NAME

bcg_cmp – equivalence comparison of normal, probabilistic, or stochastic labeled transitions systems (LTS) encoded in the BCG format

SYNOPSIS

bcg_cmp [*bcg_options*] [**-strong** | **-branching** | **-divbranching** | **-observational**] [**-normal** | **-prob** | **-rate** [**-self**]] [**-epsilon** *eps*] [**-format** *format_string*] [**-class** *class_file*] [**-diag** *diag_file***.bcg**] *input_1***.bcg** *input_2***.bcg**

where *bcg_options* is defined below (see GENERAL OPTIONS).

bcg_cmp takes as inputs the BCG graphs *input_1***.bcg** and *input_2***.bcg**, and compares these graphs according to some bisimulation relation.

DESCRIPTION

bcg_cmp implements various algorithms to perform comparison of graphs encoded in the BCG format according to strong bisimulation, branching bisimulation, divergence-sensitive branching bisimulation, or observational equivalence. A graph input by **bcg_cmp** can be:

- either a "normal" LTS, whose transitions are either labelled with "normal" labels or with the "internal" label (usually noted "tau" in the scientific literature and displayed as the character string "**i**" by the various BCG tools),

- or a "probabilistic LTS": these are LTS with "normal" labelled transitions, as well as "special" transitions, whose labels are either of the form "**prob %p**" or "*label*; **prob %p**", where **%p** denotes a floating-point number in the range]0..1] and *label* denotes a character string that does not contain the "**;**" character. For each state, the sum of "**%p**" values on special transitions leaving the state must be less or equal than 1 (see ANNEX 1 of the **bcg_min**(LOCAL) manual page for a discussion on how the probabilistic LTS model generalizes other theoretical models published in the literature),

- or a "stochastic LTS": these are LTS with "normal" labelled transitions, as well as "special" transitions, whose labels are either of the form "**rate %f**" or "*label*; **rate %f**", where **%f** denotes a strictly positive floating-point number and *label* denotes a character string that does not contain the "**;**" character (see ANNEX 2 of the **bcg_min**(LOCAL) manual page for a discussion on how the stochastic LTS model of **bcg_min** generalizes other theoretical models published in the literature).

GENERAL OPTIONS

The following *bcg_options* are currently supported: **-version**, **-create**, **-update**, **-remove**, **-cc**, **-tmp**, **-uncompress**, **-compress**, **-register**, **-short**, **-medium**, and **-size**. See the **bcg**(LOCAL) manual page for a description of these options.

PARTICULAR OPTIONS

The following options are also supported:

-strong

Perform LTS comparison according to strong bisimulation. On (Discrete or Continuous Time) Markov Chains and on Markov Reward Models, this equivalence agrees with lumpability of [KS76] (see ANNEX 1, ANNEX 2, and BIBLIOGRAPHY of the **bcg_min**(LOCAL) manual page). Default option.

-branching

Perform LTS comparison according to branching bisimulation. It is worth noticing that the notion of branching bisimulation is rather meaningless for probabilistic systems. Not a default option.

-divbranching

Perform LTS comparison according to divergence-sensitive branching bisimulation [GW96]. Divergence-sensitive branching bisimulation differs from branching bisimulation only in the way cycles of internal transitions (also called divergences) are treated. It is known that all states traversed by a cycle of internal transitions belong to the same branching equivalence class. While divergences are ignored by ordinary branching bisimulation, they are assimilated to a self-looping internal transition in each such equivalence class by divergence-sensitive branching bisimulation. Unlike branching bisimulation, divergence-sensitive branching bisimulation preserves inevitability properties. Like branching bisimulation, it is worth noticing that the notion of divergence-sensitive branching bisimulation is rather meaningless for probabilistic systems. Not a default option.

-observational

Perform LTS comparison according to observational equivalence [Mil89]. It is worth noticing that observational equivalence is computationally more expensive than branching bisimulation, so that comparison may fail even for graphs containing only few thousands of states. To reduce the risk of failure, in a first step the input graphs are automatically reduced according to branching bisimulation. This is sound because branching bisimulation is a graph relation stronger than observational equivalence. However, this optimisation is not applied if the **-class** option is set, so that **bcg_cmp** can print the equivalence classes relatively to the states of the input graphs, instead of the states of the branching minimal intermediate graph produced in the first step. This option cannot be combined with neither **-prob** nor **-rate** options. Not a default option.

-normal

Consider *input_1***.bcg** and *input_2***.bcg** as normal LTSs. With this option, labels of the form "**`rate %f`**" or "*label*`; rate %f`" or "**`prob %p`**" or "*label*`; prob %p`" are processed as ordinary labels, without special meaning attached. Default option.

-prob

Consider *input_1***.bcg** and *input_2***.bcg** as probabilistic LTSs. With this option, each label of the form "**`prob %p`**" or "*label*`; prob %p`" is recognized as denoting a probabilistic transition with probability **`%p`**. **bcg_cmp** will stop with an error message if, for some probabilistic transition, **`%p`** is out of]0..1], or if the probabilistic transitions going out of the same state have a cumulated sum strictly greater than 1. With this option, labels of the form "**`rate %f`**" or "*label*`; rate %f`" are processed as ordinary labels. Not a default option.

-rate [**-self**]

Consider *input_1***.bcg** and *input_2***.bcg** as stochastic LTSs. With this option, each label of the form "**`rate %f`**" or "*label*`; rate %f`" is recognized as denoting a stochastic transition with rate **`%f`**. **bcg_cmp** will stop with an error message if, for some stochastic transition, **`%f`** is less or equal to 0. If the **-branching** or the **-divbranching** option is selected, and some state has both an outgoing stochastic transition and an outgoing internal (i.e., "tau") transition, **bcg_cmp** will print a warning and ignore the stochastic transition in order to preserve the notion of maximal progress. With this option, labels of the form "**`prob %p`**" or "*label*`; prob %p`" are processed as ordinary labels. Not a default option.

If **-self** sub-option is given, all self loops (i.e., transitions that remain within the same equivalence class) having labels of the form "**`rate %f`**" are ignored. This implements the weak Markovian equivalences described in [Bra02] and [BHKW05]. Not a default sub-option.

-epsilon *eps*

Set the precision of floating-point comparisons to *eps*, where *eps* is a real value. When *eps* is out of [0..1[, **bcg_cmp** reports an error. Default value for *eps* is 1E-6.

-format *format_string*

Use *format_string* to control the format of the floating-point numbers contained in transition labels (these numbers correspond to the occurrences of **`%f`** and **`%p`** mentioned in section DESCRIPTION above). The value of *format_string* should obey the same conventions as the *format* parameter of function **sprintf**(3C) for values of type **`double`**. Default value for *format_string* is **`"%g"`**, meaning that floating-point numbers are printed with at most six digits after the "." (i.e., the radix character). Other values can be used, for instance **`"%.9g"`** to obtain nine digits instead of six, or by replacing the **`"%g"`** flag with other flags, namely **`"%e"`**, **`"%E"`**, **`"%f"`**, **`"%G"`**, possibly combined with additional flags (e.g., to specify precision).

-class *class_file*

If *class_file* is the character '-', then display the equivalence classes on standard output. Otherwise, display the equivalence classes in a file named *class_file*. Not a default option.

-diag *diag_file*.**bcg**

When the comparison of *input_1* and *input_2* yields FALSE, generate a diagnostic (counterexample) in BCG format (see the **bcg**(LOCAL) manual page for details) explaining this result. The diagnostic is generated in the file *diag*.**bcg**. This option has no effect when the comparison of input_1 **and** *input_2* **yields TRUE, since in this case the diagnostic would be larger than** *input_1* **and** *input_2***, and would not bring any useful information. The BCG file containing the diagnostic can be visualized using the bcg_draw**(LOCAL) and **bcg_edit**(LOCAL) tools of CADP (see respective manual pages for details).

The diagnostic is a directed acyclic graph included (modulo the preorder corresponding to the equivalence relation considered) both in *input_1* and *input_2*.

If the diagnostic is a sequence of transitions, it will also be displayed on standard output using the simple SEQ format (see the **seq**(LOCAL) manual page for the definition of this format). Not a default option.

Note: Options **-strong**, **-branching**, and **-divbranching** are mutually exclusive. If they occur simultaneously on the command-line, the option occurring last is selected.

Note: Options **-normal**, **-prob**, and **-rate** (with or without **-self** sub-option) are mutually exclusive. If they occur simultaneously on the command-line, the option occurring last is selected.

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

bcg_cmp was developped by Frederic Lang (INRIA/CONVECS). It shares a substantial amount of code with **bcg_min**. See the **bcg_min**(LOCAL) manual page for more information.

OPERANDS

input_1.**bcg**	BCG graph (input)
input_2.**bcg**	BCG graph (input)
input_1@**1.o**	dynamic library (input or output)
input_2@**1.o**	dynamic library (input or output)

FILES

$CADP/bin.'arch'/bcg_cmp	"**bcg_cmp**" binary program

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

SEE ALSO

bcg(LOCAL), **bcg_min**(LOCAL), **seq**(LOCAL), **sprintf**(3C)

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.