On-the-Fly Verification using CADP

Radu Mateescu

INRIA Rhône-Alpes / VASY
655, avenue de l'Europe
F-38330 Montbonnot Saint Martin, France
http://www.inrialpes.fr/vasy

INRIA Rhône-Alpes
http://www.inrialpes.fr

- Created in December 1992
  - 19 research projects
  - Experimental technological platforms (PC clusters, high-speed networks, robotics, virtual reality studio)
- Knowledge dissemination
  - Over 130 doctoral candidates
  - University courses (Inst. Nat. Polytechnique Grenoble, Univ. Joseph Fourier, Ecole Normale Sup. de Lyon)
- Technology transfer
  - Cooperations with Bull and W3C
  - 6 start-up companies

The VASY team (Validation of Systems)
http://www.inrialpes.fr/vasy

- Leader: Hubert Garavel
- 2 INRIA researchers: Radu Mateescu, Frédéric Lang
- 1 Bull engineer: Solofo Ramangalahy
- 1 post-doc, 1 PhD student, 3 expert engineers
- Scientific areas of interest:
  - Formal methods and specification languages
  - Model-based verification technologies
  - Industrial case-studies and applications
- Software tools:
  - The CADP verification toolbox
  - The TRAIAN compiler (E-LOTOS)

The CADP toolbox
http://www.inrialpes.fr/vasy/cadp

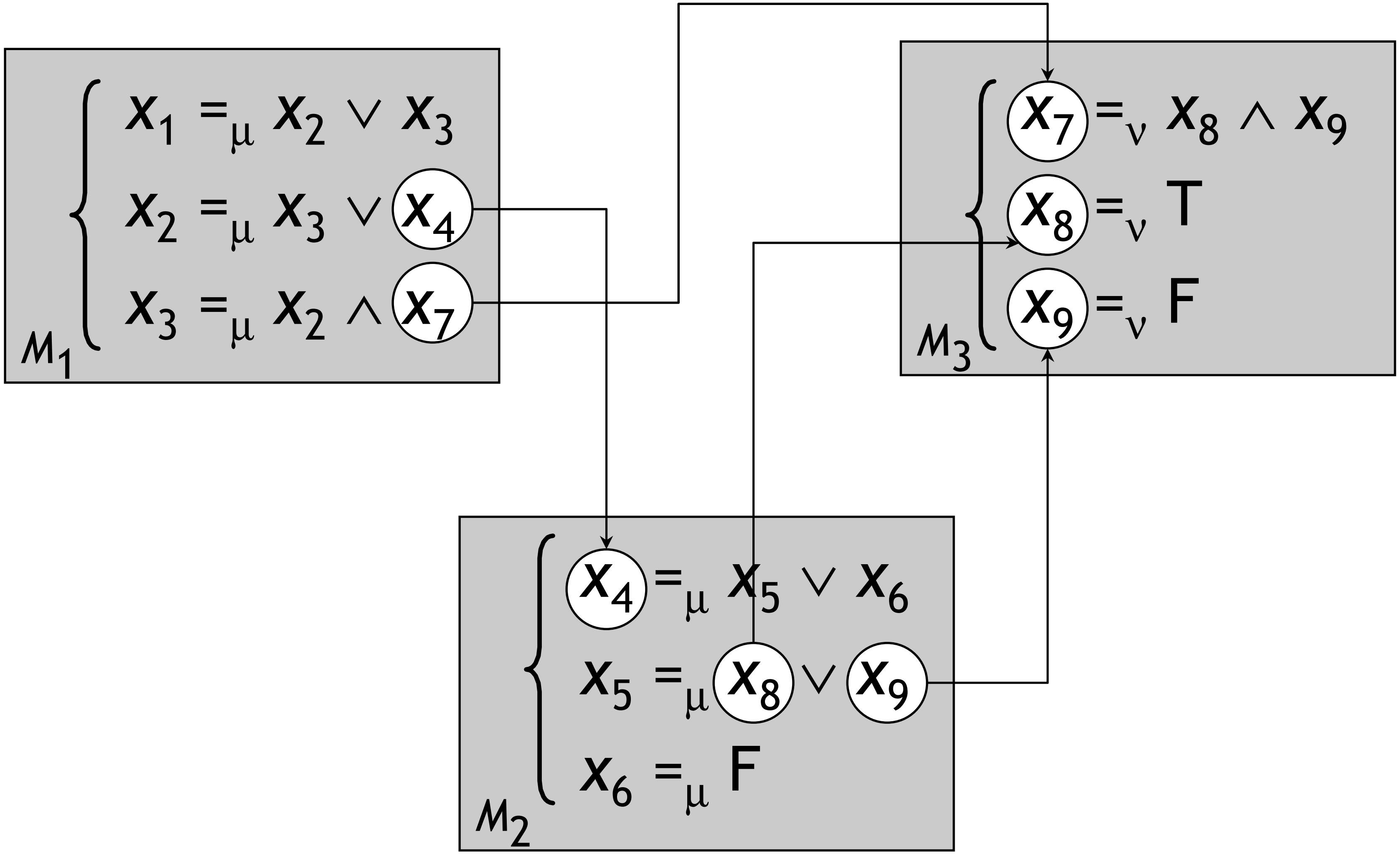
- Input languages
  - ISO formal description techniques (LOTOS, E-LOTOS)
  - Networks of communicating automata
- Functionalities
  - Compilation, rapid prototyping, interactive simulation
  - Equivalence checking, model checking
  - Compositional verification, test generation
- Applications: 65 case studies, 13 research tools
- OPEN/CAESAR [Garavel-98]
  - CADP generic environment for state space manipulation
  - Implicit state space representation (*successor function*)

Motivation

- On-the-fly verification
  - Builds the state space incrementally
  - Allows to detect errors in large systems
- Practical needs
  - Easy construction of on-the-fly verification tools
  - Generic software components for verification
- Boolean Equation Systems (BES)
  - Technology for equivalence checking and model checking
  - On-the-fly resolution and diagnostic generation

➔ *Goal: provide generic software (libraries)*

Alternation-free BES

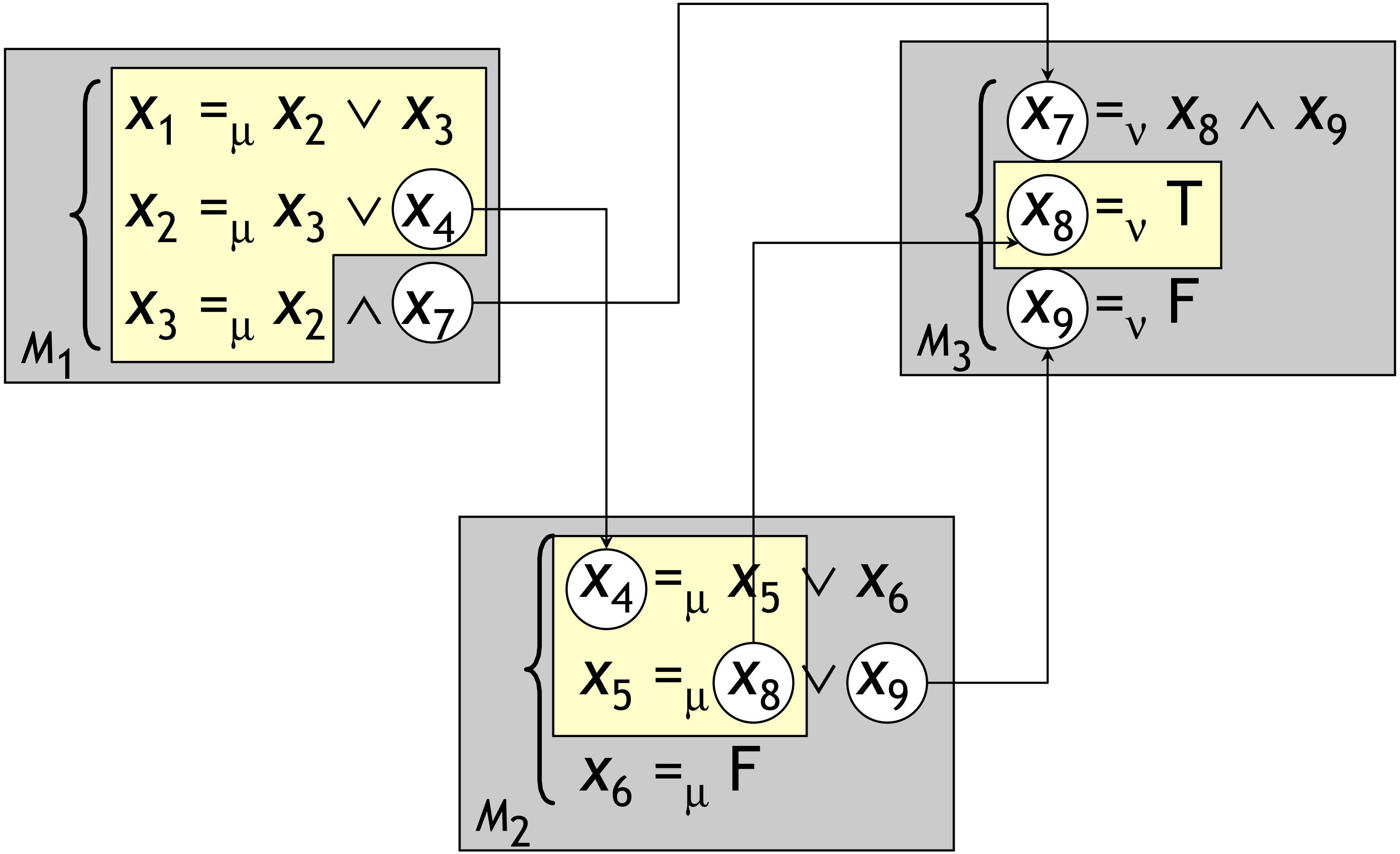


$$M_1 \left\{ \begin{aligned} x_1 &=_\mu x_2 \vee x_3 \\ x_2 &=_\mu x_3 \vee x_4 \\ x_3 &=_\mu x_2 \wedge x_7 \end{aligned} \right.$$

$$M_3 \left\{ \begin{aligned} x_7 &=_\nu x_8 \wedge x_9 \\ x_8 &=_\nu T \\ x_9 &=_\nu F \end{aligned} \right.$$

$$M_2 \left\{ \begin{aligned} x_4 &=_\mu x_5 \vee x_6 \\ x_5 &=_\mu x_8 \vee x_9 \\ x_6 &=_\mu F \end{aligned} \right.$$

On-the-fly resolution



$$M_1 \left\{ \begin{array}{l} x_1 =_\mu x_2 \vee x_3 \\ x_2 =_\mu x_3 \vee x_4 \\ x_3 =_\mu x_2 \wedge x_7 \end{array} \right.$$

$$M_3 \left\{ \begin{array}{l} x_7 =_\nu x_8 \wedge x_9 \\ x_8 =_\nu T \\ x_9 =_\nu F \end{array} \right.$$

$$M_2 \left\{ \begin{array}{l} x_4 =_\mu x_5 \vee x_6 \\ x_5 =_\mu x_8 \vee x_9 \\ x_6 =_\mu F \end{array} \right.$$

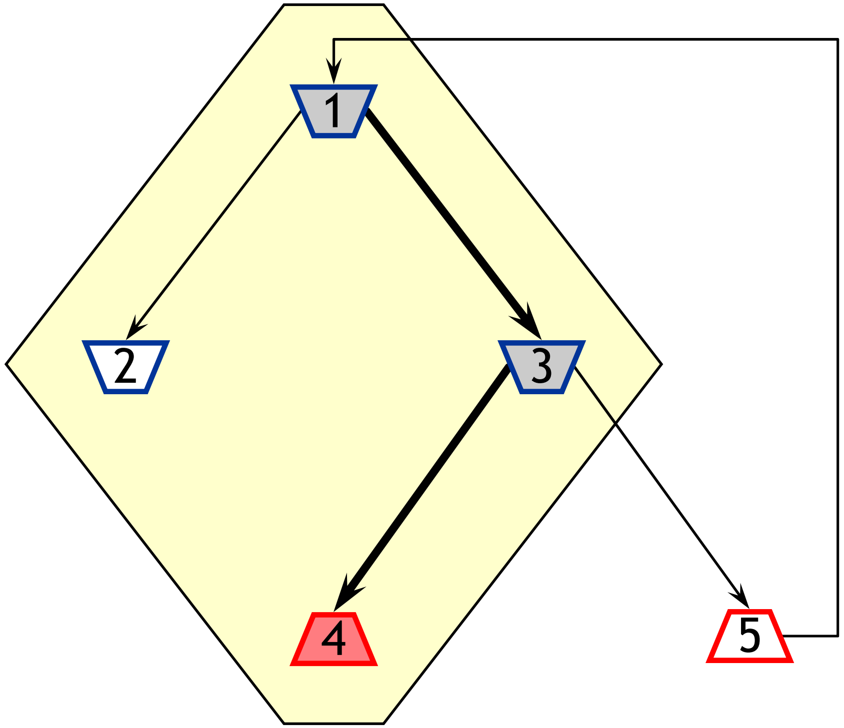
Boolean graphs

[Andersen-94]

BES (μ-block)

$$\begin{cases} x_1 =_\mu x_2 \vee x_3 \\ x_2 =_\mu F \\ x_3 =_\mu x_4 \vee x_5 \\ x_4 =_\mu T \\ x_5 =_\mu x_1 \end{cases}$$

boolean graph



▽ : ∨-variables

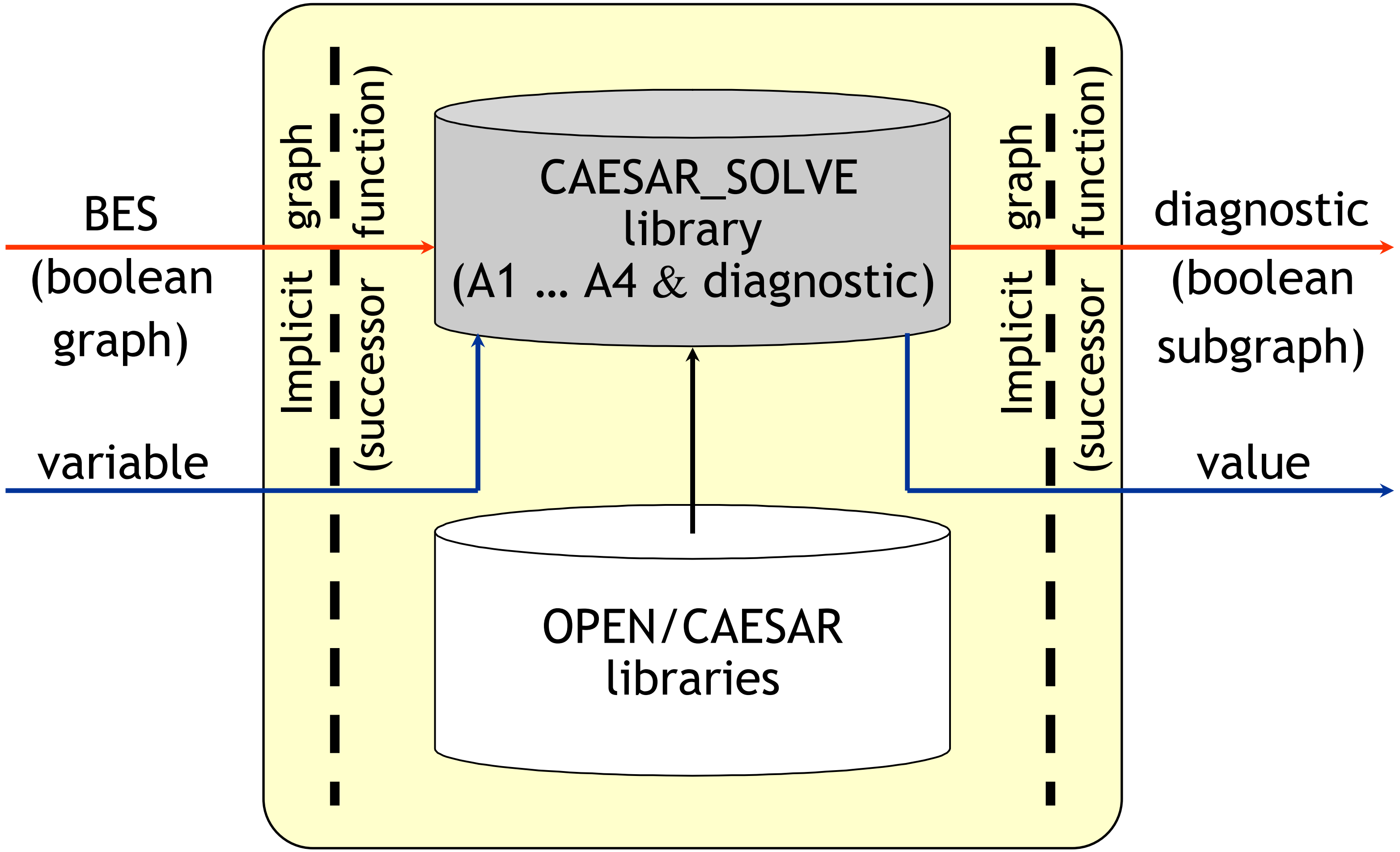
△ : ∧-variables

Resolution algorithms
[TACAS 2003]

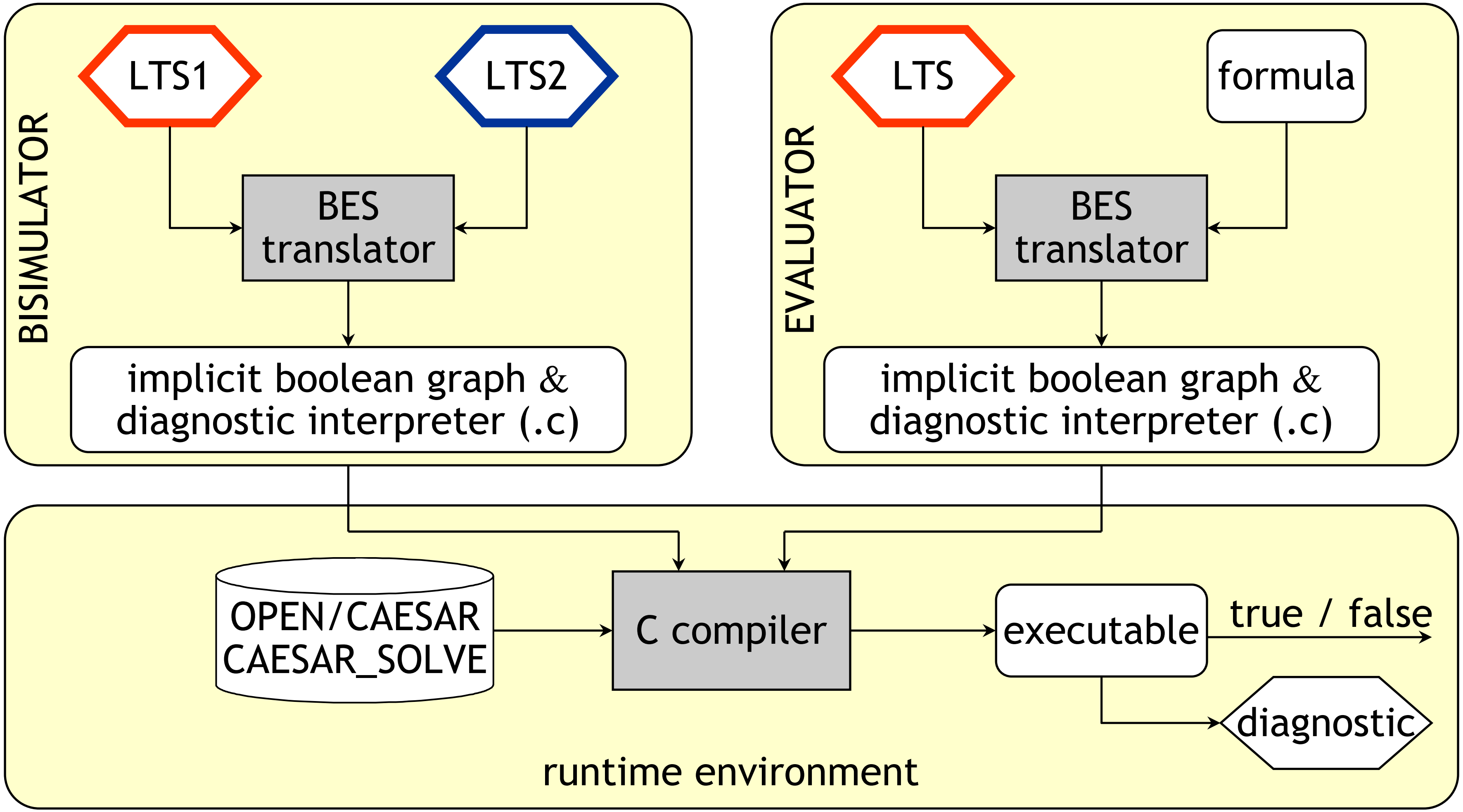
- A1 (DFS, general)
  - Memory complexity $O(|V|+|E|)$
- A2 (BFS, general)
  - Small-depth diagnostics
  - Memory complexity $O(|V|+|E|)$
- A3 (DFS, acyclic)
  - Memory complexity $O(|V|)$
- A4 (DFS, disjunctive / conjunctive)
  - Memory complexity $O(|V|)$

Time complexity $O(|V|+|E|)$

CAESAR_SOLVE library



BISIMULATOR and EVALUATOR



Algorithm usage guidelines

- A1 and A2 (diagnostic depth ↓)
  - All equivalences and their preorders
  - Alternation-free μ-calculus formulas
- A3 (memory ↓)
  - Strong equivalence: one LTS acyclic
  - Safety and $\tau^*.a$: one LTS acyclic (τ-circuits allowed)
  - Branching and observational: both LTS acyclic
  - Acyclic LTS and μ-calculus formula (via reduction)
- A4 (memory ↓)
  - All equivalences: one LTS deterministic
  - CTL, ACTL, and PDL formulas

Ongoing and future work

- New algorithms within CAESAR_SOLVE
  - Single-scan & low-memory algorithms for trace-based verification (low-depth acyclic boolean graphs)
  - Further resolution strategies (combined DFS-BFS, random exploration, …)
- New applications of CAESAR_SOLVE
  - Detection of τ-confluent transitions [CAV 2003]
  - Test generation
  - Discrete controller synthesis
  - Horn clause resolution

  } using diagnostic generation
- Distributed resolution algorithms

  ➔ Distributed equivalence checking and model checking