# Trio2Promela: a Model Checker for Temporal Metric Specifications

Domenico Bianculli [1], Angelo Morzenti [2], Matteo Pradella [3], Pierluigi San Pietro [2] and Paola Spoletini [2]

[1]University of Lugano - Faculty of Informatics, Lugano, Switzerland — [2]Politecnico di Milano - Dipartimento di Elettronica e Informazione, Milano, Italy — [3] CNR IEIIT-MI, Milano, Italy

## TRIO

- It is a first-order linear-time temporal logic with past and future operators and a quantitative metric on time.
- Used for specification of large, real-life critical real-time systems
- The language is very expressive but satisfiability is undecidable
- Verification and Validation activities
  - Testing and simulation
  - Semi-automatic theorem proving techniques
  - Decidable approximations of the specification

## Our Goal

- *Verification* of TRIO specifications using the SPIN model checker
- Dealing with a decidable subset of TRIO:
  - Natural numbers as time domain
  - Quantifiers only range on finite domains
  - Equivalent to LTL with past, but more compact and easier to use

## The Model Checking Problem: $\mathcal{M} \models \psi$

- Traditionally:
  - $\mathcal{M}$ is an operational model
  - $\psi$ is an LTL formula
- Our approach:
  - $\mathcal{M}$ is a TRIO specification (one or more axioms)
  - $\psi$ is a TRIO formula (one property)
  - We check for the satisfiability of $\neg(\mathcal{M} \rightarrow \psi)$

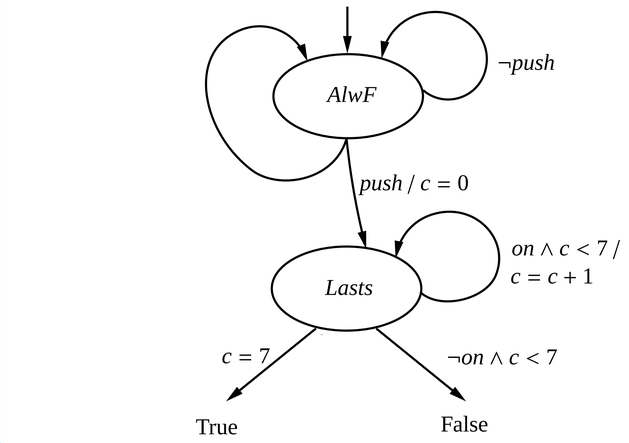
## Trio2Promela

- It takes TRIO formulae with shallow alternation of past and future operators
- It efficiently handles translation into SPIN's language, Promela
  - Alternation modeled with concurrent processes
  - Metric operators translated with counters to reduce code size
  - Optimizations are the real content of the toolkit
- For a formula $\phi$, the generated Promela code is *linear* in $|\phi|$

## TRIO formula

$$AlwF(push \rightarrow Lasts(on, 7))$$

## Alternating automaton




## Promela code

```
if
  ::ex_Lasts_1->
    if
      ::cont_lasts_1<7 && !(on==1)  -> flag = 1;
      ::cont_lasts_1<7 && (on==1)  -> cont_lasts_1++;
      ::cont_lasts_1==7 -> ex_Lasts_1=0;
    fi;
  ::!ex_Lasts_1-> skip;
fi;
if
  ::flag==1 -> s=0; goto gen;
  ::flag!=1 ->
    if
      ::!(push!=1)->  ex_Lasts_1=1; cont_lasts_1 = 0;
      ::(push!=1)->skip;
    fi;
fi;
```

## Experimental Results

Table 1: Satisfiability checking of a temporal logic model and a safety formula, both translated with Trio2Promela

| | Memory (MB) | States | Approx. time (s) |
|---|---|---|---|
| KRC1-safety | 8.2 | 105151 | 2 |
| KRC2-safety | 31.8 | 361627 | 8 |
| KRC3-safety | 171.0 | 1371870 | 46 |
| FP - 2 procs | 3.0 | 13179 | 1 |
| FP - 3 procs | 13.3 | 304047 | 2 |
| FP - 4 procs | 241.1 | 6321520 | 59 |
| FP - 5 procs | Exhausted | not completed | not completed |

Table 2: Model Checking of a Promela model against a safety formula translated with Trio2Promela, compared with the model translated with LTL2BA

| | Trio2Promela | | | LTL2BA | | |
|---|---|---|---|---|---|---|
| | (MB) | States | Approx. time (s) | (MB) | States | Approx. time (s) |
| KRC1-safety | 2.6 | 298 | 1 | 2.6 | 909 | 1 |
| KRC2-safety | 2.6 | 674 | 1 | 2.6 | 2233 | 1 |
| KRC3-safety | 2.6 | 1390 | 1 | 2.7 | 3658 | 1 |
| FP - 2 procs | 2.6 | 345 | 1 | 2.6 | 326 | 1 |
| FP - 3 procs | 2.7 | 3355 | 1 | 2.6 | 3240 | 1 |
| FP - 4 procs | 3.5 | 27977 | 1 | 3.3 | 41694 | 1 |
| FP - 5 procs | 9.7 | 215886 | 1 | 9.8 | 222940 | 1 |

Table 3: Comparison of translation times and sizes

| | Translation time (s) | | Size | |
|---|---|---|---|---|
| | LTL2BA | T2P | LTL2BA | T2P |
| Safety-KRC | <1 | <1 | 2 | 9 |
| Mutex-Fischer | <1 | <1 | 2 | 8 |
| $AlwF(A \rightarrow WithinF(B, 10))$ | <1 | <1 | 11 | 218 |
| $AlwF(A \rightarrow WithinF(B, 15))$ | 222 | <1 | 16 | 308 |
| $AlwF(A \rightarrow WithinF(B, 17))$ | <3127 | <1 | 18 | 308 |
| $AlwF(A \rightarrow WithinF(B, 20))$ | infeasible | <1 | 21 | 398 |
| $AlwF(A \wedge SomF(B))$ | <1 | <1 | 1 | 20 |
| $AlwF(A \wedge SomF(B) \wedge Lasts(C, 5))$ | <1 | <1 | 5 | 26 |
| PAX Formula | infeasible | 7 | - | 252 |

## References

[1] Angelo Morzenti, Matteo Pradella, Pierluigi San Pietro, and Paola Spoletini. Model checking TRIO specifications in Spin. In *FME 2003*, volume 2805 of *LNCS*, pages 542–561, 2003.

[2] Angelo Morzenti and Pierluigi San Pietro. Object-oriented logical specification of time-critical systems. *ACM Trans. Softw. Eng. Methodol.*, 3(1):56–98, 1994.

[3] Matteo Pradella, Pierluigi San Pietro, Paola Spoletini, and Angelo Morzenti. Practical model checking of LTL with past. In *ATVA03: 1st Workshop on Automated Technology for Verification and Analysis*, 2003.

[4] Orna Kupferman and Moshe Vardi. Weak alternating automata are not that weak. In *ISTCS'97*, pages 147–158. IEEE Computer Society Press, 1997.

[5] Gerard J. Holzmann. The model checker SPIN. *IEEE Trans. Softw. Eng.*, 23(5):279–295, 1997.

Work partially supported by the IST EU project "PLASTIC" and the MIUR project: "FIRB: Applicazione della Teoria degli Automi all'Analisi, alla Compilazione e alla Verifica di Sistemi Critici e in Tempo Reale"