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

• 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

\[
\begin{align*}
M_1 & : \\
X_1 &= \mu X_2 \lor X_3 \\
X_2 &= \mu X_3 \lor X_4 \\
X_3 &= \mu X_2 \land X_7
\end{align*}
\]

\[
M_2 : \\
\begin{align*}
X_4 &= \mu X_5 \lor X_6 \\
X_5 &= \mu X_8 \lor X_9 \\
X_6 &= \mu F
\end{align*}
\]

\[
M_3 : \\
\begin{align*}
x_7 &= \nu X_8 \land X_9 \\
x_8 &= \nu T \\
x_9 &= \nu F
\end{align*}
\]
On-the-fly resolution

\[\begin{align*}
x_1 &= \mu x_2 \lor x_3 \\
x_2 &= \mu x_3 \lor x_4 \\
x_3 &= \mu x_2 \land x_7 \\
x_4 &= \mu x_5 \lor x_6 \\
x_5 &= \mu x_8 \lor x_9 \\
x_6 &= \mu F \\
x_7 &= \nu x_8 \land x_9 \\
x_8 &= \nu T \\
x_9 &= \nu F
\end{align*}\]
Boolean graphs
[Andersen-94]

BES ($\mu$-block)

\[
\begin{align*}
  x_1 &= \mu x_2 \lor x_3 \\
  x_2 &= \mu F \\
  x_3 &= \mu x_4 \lor x_5 \\
  x_4 &= \mu T \\
  x_5 &= \mu x_1
\end{align*}
\]

$: \lor$-variables

$: \land$-variables

Diagram:

- Blue diamond: $\lor$-variables
- Red triangle: $\land$-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 $O(|V|)$
CAESAR_SOLVE library

Implicit graph (successor function)

BES (boolean graph)

variable

diagnostic

(implicit subgraph)

value

OPEN/CAESAR libraries

CAESAR_SOLVE library
(A1 ... A4 & diagnostic)
BISIMULATOR and EVALUATOR

BISIMULATOR

LTS1
BES translator
implicit boolean graph & diagnostic interpreter (.c)

OPEN/CAESAR
CAESAR_SOLVE

C compiler
runtime environment

EVALUATOR

LTS
BES translator
implicit boolean graph & diagnostic interpreter (.c)

formula

C compiler
executable
diagnostic
true / false
Algorithm usage guidelines

• A1 and A2 (diagnostic depth ↓)
  - All equivalences and their preorders
  - Alternation-free $\mu$-calculus formulas

• A3 (memory ↓)
  - Strong equivalence: one LTS acyclic
  - Safety and $\tau^*.a$: one LTS acyclic ($\tau$-circuits allowed)
  - Branching and observational: both LTS acyclic
  - Acyclic LTS and $\mu$-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 $\tau$-confluent transitions [CAV 2003]
  - Test generation
  - Discrete controller synthesis
  - Horn clause resolution

- **Distributed resolution algorithms**
  - Distributed equivalence checking and model checking