NAME

bcg_io – convert graphs from and into the BCG format

SYNOPSIS

bcg_io [*bcg_options_1*] [*input_options*] *input_filename* [*bcg_options_2*] [*output_options*] *output_filename*

DESCRIPTION

This command takes as input a graph contained in *input_filename* and generates as output a graph in *output_filename*. This command is mostly used to perform format conversion, in the case where *input_filename* and *output_filename* are not encoded in the same graph formats.

The following ‘‘input formats’’ are currently implemented and available for *input_filename*:

- BCG	Binary Coded Graphs format
- AUT	ALDEBARAN graph format
- FC2	ESPRIT project CONCUR2 format (INRIA, Sophia)
- SEQ	CADP common trace format

The following ‘‘output formats’’ are currently implemented and available for *output_filename*:

- BCG	Binary Coded Graphs format
- ASCII	ASCII dump of the BCG format
- AUT	ALDEBARAN graph format
- AUTO	AUTO and MEIJE graph format (INRIA, Sophia)
- CWB	Concurrency Workbench (LFCS, Edinburgh)
- ETMCC	Erlangen-Twente Markov Chain Checker
- FC2	ESPRIT project CONCUR2 format (INRIA, Sophia)
- GML	Graph Modelling Language (Univ. Passau)
- GRAPHVIZ	DOT format of the GRAPHVIZ tools (ATT)
- LOTOS	pseudo-LOTOS format
- MEC	MEC graph format (LaBRI, Bordeaux)
- PIPN	PIPN graph format (LAAS, Toulouse)
- SCAN	SCAN format (BULL)
- SEQ	CADP common trace format
- SQUIGGLES	SQUIGGLES graph format (CNUCE, Pisa)
- VCG	Vizualization of Compiler Graphs (Univ. Saarlandes)
- VISCOPE	VISCOPE graph format (IRISA, Rennes)
- XESAR	XESAR graph format (LGI-IMAG, Grenoble)

Conversions are allowed from any input format into any output format.

When converting from the BCG format into a non-BCG output format, the dynamic library corresponding to *input_filename* may be generated if necessary.

When converting a non-BCG input format into a non-BCG output format, a BCG graph and its dynamic library may be generated as intermediate forms. This is normally transparent to the end-user.

Conversion from an input format to itself are allowed, but are useless except in the case of the BCG format: compression parameters (see below) can be modified this way.

GENERAL OPTIONS

Two groups of general options, *bcg_options_1* and *bcg_options_2*, are currently supported.

Options *bcg_options_1* can appear at the beginning of the command line. These options are: **-version**, **-create**, **-update**, **-remove**, **-cc**, **-tmp**.

Options *bcg_options_2* can appear on the command line after *input_filename*. These options are: **-uncompress**, **-compress**, **-register**, **-short**, **-medium**, and **-size**.

See the **bcg**(LOCAL) manual page for a description of these options.

PARTICULAR OPTIONS

bcg_io is very flexible with respect to options and file suffixes (i.e., file extensions). Options or file suffixes can be omitted, provided that there is no ambiguity. **bcg_io** does its best to guess the user's intentions and automatically supplies the missing options and suffixes.

An option can be omitted iff the file suffix is present and, reciprocally, a file suffix can be omitted iff the option is present.

If *input_file* is equal to "-", then it is considered to be the standard input. In such case, the option must be present.

If *output_file* is equal to "-", then it is considered to be the standard output. In such case, the option must be present.

Note: if the output option "-" is given for producing the BCG, FC2, or XESAR formats, the standard output of the **bcg_io** should be a regular file (not a pipe), because these formats require either that an explicit output file name is given (XESAR), or that the file can be accessed randomly using the **lseek**(2) system call (BCG and FC2).

For example, the conversion of an AUT file into a FC2 file is normally done as follows:
  bcg_io -aldebaran *input_file***.aut -fc2** *output_file***.fc2**
but equivalent synopses are allowed, such as:
  bcg_io -aldebaran *input_file* **-fc2** *output_file*
or:
  bcg_io *input_file***.aut** *output_file***.fc2**
or (assuming that *output_file***.fc2** is a regular file):
 bcg_io -aldebaran - -fc2 - <*input_file***.aut >***output_file***.fc2**

If *output_file* is omitted and has only its suffix mentioned, it is assumed to be equal to *inputfile*. For example:
  bcg_io *filename***.aut .fc2**
is equivalent to:
  bcg_io *filename***.aut** *filename***.fc2**

If the input and output files are both in BCG format, e.g.:
  bcg_io *input_file***.bcg** *output_file***.bcg**
then the input file, if encoded using an old version of the BCG format, is systematically converted to the latest version of the BCG format and stored in the output file.

Finally, uncompressing or compressing a BCG file can be done as follows:
  bcg_io *input_file***.bcg -uncompress** *output_file***.bcg**
or
 bcg_io *input_file***.bcg -compress -size 2 2 4** *output_file***.bcg**

The following associations of options and filenames are currently available, for input and/or output:

-bcg *input_filename[***.bcg***]*

Read *input_filename***.bcg** encoded in the BCG graph format.

–bcg *[* **-parse | -unparse***]* *output_filename[***.bcg***]*

Write *output_filename***.bcg** encoded in the BCG graph format. General options **-uncompress**, **-compress**, **-register**, **-short**, **-medium**, and **-size** can be used to control the contents of *output_filename***.bcg**. Options **-parse** and **-unparse** can be used to control label parsing in *output_filename***.bcg** (see the **bcg_write**(LOCAL) manual page for a technical discussion about label parsing). By default, or if option **-parse** is present, label parsing is enabled. If option **-unparse** is present, label parsing is disabled. If the input file is also in BCG format, options **-parse** and **-unparse** will have no effect, as the status of label parsing used in the input BCG file will be preserved in *output_filename***.bcg**.

–ascii *[* **-small***] output_filename[* **.ascii***]*

Write *output_filename***.ascii** encoded in the ASCII dump format. If option **-small** is present, do not display the contents of the state area, the edge area, and the class area. By default, these areas are displayed.

-auto *output_filename[* **.m0***]*

Write *output_filename***.m0** encoded in the AUTO graph format.

–aldebaran *input_filename[* **.aut***]*

Read *input_filename***.aut** encoded in the AUT graph format (see the **aut**(LOCAL) manual page for a description of this format).

–aldebaran *output_filename[* **.aut***]*

Write *output_filename***.aut** encoded in the AUT graph format.

-cwb *output_filename[* **.cwb***]*

Write *output_filename***.cwb** encoded in the CWB graph format.

-etmcc *[***-format** *format_string] output_filename[***.tra***]*

Write *output_filename***.tra** encoded in the ETMCC graph format. If option **-format** *format_string* is present, it specifies the form under which floating-point numbers are printed to the output file See the **bcg_min**(LOCAL) and **determinator**(LOCAL) manual pages for a detailed description of *format_string*. By default, i.e., if option **-format** is absent, the default value of *format_string* is `"%g"`.

-fc2 *[***-net** *number] input_filename[***.fc2***]*

Read *input_filename***.fc2** encoded in the FC2 graph format. If option **-net** *number* is present, select the *number*-th automaton contained in *input_filename***.fc2**. If option **-net** is not specified, *number* is given the default 0.

-fc2 *[***-verbose***] output_filename[***.fc2***]*

Write *output_filename***.fc2** encoded in the FC2 graph format. If option **-verbose** is present, use the verbose form of FC2 (by default, the compact form of FC2 is used).

–gml *output_filename[* **.gml***]*

Write *output_filename***.gml** encoded in the GML graph format.

–graphviz *output_filename[* **.dot***]*

Write *output_filename***.dot** encoded in the DOT graph format of the GRAPHVIZ tools.

–lotos *output_filename[* **.lotos***]*

Write *output_filename***.lotos** in pseudo-LOTOS format.

–mec *output_filename[* **.mec***]*

Write *output_filename***.mec** encoded in the MEC graph format.

–pipn *output_filename[***.auto.pro***]*

Write *output_filename***.auto.pro** encoded in the PIPN graph format.

–scan *output_filename[***.scan***]*

Write *output_filename***.scan** encoded in the SCAN graph format.

–sequence *input_filename[***.seq***]*

Read *input_filename***.seq** encoded in the simple SEQ format (see the **seq**(LOCAL) manual page for a description of this format).

–sequence *output_filename[***.seq***]*

Write *output_filename***.seq** encoded in the simple SEQ format. Translating a graph to the SEQ format is only possible if the graph has no circuits and if all its states (with the possible exception of the initial state) have at most one outgoing edge.

–squiggles *output_filename[***.graph***]*

Write *output_filename***.graph** encoded in the SQUIGGLES graph format.

–vcg *output_filename[***.vcg***]*

Write *output_filename***.vcg** encoded in the VCG graph format.

–viscope *output_filename[***.trans***]*

Write *output_filename***.trans** encoded in the VISCOPE graph format.

-xesar *[***-old***] output_filename[***.gra***]*

Write *output_filename***.gra** encoded in the XESAR graph format. Three auxiliary files (*output_filename***.dp3**, *output_filename***.ge3**, and *output_filename***.tai**) are also generated. If option **-old** is present, use the old XESAR format (this format is now obsolete and does not work for graphs with more than 65536 states). By default, the new XESAR format is used.

NOTES

The letters ‘‘io’’ in **bcg_io** stand for input/output.

ENVIRONMENT VARIABLES

See the **bcg**(LOCAL) manual page for a description of the environment variables used by all the BCG application tools.

EXIT STATUS

Exit status is 0 if everything is alright, 1 otherwise.

AUTHORS

Renaud Ruffiot, Hubert Garavel, Radu Mateescu, and David Jacquemin.

ACKNOWLEDGEMENTS

Acknowledgements are due to Patrick Ernberg (SICS, Sweden) for suggesting a conversion scheme to the CWB format.

OPERANDS

filename.**bcg**	BCG graph (input or output)
filename@**1.o**	dynamic library (input or output)
filename.**ascii**	ASCII dump (output)
filename.**m0**	AUTO graph (output)
filename.**aut**	AUT graph (input or output)
filename.**cwb**	CWB graph (output)
filename.**tra**	ETMCC graph (output)
filename.**fc2**	FC2 graph (input or output)
filename.**gml**	GML graph (output)
filename.**dot**	GRAPHVIZ graph (output)
filename.**lotos**	pseudo-LOTOS (output)
filename.**mec**	MEC graph (output)
filename.**auto.pro**	PIPN graph (output)
filename.**scan**	SCAN graph (output)
filename.**seq**	SEQ trace (input or output)
filename.**graph**	SQUIGGLES graph (output)
filename.**vcg**	VCG graph (output)
filename.**trans**	VISCOPE graph (output)
filename.**gra**	XESAR graph (output)
filename.**dp3**	XESAR graph (output)
filename.**ge3**	XESAR graph (output)
filename.**tai**	XESAR graph (output)

FILES

$CADP/bin.‘arch‘/bcg_io	“**bcg_io**” binary program
$CADP/bin.‘arch‘/libBCG_IO.a	“**bcg_io**” static library #1
$CADP/bin.‘arch‘/libbcg_iodyn.a	“**bcg_io**” static library #2

See the **bcg**(LOCAL) manual page for a description of the other files.

SEE ALSO

aut(LOCAL), **bcg**(LOCAL), **bcg_min**(LOCAL), **bcg_write**(LOCAL), **determinator**(LOCAL), **exhibitor**(LOCAL), **seq**(LOCAL)

Additional information is available from the CADP Web page located at http://cadp.inria.fr

Directives for installation are given in files **$CADP/INSTALLATION_∗.**

Recent changes and improvements to this software are reported and commented in file **$CADP/HISTORY.**

BUGS

Please report bugs to Hubert.Garavel@inria.fr