ntheorem

Table of Contents

Chapter 1

Definitions and Notations

1.1 Timed Automata

Timed automata are an extension of finite state automata. They allow us to integrate time into automaton models and thus enable modelling time-dependent systems.In this section we discuss about timed automaton and define its terminology.

1.1.1 Overview

Timed automaton is a state machine, equipped with special real valued variables called *clocks*. Clocks spontaneously increase their value with time. All clocks progress with the same rate. Locations (i.e. states) in automaton have invariants that are predicates over clocks. A location in an automaton can be active as long as its invariant is satisfied. Transitions in automaton have guards that are predicates over clocks. A transition can be taken only if its guard is true. Because clock values increase, an initially false guard can become true, allowing us to model time dependent behaviours. When a transition is taken, an associated action is executed, which can reset clocks to an integer value. In the general case, a transition from location l_1 to l_2 can be described as following

$$l_1 \xrightarrow{g,r} l_2 \tag{1.1}$$

where g represents the guard and r represents the reset operations performed when transition occurs.

Let's take a look at a sample timed automaton.

In the above timed automaton the initial state is L_0 with invariant $x \in [0, 1]$. The

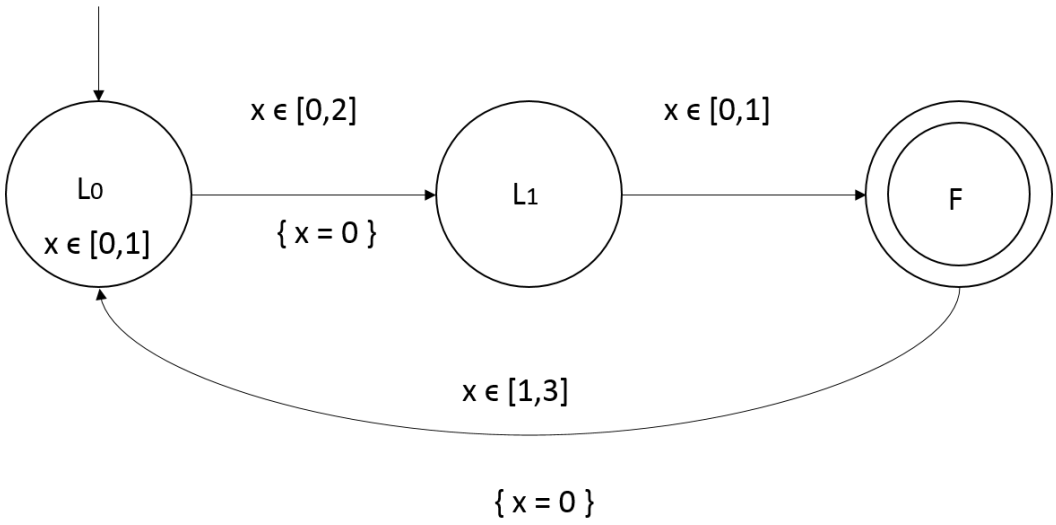
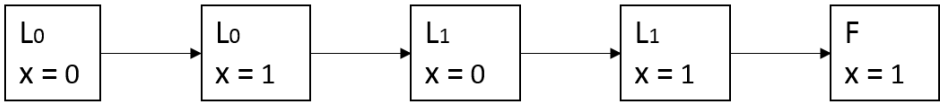


Figure 1.1: Timed Automaton with 3 states and 1 clock

transition $L_0 \rightarrow L_1$ has a guard over clock x, $x \epsilon [0, 2]$. This transition, if taken will reset the value of clock x to 0. A sample run of the timed automaton is,



1.1.2 Semantics of Timed Automata

To define the semantics of timed automata we define the following terms. Let C denote the set of all clocks. A clock valuation is a function $u : C \rightarrow \mathbb{R}^+$ which maps each clock to a real value. A simple valuation is the function $u_0 = 0 \; \forall \, x \, \epsilon \, C$. A clock valuation that satisfies that $\forall \; x \; \epsilon \; C, \; u(x) \; \epsilon \; \mathbb{N}_0$ is called an integer clock valuation u.

Guards are predicates over clocks and are of the form $x \;\; \bowtie \;\; n$ or $x - y \;\; \bowtie \;\; n$ where $n \; \epsilon \; \mathbb{N}$, $x, y \; \epsilon \; C$ and $\bowtie \;\; \epsilon <, \leq, >, \geq, =$. $B(C)$ denotes the set of all possible conjunctions over the clock variable conditions of C.

Definition 1. *A timed automaton A is a tuple (L, l_0, R, C, E, I) where L denotes*

the set of locations in the timed automaton, l_0 is the initial location, R is a set of reset operations, C a set of clocks and $E \subseteq L \times R \times B(C) \times L$ denotes the set of edges(between locations, with guard ($g \epsilon B(C)$) and a reset($r \epsilon R$), while I assigns invariants to locations.)

1.1.3 Region Graphs in Timed Automaton

If a timed automaton has n clocks, then any clock valuation can be represented as a point in n-dimensional euclidean space. Since in our representation guards over clocks are of the form $x < T$ or $x - y < T$ where x and y are clocks and T ϵ $\mathbb{N}$, there are regions in the n-dimensional euclidean space in which if one point satisfies some guard then all other points in the region also satisfy that guard. Consider the case $n = 2$, for which the region graph is,

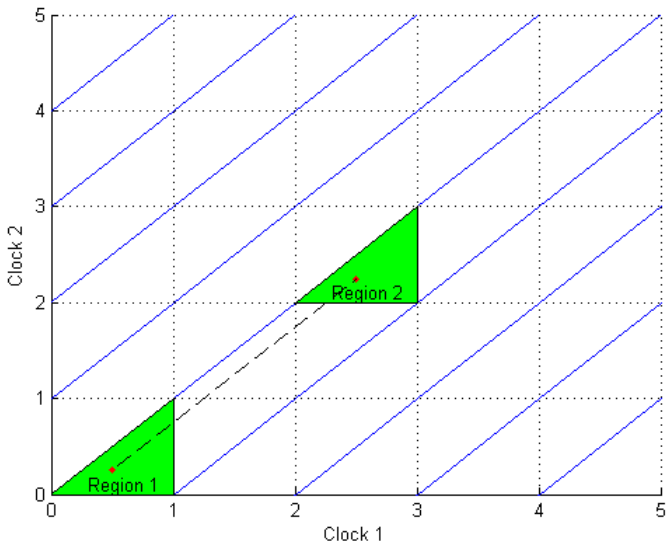


Figure 1.2: Region Graph for $n = 2$

In the above figure, shaded regions R_1(Region 1) and R_2(Region 2) represents two such regions.

For a clock valuation u_a in region R_a such that after waiting for some time, it becomes

u_b in region R_b, then all other points in region R_a can also, after waiting for some time reach region R_b. Further it can be observed that, a corner point (an integer valuation) in R_a after waiting for some integer time can reach the corresponding corner point (also an integer valuation) in R_b. These propositions hold true for all values of n.

1.2 Modelling in UPPAAL

Chapter 2

Results

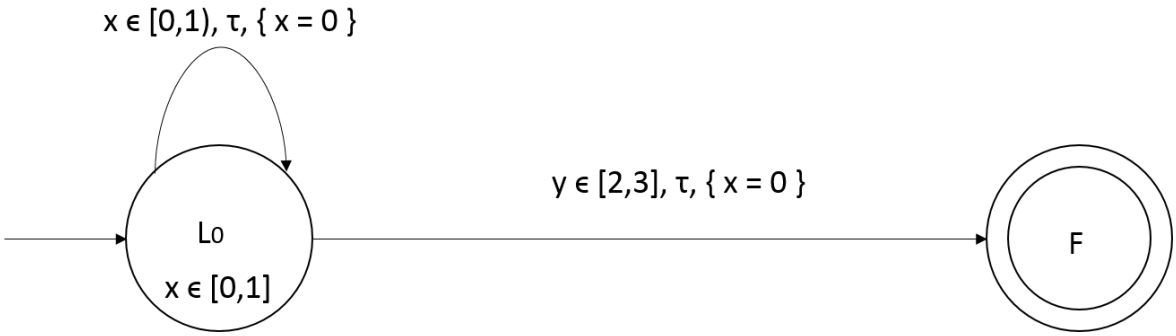
Theorem 1. *Let $A = \{A_1, A_2 ..., A_n\}$ be a network of n timed automata from the class LSC, with semantics (S, s_0,→). Consider $\bar{l}^a, \bar{l}^b, \bar{l}^c \epsilon (L_1 \times L_2 \times ... \times L_n)$ and valuations $w_a = (u_a, v_a)$, $w_b = (u_b, v_b)$, $w_c = (u_c, v_c)$ such that $(\bar{l}^a, w_a) \rightarrow (\bar{l}^b, w_b) \rightarrow (\bar{l}^c, w_c)$. If w_a is an integer evaluation then there exists integer valuations w'_b and w'_c such that*

$$(\bar{l}^a, w_a) \rightarrow (\bar{l}^b, w'_b) \rightarrow (\bar{l}^c, w'_c) \tag{2.1}$$

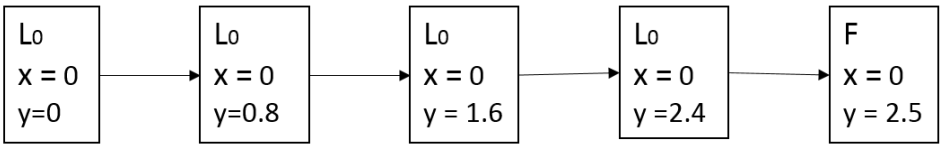
The above theorem is from [1] and its proof and semantics are same as in [1].

Proposition 1. *Theorem 2 is false.*

Proof. Consider the timed automaton,



Now, consider the following run on the above timed automaton,

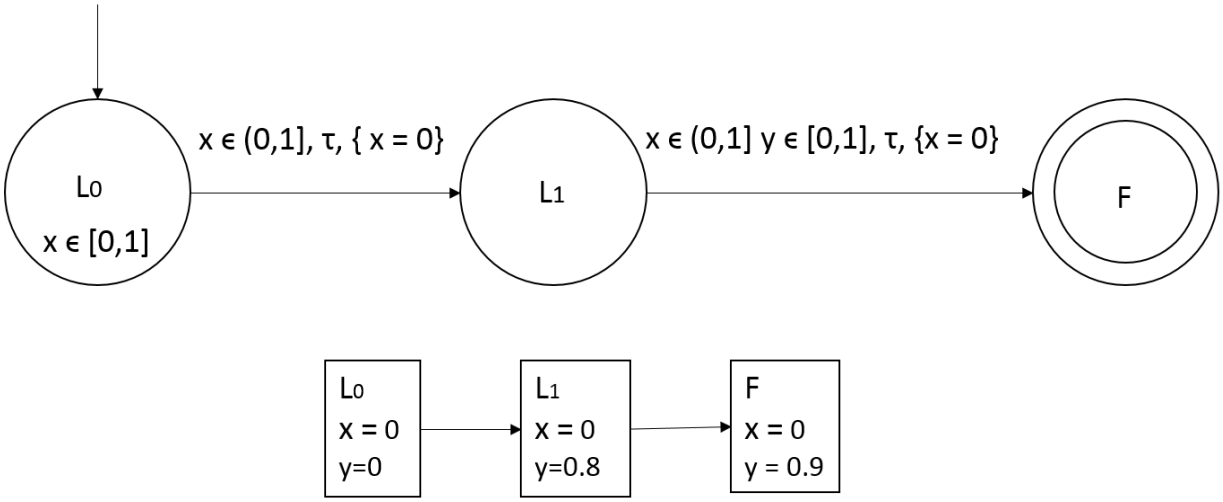


The timed automaton is from the class LSC and the initial clock valuation is also an integral valuation. However, there does not exist any sequence of integral valuations $w_1, w_2, ..., w_n$ such that $(\bar{L}_0, w_1) \to (\bar{L}_0, w_2) \to ... \to (F, w_n)$

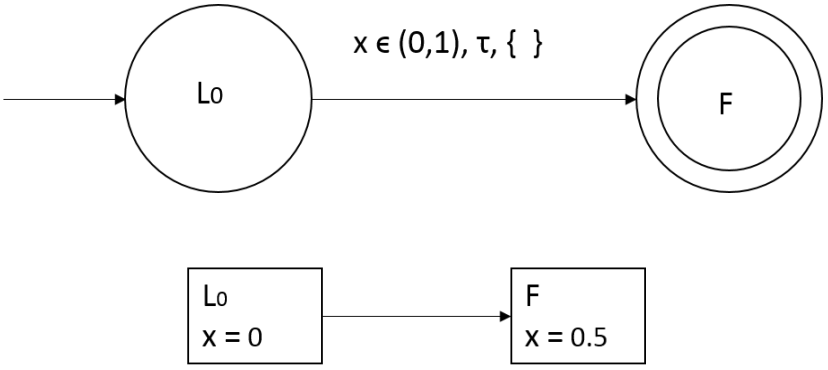
□

Remark 1. *Theorem* 1 *is false for automatons of the class RSC and Open Interval too.*

Proof. Consider the following automaton from the class RSC and the corresponding run,



Consider the following automaton from the class Open Interval and the corresponding run,



□

Theorem 2. *Let $A = \{A_1, A_2 ..., A_n\}$ be a network of n timed automata from the class Closed Interval, with semantics $(S, s_0, \rightarrow)$. Consider $\bar{l}^1, \bar{l}^2, ..., \bar{l}^n \epsilon (L_1 \times L_2 \times ... \times L_m)$ and valuations $w_1 = (u_1, v_1)$, $w_2 = (u_2, v_2)$,..., $w_3 = (u_3, v_n)$ such that there exists a run R, $(\bar{l}^1, w_1) \rightarrow (\bar{l}^2, w_2) \rightarrow ... \rightarrow (\bar{l}^n, w_n)$. If w_1 is an integer evaluation then there exists integer valuations w'_1, w'_2,..., w'_n such that*

$$(\bar{l}^1, w'_1) \rightarrow (\bar{l}^2, w'_2) \rightarrow ... \rightarrow (\bar{l}^n, w'_n) \tag{2.2}$$

Proof. We are going to prove this by using induction on the length of run R.

$w'_1 = w_1$. It is given that w_1 is an integer clock valuation.

Consider the k^{th} transition of R, $(\bar{l}^k, w_k) \rightarrow (\bar{l}^{k+1}, w_{k+1})$

Induction Step: There exists integer clock valuations $w'_1, ..., w'_k$ such that $(\bar{l}^1, w'_1) \rightarrow (\bar{l}^2, w'_2) \rightarrow ... \rightarrow (\bar{l}^k, w'_k)$. w'_k and w_k lie in the same region R_k.

Consider the transition, $(\bar{l}^k, w_k) \rightarrow (\bar{l}^{k+1}, w_{k+1})$. This can also be written as, $(\bar{l}^k, w_k) \rightarrow (\bar{l}^k, \bar{w}_k) \rightarrow (\bar{l}^{k+1}, w_{k+1})$. Here we wait for some time to reach from w_k to $\bar{w}_k$, $\bar{w}_k \epsilon\ g$ and $r(\bar{w}_k) = w_{k+1}$.

From propositions in section 1.1.3, it can be inferred that $\exists$ an integer valuation $\bar{w}'_k$ in region $\bar{R}_k$ such that after waiting for some time, we can reach $\bar{w}'_k$ from w'_k and both $\bar{w}'_k$ and $\bar{w}_k$ lie in the same region $\bar{R}_k$.Also w'_{k+1} is an integer valuation since reset operations update a clock to an integer value.

□

References

[1] Miroslav Pajic, Insup Lee, Rahul Mangharam, Oleg Sokolsky, UPP2SF: Translating UPPAAL Models to SimulinkJournal, Universitty of Pennsylvania, Tech. Rep., Oct 2011