MFS TP1 23/24 - Design in Alloy

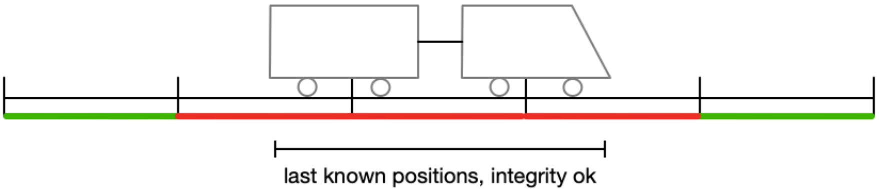
Context

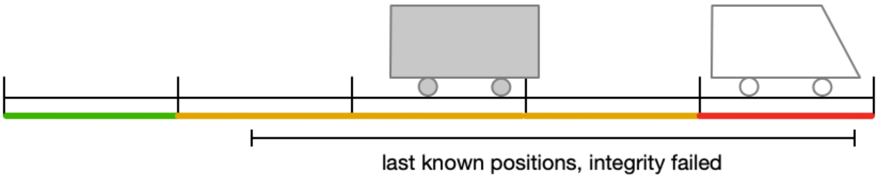
Trains move according to a *movement authority* (MA) issued by a control center. Traditionally, the control center issues MAs based on information collected trackside, such as balises, to identify track sections occupied by a train. However, this is costly and provides coarse-grained information. Currently, trackside information is combined with on-board information reported directly by the train, providing finer-grained information and allowing more efficient traffic. But since communication problems or train malfunctions may occur, trackside information is still used as a fail safe. Nonetheless, proposals have been made to completely forgo the trackside detectors and fully rely on the on-board train information, such as the Level 3 of the European Train Control System (ETCS). Here, we'll consider a simplified version of this protocol.

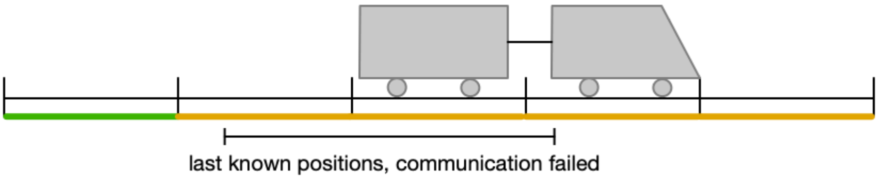
In systems without trackside detectors, the control center sees the train track as a sequence of virtual sub-sections (VSSs) whose occupation is determined by the information communicated by the trains. Trains must also employ a Train Integrity Monitoring System (TIMS) in order to detect any disconnected car. Given this, each VSS can be in 3 states, which are then used to issue MAs:

- **Free**: if the control center knows for sure that the VSS is free
- **Occupied**: if the control center knows for sure that the VSS is occupied
- **Unknown**: if the control center cannot conclude whether the VSS is free or occupied

There are two main reasons for the state to become unknown. Communication with the train has failed, and the control center does not know the train current position; or the train reported a loss of integrity, meaning there is a car somewhere along the track after the last reported position. Below are some example scenarios.







Goal

The goal of the first project of MFS 23/24, **TP1**, is to *model*, *validate* and *verify* the software controller of a Level 3 train control center using Alloy. Each student group is free to choose the level of abstraction, features, and properties to verify. The goal is to exercise the different phases of trustworthy software design studied in the course, and not achieve a realistic system. It's acceptable to have expected properties failing, as long as properly justified. In particular, the following topics should be covered:

- **Structural design (8pts)**
    - **(4pts) Model** the structure of the system, alongside any additional constraints
      Examples: define how the track and the trains can be configured.
    - **(2pts) Validate** the structural model of the design

      Examples: define run commands to explore expected configurations, define a theme to better visualize such configurations.
    - **(2 pts) Specify** and **verify** desirable structural properties

      Examples: define properties that should hold for every valid track and train configuration of the robot, explain why they don't hold if verification fails.
- **Behavioral design (12pts)**
    - **(1 pts) Model** the dynamic elements of the system
      Examples: define how the train positions and VSS states are represented
    - **(4 pts) Model** the events that allow the system to evolve

      Examples: define how the train moves, and how the VSS state is calculated
    - **(3 pts) Validate** the behavioral model of the design

      Examples: define run commands to explore expected scenarios, adapt the theme to better visualize mutable elements.
    - **(4 pts) Specify** and **verify** desirable behavioral properties

      Examples: define temporal properties that should hold for every execution trace, explain why they don't hold if verification fails.

Submission

Projects will be developed in groups of 3 students. The submission will be via the Moodle assignment "MFS TP1" by **23:59 March 30**, and should contain the following files:

- The Alloy file(s) describing the model (including a theme to support visualization);
- A short report (max 5 pages) describing and discussing the steps listed above.

A short discussion with the teacher regarding the developed models and the followed strategy will take place in the class after submission on **April 2**. After that discussion groups will be able to submit an updated version of the report until **23:59 April 5** if they wish to integrate the feedback from the teacher.