NAME

exp.open – OPEN/CAESAR connection for EXP networks of communicating automata

SYNOPSIS

exp.open [**-branching** | **-deadpreserving** | **-strong** | **-weaktrace**] [**-case**] [**-debug**] [**-lotos** | **-elotos** | **-ccs** | **-csp** | **-mcrl**] [**-hidden** *string*] [**-termination** *string*] [**-coaction** *string*] [**-depend**] [**-history**] [**-info**] [**-inline**] [**-interface** *interface_directives*] [**-interfaceuser**] [**-labels**] [**-network** *format*] [**-nocheck**] [**-prob**] [**-rate**] [**-silent** | **-verbose**] [**-unparse**] [**-version**] *filename*[**.exp**] [*cc_options*] *prog*[**.a**|**.c**|**.o**] [*prog_options*]

DESCRIPTION

Taking as input *filename***.exp,** which describes a network of communicating automata in the EXP 2.0, see **exp**(LOCAL), and an OPEN/CAESAR program *prog*[**.a**|**.c**|**.o**], **exp.open** generates an OPEN/CAESAR graph module *filename***.c**. This file is then compiled into *filename***.o** and an executable program *prog* resulting from the combination of *filename***.o** and *prog*[**.a**|**.c**|**.o**] is produced. Finally, *prog* is executed.

According to the principles of the OPEN/CAESAR architecture, *prog* is obtained by combining three different modules:

- the graph module is generated from *filename***.exp**
- the storage module is the standard OPEN/CAESAR library
- the exploration module is *prog*[**.a**|**.c**|**.o**]

PROCESSING OF THE EXPLORATION MODULE

The exploration module *prog*[**.a**|**.c**|**.o**] is supposed to contain an OPEN/CAESAR application program, such as **evaluator**(LOCAL), **generator**(LOCAL), **ocis**(LOCAL)...

The exploration module can be supplied in three different forms. It can be either an archive file (with **.a** suffix), or a source C program (with **.c** suffix) or an object code file (with **.o** suffix).

If *prog***.a** is not present in the current directory, **exp.open** attempts to fetch it in the OPEN/CAESAR binary library **$CADP/bin.‘arch‘**.

If *prog***.c** is not present in the current directory, **exp.open** attempts to fetch it in the OPEN/CAESAR source library **$CADP/src/open_caesar**.

If *prog***.o** is not present in the current directory, **exp.open** attempts to fetch it in the OPEN/CAESAR binary library **$CADP/bin.‘arch‘**.

If no suffix (**.a**, **.c**, **.o**) is specified on the command line for the exploration module *prog*, **exp.open** will make successive attempts to fetch this exploration module: first, as a source C program with **.c** suffix; then as an archive file with **.a** suffix; finally as an object code file with **.o** suffix.

OPTIONS

-branching

Perform on-the-fly partial order reduction modulo branching bisimulation. This yields a generally smaller graph, which is equivalent modulo branching bisimulation to the graph obtained using the **-strong** option. The used technique is based on prioritization of so-called *tau-confluent* transitions [Pace-Lang-Mateescu-03]. This is not a default option.

If the **-branching** option is used in combination with **-rate**, then also attempt on-the-fly partial order reduction modulo stochastic branching bisimulation (which is weaker than branching bisimulation), by giving priority to hidden actions over stochastic transitions (see the **-rate** option

below), thus taking an account of the maximal progress of hidden actions.

-case Force the distinction between lowercase and uppercase characters in labels occurring within the operators used in *filename***.exp**. This is the default option if no reference language is selected or if the reference language is E-LOTOS or mCRL. In other cases, labels occurring within the operators used in *filename***.exp** are automatically turned to uppercase. Therefore, labels in LTSs should also be uppercase, except possibly the strings representing the hidden label, termination label, co-action prefix, and the "prob" and "rate" keywords used to denote probabilistic and stochastic transitions (see the **-prob** and **-rate** options below).

-ccs Set CCS as the reference language. This is not a default option. See Section LANGUAGE PARAMETERS of **exp**(LOCAL) for details.

-coaction *string*
Set *string* so as to prefix CCS co-action labels; *string* is named *co-action prefix*. See Section CCS PARALLEL COMPOSITION of **exp**(LOCAL) for more information about the co-action prefix.

-csp Set CSP as the reference language. This is not a default option. See Section LANGUAGE PARAMETERS of **exp**(LOCAL) for details.

-deadpreserving
Perform on-the-fly partial order reduction preserving deadlocks. This yields a generally smaller graph, which contains the same deadlocks as the graph obtained using the **-strong** option. This is not a default option.

-debug Undocumented option.

-depend
Display the list of EXP files included (directly or transitively) in *filename***.exp**, followed by the list of communicating automata, hide, rename, and cut files used in the EXP behaviour and stop. This list may be incomplete if the EXP behaviour is syntactically incorrect. Not a default option.

-elotos Set E-LOTOS as the reference language. This is not a default option. See Section LANGUAGE PARAMETERS of **exp**(LOCAL) for details.

-hidden *string*
Set *string* as denoting the hidden label in BCG files of both the communicating automata and of the automaton corresponding to their composition. The default value depends on the reference language, see Section LANGUAGE PARAMETERS of **exp**(LOCAL) for details.

Note that many CADP tools (such as for instance **bcg_min**(LOCAL), **aldebaran**(LOCAL), etc.) require the hidden label to be written "i". If it is written differently, e.g., "tau", then one may use the "-hidden i" option and hide "tau" in each communicating automaton, by using the hiding operator of EXP 2.0.

Note also that the hidden label is usually written "tau" in the FC2 format. During conversion from FC2 communicating automata into BCG, "tau" labels are automatically renamed into "i" by the **bcg_io**(LOCAL) tool. Therefore, since bcg_io is systematically called to translate FC2 components into the BCG format, the hidden label should be set to "i", using "-hidden i", even though some component is in the FC2 format, with "tau" denoting the hidden label.

-history

Record a history of each label. The history can be read using the CAESAR_INFORMATION_LABEL function of the OPEN/CAESAR API. With the **-history** option, it is possible to set FORMAT_LABEL (see the OPEN/CAESAR manual) to a natural number up to 3 (instead of 2 otherwise):

o The behaviour of CAESAR_INFORMATION_LABEL with FORMAT_LABEL set to 0 or 1 is described in the OPEN/CAESAR documentation.

o If FORMAT_LABEL is equal to 2, then information about the synchronisations involved in the computation of each label is displayed under the form of a *synchronisation vector*.

o If FORMAT_LABEL is equal to 3, then the displayed synchronisation vector is extended with information about hidings and renamings performed to produce the label.

This is not a default option.

-info Print structural information about the LTSs referenced in *filename***.exp** and stop. See **bcg_info**(LOCAL) for more information.

-inline Generate an OPEN/CAESAR graph module that does not depend on BCG files. This option cannot be combined with **-branching**, **-deadpreserving**, **-weaktrace**, and/or the priority operator. Debugging option, not available in official releases of CADP.

-interface *interface_directives*

This option allows to generate a refined interface as explained in the article [Lang-06].

This option assumes that the composition of LTSs stored in *filename***.exp** corresponds to a system of concurrent processes *S* as follows: The concurrent architecture of *filename***.exp** is the same as the concurrent architecture of *S*, and each LTS in *filename***.exp** represents either the state space (named concrete LTS in the sequel) or simply the set of labels (named abstract LTS in the sequel) of the corresponding process in *S*; States and transitions of abstract LTSs are irrelevant.

Consider processes *P0, P1, ..., Pm* of *S*, such that, in *filename***.exp**, the LTS corresponding to *P0* is abstract and the LTSs corresponding to *P1, ..., Pm* are concrete. The **-interface** option allows to synthesize an interface representing the synchronization constraints imposed on *P0* by *P1, ..., Pm*. This interface has the form of an OPEN/CAESAR graph module stored in a file named *filename***.c** and a list of synchronisation labels stored in a file named *filename***.sync**. The graph module can be translated into an explicit LTS using the **generator**(LOCAL) tool. The resulting LTS can then be given, together with *filename***.sync**, to the **projector**(LOCAL) tool so as to restrict the behaviour of *P0*.

The *interface_directives* argument has the form *"nat:nat_list"*, where *nat* is a natural number and *nat_list* is a list of natural numbers separated by blank characters. Each of these natural numbers is an index corresponding to the rank of occurrence of an LTS in *filename***.exp** (once eventual **.exp** file names have been substituted by the expression stored in the corresponding **.exp** files). Index 1 represents the leftmost LTS. The left-hand side of ":" is the index of the LTS corresponding to *P0*. The right-hand side of ":" is the list of indices of the LTSs corresponding to *P1, ..., Pm*. *interface_directives* must be parsed as a single argument on the command line and thus must be enclosed in quotes.

-interfaceuser

Indicate that some of the automata in *filename***.exp** have been obtained by semi-composition with "user-given" restriction interfaces, and compute the associated validation predicates. Note that this option does not make sense outside a compositional verification process using restriction interfaces. See **projector**(LOCAL) and **svl**(LOCAL) for more information about using restriction interfaces. This is not a default option.

-labels Display the number of labels followed by the list of labels potentially occurring in the state space of the input network of communicating automata and stop. If the **-interfaceuser** option is set, do not print the labels representing validation predicates (see **-interfaceuser** option).

-lotos Set LOTOS as the reference language. This is not a default option. See Section LANGUAGE PARAMETERS of **exp**(LOCAL) for details.

-mcrl Set mCRL as the reference language. This is not a default option. See Section LANGUAGE PARAMETERS of **exp**(LOCAL) for details.

-network *format*
Generate a network equivalent to *filename***.exp** in one of "nupn", "pep", "tina", "fc2", or "txt" *format* and stop:

If *format* is "nupn", **exp.open** generates a file named *filename***.nupn**, containing a Petri net in the NUPN (*Nested Unit Petri Net*) file format [Garavel-15-a] (see **caesar.bdd**(LOCAL) for a description of the NUPN format);

If *format* is "pep", **exp.open** generates a file named *filename***.ll_net**, containing a Petri net in the low-level PEP file format [Best-Grahlmann-98];

If *format* is "tina", **exp.open** generates a file named *filename***.tpn**, containing a Petri net in the ‘‘tpn’’ format of the TINA toolbox [Berthomieu-Ribet-Vernadat-04];

If *format* is "fc2", **exp.open** generates a file named *filename***.fc2**, containing a network of automata in the FC2 format [Bouali-Ressouche-Roy-deSimone-96].

If *format* is "txt", **exp.open** generates a file named *filename***.txt**, containing a description of the network of automata in an undocumented textual format. This description includes the list of files containing the communicating automata, the list of labels potentially occurring in the product and, for each label, the list of synchronization vectors.

The **bcg_io**(LOCAL) and **fc2link** tools are called internally to make the conversion from EXP to FC2. Note however that **fc2link** is not provided within CADP but belongs to the Fc2Tools distribution, which can be downloaded at http://www-sop.inria.fr/meije/verification.

Moreover, when converting EXP to FC2, the hidden event must be written "i" (see **-hidden** option above and Section LANGUAGE PARAMETERS of **exp**(LOCAL) for details) because this is required by **bcg_io**(LOCAL) and **fc2link**.

This option does not require an exploration module. This is not a default option.

This option is not available if *filename***.exp** contains a priority operator.

-nocheck

Parsing of EXP behaviours is generally followed by a static semantics verification phase to verify that behaviours are well-formed. Option **-nocheck** skips this verification phase. This option should be used with caution since the semantics of ill-formed behaviours is undefined. This is not a default option.

-prob Consider the LTSs composed in *filename***.exp** as "probabilistic LTSs" (see the **bcg_min**(LOCAL) manual page for details about probabilistic LTSs). Labels 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, are interpreted as "special" transitions, named "probabilistic". With this option, probabilistic transitions can always execute asynchronously. If a parallel composition attempts to synchronize probabilistic transitions explicitly, then **exp.open** issues a warning message.

-rate Consider the LTSs composed in *filename***.exp** as "stochastic LTSs" (see the **bcg_min**(LOCAL) manual page for details about stochastic LTSs). Labels of the form "rate %f" or "*label*; rate %f", where %f denotes a stricly positive floating-point number and *label* denotes a character string that does not contain the ";" character, are interpreted as "special" transitions, named "stochastic". With this option, stochastic transitions can always execute asynchronously. If a parallel composition attempts to synchronize stochastic transitions explicitly, then **exp.open** issues a warning message.

-ratebranching
This option is obsolete and should be replaced by the combination of options **-rate -branching**.

-silent Execute silently. Opposite of **-verbose**. Default option is **-verbose**.

-strong Do not perform partial order reduction of the graph. This is a default option.

-termination *string*
Set *string* as denoting the gate used to express behaviour termination. The default value depends on the reference language, see Section LANGUAGE PARAMETERS of **exp**(LOCAL) for details.

-unparse
Use the "**-bcg -unparse**" options of **bcg_io** while converting LTSs in AUT, FC2, or SEQ formats into BCG. See the **bcg_io**(LOCAL) manual page for details about these options.

-verbose
Report activities and progress, including errors, to the user's screen. Opposite of **-silent**. Default option is **-verbose**.

-version
Display the version number and stop.

-weaktrace
Perform on-the-fly partial order reduction modulo weak trace equivalence. This yields a generally smaller graph, which is equivalent modulo weak trace equivalence to the graph obtained using the **-strong** option. This is not a default option.

cc_options

if any, are passed to the C compiler.

prog_options
if any, are passed to *prog*.

EXIT STATUS

Exit status is 0 if everything is all right, > 0 otherwise.

BIBLIOGRAPHY

[Berthomieu-Ribet-Vernadat-04]
B. Berthomieu, P.-O. Ribet, and F. Vernadat. The tool TINA - Construction of Abstract State Spaces for Petri Nets and Time Petri Nets. In International Journal of Production Research, Vol. 42, No 14, July 2004.

[Best-Grahlmann-98]
Eike Best and Bernd Grahlmann. "PEP Documentation and User Guide." http://parsys.informatik.uni-oldenburg.de/~pep/paper.html. 1998."

[Bouali-Ressouche-Roy-deSimone-96]
Amar Bouali, Annie Ressouche, Valerie Roy, and Robert de Simone. The Fc2Tools set: a Toolset for the Verification of Concurrent Systems. In R. Alur and T.A. Henzinger, editors, Proceedings of the 8th Conference on Computer-Aided Verification (New Brunswick, New Jersey, USA). Lecture Notes in Computer Science volume 1102, Springer-Verlag, 1996.

[Brookes-Hoare-Roscoe-84]
S. D. Brookes, C. A. R. Hoare, and A. W. Roscoe. "A Theory of Communicating Sequential Processes." In Journal of the ACM, vol. 31, number 3, pages 560-599. ACM, 1984.

[Garavel-15-a]
Hubert Garavel. "Nested-Unit Petri Nets: A Structural Means to Increase Efficiency and Scalability of Verification on Elementary Nets." In R. Devillers and A. Valmari, editors, Proceedings of the 36th International Conference on Application and Theory of Petri Nets and Concurrency (Brussels, Belgium). Lecture Notes in Computer Science volume 9115, Springer-Verlag, 2015. Available from http://cadp.inria.fr/publications/Garavel-15-a.html

[Garavel-Sighireanu-99]
Hubert Garavel and Mihaela Sighireanu. "A Graphical Parallel Composition Operator for Process Algebras." In J. Wu, Q. Gao, and S.T. Chanson, editors, Proceedings of the Joint International Conference on Formal Description Techniques for Distributed Systems and Communication Protocols, and Protocol Specification, Testing, and Verification FORTE/PSTV'99 (Beijing, China). Kluwer Academic Publishers, 1999. Available from http://cadp.inria.fr/publications/Garavel-Sighireanu-99.html

[Groote-Ponse-90]
J.F. Groote and A. Ponse. "The syntax and semantics of mCRL." In A. Ponse, C. Verhoef and S.F.M. van Vlijmen, editors, Algebra of Communicating Processes '94, Workshops in Computing Series, Springer-Verlag, pp. 26-62, 1995. Also appeared as: Technical Report CS-R9076, CWI, Amsterdam, 1990.

[ISO-89]

ISO/IEC. "LOTOS --- A Formal Description Technique Based on the Temporal Ordering of Observational Behaviour." International Organization for Standardization --- Information Processing Systems --- Open Systems Interconnection. International Standard number 8807. Geneva, September 1989.

[ISO-01]
ISO/IEC. "Enhancements to LOTOS (E-LOTOS)." International Organization for Standardization --- Information Technology. International Standard number 15437:2001. Geneva, September 2001.

[Lang-05]
Frederic Lang. "EXP.OPEN 2.0: A Flexible Tool Integrating Partial Order, Compositional, and On-the-fly Verification Methods." In J. van de Pol, J. Romijn and G. Smith, editors, Proceedings of the 5th International Conference on Integrated Formal Methods IFM'2005 (Eindhoven, The Netherlands). Lecture Notes in Computer Science volume 3771, Springer-Verlag, 2005. Available from http://cadp.inria.fr/publications/Lang-05.html

[Lang-06]
Frederic Lang. "Refined Interfaces for Compositional Verification." In E. Najm, J.-F. Pradat-Peyre and V. Viguie Donzeau-Gouge, editors, Proceedings of the 26th IFIP WG 6.1 International Conference on Formal Techniques for Networked and Distributed Systems FORTE'2006 (Paris, France). Lecture Notes in Computer Science volume 4229, Springer-Verlag, 2006. Available from http://cadp.inria.fr/publications/Lang-06.html

[Milner-89]
Robin Milner. "Communication and Concurrency." Prentice-Hall, 1989.

[Pace-Lang-Mateescu-03]
Gordon Pace, Frederic Lang, and Radu Mateescu. "Calculating tau-confluence compositionally." In W.A. Hunt Jr. and F. Somenzi, editors, 15th Computer-Aided Verification conference (CAV 2003), Lecture Notes in Computer Science volume 2725, Springer-Verlag, 2003. Available from http://cadp.inria.fr/publications/Pace-Lang-Mateescu-03.html

AUTHORS

Versions 1.∗: Marius Bozga, Jean-Claude Fernandez, and Laurent Mounier.

Versions 2.∗: Frederic Lang and Hubert Garavel.

OPERANDS

filename.**exp**	network of communicating automata (input)
filename.**c**	graph module for filename.exp (output)
filename.**fc2**	FC2 network (output, with **-network fc2** option)
filename.**ll_net**	low level PEP Petri net (output, with **-network pep** option)
filename.**nupn**	NUPN Petri net (output, with **-network nupn** option)
filename.**tpn**	TINA Petri net (output, with **-network tina** option)
prog.**a**	exploration module (archive, input)
prog.**c**	exploration module (source, input)
prog.**o**	exploration module (object code, input)
prog	executable program (output)

FILES

$CADP/com/exp.open	"exp.open" shell script
$CADP/bin.'arch'/libexp_open.a	"exp.open" static library
$CADP/bin.'arch'/exp2c	"exp.open" graph module generator
$CADP/incl/caesar_*.h	OPEN/CAESAR interfaces
$CADP/bin.'arch'/libcaesar.a	OPEN/CAESAR library
$CADP/src/open_caesar/*.c	exploration modules (source)
$CADP/bin.'arch'/*.a	exploration modules (archive)
$CADP/bin.'arch'/*.o	exploration modules (object code)

SEE ALSO

aldebaran(LOCAL), **aut**(LOCAL), **bcg**(LOCAL), **bcg_io**(LOCAL), **caesar_hide_1**(LOCAL), **caesar_rename_1**(LOCAL), **exp**(LOCAL), **lotos.open**(LOCAL), **projector**(LOCAL), **regexp**(LOCAL), **seq**(LOCAL), **svl**(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