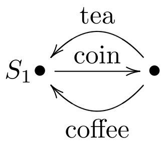
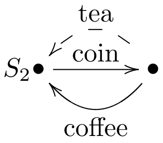
Bachelor/Master Thesis:
A Tool for Modal Transition Systems

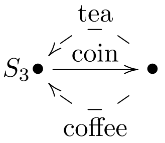
Modal transition systems

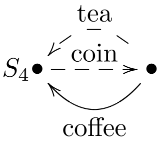
Modal transition systems (MTS) is an automata-like formalism used e.g. for the specification of a variety of software systems. The following figure depicts an implementation of a vending machine, which after inserting a coin produces tea or coffee. It is an example of a labelled transition system (LTS), i.e. an automaton without final states.



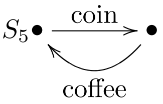
At an early stage of design, we might not know if some behaviour will be present in the final implementation or not. To express this, we use *modal* transition systems. These are equipped with so called *may* transitions, which are drawn as broken arrows and denote a behaviour that may or may not be present. For instance, S_2 must accept a coin and then must be able to produce coffee (unbroken arrow) and may also be able to produce tea. S_3 captures the demands of the owner of the machine: the machine must accept the coin and we do not care what it offers then. On the other hand, in S_4 we have a guarantee that if the machine accepts a coin, we must be offered coffee, which captures the coffee-consumer's viewpoint.

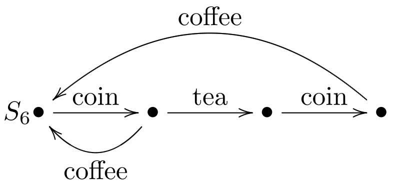


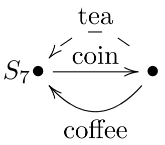




In this way, MTS are able to capture specifications of the systems. These systems are then refined (the behaviour is made more and more precise) until we get an LTS where all behaviour is completely known. In the simplest case, this refinement means that a may transition is either implemented (and becomes a normal transition) or is omitted (and disappears). Thus, we can see that both LTS S_1 and S_5 are refinements of S_2. However, note that in general S_2 is also refined by the more complicated S_6.







Furthermore, we might be given the seller's (S_3) and buyer's (S_4) specifications separately and ask whether there is a system that implements both requirements. In this example, both S_1 and S_5 do so. In addition, we can also say that S_7 is the most general MTS that captures both requirements together.

There is also a *disjunctive* extension of MTS, where we can specify that e.g. either coffee or tea has to be offferred, but we need not say which one.

Moreover, we want to check properties of MTS such as *the action coffee will eventually be executed* or *it always holds that whenever the action coin is executed, in the next step either tea or coffee is executed*. These properties are also expressible as logical formulae in some logics, e.g. LTL or CTL. Verifying that these formulae are satisfied in the system is called model checking.

Description of the task

The main aim of the thesis is to create a tool for (disjunctive) MTS and experimentally test it. More precisely, the following features should be included:

- simple but user-friendly environment for inputting MTS together with a simple web interface,
- implementation of algorithms deciding whether (1) an MTS refines another MTS, (2) a set of MTS have a common implementation and what is the most general MTS that refines all of them, (3) determinization of MTS,
- in the case of a Master thesis, model checking algorithms for LTL and CTL should also be implemented and all algorithms should be optimized so that they run in reasonable time despite the theoretically higher complexity of some of these,
- the implementation should allow future developers to easily extend or modify the functionality.

Furthermore, the implementation should be tested on some examples and experimental data (especially running time) should be evaluated.

Prerequisites

A basic knowledge of finite automata and operations such as their intersection and determinization is recommended. In the case of a Master thesis, a basic knowledge of model checking (on the level of either a model-checking course or automata course) is an advantage but not a necessity. The Master thesis version also offers research oriented questions in case of interest. For recent development and some of the algorithms, see e.g. [1, 3, 2].

References

[1] A. Antonik, M. Huth, K. G. Larsen, U. Nyman, and A. Wasowski. 20 years of modal and mixed specifications. *Bulletin of the EATCS no. 95*, pages 94–129, 2008.

[2] Nikola Benes, Ivana Cerna, and Jan Kretinsky. Disjunctive modal transition systems and generalized LTL model checking. Technical report FIMU-RS-2010-12, Faculty of Informatics, Masaryk University, Brno, 2010. http://www.fi.muni.cz/reports/files/2010/FIMU-RS-2010-12.pdf.

[3] Nikola Benes, Jan Kretinsky, Kim Guldstrand Larsen, and Jiri Srba. On determinism in modal transition systems. *Theor. Comput. Sci.*, 410(41):4026–4043, 2009.