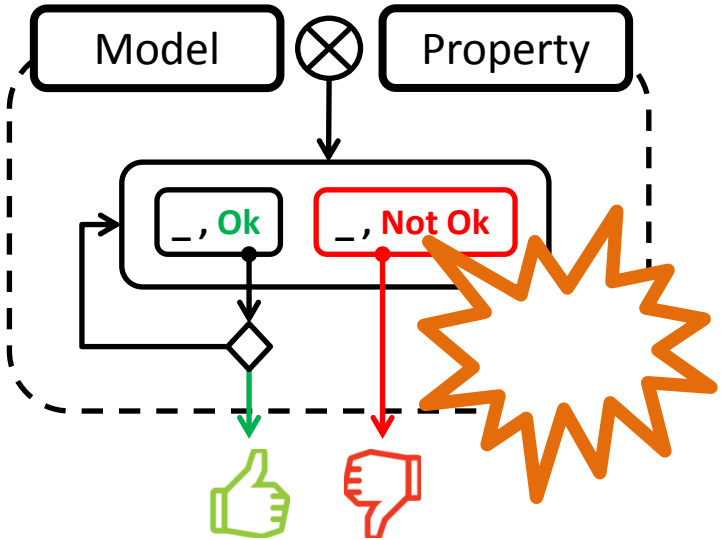
Partially Bounded Context-Aware Verification

LE ROUX Luka & TEODOROV Ciprian

Lab-STICC, ENSTA Bretagne, Brest, France

Introduction

Model-Checking



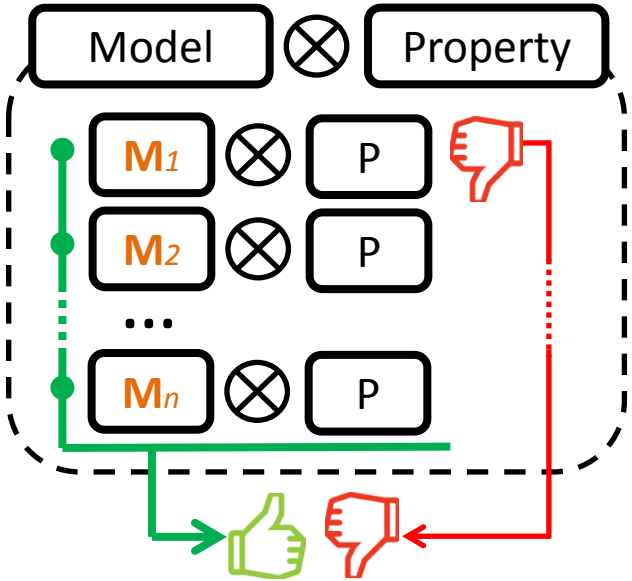
Exhaustive and automatic formal method
[ClarkeEmerson82, QueillleSifakis82]

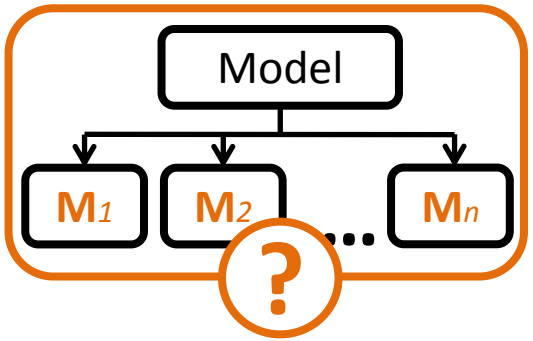
- Major algorithmic breakthroughs
  [ClarkeEmersonSifakis09]
  - *Symbolic approach (OBDDs)*
  - *Partial order reduction*
  - *Bounded Model Checking*
  - *Abstraction Refinement Loop* (CEGAR)

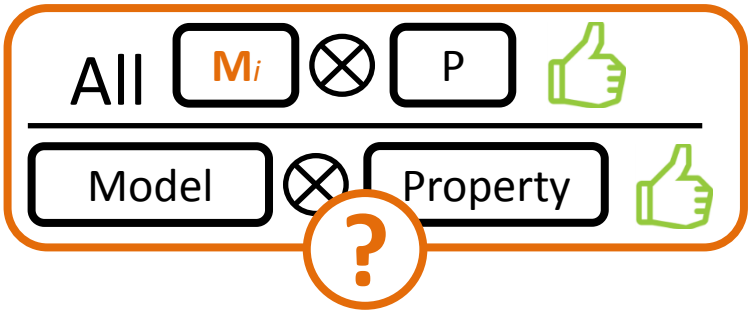
- When scalability issues persist
  - Refine the specifications
  - Narrow the modeling scope
  - **Split the analysis**

Introduction

Splitting the analysis

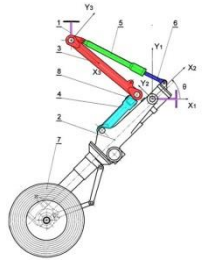


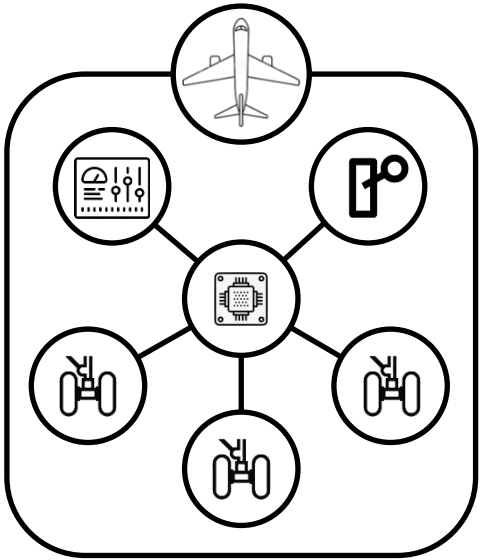




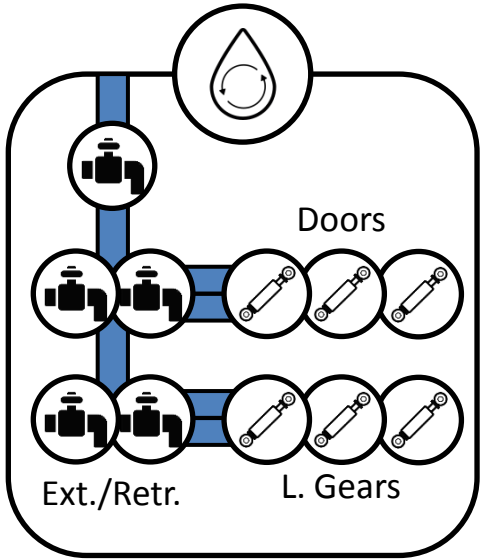
Case Study

Landing Gear System [F. Boniol, V. Wiels, ABZ'2014]

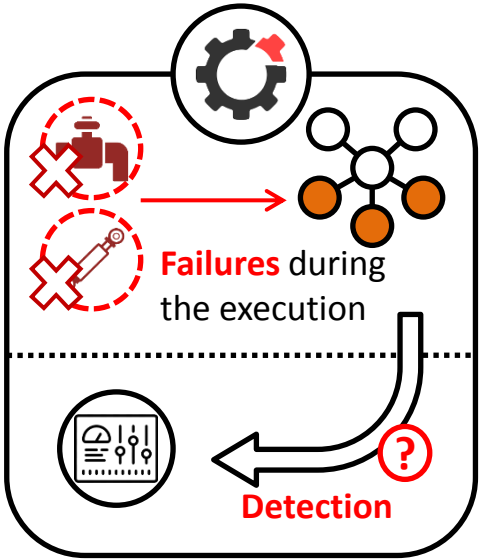




Overview

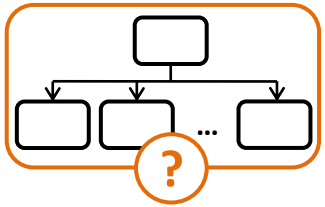


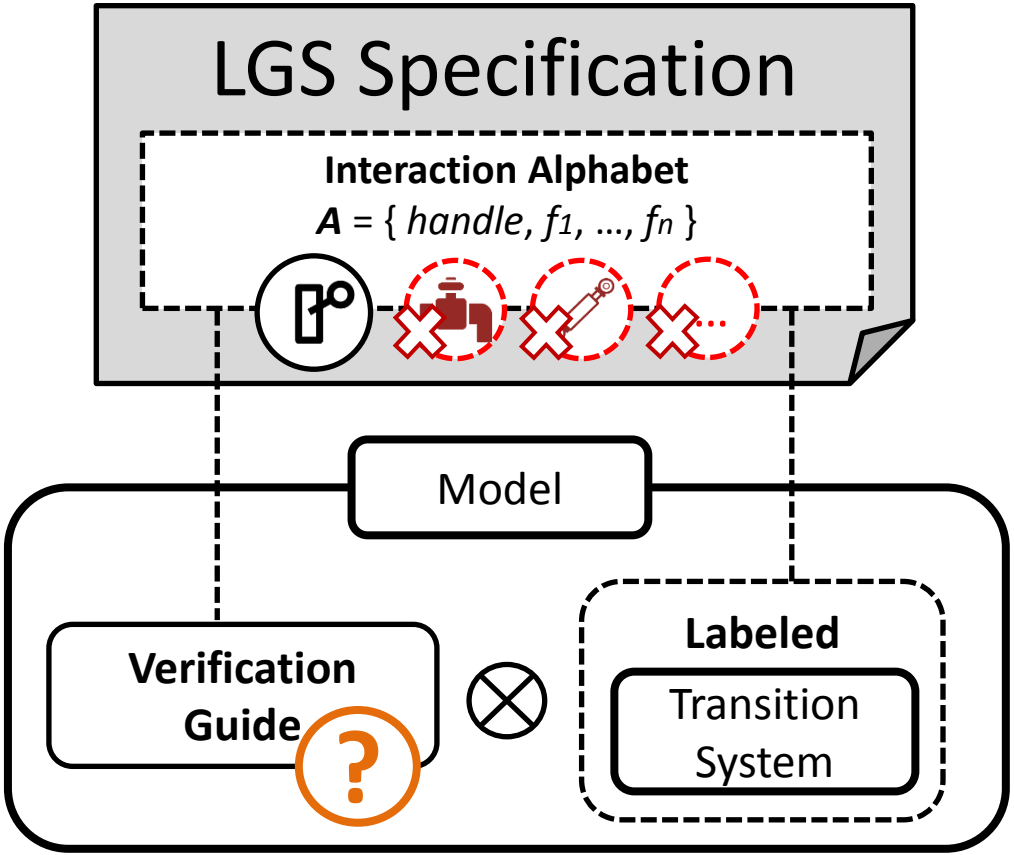
Hydraulic Parts
(Extension / Retraction)

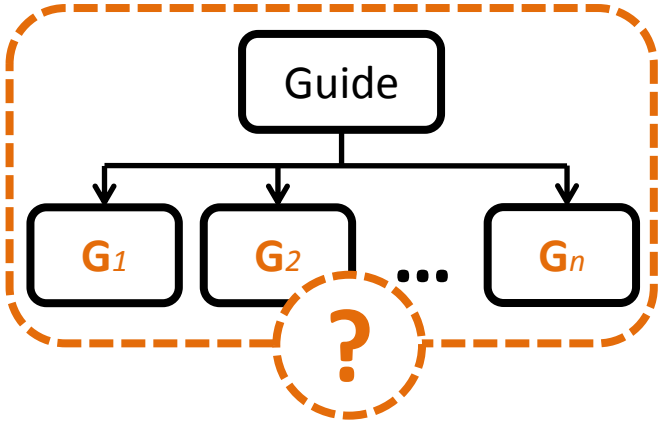


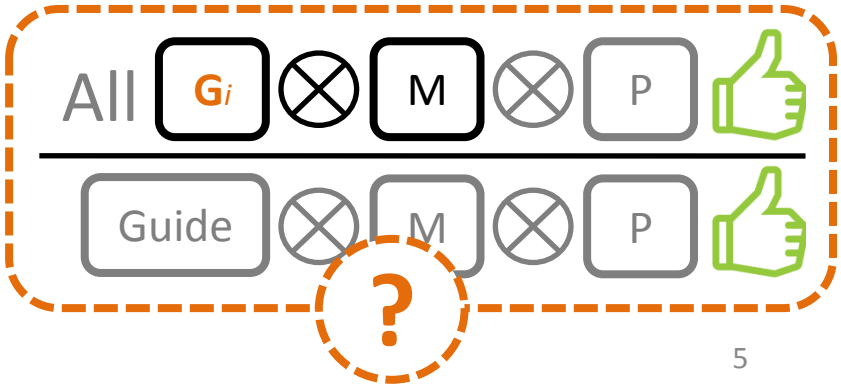
Failures Injection
& Requirements

Context-Aware Verification [STTT'17]



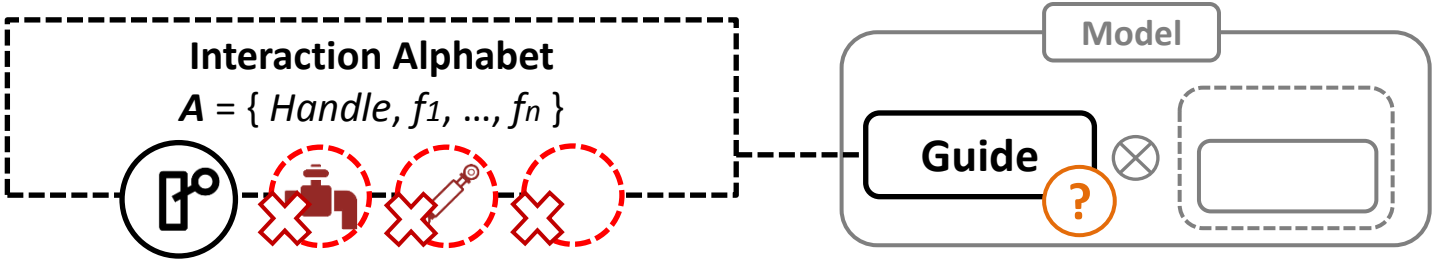






xGDL

Operators



a	Interaction
⊥	Empty term
C_1 ; C_2	Sequence
C_1 □ C_2	Alternative
C ?	Optional

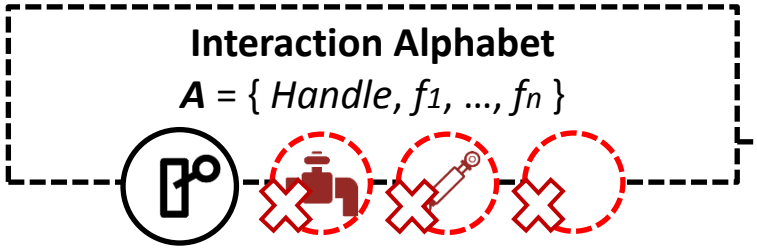
C *	Repetition (*0+*)
C +	Repetition (*1+*)
C {*i*, *j*}	Repetition (*bounded*)
C_1 || C_2	Parallel interleaving
{*i*, *j*} of [C_1, C_2, …, C_n]	Permutations

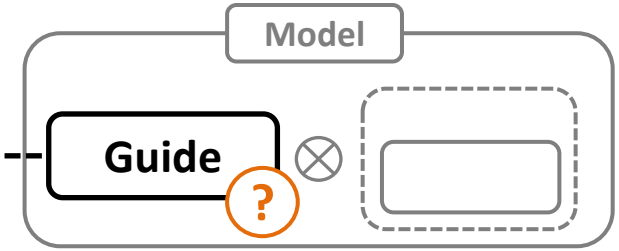
Examples

Pilot	handle *	*« Handle the landing gears at will »*
Failures	{0, 3} of [f_1, f_2, …, f_{12}]	*« 0 to 3 unique failures among a set of 12 »*
Guide	Pilot || Failures	*« 0 to 3 unique failures, arbitrarily injected »*

xGDL

Compilation





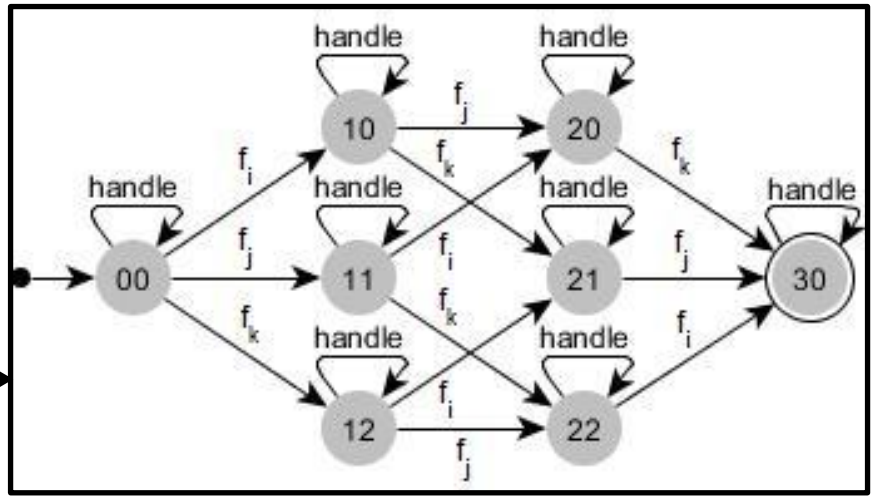
xGDL expression → NFA → DFA → **xGDL** Guide

Semantics

$$\frac{C_1 \xrightarrow{a} C_1'}{C_1||C_2 \xrightarrow{a} C_1'||C_2} \qquad \frac{C_2 \xrightarrow{a} C_2'}{C_1||C_2 \xrightarrow{a} C_1||C_2'}$$

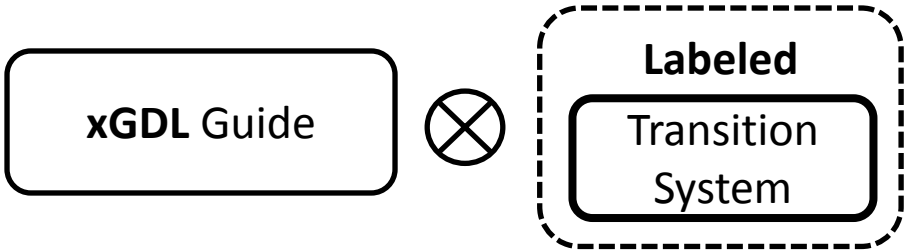
$$\frac{}{\perp||C \xrightarrow{\tau} C} \qquad \frac{}{C||\perp \xrightarrow{\tau} C}$$

*handle** **|| {***0 , 3* **} of [***fi , fj , fk* **]**



xGDL

Composition



- Initial states $G_0 \times S_0$
- Synchronisation $a \neq \tau,\ (g,s) \xrightarrow{a} (g',s') \Leftrightarrow g \xrightarrow{a} g' \wedge s \xrightarrow{a} s'$
- Stuttering steps $(g,s) \xrightarrow{\tau} (g',s') \Leftrightarrow g = g' \wedge s \xrightarrow{\tau} s'$

Always possible to produce a *« neutral element »*

$$A = \{a_1, ..., a_n\}, G_{neutral} = (a_1 \square ... \square a_n)*$$

Initial Guide

Production & Soundness

LGS Requirements

[…] *Failures are irreversible*

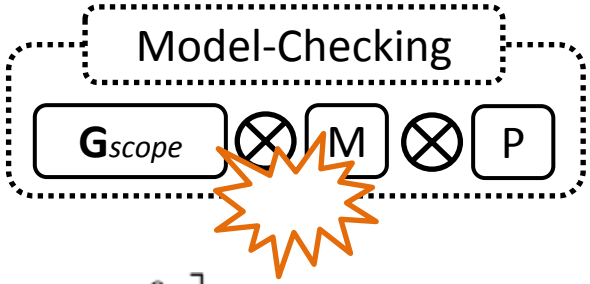
[…] *Four or more failures is outside the scope*

$$
\begin{aligned}
G_{neutral} &= (handle \square f_1 \square ... \square f_n)* \\
&= handle* \;||\; (f_1 \square ... \square f_n)* \\
G_{scope} &= handle* \;||\; \{0, n\} of \left[f_1, ..., f_n\right] \quad (uniqueness) \\
G_{scope} &= handle* \;||\; \{0, 3\} of \left[f_1, ..., f_n\right] \quad (at\ most\ 3)
\end{aligned}
$$

$$
\frac{\mathbf{G}_{scope} \otimes \text{M} \otimes \text{P}}{\text{M} \otimes \text{P}}
$$

Splitting the analysis

Illustration

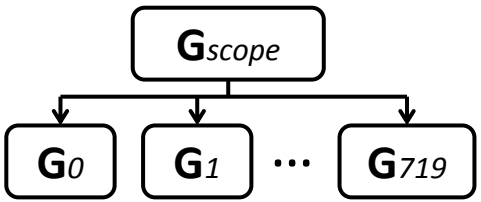


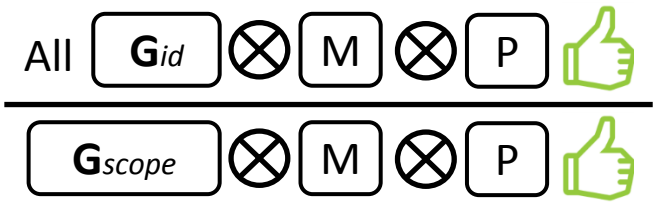
$$G_{scope} = handle * \quad || \quad \{0,3\} of \left[f_1, ..., f_n\right]$$

At most three failures may happen in one execution.
There are 720 distinct subsets of three failures.

$$G^3_{id} = handle * \quad || \quad \{0,3\} of \left[f_i, f_j, f_k\right]$$

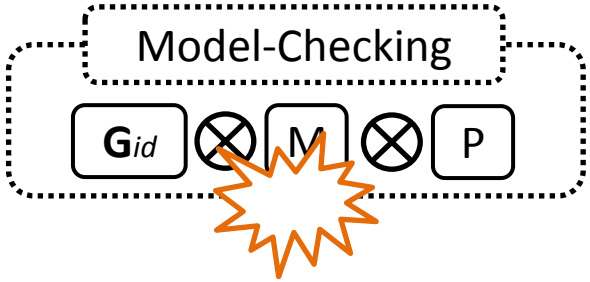
$$language(G_{scope}) \quad = \quad \cup_{id=0}^{719} language(G^3_{id})$$

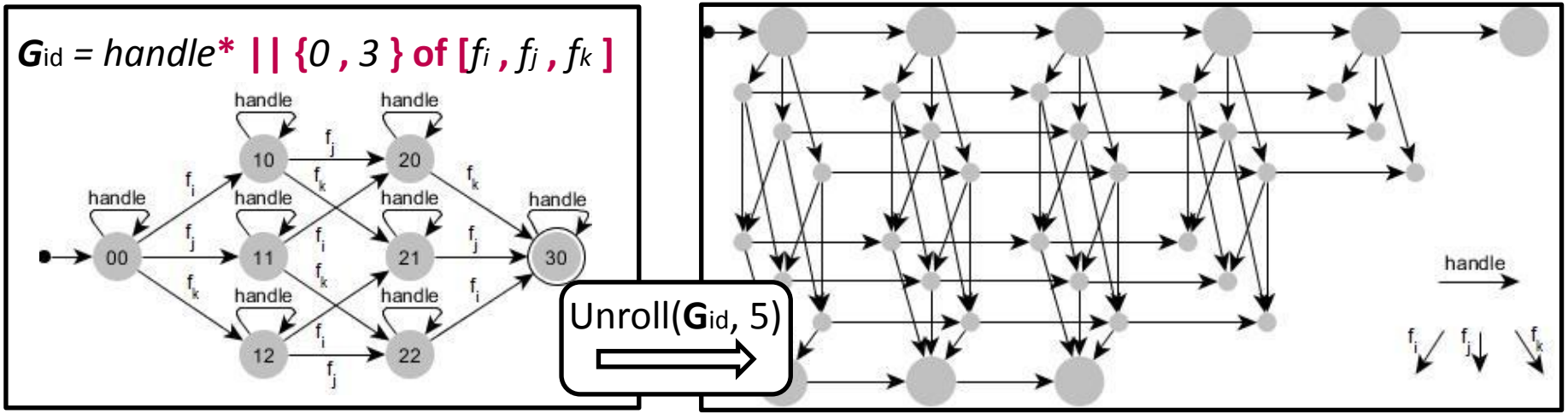




Partially Bounded

Unrolling the guide





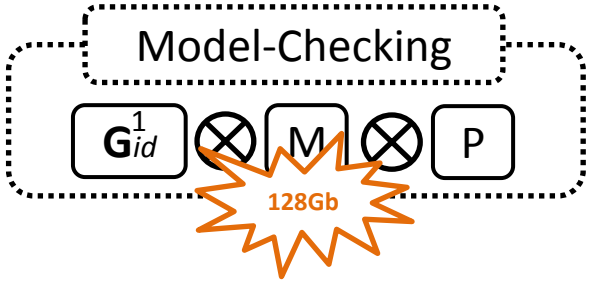
DAG specific algorithms from CaV literature

- Split: an automatic, recursive decomposition
- PastFree[ze]: reduces memory load

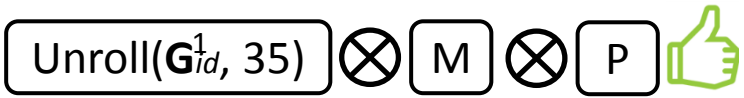
Soundness ?

Partially Bounded

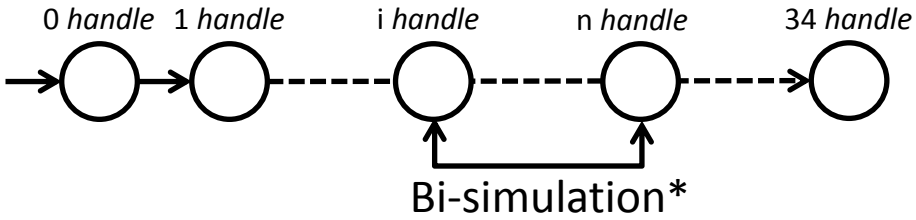
Soundness



$$G^1_{id} = handle * \quad || \quad f_i$$



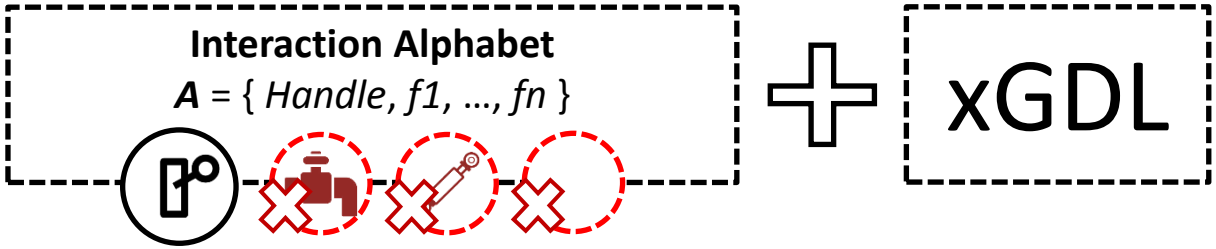
Resulting state space (indexed) :

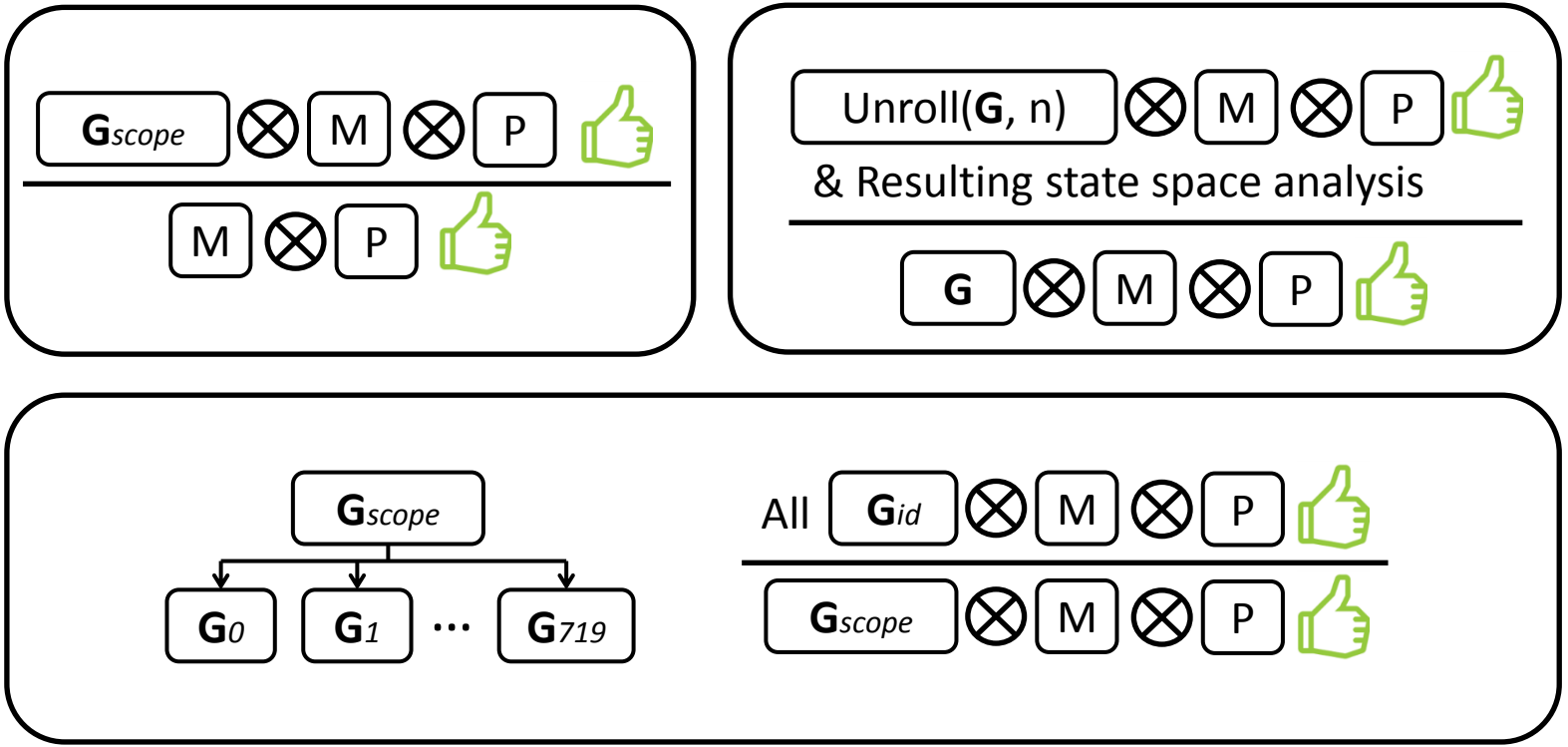


Failure	f_{1_1}	f_{1_2}	f_{2_1}	f_{2_2}	f_{3_1}	f_{3_2}	f_{4_1}	f_{4_1}	f_{5_1}	f_{5_2}	f_{6_1}	f_{6_2}	f_7, f_8, f_9	f_{10}, f_{11}, f_{12}
Bound	16	16	18	17	20	20	18	20	20	X	18	X	20	20

Table 2. Unrolling bounds required for completeness

Conclusion


Interaction Alphabet
A = { Handle, f1, …, fn }
xGDL


Gscope ⊗ M ⊗ P
M ⊗ P
Unroll(G, n) ⊗ M ⊗ P
& Resulting state space analysis
G ⊗ M ⊗ P
Gscope
G0 G1 … G719
All Gid ⊗ M ⊗ P
Gscope ⊗ M ⊗ P

Future Works

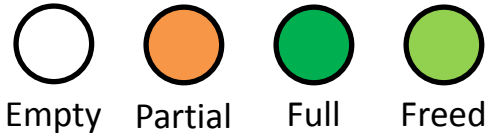
- PastFree[ze] with DFAs (cycles)
- Tooling / automation of the induced state clusters bi-simulation
- Usage in a collective and heterogeneous verification task

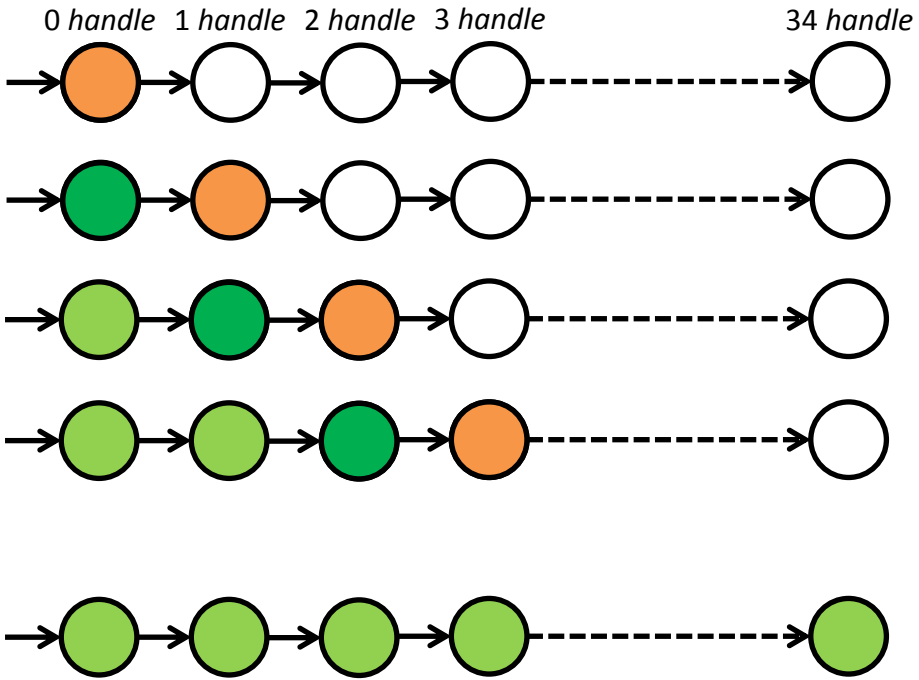
Tusen takk!

(thank you!)

Questions

PastFree[ze]


Empty
Partial
Full
Freed


0 handle
1 handle
2 handle
3 handle
34 handle