

**NAME**

**bcg\_graph** – generate various kinds of useful BCG graphs

**SYNOPSIS**

**bcg\_graph** [*bcg\_options*] **–chaos** [*labelfile* | **–labels** *number pattern1*] [**–monitor**] [**–verbose**] *output* [.bcg]

or

**bcg\_graph** [*bcg\_options*] **–bag** *size* [*labelfile* | **–labels** *number pattern1 pattern2*] [**–monitor**] [**–verbose**] *output* [.bcg]

or

**bcg\_graph** [*bcg\_options*] **–fifo** *size* [*labelfile* | **–labels** *number pattern1 pattern2*] [**–monitor**] [**–verbose**] *output* [.bcg]

**DESCRIPTION**

This command produces a BCG graph named *output.bcg* the structure of which is specified by the command-line options. At present, three kinds of graphs can be generated using this command: chaos automata, bag buffers, and FIFO buffers (see below for detailed definitions).

Note: Such particular graphs could be easily described in a high-level language such as LOTOS or LNT, and then translated to BCG. However, the **bcg\_graph** tool contains specialized algorithms designed for time and memory efficiency and ensuring that each generated graph is minimal wrt to branching bisimulation.

Note: Unless specified otherwise, **bcg\_graph** generates graphs that are minimal for strong equivalence (i.e., that do not contain redundant states or transitions). However, this property is only ensured if the specified labels are pairwise different.

**GENERAL OPTIONS**

The following *bcg\_options* are currently supported: **–version**, **–tmp**, **–uncompress**, **–compress**, **–register**, **–short**, **–medium**, and **–size**. See the **bcg(LOCAL)** manual page for a description of these options.

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

**PARTICULAR OPTIONS**

The following options and arguments are also supported:

**–chaos**

Generate a chaos automaton. This option must occur immediately after *bcg\_options* on the command-line.

**–bag** *size*

Generate a bag buffer with *size* places (with *size*  $\geq 0$ ). This option must occur immediately after *bcg\_options* on the command-line.

**–fifo** *size*

Generate a FIFO buffer with *size* places (with *size*  $\geq 0$ ). This option must occur immediately after *bcg\_options* on the command-line.

#### *labelfile*

Specify that the labels of the generated graph are listed in file *labelfile* containing an exhaustive list of labels, one label per line. See Section LABEL DEFINITIONS below for details.

#### **-labels** *number pattern1* [*pattern2*]

Specify that the labels of the generated graph are defined by *pattern1* (and possibly *pattern2*) instantiated by integers in the range  $[1..number]$  (with *number*  $\geq 0$ ). See Section LABEL DEFINITIONS below for a definition of patterns. This option is forbidden if *labelfile* is given and mandatory otherwise.

#### **-monitor**

Open a window for monitoring the graph generation in real-time. Not a default option.

#### **-verbose**

Display the number of states and transitions of the generated graph. Not a default option.

### CHAOS AUTOMATA

Chaos automata (**-chaos** option) accept all messages in a given set of messages.

A chaos automaton is determined by the number *N* (with *N*  $\geq 0$ ) of different messages it can handle. It has a single state and *N* looping transitions, each labelled by a different message.

### BAG BUFFERS

Bag buffers (**-bag** option) model communication media that do not lose messages but do not preserve the ordering of messages (messages are reset in any order).

A bag buffer is determined by its size *P* (with *P*  $\geq 0$ ), which is the maximal number of messages that can be stored into the buffer, and the number *N* (with *N*  $\geq 0$ ) of different messages (there are *N* labels corresponding to message inputs and *N* labels corresponding to message outputs; see the notions of get-label and put-label below).

The number of states of a bag buffer is equal to

$$(N+P)! / (N! * P!)$$

The number of transitions of a bag buffer is equal to

$$\text{if } N=0 \text{ or } P=0 \text{ then } 0 \text{ else } 2*(N+P-1)!/(N-1)!*(P-1)!$$

### FIFO BUFFERS

FIFO buffers (**-fifo** option) model communication media that do not lose messages and preserve the ordering of messages (messages are resent in the same order as they were received).

A FIFO buffer is determined by its size *P* (with *P*  $\geq 0$ ), which is the maximal number of messages that can be stored into the buffer, and the number *N* (with *N*  $\geq 0$ ) of different messages (there are *N* labels corresponding to message inputs and *N* labels corresponding to message outputs; see the notions of get-label and put-label below).

The number of states of a FIFO buffer is equal to

$$\sum_{k \text{ in } 0..P} N^k$$

or also

if  $N=1$  then  $P+1$  else  $(N^{P+1}-1)/(N-1)$   
 The number of transitions of a FIFO buffer is equal to  
 $2 * N * \sum_{k \text{ in } 0..P-1} N^k$   
 or also  
 if  $N=1$  then  $2 * P$  else  $2 * N * (((N^P)-1)/(N-1))$

## LABEL DEFINITIONS

For some forms of graphs (such as *-chaos*), a single list of labels is needed.

For other forms of graphs (such as *-bag* and *-fifo*), two lists of labels are needed: the get-labels (corresponding to input messages) and the put-labels (corresponding to output messages). There is a pairwise correspondence between both lists, as each put-label emitted corresponds to a get-label received previously.

### A. LABELS DEFINED USING A FILE

A *labelfile* contains a list of labels separated with newline characters. Labels are strings of characters without newline characters. Labels can be enclosed between double quotes, which will be ignored, and may contain spaces. For instance, **A**, **"B"**, and **"A !1 !CONS(3, 4)"** are valid labels. In the label definition file, lines that contain only spaces (including empty lines) are ignored. On each line, leading and trailing spaces are ignored, unless they are enclosed in quotes.

For graphs requiring two lists of labels, the position of each label in the file determines whether it is a put-label or a get-label. In the *labelfile*, pairs of corresponding labels alternate strictly. Each get-label occurs on an odd-numbered line and is followed by its corresponding put-label on an even-numbered line (empty lines notwithstanding). Therefore, there must be an even number of labels in *labelfile*.

The degenerate case in which *labelfile* contains no label (e.g., the file is empty) is permitted and no warning will be issued.

### B. LABELS DEFINED USING PATTERNS

A *pattern* is a character string to be used as the *format* argument of the C function **printf()**. It should contain exactly one occurrence of **"%d"** (and no occurrence of other **%**-arguments such as **"%c"**, **"%f"**, etc.). If needed, the **"%"** character can be specified as **"%%"**.

Patterns are used to generate labels, the **"%d"** argument being substituted by an integer in the range **[1..number]**. If *number* is null, the generated set of labels is empty.

For graphs requiring two lists of labels, *pattern1* defines the get-labels and *pattern2* defines the put-labels; each put-label is associated with the get-label instantiated with the same number.

## EXAMPLES

- **"bcg\_graph -fifo 2 labelfile output.bcg"** will generate a graph named **output.bcg** modeling a 2-place FIFO buffer whose get- and put-labels are defined in **labelfile**.
- **"bcg\_graph -chaos -labels 4 "A%d" output.bcg"** will generate a graph named **output.bcg** consisting of a single state and four transitions labeled **A1** through **A4**.
- **"bcg\_graph -bag 2 -labels 3 "GET \!%d" "PUT \!%d" output.bcg"** will generate a graph named **output.bcg** modeling a 2-place bag buffer whose labels are named **GET !1, PUT !1, GET !2, PUT !2, GET !3, and PUT !3** respectively. Note: the **"!"** characters are escaped with backslashes in order to avoid problems with csh/tcsh history substitution features.

## LIMITATIONS

**bcg\_graph** cannot generate graphs with more than  $2^{32}$  transitions (note that the number of transitions is always larger than the number of states).

Moreover, if the available memory is insufficient, **bcg\_graph** might be unable to generate large graphs (even with less than  $2^{32}$  transitions).

**bcg\_graph** does not check if the specified labels are pairwise different.

#### 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

Frederic Tronel, Frederic Lang, Hubert Garavel, and David Champelovier

#### OPERANDS

<i>labelfile</i>	Label definition file (optional input)
<i>output.bcg</i>	BCG graph (output)

#### FILES

**\$CADP/bin.'arch'/bcg\_graph**      “**bcg\_graph**” binary program

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

#### SEE ALSO

**bcg(LOCAL)**, **bcg\_info(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 [cadp@inria.fr](mailto:cadp@inria.fr)