Compositional verification applied to RERS 2019

Frédéric Lang
Inria & LIG – CONVECS team (Grenoble, France)

Franco Mazzanti
ISTI-CNR – FM&&T group (Pisa, Italy)
Approach

• Categories: parallel CTL and parallel LTL
• Main tool: CADP (cadp.inria.fr)
• Auxiliary tools:
  – SPOT (spot.lrde.epita.fr) Translation of LTL to Büchi automata
  – KandISTI/FMC (fmt.isit.cnr.it/kandisti) Cross-checking of CTL results
  – nuXmv (nuxmv.fbk.eu) Cross-checking of LTL results
• Main technique: Compositional verification
The CADP toolbox

http://cadp.inria.fr

• Developed by Inria/CONVECS for > 30 years
• Model & equivalence checking, rapid prototyping, test case generation, ... (> 80 tools and libraries)
• Enumerative techniques: LTS model
• Main languages and tools used in this work:
  – LNT system description language,
  – MCL property description language,
  – EVALUATOR model checker ,
  – BCG_MIN LTS minimization tool,
  – SVL scripting language and compiler, ...
RERS parallel verification tasks

- System description $P_1 \parallel \ldots \parallel P_n$
  - 9 system descriptions from 8 to 70 parallel processes and from 29 to 234 actions
  - We used the DOT representation
  - Automated translation from DOT to LNT

- Property $\varphi$
  - 20 CTL properties for each system description
  - 20 LTL properties for each system description
CTL compositional verification

• Results of [MW14] are used to infer from \( \varphi \)
  – a set of actions \( H \) that can be hidden
  – an equiv. relation \( R \) that preserves \( \varphi \) (improved)
• A reduced model \( M \) is obtained using SVL as smart \( R \) reduction of hide \( H \) in \( P_1 \parallel ... \parallel P_n \)
• \( \varphi \) is verified on \( M \) using EVALUATOR:
  \[
P_1 \parallel ... \parallel P_n \models \varphi \quad \text{iff} \quad M \models \varphi
  \]

CTL results

• **All 180 CTL properties verified** on this laptop:
  – 158 min. CPU (≈ 2.5 hours) / ≈ 5 hours elapsed
  – 200 MB memory
  – Largest intermediate LTS ≤ 3363 states

• Cross-checking with **KandISTI/FMC**:
  – on the fly, explicit verification on unreduced LTS
  – 126 problems solved out of 180 (70 %)
    max 2h, 64 GB memory available
  – **CADP** results confirmed
LTL compositional verification

• Reduced model $M$ obtained using same approach
• Use of Büchi automaton $B$
  – Automated translation of $\neg \varphi$ to transition-based Büchi automaton using SPOT (HOA format)
  – Automated encoding from HOA to LNT
  – Accepting transitions encoded by action ACC
• EVALUATOR is used to verify the acceptance condition encoded as an MCL formula:
  
  $P_1 || \ldots || P_n \models \varphi \iff M || B \models \neg<true^* . ACC> @$
LTL results

• All 180 LTL properties verified on this laptop:
  – 144 min. CPU (≈ 2.5 hours) / ≈ 5 hours elapsed
  – 200 MB memory
  – Largest intermediate LTS ≤ 1068 states

• Cross-checking with nuXmv:
  – LTL verification on the reduced LTS (risk)
  – all problems solved
  – CADP results confirmed
Conclusion

• **Compositional verification is effective** to solve **CTL** and **LTL** parallel benchmarks of RERS 2019

• **Causes of success:**
  – Expressive languages (LNT, MCL, SVL, ...)
  – Efficient tools
  – Team working: combination of expertises, synergy
  – Hard work and tenacity

• **Diversity of approaches** ⇒ trust increases

• **New results**: papers in preparation