Parameter r must be a number from [0,1]. Otherwise, function returns biddyNull.

More info

Macro Biddy_Random(support,r) is defined for use with anonymous manager.

Definition at line 10446 of file biddyMainGDD.c.

5.5.2.81 Biddy_Edge Biddy_Managed_RandomSet (Biddy_Manager *MNG,* Biddy_Edge *unit,* double *r*)

Function Biddy_Managed_RandomSet generates a random BDD.

Description

The represented set is a random combination set determined by the parameter unit whilst the parameter r determines the ratio between the number of set's subsets and the number of all possible subsets. Parameter set is a set containing only one subset which consist of all elements, i.e. it is a set {{x1,x2,...,xn}}.

Side effects

Parameter r must be a number from [0,1]. Otherwise, function returns biddyNull.

More info

Macro Biddy_RandomSet(unit,r) is defined for use with anonymous manager.

Definition at line 10575 of file biddyMainGDD.c.

5.6 biddyStat.c File Reference

File biddyStat.c contains statistical functions.

```
#include "biddyInt.h"
```

Functions

- unsigned int Biddy_Managed_NodeNumber (Biddy_Manager MNG, Biddy_Edge f)
  Function Biddy_Managed_NodeNumber.
- unsigned int Biddy_NodeMaxLevel (Biddy_Edge f)
  Function Biddy_NodeMaxLevel.
- float Biddy_NodeAvgLevel (Biddy_Edge f)
  Function Biddy_NodeAvgLevel.
- Biddy_Variable Biddy_Managed_VariableTableNum (Biddy_Manager MNG)
  Function Biddy_Managed_VariableTableNum returns number of used variables.
- unsigned int Biddy_Managed_NodeTableSize (Biddy_Manager MNG)
  Function Biddy_Managed_NodeTableSize returns the size of node table.
- unsigned int Biddy_Managed_NodeTableBlockNumber (Biddy_Manager MNG)
  Function Biddy_Managed_NodeTableBlockNumber.
- unsigned int Biddy_Managed_NodeTableGenerated (Biddy_Manager MNG)

Function Biddy_Managed_NodeTableGenerated.

- unsigned int Biddy_Managed_NodeTableMax (Biddy_Manager MNG)

  Function Biddy_Managed_NodeTableMax returns maximal (peek) number of nodes in node table.
- unsigned int Biddy_Managed_NodeTableNum (Biddy_Manager MNG)

  Function Biddy_Managed_NodeTableNum returns number of all nodes currently in node table.
- unsigned int Biddy_Managed_NodeTableNumVar (Biddy_Manager MNG, Biddy_Variable v)

  Function Biddy_Managed_NodeTableNumVar returns number of nodes with a given variable currently in node table.
- unsigned int Biddy_Managed_NodeTableGCNumber (Biddy_Manager MNG)

  Function Biddy_Managed_NodeTableGCNumber.
- unsigned int Biddy_Managed_NodeTableSwapNumber (Biddy_Manager MNG)

  Function Biddy_Managed_NodeTableSwapNumber.
- unsigned int Biddy_Managed_NodeTableSiftingNumber (Biddy_Manager MNG)

  Function Biddy_Managed_NodeTableSiftingNumber.
- unsigned int Biddy_Managed_NodeTableResizeNumber (Biddy_Manager MNG)

  Function Biddy_Managed_NodeTableResizeNumber.
- unsigned int Biddy_Managed_NodeTableITENumber (Biddy_Manager MNG)

  Function Biddy_Managed_NodeTableITENumber.
- unsigned long long int Biddy_Managed_NodeTableITERecursiveNumber (Biddy_Manager MNG)

  Function Biddy_Managed_NodeTableITERecursiveNumber.
- unsigned int Biddy_Managed_NodeTableANDORNumber (Biddy_Manager MNG)

  Function Biddy_Managed_NodeTableANDORNumber.
- unsigned long long int Biddy_Managed_NodeTableANDORRecursiveNumber (Biddy_Manager MNG)

  Function Biddy_Managed_NodeTableANDORRecursiveNumber.
- unsigned int Biddy_Managed_NodeTableXORNumber (Biddy_Manager MNG)

  Function Biddy_Managed_NodeTableXORNumber.
- unsigned long long int Biddy_Managed_NodeTableXORRecursiveNumber (Biddy_Manager MNG)

  Function Biddy_Managed_NodeTableXORRecursiveNumber.
- clock_t Biddy_Managed_NodeTableGCTime (Biddy_Manager MNG)

  Function Biddy_Managed_NodeTableGCTime.
- unsigned long long int Biddy_Managed_NodeTableGCObsoleteNumber (Biddy_Manager MNG)

  Function Biddy_Managed_NodeTableGCObsoleteNumber.
- clock_t Biddy_Managed_NodeTableDRTime (Biddy_Manager MNG)

  Function Biddy_Managed_NodeTableDRTime.
- unsigned int Biddy_Managed_FormulaTableNum (Biddy_Manager MNG)

  Function Biddy_Managed_FormulaTableNum returns number of known formulae.
- unsigned int Biddy_Managed_ListUsed (Biddy_Manager MNG)

  Function Biddy_Managed_ListUsed.
- unsigned int Biddy_Managed_ListMaxLength (Biddy_Manager MNG)

  Function Biddy_Managed_ListMaxLength.
- float Biddy_Managed_ListAvgLength (Biddy_Manager MNG)

  Function Biddy_Managed_ListAvgLength.
- unsigned long long int Biddy_Managed_OPCacheSearch (Biddy_Manager MNG)

  Function Biddy_Managed_OPCacheSearch.
- unsigned long long int Biddy_Managed_OPCacheFind (Biddy_Manager MNG)

  Function Biddy_Managed_OPCacheFind.
- unsigned long long int Biddy_Managed_OPCacheOverwrite (Biddy_Manager MNG)

  Function Biddy_Managed_OPCacheOverwrite.
- unsigned int Biddy_Managed_NodeNumberPlain (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_NodeNumberPlain.
- unsigned int Biddy_Managed_DependentVariableNumber (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_DependentVariableNumber.

- void Biddy_Managed_NodeVarNumber (Biddy_Manager MNG, Biddy_Edge f, unsigned int *n, unsigned int *v)

  Function Biddy_Managed_NodeVarNumber.
- unsigned long long int Biddy_Managed_CountPaths (Biddy_Manager MNG, Biddy_Edge f)

  Function Biddy_Managed_CountPaths count the number of 1-paths.
- double Biddy_Managed_CountMinterm (Biddy_Manager MNG, Biddy_Edge f, unsigned int nvars)

  Function Biddy_Managed_CountMinterm.
- double Biddy_Managed_DensityFunction (Biddy_Manager MNG, Biddy_Edge f, unsigned int nvars)

  Function Biddy_Managed_DensityFunction calculates the ratio of the number of on-set minterms to the number of all minterms.
- double Biddy_Managed_DensityBDD (Biddy_Manager MNG, Biddy_Edge f, unsigned int nvars)

  Function Biddy_Managed_DensityBDD calculates the ratio of the number of on-set minterms to the number of nodes.
- unsigned long long int Biddy_Managed_ReadMemoryInUse (Biddy_Manager MNG)

  Function Biddy_Managed_ReadMemoryInUse report memory consumption of main data strucutures (nodes, node table, variable table, ordering table, formula table, ITE cache, EA cache, RC cache) in bytes.
- void Biddy_Managed_PrintInfo (Biddy_Manager MNG, FILE *f)

  Function Biddy_Managed_PrintInfo prepare a file with stats.

5.6.1 Detailed Description

File biddyStat.c contains statistical functions.

Description

```
PackageName [Biddy]
Synopsis    [Biddy provides data structures and algorithms for the
             representation and manipulation of Boolean functions with
             ROBDDs. A hash table is used for quick search of nodes.
             Complement edges decreases the number of nodes. An automatic
             garbage collection with a system age is implemented.
             Variable swapping and sifting are implemented.]

FileName    [biddyStat.c]
Revision    [$Revision: 252 $]
Date        [$Date: 2017-03-17 23:30:03 +0100 (pet, 17 mar 2017) $]
Authors     [Robert Meolic (robert.meolic@um.si),
             Ales Casar (ales@homemade.net)]
```

Copyright

More info

See also: biddy.h, biddyInt.h

5.6.2 Function Documentation

5.6.2.1 unsigned int Biddy_Managed_NodeNumber (Biddy_Manager *MNG*, Biddy_Edge *f*)

Function Biddy_Managed_NodeNumber.

Description

Count number of nodes in a BDD.

Side effects

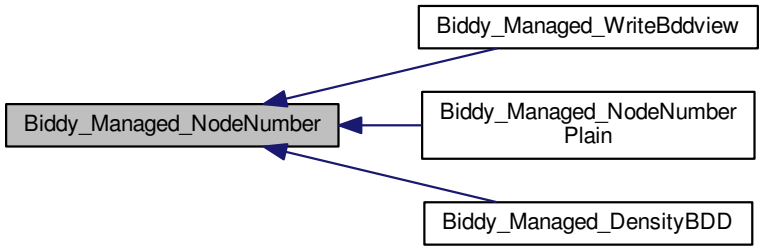
This function must be managed because node selection is used.

More info

Macro Biddy_Managed_NodeNumber(f) is defined for use with anonymous manager.

Definition at line 85 of file biddyStat.c.

Here is the caller graph for this function:



5.6.2.2 unsigned int Biddy_NodeMaxLevel (Biddy_Edge *f*)

Function Biddy_NodeMaxLevel.

Description

Side effects

More info

Definition at line 119 of file biddyStat.c.

5.6.2.3 float Biddy_NodeAvgLevel (Biddy_Edge *f*)

Function Biddy_NodeAvgLevel.

Description

Side effects

More info

Definition at line 147 of file biddyStat.c.

5.6.2.4 Biddy_Variable Biddy_Managed_VariableTableNum (Biddy_Manager *MNG*)

Function Biddy_Managed_VariableTableNum returns number of used variables.

Description

Side effects

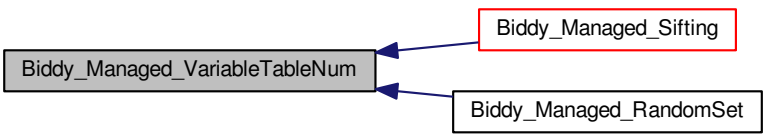
Variable '1' is included.

More info

Macro Biddy_VariableTableNum() is defined for use with anonymous manager.

Definition at line 179 of file biddyStat.c.

Here is the caller graph for this function:



5.6.2.5 unsigned int Biddy_Managed_NodeTableSize (Biddy_Manager *MNG*)

Function Biddy_Managed_NodeTableSize returns the size of node table.

Description

Side effects

More info

Macro Biddy_NodeTableSize() is defined for use with anonymous manager.

Definition at line 204 of file biddyStat.c.

5.6.2.6 unsigned int Biddy_Managed_NodeTableBlockNumber (Biddy_Manager *MNG*)

Function Biddy_Managed_NodeTableBlockNumber.

Description

Side effects

More info

Macro Biddy_NodeTableBlockNumber() is defined for use with anonymous manager.

Definition at line 234 of file biddyStat.c.

5.6.2.7 unsigned int Biddy_Managed_NodeTableGenerated (Biddy_Manager *MNG*)

Function Biddy_Managed_NodeTableGenerated.

Description

Side effects

More info

Macro Biddy_NodeTableGenerated() is defined for use with anonymous manager.

Definition at line 259 of file biddyStat.c.

5.6.2.8 unsigned int Biddy_Managed_NodeTableMax (Biddy_Manager *MNG*)

Function Biddy_Managed_NodeTableMax returns maximal (peek) number of nodes in node table.

Description

Side effects

More info

Macro Biddy_NodeTableMax() is defined for use with anonymous manager.

Definition at line 285 of file biddyStat.c.

5.6.2.9 unsigned int Biddy_Managed_NodeTableNum (Biddy_Manager *MNG*)

Function Biddy_Managed_NodeTableNum returns number of all nodes currently in node table.

Description

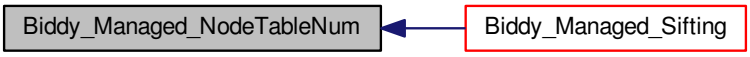
Side effects

More info

Macro Biddy_NodeTableNum() is defined for use with anonymous manager.

Definition at line 311 of file biddyStat.c.

Here is the caller graph for this function:



5.6.2.10 unsigned int Biddy_Managed_NodeTableNumVar (Biddy_Manager *MNG*, Biddy_Variable *v*)

Function Biddy_Managed_NodeTableNumVar returns number of nodes with a given variable currently in node table.

Description

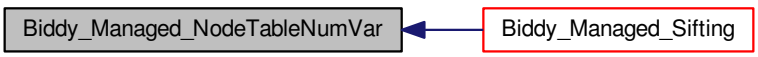
Side effects

More info

Macro Biddy_NodeTableNumVar(v) is defined for use with anonymous manager.

Definition at line 337 of file biddyStat.c.

Here is the caller graph for this function:



5.6.2.11 unsigned int Biddy_Managed_NodeTableGCNumber (Biddy_Manager *MNG*)

Function Biddy_Managed_NodeTableGCNumber.

Description

Side effects

More info

Macro Biddy_NodeTableGCNumber() is defined for use with anonymous manager.

Definition at line 362 of file biddyStat.c.

5.6.2.12 unsigned int Biddy_Managed_NodeTableSwapNumber (Biddy_Manager *MNG*)

Function Biddy_Managed_NodeTableSwapNumber.

Description

Side effects

More info

Macro Biddy_NodeTableSwapNumber() is defined for use with anonymous manager.

Definition at line 387 of file biddyStat.c.

5.6.2.13 unsigned int Biddy_Managed_NodeTableSiftingNumber (Biddy_Manager *MNG*)

Function Biddy_Managed_NodeTableSiftingNumber.

Description

Side effects

More info

Macro Biddy_NodeTableSiftingNumber() is defined for use with anonymous manager.

Definition at line 413 of file biddyStat.c.

5.6.2.14 unsigned int Biddy_Managed_NodeTableResizeNumber (Biddy_Manager *MNG*)

Function Biddy_Managed_NodeTableResizeNumber.

Description

Side effects

More info

Macro Biddy_NodeTableResizeNumber() is defined for use with anonymous manager.

Definition at line 439 of file biddyStat.c.

5.6.2.15 unsigned int Biddy_Managed_NodeTableITENumber (Biddy_Manager *MNG*)

Function Biddy_Managed_NodeTableITENumber.

Description

Side effects

More info

Macro Biddy_NodeTableITENumber() is defined for use with anonymous manager.

Definition at line 464 of file biddyStat.c.

5.6.2.16 unsigned long long int Biddy_Managed_NodeTableITERecursiveNumber (Biddy_Manager *MNG*)

Function Biddy_Managed_NodeTableITERecursiveNumber.

Description

Side effects

Recursive ITE calls are counted only if Biddy is compiled using directive BIDDYEXTENDEDSTATS_YES.

More info

Macro Biddy_NodeTableITERecursiveNumber() is defined for use with anonymous manager.

Definition at line 492 of file biddyStat.c.

5.6.2.17 unsigned int Biddy_Managed_NodeTableANDORNumber (Biddy_Manager *MNG*)

Function Biddy_Managed_NodeTableANDORNumber.

Description

Side effects

More info

Macro Biddy_NodeTableANDORNumber() is defined for use with anonymous manager.

Definition at line 524 of file biddyStat.c.

5.6.2.18 unsigned long long int Biddy_Managed_NodeTableANDORRecursiveNumber (Biddy_Manager *MNG*)

Function Biddy_Managed_NodeTableANDORRecursiveNumber.

Description

Side effects

Recursive AND/OR calls are counted only if Biddy is compiled using directive BIDDYEXTENDEDSTATS_YES.

More info

Macro Biddy_NodeTableANDORRecursiveNumber() is defined for use with anonymous manager.

Definition at line 552 of file biddyStat.c.

5.6.2.19 unsigned int Biddy_Managed_NodeTableXORNumber (Biddy_Manager *MNG*)

Function Biddy_Managed_NodeTableXORNumber.

Description

Side effects

More info

Macro Biddy_NodeTableXORNumber() is defined for use with anonymous manager.

Definition at line 583 of file biddyStat.c.

5.6.2.20 unsigned long long int Biddy_Managed_NodeTableXORRecursiveNumber (Biddy_Manager *MNG*)

Function Biddy_Managed_NodeTableXORRecursiveNumber.

Description

Side effects

Recursive XOR calls are counted only if Biddy is compiled using directive BIDDYEXTENDEDSTATS_YES.

More info

Macro Biddy_NodeTableXORRecursiveNumber() is defined for use with anonymous manager.

Definition at line 611 of file biddyStat.c.

5.6.2.21 clock_t Biddy_Managed_NodeTableGCTime (Biddy_Manager *MNG*)

Function Biddy_Managed_NodeTableGCTime.

Description

Side effects

More info

Macro Biddy_NodeTableGCTime() is defined for use with anonymous manager.

Definition at line 642 of file biddyStat.c.

5.6.2.22 unsigned long long int Biddy_Managed_NodeTableGCObsoleteNumber (Biddy_Manager *MNG*)

Function Biddy_Managed_NodeTableGCObsoleteNumber.

Description

Return the number of nodes deleted by GC.

Side effects

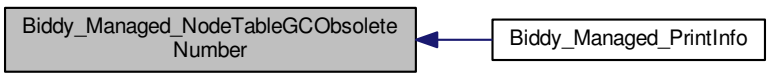
Obsolete nodes deleted by GC are counted only if Biddy is compiled using directive BIDDYEXTENDEDSTATS_↩ YES.

More info

Macro Biddy_NodeTableGCObsoleteNumber() is defined for use with anonymous manager.

Definition at line 671 of file biddyStat.c.

Here is the caller graph for this function:



5.6.2.23 clock_t Biddy_Managed_NodeTableDRTime (Biddy_Manager *MNG*)

Function Biddy_Managed_NodeTableDRTime.

Description

Side effects

More info

Macro Biddy_NodeTableDRTime() is defined for use with anonymous manager.

Definition at line 710 of file biddyStat.c.

5.6.2.24 unsigned int Biddy_Managed_FormulaTableNum (Biddy_Manager *MNG*)

Function Biddy_Managed_FormulaTableNum returns number of known formulae.

Description

Side effects

Formulae '0' and '1' are included.

More info

Macro Biddy_FormulaTableNum() is defined for use with anonymous manager.

Definition at line 736 of file biddyStat.c.

5.6.2.25 unsigned int Biddy_Managed_ListUsed (Biddy_Manager *MNG*)

Function Biddy_Managed_ListUsed.

Description

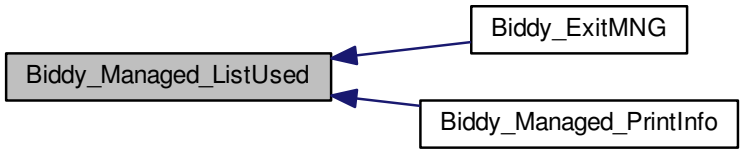
Side effects

More info

Macro Biddy_ListUsed() is defined for use with anonymous manager.

Definition at line 761 of file biddyStat.c.

Here is the caller graph for this function:



5.6.2.26 unsigned int Biddy_Managed_ListMaxLength (Biddy_Manager *MNG*)

Function Biddy_Managed_ListMaxLength.

Description

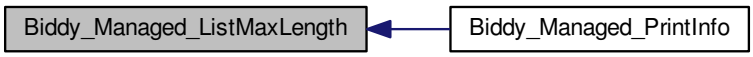
Side effects

More info

Macro Biddy_ListMaxLength() is defined for use with anonymous manager.

Definition at line 794 of file biddyStat.c.

Here is the caller graph for this function:



5.6.2.27 float Biddy_Managed_ListAvgLength (Biddy_Manager *MNG*)

Function Biddy_Managed_ListAvgLength.

Description

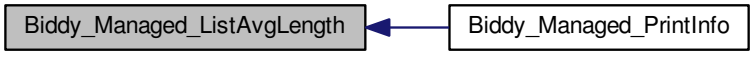
Side effects

More info

Macro Biddy_ListAvgLength() is defined for use with anonymous manager.

Definition at line 858 of file biddyStat.c.

Here is the caller graph for this function:



5.6.2.28 unsigned long long int Biddy_Managed_OPCacheSearch (Biddy_Manager *MNG*)

Function Biddy_Managed_OPCacheSearch.

Description

Side effects

More info

Macro Biddy_OPCacheSearch() is defined for use with anonymous manager.

Definition at line 900 of file biddyStat.c.

5.6.2.29 unsigned long long int Biddy_Managed_OPCacheFind (Biddy_Manager *MNG*)

Function Biddy_Managed_OPCacheFind.

Description

Side effects

More info

Macro Biddy_OPCacheFind() is defined for use with anonymous manager.

Definition at line 925 of file biddyStat.c.

5.6.2.30 unsigned long long int Biddy_Managed_OPCacheOverwrite (Biddy_Manager *MNG*)

Function Biddy_Managed_OPCacheOverwrite.

Description

Side effects

More info

Macro Biddy_OPCacheOverwrite() is defined for use with anonymous manager.

Definition at line 950 of file biddyStat.c.

5.6.2.31 unsigned int Biddy_Managed_NodeNumberPlain (Biddy_Manager *MNG*, Biddy_Edge *f*)

Function Biddy_Managed_NodeNumberPlain.

Description

Count number of nodes in a corresponding BDD without complement edges.

Side effects

More info

Macro Biddy_Managed_NodeNumberPlain(f) is defined for use with anonymous manager.

Definition at line 976 of file biddyStat.c.

5.6.2.32 unsigned int Biddy_Managed_DependentVariableNumber (Biddy_Manager *MNG*, Biddy_Edge *f*)

Function Biddy_Managed_DependentVariableNumber.

Description

Count number of dependent variables. For OBDD, the number of dependent variables is the same as the number of variables in the graph. For ZBDD and TZBDD, this is not true.

Side effects

More info

Macro Biddy_DependentVariableNumber(f) is defined for use with anonymous manager.

Definition at line 1026 of file biddyStat.c.

Here is the caller graph for this function:

5.6.2.33 void Biddy_Managed_NodeVarNumber (Biddy_Manager *MNG*, Biddy_Edge *f*, unsigned int * *n*, unsigned int * *v*)

Function Biddy_Managed_NodeVarNumber.

Description

Count number of nodes and number of variables existing in the graph. For OBDD, the number of variables existing in the graph is the same as the number of dependent variables. For ZBDD and TZBDD, this is not true.

Side effects

More info

Macro Biddy_NodeVarNumber(f,n,v) is defined for use with anonymous manager.

Definition at line 1099 of file biddyStat.c.

Here is the caller graph for this function:

5.6.2.34 unsigned long long int Biddy_Managed_CountPaths (Biddy_Manager *MNG,* Biddy_Edge *f*)

Function Biddy_Managed_CountPaths count the number of 1-paths.

Description

Side effects

Implemented for OBDD, ZBDD, and TZBDD. TO DO: implement this using GNU Multiple Precision Arithmetic Library (GMP).

More info

Macro Biddy_CountPaths(f) is defined for use with anonymous manager.

Definition at line 1151 of file biddyStat.c.

5.6.2.35 double Biddy_Managed_CountMinterm (Biddy_Manager *MNG,* Biddy_Edge *f,* unsigned int *nvars*)

Function Biddy_Managed_CountMinterm.

Description

Parameter nvars is a user-defined number of dependent variables. If nvars == 0 then number of variables existing in the graph is used. For combination sets, this function coincides with combination counting.

Side effects

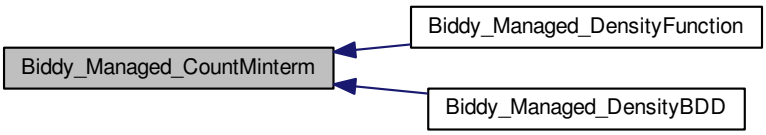
We are using GNU Multiple Precision Arithmetic Library (GMP). For ZBDDs, this function coincides with 1-path cou

More info

Macro Biddy_CountMinterm(f,nvars) is defined for use with anonymous manager. Macros Biddy_Managed_Count↩ Combination(MNG,f,nvars) and Biddy_CountCombination(f,nvars) are defined for use with combination sets.

Definition at line 1202 of file biddyStat.c.

Here is the caller graph for this function:



5.6.2.36 double Biddy_Managed_DensityFunction (Biddy_Manager *MNG*, Biddy_Edge *f*, unsigned int *nvars*)

Function Biddy_Managed_DensityFunction calculates the ratio of the number of on-set minterms to the number of all minterms.

Description

If nvars == 0 then number of dependent variables is used.

Side effects

More info

Macro Biddy_DensityFunction(f,nvars) is defined for use with anonymous manager.

Definition at line 1328 of file biddyStat.c.

5.6.2.37 double Biddy_Managed_DensityBDD (Biddy_Manager *MNG*, Biddy_Edge *f*, unsigned int *nvars*)

Function Biddy_Managed_DensityBDD calculates the ratio of the number of on-set minterms to the number of nodes.

Description

If nvars == 0 then number of dependent variables is used.

Side effects

More info

Macro Biddy_DensityBDD(f,nvars) is defined for use with anonymous manager.

Definition at line 1382 of file biddyStat.c.

5.6.2.38 unsigned long long int Biddy_Managed_ReadMemoryInUse (Biddy_Manager *MNG*)

Function Biddy_Managed_ReadMemoryInUse report memory consumption of main data strucutures (nodes, node table, variable table, ordering table, formula table, ITE cache, EA cache, RC cache) in bytes.

Description

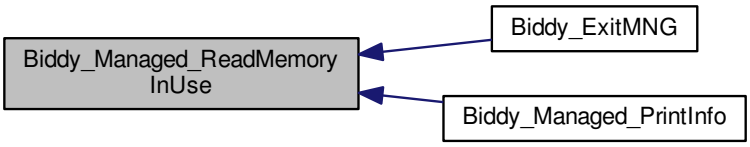
Side effects

More info

Macro Biddy_ReadMemoryInUse() is defined for use with anonymous manager.

Definition at line 1428 of file biddyStat.c.

Here is the caller graph for this function:



5.6.2.39 void Biddy_Managed_PrintInfo (Biddy_Manager *MNG*, FILE * *f*)

Function Biddy_Managed_PrintInfo prepare a file with stats.

Description

Side effects

More info

Macro Biddy_PrintInfo(f) is defined for use with anonymous manager.

Definition at line 1514 of file biddyStat.c.

Index