

1. **Additive clock constraints** In the paper ‘Timed automata and additive clock constraints’, Berard et. al. have studied timed automata with additive clock constraints. It has been shown that emptiness checking is undecidable for timed automata with 4 clocks where clock constraints of the form  $x + y \sim c$  is allowed, where  $\sim \in \{<, \leq, =, \geq, >\}$ . The emptiness problem has been proved to be decidable for timed automata with 2 clocks. It has been shown in that paper that timed automata with additive clock constraints are strictly more expressive than the Alur-Dill timed automata that do not have the additive clock constraints. The undecidability proof involving 4 clocks uses encoding of two counter machine. The problem with 3 clocks is known to be open and want to investigate this problem.
2. **Bounded prebisimulation** Given a timed automata,  $Q$  denotes the set of timed automata states. A state is a tuple of a location and a valuation.

**Definition 1.** A bounded prebisimulation relation  $\preceq_\Delta \subseteq Q \times Q$  can be defined in the following way. For two timed automata states  $p$  and  $q$ ,  $p \preceq_\Delta q$  if the following conditions hold true.

- (i)  $\forall a \in Act, \forall p',$  if  $p \xrightarrow{a} p'$  then  $\exists q'$  such that  $q \xrightarrow{a} q'$  and  $p' \preceq_\Delta q'$ .
- (ii)  $\forall a \in Act, \forall p',$  if  $q \xrightarrow{a} q'$  then  $\exists p'$  such that  $p \xrightarrow{a} p'$  and  $p' \preceq_\Delta q'$ .
- (iii)  $\forall d \in \mathbb{R}_{\geq 0},$  if  $p \xrightarrow{d} p'$  then  $\exists q'$  and  $d' \in \mathbb{R}_{\geq 0}$  such that  $q \xrightarrow{d'} q'$  and  $d' - d \geq \Delta$  and  $p' \preceq_\Delta q'$ .
- (iv)  $\forall d \in \mathbb{R}_{\geq 0},$  if  $q \xrightarrow{d} q'$  then  $\exists q'$  and  $d' \in \mathbb{R}_{\geq 0}$   $p \xrightarrow{d'} p'$  and  $d - d' \geq \Delta$  such that  $p' \preceq_\Delta q'$ .

Here  $p$  is faster than  $q$  by  $\Delta$  or more. This problem can be checked by playing the corner point prebisimulation game between the challenger and the defender where for a delay move of the challenger, the defender if plays on the graph of  $q$ , chooses a delay that is at least  $\Delta$  greater than the delay chosen by the challenger. If the defender on the other hand chooses the  $p$  side, the delay made by it should be at least  $\Delta$  less than the delay made by the challenger.

3. **Weak prebisimulation** Considering  $\tau$  to be an internal action, we define a weak transition  $\Rightarrow$  as  $\xrightarrow{\tau^*} \xrightarrow{a} \xrightarrow{\tau^*}$ , where  $a \in Act$  is an action.

**Definition 2.** A weak prebisimulation relation  $\preceq_w \subseteq Q \times Q$  can be defined in the following way. For two timed automata states  $p$  and  $q$ ,  $p \preceq_w q$  if the following conditions hold true.

- (i)  $\forall a \in Act, \forall p',$  if  $p \xrightarrow{a} p'$  then  $\exists q'$  such that  $q \Rightarrow q'$  and  $p' \preceq_w q'$ .
- (ii)  $\forall a \in Act, \forall p',$  if  $q \xrightarrow{a} q'$  then  $\exists p'$  such that  $p \Rightarrow p'$  and  $p' \preceq_w q'$ .
- (iii)  $\forall d \in \mathbb{R}_{\geq 0},$  if  $p \xrightarrow{d} p'$  then  $\exists q'$  and  $d' \in \mathbb{R}_{\geq 0}$  such that  $q \xrightarrow{d'} q'$  and  $d' \geq d$  and  $p' \preceq_w q'$ .
- (iv)  $\forall d \in \mathbb{R}_{\geq 0},$  if  $q \xrightarrow{d} q'$  then  $\exists q'$  and  $d' \in \mathbb{R}_{\geq 0}$   $p \xrightarrow{d'} p'$  and  $d' \leq d$  such that  $p' \preceq_w q'$ .

Given two timed automaton states, we want to check how we can decide a weak timed performance prebisimulation relation between them.

4. **Synthesizing timed automaton preserving the timed language with least possible number of locations such that there is no increase in the number of clocks:** In the paper ‘Folk Theorems on the Determinization and Minimization of Timed Automata’, Tripakis has discussed minimizing the resources of a timed automaton while preserving the timed language. It has been shown that whether a timed automaton exists with lesser number of clocks or whether there exists a timed automaton in which the maximum constant used is less than the maximum constant used in the original automaton is undecidable. It has been discussed in the conclusion that the answer to the problem is not known while possibly it may be allowed to increase the number of locations of the timed automaton. However, since increasing the number of clocks increases the size of the region graph with an exponential factor and hence minimizing the number of locations without increasing the number of clocks is of practical importance.
5. **Faster Timed Automata Synthesis without preserving structure and preserving untimed language:** One can always create the region graph which is an NFA preserving the untimed language. One naive and simple approach to synthesize the fastest possible timed automaton is to remove the epsilon transitions by transforming the NFA to a DFA and on each edge add a transition with the clock constraint  $x = 0$ . In the synthesized automaton thus, there is a single clock  $x$  such that all the transitions take place at  $x = 0$ . This is the fastest possible timed automaton but may not preserve any non-Zeno behaviour if the original timed automaton is non-Zeno. Also the NFA to DFA conversion may give an exponential blow up to the number of locations of the timed automaton.  
We would also like to investigate how to synthesize a timed automaton that will preserve the non-Zeno behaviour if the original timed automaton is non-Zeno.
6. **Faster Timed Automata Synthesis while preserving structure and preserving untimed language:** In this problem, given a timed automaton, we want to synthesize the fastest possible timed automaton just by changing the clock constraints, the edges and the locations remaining the same such that the synthesized automaton accepts the same untimed language.

**Definition 3.** A zone  $z_2$  is a delay successor zone of zone  $z_1$  if  $z_2$  can be reached from  $z_1$  after elapsing some delay, i.e. in the zone graph, if  $z_1$  and  $z_2$  are associated to some location  $l$  of the corresponding timed automaton, then in the zone graph, we have  $(l, z_1) \xrightarrow{\varepsilon} (l, z_2)$ . Since  $\varepsilon$  is also reflexive, note that delay successor is also a reflexive relation.

Also  $z_2$  is an immediate delay successor zone of  $z_1$ , if  $z_2$  is not same as  $z_1$  and for all delay successor zones  $z_3$  of  $z_1$  such that  $z_3$  is not same as  $z_1$ ,  $z_3$  is also a delay successor zone of  $z_2$ .

One of the suggested methods is to create the zone graph for the timed automaton and then merging some of the nodes of the zone graph. We consider every pair of nodes such that for their corresponding zones, one is an immediate delay successor of another, i.e. their locations are the same and

check whether the two zones belong to an equivalence class in the sense of equivalence classes obtained through Myhill Nerode minimization.

We introduce an additional clock and the resultant timed automaton has a single clock. However, this synthesis procedure does not preserve the non-Zeno behaviour if the original automaton does not allow any Zeno behaviour.

**7. Substring of timed language:**