A likelihood-ratio test for identifying probabilistic deterministic real-time automata from positive data[1]

Sicco Verwer[a] Mathijs de Weerdt [b] Cees Witteveen [b]

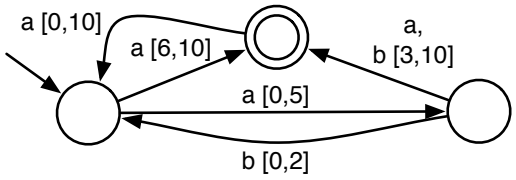
[a]*Eindhoven University of Technology* [b]*Delft University of Technology*

1 Real-time automata

Timed automata (TAs) are finite state models that represent timed events using an *explicit* notion of time, i.e., using numbers. They can be used to model and reason about real-time systems. In these system, each occurrence of a symbol (event) is associated with a time value, i.e., its time of occurrence. TAs can be used to accept or generate a sequence $(a_1, t_1)(a_2, t_2)(a_3, t_3) \dots (a_n, t_n)$ of symbols $a_i \in \Sigma$ paired with time values $t_i \in \mathbb{N}$, called a *timed string*. Every time value t_i in a timed string represents the time (delay) until the occurrence of symbol a_i since the occurrence of the previous symbol a_{i-1}.

In previous work [2], we described the RTI algorithm for identifying (learning) a subclass of TAs known as deterministic real-time automata (DRTAs) from labeled data, i.e., from an input sample $S = (S_+, S_-)$. The idea behind identification is that it is often easier to find examples of the behavior of a real-time system than to specify the system in a direct way. An identification algorithm then provides a way to find a TA model that characterizes the (behavior of the) real-time system that produced these examples.

The RTI algorithm is based on the currently best-performing algorithm for the identification of deterministic finite state automata (DFAs), called evidence-driven state-merging (ESDM) [1]. The only difference between DFAs and DRTAs are that DRTAs contain time constraints, as shown by the following example:



The above figure shows the transition graph of a DRTA. The start state is indicated by the sourceless arrow. The topmost state is an end state, indicated by the double circle. Every state transition contains both a label and a time constraint. The DRTA accepts and rejects timed strings not only based on their event symbols, but also based on their time values. For instance, it accepts $(a, 4)(b, 2)$ (state sequence: left $\rightarrow$ bottom $\rightarrow$ top) and $(a, 6)(a, 5)(a, 6)$ (left $\rightarrow$ top $\rightarrow$ left $\rightarrow$ top), and rejects $(a, 6)(b, 2)$ (left $\rightarrow$ top $\rightarrow$ reject) and $(a, 5)(a, 5)(a, 6)$ (left $\rightarrow$ bottom $\rightarrow$ top $\rightarrow$ left).

2 Identifying real-time automata

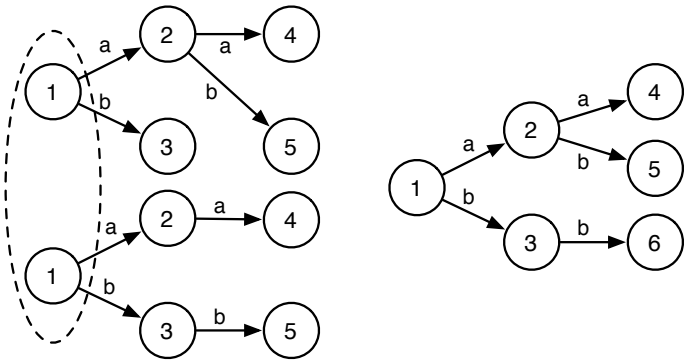
The RTI algorithm is efficient in both run-time and convergence. In practice, however, it can sometimes be difficult to apply RTI. The reason being that data can often only be obtained from actual observations of the process to be modeled. From such observations we only obtain timed strings that have actually

[1]This is an extended abstract of a paper to be published in the ICGI 2010 proceedings.

been generated by the system. In other words, we only have access to the positive data S_+, also known as *unlabeled data*.

In this paper, we adapt the RTI algorithm to this setting. A straightforward way to do this is to make the model probabilistic, and to check for consistency using statistics. To this aim, we introduce probabilistic DRTAs (PDRTAs) and a new likelihood-ratio test for this consistency check. A PDRTA is a DRTA with probability distributions that model the probability of observing a certain timed event (a, t) given the current state q of the PDRTA, i.e., $Pr(O = (a, t) \mid q)$.

The likelihood-ratio test is a common way to test nested hypotheses. A hypothesis H is called *nested* within another hypothesis H' if the possible distributions under H form a strict subset of the possible distributions under H'. Adding the likelihood-ratio test to a state-merging algorithm such as RTI is remarkably straightforward. Suppose that we want to test whether we should perform a merge (or split) of two states. Thus, we have to make a choice between two PDRTAs (models): the PDRTA $\mathcal{A}$ with two seperate states, and the PDRTA $\mathcal{A}'$ where these states are modeled as one. Clearly, $\mathcal{A}'$ is nested in $\mathcal{A}$. Thus all we need to do is to compute the maximized likelihood of S_+ under $\mathcal{A}$ and $\mathcal{A}'$, and apply the likelihood-ratio test:



The likelihood-ratio test tests whether using the left model (two prefix trees) instead of the right model (a single prefix tree) results in a significant increase in the likelihood of the data with respect to the number of additional parameters (states). The result of adding this to RTI is the RTI+ algorithm, which stands for real-time identification from positive data. The RTI+ algorithm is a polynomial time algorithm that converges efficiently.

3 Results

In order to evaluate the RTI+ algorithm, we test it on artificially generated data. First we generate a random PDRTA (without final states), and then we generate data using the distributions of this PDRTA. We performed such a test multiple times and using differently sized random PDRTAs using data sets of size 2000. The results of these tests are encouraging for up to 8 states, a size 4 alphabet, and 4 splits. When either of these values is increased, the algorithm needs more than 2000 examples to come up with a similar PDRTA. These results are encouraging because PDRTAs of this size are complex enough to model interesting real-time systems.

The likelihood-ratio test used by RTI+ is designed specifically for the purpose of identifying a PDRTA from unlabeled data. Although many algorithms like RTI+ exist for the problem of identifying (probabilistic) DFAs, none of these algorithms uses the non-timed version of the likelihood-ratio test of RTI+. Hence, since this test can easily be modified in order to identify (probabilistic) DFAs using for instance EDSM, it also contributes to the current state-of-the-art in DFA identification.

References

[1] Kevin J. Lang, Barak A. Pearlmutter, and Rodney A. Price. Results of the Abbadingo one DFA learning competition and a new evidence-driven state merging algorithm. In *Grammatical Inference*, volume 1433 of *LNCS*. Springer, 1998.

[2] Sicco Verwer, Mathijs de Weerdt, and Cees Witteveen. An algorithm for learning real-time automata. In *Proceedings of the Sixteenth Annual Machine Learning Conference of Belgium and the Netherlands*, pages 128–135, 2007.