



Deciding Bisimilarity for Alternating Timed Automata

Chris Chilton

Based on work with James Worrell

Trinity Term 2008

chris.chilton@balliol.ox.ac.uk

Oxford University Computing Laboratory
Parks Road, Oxford, OX1 3QD, UK
www.comlab.ox.ac.uk

Contents

1	Introduction	1
2	Background On Bisimulation	3
2.1	Transition systems	3
2.2	Bisimulations and bisimilarity	4
2.3	Non-bisimilarity	5
2.4	Semi-decidability of non-bisimilarity	7
3	Timed Transition Systems	9
4	Alternating Timed Automata	11
4.1	Clock constraints	11
4.2	Alternating timed automata	12
4.3	Transition systems, bisimilarity and selectors	13
4.4	Problem	14
5	Non-Bisimilarity Is Semi-Decidable	15
5.1	Abstracting time	15
5.2	Rational bisimilarity	16
5.3	Semi-decidability of non-bisimilarity	19
6	Bisimilarity Is Semi-Decidable	21
6.1	Congruences	21
6.2	Bisimulation bases	22
6.3	A bisimulation base for bisimilarity	23
6.4	Well-quasi-orders	26
6.5	A finite bisimulation base	26
6.6	Bisimilarity is semi-decidable	28

7 Summary**31**

Introduction

Alternating timed automata (ATAs) are powerful models for computation, with numerous applications to many branches of computer science and mathematics.

ATAs stem from the foundational timed automata of Alur and Dill [1], by generalising the non-determinism to the richer concept of alternation. In light of this fact, it is known that ATAs are closed under all Boolean operators making them particularly well suited to the modelling of temporal logics [2, 3].

Temporal logics themselves have been growing in popularity in recent years as they are notably adept to the model-checking and analysing of real-time systems, for example, verifying the correctness of hardware designs and critical software components. A reasonable question to ask about these systems is do they satisfy some safety constraint?

Questions of this nature are intimately related with language inclusion. Indeed, one way of addressing the question is to construct temporal formulas for the system concerned and the specification. Models for these formulas can then be assembled; perhaps by constructing ATAs. We can then check that the safety constraint is maintained by checking for the inclusion of the language for the system model within the language for the specification model.

It is known for one-clock alternating timed automata that language inclusion/equivalence is decidable (proved in [4], inspired by [5]), yet it has non-primitive recursive complexity. Bisimilarity on the other hand often has much lower complexity than language equivalence, and furthermore implies the latter, although the converse is not true. This makes bisimilarity a more practical and preferable test to perform on automata; especially considering that in practice, two language equivalent automata are often bisimilar. Thus it is certainly worthwhile to pursue our desired result.

Bisimulation equivalence was first introduced in 1980 by Robin Milner [6] in an attempt to provide a branching-time semantics of equivalence for process algebras. The theory was based primarily on Milner's own CCS, but can equally be applied to other process calculi, such as Hoare's CSP, the π -calculus or even to the more general notion of transition systems.

Linear-time equivalencies, for example trace-semantics, only provide an indication of what can be communicated, and as such are incapable of distinguishing processes that behave in fundamentally different ways. On the other hand, the theory of bisimilarity provides a notion of equivalence that prevails the inadequacy of linear-time equivalencies. Not only can we be sure that two processes are language equivalent, we can also be sure that they behave consistently, without one of them exhibiting *impar* behaviour by dead-locking or

diverging¹ unexpectedly. For a comprehensive study of bisimilarity and its application to process algebras, please consult [7, 6, 8].

As we remarked earlier, bisimilarity checking often has much lower complexity than language equivalence. It is known for finite automata that bisimilarity checking is in PTIME, whereas language inclusion is in PSPACE. Perhaps more striking is the affirmative fact that for timed automata, bisimilarity checking is in EXPTIME and language inclusion is undecidable.

In this project we will show that bisimilarity for one-clock alternating timed automata is decidable. We will do this by showing that bisimilarity equivalence is semi-decidable and that non-bisimilarity (bisimilarity inequivalence) is also semi-decidable. The result seems likely given that language inclusion is decidable on these automata [4] and bisimilarity is decidable on the automata of Alur and Dill [9].

Our semi-decision procedure for bisimilarity inequivalence will make use of the stratification of non-bisimilarity, so we need only find an $n \in \mathbb{N}$ such that the automata are non-bisimilar within n -steps. We will also highlight a correspondence between bisimilarity and rational bisimilarity so that we need only concern ourselves with rational time evolutions, which there are countably many of. Our procedure will then be able to enumerate all time evolutions and employ a familiar procedure for non-bisimilarity in the untimed case.

As for the semi-decision procedure for bisimilarity, we will first show that bisimilarity is a congruence, allowing us to construct a bisimulation base that can generate the bisimilarity relation. Using the theory of well-quasi-orders and Graham Higman's work into group theory [11], we will show that we only need to consider the fractional values of clocks, allowing us to abstract away the real-valued time evolutions. This will allow us to show that our bisimulation base is finite, and can be found by enumeration.

In Chapter 2 we introduce the notion of transition systems, bisimulations, bisimilarity and non-bisimilarity in the untimed case. We show that non-bisimilarity is semi-decidable by exhibiting a semi-decision procedure. In Chapter 3 we extend the concepts of Chapter 2 to the timed case. Chapter 4 introduces alternating timed automata to as much depth as we require and concludes by stating formally the problem we are attempting to solve. Chapters 5 and 6 show that non-bisimilarity and bisimilarity are both semi-decidable. Both finish by revealing a semi-decision procedure for non-bisimilarity and bisimilarity respectively. Finally, Chapter 7 provides an overview of our work and suggests further extensions.

¹It should be noted that we have over-simplified bisimilarity up to this point. Milner's bisimulation equivalence, otherwise known as *weak bisimilarity*, only concerns itself with externally visible events, whereas *strong bisimilarity* treats silent τ transitions as any other event. This makes strong bisimilarity capable of distinguishing between different divergence capabilities. Since alternating timed automata cannot make τ transitions, we will not consider this distinction any further.

Background On Bisimulation

In this chapter we present the notion of a transition system, and the definitions of bisimulation, bisimilarity and non-bisimilarity. We show that bisimilarity is both an equivalence relation and a bisimulation. We conclude by proving that non-bisimilarity for an image-finite transition system is the union of stepped non-bisimilarity and that non-bisimilarity is semi-decidable on recursive image-finite transition systems. The results in this chapter are not new, but are crucial for the subsequent development.

2.1 Transition systems

We first introduce transition systems, which will be fundamental models for explaining the semantics of alternating timed automata.

Definition 2.1: A Σ -labelled transition system \mathcal{T} is a triple (Q, Σ, \rightarrow) , where

- Q is a (possibly infinite) set of states
- Σ is a (possibly infinite) alphabet
- $\rightarrow \subseteq Q \times \Sigma \times Q$ is the transition relation.

The system can make a transition from state q to q' reading $\sigma \in \Sigma$, written as $q \xrightarrow{\sigma} q'$, if and only if $(q, \sigma, q') \in \rightarrow$. \diamond

The transition systems that we generate will have a number of useful properties, which we characterise below.

Definition 2.2: A transition system \mathcal{T} is said to be *image-finite* if for each $q \in Q$ and $\sigma \in \Sigma$ the set $\{q' | q \xrightarrow{\sigma} q'\}$ is finite. Note that if Q is finite then \mathcal{T} is automatically image-finite. \diamond

Definition 2.3: Transition system \mathcal{T} is *recursive* if Q and Σ are both recursive sets¹, and $\rightarrow \subseteq Q \times \Sigma \times Q$ is a recursive relation. \diamond

¹A set S is said to be recursive if membership in S is decidable.

2.2 Bisimulations and bisimilarity

Now we introduce a number of relations over transition systems. Let \mathcal{T} be a transition system, with set of states Q , as defined in the previous section.

Definition 2.4: A *bisimulation* on \mathcal{T} is defined to be a binary relation $R \subseteq Q \times Q$, such that if $p R q$, then

$$\begin{aligned} & (\forall p' : p \xrightarrow{\sigma} p' : (\exists q' : q \xrightarrow{\sigma} q' : p' R q')) \\ \wedge & (\forall q' : q \xrightarrow{\sigma} q' : (\exists p' : p \xrightarrow{\sigma} p' : p' R q')) \end{aligned}$$

◇

Equipped with the concept of a bisimulation, we can go on to define the bisimilarity relation.

Definition 2.5: The *bisimilarity* relation on \mathcal{T} is defined to be the union of all the individual bisimulations on \mathcal{T} , and is denoted by \sim . More precisely

$$\sim = \bigcup \{R \mid R \text{ is a bisimulation on } \mathcal{T}\}$$

◇

It is well-known that bisimilarity is an equivalence relation, as the following lemma demonstrates.

Lemma 2.6: \sim is an equivalence relation.

Proof: Consider separately the cases of reflexivity, symmetry and transitivity.

- *Reflexivity:* Note that $Id_Q \triangleq \{(q, q) \mid q \in Q\}$ is a bisimulation, thus is contained within \sim .
- *Symmetry:* Assume $p \sim q$, then there exists a bisimulation R such that $p R q$. Observe that $R^{-1} \triangleq \{(y, x) \mid (x, y) \in R\}$ is a bisimulation and that $q R^{-1} p$, so $q \sim p$.
- *Transitivity:* Assume $p \sim q$ and $q \sim r$; then there exists bisimulations R and S such that $p R q$ and $q S r$. We now show that the relational composition $R; S$, alternatively written as $S \circ R$, is a bisimulation.

If $p R; S q$, then there exists an r such that $p R r$ and $r S q$. So for all p' reachable from p by reading a σ , there is an r' reachable from r by reading that same symbol such that $p' R r'$. Since S is a bisimulation it must be the case that there exists a q' reachable from q by reading the same σ such that $r' S q'$. Thus it holds that $p' R; S q'$.

□

Crucially the bisimilarity relation is itself a bisimulation, as the proceeding lemma shows. This rather neat result allows us to condense our proofs in later parts of the project.

Lemma 2.7: \sim is a bisimulation on \mathcal{T} .

Proof: Note that \sim is always non-empty if $Q \neq \emptyset$. So let $(p, q) \in \sim$, then there exists a bisimulation R such that $p R q$. So $(\forall p' : p \xrightarrow{\sigma} p' : (\exists q' : q \xrightarrow{\sigma} q' : p' R q'))$. But $R \subseteq \sim$, hence $(\forall p' : p \xrightarrow{\sigma} p' : (\exists q' : q \xrightarrow{\sigma} q' : p' \sim q'))$.

Thus \sim is a bisimulation. \square

Given that bisimilarity is both an equivalence relation and a bisimulation, we can give an alternative definition of \sim , which is particularly well-suited to syntactic proofs.

Corollary 2.8: Bisimilarity on \mathcal{T} is the largest equivalence relation R on Q exhibiting the property

$$p R q \Leftrightarrow (\forall p' : p \xrightarrow{\sigma} p' : (\exists q' : q \xrightarrow{\sigma} q' : p' R q'))$$

Note that the implication has now been replaced by a biconditional.

Proof: Any relation R satisfying the formula above is a bisimulation, so $R \subseteq \sim$. Furthermore, bisimilarity is contained within R , as we now show.

Define $R \triangleq \{(p, q) \mid (\forall p' : p \xrightarrow{\sigma} p' : (\exists q' : q \xrightarrow{\sigma} q' : p' \sim q'))\}$, and suppose that $p R q$ and $p \xrightarrow{\sigma} p'$, then $(\exists q' : q \xrightarrow{\sigma} q' : p' \sim q')$. But, if $p' \sim q'$, then $p' R q'$ by definition of R ; hence $\sim \subseteq R$. Thus we have shown that $\sim = R$. \square

2.3 Non-bisimilarity

Acquainted with bisimulations and bisimilarity, we can now introduce the complement relation of bisimilarity known as non-bisimilarity. This relation will be of the utmost importance in proving the decidability of bisimilarity.

Definition 2.9: The complement relation of bisimilarity is called *non-bisimilarity* and is denoted by $\not\sim$.

We define a sequence of relations on the set of states Q that approximate non-bisimilarity from below. First, define $\not\sim_0$ (0-step non-bisimilarity) to be the empty relation. Then $p \not\sim_{n+1} q$ if, and only if

$$\begin{aligned} & (\exists p' : p \xrightarrow{\sigma} p' : (\forall q' : q \xrightarrow{\sigma} q' : p' \not\sim_n q')) \\ \vee & (\exists q' : q \xrightarrow{\sigma} q' : (\forall p' : p \xrightarrow{\sigma} p' : p' \not\sim_n q')) \end{aligned}$$

\diamond

We now present the notion of stepped bisimilarity as the complement of stepped non-bisimilarity. This concept is introduced to assist in simplifying the proof of a fact relating to non-bisimilarity.

Definition 2.10: We define a sequence of relations over Q that represent the negation of non-stepped bisimilarity, namely stepped bisimilarity. First note that \sim_0 is defined to be $Q \times Q$. Now we define \sim_n , which is $(Q \times Q) \setminus \not\sim_n$, in the case that $n > 0$.

$$\begin{aligned}
& p \sim_n q \\
\Leftrightarrow & \neg((\exists p' : p \xrightarrow{\sigma} p' : (\forall q' : q \xrightarrow{\sigma} q' : p' \not\sim_n q')) \vee \\
& (\exists q' : q \xrightarrow{\sigma} q' : (\forall p' : p \xrightarrow{\sigma} p' : p' \not\sim_n q))) \\
\Leftrightarrow & (\forall p' : p \xrightarrow{\sigma} p' : \neg(\forall q' : q \xrightarrow{\sigma} q' : p' \not\sim_n q)) \wedge \\
& (\forall q' : q \xrightarrow{\sigma} q' : \neg(\forall p' : p \xrightarrow{\sigma} p' : p' \not\sim_n q)) \\
\Leftrightarrow & (\forall p' : p \xrightarrow{\sigma} p' : (\exists q' : q \xrightarrow{\sigma} q' : p' \sim_{n-1} q)) \wedge \\
& (\forall q' : q \xrightarrow{\sigma} q' : (\exists p' : p \xrightarrow{\sigma} p' : p' \sim_{n-1} q))
\end{aligned}$$

◇

It is folklore that for image-finite transition systems $\not\sim = \bigcup_{n \in \mathbb{N}} \not\sim_n$, as the next lemma reveals.

Lemma 2.11: For an image-finite transition system $\not\sim = \bigcup_{n \in \mathbb{N}} \not\sim_n$.

Proof: First observe that $\not\sim = \bigcup_{n \in \mathbb{N}} \not\sim_n$ if, and only if, $\sim = \bigcap_{n \in \mathbb{N}} \sim_n$.

We will show that each side of the second equality is a subset of the other.

- Proof of $\sim \subseteq \bigcap_{n \in \mathbb{N}} \sim_n$. We show $\sim \subseteq \sim_n$ for each $n \in \mathbb{N}$ by induction on n .

Base case $n = 0$: $\sim_0 = Q \times Q$, hence $\sim \subseteq \sim_0$.

Inductive case: Assume $\sim \subseteq \sim_n$, then show $\sim \subseteq \sim_{n+1}$.

$$\begin{aligned}
& (p, q) \in \sim \\
\Leftrightarrow & (\forall p' : p \xrightarrow{\sigma} p' (\exists q' : q \xrightarrow{\sigma} q' : p' \sim q)) \\
\Rightarrow & (\forall p' : p \xrightarrow{\sigma} p' (\exists q' : q \xrightarrow{\sigma} q' : p' \sim_n q)) \\
\Leftrightarrow & (p, q) \in \sim_{n+1}
\end{aligned}$$

- Proof of $\sim \supseteq \bigcap_{n \in \mathbb{N}} \sim_n$.

Recall \sim is the union of all bisimulations, so it is sufficient to prove that $\bigcap_{n \in \mathbb{N}} \sim_n$ is a bisimulation.

$$\begin{aligned}
& (p, q) \in \bigcap_{n \in \mathbb{N}} \sim_n \\
\Leftrightarrow & (\forall n : n \in \mathbb{N} : p \sim_{n+1} q) \\
\Leftrightarrow & (\forall n : n \in \mathbb{N} : (\forall p' : p \xrightarrow{\sigma} p' (\exists q' : q \xrightarrow{\sigma} q' : p' \sim_n q'))) \\
\Rightarrow & \{\text{by image-finiteness}\} \\
& (\forall p' : p \xrightarrow{\sigma} p' (\exists q' : q \xrightarrow{\sigma} q' : (\forall n : n \in \mathbb{N} : p' \sim_n q'))) \\
\Leftrightarrow & (\forall p' : p \xrightarrow{\sigma} p' (\exists q' : q \xrightarrow{\sigma} q' : p' \bigcap_{n \in \mathbb{N}} \sim_n q'))
\end{aligned}$$

Hence $\bigcap_{n \in \mathbb{N}} \sim_n$ is a bisimulation.

□

2.4 Semi-decidability of non-bisimilarity

In this final section we consider what it means for a language to be semi-decidable. Furthermore, we show that non-bisimilarity is semi-decidable.

Definition 2.12: A language \mathcal{L} is said to be *semi-decidable* if, and only if, there exists a deterministic Turing Machine \mathcal{M} such that $w \in \mathcal{L}$ implies \mathcal{M} accepts w and if $w \notin \mathcal{L}$, then \mathcal{M} diverges. \diamond

It is possible to weaken the requirement that \mathcal{M} be deterministic, as the proceeding lemma shows.

Lemma 2.13: A language \mathcal{L} is *semi-decidable* if, and only if, there exists a non-deterministic Turing Machine \mathcal{N} such that $w \in \mathcal{L}$ if, and only if, \mathcal{N} has an accepting computation on w .

Proof: The computation of \mathcal{N} on w can be seen as a finitely-branching tree, in which $w \in \mathcal{L}$ if, and only if, the tree has an accepting leaf. The corresponding deterministic semi-decision procedure performs breadth-first search on this tree. \square

Thus the expressiveness of deterministic and non-deterministic semi-decision procedures is the same, so we are free to present either as an algorithm for semi-deciding non-bisimilarity. We will favour the second option, as non-deterministic algorithms are sometimes clearer to read than their deterministic counterparts.

Proposition 2.14: If \mathcal{T} is recursive and image-finite, then $\not\sim$ is semi-decidable.

Proof: To show that $p \not\sim q$, it is sufficient by Lemma 2.11 to find an $n \in \mathbb{N}$ such that $p \not\sim_n q$. We therefore present a non-deterministic semi-decision procedure for non-bisimilarity making use of this fact.

```

1 NON-BISIM( $p, q$ )
2   //  $\mathbb{N}$  is recursive;
3   Guess  $n \in \mathbb{N}$ ;
4   NON-BISIM-STEP( $p, q, n$ );
5   return true;
6 end

7 NON-BISIM-STEP( $p, q, n$ )
8   if  $n = 0$  then
9     | diverge;
10  else
11    //  $\Sigma$  is recursive;
12    Guess  $a \in \Sigma$ ;
13    if (true  $\sqcap$  false) then
14      | //  $Q$  and  $\rightarrow$  are recursive;
15      | Guess  $p' \in Q$  such that  $p \xrightarrow{a} p'$ ;
16      | // Finitely many iterations by image-finiteness;
17      | foreach  $q \xrightarrow{a} q'$  do
18        | NON-BISIM-STEP( $p', q', n - 1$ );
19      | end
20    else
21      | //  $Q$  and  $\rightarrow$  are recursive;
22      | Guess  $q' \in Q$  such that  $q \xrightarrow{a} q'$ ;
23      | // Finitely many iterations by image-finiteness;
24      | foreach  $p \xrightarrow{a} p'$  do
25        | NON-BISIM-STEP( $p', q', n - 1$ );
26      | end
27    end
28  end
29 end

```

Algorithm 1: Semi-decision procedure for bisimilarity inequivalence

This code can clearly be resolved to a deterministic procedure that is always guaranteed to terminate in the case that $p \not\sim q$. A comment within the algorithm justifies that the following line is valid, or in some cases, terminable.

We now briefly relate the procedure $\text{NON-BISIM-STEP}(p, q, n)$ to the definition of $p \not\sim_n q$. The non-deterministic choice of **true** and **false** corresponds to the disjunction in the definition of $p \not\sim_n q$. The guessing of a symbol and state characterises the existential quantifiers and the **foreach** constructs relate to the universal quantifiers. \square

In this chapter we have introduced the main concepts relating to bisimilarity in the untimed case. We have shown that there is a simple semi-decision algorithm for non-bisimilarity, which we hope only requires minor alteration for application to the timed case. In Chapter 3 we extend and augment these introductory notions with the expressivity of time.

Timed Transition Systems

We now extend the transition systems considered in the previous chapter to include time. As a result, it is necessary to refine the definitions of a bisimulation and bisimilarity to capture the transition system's ability to make timed transitions.

Definition 3.1: A *timed transition system* \mathcal{T} is a tuple $(Q, \Sigma, \rightarrow, \rightsquigarrow)$, where

- Q is a (possibly infinite) set of states
- Σ is a (possibly infinite) alphabet
- $\rightarrow \subseteq Q \times \Sigma \times Q$ is the Σ -labelled transition relation
- $\rightsquigarrow \subseteq Q \times \mathbb{R}_0^+ \times Q$ is the time-labelled transition relation.

The system can make a Σ -labelled transition from p to q reading σ , written as $p \xrightarrow{\sigma} q$, if, and only if, $(p, \sigma, q) \in \rightarrow$.

Furthermore, the system can make a time-labelled transition from p to q in time t , written as $p \rightsquigarrow^t q$, if, and only if, $(p, t, q) \in \rightsquigarrow$. ◇

Henceforth let \mathcal{T} be the timed transition system defined in the previous definition, then we can define a timed-bisimulation on \mathcal{T} as follows.

Definition 3.2: A *timed bisimulation* on timed transition system \mathcal{T} is a relation R on $Q \times Q$ satisfying

$$p R q \Rightarrow (\forall p' : p \xrightarrow{\sigma} p' : (\exists q' : q \xrightarrow{\sigma} q' : p' R q')) \wedge (\forall q' : q \xrightarrow{\sigma} q' : (\exists p' : p \xrightarrow{\sigma} p' : p' R q')) \wedge$$

$$(\forall p' : p \rightsquigarrow^t p' : (\exists q' : q \rightsquigarrow^t q' : p' R q')) \wedge (\forall q' : q \rightsquigarrow^t q' : (\exists p' : p \rightsquigarrow^t p' : p' R q'))$$

Note that a timed bisimulation is simply an ordinary bisimulation over the label set $\Sigma \cup \mathbb{R}_0^+$.

The formula above formally captures the intuition that whenever p can make a σ transition, q can match it with the same symbol, and vice versa. Similarly if p can make a timed transition t then, q can also match that transition with the same t and vice versa. ◇

The definition of timed bisimilarity now carries over in the instinctive way from the untimed case.

Definition 3.3: The *timed bisimilarity* relation on timed transition system \mathcal{T} is defined to be the union of all the timed bisimulations on \mathcal{T} and is denoted by \sim . Mathematically speaking

$$\sim = \bigcup \{R \mid R \text{ is a timed bisimulation on } \mathcal{T}\}$$

◇

From here on let \sim be the timed bisimilarity relation on \mathcal{T} . Again, as for the untimed case, timed bisimilarity is both an equivalence relation and a bisimulation.

Lemma 3.4: \sim is an equivalence relation and a bisimulation.

Proof: Note that timed transition system $(Q, \Sigma, \rightarrow, \rightsquigarrow)$ is synonymous to the untimed transition system $(Q, \Sigma \cup \mathbb{R}_0^+, \rightarrow \cup \rightsquigarrow)$. Moreover, the timed bisimulations on the former are equal to the bisimulations on the latter, hence the bisimilarity relation on both of the transition systems is the same. Thus \sim is both an equivalence relation and bisimulation. \square

We have now completed our extension of the definitions in Chapter 2 to the timed case. In the subsequent chapter (Chapter 4) we will introduce alternating timed automata, and link these computational models back to the theory in this and the previous chapters.

Alternating Timed Automata

In this chapter we introduce alternating timed automata (ATAs) and describe their states and configurations. We see how to generate a timed transition system from an ATA and conclude by formally stating the problem that we are attempting to solve: deciding timed bisimilarity for 1-clock ATAs.

4.1 Clock constraints

Before defining alternating timed automata we need to consider the ways in which we will impose time constraints on them. We will do this by using a collection of clock variables and Boolean conditions.

Once the automaton has been activated, we will start a global real-valued clock. As a symbol in the alphabet is read by the automaton, we will record the time at which that symbol was read according to the global clock.

The automaton possesses local clock variables, which will pass time at exactly the same rate as the global clock. However we will allow local clock variables to be reset to 0, as requested, on the completion of a transition between locations. This will allow us to build up complicated behaviours on our timed automata, by placing constraints on when transitions can and can't occur.

The remainder of this section builds up the notion of clock constraints in a more formal setting.

Definition 4.1: Let X be a set of clock variables, then $\Phi(X)$ is the set of clock constraints φ , such that

$$\varphi ::= x \leq c \mid x \geq c \mid \neg\varphi \mid \varphi \wedge \varphi$$

where $x \in X$ and $c \in \mathbb{Q}_0^+$. ◇

Definition 4.2: A *clock valuation* v is a mapping $v : X \rightarrow \mathbb{R}_0^+$. We say that v satisfies φ , a clock constraint on X , if, and only if, $v(\varphi)$ evaluates to *true*, where v is extended to a function $v : \Phi(X) \rightarrow \{false, true\}$ in the natural way. ◇

Definition 4.3: Let v be a clock valuation for the set of clock variables X and let $t \in \mathbb{R}_0^+$. We write $v + t$ for the function that applied to any $x \in X$ yields $v(x) + t$. Also let $Y \subseteq X$, then define a valuation $[Y \mapsto t]v$ by

$$\begin{aligned} [Y \mapsto t]v(x) &= t && \text{(for } x \in Y) \\ [Y \mapsto t]v(x) &= v(x) && \text{(for } x \notin Y) \end{aligned}$$

◇

4.2 Alternating timed automata

Given the theory on clock constraints we can now go on to provide a general description of alternating timed automata.

Definition 4.4: Given a set Q , $\mathcal{B}^+(Q)$ is defined to be the set of positive Boolean formulas β over Q , such that

$$\beta ::= Q \mid \beta \vee \beta \mid \beta \wedge \beta$$

◇

Definition 4.5: An *alternating timed automaton* \mathcal{A} is a 6-tuple $(\Sigma, S, s_0, X, \delta, F)$, where

- Σ is a finite alphabet
- S is a finite set of locations
- $s_0 \in S$ is the initial location
- X is a finite set of clock variables
- $\delta : S \times \Sigma \times \Phi(X) \rightarrow \mathcal{B}^+(S \times \mathcal{P}(X))$ is the transition (partial) function
- $F \subseteq S$ is the set of accepting locations

We impose a *partitioning condition* over the clock-constraint state-space $(\mathbb{R}_0^+)^X$ of \mathcal{A} . Fix $s \in S$ and $\sigma \in \Sigma$, and let

$$P = \{ [\varphi] \mid \varphi \in \Phi(X) \text{ and } \delta(s, \sigma, \varphi) \text{ is defined} \}$$

then we require the following constraints to hold

- Coverage: $\bigcup_{p \in P} p = (\mathbb{R}_0^+)^X$
- Disjointedness: $(\forall p, p' : p, p' \in P \wedge p \neq p' : p \cap p' = \emptyset)$

◇

Definition 4.6: A *state* of \mathcal{A} is a pair (s, v) containing the current location $s \in S$ and a clock valuation $v : X \rightarrow \mathbb{R}_0^+$. Hence the set of states Q is defined to be $S \times (X \rightarrow \mathbb{R}_0^+)$. \diamond

Definition 4.7: A *configuration* is defined to be a set of states. We say that a configuration is accepting if, and only if, every location within the configuration is a member of F . The set of configurations is simply the powerset of the set of states Q , and is written as $\mathcal{P}(Q)$. \diamond

4.3 Transition systems, bisimilarity and selectors

In this section we see how to generate a transition system from an alternating timed automaton \mathcal{A} , as defined in the previous section. Furthermore, we introduce the notion of a selector, which we use to resolve the non-determinism introduced by the transition relation (i.e. the disjuncts) of the automaton.

Definition 4.8: Let $M \subseteq Q$, then we say M is a *model* for $\beta \in \mathcal{B}^+(S \times \mathcal{P}(X))$ with respect to clock valuation v if, and only if, the satisfaction relation $(M, v) \models \beta$ holds.

The satisfaction relation is defined over the structure of β as follows

- $(M, v) \models \beta_1 \wedge \beta_2 \Leftrightarrow (M, v) \models \beta_1 \wedge (M, v) \models \beta_2$
- $(M, v) \models \beta_1 \vee \beta_2 \Leftrightarrow (M, v) \models \beta_1 \vee (M, v) \models \beta_2$
- $(M, v) \models (s, R) \Leftrightarrow (s, [R \mapsto 0]v) \in M$

We say that model M of β is minimal with respect to clock valuation v if, and only if, $(\nexists N : N \subset M : (N, v) \models \beta)$. \diamond

Definition 4.9: The *timed transition system* for \mathcal{A} is a quadruple $\mathcal{T}_{\mathcal{A}} = (\mathcal{P}(Q), \Sigma, \rightarrow, \rightsquigarrow)$.

$\rightarrow \subseteq \mathcal{P}(Q) \times \Sigma \times \mathcal{P}(Q)$ is defined to be the Σ -labelled transition system. We say that $(Q_1, \sigma, Q_2) \in \rightarrow$, written as $Q_1 \xrightarrow{\sigma} Q_2$, if, and only if

$$Q_2 = \bigcup_{(s,v) \in Q_1} \{M \mid \exists \varphi \in \Phi(X), \delta(s, \sigma, \varphi) \text{ is defined, } (M, v) \models \delta(s, \sigma, \varphi) \text{ and } M \text{ is minimal}\}$$

Nota bene: By the partitioning condition there is guaranteed to be a single $\varphi \in \Phi(X)$ such that $\delta(s, \sigma, \varphi)$ is defined when there is a transition from location $s \in S$ reading $\sigma \in \Sigma$.

\rightsquigarrow is defined to be the time-labelled transition system $\mathcal{P}(Q) \times \mathbb{R}_0^+ \times \mathcal{P}(Q)$. We say that $(Q_1, t, Q_2) \in \rightsquigarrow$, written as $Q_1 \xrightarrow{t} Q_2$, if, and only if, $Q_2 = \{(s, v + t) \mid (s, v) \in Q_1\}$. \diamond

Definition 4.10: Bisimilarity of ATAs

The notion of bisimilarity for ATAs is derived from the definition of timed bisimilarity in Chapter 3.

Let $\mathcal{B} = (\Sigma, S', s'_0, X', \delta', F')$ be an additional alternating timed automaton sharing the same alphabet as \mathcal{A} . Furthermore, let $\mathbf{0}_{\mathcal{A}} : X \rightarrow \mathbb{R}_0^+$ and $\mathbf{0}_{\mathcal{B}} : X' \rightarrow \mathbb{R}_0^+$ be zero functions.

We define \mathcal{A} and \mathcal{B} to be bisimilar if, and only if, $\{(s_0, \mathbf{0}_{\mathcal{A}})\} \sim \{(s'_0, \mathbf{0}_{\mathcal{B}})\}$ on the timed transition system $\mathcal{T}_{\mathcal{A}, \mathcal{B}}$ formed by taking the disjoint union of $\mathcal{T}_{\mathcal{A}}$ and $\mathcal{T}_{\mathcal{B}}$. \diamond

Definition 4.11: A *selector* for a transition $C \xrightarrow{\sigma} C'$, where $C = \{(s_i, v_i)\}_{i=1}^n$, is a function $f : \{1, \dots, n\} \rightarrow \mathcal{B}^+(S \times \mathcal{P}(X))$ such that $f(i)$ is a disjunct of $\delta(s_i, \sigma, \varphi_i)$ in disjunctive normal form, where φ_i is the uniquely determined clock constraint satisfying v_i .

Observe that $f(i)$ is a conjunction of pairs of locations and clock-resets, so let A_i be the set of pairs occurring in $f(i)$, such that $f(i) = \bigwedge A_i$. Now, the new state C' is derived as follows

$$C' = \bigcup_{i=1}^n \{(s', [X \mapsto 0]v_i) \mid (s', X) \in A_i\}$$

In effect, the purpose of a selector is to resolve the non-deterministic choice inherent in the transition system. In the next chapter we will present a lemma on selectors that will be vital for proving non-bisimilarity is semi-decidable. \diamond

4.4 Problem

We have now finished our coverage of the preliminary concepts required for this project, thus putting us in a position to formally define the problem we are addressing.

1-CLOCK ATA BISIMILARITY

Instance: Two 1-clock alternating timed automata denoted by \mathcal{A} and \mathcal{B} .

Question: Is \mathcal{A} bisimilar to \mathcal{B} ?

We say that an ATA has one clock if the set of clock variables defined in Definition 4.5 is a singleton. As we are restricting ourselves to one-clock ATAs from here on, we will take the liberty of replacing the clock valuation with a single real number, which will represent the value of the clock. Hence the set of all states Q will now be defined as $S \times \mathbb{R}_0^+$ (*cf.* Definition 4.5).

This subtly changes the definition of bisimilarity such that \mathcal{A} and \mathcal{B} are bisimilar if, and only if, $\{(s_0, 0)\} \sim \{(s'_0, 0)\}$ on the timed transition system formed by taking the disjoint union of $\mathcal{T}_{\mathcal{A}}$ and $\mathcal{T}_{\mathcal{B}}$.

We need to restrict to one clock since bisimilarity is undecidable for two-clock ATAs. This is an easy consequence of the fact that reachability is undecidable for two-clock ATAs.

Non-Bisimilarity Is Semi-Decidable

Henceforth let \mathcal{A} and \mathcal{B} denote two one-clock alternating timed automata. Furthermore, let $\mathcal{T}_{\mathcal{A},\mathcal{B}} = (\mathcal{P}(S \times \mathbb{R}_0^+), \Sigma, \rightarrow, \rightsquigarrow)^1$ denote the timed transition system formed by taking the disjoint union of $\mathcal{T}_{\mathcal{A}}$ and $\mathcal{T}_{\mathcal{B}}$. We now begin to show that non-bisimilarity for one-clock alternating timed automata is semi-decidable.

5.1 Abstracting time

In order to show that bisimilarity is semi-decidable for alternating timed automata, we need to be able to move from the world of real-valued time evolutions to rational ones. It is common knowledge that \mathbb{R}_0^+ is uncountable, so we will be unable to enumerate the whole of this set, which is a necessity for our semi-decision procedure. On the other hand, \mathbb{Q}_0^+ is enumerable, so if we can show a correspondence between real and rational time evolutions, it will be sufficient to prove that rational bisimilarity holds for the alternating timed automata.

Definition 5.1: Let k be the largest constant occurring in the clock constraints of \mathcal{A} and \mathcal{B} , then we define $REG = \{\{i\} | 0 \leq i \leq k, i \in \mathbb{N}\} \cup \{(i, i+1) | 0 \leq i < k, i \in \mathbb{N}\} \cup \{(k, \infty)\}^2$ to be a partition of \mathbb{R}_0^+ .

The function $reg : \mathbb{R}_0^+ \rightarrow REG$ is defined to take real-valued times t to the corresponding element of REG to which t belongs.

Now we can extend reg , so that it applies to configurations (collections of states) rather than a particular state. The function $abs : \mathcal{P}(S \times \mathbb{R}_0^+) \rightarrow \mathcal{P}(S \times REG)$ is defined to be $abs(C) = \{(s, reg(t)) | (s, t) \in C\}$, where C is an arbitrary configuration in $\mathcal{P}(S \times \mathbb{R}_0^+)$. \diamond

Definition 5.2: The *time abstraction function* $H : \mathcal{P}(S \times \mathbb{R}_0^+) \rightarrow (\mathcal{P}(S \times REG))^*$ is defined in the following manner.

Given $C \subseteq (S \times \mathbb{R}_0^+)$, we define an equivalence relation, \equiv , on C by $(s, t) \equiv (s', t')$ if, and only if, $t - \lfloor t \rfloor = t' - \lfloor t' \rfloor$.

Let C_1, \dots, C_n be the set of equivalence classes of \equiv , such that, if $i < j$, $(s, t) \in C_i$ and $(s', t') \in C_j$ then $t - \lfloor t \rfloor < t' - \lfloor t' \rfloor$. Now define $H(C) = abs(C_1) \dots abs(C_n)$.

¹Note that S is playing the role of the disjoint union of locations from \mathcal{A} and \mathcal{B} . Moreover, configurations of \mathcal{A} and \mathcal{B} are members of $\mathcal{P}(S \times \mathbb{R}_0^+)$.

²The observant reader may have noticed that in Definition 4.1 we allowed clock comparisons with rational numbers. We have now implicitly scaled time by the greatest common denominator of the aforementioned values so that all comparisons are with integers.

Intuitively, H orders states by the fractional value of their clock values. As an example, consider the configuration $D = \{(s, 0.3), (t, 1.3), (u, 1.5), (u, 3.4)\}$ for an ATA \mathcal{C} . If the greatest clock constraint in \mathcal{C} is 2, then $H(D) = \{(s, (0, 1)), (t, (1, 2))\} \{(u, (2, \infty))\} \{(u, (1, 2))\}$. \diamond

Lemma 5.3: Bisimulation Lemma

If $H(C) = H(D)$ for configurations C and D , then $(\forall t \in \mathbb{R}_0^+ :: (\exists s \in \mathbb{R}_0^+ :: H(C + t) = H(D + s)))$. In effect, this states that if C and D are H equivalent, then there is some $s \in \mathbb{R}_0^+$, specific to each $t \in \mathbb{R}_0^+$ such that configurations $C + t$ and $D + s$ are H equivalent. That is, H -equivalence is a kind of time-abstract bisimulation.

Proof: For a proof of the lemma please refer to Lemma 12 in [2]. \square

Lemma 5.4: Selection Lemma

Let $C = \{(s_i, v_i) | i \in I\}$ and $D = \{(t_j, w_j) | j \in J\}$ be configurations of \mathcal{A} , where \mathcal{A} is described in Definition 4.5. If $H(C) = H(D)$ and $C \xrightarrow{\sigma} C'$ has the same selector as $D \xrightarrow{\sigma} D'$, then $H(C') = H(D')$.

Proof: Since $H(C) = H(D)$, we know that $|C| = |D|$ and for each $(s, v) \in C$ there exists a $(t, w) \in D$ (and vice versa) such that $H(\{(s, v)\}) = H(\{(t, w)\}) = \{(s, \text{reg}(v))\} = \{(t, \text{reg}(w))\}$. Thus $s = t$ and $\text{reg}(v) = \text{reg}(w)$.

As a result, both v and w coincide with a unique clock-constraint $\varphi \in \Phi(X)$, hence $\delta(s, \sigma, \varphi) = \delta(t, \sigma, \varphi) = B$, for some $B \in \mathcal{B}^+(S \times \mathcal{P}(X))$. Now let $\bigwedge A$ be the disjunct of B as prescribed by the selector. So we have

$$\begin{aligned} (s, v) &\xrightarrow{\sigma} \{(s', [X \mapsto 0]v) | (s', X) \in A\} \\ (t, w) &\xrightarrow{\sigma} \{(s', [X \mapsto 0]w) | (s', X) \in A\} \end{aligned}$$

Since $\text{reg}(v) = \text{reg}(w)$, it's the case that $H(\{(s', [X \mapsto 0]v)\}) = H(\{(s', [X \mapsto 0]w)\})$ for $(s', X) \in A$. This applies equally to the other states in C and D , so $H(C') = H(D')$. \square

Equipped with this nomenclature, we can prove an important result that H equivalence preserves bisimilarity. This result is central in showing that we can switch between real and rational timed transitions without affecting bisimilarity.

5.2 Rational bisimilarity

Now we introduce the rational version of transition systems, bisimulations and bisimilarity.

Definition 5.5: If $\mathcal{T}_A = (Q, \Sigma, \rightarrow, \rightsquigarrow)$, then the *rational timed transition system* $\mathcal{T}_A^{\mathbb{Q}}$ is the tuple $(Q', \Sigma, \rightarrow, \rightsquigarrow^{\mathbb{Q}})$, where $\rightsquigarrow^{\mathbb{Q}}$ is \rightsquigarrow restricted to times drawn from \mathbb{Q}_0^+ and Q' is Q with clock values restricted to rational values. \diamond

Definition 5.6: A *rational bisimulation* is a relation R on $Q \times Q$ identical to that in Definition 3.2, except that time evolutions are restricted to rationals. \diamond

Definition 5.7: The *rational timed bisimilarity* relation $\sim_{\mathbb{Q}}$ for timed transition system $\mathcal{T}_{\mathcal{A},\mathcal{B}}^{\mathbb{Q}}$ is defined to be the union of all of the rational timed bisimulations on $\mathcal{T}_{\mathcal{A},\mathcal{B}}^{\mathbb{Q}}$.

$$\sim_{\mathbb{Q}} = \bigcup \{R \mid R \text{ is a rational timed bisimulation on } \mathcal{T}_{\mathcal{A},\mathcal{B}}^{\mathbb{Q}}\}$$

Note that $\sim_{\mathbb{Q}}$ is an equivalence relation and is itself a rational bisimulation. \diamond

The next result shows that if two configurations are bisimilar on a transition system, then they are also rationally bisimilar on the corresponding rational timed transition system and vice versa. This means that henceforth we can restrict our attention to rational transition systems.

Proposition 5.8: $C \sim D$ on $\mathcal{T}_{\mathcal{A},\mathcal{B}}$ if, and only if, $C \sim_{\mathbb{Q}} D$ on $\mathcal{T}_{\mathcal{A},\mathcal{B}}^{\mathbb{Q}}$, where C and D both have rational clock valuations.

Proof: *If direction:* Assume $C \sim_{\mathbb{Q}} D$, then we wish to show that $C \sim D$. Define

$$R \triangleq \{(E, F) \mid \exists E', F' :: H(E \cup F) = H(E' \cup F') \wedge E' \sim_{\mathbb{Q}} F'\}$$

and observe that $C R D$ (simply take the E', F' equal to C, D). It remains to prove that R is a timed bisimulation, for then $R \subseteq \sim$.

- We begin by considering the Σ -labelled transitions.

Suppose $E R F$ and $E \xrightarrow{\sigma} G$, and let E', F' be configurations such that $H(E \cup F) = H(E' \cup F')$ and $E' \sim_{\mathbb{Q}} F'$. We want to find a matching transition $F \xrightarrow{\sigma} I$ such that $G R I$.

Since $H(E \cup F) = H(E' \cup F')$, there is a transition $E' \xrightarrow{\sigma} G'$ with the same selector as $E \xrightarrow{\sigma} G$. Given that $E' \sim_{\mathbb{Q}} F'$, there is a transition $F' \xrightarrow{\sigma} I'$ with $G' \sim_{\mathbb{Q}} I'$.

Again, by $H(E \cup F) = H(E' \cup F')$ there is a transition $F \xrightarrow{\sigma} I$ with the same selector as $F' \xrightarrow{\sigma} I'$. Now we see that $H(G \cup I) = H(G' \cup I')$ and $G' \sim_{\mathbb{Q}} I'$, hence $G R I$.

$$\begin{array}{ccc} E & \xrightarrow{\sigma} & G \\ R & & R \\ F & \xrightarrow{\sigma} & I \end{array} \qquad \begin{array}{ccc} E' & \xrightarrow{\sigma} & G' \\ \sim_{\mathbb{Q}} & & \sim_{\mathbb{Q}} \\ F' & \xrightarrow{\sigma} & I' \end{array}$$

- Now we consider the timed transitions.

$$\begin{aligned} & C R D \\ \Leftrightarrow & (\exists E', F' :: H(C \cup D) = H(E' \cup F') \wedge E' \sim_{\mathbb{Q}} F') \\ \Rightarrow & \{\text{rational bisimulation lemma and } \sim_{\mathbb{Q}} \text{ is a bisimulation}\} \\ & (\forall t \in \mathbb{Q}_0^+ :: (\exists E', F', s :: H((C \cup D) + t) = H((E' \cup F') + s) \wedge (E' + s) \sim_{\mathbb{Q}} (F' + s))) \\ \Leftrightarrow & (\forall t \in \mathbb{Q}_0^+ :: (\exists E'', F'' :: H((C \cup D) + t) = H(E'' \cup F'') \wedge E'' \sim_{\mathbb{Q}} F'')) \\ \Leftrightarrow & (\forall t \in \mathbb{Q}_0^+ :: (C + t) R (D + t)) \end{aligned}$$

Hence R is a bisimulation as required.

Only if direction:

Assume $C \sim D$ and define

$$R \triangleq \{(E, F) \mid E, F \text{ are rational} \wedge \exists E', F' :: H(E \cup F) = H(E' \cup F') \wedge E' \sim F'\}$$

First note that $C R D$ as we can take $E' = C$ and $F' = D$. We now show that R is a rational-timed bisimulation, because then $R \subseteq \sim_{\mathbb{Q}}$.

- We first consider the Σ -labelled transitions.

Suppose $C R D$ and $C \xrightarrow{\sigma} C'$, and let E, F be configurations such that $H(C \cup D) = H(E \cup F)$ and $E \sim F$. We want to find a matching transition $D \xrightarrow{\sigma} D'$ such that $C' R D'$.

Since $H(C \cup D) = H(E \cup F)$, there is a transition $E \xrightarrow{\sigma} E'$ with the same selector as $C \xrightarrow{\sigma} C'$. Given that $E \sim F$, there is a transition $F \xrightarrow{\sigma} F'$ with $E' \sim F'$.

Again, by $H(C \cup D) = H(E \cup F)$ there is a transition $D \xrightarrow{\sigma} D'$ with the same selector as $F \xrightarrow{\sigma} F'$. Now we see that $H(C' \cup D') = H(E' \cup F')$ and $E' \sim F'$, hence $C' R D'$.

$$\begin{array}{ccc} C & \xrightarrow{\sigma} & C' & & E & \xrightarrow{\sigma} & E' \\ R & & R & & \sim & & \sim \\ D & \xrightarrow{\sigma} & D' & & F & \xrightarrow{\sigma} & F' \end{array}$$

So R satisfies the Σ -labelled transitions clause of timed-bisimulations.

- We now consider the timed transitions clause.

$$\begin{aligned} & C R D \\ \Leftrightarrow & (\exists E', F' :: H(C \cup D) = H(E' \cup F') \wedge E' \sim F') \\ \Rightarrow & \{\text{rational bisimulation lemma and } \sim \text{ is a bisimulation}\} \\ & (\forall t \in \mathbb{Q}_0^+ :: (\exists E', F', s :: H((C \cup D) + t) = H((E' \cup F') + s) \wedge (E' + s) \sim (F' + s))) \\ \Leftrightarrow & (\forall t \in \mathbb{Q}_0^+ :: (\exists E'', F'' :: H((C \cup D) + t) = H(E'' \cup F'') \wedge E'' \sim F'')) \\ \Leftrightarrow & (\forall t \in \mathbb{Q}_0^+ :: (C + t) R (D + t)) \end{aligned}$$

Thus R is a rational timed bisimulation and so $C \sim_{\mathbb{Q}} D$. □

5.3 Semi-decidability of non-bisimilarity

At last, we are in a position to present the fruit of our labour in this chapter, namely that non-bisimilarity is semi-decidable. This is a key milestone in the project, as we are half way to proving the decidability of bisimilarity.

Proposition 5.9: $\mathcal{T}_{A,B}^{\mathbb{Q}}$ is recursive and image-finite.

Proof: This follows directly from the definition of ATAs and the fact that we restrict ourselves to rational clock values. \square

We now present an adaptation of the semi-decision procedure in Proposition 2.14 for timed non-bisimilarity.

```

1 NON-BISIM( $C, D$ )
2   //  $\mathbb{N}$  is recursive;
3   Guess  $n \in \mathbb{N}$ ;
4   NON-BISIM-STEP( $C, D, n$ );
5   return true;
6 end
7 NON-BISIM-STEP( $C, D, n$ )
8   if  $n = 0$  then
9     diverge;
10  else
11    //  $\Sigma \cup \mathbb{Q}_0^+$  is recursive;
12    Guess  $a \in \Sigma \cup \mathbb{Q}_0^+$ ;
13    if ( $\text{true} \sqcap \text{false}$ ) then
14      //  $\rightarrow$  is recursive;
15      Guess  $C'$  such that  $C \xrightarrow{a} C'$ ;
16      // Finitely many iterations by image-finiteness;
17      foreach  $D \xrightarrow{a} D'$  do
18        NON-BISIM-STEP( $C', D', n - 1$ );
19      end
20    else
21      //  $\rightarrow$  is recursive;
22      Guess  $D'$  such that  $D \xrightarrow{a} D'$ ;
23      // Finitely many iterations by image-finiteness;
24      foreach  $C \xrightarrow{a} C'$  do
25        NON-BISIM-STEP( $C', D', n - 1$ );
26      end
27    end
28  end
29 end

```

Algorithm 2: Semi-decision procedure for bisimilarity inequivalence

Nota bene: Comments within the algorithm justify that the proceeding line is terminable and/or valid.

Finally, we give an insight into formalising the above algorithm as a deterministic procedure that can be performed on a computer. Also included is a proof that the derivation of the algorithm is correct.

Proposition 5.10: Non-bisimilarity between one-clock alternating timed automata is semi-decidable. This is equivalent to showing that $\not\sim$ on $\mathcal{T}_{\mathcal{A},\mathcal{B}}$ is semi-decidable.

Proof: By Proposition 5.8 it is sufficient to prove that $\not\sim$ on $\mathcal{T}_{\mathcal{A},\mathcal{B}}^{\mathbb{Q}}$ is semi-decidable. Moreover, given that $\not\sim = \bigcup_{n \in \mathbb{N}} \not\sim_n$, we need only prove that each of the $\not\sim_n$ are semi-decidable on $\mathcal{T}_{\mathcal{A},\mathcal{B}}^{\mathbb{Q}}$, so we proceed by induction on n .

Base case $n = 0$: $\not\sim_0 = \emptyset$, hence this is trivially satisfied.

Inductive case: Assume $\not\sim_n$ is semi-decidable on $\mathcal{T}_{\mathcal{A},\mathcal{B}}^{\mathbb{Q}}$, that is, there exists an algorithm that semi-decides whether $p \not\sim_n q$. Now show that $\not\sim_{n+1}$ is semi-decidable.

We first note that Σ is finite and \mathbb{Q}_0^+ is countably infinite. Therefore we can enumerate the whole of $\Sigma \cup \mathbb{Q}_0^+$ in the following way $[\sigma_0, \sigma_1, \dots, \sigma_{n-1}, q_0, q_1, \dots]$, where $n = |\Sigma|$.

We now formalise the algorithm that semi-decides whether $C \not\sim_{n+1} D$.

The i -th stage of the algorithm will do the following:

1. Take the first i symbols from $\Sigma \cup \mathbb{Q}_0^+$ in the order prescribed in the enumeration above. Let this collection of symbols be denoted by the set P .
2. For each configuration C' reachable from C by reading a symbol in P , consider whether $C' \not\sim_n D'$ by running the semi-decision procedure for $\not\sim_n$ in i steps for each D' that is reachable from D by reading the same symbol. Note that this step is valid since there are only finitely many such D' by image-finiteness.
3. If there is a particular symbol in P that causes $C' \not\sim_n D'$ to accept for all valid D' within i steps, then we accept as it is the case that $C \not\sim_{n+1} D$, otherwise we re-run the algorithm after incrementing i .

This has shown that $\not\sim_{n+1}$ is semi-decidable, hence by induction $\not\sim$ is semi-decidable. \square

In the next chapter we move a step closer to proving full decidability of bisimilarity by proving that bisimilarity equivalence is semi-decidable.

Bisimilarity Is Semi-Decidable

This penultimate chapter will complete our algorithm for deciding the bisimilarity of two one-clock alternating timed automata. In the previous chapter we showed that non-bisimilarity was semi-decidable, so we now show that bisimilarity is semi-decidable. Full decidability then holds since we can run our two algorithms in parallel, halting if either one of the algorithms halts.

6.1 Congruences

In this section we define the notion of a congruence and show that bisimilarity is a congruence. Recall that the states of $\mathcal{T}_{\mathcal{A},\mathcal{B}}$ are configurations of \mathcal{A} or \mathcal{B} (i.e. sets of states of \mathcal{A} or \mathcal{B}).

Definition 6.1: Say that a relation R on $\mathcal{T}_{\mathcal{A},\mathcal{B}}$ is a *congruence* if $C R D$ and $C' R D'$ imply that $(C \cup C') R (D \cup D')$ holds. \diamond

Now we can show that the bisimilarity relation is a congruence.

Proposition 6.2: The bisimilarity relation \sim is a congruence on $\mathcal{T}_{\mathcal{A},\mathcal{B}}$.

Proof: Let C, C' be configurations of \mathcal{A} and D, D' configurations of \mathcal{B} , such that $C \sim D$ and $C' \sim D'$. Then we wish to show that $C \cup C' \sim D \cup D'$.

First define a relation R to be

$$\{(E, F) \mid \exists E', F', E'', F'' :: (E' \cup E'' = E), (F' \cup F'' = F), (E' \sim F') \text{ and } (E'' \sim F'')\}$$

Note that $(C \cup C') R (D \cup D')$. We wish to show that R is a timed bisimulation.

- First consider the timed-transitions:

$$\begin{aligned} & E R F \\ \Leftrightarrow & \exists E', F', E'', F'' :: E' \cup E'' = E \wedge F' \cup F'' = F \wedge E' \sim F' \wedge E'' \sim F'' \\ \Leftrightarrow & \forall t \in \mathbb{R}_0^+ :: (\exists E', F', E'', F'' :: E' \cup E'' = E \wedge F' \cup F'' = F \wedge \\ & (E' + t) \sim (F' + t) \wedge (E'' + t) \sim (F'' + t)) \\ \Leftrightarrow & \forall t \in \mathbb{R}_0^+ :: (\exists E', F', E'', F'' :: (E' + t) \cup (E'' + t) = (E + t) \wedge \\ & (F' + t) \cup (F'' + t) = (F + t) \wedge (E' + t) \sim (F' + t) \wedge (E'' + t) \sim (F'' + t)) \\ \Leftrightarrow & \forall t \in \mathbb{R}_0^+ :: ((E + t) R (F + t)) \end{aligned}$$

So R satisfies the timed clause of a timed bisimulation.

- Now consider the Σ -labelled transitions:

Suppose $E R F$, then there exists configurations E', F', E'', F'' such that $E = E' \cup E''$, $F = F' \cup F''$, $E' \sim F'$ and $E'' \sim F''$.

Assume that $E \xrightarrow{\sigma} G$, then $E' \xrightarrow{\sigma} G'$ and $E'' \xrightarrow{\sigma} G''$, where $G = G' \cup G''$.

Since $E' \sim F'$ and $E'' \sim F''$ there exist transitions $F' \xrightarrow{\sigma} I'$ and $F'' \xrightarrow{\sigma} I''$, and so $F \xrightarrow{\sigma} I$, where $I = I' \cup I''$.

Observe that $G' \sim I'$ and $G'' \sim I''$, hence $G R I$ by definition of R . Thus R satisfies the Σ -labelled clause of a timed bisimulation.

Therefore R is a timed bisimulation as required. \square

6.2 Bisimulation bases

Our semi-decision procedure will rely upon the fact that the bisimilarity relation can be constructed from a finite bisimulation base. More succinctly, we wish to show that bisimilarity is finitely generated. Accordingly, we now present definitions of the closure of a relation and of bisimulation bases.

Definition 6.3: Given a relation R on $\mathcal{T}_{\mathcal{A},\mathcal{B}}$, the *closure* of R , denoted by \overline{R} , is the smallest relation that contains R and is a congruence. Formally speaking

$$\overline{R} \triangleq \{(C, D) \mid C = C_1 \cup \dots \cup C_n \wedge D = D_1 \cup \dots \cup D_n \wedge (C_i, D_i) \in R\}$$

\diamond

Definition 6.4: A relation R on $\mathcal{T}_{\mathcal{A},\mathcal{B}}$ is a *bisimulation base* if $C R D$ implies

- $(\forall C' : C \xrightarrow{\sigma} C' : (\exists D' : D \xrightarrow{\sigma} D' : C' \overline{R} D'))$
- $(\forall C' : C \xrightarrow{\tau} C' : (\exists D' : D \xrightarrow{\tau} D' : C' \overline{R} D'))$

\diamond

The next proposition shows that the notion of a bisimulation base is sound: if two configurations are related by a bisimulation base, then they really are bisimilar.

Proposition 6.5: If R is a bisimulation base, then \overline{R} is a bisimulation.

Proof: We wish to show that \overline{R} is a timed bisimulation. As usual, we will consider the timed and Σ -labelled transitions separately.

- Time labelled transitions:

Suppose $E \bar{R} F$, then $E = E_1 \cup \dots \cup E_n$ and $F = F_1 \cup \dots \cup F_n$ for some E_i, F_i such that $E_i R F_i$.

Given $t \in \mathbb{R}_0^+$, we can write $E_i + t = E_{i,1} \cup \dots \cup E_{i,m_i}$ and $F_i + t = F_{i,1} \cup \dots \cup F_{i,m_i}$ such that $E_{i,j} R F_{i,j}$, because R is a bisimulation base.

Now we know $(E + t) \bar{R} (F + t)$ since

$$\begin{aligned} E + t &= \bigcup_{i=1}^n (E_i + t) = \bigcup_{i=1}^n \bigcup_{j=1}^{m_i} E_{i,j} \\ F + t &= \bigcup_{i=1}^n (F_i + t) = \bigcup_{i=1}^n \bigcup_{j=1}^{m_i} F_{i,j} \end{aligned}$$

and $E_{i,j} R F_{i,j}$.

- Σ -labelled transitions:

Suppose $E \bar{R} F$, then $E = E_1 \cup \dots \cup E_n$ and $F = F_1 \cup \dots \cup F_n$ for some E_i, F_i such that $E_i R F_i$.

Assume $E \xrightarrow{\sigma} G$, then $E_i \xrightarrow{\sigma} G_{i,1} \cup \dots \cup G_{i,m_i}$, where $G = \bigcup_{i=1}^n \bigcup_{j=1}^{m_i} G_{i,j}$.

Because R is a bisimulation base, we know that $G_{i,j} R I_{i,j}$, thus $F_i \xrightarrow{\sigma} I_{i,1} \cup \dots \cup I_{i,m_i}$. Therefore $G \xrightarrow{\sigma} I$, where $I = \bigcup_{i=1}^n \bigcup_{j=1}^{m_i} I_{i,j}$, which allows us to conclude $G \bar{R} I$ as required.

Hence \bar{R} is a timed bisimulation. □

6.3 A bisimulation base for bisimilarity

In this section we construct a bisimulation base for the bisimilarity relation. This is accomplished by closely mirroring the methodology employed by Burkart, Caucal, Moller and Steffen [10] in semi-deciding bisimilarity for basic parallel processes (BPPs). However, the pairs of configurations that we are dealing with are more complex objects than the BPPs, so we will have to considerably refine the approach to be applicable to our setting.

We first define two well-founded orders over pairs of configurations that allow us to rigorously show that the closure of our bisimulation base is equal to the bisimilarity relation.

Definition 6.6: Let (C, D) and (C', D') be pairs of configurations, then the \sqsubseteq relation on pairs of configurations is a well-founded-order¹ defined by $(C, D) \sqsubseteq (C', D')$ if, and only if, $C \subseteq C'$ and $D \subseteq D'$. *Nota bene:* $(C, D) \sqsubset (C', D')$ if, and only if, $(C, D) \sqsubseteq (C', D')$ and $C \cup D \subset C' \cup D'$. \diamond

Definition 6.7: Let (C, D) and (C', D') be pairs of configurations, then the strict preorder $<$ on pairs of configurations is a well-founded-order defined by $(C, D) < (C', D')$ if, and only if,

- $|C \cup D| < |C' \cup D'|$; or
- $|C \cup D| = |C' \cup D'|$ and $|C \cap D| < |C' \cap D'|$.

\diamond

Prior to constructing our bisimulation base, we will introduce a piece of syntactic sugar that will assist us in that process.

Definition 6.8: The generic min_{\leq} operator on sets of pairs of configurations is defined to be

$$min_{\leq}(T) \triangleq \{(C, D) \in T \mid (\nexists (C', D') \in T : (C, D) \neq (C', D') : (C', D') \leq (C, D))\}$$

\diamond

Now we begin to construct our bisimulation base by first defining a set S and showing how the bisimulation base relates to S .

Definition 6.9: The set S of pairs of configurations is defined to be the set of all bisimilar pairs modulo reflexivity.

$$S \triangleq \{(C, D) \mid C \sim D \wedge C \neq D\}$$

The set $min_{\sqsubseteq}(S)$ of pairs of configurations is defined to be the set of all minimal bisimilar pairs in S with respect to the \sqsubseteq relation. This will be our bisimulation base. \diamond

Finally, we can show that the closure of the set $min_{\sqsubseteq}(S)$ is equal to \sim , that is, $min_{\sqsubseteq}(S)$ really is a bisimulation base.

Proposition 6.10: $\overline{R} = \sim$, where $R = min_{\sqsubseteq}(S)$.

Proof: Only if direction: $\overline{R} \subseteq \sim$

By definition $R \subseteq \sim$ and \sim is a congruence, so the closure of R , written \overline{R} , is contained within \sim .

If direction: $\sim \subseteq \overline{R}$

¹ (P, \leq) is a *well-founded-order* if there is no infinite descending sequence $x_1 \geq x_2 \geq x_3 \geq \dots$.

Suppose for a contradiction that $C \sim D$ and $\neg(C \bar{R} D)$. Without loss of generality we may assume that (C, D) is a minimal counter-example with respect to $<$ (this follows since a set of configurations with the ordering relation $<$ yields a well-founded ordering).

We know, *a fortiori*, $\neg(C R D)$, since $\neg(C \bar{R} D)$ and $R \subseteq \bar{R}$. Furthermore, we know $C \neq D$, because \bar{R} is an equivalence relation and $\neg(C \bar{R} D)$, hence $(C, D) \in S$. Therefore, there exists a pair $(C', D') \in R$ such that $(C', D') \sqsubseteq (C, D)$, thus $C' \subseteq C$ and $D' \subseteq D$.

Let $C'' = C \setminus C'$ and $D'' = D \setminus D'$. Since $C' \sim D'$, we have $C \sim (C' \cup D'')$ and $D \sim (D' \cup C'')$ (*Proof: $C \sim D \Leftrightarrow C \sim (D' \cup D'') \Leftrightarrow C \sim (C' \cup D'')$, the second clause follows *mutatis mutandis**).

Now either $(C, C' \cup D'') < (C, D)$ or $(D, D' \cup C'') < (C, D)$. This follows directly from the preceding lemma (Lemma 6.11).

Suppose that $(C, C' \cup D'') < (C, D)$, then $C \bar{R} (C' \cup D'') \Leftrightarrow C \bar{R} (D' \cup D'') \Leftrightarrow C \bar{R} D$. So this case leads to a contradiction.

Instead suppose that $(D, D' \cup C'') < (C, D)$, then $D \bar{R} (D' \cup C'') \Leftrightarrow D \bar{R} (C' \cup C'') \Leftrightarrow C \bar{R} D$. This case also leads to a contradiction. Therefore we must disregard our assumption and conclude $\sim \subseteq \bar{R}$. \square

The following lemma is used as part of the preceding proposition.

Lemma 6.11: If $C = C' \dot{\cup} C''$, $D = D' \dot{\cup} D''$ and $C' \neq D'$ then either $(C, C' \cup D'') < (C, D)$ or $(D, D' \cup C'') < (C, D)$ holds.

Proof: If $|C \cup C' \cup D''| < |C \cup D|$ or $|D \cup D' \cup C''| < |C \cup D|$ then the lemma holds since the first clause of the definition of $<$ on pairs of configurations satisfies this case. So make the assumption that $|C \cup C' \cup D''| \geq |C \cup D|$ and $|D \cup D' \cup C''| \geq |C \cup D|$. It is clear to see that the \geq symbol can be replaced with $=$, because $C \cup C' \cup D'' \subseteq C \cup D$ and $D \cup D' \cup C'' \subseteq C \cup D$.

But this also implies that $D' \subseteq C$ and $C' \subseteq D$, since it must be the case that $C \cup C' \cup D'' = C \cup D$ and $D \cup D' \cup C'' = C \cup D$. Thus take $D' \subseteq C$ and $C' \subseteq D$ as our assumption. Now if $|C \cap (C' \cup D'')| < |C \cap D|$ or $|D \cap (D' \cup C'')| < |C \cap D|$, then clause 2 of the definition of $<$ is satisfied.

So again, make the assumption that $|C \cap (C' \cup D'')| = |C \cap D|$ and $|D \cap (D' \cup C'')| = |C \cap D|$. Since $D' \subseteq C$ and $C' \subseteq D$ we may conclude that $C \cap (C' \cup D'') = C \cap D$ and $D \cap (D' \cup C'') = C \cap D$. Appealing to distributivity of \cap over \cup it follows that $C' \cup (C \cap D'') = C \cap D$ and $D' \cup (D \cap C'') = C \cap D$. But now $C \cap D' \subseteq C'$ and $D \cap C' \subseteq D'$. This implies $D' \subseteq C'$ and $C' \subseteq D'$, which is equivalent to $C' = D'$ contradicting one of the premises of the lemma, hence this final case cannot occur.

Therefore the lemma holds as we have exhausted all cases. \square

We have now constructed a bisimulation base $\min_{\sqsubseteq}(S)$, although this relation is not finite. We next show using the theory of well-quasi-orders that $\min_{\sqsubseteq}(S)$ is finite up to H -equivalence, where H is the time abstraction function defined in Definition 5.2.

6.4 Well-quasi-orders

Definition 6.12: A quasi-order (P, \leq) ² is said to be a *well-quasi-order* (wqo) if for every infinite sequence p_1, p_2, p_3, \dots in P , there exists $i < j$ such that $p_i \leq p_j$. \diamond

Example 6.13:

- (\mathbb{Z}, \leq) is not a wqo; take the sequence $-1, -2, -3, \dots$
- $(\mathcal{P}(\mathbb{N}), \subseteq)$ is not a wqo; take the sequence $\{0\}, \{1\}, \{2\}, \dots$
- (\mathbb{N}, \leq) is a wqo; every infinite sequence satisfies the condition of a well-quasi-order.
- $(\mathbb{N} \times \mathbb{N}, \leq_{lex})$ is a wqo; every infinite sequence satisfies the condition of a well-quasi-order.

Definition 6.14: Let Σ be an alphabet, then Σ^* is the set of words over Σ . We define the binary subword relation \preceq over Σ^* as follows.

Given $w, w' \in \Sigma^*$ such that $w = w_1 \dots w_m$ and $w' = w'_1 \dots w'_n$ we say that $w \preceq w'$ if, and only if, $m \leq n$ and there exists a strictly increasing function $f : \{1, \dots, m\} \rightarrow \{1, \dots, n\}$, such that $w_i = w'_{f(i)}$ for each $i \in \{1, \dots, m\}$.

As an example, *Higman* \preceq *Highmountain*. \diamond

Lemma 6.15: Higman's Lemma

In the case that Σ is finite, the subword relation \preceq is a well-quasi-order.

Proof: For a proof of the lemma, please consult [11]. \square

6.5 A finite bisimulation base

The relation $min_{\sqsubseteq}(S)$ in Section 6.3 is clearly a bisimulation base, however it has not been shown that it is finite up to H -equivalence. In this section we show that $min_{\sqsubseteq}(S)$ is finite up to H -equivalence indirectly by deriving a relation that coincides directly with $min_{\sqsubseteq}(S)$ and showing that this relation is finite up to H -equivalence.

Definition 6.16: Let (C, D) and (C', D') be pairs of configurations, then we write $(C, D) \trianglelefteq (C', D')$ if, and only if, $H(C \cup D) \preceq H(C' \cup D')$. \diamond

Recall from Definition 6.9 the set S . We now show that $min_{\sqsubseteq}(S)$, our bisimulation base, is also the set of minimal elements of S with respect to the \trianglelefteq relation.

²A quasi-order (also known as a pre-order) is a partial-order without the anti-symmetric clause, so \leq is a reflexive and transitive binary relation on P .

Proposition 6.17: $\min_{\sqsubseteq}(S) = \min_{\sqsubset}(S)$

Proof: $\min_{\sqsubset}(S) \subseteq \min_{\sqsubseteq}(S)$:

Take $(C, D) \in \min_{\sqsubset}(S)$ and suppose that there exists $(C', D') \in S$ such that $(C', D') \sqsubset (C, D)$. Then $C' \subseteq C$ and $D' \subseteq D$ (with at least one of these subsets being proper) by the definition of \sqsubset . Thus it follows that $H(C' \cup D') \preceq H(C \cup D)$, which implies $(C', D') \sqsubseteq (C, D)$, contradicting the inclusion of (C, D) in $\min_{\sqsubset}(S)$. Hence, $(C, D) \in \min_{\sqsubseteq}(S)$.

$\min_{\sqsubseteq}(S) \subseteq \min_{\sqsubset}(S)$:

Take $(C, D) \in \min_{\sqsubseteq}(S)$ and assume for a contradiction that there exists $(C', D') \in S$ such that $(C', D') \sqsubset (C, D)$. Then $H(C' \cup D') \prec H(C \cup D)$. By Lemma 6.18 it follows that there exists sets C'' and D'' such that $C'' \subseteq C$ and $D'' \subseteq D$ (with at least one proper), and $H(C' \cup D') = H(C'' \cup D'')$. But then $(C'', D'') \in S$ and $(C'', D'') \sqsubset (C, D)$, contradicting the inclusion of (C, D) in $\min_{\sqsubseteq}(S)$. Hence $(C, D) \in \min_{\sqsubset}(S)$. \square

The next lemma was used as part of the preceding proposition.

Lemma 6.18: $(C, D) \sqsubset (C', D')$ if, and only if, there exists sets C'' and D'' such that $C'' \subseteq C'$, $D'' \subseteq D'$, $(C'' \cup D'') \subset (C' \cup D')$ and $H(C \cup D) = H(C'' \cup D'')$.

Proof: *Only if direction:*

Suppose $(C, D) \sqsubset (C', D')$, then $H(C \cup D) \prec H(C' \cup D')$. If $H(C \cup D) = w_1 w_2 \dots w_m$ and $H(C' \cup D') = w'_1 w'_2 \dots w'_n$, then because $w_1 \dots w_m \prec w'_1 \dots w'_n$ there exists a strictly increasing function $f : \{1 \dots m\} \rightarrow \{1 \dots n\}$, where $w_i = w'_{f(i)}$ and $m < n$.

Take $G \hat{=} \{1 \dots n\} \setminus \{f(i) | i \in \{1 \dots m\}\}$ and note that this set is non-empty, because we are dealing with the proper subword relation. Furthermore, partition $C' \cup D'$ into sets $E'_1 \dots E'_n$ such that $w'_i = H(E'_i)$.

Now we can construct the sets C'' and D'' as follows:

$$\begin{aligned} C'' &\hat{=} C' \setminus \left(\bigcup \{E'_i | i \in G\} \right) \\ D'' &\hat{=} D' \setminus \left(\bigcup \{E'_i | i \in G\} \right) \end{aligned}$$

Thus $C'' \subseteq C'$ and $D'' \subseteq D'$ as required and $(C'' \cup D'') \subset (C' \cup D')$ since G is non-empty. Moreover, $H(C \cup D) = H(C'' \cup D'')$ by construction of the sets C'' and D'' .

If direction:

This direction is trivial, and so is left as an exercise for the reader. However, note the subtlety in that $H(C \cup D)$ and $H(C' \cup D')$ must be related by the proper subword order. \square

Given the equivalence of $\min_{\sqsubseteq}(S)$ and $\min_{\sqsubset}(S)$, we now wish to show that $\min_{\sqsubset}(S)$ is finite up to H -equivalence. This follows neatly from the properties of well-quasi-orders.

Proposition 6.19: $\min_{\triangleleft}(S)$ is finite up to H -equivalence.

Proof: Let Q be the set of pairs of configurations, then (Q, \triangleleft) is a well-quasi-order according to Lemma 6.15 (Higman's Lemma). Observe that $S \subseteq Q$, now we wish to show that $\min_{\triangleleft}(S)$ is finite up to H -equivalence.

Suppose $\min_{\triangleleft}(S)$ is not finite up to H -equivalence; then there exists an infinite sequence x_1, x_2, x_3, \dots such that $H(x_i) \neq H(x_j)$ for $i \neq j$.

But since (Q, \triangleleft) is a well-quasi-order, it must be the case that x_1, x_2, x_3, \dots saturates. Therefore, there exists $p < q$ such that $x_p \triangleleft x_q$, which is equivalent to $H(x_p) \preceq H(x_q)$. However, by minimality of the x_i it must be the case that $H(x_p) = H(x_q)$, which is a contradiction.

Therefore $\min_{\triangleleft}(S)$ is finite up to H -equivalence. \square

Thus we have shown in this section that we may partition our bisimulation base $\min_{\sqsubseteq}(S)$ into a finite number of equivalence classes, by taking the quotient of $\min_{\sqsubseteq}(S)$ with respect to H -equivalence. Consequentially, our semi-decision procedure for bisimilarity need only concern itself with a single pair of configurations in each equivalence class.

6.6 Bisimilarity is semi-decidable

In this section we show that bisimilarity equivalence is semi-decidable. Recall from Chapter 5 that there is a correspondence between timed bisimilarity and rationally timed bisimilarity, so we will deal exclusively with configurations on $\mathcal{T}_{\mathcal{A}, \mathcal{B}}^{\mathbb{Q}}$.

Henceforth, let Q be the set of pairs of configurations of $\mathcal{T}_{\mathcal{A}, \mathcal{B}}^{\mathbb{Q}}$, where the first component of a pair is a configuration of \mathcal{A} and the second component is a configuration of \mathcal{B} .

Prior to presenting our semi-decision procedure for bisimilarity equivalence we introduce the concept of a skeleton, which we will make use of to simplify our code.

Definition 6.20: A *skeleton* for a bisimulation base is a relation R such that if $C R D$ and $C \xrightarrow{\sigma} C'$ then there exists a D' such that $D \xrightarrow{\sigma} D'$ with decompositions $C' = C'_1 \cup \dots \cup C'_n$ and $D' = D'_1 \cup \dots \cup D'_n$ such that (C'_j, D'_j) is H -equivalent to some pair $(C''_j, D''_j) \in R$, and similarly for time evolutions. \diamond

Proposition 6.21: There is a program to decide whether a finite relation is a skeleton for a bisimulation base.

Proof: It is not too difficult to derive a correct program for deciding whether a finite relation is a skeleton for a bisimulation base, however we choose to omit the code for concision. \square

Proposition 6.22: Bisimilarity is Semi-Decidable

We now show that bisimilarity equivalence is semi-decidable by stating a procedure to determine whether two one-clock ATAs \mathcal{A} and \mathcal{B} , with initial configurations C and D respectively, are bisimilar.

1. **Generate all finite binary relations on pairs of configurations.** Clearly the set of all pairs of configurations Q is infinite, but it is possible to enumerate all finite subsets of Q as we require them. Let the sets of the enumeration be R_1, R_2, R_3, \dots . On the generation of each R_i perform the following:
 - (a) **Check R_i is a skeleton.** Check whether the relation R_i is a skeleton for a bisimulation base. This is decidable, according to Proposition 6.21.
 - (b) **Check for initial configurations.** If R_i is a skeleton for a bisimulation base then check that the initial configuration pair (C, D) is contained within R_i . If it is, then halt in the accept state. Otherwise proceed with the next set of configurations.

Proof: Note that if \mathcal{A} and \mathcal{B} are bisimilar, with initial configurations C and D respectively, then $C \sim D$ by Definition 6.9. Hence $C \overline{R} D$, where $R = \min_{\subseteq}(S)$, by Proposition 6.10. Unfortunately R may not be finite, therefore we will be unable to systematically generate a subset of Q that is a bisimulation base and whose closure contains the pair (C, D) .

Instead, let R_H denote R quotiented by H -equivalence. Then the closure of R_H is contained within the H -closure of \sim . Furthermore, we know that R_H is finite by Proposition 6.17 and Proposition 6.19. As a result, we are able to enumerate all such relations R_H . So we now check that each R_H is a skeleton (a bisimulation base allowing for H -equivalence); this is equivalent to checking that R is a bisimulation base. Proposition 6.21 ensures that this can be performed in finite time.

Now to check that (C, D) is H -equivalent to some pair in the closure of R_H . If (C, D) is H -equivalent to some pair in the closure of R_H , then $\overline{R_H} = \overline{R_H \cup \{(C, D)\}}$ up to H -equivalence. So rather than compute the closure of R_H and then check for membership of (C, D) , instead find another skeleton that contains (C, D) .

This method is guaranteed to terminate if \mathcal{A} and \mathcal{B} are bisimilar, because we can enumerate all finite subsets of Q . \square

The above procedure may be expressed in pseudo-code as follows.

Corollary 6.23: Semi-Decision Procedure for Bisimilarity

```

1 BISIM( $C, D$ )
2   //  $\mathcal{P}(Q)$  is enumerable;
3   foreach  $R \subseteq_{finite} Q$  do
4     // IS-SKELETON( $R$ ) is decidable,  $R$  is finite;
5     if IS-SKELETON( $R$ ) and  $(C, D) \in R$  then
6       | return true
7     end
8   end
9 end

```

Algorithm 3: Semi-decision procedure for bisimilarity equivalence

Nota bene: Comments within the code justify the validity of the proceeding line. \diamond

In this chapter we have shown that bisimilarity equivalence is semi-decidable by deriving a semi-decision procedure. The final chapter presents a summary of our work and poses further questions about what we have derived.

Summary

This project shows that bisimilarity for one-clock alternating timed automata is decidable, by exhibiting two semi-decision procedures for bisimilarity and non-bisimilarity. Full bisimilarity holds by running the two algorithms in parallel, or on a more traditional computing device, interleaving the computations of both procedures.

We now show where our results lie within a subset of the plethora of differing computational models. For each model we consider the decidability of bisimilarity, language equivalence and similarity.

Model	Bisimilarity	Language equivalence	Similarity
Finite automata	P _{TIME} -complete	P _{SPACE} -complete	P _{TIME} -complete
Timed automata	EX _{PTIME} -complete	Undecidable (NPR for 1-clock automata)	EX _{PTIME} -complete
Alternating timed automata	Undecidable (Decidable for 1-clock automata)	Undecidable (NPR for 1-clock automata)	Undecidable

Table 7.1: Bisimilarity, language equivalence and simulation results [4, 12, 5, 13, 9]

Nota bene: We let NPR denote the class of languages bounded by a non-primitive recursive function.

As stated in the introduction, bisimilarity is a useful property to check for a model because it implies language equivalence, and often has a lower complexity than the latter. This makes our result a valued contribution to the varied areas of computer science that can be modelled by one-clock ATAs.

Although we have yet to give a lower bound on the complexity of our result, we believe that it is possible to exhibit a polynomial-time reduction from the reachability problem for lossy one-channel systems. This would show that the lower bound for our problem is a non-primitive recursive function.

The results we have provided here are purely theoretical, although we have given pseudo-code for how a decider would work. A natural extension to this project would be to implement a bisimilarity checker that takes two encodings of ATAs and delivers a yes/no answer as to

whether the two automata are bisimilar. This code would inevitably make use of novel data structures and optimisations that our procedures blatantly ignore.

Looking forward, we intend to investigate the decidability of similarity for one-clock alternating timed automata, which we believe is also decidable, despite the undecidability results for an arbitrary number of clocks (see Table 7.1). For this we anticipate using a similar method of finding two semi-decision procedures for similarity equivalence and inequivalence, although inescapably these procedures will have considerable differences in comparison to those for bisimilarity.

In terms of further work, it would be worthwhile looking at undecidability results for other timed computational models to see whether we can apply the all powerful H -function to them, or indeed any of the other results we have derived through this project. The primary objective of this would be to provide a decidability framework for the differing properties of various computational models, as has been started by Ouaknine, Worrell and numerous others. This would bring all of the known results together into a single coherent whole.

Bibliography

- [1] R. Alur and D. Dill, *A theory of timed automata* (Proceedings of 17th International Colloquium on Automata, Languages & Programming, 1990).
- [2] J. Ouaknine and J. Worrell, *On the decidability of Metric Temporal Logic* (Proceedings of LICS 2005, IEEE Computer Society Press, 2005).
- [3] T. A. Henzinger and V. Prabhu, *Timed alternating-time temporal logic* (Formal Modeling and Analysis of Timed Systems, 4th International Conference, FORMATS 2006, Paris, France, LCNS 4202, 2006).
- [4] S. Lasota and I. Walukiewicz, *Alternating Timed Automata* (ACM Transactions on Computational Logic, Vol. V, No. N, September 2006).
- [5] J. Ouaknine and J. Worrell, *On the language inclusion problem for timed automata: Closing a decidability gap* (Proceedings of LICS 2004, IEEE Computer Society Press, 2004).
- [6] R. Milner, *A calculus of communication systems* (LNCS 92, Springer-Verlag, 1980).
- [7] A. W. Roscoe, *The theory and practice of concurrency* (Prentice Hall, 1998).
- [8] F. Moller, S. Smolka, and J. Srba, *On the Computational Complexity of Bisimulation, Redux* (Information and Computation, Volume 194, Issue 2, Academic Press Inc, 2004).
- [9] F. Laroussinie and P. Schnoebelen, *The State Explosion Problem from Trace to Bisimulation Equivalence* (FOSSACS 2000, LCNS 1784, Springer-Verlag Berlin Heidelberg, 2000).
- [10] O. Burkart, D. Caucal, F. Moller, and B. Steffen, *Verification on Infinite Structures* (Handbook of Process Algebra, Elsevier Publishers, 2001).
- [11] G. Higman, *Ordering by divisibility in abstract algebras* (Proceedings of the London Mathematical Society, 1952).
- [12] P. A. Abdulla, J. Deneux, J. Ouaknine, and J. Worrell, *Decidability and Complexity Results for Timed Automata via Channel Machines* (Proceedings of ICALP 05, LNCS 3580, Springer, 2005).
- [13] R. Alur, *Timed Automata* (Computer Aided Verification, 1999).

Acknowledgements

I am indebted to my supervisor, James Worrell, for the assistance he has provided me with in the development of this project. I am particularly grateful for the inspiration and support offered with the proofs, most notably in Chapters 5 and 6, and for the guidance and overall strategy on how to reach our result.

Although our result is new, it is influenced by the work of Olaf Burkart, Didier Caucal, Faron Moller and Bernhard Steffen [10] on process algebras. We adapted the methods employed in that paper to generate a finite bisimulation base specific to the one-clock ATAs we were considering.

The H function was first introduced by Joël Ouaknine and James Worrell in their work on the decidability results for language inclusion and universality of timed automata [5]. It is this function that allowed us to derive semi-decision procedures for both bisimilarity and non-bisimilarity by eliding the real-values of clocks.

Many of the finiteness results in Chapter 6 relied upon the foundational work of Graham Higman¹ into the area of group theory.

I am also grateful to Gilles Bertrand and others, whose web tutorials allowed me to customise the formatting for this L^AT_EX document.

Chris Chilton
Oxford, Trinity 2008

¹19 January 1917 - 8 April 2008