Automating the Analysis of the Finite State Machines at CERN

Yi Ling Hwong Jeroen Keiren Vincent Kusters
Sander Leemans Tim Willemse

Namur, February 9, 2012

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

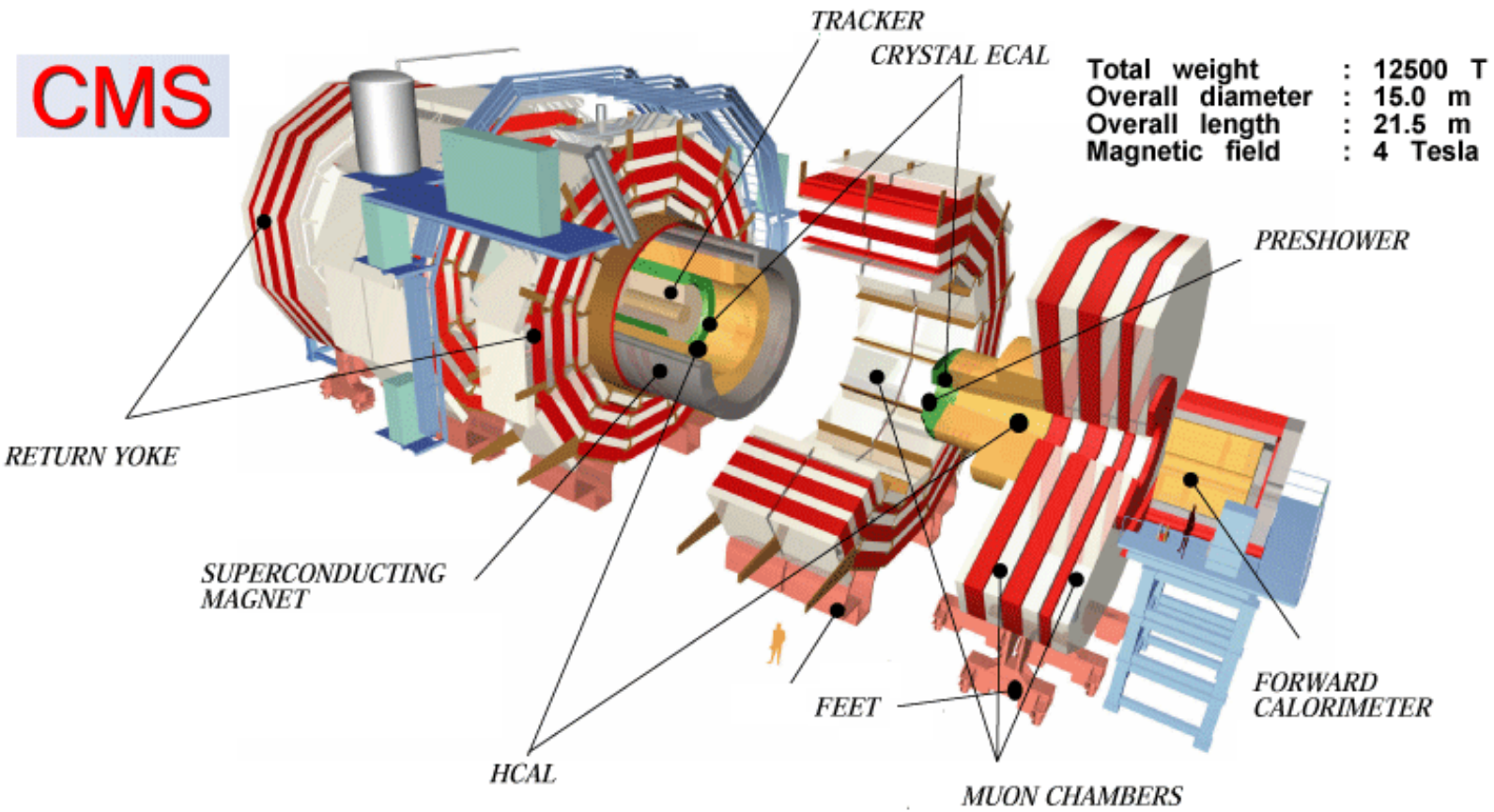
/ Department of Mathematics and Computer Science

CERN

Thanks to Vincent Kusters for the slides!

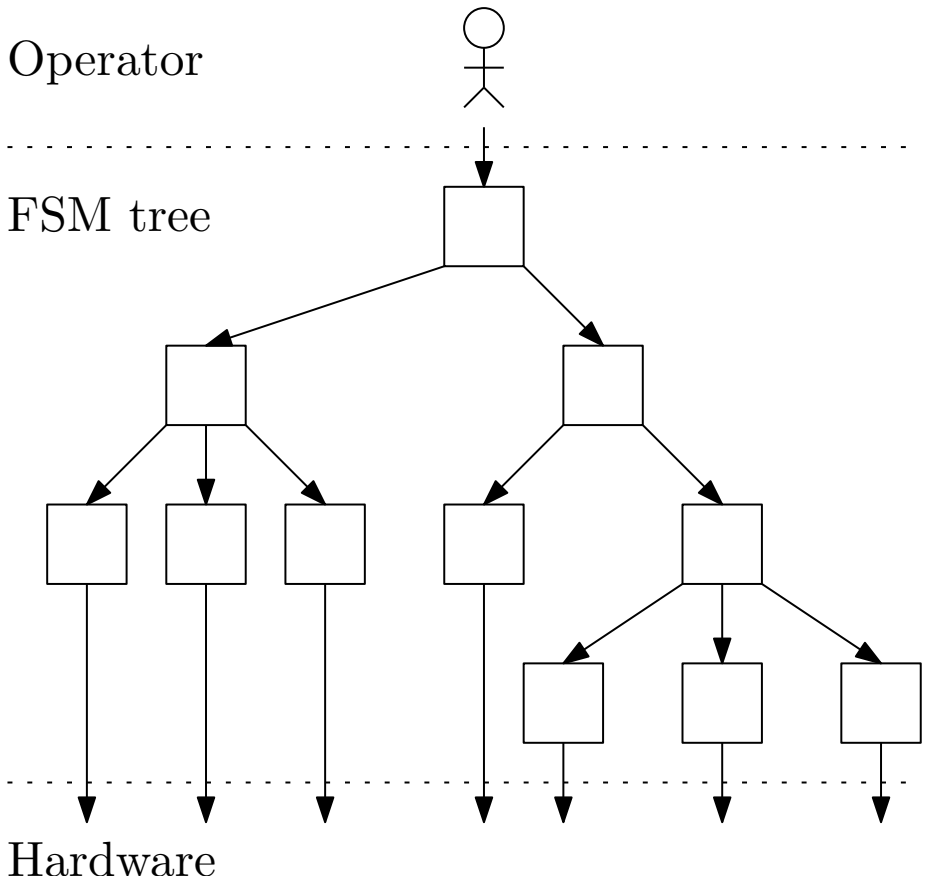
Large Hadron Collider

Compact Muon Solenoid



Finite State Machines

CMS uses over 30,000 Finite State Machines to do the supervisory control of the detector.

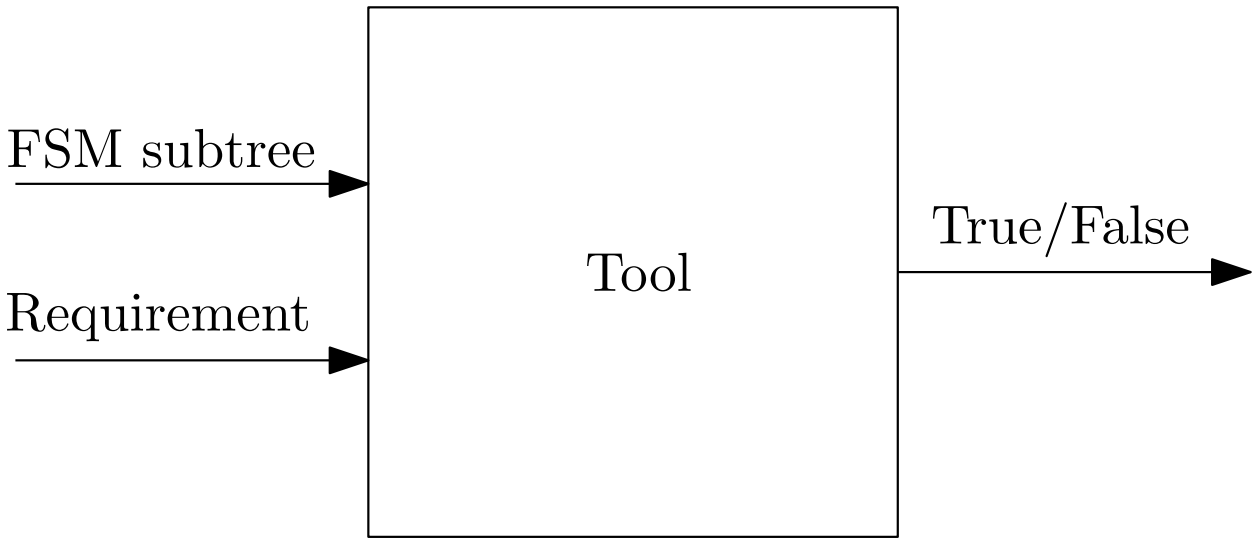


Problem

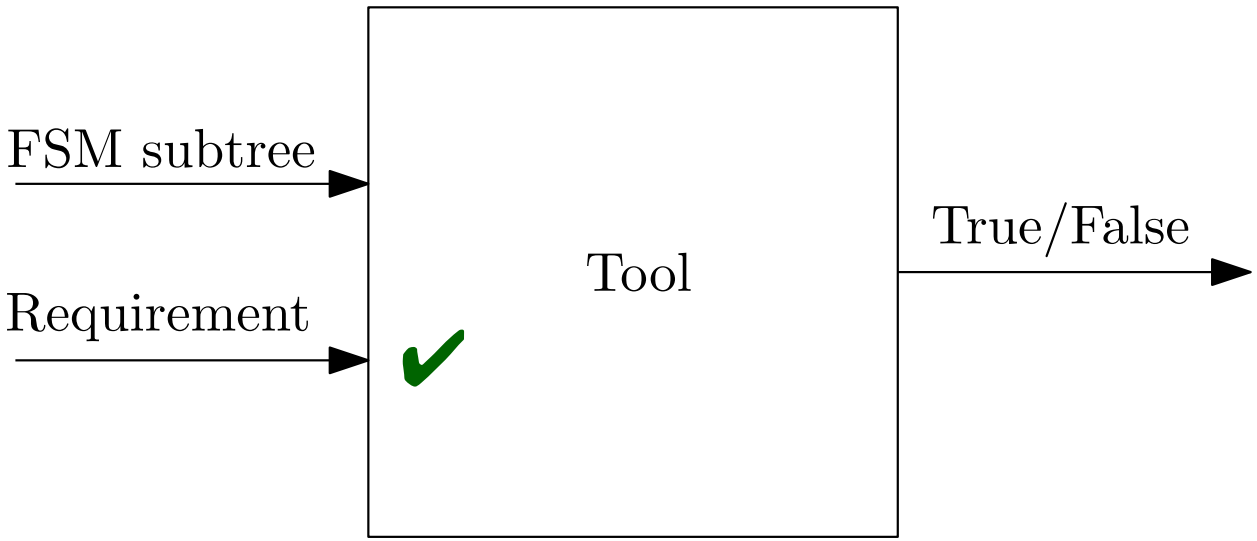
The system exhibited some problems during operation.

Find errors in CMS finite state machines using mCRL2 based verification tools.

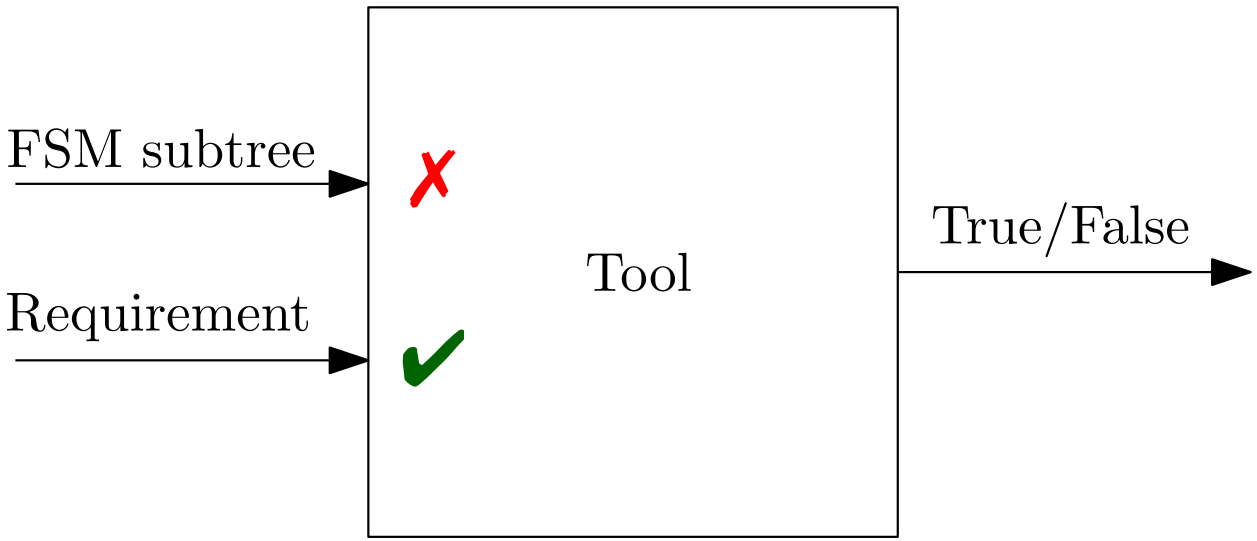
Objective



Objective



Objective



State Manager Language

SML

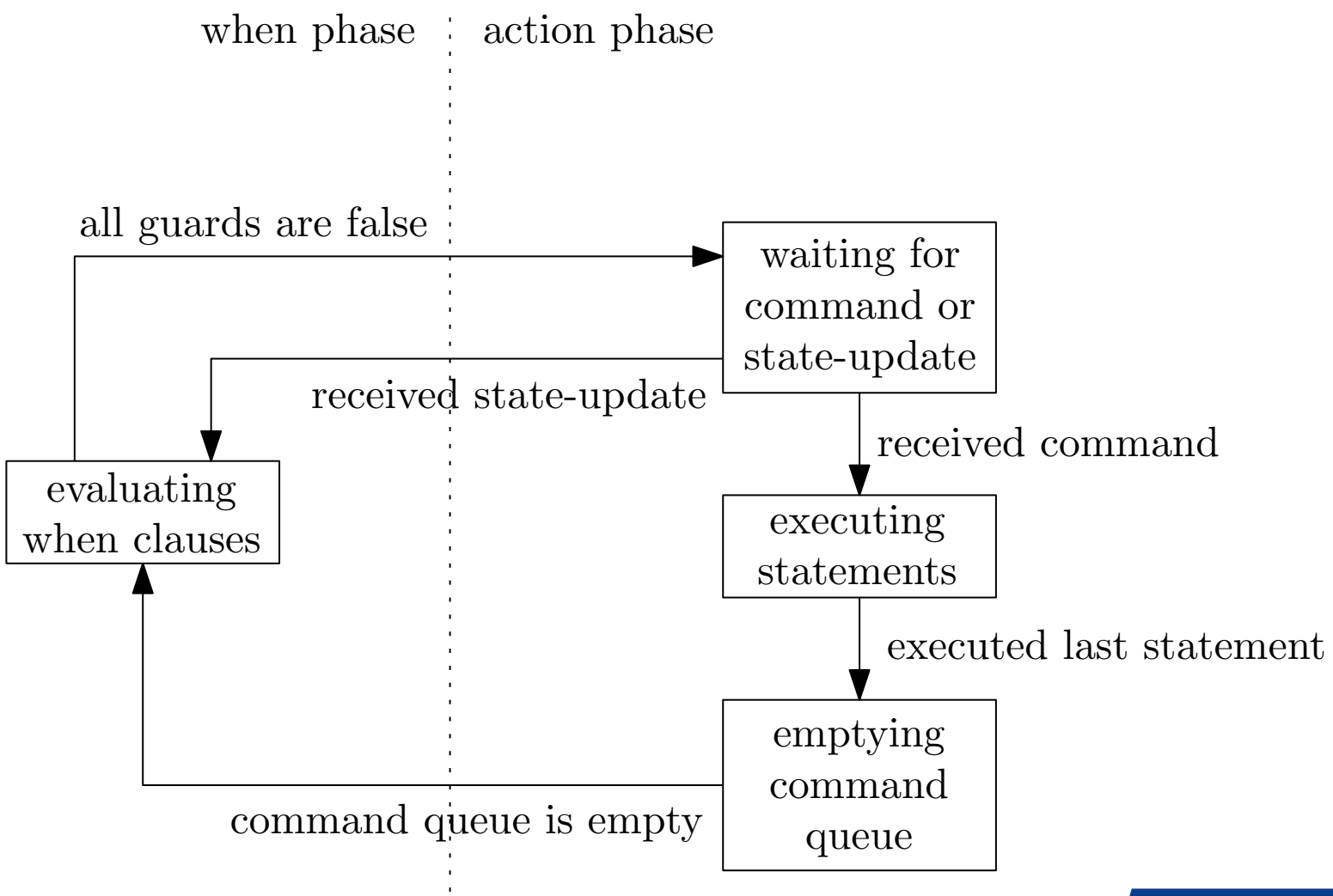
```
class: $FWPART_$TOP$RPC_Chamber_CLASS
  state: OFF
    when (($ANY$FwCHILDREN in_state ERROR) or
          ($ANY$FwCHILDREN in_state TRIPPED))
          move_to ERROR

    ...

    action: STANDBY
      do STANDBY $ALL$RPC_HV
      do ON $ALL$RPC_LV
```

Formalization

Approach to formalizing SML:

1. Interview developers.
2. Formalize a small subsystem in the mCRL2 process algebra.
3. Validate using simulation and model checking.
4. Automate translation.

Phases


when phase
action phase
all guards are false
waiting for command or state-update
received state-update
received command
evaluating when clauses
executing statements
executed last statement
emptying command queue
command queue is empty

Translating SML to mCRL2

SML

```
class: Chamber
  state: S0
    when C1 move_to S1
    when C2 move_to S2
    ...
```

Translating SML to mCRL2

SML

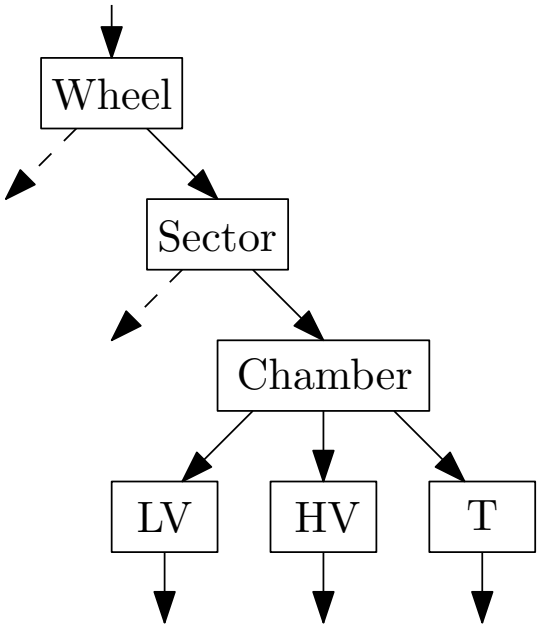
```
class: Chamber
  state: S0
    when C1 move_to S1
    when C2 move_to S2
    ...
```

mCRL2

```
proc Chamber(id, p, chs, state, phase) =
  isS0(state) && isWhenPhase(phase) && C1 ->
    move_to(id, S1) . Chamber(id, p, chs, S1, phase) <>
  ...
  isS0(state) && isWhenPhase(phase) ->
    send_state(id, p, state).
    move_phase(id, ActionPhase).
    Chamber(id, p, chs, state, ActionPhase)
```

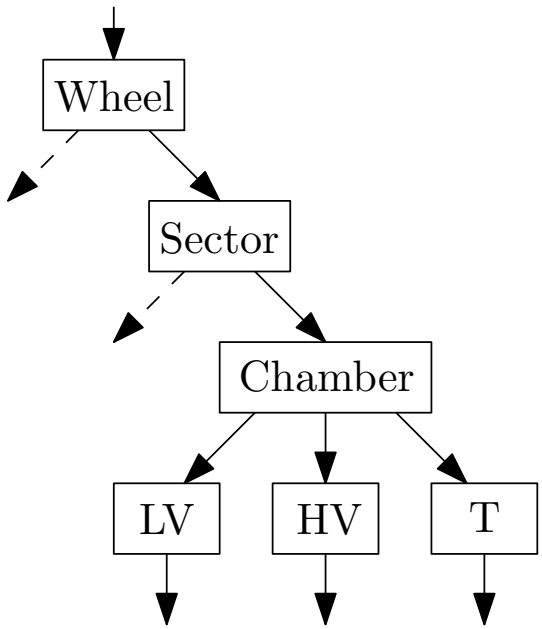
Validation

Translation was validated in a case study.



Global and local properties

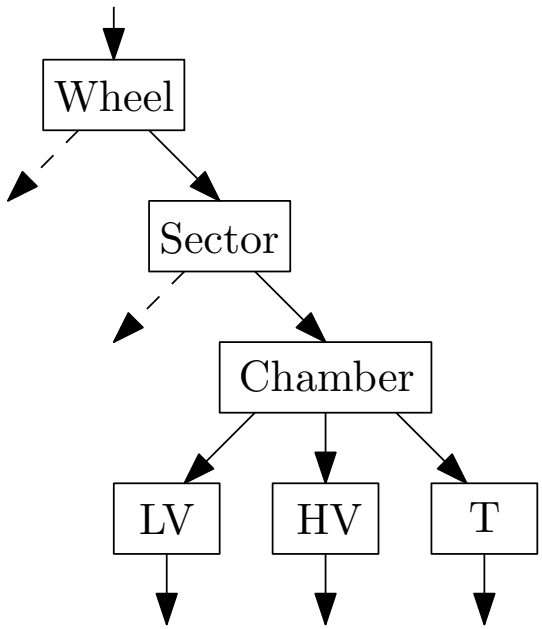
With mCRL2 we can check a wide range of properties of the system. This includes global properties like propagation of commands.



Global and local properties

With mCRL2 we can check a wide range of properties of the system. This includes global properties like propagation of commands.

Checking properties suffers from the state-space explosion problem.

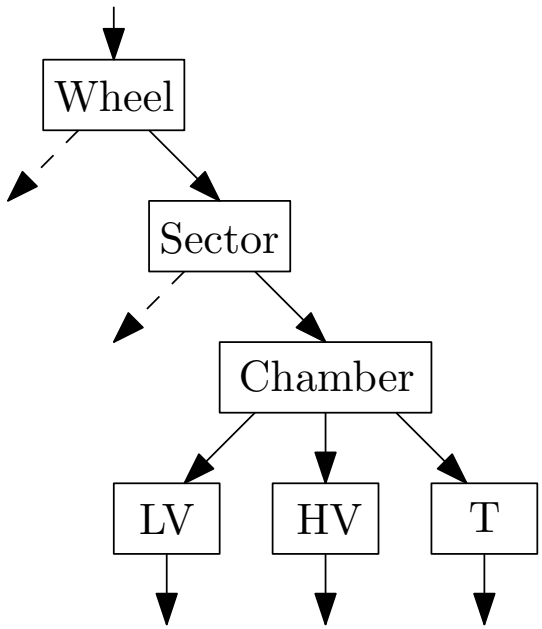


Global and local properties

With mCRL2 we can check a wide range of properties of the system. This includes global properties like propagation of commands.

Checking properties suffers from the state-space explosion problem.

Observation: checking for some properties can be done locally.



move_to loop example

Example

```
class ECALfw_CoolingDee:
  state: ERROR
    when ( $ANY$FwCHILDREN in_state NO_CONNECTION )
      move_to NO_CONNECTION
    when ( $ALL$FwCHILDREN in_state OK )
      move_to OK

  state: NO_CONNECTION
    when ( $ALL$FwCHILDREN in_state OK )
      move_to OK
    when ( $ANY$FwCHILDREN in_state ERROR )
      move_to ERROR
```

Results

- Formalized most of SML in mCRL2.
- Formalization in mCRL2 basis for discussion with developers.
- Verification possible with mCRL2.
- Large scale verification of local problems done using SMT.
- Errors found in 6/40 systems.