ICTEAM seminar

Formal Verification of Railway Interlocking

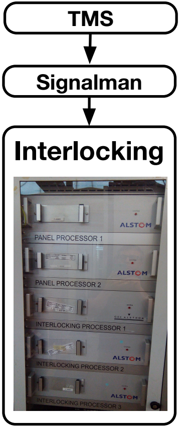
Christophe Limbrée

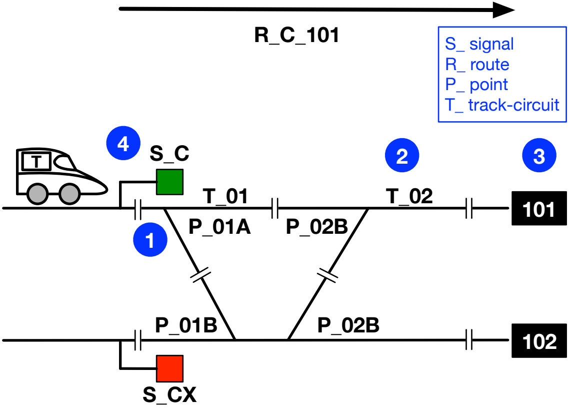
Ecole Polytechnique de Louvain

20 November 2017

Elements of railway signaling (e.g., signals, points, LX, . . .)

Interlocking context





S_ signal
R_ route
P_ point
T_ track-circuit

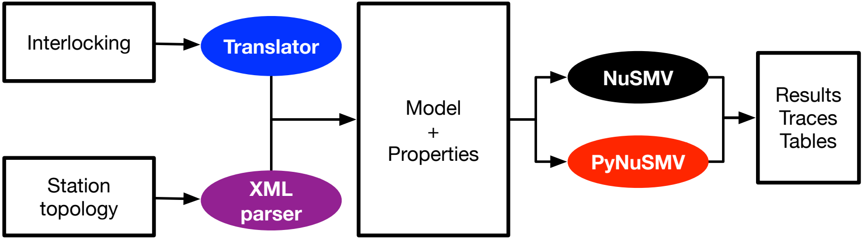
Research subject

```
*Q_R(C_101)
  if  R_C_101 xs
    P_01A cfn , P_01B cfn,
        P_02AB cfr, P_02B cfr
    U_IR(01A) f , U_IR(02B) f
  then  R_C_101 s
    P_01A cn , P_01B cn , P_02A
        cr , P_02B cr ,
    U_IR(01A)  l , U_IR(02B) l
    U_CBSPA(101) l
    S_C clear bpull
```

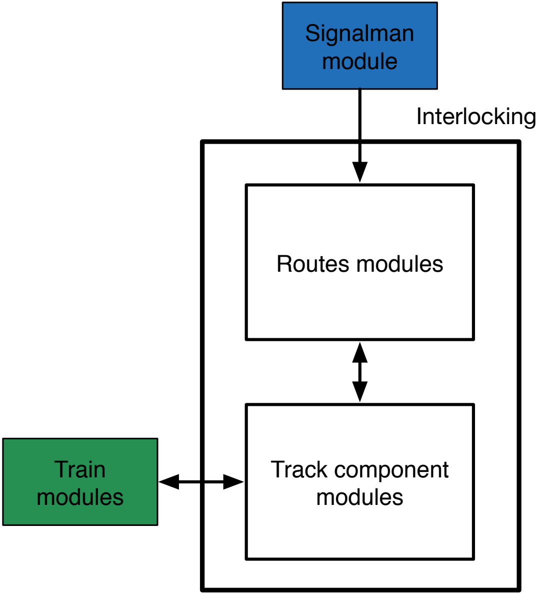
V&V:

- Combinatorial testing
- ≥ 2 month/int.
- Tedious job
- Error prone

Process of verification



Model - high level hierarchy



Model checking - Formal foundations

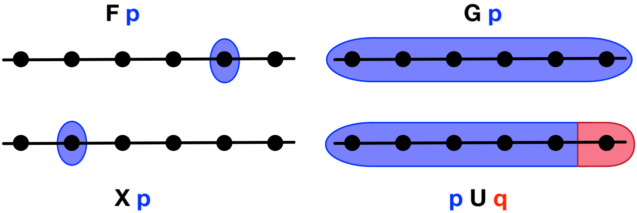
Model checking [2] Given a model of a system, exhaustively and automatically check whether this model meets a given property.

$$M, s \models P \text{ with } P \text{ a property}$$

[1] define a **Kripke structure** over AP as a 4-tuple $M = (S, I, R, L)$ consisting of:

- a finite set of states S.
- a set of initial states $I \subseteq S$.
- a transition relation $R \subseteq S \times S$ (i.e., $\forall s \in S \; \exists s' \in S$ such that $(s, s') \in R$).
- a labeling function $L : S \rightarrow 2AP$ with AP a set of atomic propositions.

Properties (MC [4], invariants, temporal logic (e.g.; LTL)



```
   INVARSPEC !(train_collision)
   INVARSPEC (train_on_switch -> !switch_moves)
   LTLSPEC   G(train_enters -> X signal_goes_red)
```

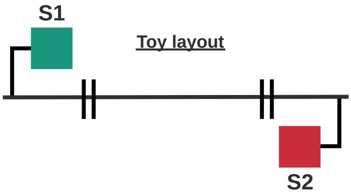
A toy example

```
MODULE main
VAR
S1 : {red, green};
S2 : {red, green};

ASSIGN
init(S1) := red;
init(S2) := red;

next(S1) :=
 case
  S1 = red : {red, green};
  TRUE     : red;
 esac;
next(S2) :=
 case
  S2 = red : {red, green};
  TRUE     : red;
 esac;

--INVARSPEC !(S1 = green &
     S2 = green)
```



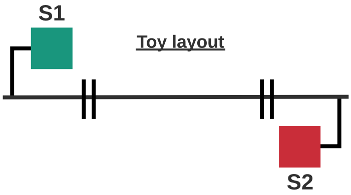
A toy example

```
MODULE main
VAR
S1 : {red, green};
S2 : {red, green};

ASSIGN
init(S1) := red;
init(S2) := red;

next(S1) :=
 case
  S1 = red  : {red, green};
  TRUE      : red;
 esac;
next(S2) :=
 case
  S2 = red  : {red, green};
  TRUE      : red;
 esac;

--INVARSPEC !(S1 = green &
     S2 = green)
```



```
nuXmv > check_invar
-- invariant !(S1 =
    green & S2 = green)
      is false
Trace Type:
    Counterexample
  -> State: 1.1 <-
    S1 = red
    S2 = red
  -> State: 1.2 <-
    S1 = green
    S2 = green
```

Conclusions and questions

- From Interlocking to model in SMV
  - Namêche
  - Braine l'Alleud
  - Ottignies
- Compositional approach for larger stations [3]
- Bottleneck → Space state explosion

→ **Questions**

Biblio

Jr. Edmund M. Clarke, Orna Grumberg, and Doron A. Peled.
Model Checking.
The MIT Press, 1999.

Clarke, Edmund M.
25 Years of Model Checking.
chapter The Birth of Model Checking, pages 1–26. Springer-Verlag, Berlin, Heidelberg, 2008.

Limbrée, Christophe and Cappart, Quentin and Pecheur, Charles and Tonetta, Stefano.
Verification of railway interlocking-compositional approach with ocra.
In *International Conference on Reliability, Safety and Security of Railway Systems*, pages 134–149. Springer International Publishing, 2016.

Roberto Cavada and Alessandro Cimatti and Michele Dorigatti and Alberto Griggio and Alessandro Mariotti and Andrea Micheli and Sergio Mover and Marco Roveri and Stefano Tonetta.
The nuXmv Symbolic Model Checker.
In *Computer Aided Verification - 26th International Conference, CAV 2014, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 18-22, 2014. Proceedings*, pages 334–342, 2014.