Compiling, porting and modifying MAS

The Modula-2 Algebra System
Version of 9 March 1998 for MAS Version 1.0

Heinz Kredel
Michael Pesch

Heinz Kredel
Universität Mannheim
D-68131 Mannheim
Germany
kredel@rz.uni-mannheim.de

Michael Pesch
Universität Passau
D-94030 Passau
Germany
pesch@alice.fmi.uni-passau.de

Available from the MAS WWW page (http://alice.fmi.uni-passau.de/mas.html)

1 Introduction

MAS, the Modula-2 Algebra System, is an experimental computer algebra system. MAS combines imperative programming facilities with algebraic specification capabilities for design and study of algebraic algorithms. It contains a large library of implemented up-to-date Gröbner basis algorithms for nearly all algebraic structures where such methods exist. MAS further includes algorithms for real quantifier elimination, parametric real root counting, and for computing in (commutative and noncommutative) polynomial rings.

This manual explains how to compile the MAS sources. Moreover, it gives some hints for those who whish to modify the MAS source or port MAS to another architecture. However, this manual does not explain the usage of MAS. For this purpose refer to *MAS, Modula-2 Algebra System, Interactive Usage* which is available from the MAS WWW page (`http://alice.fmi.uni-passau.de/mas.html`)

2 Obtaining MAS and its Prerequisites

The place to look for MAS is the MAS WWW page (`http://alice.fmi.uni-passau.de/mas.html`) or the MAS ftp site (`ftp://alice.fmi.uni-passau.de/pub/mas`). There you will find the MAS documentation, the sourcecode of MAS and binaries for various architectures.

The sourcecode can be found following the URL `ftp://alice.fmi.uni-passau.de/pub/mas/mas-1.01`

Currently available binaries are:

`ftp://alice.fmi.uni-passau.de/pub/mas/mas-hppa1.1-hp-hpux9.03-1.00.tar.gz`
: MAS binary for HPPA running HPUX

`ftp://alice.fmi.uni-passau.de/pub/mas/mas-i386-unknown-os2-1.00.tar.gz`
: MAS binary for Intel PC running OS/2

`ftp://alice.fmi.uni-passau.de/pub/mas/mas-i486-unknown-linux-1.00.tar.gz`
: MAS binary for Intel PC running LINUX

`ftp://alice.fmi.uni-passau.de/pub/mas/mas-mab-next-nextstep3-1.00.tar.gz`
: MAS multi-architecture binary for Nextstep

`ftp://alice.fmi.uni-passau.de/pub/mas/mas-rs6000-ibm-aix3.2.5-1.00.tar.gz`
: MAS binary for RS/6000 running AIX 3

`ftp://alice.fmi.uni-passau.de/pub/mas/mas-sparc-sun-sunos4.1.3C-1.00.tar.g`
: MAS binary for Sun Sparc running SunOS

The documentation has the URLs:

MAS,Modula-2 Algebra System, Specifications, Definition Modules, Indexes
(`ftp://alice.fmi.uni-passau.de/pub/mas/masdef.ps.gz`)
MAS,Modula-2 Algebra System, Interactive Usage
(`ftp://alice.fmi.uni-passau.de/pub/mas/mastut.ps.gz`)
this manual (`ftp://alice.fmi.uni-passau.de/pub/mas/mascomp.ps.gz`)

Patches for some prerequisites are available:

`ftp://alice.fmi.uni-passau.de/pub/ComputerAlgebraSystems/mas/makemake-patches.gz`
: Patches for the Makemake program.

`ftp://alice.fmi.uni-passau.de/pub/ComputerAlgebraSystems/mas/mtc-patches.gz`
: Patches for the Modula-2 to C translator MTC.

`ftp://alice.fmi.uni-passau.de/pub/ComputerAlgebraSystems/mas/reuse-patches.gz`
: Patches for the Reuse library.

Moreover, all the tools mentioned in this manual are also available from this site.

Send bug reports, remarks and questions about MAS to `mas@alice.fmi.uni-passau.de`.

3 Invoking MAS

MAS can be run from the command line by entering `mas`. This will start an interactive MAS session. Its general usage is `mas [`*Options*`] [`*File*`]` where *File* is a MAS input file. The default extension for such files is '`.mas`'.

The following *Options* are available:

`--help` display help and exit

`--version`
: display version information and exit

`-C`
`--copyright`
: display copyright and copying conditions

`-c [`*COMMAND*`]`
`--command[=`*COMMAND*`]`
: execute MAS command *COMMAND*

`-e`
`--exit` exit after all input files have been read

`-E`
`--exit-on-error`
: exit if an error occurs

`-f` *FILE*
`--file=`*FILE*
: input initialization file *FILE* instead of default

`-m` *SIZE*
`--memorysize=`*SIZE*
: size of MAS memory in kbytes

`-o[`*FILE*`]`
`--output[=`*FILE*`]`
: write output to *FILE*

`-R`
`--no-readline`
: do not use the readline library

An interactive session can be terminated by typing `EXIT`.

While MAS is running, sending the `SIGUSR1` signal (by entering `kill -USR1` *pid-of-mas*) to MAS will cause MAS to print information about its current state.

MAS reads an initialization file called '`~/.bashrc`'. If this file does not exists, a warning message will be printed, which can be ignored.

4 Compiling MAS

To compile MAS simply run `configure` and then `gnumake`. Ideally, this will be all you have to do. Otherwise, read on.

4.1 Prerequisites

It may happen, that not all the prerequisites (the MODULA to C compiler mtc for instance) are available on your machine. Chapter 2 [Obtaining MAS and its Prerequisites], page 3, for information about obtaining the prerequisites.

To compile MAS you must have a Bourne shell (bash is recommended), GNU make, mtc, the reuse library and an ANSI-C compiler (gcc is recommended) available. We recommend further to have the readline and kpathsea libraries available. If you want to modifify and then compile MAS you will also need GNU configure, awk and sed. These prerequisites will be described in the following.

4.1.1 mtc

The MAS source code is written in the Modula-2 programming language. To compile MAS you need a Modula-2 compiler. The recommended one (which is the one we actually use) is Mtc. Mtc is the Modula-2 to C Translator from the Gesellschaft fuer Mathematik und Datenverarbeitung (German National Research Center for Computer Science) Forschungsstelle fuer Programmstrukturen an der Universität Karlsruhe. It translates the Modula-2 sources to C. The C sources can then be compiled with some ANSI-C Compiler like gcc.

Mtc can be obtained from the GMD ftp site (`ftp://ftp.gmd.de`).

To compile mtc just follow the installation instructions that come with it. We recommend too choose the '`m2c` ' (not the '`src`') directory when this choice is required. Additionally the files '`SYSTEM_.c`' and '`SYSTEM_.h`' (which contain information like the bit size of an `int`) may need to be adapted to your machine. The modified files we use are available from `http://alice.fmi.uni-passau.de/mas.html, the MAS WWW page`. Note that, due to the way the MAS garbage collector works, `StackAlloc` must be defined in '`SYSTEM_.h`' (and `alloca()` must be available on your system, which usually is the case).

4.1.2 Reuse Library

MAS uses many functions from the Reuse Modula-2 library. To compile MAS you must have this library and the associated header files available. Like Mtc the library has been developed by the GMD and is available from the GMD ftp site (`ftp://ftp.gmd.de`).

Configure will try to find the Reuse library and fail if it can not find it. If this happens on a system where Reuse is available, it means that Reuse is not correctly installed. To get around this before running `configure` set the environment variable `reuse_prefix` to a directory such the library file '`libreuse.a`' can be found in '`$reuse_prefix/lib`' the corresponding header ('`*.h`') files can be found in in '`$reuse_prefix/lib/include`' and the corresponding Modula-2 definition

module ('*.md') files can be found in '$reuse_prefix/lib/reuse'. Alternatively you may set the environment variable reuse_a to the location of the library file, reuse_h to the location of the include files and reuse_md to the location of the definition module files.

To compile the Reuse library , goto the m2c directory, and edit the Makefile, as needed. The files 'SYSTEM_.c', 'SYSTEM_.h.', 'System.c' may need to be edited too (the first two are the same as for Mtc). Then run make.

4.1.3 GNU Readline Library

Optionally, MAS can use the GNU readline library for more comfortable interactive input. It can be obtained from the GNU ftp site (ftp://prep.ai.mit.edu).

Configure will automatically set up MAS to use readline, if readline is found. Otherwise MAS will be compiled without readline. If readline is present on your system but not found by configure, you may (before running configure) set the environment variable readline_prefix to a directory where the library file 'libreadline.a' can be found in '$readline_prefix/lib' and the corresponding header ('*.h') files can be found in in '$readline_prefix/include'. Alternatively you may set the environment variable readline_a to the location of the library file and readline_h to the location of the include files.

4.1.4 Kpathsearch Library

Optionally, MAS can use the kpathsea library by K. Berry for finding input files, etc.. It is part of the TEX distribution by K. Berry and can be obtained from the TUG ftp site (ftp://ftp.tug.org).

Configure will automatically set up MAS to use kpathsea, if kpathsea is found. Otherwise MAS will be compiled without kpathsea. If kpathsea is present on your system but not found by configure, you may (before running configure) set the environment variable kpathsea_prefix to a directory where the library file 'libkpathsea.a' can be found in '$kpathsea_prefix/lib' and the corresponding header ('*.h') files can be found in in '$kpathsea_prefix/include'. Alternatively you may set the environment variable kpathsea_a to the location of the library file and kpathsea_h to the location of the include files.

4.1.5 Makemake

If you want to rebuild the Makefiles, you need Makemake to generate the dependencies for the Modula-2 files. Makemake comes with Mtc and can be compiled following the instructions there.

4.2 Compiling

Unpack the MAS source code. Then create a new directory and go there. The next steps are running configure and gnumake.

4.2.1 Running Configure

For information about the options of configure see the file 'INSTALL'.

Normally, running `configure` will do. See Section 4.1.2 [Reuse Library], page 7, Section 4.1.3 [GNU Readline Library], page 8, or Section 4.1.4 [Kpathsearch Library], page 8, if these libraries are not found by configure.

4.2.2 Running Gnumake

Normally, running `gnumake` will create a MAS binary and the libraries. To create an optimized version of MAS, run `gnumake mas-opt`.

If you want do recreate the dependecies in the Makefiles (e.g. because you added new source files), use `gnumake MKMK=y`.

4.3 Miscellaneous

4.3.1 Code Optimization

The usual optimization options of the C compiler (the `-O` flag) must not be used when compiling MAS. The reason is that one of the optimizations compilers will usually do is to place variables in registers. Unfortunately, the MAS garbage collector will not check these registers for valid adresses.

Nevertheless, the Gnu C Compiler, gcc, allows to select the optimizations seperately. All optimizations not involving registers can be applied. Configure will set the necessary options automatically, if you are using gcc.

Table of Contents