Distributed On-the-Fly Model Checking and Test Case Generation

Christophe Joubert and Radu Mateescu

INRIA Rhône-Alpes / VASY

http://www.inrialpes.fr/vasy

Context and motivation

- Explicit-state verification of concurrent systems
- Combine two approaches to fight state explosion
  - *On-the-fly* verification
    - Incremental state space construction
  - *Distributed* verification
    - State space exploration using several machines connected by a network

Two problems

- Model checking of alt-free μ-calculus
- Conformance test case generation

One solution

- Translation to a boolean equation system resolution
- Use of diagnostic generation

Outline

- Boolean equation systems
- Distributed local resolution algorithm
- Model checking of alternation-free mu-calculus
- Conformance test case generation
- Performance measures
- Conclusion and future work

Boolean equation systems

(alternation-free)

$$M_1 \quad \begin{cases} x_1 =_\mu x_2 \vee x_3 \\ x_2 =_\mu x_3 \vee x_4 \\ x_3 =_\mu x_2 \wedge x_7 \end{cases}$$

$$M_2 \quad \begin{cases} x_4 =_\mu x_5 \vee x_6 \\ x_5 =_\mu x_8 \vee x_9 \\ x_6 =_\mu \mathrm{F} \end{cases}$$

$$M_3 \quad \begin{cases} x_7 =_\nu x_8 \wedge x_9 \\ x_8 =_\nu \mathrm{T} \\ x_9 =_\nu \mathrm{F} \end{cases}$$

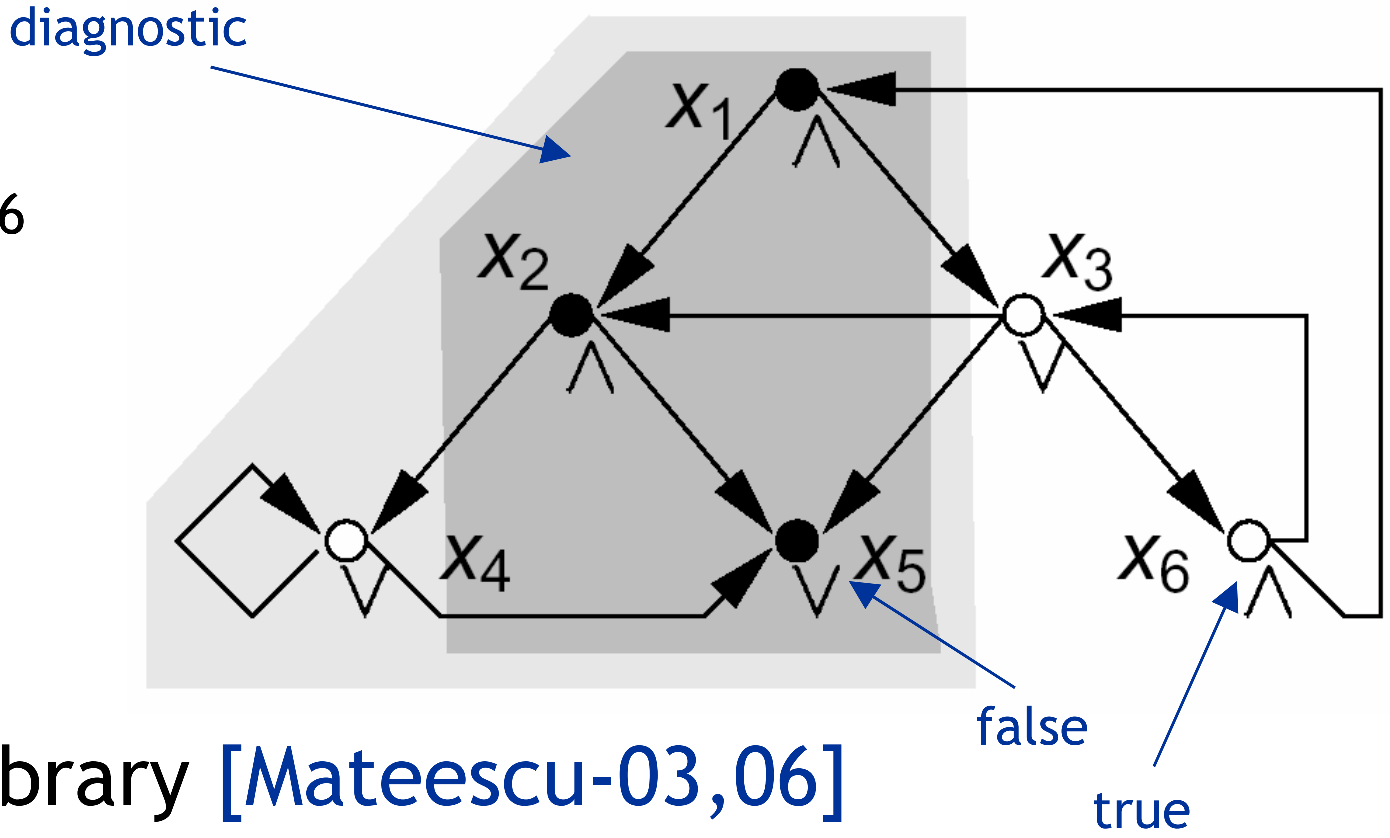
Sequential local resolution

BES

$$\begin{cases} x_1 =_\nu x_2 \wedge x_3 \\ x_2 =_\nu x_4 \wedge x_5 \\ x_3 =_\nu x_2 \vee x_5 \vee x_6 \\ x_4 =_\nu x_4 \vee x_5 \\ x_5 =_\nu \text{false} \\ x_6 =_\nu x_1 \wedge x_3 \end{cases}$$

boolean graph

[Andersen-94]



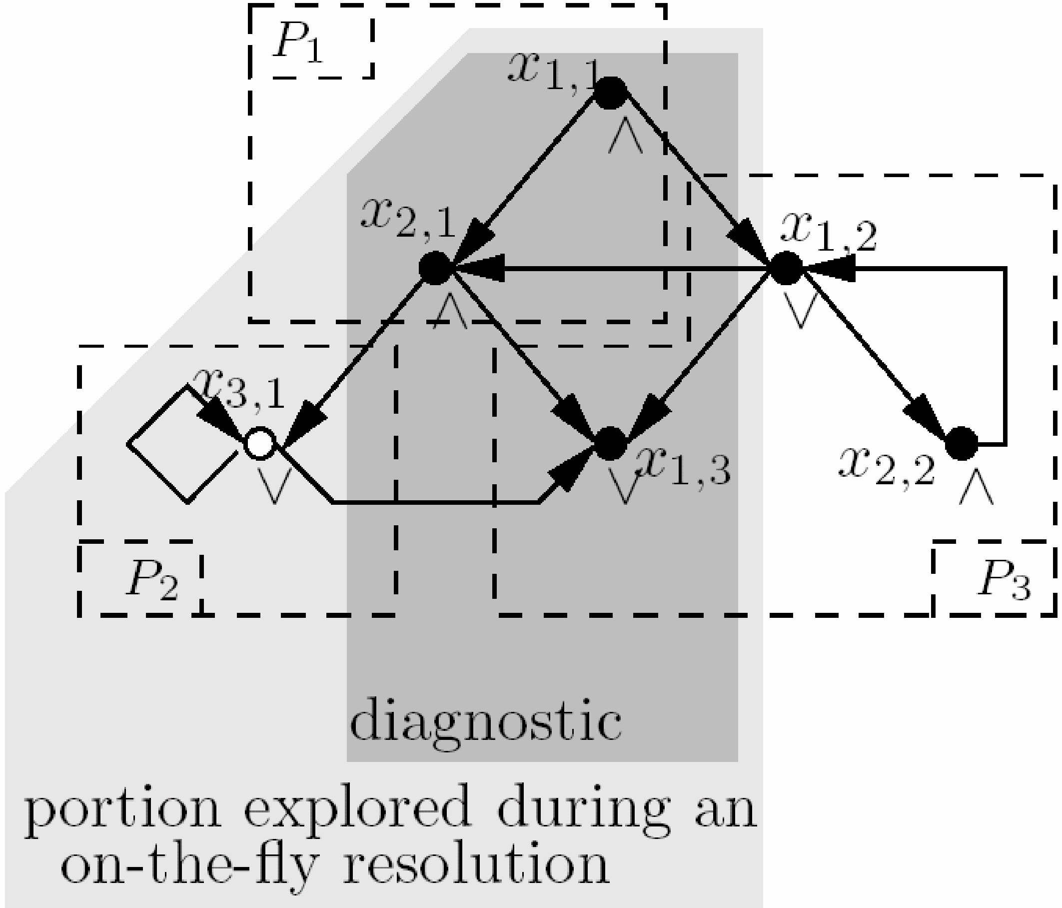
- Caesar_Solve library [Mateescu-03,06]
  - 5 resolution algorithms + diagnostic generation

Distributed local resolution

block 1 $\begin{cases} x_{1,1} \stackrel{\nu}{=} x_{2,1} \wedge x_{1,2} \\ x_{2,1} \stackrel{\nu}{=} x_{3,1} \wedge x_{1,3} \\ x_{3,1} \stackrel{\nu}{=} x_{3,1} \vee x_{1,3} \end{cases}$

block 2 $\begin{cases} x_{1,2} \stackrel{\mu}{=} x_{2,1} \vee x_{1,3} \vee x_{2,2} \\ x_{2,2} \stackrel{\mu}{=} x_{1,2} \end{cases}$

block 3 $\begin{cases} x_{1,3} \stackrel{\nu}{=} \text{false} \end{cases}$

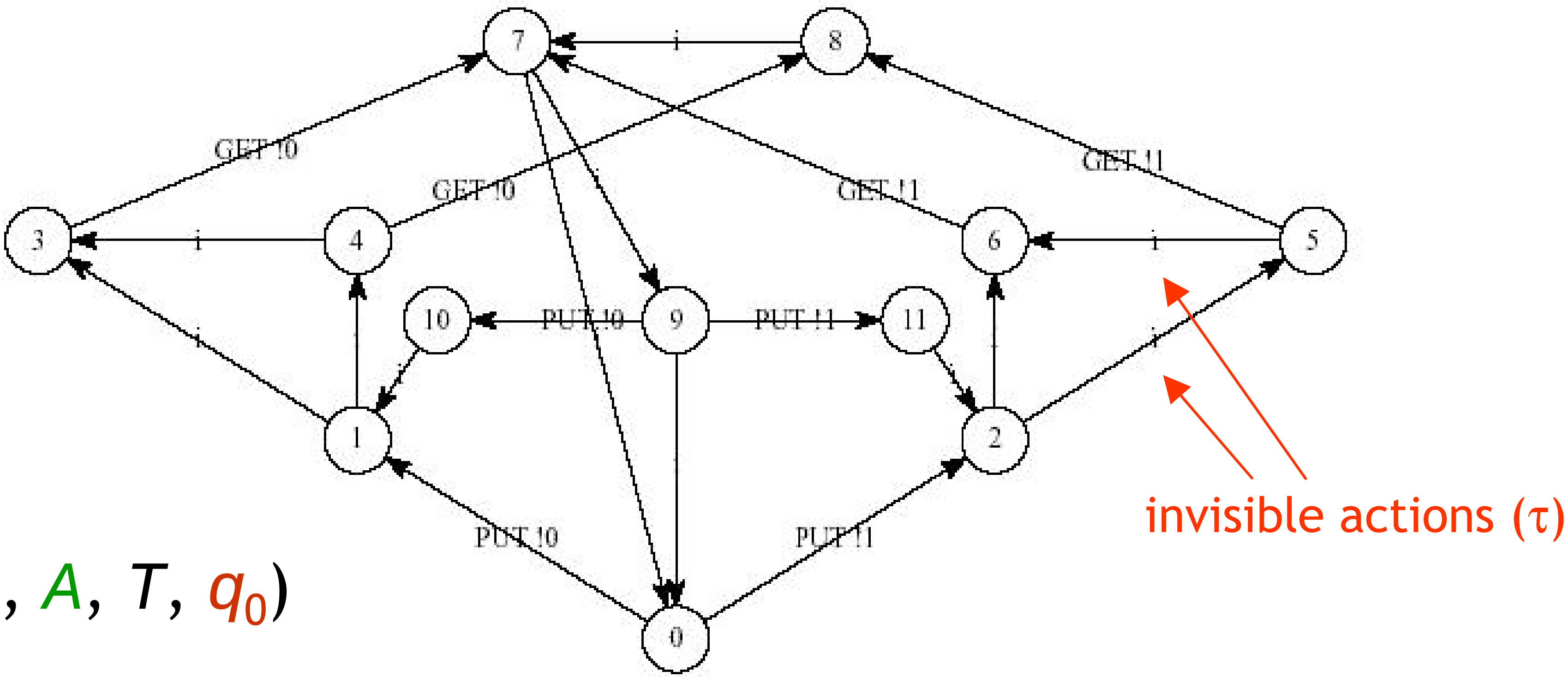


- MB-DSolve algorithm
  - Two distributed BFS traversals of the boolean graph (forward expansion and backward stabilization)
  - Partial distributed termination detection (stabilization of a portion of a block)

Related work
(distributed model checking)

- Linear temporal logic
  - Safety properties [Lerda-Sisto-99]
    - Distributed non-nested DFS
  - Full LTL [Barnat-Brim-Stribrna-01]
    - Distributed nested DFS
- Modal μ-calculus
  - Alternation depth 1 [Bollig-Leucker-Weber-02]
  - Alternation depth 2 [Leucker-Somla-Weber-03] [Holmen-Leucker-Lindstrom-04]
    - Distributed game graph exploration
    - UppDMC tool

Labelled Transition Systems

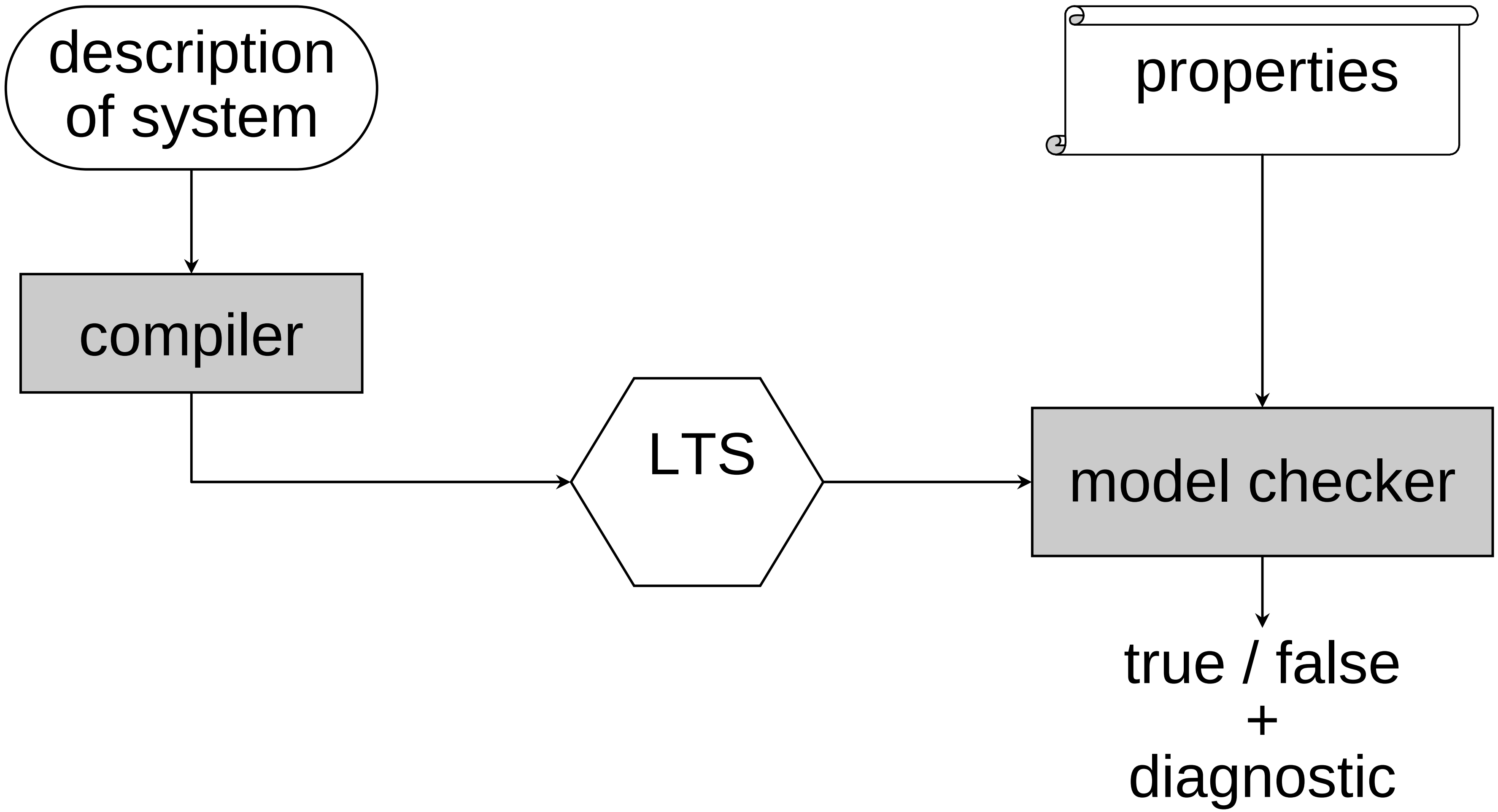


$M = (Q, A, T, q_0)$

CADP toolbox (http://www.inrialpes.fr/vasy/cadp)

- Explicit representation (succ/pred function)
  - BCG (Binary Coded Graphs)
- Implicit representation (successor function)
  - OPEN/CAESAR [Garavel-98]

Model checking



Modal mu-calculus

Let $M = (Q, A, T, q_0)$ be an LTS.

Action formulas

$$\alpha ::= a \mid \neg\alpha \mid \alpha_1 \vee \alpha_2 \mid \alpha_1 \wedge \alpha_2$$

State formulas

$$\varphi ::= \mathsf{F} \mid \mathsf{T} \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2$$
$$\mid \langle \alpha \rangle \varphi \mid [\alpha]\varphi$$
$$\mid X \mid \mu X . \varphi \mid \nu X . \varphi$$

Alternation-free fragment

- No mutual recursion between minimal and maximal fixed point variables [Emerson-Lei-86]
- Example:

  "every SEND is eventually followed by a RECV"

  $\nu X \,.\, [\,\mathrm{SEND}\,]\,(\mu Y \,.\, \langle\, \mathrm{T} \,\rangle\, \mathrm{T} \wedge [\,\neg \mathrm{RECV}\,]\, Y\,) \wedge [\,\mathrm{T}\,]\, X$
- Equational form HMLR [Larsen-88]:

  $\{\, X =_{\nu} [\,\mathrm{SEND}\,]\, Y \wedge [\,\mathrm{T}\,]\, X \,\}$

  $\{\, Y =_{\mu} \langle\, \mathrm{T} \,\rangle\, \mathrm{T} \wedge [\,\neg \mathrm{RECV}\,]\, Y \,\}$

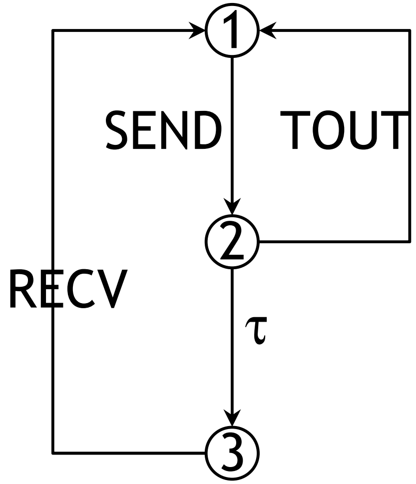
  (no cyclic dependencies between blocks)

Translation to BESs

- Principle: $s \models X$ iff X_s is true
- Formula: $\{ X =_\nu [\,\mathrm{SEND}\,]\, Y \wedge [\,\mathrm{T}\,]\, X \}$

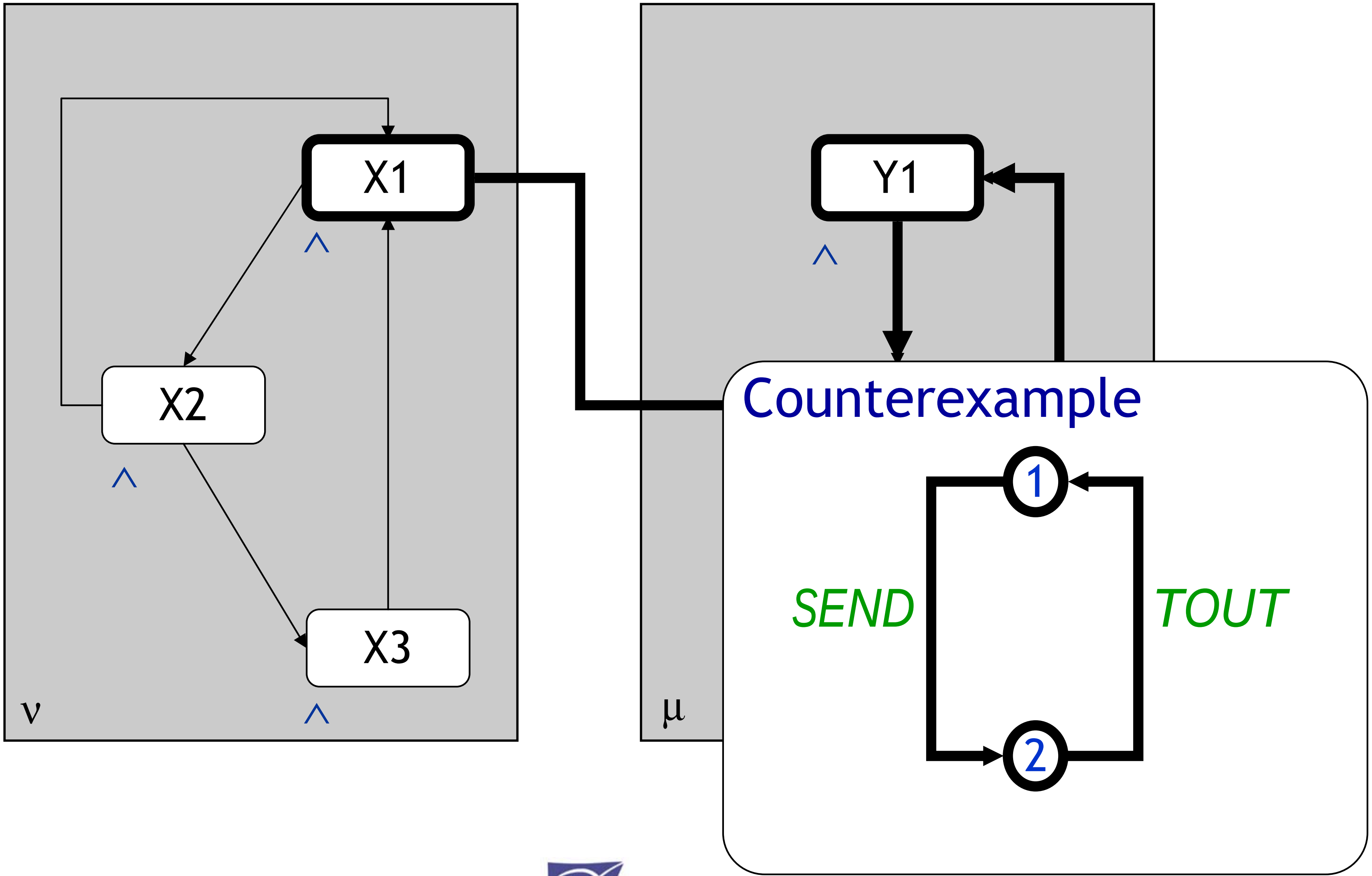
  $\{ Y =_\mu \langle \mathrm{T} \rangle\, \mathrm{T} \wedge [\,\neg\mathrm{RECV}\,]\, Y \}$
- BES: $\{ X_s =_\nu (\wedge_{s \xrightarrow{\mathrm{SEND}} s'} Y_{s'}) \wedge (\wedge_{s \rightarrow s'} X_{s'}) \}$

  $\{ Y_s =_\mu (\vee_{s \rightarrow s'} \mathrm{T}) \wedge (\wedge_{s \xrightarrow{\neg\mathrm{RECV}} s'} Y_{s'}) \}$



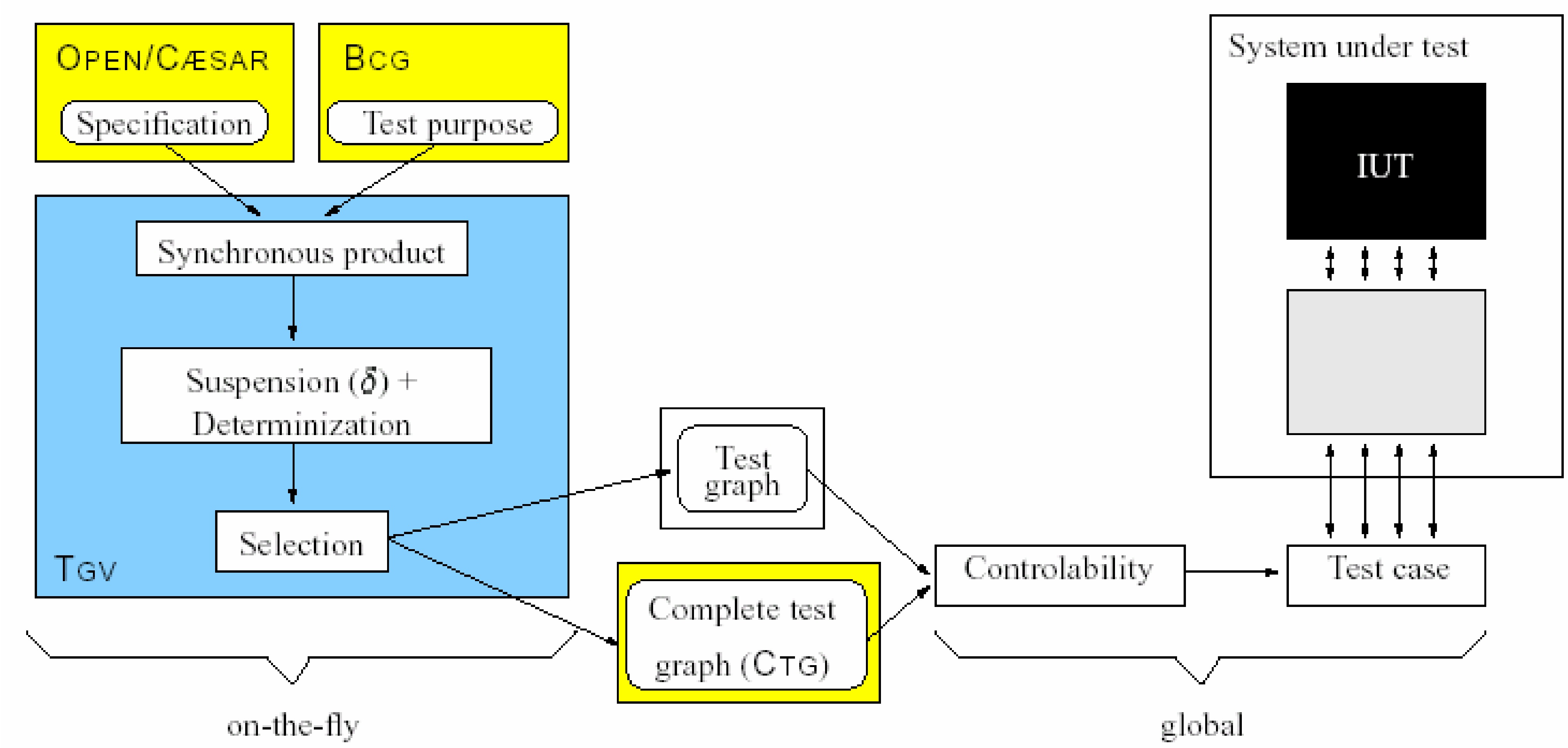
$$\begin{cases} X_1 =_\nu Y_2 \wedge X_2 \\ X_2 =_\nu X_1 \wedge X_3 \\ X_3 =_\nu X_1 \end{cases} \qquad \begin{cases} Y_1 =_\mu Y_2 \\ Y_2 =_\mu Y_1 \wedge Y_3 \\ Y_3 =_\mu \mathrm{T} \end{cases}$$

Local resolution with diagnostic



Conformance test generation using TGV

(Test Generation based on Verification technology)



[Fernandez-Jard-Jeron-Viho-96] [Jard-Jeron-05]

Translation into BES resolution with diagnostic

- *L2A* (*lead to accept*): all states of the synchronous product *Spec* × *TP* from which an accepting state can be reached

$$\phi_{l2a} = \phi_{acc} \wedge \nu X \,.\, [\,-\,]\,(\phi_{acc} \Rightarrow X\,)$$

$$\phi_{acc} = \mu Y \,.\, acc \vee \langle\,-\,\rangle\, Y$$

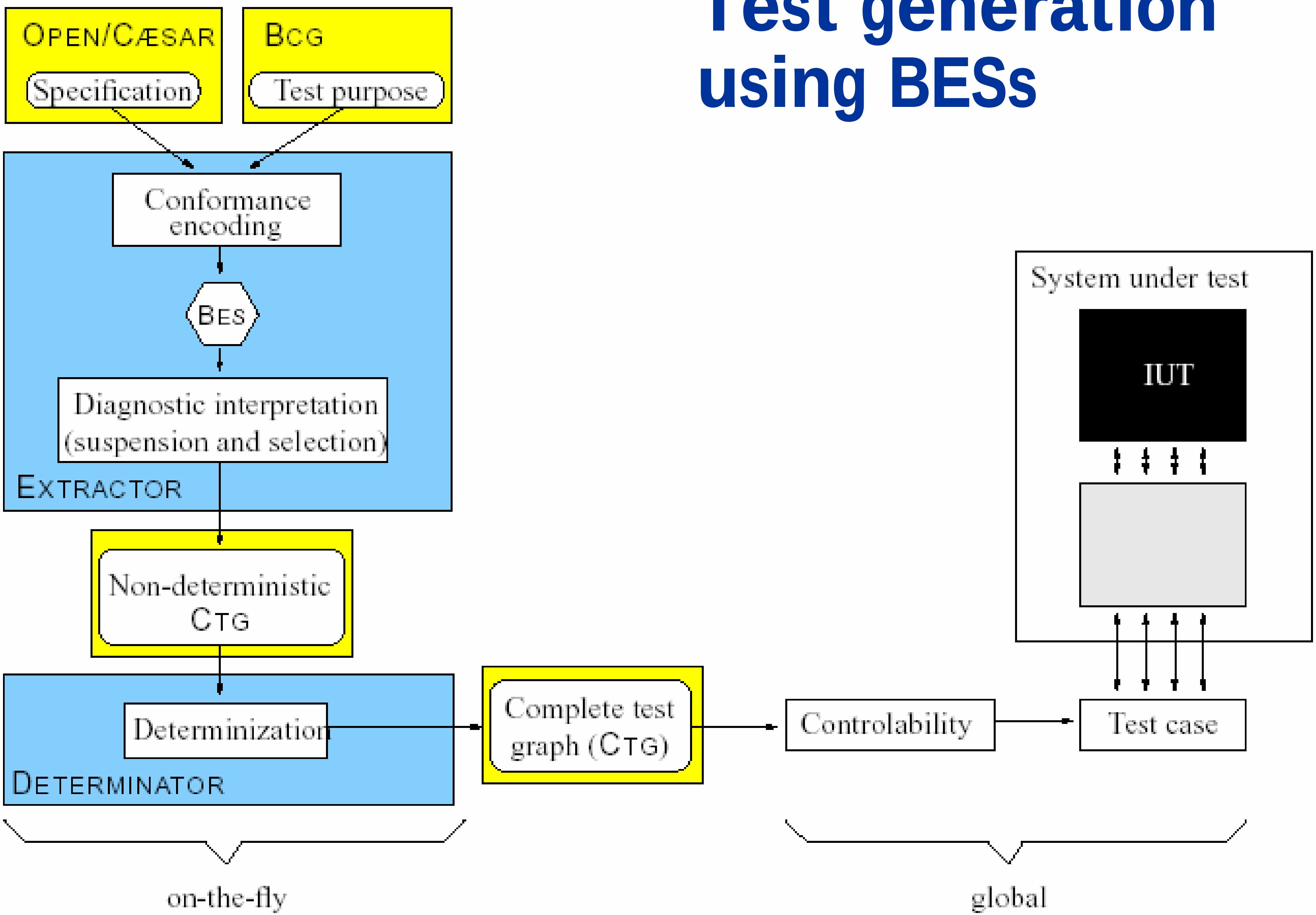
- Translation to a BES:

$$s \;|\!= \phi_{l2a} \;=\; Y_s \wedge X_s$$

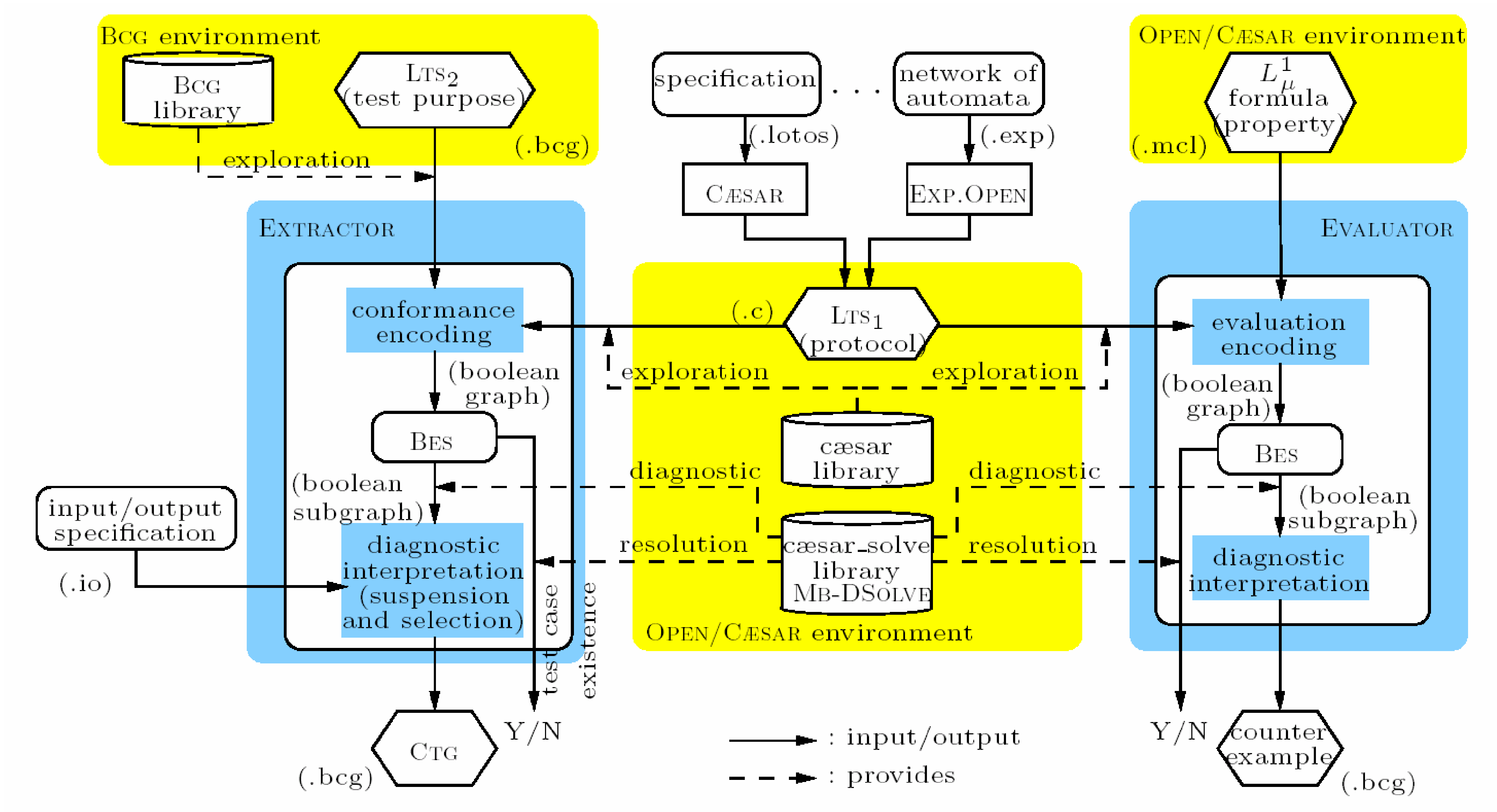
$$\{\, X_s =_\nu \bigwedge_{s \to s'} (Z_{s'} \vee X_{s'}\,)\,\} \qquad \{\, Y_s =_\mu acc_s \vee \bigvee_{s \to s'} Y_{s'} \,\}$$

$$\{\, Z_s =_\nu \neg acc_s \wedge \bigwedge_{s \to s'} Z_{s'} \,\}$$

Test generation using BESs



Tools architecture

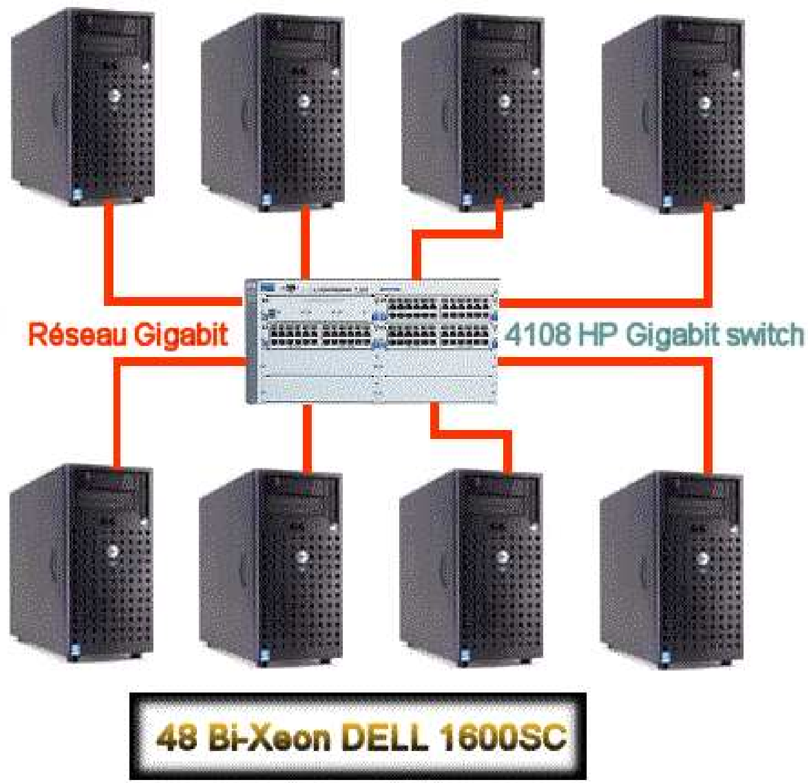


Experiments

- IDPOT cluster

  48 bi-Xeon

  2.4 GHz, 1.5 Gb

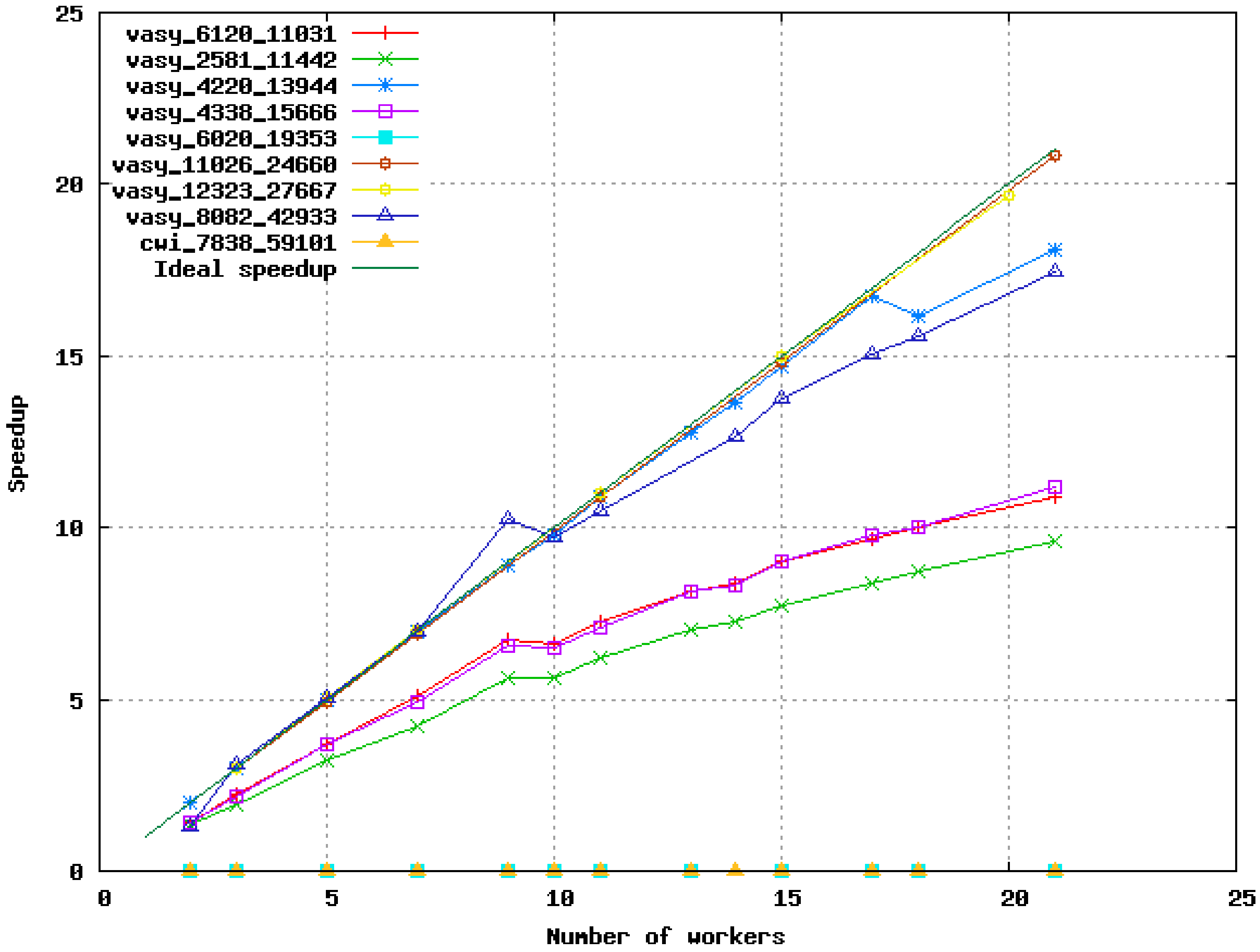


- VLTS benchmark suite

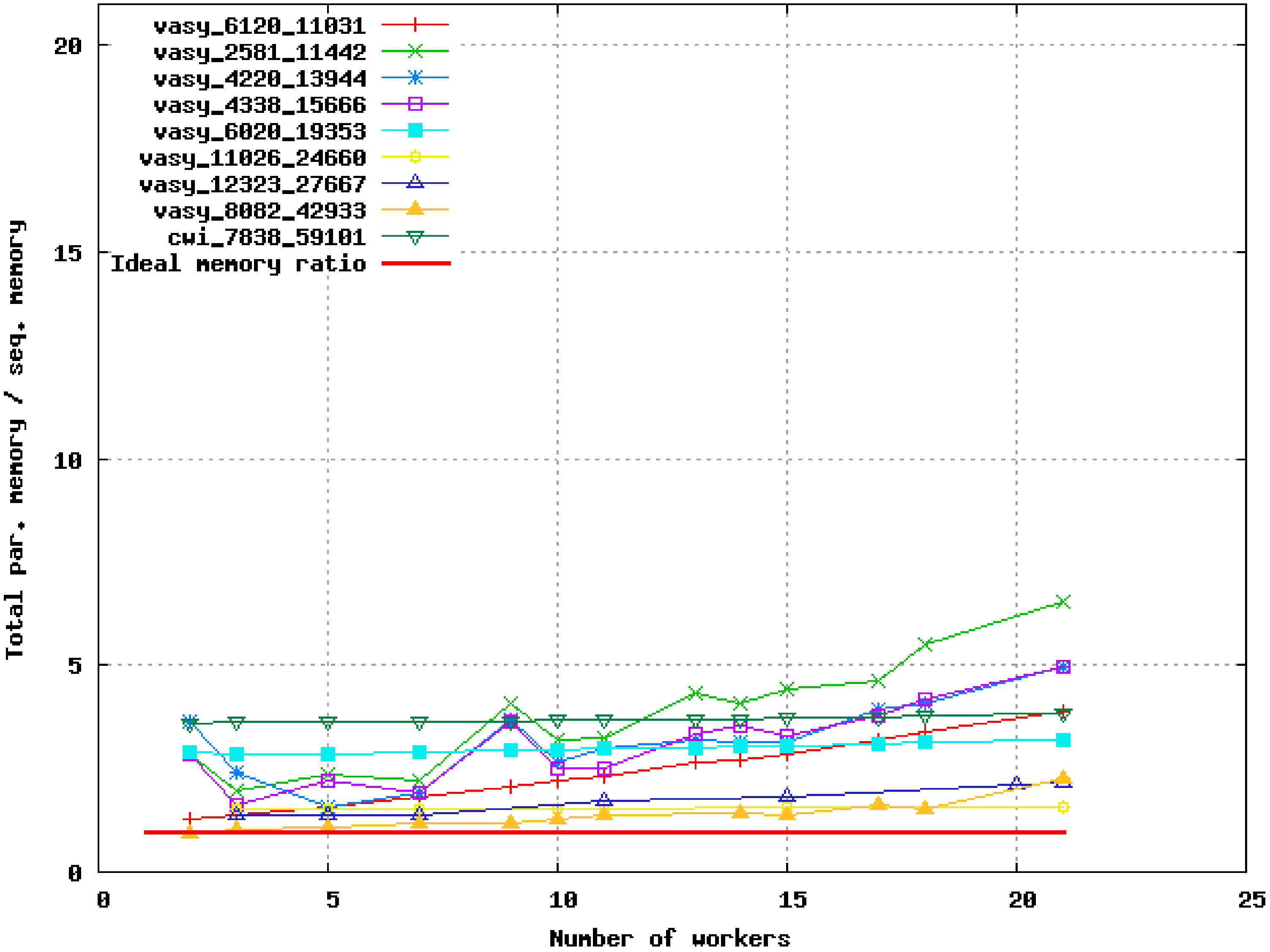
  http://www.inrialpes.fr/vasy/cadp/resources/benchmark_bcg.html

Distributed vs. sequential Evaluator

(speedup, absence of deadlock, VLTS)



Distributed vs. sequential Evaluator
(memory overhead, absence of deadlock, VLTS)



Distributed Evaluator vs. UppDMC
(absence of deadlock, VLTS)

EXAMPLE	absence of deadlock				
	truth	U (s)	U (MB)	E (s)	E (MB)
vasy_2581_11442	false	44	461	2	272
vasy_4220_13944	false	56	726	21	294
vasy_4338_15666	false	64	745	2	313
vasy_6020_19353	true	59	1 085	24	1 239
vasy_6120_11031	false	95	947	1	170
cwi_7838_59101	true	149	1 531	46	2 298
vasy_8082_42933	false	162	1 374	2	268

Evaluator: 21 Xeon / 2.4 GHz / 1.5 Gb
UppDMC: 25 bi-Pentium III / 500 MHz / 512 Mb

Distributed Evaluator vs. UppDMC

(presence of livelock, VLTS)

EXAMPLE	presence of livelock				
	truth	U (s)	U (MB)	E (s)	E (MB)
vasy_2581_11442	false	47	n.c.	7	844
vasy_4220_13944	false	67	n.c.	622	1 149
vasy_4338_15666	false	64	n.c.	11	1 203
vasy_6020_19353	true	125	n.c.	8	1 442
vasy_6120_11031	false	108	n.c.	13	1 092
cwi_7838_59101	true	314	n.c.	16	2 793
vasy_8082_42933	false	134	n.c.	24	2 401

Evaluator: 21 Xeon / 2.4 GHz / 1.5 Gb
UppDMC: 25 bi-Pentium III / 500 MHz / 512 Mb

Sequential Extractor vs. TGV

(generic TP – accepting state after 10 visible actions, VLTS)

Example	TGV				(sequential) Extractor					
	time	MB	states	trans.	time	%	MB	%	states	trans.
vasy_164_1619	15'8s	242	100 319	231 266	3'47s	75	210	13	438 861	2 982 696
vasy_166_651	20'23s	242	170 657	586 602	1'41s	92	113	53	444 542	1 504 985
cwi_371_641	6'5s	1600	125 894	597 445	5'20s	12	310	81	1 912 260	3 163 177
vasy_386_1171	9s	11	3 319	3 892	7s	22	10	9	5 561	6 324
vasy_1112_5290	23s	33	10 827	20 888	13s	44	28	15	15 008	41 225
b256	597'4s	2322	264 194	854 786	139'22s	77	2772	-2	12 139 232	39 020 231

TGV:
- 1.82 times slower than Extractor + Determinator
- Produces CTGs between 30% and 50% smaller

"raw" CTGs
(contain τ-transitions)

Distributed Extractor + Determinator

(generic TP, 7 nodes, VLTS)

Example	(distributed) Extractor		Determinator			
	time	Mb	time	Mb	states (final)	transitions (final)
vasy_164_1619	4'39s	470	4'40s	55	103 658	975 594
vasy_166_651	2'59s	335	2'27s	50	173 259	801 675
cwi_371_641	12'4s	880	25'8s	185	127 218	777 278
vasy_386_1171	16s	104	15s	6	2 452	3 894
vasy_1112_5290	27s	228	17s	7	8 369	41 225
b256	180'	6127	19'	459	527 875	1 709 058

final CTGs
(without τ-transitions)
strongly equivalent to
those produced by TGV

Conclusion and future work

- Summary
  - MB-DSolve: distributed local resolution of multi-block BESs
  - Generic implementation using OPEN/CAESAR
  - Two applications distributed & on-the-fly:
    - Model checking of alt-free mu-calculus (Evaluator 3.5)
    - Conformance test case generation (Extractor)
  - Good speedups w.r.t. sequential versions
  - Performance comparable with state-of-the-art tools (UppDMC, TGV)
- Ongoing and future work
  - Further experiments and benchmarks
  - Handling of heterogeneous architectures (grids)
  - Other applications (discrete controller synthesis)