Journal of Universal Computer Science, vol. 5, no. 9 (1999), 574-587
submitted: 10/6/99, accepted: 15/7/99, appeared: 28/9/99 © Springer Pub. Co.

Decomposition of Timed Automata

Kahn Mason
Engineering-Economic Systems and Operations Research
Stanford University, USA
kmason@leland.stanford.edu

Padmanabhan Krishnan
Department of Computer Science
University of Canterbury, New Zealand
paddy@cosc.canterbury.ac.nz

Abstract: In this paper we extend the notion of homomorphisms and decomposition to timed automata. This is based on the classical Hartmanis-Stearns decomposition results for finite state automata. As in the classical theory, the existence of non-trivial orthogonal partitions is both necessary and sufficient for non-trivial decompositions. Of course, now these partitions have to include both the set of states and the set of timers (or clocks) in the system. We present an example which illustrate the various issues.

Key Words: timed automata, parallel decomposition, state and clock partition
Category: F.1.1, J.7

1 Introduction

Abstraction is a technique used to overcome the state explosion problem in program verification. The paper [Clarke et al., 1994] illustrates the key issues. In the context of timed systems, approximation techniques are useful as shown in [Alur et al., 1995]. These approaches help one obtain a smaller system from a given system without violating the properties of interest. The aim of this paper is to consider decomposition techniques for timed systems. In the context of finite automata, the earliest work on a systematic approach to decomposition is presented in [Hartmanis and Stearns, 1966]. Given a finite automaton, one can obtain smaller systems by identifying (or merging) states. But not all reduced automata are desirable. When one merges the various states, it is important to preserve the original behaviour. Hence the reduced automaton must be consistent with the original automaton. This gives rise to the notion of admissible partitions and quotient automata. Not all combinations of smaller automata can be related to the original automaton. The theory shows that if one combines orthogonal subsystems, the combined system can be said to realise the original automaton. The notion of realisation is precisely captured by the existence of a suitable homomorphism.

The main aim of this paper is to extend the decomposition results for finite automata to timed automata initially presented in [Alur and Dill, 1994]. However rather than adopting a purely state based approach we have to take the effect of the clocks available to timed automata. We proceed with this in much the same fashion as for finite automata. That is, we specify the parallel composition used to combine automata, define a suitable notion of homomorphism and admissible decomposition. The main theorem of this paper identifies necessary and sufficient conditions for the existence of a parallel decomposition of a timed automaton.

This decomposition result forms the first step in understanding the structure of timed automata. It provides the basis for developing work on compositional techniques, abstractions and approximations. Theoretically such a technique could be used to model check state clock logic (SCL) formulae as described in [Raskin and Schobbens, 1997]. This is primarily because of the direct link between the temporal formulae and the clocks of the state clock automata.

In the next section we quickly review some preliminary material and establish the notation we use in the rest of the paper. This is followed by the main section of the paper where all the theoretical issues related to the decomposition of timed automata are presented. Section 4 illustrates the various ideas on two examples.

2 Preliminaries

We represent a finite automaton $\mathcal{M}$ as (Q, Σ, E, s, F) where Q is a finite set of states, Σ the finite input alphabet, $E \subseteq (Q \times Q \times \Sigma)$ the set of transitions, s the start state and F the set of final states. We use E instead of δ to avoid confusion as δ is used to present timing constraints in timed automata [Alur and Dill, 1994]. In certain contexts we write $\longrightarrow$ to denote the transition relation. When representing automata we let subscripts and superscripts to flow onto the components. For example, the states of an automaton $\mathcal{M}_1$ will be represented by Q_1 etc.

A timed automaton [Alur and Dill, 1994] is a finite automaton equipped with a set of timers (or clocks) C. Assume a set of timing constraints $\Phi(C)$ expressed over C as follows.

$$\beta ::= x \leq c \mid c \leq x \mid \neg\beta \mid \beta_1 \wedge \beta_2$$

where x is a timer in C and c a rational constant. Hence a timed automaton is a structure (Q, Σ, C, E, s, F) were Q is the finite set of states, Σ the input alphabet, C the set of available clocks or timers, E the transition relation is a subset of $(Q \times Q \times \Sigma \times 2^X \times \Phi(C))$, s the start state and F a Muller acceptance condition. An element of the transition relation of the form (q, q', a, X, β) indicates that the automaton can move from state q to state q' on the input a provided the timing constraint β is satisfied. When the move is made all the timers mentioned in X are reset to 0.

In order to define the behaviour of such an automaton, a notion of *extended state* is useful. Given a set of clocks C, a time valuation ν is a map from C to $\mathbb{R}^{\geq 0}$. Let V represent the set of all time valuations. We define $Q \times V$ to represent the set of extended states. That is, an extended state represents the state of the automaton as well as the values held in the various timers.

We let $[X \rightarrow 0]\nu$ to correspond to the valuation where all the clocks in X are set to 0 and the other clocks are unchanged. We also let $\nu + t$ correspond to the valuation where the value of all the clocks are incremented by t. We write $(q, \nu) \xrightarrow{a,t} (q', \nu')$ if there is an edge (q, q', a, X, β) such that $\nu + t$ satisfies β and $\nu' = [X \rightarrow 0](\nu + t)$. In other words, the a occurs t units of time after the automaton enters the state q. A real-time automaton is deterministic if for each extended state and input action and time there is only applicable transition.

The acceptance behaviour of timed automata is characterised by timed words. A timed word is a pair (α, τ) where α is an infinite sequence over Σ and τ an increasing non-Zeno sequence over the positive reals ($\mathbb{R}^+$). A run of a automaton over a timed word (α, τ) is a sequence of extended states σ where $\sigma(0) = (s, 0)$ and for each i greater than 0, $\sigma(i) \xrightarrow{a,t} \sigma(i+1)$ where $\alpha(i) = a$ and $\tau(i) - \tau(i-1) = t$. A run is an accepting run if the set of states that occur infinitely often in σ belongs to F (i.e., the usual Muller condition.) Next we present the definition of parallel composition of two timed automata as defined in [Alur and Dill, 1994].

Definition 1. Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two timed automata over identical alphabets. We define their *parallel composition*, $\mathcal{M}_1 \| \mathcal{M}_2$ to be the automaton $\mathcal{M}_\|$, where :

1. $Q_\| = (Q_1 \times Q_2)$, $\Sigma_\| = \Sigma_1 = \Sigma_2$, $s_\| = (s_1, s_2)$, $C_\| = C_1 \uplus C_2$ and $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2)$
2. $E_\| \subseteq (Q_\|, Q_\|, \Sigma_\|, 2^{C_\|}, \Phi(C_\|))$ is equal to $\{e_\| \mid \exists e_1 \in E_1, e_1 = (q_1, r_1, a, X_1, \delta_1) \in E_1, \exists e_2 \in E_2, e_2 = (q_2, r_2, a, X_2, \delta_2) \in E_2$, where $e_\| = ((q_1, q_2), (r_1, r_2), a, X_1 \uplus X_2, \delta_1 \wedge \delta_2)\}$

Here $\uplus$ denotes disjoint union. We can also assume that the set of clocks used in the two automata are disjoint in which case set union can be used. This definition makes it clear that the parallel composition used corresponds to language intersection. We let $(\mathcal{F}_1, \mathcal{F}_2)$ denote the appropriate Cartesian product over Muller acceptance criteria. This concludes our discussion of timed automata.

Now we present a quick overview of the classical decomposition theory. A finite automaton $\mathcal{M}_1$ is *homomorphic* to another automaton $\mathcal{M}_2$ if $\Sigma_1 = \Sigma_2$ and we can find a map $\phi \colon Q_1 \rightarrow Q_2$ such that if $(q, r, a) \in E_1$ then $(\phi(q), \phi(r), a) \in E_2$, $\phi(s_1) = s_2$ and $\phi[F_1] \subseteq F_2$ where $\phi[F_1]$ denotes the point-wise application of ϕ to elements of F_1. If ϕ is injective the homomorphism is an epimorphism. If $\mathcal{M}_1$ is epimorphic to $\mathcal{M}_2$, ϕ is said to represent a state behaviour assignment. That is, $\mathcal{M}_1$ is structurally contained in $\mathcal{M}_2$. As this containment must be related to the notion of implementation, the theory is used mainly in the context of deterministic automata.

Given a partition over a set S we let $\bot_S$ denote the *finest* partition viz., the set of singletons. We also let $\top_S$ denote the *coarsest* partition viz., the set containing S. Every other partition is *non-trivial*. Given an partition π and $x \in S$ we let $\pi|_x$ denote the equivalence class containing x.

Given an automaton $\mathcal{M}$, a partition π over Q is *admissible* if and only if for every X belonging to π and for every a in the input alphabet, there is a Y belonging to π such that $X \xrightarrow{a} Z$ and $Z \subseteq Y$. This means that for each input and each member of π the point-wise application of the transition relation to that member is contained within another member of π. Two partitions are *orthogonal* if for each pairwise intersection of the elements yields either the empty set or a singleton set.

Proposition 2 [Holcombe, 1982]. A given automaton $\mathcal{M}$ has two non-trivial orthogonal partitions each of which is admissible, iff it has a non-trivial decomposition into $(\mathcal{M}_1 \| \mathcal{M}_2)$.

This completes our review of the notation and concepts used in the rest of the paper.

3 Composition and Decomposition

In this section we present the formal description for the comparison of timed automata as well as the decomposition of a single automaton into two automata. This represents the main contribution of the paper.

3.1 Comparing Timed Automata

As for finite automata, we will define our homomorphisms as maps that correspond to some form of structural inclusion. One could try to base the definitions of homomorphisms, partitions etc. on the infinite state space of the whole system. However this has to be done with care. As the timing constraints in Alur-Dill automata involve only rational constants, not all partitions will yield an automata describable using rational constants. Hence partitions have to be based on some form of clock regions. A partition obtained from the clock regions may not be easily related to the original automaton. This is because one could construct timing constraints which are not related to the constraints present in the original automaton.

In our development, we will focus on the syntactical aspects of timed automata. Because the formal description of a timed automaton has clocks as well as states, both these will be mapped. Thus, we require maps for both the states and the clocks. Of course, such a map on clocks should be semantically meaningful. Although the states map forward, in order to get the extended states to correspond, the time states must be composed with the inverse of the clock maps. A homomorphism doesn't require that they be injective, but our concept of structural inclusion will impose this requirement. As for finite automata the starting states and accepting set of states must correspond. Also we will consider only time deterministic automata so that we get notion of implementation similar to that for finite automata. Thus the notion of "smaller" is measured by the number of states (not extended states) and the number of clocks used by the system.

Let the "smaller" automaton be in a particular extended state, say (q, ν) and be able to exhibit an a at time t and move to another extended state, (r, ξ). We want the larger automaton to be able to perform the "same" transition. What we mean by the "same" transition is that there will be an associated extended state in the larger automaton such that the extended state associated with (q, ν) can perform a transition to the extended state associated with (r, ξ). Thus we need to introduce a way of associating extended states of the smaller automaton with those of the larger automaton.

The choice of which state in the larger automaton to associate with a given state in the smaller automaton is very natural. It is the same as with untimed automaton. If the state in the smaller automaton is q and the state map is ϕ then the state in the larger automaton that corresponds to q is its image under ϕ, that is $\phi(q)$ is associated with q.

$$\begin{tikzcd}
C^1 \arrow[r, "\nu^1"] \arrow[d, "\eta"'] & \mathbb{R}^{\geq 0} \\
C^2 \arrow[ur, "\nu^2"'] &
\end{tikzcd}$$

Figure 1: Time-state correspondences.

The time states are not as easy, because each time state is itself a function. If the clock map is η and the two time states are ν^1 in the smaller automaton and ν^2 (which we desire to find) in the larger automaton, then we have Figure 1.

The natural correspondence (see Figure 1) commutes. That is, choose ν^2 so that $\nu^2 \circ \eta = \nu^1$. This means that $\nu^2 = \nu^1 \circ \eta^{-1}$ over any domain where η^{-1} is a function. When η^{-1} is not a function then, in order to keep the concept of inclusion ν^2 will have to satisfy any clock constraints that ν^1 could. To enable this we let $\nu^1 \circ \eta^{-1}$ only be defined over the largest domain where η^{-1} is a function. That is, for all other points, x, $\nu^1 \circ \eta^{-1}(x) = *$ where $*$ is a don't care value that satisfies any clock constraint. This does not affect any results as structural inclusion will require that η be injective. Putting all these observations together gives us the following definition.

Definition 3. Given two timed automata, $\mathcal{M}_1$ and $\mathcal{M}_2$, a *homomorphism* between $\mathcal{M}_1$ and $\mathcal{M}_2$ is an ordered pair (ϕ, η) where $\phi : Q_1 \rightarrow Q_2$ and $\eta : C_1 \rightarrow C_2$ are such that:

1. $\phi(s_1) = s_2$ and $\phi[\mathcal{F}] \subseteq \mathcal{F}_2$.
2. If $\quad (q, \nu) \xrightarrow{a,t} (r, \xi) \quad$ then $\quad (\phi(q), \nu \circ \eta^{-1}) \xrightarrow{a,t} (\phi(r), \xi \circ \eta^{-1})$

We write $(\phi, \eta) : \mathcal{M}_1 \rightsquigarrow \mathcal{M}_2$ if the above holds. If both ϕ and η are total and injective then we say the homomorphism is an *monomorphism*. If both ϕ and η are onto we say the homomorphism is an *epimorphism*.

Note that ϕ and η are maps over finite sets and hence the pair (ϕ, η) is a structural map. But the second requirement is over the extended state. Based on this definition the following two results can be easily proven.

Lemma 4. *If $\mathcal{M}_1$ is homomorphic to $\mathcal{M}_2$ then the language accepted by $\mathcal{M}_1$ is contained in the language accepted by $\mathcal{M}_2$.*

Proof: The proof follows directly from the definitions. One has to translate one accepting run into another accepting run. □

Informally, by a component-wise combination of the two morphisms, an appropriate morphism to the parallel composition can be exhibited. The result is useful when a system is decomposed into more than two subcomponents.

Lemma 5. *If $\mathcal{M}$ is monomorphic to both $\mathcal{M}_1$ and $\mathcal{M}_2$ then it is monomorphic to $\mathcal{M}_1 \| \mathcal{M}_2$.*

Proof: Let $\mathcal{M}_{\|} = \mathcal{M}_1 \| \mathcal{M}_2$. For each $i \in \{1, 2\}$, let $(\phi_i, \eta_i) : \mathcal{M} \rightsquigarrow \mathcal{M}_1$. Define $\phi_{\|}$ to be such that $\phi_{\|}(x, y) = (\phi_1(x), \phi_2(y))$ and $\eta_{\|} = (\eta_1 \uplus \eta_2)$. Based on this the following observations follow.

1. $\phi_{\|}(s) = (s_1, s_2) = s_{\|}$ and $\phi_{\|}[\mathcal{F}] = (\phi_1(\mathcal{F}), \phi_2(\mathcal{F})) \subseteq (\mathcal{F}_1, \mathcal{F}_2) = \mathcal{F}_*$.
2. If $(q, \nu) \xrightarrow{a,t} (r, \xi)$ by the definition of monomorphism it is the case that for each $i \in \{1, 2\}$, $(\phi_i(q), \nu \circ \eta_i^{-1}) \xrightarrow{a,t} (\phi_i(r), \xi \circ \eta_i^{-1})$. Therefore, from the definition of parallel composition the transition $(q_*, \nu_*) \xrightarrow{a,t} (r\ast_{,}\, \xi_*)$ is possible where
   $q_* = (\phi_1(q), \phi_2(q))$, $\nu_* = \nu_1 \circ \eta^{-1} \uplus \nu_2 \circ \eta^{-1}$, $r_* = (\phi_1(r), \phi_2(r))$ and $\xi_* = \xi \circ \eta_1^{-1} \uplus \nu \circ \eta_2^{-1}$.

   This implies that $(\phi_{\|}(q), \nu \circ \eta_{\|}^{-1}) \xrightarrow{a,t} (\phi_{\|}(r), \xi \circ \eta_{\|}^{-1})$.

Together the above imply that $\mathcal{M}$ is homomorphic to $\mathcal{M}_1 \| \mathcal{M}_2$. Note that ϕ_i, η_i being injective for each $i = 1, 2$ implies that $\phi_{\|}, \eta_{\|}$ are both injective, thus ensuring that the homomorphism is a monomorphism. □

This completes the formal definition concerning the comparisons of timed automata.

3.2 Decomposition

Based on the above definition of structure relationship, we define the desired goal of a decomposition precisely.

Definition 6. A timed automaton, $\mathcal{M}$, has a *parallel decomposition* if there exist two timed automata, $\mathcal{M}_1$ and $\mathcal{M}_2$, such that $|Q_1|, |Q_2| < |Q|$ and there is a *monomorphism* $(\phi, \eta) : \mathcal{M} \rightsquigarrow \mathcal{M}_1 \| \mathcal{M}_2$.

This definition makes it clear that we are not interested in arbitrary homomorphisms. We are interested in mono-morphisms or homomorphisms which maintain structural containment. Hence all future references to parallel decomposition assumes the existence of a pair of monomorphic maps.

The requirements on the state spaces are to ensure that the decomposition yields “smaller” automata. But this is only on the “untimed” states. The following is a picture showing the structural relationship. The wavy arrow represents a monomorphism and the line-double arrow represents the part-whole relationship of a parallel composition. The dotted arrows describe the relationships that we will investigate to determine the requirements on $\mathcal{M}$ in order for it to have a parallel decomposition.

As we will be using a partition based approach, the following two concepts are useful. The first concept is as for finite automata. That is, for each state, q_1, of $\mathcal{M}_1$, there will correspond a member of π_1 that contains all the states of $\mathcal{M}$ that map (under ϕ) to states of $\mathcal{M}_{\|}$ where the first component is in

$$\begin{tikzcd}
& \mathcal{M} \arrow[d, rightsquigarrow] \arrow[ddl, dotted] \arrow[ddr, dotted] & \\
& \mathcal{M}_{\|} & \\
\mathcal{M}_1 \arrow[rr, dash] & {} \arrow[u, Rightarrow] & \mathcal{M}_2
\end{tikzcd}$$

Figure 2: The relationship between $\mathcal{M}$ and $\mathcal{M}_{\|}$.

state q_1. In other words, all information, be it state or clock information, in the parallel automaton that comes from the second component is ignored. The second concept concerns the set of clocks. That is, there is a natural partition of the clocks of $\mathcal{M}_{\|}$ corresponding to the clocks that are mapped into each component. Note that the term partition is justified by η being total and $C_{\|}$ being $C_1 \uplus C_2$.

Definition 7. If $\mathcal{M}$ has a parallel decomposition and $(\phi, \eta) : \mathcal{M} \rightsquigarrow \mathcal{M}_{\|}$ then the *induced state partitions* of $\mathcal{M}$ are given by

$\pi_1 = \{\phi^{-1}[(q_1, Q_2)] | q_1 \in Q_1\}$ and
$\pi_2 = \{\phi^{-1}[(Q_1, q_2)] | q_2 \in Q_2\}$.

The *induced clock partition* of $\mathcal{M}$ is given by $\{\eta^{-1}[C_1], \eta^{-1}[C_2]\}$.

The injectivity of the state map forces the induced state partitions to be orthogonal. This is because each state in $\mathcal{M}$ corresponds to precisely one state in $\mathcal{M}_{\|}$ which in turn corresponds to precisely one state in each of the component automata.

The intuitions behind the definition are as follows. A member of the first state partition consists of states in the original automaton that have the same first component when relabelled by ϕ, and similarly for the second state partition. Because the relabelling was one-to-one, the pair of members—one from each induced state partition—defined by a given state in the original automaton is unique, and each state is contained in both members, so that the partitions are orthogonal.

Lemma 8. *The induced state partitions, given in Definition 7, are orthogonal.*

Proof: Note that ϕ is one-to-one and that its inverse is a partial function. Therefore,

$$\begin{aligned}
\pi_1.\pi_2 &= \{U \cap V | U \in \pi_1, V \in \pi_2\} \\
&= \{\phi^{-1}[(q_1, Q_2)] \cap \phi^{-1}[(Q_1, q_2)] | q_1 \in Q_1, q_2 \in Q_2\} && \textit{Definition } 7 \\
&= \{\phi^{-1}[\{(q_1, q_2)\}] | (q_1, q_2) \in Q_{\|}\} \\
&= \{\phi^{-1}[\{q_{\|}\}] | q_{\|} \in Q_{\|}\} && \textit{Definition } 1 \\
&= \bot_Q && \textit{because } \phi \textit{ is total}
\end{aligned}$$

□

It is easy to show that the induced state partitions are non-trivial. If either of the state partitions were trivial then one of them would have to be $\perp_Q$ contradicting the assumption that $|Q_1|, |Q_2| < |Q|$.

We have presented a few properties that will be satisfied if one is given a parallel decomposition of a given automata. However, we have not still addressed how one can arrive at this from the original automaton. It is this process that requires state and clock partitions. The next section discusses this issue.

3.3 Admissible Partitions

As we are interested in deterministic behaviour, we need to impose certain requirements that force the appropriate partitions to induce a well-defined transition function. In creating the component automaton join states together in an arbitrary manner does not guarantee determinacy. If some extended states are to be joined, time determinacy requires for each symbol a and time step t, the extended states that could be reached by a transition on an a taking time t must also be joined. Hence the notion of admissibility needs to take into account the various clocks.

Definition 9. Given a timed automaton $\mathcal{M}$, a partition $\pi \in \Pi(Q)$, and a set of clocks, $S \subseteq C$, we say that (π, S) is an *admissible pair* for $\mathcal{M}$ if the following condition is satisfied. For every pair of edges $e, e' \in E$ of the form $e = (q, r, a, X, \delta)$ and $e' = (q', r', a, X', \delta')$, if $\pi|_q = \pi|_{q'}$ and $(\delta \wedge \delta')|_S$ is satisfiable over $(\mathbb{R}^+)^S$ then $\pi|_r = \pi|_{r'}$ and $X \cap S = X' \cap S$.

Note that this definition is based on the structure of the automaton. As determinacy is actually based on the extended states, the following result is useful.

Lemma 10. *Given a timed automaton $\mathcal{M}_*$, a partition $\pi \in \Pi(Q_*)$, and a set of clocks, $S \subseteq C_*$, (π, S) is an* admissible pair *for $\mathcal{M}_*$ iff the following condition is satisfied. For every $q, q' \in Q_*$ and $\nu, \nu' \in (\mathbb{R}^+)^{C_*}$, if $(q, \nu) \xrightarrow{a,t} (r, \xi)$ and $(q', \nu') \xrightarrow{a,t} (r', \xi')$ with $\pi|_q = \pi|_{q'}$ and $\nu|_S = \nu'|_S$ then $\pi|_r = \pi|_{r'}$ and $\xi|_S = \xi'|_S$.*

Proof: We first demonstrate that admissibility implies the condition in the hypothesis, before demonstrating the converse. If (π, S) is an admissible pair for $\mathcal{M}_*$ then for every $q, q' \in Q_*$ and $\nu, \nu' \in (\mathbb{R}^+)^{C_*}$ the following holds. If $(q, \nu) \xrightarrow{a,t} (r, \xi)$ and $(q', \nu') \xrightarrow{a,t} (r', \xi')$ with $\pi|_q = \pi|_{q'}$ and $\nu|_S = \nu'|_S$ then by the definition of transition, there must be edges of the form $e = (q, r, a, X, \delta)$ and $e' = (q', r', a, X', \delta')$ in E so that
$\nu + t \models \delta$, $\nu' + t \models \delta'$ with $\xi = [X \to 0](\nu + t)$ and $\xi' = [X' \to 0](\nu' + t)$.

Now because $\nu \models \delta$, it follows that $\nu|_S \models \delta|_S$. Similarly $\nu'|_S \models \delta'|_S$, so that $\delta|_S \wedge \delta'|_S$ is satisfiable, and hence $(\delta \wedge \delta')|_S$ is satisfiable over $(\mathbb{R}^+)^S$.

Admissibility now requires that $\pi|_r = \pi|_{r'}$ and $X \cap S = X' \cap S$. Because $X \cap S = X' \cap S$ and $\nu|_S = \nu'|_S$ we know the following.

$\xi|_S = ([X \to 0](\nu + t))|_S = [X \cap S \to 0](\nu + t)|_S = [X \cap S \to 0](\nu|_S + t)$

By substituting X' for X and ν' for ν we get $\xi|_S = \xi'|_S$. This together with $\pi|_r = \pi|_{r'}$ demonstrates the equivalence in one direction.

For the converse, take an arbitrary pair of edges $e, e' \in E$ of the form $e = (q, r, a, X, \delta), e' = (q', r', a, X', \delta')$ where $\pi|_q = \pi|_{q'}$ and $(\delta \wedge \delta')|_S$ is satisfiable. We essentially have to prove only that $X \cap S = X' \cap S$.

Because $(\delta \wedge \delta')|_S$ is satisfiable over $(\mathbb{R}^+)^S$, there is some $\nu_S \in (\mathbb{R}^+)^S$ so that $\nu_S \models (\delta \wedge \delta')|_S = \delta|_S \wedge \delta'|_S$. Furthermore, as $\xi|_S = \xi'|_S$ and $\nu|_s = \nu'|_S$, it is easy to construct a ν such that $\nu|_S = \nu_S$. The new clock valuation can now be use that $\xi|_S = ([X \rightarrow 0]\nu)|_S = [X \cap S \rightarrow 0]\nu_S$. Similarly, $\xi'|_S = ([X' \rightarrow 0]\nu)|_S = [X' \cap S \rightarrow 0]\nu_S$.

This then completes the proof. □

Another way of viewing this description of admissible pairs is as follows. Given a member of the partition, U, and an action and a time step a and t, any two transitions from a member of U on a taking t units of time must both go into the same member of π. This holds provided the time states have the same values for members of S. This means we can find admissible partitions as a recursive fixed point because the system (as defined by the number of states and available clocks) is finite. Note that this construction is concerned only with satisfiability of timing constraints over the set of clocks S. Such a clock valuation may or may not satisfy the original timing constraint.

The reason we define admissible pairs is that in order to generate the component automata from $\mathcal{M}$ we will need to group some states of $\mathcal{M}$ in precisely this sort of a consistent manner, as the following result shows. This result now can be viewed as a concrete realisation related to Definition 7.

Lemma 11. *If $\mathcal{M}$ has a parallel decomposition with induced state partitions π_1 and π_2 and induced clock partition (S_1, S_2) then (π_1, S_1) and (π_2, S_2) are admissible where the subscripts refer to the component which generated the associated partition on $\mathcal{M}$.*

Proof: Let the decomposition be realised by (ϕ, η) which maps $\mathcal{M}$ to $\mathcal{M}_1 \| \mathcal{M}_2$. As (ϕ, η) is a homomorphism, every extended state and hence every transition in $\mathcal{M}$ can be reflected to an appropriate state and transition in $\mathcal{M}_1 \| \mathcal{M}_2$.

That is, for an arbitrary pair of transitions T, T' of $\mathcal{M}$ whose source time states are identical over S_1 and whose source states come from the same member of π_1, that is the first component of their images under ϕ are identical. The definition of homomorphism gives corresponding transitions in $\mathcal{M}_1 \| \mathcal{M}_2$, with the images of the states under ϕ and the corresponding time states (Figure 1).

Consider the transitions on $\mathcal{M}_1$ (the other case is similar). The requirements on T and T' ensure that the source extended states of the two (one for each of T and T') corresponding transitions on $\mathcal{M}_1$ are identical. Time determinacy of $\mathcal{M}_1$ now forces the target extended states to also be identical. Thus the target states of T and T' are in the same member of π_1 and the target time states are identical over $S_1 = \eta^{-1}[C_1]$. □

3.4 Quotient Machines

Having defined admissible partitions, we can now use the existence of orthogonal partitions to complete the decomposition theory. We first define the notion of a quotient automaton induced by a partition. The properties of the quotient automaton then automatically lead to the desired theorem.

Definition 12. If we have a timed automaton $\mathcal{M}$ and π is a partition of Q, with $S \subseteq C$, then we define the *quotient automaton* of $\mathcal{M}$ with (π, S) to be the automaton $\mathcal{M}_\pi$ where :

$Q_\pi = \pi$, $\varSigma_\pi = \varSigma$, $s_\pi = \pi|_s$, $C_\pi = S$ and $\mathcal{F}_\pi = \pi|_{\mathcal{F}}$.
E_π is defined so that $e_\pi \in E_\pi$ iff $e_\pi = (\pi|_q, \pi|_r, a, X \cap S, \delta_\pi)$ for some $e = (q, r, a, X, \delta) \in E$, where δ_π is the minimal fixed point of the recursion

$$\delta_\pi = \delta|_S \vee (\bigvee\{\delta'|_S \mid (q', r', a, X', \delta') \in E, \ \pi|_{q'} = \pi|_q \text{ and } \delta'|_{\mathrm{S}} \wedge \delta_\pi \text{ is satisfiable } \} \ )$$

The starting point for the recursion is $\delta_\pi = F$. Minimality here is with respect to the partial order generated by implication, $\Rightarrow$.

The automaton $\mathcal{M}_\pi$ is denoted by $M/(\pi, S)$

The process of grouping together states and ignoring clocks, then considering the induced edge structure led to the concept of admissibility. Determinacy in the edge structure thus generated requires admissibility. This inducing of edge structures is encapsulated by the forming of quotient automata, and so deterministic quotient automata will require admissible pairs. What is not as transparent is the fact that the argument used to show the deterministic quotient automata require admissible pairs is reversible, so that admissible pairs always generate deterministic quotient automata.

Lemma 13. *$\mathcal{M}/(\pi, S)$ is deterministic iff (π, S) is admissible.*

Proof: Consider two arbitrary edges of $\mathcal{M}_\pi = \mathcal{M}/(\pi, S)$, $e_\pi = (q_\pi, r_\pi, a, X_\pi, \delta_\pi)$ and $e'_\pi = (q'_\pi, r'_\pi, a, X'_\pi, \delta'_\pi)$, with the same action and $q_\pi = q'_\pi$.

Determinism of $\mathcal{M}_\pi$ requires that if there is some time state, ν_π, so that $\nu_\pi \models \delta_\pi$ and $\nu_\pi \models \delta'_\pi$ then $e_\pi = e'_\pi$ because both edges can be traversed from (q_π, ν_π). This means that if $\delta_\pi \wedge \delta'_\pi$ is satisfiable then $e_\pi = e'_\pi$.

Now, e_π and e'_π, by virtue of their being edges of a quotient automaton, are of the form $e_\pi = (\pi|_q, \pi|_r, a, X \cap S, \delta|_\pi)$ and $e'_\pi = (\pi|_{q'}, \pi|_{r'}, a, X' \cap S, \delta|'_\pi)$ for some $(q, r, a, X, \delta), (q', r', a, X', \delta') \in E$.

Note that $\pi|_q = q_\pi = q'_\pi = \pi|_{q'}$ by the choice of edges above and $(\delta \wedge \delta')|_S = \delta|_S \wedge \delta'|_S = \delta_\pi \wedge \delta'_\pi$. Determinism thus requires that if $(\delta \wedge \delta')|_S$ is satisfiable then $e_\pi = e'_\pi$ which means that both $r_\pi = r'_\pi$ and $X_\pi = X'_\pi$, or alternatively both $\pi|_r = \pi|_{r'}$ and $X \cap S = X' \cap S$ which is the requirement for (π, S) to be admissible.

For the converse, if $\mathcal{M}_\pi$ is not deterministic then there is some extended state of $\mathcal{M}_\pi$ which can make two distinct transitions, that is satisfy the requirements of constraints from two distinct edges. Writing everything in terms of the original automaton as above, this means it can't be the case that both $\pi|_r = \pi|_{r'}$ and $X \cap S = X' \cap S$. This means that (π, S) is not admissible as required. □

Once the quotient automaton has been generated there is a natural correspondence between the original automaton and its quotient. That is, map each state of the original automaton to the state of the quotient automaton which contains it. This is a well defined map since the states of the quotient automaton are disjoint sets of states from the original automaton. Similarly, since the clock set of the quotient automaton is contained in the clock set of the original automaton, there is a natural partial map between the two, that is, the identity restricted to the clocks of the quotient automaton. Because the quotient automaton preserves the edges of the original automaton, the maps are onto maps. This result, when combined with Lemma 5 provides sufficient conditions to match the necessary ones on the induced state and clock partitions.

Lemma 14. *For any quotient automaton $\mathcal{M}/(\pi, S)$, the original automaton $\mathcal{M}$ is epimorphic to the quotient automaton $\mathcal{M}/(\pi, S)$.*

Proof: Let $\mathcal{M}_\pi = \mathcal{M}/(\pi, S)$. Define $\phi \in Q_\pi^Q$ by $\phi(q) = \pi|_q$, and $\eta : C \to C_\pi$ to be $\{(c, c) | c \in C_\pi\}$. That is, $\phi(q)$ is a total function identifies the set in the partition containing q and η is a partial map that identifies only the relevant clocks.

The above definition implies the following.

1. $\phi(s) = \pi|_s = s_\pi$, from Definition 12.
2. If $(q, \nu) \xrightarrow{a,t} (r, \xi)$ in $\mathcal{M}$ then we know $(q, r, a, X, \delta) \in E$ for some $(X, \delta) \in (2^C, \Phi(C))$ where $\nu + t \models \delta$ and $\xi = [X \to 0](\nu + t)$. Thus, $(\pi|_q, \pi|_r, a, X \cap X, \delta_\pi) \in E_\pi$ where $\delta|_S \Rightarrow \delta_\pi$

   Now $\nu \circ \eta^{-1} + t = \nu|_S + t = (\nu + t)|_S \models \delta|_S$, so that $(\pi_q, \nu|_S) \xrightarrow{a,t} (\pi_r, \xi')$ in $\mathcal{M}_\pi$ where $\xi' = [X \cap S \to 0](\nu|_S + t) = ([X \to 0](\nu + t))|_S = \xi|_S$ Thus, from Definition 12 $(\phi(q), \nu \circ \eta^{-1}) \xrightarrow{a,t} (\phi(r), \xi \circ \eta^{-1})$ in $\mathcal{M}_\pi$.
3. $\phi[\mathcal{F}] = \pi|_\mathcal{F} = \mathcal{F}_\pi$, from Definition 12.

Thus (ϕ, η) is a homomorphism. Now, because no member of π is empty, ϕ is onto. Similarly, since $S \subset X$, η is onto, and thus (ϕ, η) is a monomorphism as required. □

The above result (which related the original automaton and one of the quotients) can now be used to obtain the desired theorem. The theorem of interest takes the parallel composition of two quotient automata.

Theorem 15. *A timed automaton $\mathcal{M}$ has a parallel decomposition (i.e., a suitable monomorphism) iff it has two orthogonal non-trivial state partitions, π_1 and π_2, and there is a clock partition, $\{C_1, C_2\}$, so that both (π_1, C_1) and (π_2, C_2) are admissible.*

Proof: If $\mathcal{M}$ has a parallel decomposition, then let π_1 and π_2 be the induced state partitions defined in Definition 7, and let C_1 and C_2 be defined so that $\{C_1, C_2\}$ is the induced state partition defined in Definition 7.

From the definitions, π_1 and π_2 are orthogonal and non-trivial, and by Lemma 11 both (π_1, C_1) and (π_2, C_2) are admissible. Thus, constructively, $\mathcal{M}$ has two orthogonal non-trivial state partitions, π_1 and π_2, and there is a clock partition, $\{C_1, C_2\}$, so that both (π_1, C_1) and (π_2, C_2) are admissible.

For the converse, suppose that $\mathcal{M}$ has two orthogonal non-trivial state partitions, π_1 and π_2, and there is a clock partition, $\{C_1, C_2\}$, so that both (π_1, C_1) and (π_2, C_2) are admissible. Let $\mathcal{M}_1 = \mathcal{M}/(\pi_1, C_1)$ and $\mathcal{M}_2 = \mathcal{M}/(\pi_2, C_2)$. Now, from Lemma 13, both $\mathcal{M}_1$ and $\mathcal{M}_2$ are deterministic. Also, from Lemma 14, $\mathcal{M}$

is epimorphic to both $\mathcal{M}_1$ and $\mathcal{M}_2$. In particular, it is homomorphic to the two quotient automata, which from Lemma 5 means that it is homomorphic to their parallel composition. The orthogonality of π_1 and π_2 ensures that the state map is injective, so that $\mathcal{M}$ is monomorphic to the parallel composition of $\mathcal{M}_1$ and $\mathcal{M}_2$. □

4 Example

We will present an example to demonstrate the application of the above results. Consider the automata shown in Figure 3. They describes the behaviour of a buffer of size two. The buffer can be reset after processing one or two inputs, but the buffer can only perform one push (or pop) per unit of time. The actions intuitive meanings are as follows; "*a*" represents a push into the buffer, "*b*" represents a pop from the buffer, and "*c*" represents resetting the buffer. In BUFF successive "*a*"'s (or "*b*"'s) must be at least one time unit apart. To keep the presentation simple we accept all cycles. States 4 and 7 represent underflow and overflow states respectively and there are not transitions out of that state. In order to find appropriate admissible pairs it is necessary to partition the state set twice (once for each admissible pair) and the clock set once (into two sets, one for each admissible pair).

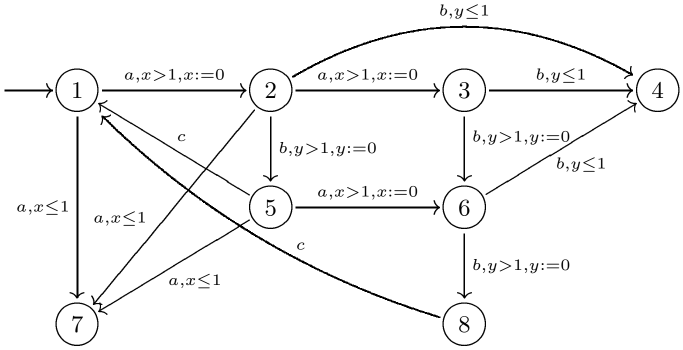
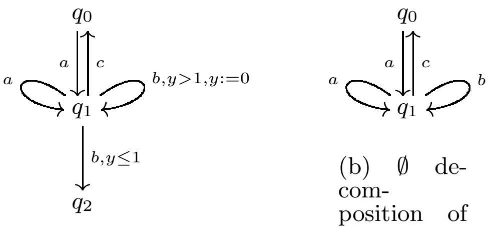


Figure 3: BUFF

Looking at just the clock set, we note that one of the two admissible pairs will not contain the clock y. Now because there is a transition on a b out of state 2 in to each of states 4 and 5, and each of the constraints restricted to $\{x\}$ is $True$, then states 4 and 5 must be grouped together. In a similar manner states 4 and 6 must be grouped together because of the b transitions out of state 3. Similarly states 4 and 8 must be grouped together because of the b transitions out of state 6. Thus states 4, 5, 6 and 8 must all be grouped together. Denote this partition of $\{\{1\},\{2\},\{3\},\{7\},\{4,5,6,8\}\}$ by π_1.

Similarly if we restrict the clock set to contain only y, states 2 and 7 must be grouped together. Similarly, states 3 and 7 must be grouped because of the a transitions out of state 2. The a transitions out of state 5 force states 6 and

7 to be grouped as well. The b transitions with timing constraint $y > 1, y := 0$ require that states 5 and 8 must also be grouped with 2,3,6, and 7, giving the picture in Figure 4(a). with $q_0 = \{1\}$, $q_1 = \{2,3,4,5,6,7,8\}$ and $q_2 = \{4\}$. Because the partitions are not orthogonal—their product still groups 5, 6, and 8—we know there can be no pair of orthogonal admissible pairs for BUFF where the two clock partitions are $\{x\}$ and $\{y\}$. Thus, if there is any pair of partitions to satisfy the hypothesis of Theorem 15, then the clock sets must be C and $\emptyset$. Repeating the process will yield the automaton shown in Figure 4(b) where $q_0 = \{1\}$ and $q_1 = \{2,3,4,5,6,7,8\}$.



(a) Using only $\{y\}$: BUFF

(b) $\emptyset$ decomposition of BUFF

Figure 4: Partitions

The other partition, will have the clock set $C = \{x, y\}$, but in order to satisfy the requirements of Theorem 15, it must be both non-trivial and orthogonal to the partition just derived. This is possible and the result is shown in Figure 5. Thus BUFF does have a parallel decomposition. One of the components is a timed automaton and the other is an untimed one.

In particular instances, one can explore if a timed automaton can be split into an untimed automaton and a timed one. This involves exploring state partitions induced by the trivial clock partition.

5 Conclusion and Future Work

We have presented a generalisation of the standard state based decomposition theory for timed automata. In doing so we have followed the steps necessary for the standard theory. Of course, the technical details were different due to the presence of time. The theory developed is applicable to the general class of real-time automata.

The aim was to present a base theory which can be specialised to other situations. One such specialisation under consideration is state clock automata (SC automata) introduced in [Raskin and Schobbens, 1997]. An SC automaton is like a timed automaton where states are decorated with collections of atomic

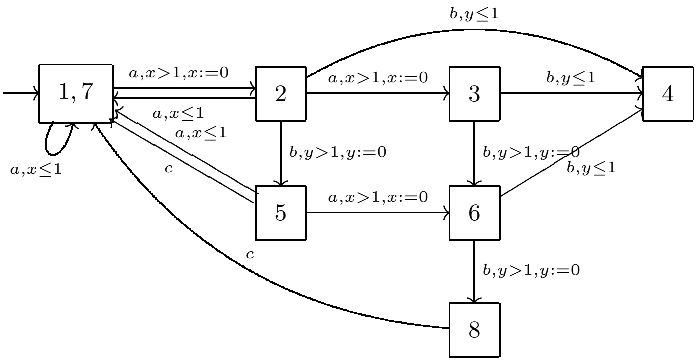


Figure 5: A partition orthogonal to the one in Figure 4(b)

propositions. The automaton has two clocks for each atomic propositions (acting as history and prophecy clocks). They also develop a temporal logic which can be translated into SC automata. The process of the translation introduces a pair of clocks for each basic property. Our aim to adapt this general theory to obtain an abstraction theorem suitable for model checking. For that endeavour Lemma 14 will be most suitable. We also hope that general theory can shed some light on compositional techniques. That is, one can impose restrictions on the structure of the components to facilitate compositional verification. We are also investigating the construction of a tool which takes user input to guide the decomposition process. This is based on techniques investigated for untimed systems in [Kaltenbach, 1996].

References

[Alur and Dill, 1994] Alur, R. and Dill, D. (1994). A Theory of Timed Automata. *Theoretical Computer Science*, 126:183–235.

[Alur et al., 1995] Alur, R., Itai, A., Kurshan, R., and Yannakakis, M. (1995). Timing Verification by Successive Approximation. *Information and Computation*, 118:142–157.

[Clarke et al., 1994] Clarke, E. M., Grumberg, O., and Long, D. E. (1994). Model checking and abstraction. *ACM Transactions on Programming Languages and Systems*, 16(5):1512–1542.

[Hartmanis and Stearns, 1966] Hartmanis, J. and Stearns, R. E. (1966). *Algebraic Structure Theory of Sequential Machines*. Prentice Hall.

[Holcombe, 1982] Holcombe, W. M. L. (1982). *Algebraic Automata Theory*. Cambridge University Press.

[Kaltenbach, 1996] Kaltenbach, M. (1996). *Interactive Verification by Exploiting Program Design Knowledge: A Model Checker for UNITY*. PhD thesis, Department of Computer Science, University of Texas Austin.

[Raskin and Schobbens, 1997] Raskin, J.-F. and Schobbens, P.-Y. (1997). State Clock Logic: A Decidable Real-Time Logic. In Maler, O., editor, *Hybrid and Real-Time Systems*, volume 1201 of LNCS, pages 33–47. Springer-Verlag.