NAME

determinator − elimination of nondeterminism for stochastic systems

SYNOPSIS

bcg_open [*bcg_opt*] *spec*[**.bcg**] [*cc_opt*] **determinator** [*determinator_opt*] *result*[**.bcg**]

or:

exp.open [*exp_opt*] *spec*[**.exp**] [*cc_opt*] **determinator** [*determinator_opt*] *result*[**.bcg**]

or:

fsp.open [*fsp_opt*] *spec*[**.lts**] [*cc_opt*] **determinator** [*determinator_opt*] *result*[**.bcg**]

or:

lnt.open [*lnt_opt*] *spec*[**.lnt**] [*cc_opt*] **determinator** [*determinator_opt*] *result*[**.bcg**]

or:

lotos.open [*lotos_opt*] *spec*[**.lotos**] [*cc_opt*] **determinator** [*determinator_opt*] *result*[**.bcg**]

or:

seq.open [*seq_opt*] *spec*[**.seq**] [*cc_opt*] **determinator** [*determinator_opt*] *result*[**.bcg**]

DESCRIPTION

Taking as input an extended Markovian model expressed either as a BCG graph *spec***.bcg**, a composition expression *spec***.exp**, an FSP program *spec***.lts**, a LNT program *spec***.lnt**, a LOTOS program *spec***.lotos**, or a SEQ file *spec***.seq**, **determinator** generates a reduced model by removing stochastic nondeterminism on-the-fly.

Extended Markovian models are state-transition models containing ordinary, probabilistic and/or stochastic transitions. Stochastic determinization consists in trying to convert the extended Markovian model *spec* into a continuous-time Markov chain (CTMC) by removing local sources of nondeterminism. It might fail if *spec* does not satisfy certain conditions. If it succeeds, the CTMC is written to *result***.bcg**; otherwise, an error message is issued. See section STOCHASTIC DETERMINIZATION below for details.

Note: Since March 2006, determinization of ordinary Labelled Transition Systems is no longer supported by **determinator**. Option **-rate** becomes the default option. The formerly available options **-normal** and **-tauclosure** are now deprecated. Use the **reductor**(LOCAL) tool instead, as explained below.

OPTIONS

The options *bcg_opt*, if any, are passed to **bcg_lib**(LOCAL).

The options *exp_opt*, if any, are passed to **exp.open**(LOCAL).

The options *fsp_opt*, if any, are passed to **fsp.open**(LOCAL).

The options *lnt_opt*, if any, are passed to **lnt.open**(LOCAL).

The options *lotos_opt*, if any, are passed to **caesar**(LOCAL) and to **caesar.adt**(LOCAL).

The options *seq_opt*, if any, are passed to **seq.open**(LOCAL).

The options *cc_opt*, if any, are passed to the C compiler.

The following options *determinator_opt* are currently available:

-epsilon *eps*
Set the precision of certain floating-point comparisons to *eps*, where *eps* is a real value. When *eps* is out of [0..1[, **determinator** 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 STOCHASTIC DETERMINIZATION below). 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"`, or `"%G"`, possibly combined with additional flags (e.g., to specify precision).

–hide [–total | –partial | –gate] *hiding_filename*
Use the hiding rules defined in *hiding_filename* to hide (on the fly) the labels of the CTMC being generated. See the **caesar_hide_1**(LOCAL) manual page for a detailed description of the appropriate format for *hiding_filename*.

The **-total**, **-partial**, and **-gate** options specify the "total matching", "partial matching", and "gate matching" semantics, respectively. See the **caesar_hide_1**(LOCAL) manual page for more details about these semantics. Option **-total** is the default.

–rename [–total|–single|–multiple|–gate] *renaming_filename*
Use the renaming rules defined in *renaming_filename* to rename (on the fly) the labels of the CTMC being generated. See the **caesar_rename_1**(LOCAL) manual page for a detailed description of the appropriate format for *renaming_filename*.

The **-total**, **-single**, **-multiple**, and **-gate** options specify the "total matching", "single partial matching", "multiple partial matching", and "gate matching" semantics, respectively. See the **caesar_rename_1**(LOCAL) manual page for more details about these semantics. Option **-total** is the default.

As for the **bcg_labels**(LOCAL) tool, several hiding and/or renaming options can be present on the command-line, in which case they are processed from left to right.

-monitor
Open a window for monitoring in real-time the generation of *result*.**bcg**. Not a default option.

-uncompress, -compress, -register, -short, -medium, -size
These options control the form under which the BCG graph *result*.**bcg** is generated. See the **bcg**(LOCAL) manual page for a description of these options.

-tmp This option specifies the directory in which temporary files are to be stored. See the **bcg**(LOCAL) manual page for a description of this option.

DEPRECATED OPTIONS

-rate This option is supported for backward compatibility but has no effect.

-normal

This option triggers an error message. Use ‘‘**reductor -trace**’’ instead of ‘‘**determinator -normal**’’.

-tauclosure

This option triggers an error message. Use ‘‘**reductor -weaktrace**’’ instead of ‘‘**determinator -normal -tauclosure**’’.

STOCHASTIC DETERMINIZATION

The input of **determinator** is an extended Markovian model combining features from discrete-time and continuous-time Markov chains. All transition labels must have one of the following forms:

- "`rate` *%f*" (called a stochastic transition), or
- "*label*`; rate` *%f*" (called a labelled stochastic transition), or
- "`prob` *%p*" (called a probabilistic transition), or
- "*label*`; prob` *%p*" (called a labelled probabilistic transition), or
- "*label*" (called an ordinary transition).

where `%f` denotes a strictly positive floating-point number, `%p` denotes a floating-point number in the range]0..1], and *label* denotes a character string that does not contain the "`;`" character (*label* may be equal to the internal action, often noted "i" or "tau").

On the opposite, the expected output of **determinator** is a continuous-time Markov chain, i.e., a model containing stochastic transitions only.

See also **bcg_min**(LOCAL) for a discussion about the various probabilistic and stochastic models present in the literature.

States are classified as follows:

- A state is called *decision* if it is the source state of at least one ordinary transition.
- A state is called *vanishing* if it is not decision and the source state of at least one (possibly labelled) probabilistic transition.
- A state is called *tangible* if it is neither decision nor vanishing.

Note: the **bcg_steady**(LOCAL) and **bcg_transient**(LOCAL) tools rely on the same notions of vanishing and tangible states, but do not have to consider decision states as they do not accept ordinary transitions.

In order to be accepted by **determinator**, the input model must satisfy two conditions (otherwise, **determinator** will emit an error message and stop):

- No decision state can be the source state of a (possibly labelled) probabilistic transition.
- The model may not contain cycles of ordinary and/or (possibly labelled) probabilistic transitions; consequently, it necessarily contains at least one tangible state.

Note that if there exists an ordinary transition or a (labelled) probabilistic transition from a state *S1* to a state *S2*, then all (labelled) stochastic transitions from *S1*, if any, are discarded, thus expressing that ordinary and probabilistic transitions are instantaneous.

Note: The sum of **`%p`** values on (possibly labelled) probabilistic transitions leaving a vanishing state needs not be equal to 1; if this sum is different from 1, then probabilistic values will be normalized (i.e., divided by this sum) during determinization.

The stochastic determinization algorithm used in **determinator** is a variant of Deavours-Sanders' algorithm [DS99]. In a nutshell, it starts from the initial state of the input model and recursively explores tangible states as follows. When in a tangible state *S1*, the algorithm inspects all states *S2* reachable from *S1* by following one single (labelled) stochastic transition, the rate of which will be noted **`%f`**:

- If *S2* is a tangible state, then the transition from *S1* to *S2* is kept in *result*.**bcg**, and *S2* will be explored recursively;
- If *S2* is a decision or a vanishing state, then the algorithm checks a local confluence property, namely, for each tangible state *S3* reachable following only ordinary and probabilistic transitions, that the probability **`%p`** to reach *S3* from *S2* does not depend on the choice of the ordinary transitions followed. If so, a new stochastic transition from *S1* to *S3* with rate **`%p*%f`** is added to *result*.**bcg**, and *S3* will be explored recursively. If not, **determinator** stops with an error message.

Note: if the initial state *S0* is not tangible, and if one single tangible state *S* is reachable from *S0* by following ordinary and/or probabilistic transitions only, then *S* will form the initial state of *result*.**bcg**. Otherwise, for each tangible state *S* reachable from *S0* by following ordinary and/or probabilistic transitions only, a probabilistic transition from *S0* to *S* (labelled with the probability to reach *S*) will be created; this is the only case where *result*.**bcg** will contain a vanishing state, i.e., the only case where **determinator** does not produce a continuous-time Markov chain, strictly speaking.

EXIT STATUS

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

DIAGNOSTICS

When the source is erroneous, error messages are issued.

AUTHORS

The first version of the stochastic determinization was written by Christophe Joubert (INRIA/VASY) and Holger Hermanns (Saarland University and University of Twente). Frederic Lang (INRIA/VASY) deeply revised the code. Frederic Lang and Hubert Garavel (both at INRIA/VASY) wrote the current **determinator** manual page.

OPERANDS

spec.**bcg**	BCG graph (input)
spec.**exp**	network of communicating LTSs (input)
spec.**lts**	FSP specification (input)
spec.**lnt**	LNT specification (input)
spec.**lotos**	LOTOS specification (input)
spec.**seq**	SEQ file (input)
result.**bcg**	BCG graph (output)

SEE ALSO

OPEN/CAESAR Reference Manual, **bcg**(LOCAL), **bcg_min**(LOCAL), **bcg_open**(LOCAL), **bcg_steady**(LOCAL), **bcg_transient**(LOCAL), **caesar**(LOCAL), **caesar.adt**(LOCAL), **exp**(LOCAL), **exp.open**(LOCAL), **fsp.open**(LOCAL), **lnt.open**(LOCAL), **lotos**(LOCAL), **lotos.open**(LOCAL), **seq**(LOCAL), **seq.open**(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

BIBLIOGRAPHY

[DS99] D. Deavours and W. Sanders. An Efficient Well-Specified Check. In Proceedings of the International Workshop on Petri Nets and Performance Models (PNPM'99), pages 124-133. IEEE Computer Society Press, 1999.

[HJ03] H. Hermanns and Ch. Joubert. A Set of Performance and Dependability Analysis Components for CADP. In Proceedings of TACAS'2003, LNCS 2619, pages 425-430, Springer Verlag. Available from http://cadp.inria.fr/publications/Hermanns-Joubert-03.html