POLITECNICO MILANO 1863

Model Checking of Battery-Powered Railway Lines

Formal Methods for Concurrent and Real-Time Systems
a.y. 21/22

Prof. Pierluigi San Pietro
Livia Lestingi

1 Introduction

The fight to reduce greenhouse gas emissions is bringing together researchers and manufacturers from all over the world. In particular, rechargeable batteries as a source of power in place of fossil fuels are already widespread in cars and making their way into the rail transport sector. Battery-powered trains are already operative in several countries like Japan, Austria, and Britain.[1] Italy is also planning on producing and deploying fully-electric trains starting mid-2022, thanks to a deal with Hitachi Rail.

Like any electric vehicle, trains can cover a **limited distance** running only on battery power before needing to recharge. Electric trains can either recharge *en route* (if the line is electrified) or while in a **station**: we will only consider the second case for this project. Nevertheless, trains must still reach the following station on time. In case of excessive **delay**, the company is obliged to issue monetary compensation to the passengers.

As engineers with a background in formal methods, you are in charge of formally verifying an imaginary railway line operating only battery-powered trains. Precisely, given a set of simplifying assumptions, you will **model** the main actors of the system as a network of **Timed Automata** (TA) whose behavior depends on specific key parameters. You will then investigate alternative configurations to **verify** whether trains always have sufficient power to reach their destination without causing service disruption.

Section 2 explains the mandatory entities to be modeled, how they are supposed to synchronize and their main characteristics.

Section 3 explains the mandatory properties to be verified.

Section 4 provides instructions on the modeling and verification tool that all teams must use.

Section 5 provides instructions on how to deliver your project.

2 Model

The upcoming subsections introduce the mandatory features to be formally modeled. Unless explicitly stated, you decide what to model as a TA feature and what via a variable for each described entity.

Railway Line

A line is a sequence of stations with two terminal stops. After reaching a terminal station, a train starts traveling again along the line in the opposite direction.

Each line is characterized by the **distance** between each pair of consecutive stations. You can assume that all trains stop at all stations.

Each line is also characterized by the **maximum delay** allowed between two consecutive stations before the company is obliged to reimburse the passengers. You can assume that the maximum delay $\mathsf{D}_{\mathsf{i,i+1}}$ allowed between consecutive stations s_{i} and $\mathsf{s}_{\mathsf{i+1}}$ is the same while traveling in both directions.

[1] Learn more at: https://www.youtube.com/watch?v=Q4JpoR0mJls.

▷ Your model must feature 1 line (trains do not switch between different lines within the same model), with (at least) 3 stations.

Station

Stations host trains while they recharge and until they leave for the following station.

Each station has a maximum number of **tracks** available, indicated as T. Parameter T can be different for different stations in the same line. When a train reaches a station, it occupies one of the tracks.

There can never be more trains in a station than its available tracks.

▷ Stations in your model must have at least one track ($\mathsf{T} \geq 1$).

▷ At least one station in the line must have less tracks than the total number of trains circulating in the line (constituting, thus, a potential bottleneck).

Train

Each train can either be waiting in a station or traveling between two stations.

Assume that trains travel at **constant speed**, indicated as V. Two trains can have different values for parameter V. When a train is traveling, the time required to reach the destination is equal to the distance between the two stations divided by V (pay attention to the units of measurement).[2]

When a train is done traveling, it asks the target station for the permission to enter. If there is at least one track available, the station approves the request and the train can enter. Otherwise, the station turns down the request. If the train cannot enter, it periodically re-submits the request every $\mathsf{R} > 0$ time instants.

Trains are exclusively battery-powered. It is not necessary to model the relationship between battery voltage and residual driving distance. Let the **distance** that can be covered with the residual charge be indicated as C. Variable C grows while the train recharges, and decreases when it is traveling. When C drops to 0, the train can no longer move. C can never grow above a threshold $\mathsf{C}_{\mathsf{max}}$ (the distance that can be covered when the battery is fully charged). Two trains can have different maximum ranges (i.e., values of $\mathsf{C}_{\mathsf{max}}$).

When a train is traveling or waiting outside a station, C decreases **linearly** with time. When a train is parked inside a station, C increases **linearly** with time. Time-dynamics of C are summarized by Eq. 1, where $\mathsf{c}_{\mathsf{dis}}, \mathsf{c}_{\mathsf{rec}} > 0$ are constant parameters (that may vary between trains).

$$\mathsf{C}(t) = \begin{cases} -\mathsf{c}_{\mathsf{dis}} \cdot t & \text{if moving/waiting and } \mathsf{C} > \mathsf{0} \\ \mathsf{c}_{\mathsf{rec}} \cdot t & \text{if parked and } \mathsf{C} < \mathsf{C}_{\mathsf{max}} \\ \mathsf{C}(t-1) & \text{otherwise} \end{cases} \quad (1)$$

[2] Uppaal has no feature to explicitly specify measurement units, but make sure that parameter values are consistent.

Assume that, while in a station, a train is always recharging. Let the time it spends in a station before leaving again be indicated as W. Variable W must have a lower bound to allow passengers to get on and off the train (i.e., the train cannot enter and leave the station in 0 time units). W is updated through "recharge policies". You are in charge of designing the recharge policy of the railway line. Remember that:

- the train should reach the following station within a deadline.
- the train should have enough power to reach the following station.

▷ Your model must feature at least 3 trains.

▷ Size their parameters so that the system configuration is not trivial (e.g., $\mathsf{c}_{\mathsf{rec}} \approx \infty$ is not allowed).

3 Properties

You have to formally verify that the railway line operates **safely**, which depends on the set of parameters and the recharge policy.

The **mandatory** safety properties to be verified are:

P1. It never happens that a train reaches the destination after the deadline.

P2. It never happens that a train runs out of power (C drops to 0) while traveling or waiting to enter a station.

Additional correctness properties (for example, "the number of trains parked in a station never exceeds the available tracks") are welcome though not mandatory. Nevertheless, keep in mind that this is an additional tool to check the *soundness* of your model.

For each line you formally model (remember that 1 is mandatory, but you can model multiple ones if you find it useful), deliver at least two different configurations. The configurations should feature either different parameters (characterizing the line, stations, and the trains) or different recharge policies. Precisely, you have to deliver at least one configuration that violates one (or both) safety properties and at least one that verifies both.

4 Modeling Tool

The project will be carried out using the Uppaal[3] tool.

The system must be modeled as a Network of **Timed Automata** (NTA) through the Uppaal GUI. Properties must be expressed in **TCTL** logic and verified through the Uppaal engine.

[3] Uppaal.org

5 Delivery Instructions

It is **mandatory** to deliver:

- a .pdf report (max 10 pages excluding front cover and bibliography, no constraint on the template) describing:
  - the model (emphasis on **critical** modeling choices you have made);
  - the system configurations you have chosen;
  - experimental results;
- the .xml Uppaal model. Make sure:
  - it is the same as what you present in the report
  - it is test-ready (i.e., it includes the queries you ran for the experiments and system configurations are easily selectable).

The deadline for the delivery is **June 24** (this includes students participating in the exchange program with UIC).

To submit your work, send an email to livia.lestingi@polimi.it with the report, Uppaal files, and any additional material you want me to evaluate.