Assignment 4
Translational Semantics in AToMPM

Simon Van Mierlo
simon.vanmierlo@uantwerpen.be

1 Practical Information

The goal of this assignment is to build a rule-based transformation that translates railway models to Petrinets. You then use this Petrinet to prove properties on railway models that are not trivially proved 'on sight' or by other methods.

Use the /Formalisms/__Transformations__/Transformation/MoTiF and /Formalisms/__Transformations__/TransformationRule/TransformationRule formalisms in combination with your RAMified domain-specific language to create the rules and schedule of your transformation.

Write a report that includes a clear explanation of your complete solution, the modelling choices you made, as well as an explanation of your testing process. Also mention possible difficulties you encountered during the assignment, and how you solved them. You will have to complete this assignment in groups of 2. Submit your assignment (report in pdf, abstract and concrete syntax definition, example models, and the complete rule-based transformation) on Blackboard before Friday, November 25, 14:00h.

Contact Simon Van Mierlo (simon.vanmierlo@uantwerpen.be) if you have a problem.

2 Requirements

The different parts of this assignment are listed below:

1. Implement a rule-based transformation that transforms railway models into Petrinets models.
   - Make sure this transformation is complete (*i.e.*, every well-formed railway model produces a well-formed Petrinet (in the /Formalisms/PN formalism)). For this assignment, you only have to consider models with two trains, whose end stations are reachable from their start stations.

- We will use Petrinets to find a schedule for the two trains. This means the source model of your transformation should not contain a schedule for the trains. In principle, this means the source model is not well-formed (because each train needs to have an associated schedule). You can also include two empty schedules, to satisfy the constraint.
- Do not remove your railway model during transformation. Instead, keep it and create traceability links between created Places and Transitions and the railway element they correspond to. Use the /Formalisms/GenericGraph formalism for traceability links. These links will greatly help in implementing your transformation.
- Petrinets are used to analyse safety properties of a system, such as deadlock. Needless to say, the Petrinets resulting from your railway models will be abstractions. These abstractions need to be useful with respect to the properties you want to prove. For this exercise, you will have to prove, for a model with two trains, whether there is a way to schedule these trains fully independently. "Fully independent" means that they will, on their way from the start to the end station, never enter a segment that was entered by the other train. Knowing that two trains can run fully independently is useful, as a scheduler then would not need to worry about those two trains ever crashing. Make sure to fully explain in your report how your Petrinets representation is used to prove this property.

2. Create test cases that test your Railway-to-PetriNet mapping.

3 Useful Links

- AToMPM main page: `http://www-ens.iro.umontreal.ca/~syriani/atompm/atompm.htm`
- AToMPM user manual: `https://msdl.uantwerpen.be/documentation/AToMPM/`
- AToMPM git repository: `https://msdl.uantwerpen.be/git/simon/AToMPM` (includes installation instructions)
- Useful tutorials can be found on the 'Tutorials & Demos' page on the main website.