Systeme Hoher Qualität und Sicherheit
Vorlesung 12 vom 20.01.2014: NuSMV and Spin

Christoph Lüth & Christian Liguda

Universität Bremen

Wintersemester 2013/14

Where are we?

- Lecture 1: Concepts of Quality
- Lecture 2: Concepts of Safety and Security, Norms and Standards
- Lecture 3: Quality of the Software Development Process
- Lecture 4: Requirements Analysis
- Lecture 5: High-Level Design & Formal Modelling
- Lecture 6: Detailed Specification, Refinement & Implementation
- Lecture 7: Testing
- Lecture 8: Program Analysis
- Lecture 9: Verification with Floyd-Hoare Logic
- Lecture 10: Verification Condition Generation
- Lecture 11: Model-Checking with LTL and CTL
- **Lecture 12: NuSMV and Spin**
- Lecture 13: Conclusions

Organisatorisches

- Fachgesprächstermine über Stud.IP (2./3. Februar).
- Für eine Modulprüfung: bitte zwei **aufeinanderfolgende** Termine buchen.
- Fachgespräche in der Gruppe, Prüfung alleine.
- Helft uns, die Veranstaltung zu verbessern: Nehmt an der **Evaluation** unter Stud.IP teil!

Introduction

- In the last lecture, we saw how to model systems as **finite-state machines**, and how to specify properties about these in temporal logic — namely, **linear temporal logic** (LTL) and **computational tree logic** (CTL).

- The idea was to allow **automatic** verification or disproving of the properties by **model-checkers** which enumerate the system states.

- Today, we look at two prominent model-checkers: **NuSMV2** and **Spin**. If time permits, we might also look at an interactive theorem prover.

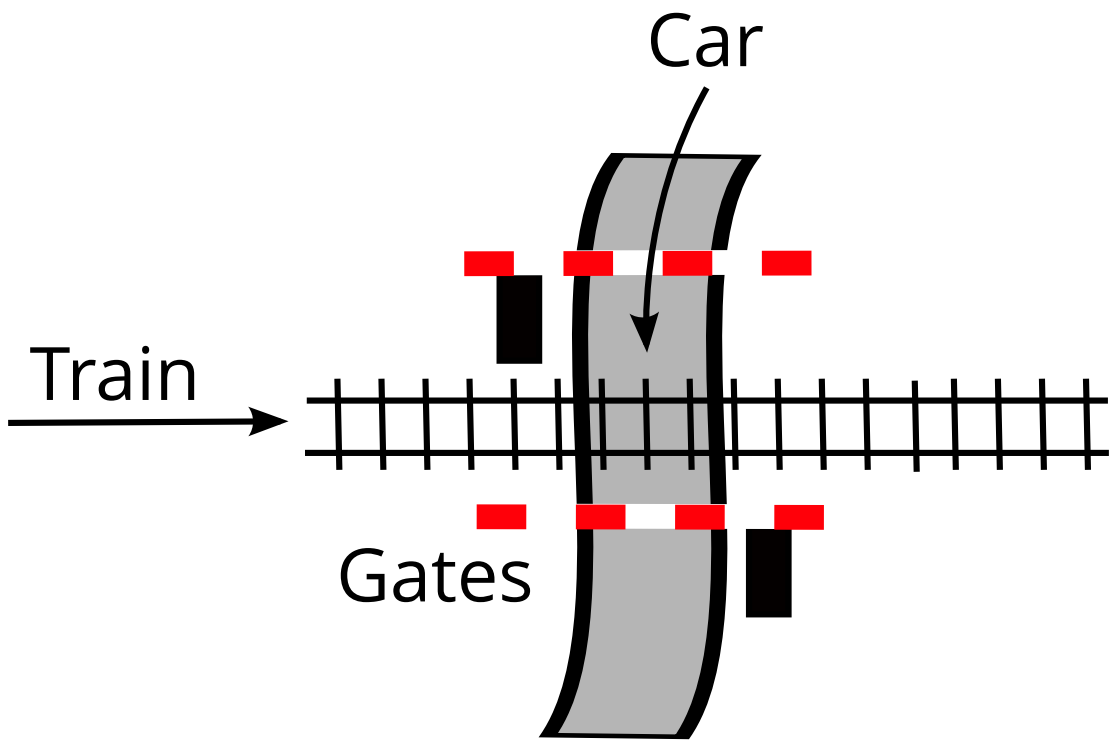
NuSMV

- **NuSMV2** originated with SMV model checker (Edmund Clarke, Ken McMillan). SMV was the first m/c to use BDDs (binary decision diagrams) to represent the transition relation, allowing for much more compact state representation (around 1990). As a result, it could represent up to 10^{20} states.
- **NuSMV2** is currently maintained by CMU, FBK-irst (Trentino, Italy), University of Genoa and University of Trentino.
- It allows simulation, tracing, and supports both LTL and CTL specifications.
- Web Site: `http://nusmv.fbk.eu/`

Spin

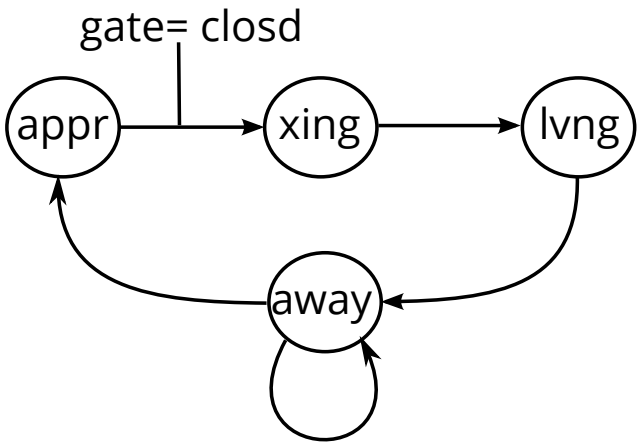
- **Spin** was written by Gerard Holzman. It originated with a protocol analyser (PAN) in 1980, which became Spin in 1989.
- Spin uses the language **Promela** for modelling. As opposed to NuSMV, it allows to model **processes** and communication between them via **channels**. The key difference is that Spin is **asynchronous**, whereas NuSMV is **synchronous**.
- Spin generates a program representing the model, which does the actual model-checking. Besides higher speed, it allows a much more flexible approach to modelling (e.g. one can inject C code into the Promela model).
- Web Site: `http://spinroot.com/`

Recall: The Railway Crossing

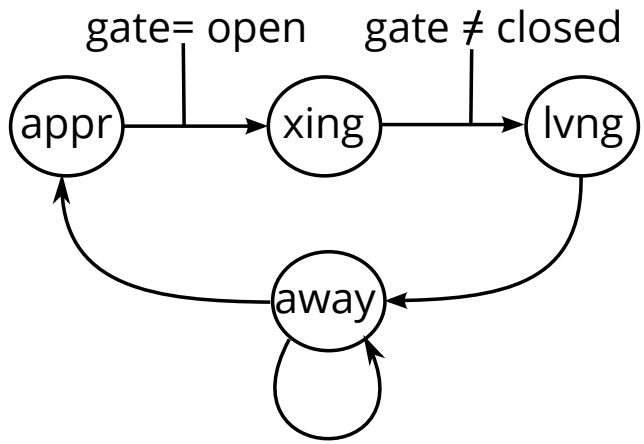


Modelling the Railway Crossing

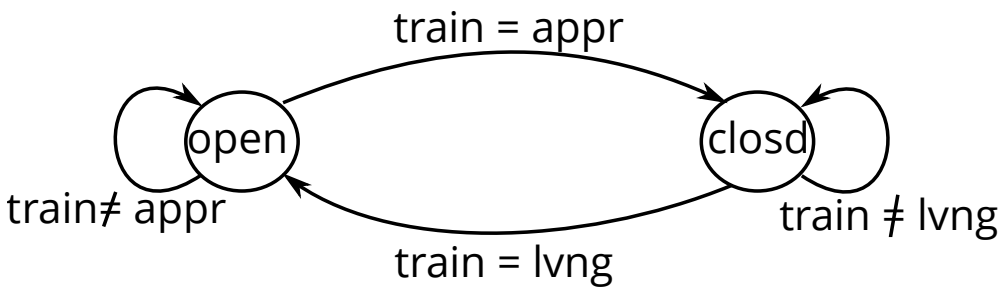
States of the train:



States of the car:



States of the gate:



Summary

- NuSMV vs. Spin:
  - Spin (Promela) is more **concrete**, closer to a programming language.
  - NuSMV supports CTL as well as LTL.
- Model-checking:
  - Can we trust the results? If it finds errors, we get **counter-examples**, but how reliable are positive results?
  - And just how good is our model?