Abstract

Temporal Logics are a popular formalism for specification of properties in the verification of reactive systems. They can be employed to reason about the behavior of systems with the evolution of time. For example one can specify that an event corresponding to the request of a resource is eventually followed by an event corresponding to its grant. Linear-time Temporal Logic (LTL) is one such formalism which apart from its applications in verification has strong theoretical links with other well-known logic and automata based formalisms. Though it can be used to specify a wide range of qualitative behaviors of systems, like the example above, it does not suffice to verify quantitative properties which have explicit references to the times of occurrences of events. Such properties are common in real-time systems, which have strict real-time constraints, like a request should be satisfied within 5 time units. Metric Temporal Logic (MTL) is an extension of LTL to verify real-time properties.

In this thesis we study issues related to the expressiveness of MTL. There have been two ways in which the formulas in the logic have been interpreted in the literature, which have come to be known as the "pointwise" and "continuous" semantics. In the pointwise semantics one can assert a property at only the action points corresponding to the occurrence of events where as in the continuous semantics a property can be asserted at any real time point. In this thesis we compare the expressiveness of MTL with respect to the two semantics for its various syntactic variants. We also give a characterization of the languages definable in MTL in terms of a necessary "counter-freeness" property. Finally we show that MTL is expressively complete in that there are natural logics characterizing it.

Publications based on the Thesis Work

1. Deepak D'Souza and Pavithra Prabhakar. On the expressiveness of MTL in the pointwise and continuous semantics. To appear in *Formal Methods Letters*, Software Tools for Technology Transfer.

2. Pavithra Prabhakar and Deepak D'Souza. On the expressiveness of MTL with past operators. To appear in *FORMATS'06*.

3. Fabrice Chevalier, Deepak D'Souza and Pavithra Prabhakar. On continuous timed automata with input-determined guards. *Technical Report* IISc-CSA-TR-2006-7, Indian Institute of Science, Bangalore 560012, India, June 2006.

List of Figures

Contents

Chapter 1

Introduction

This thesis focuses on the expressiveness of Metric Temporal Logic (MTL) which is a well-known temporal logic for specifying and verifying timing properties of real-time systems. In the next section we describe the main ideas behind temporal logic based verification, which is the context in which our work is set.

1.1 Temporal logic based verification

In this section we discuss the *model-checking* of *temporal properties* of *reactive systems*. A reactive system is one which interacts with its environment frequently. At any particular instant of time, it is in a certain state, which it can change upon the occurrence of an event internal or external to the system. A temporal property essentially reasons about the behavior of a system with the evolution of time. Model checking is a fully automated verification technique which, unlike testing, does an exhaustive check on all the possible behaviors of a system for violations of a desired property.

The first step in model-checking is to produce a model of the system. In order to verify qualitative temporal properties of a reactive system, it often suffices to model the system as a *finite-state transition system* (FSTS), which is a finite state automaton (FSA) where all states are specified as final. For example, Figure 1.1 gives a FSTS which models a microwave oven [11]. The labels on a state indicate the properties that the oven satisfies when in that state. For example, a state label *heat* specifies that the oven is getting heated. Properties such as "is an error state reachable" or "is every start state eventually followed by a heat state" can now be checked.

The next step is to provide a formal description of the property to be verified. The property is usually specified as a *Linear-time Temporal Logic*

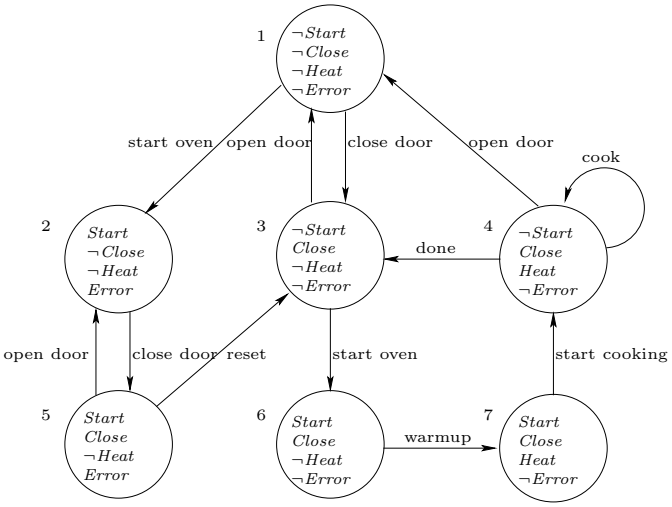


Figure 1.1: Microwave oven example.

(LTL) formula as initially proposed in [21]. LTL extends propositional logic with temporal operators, which in some sense help in reasoning about the transitions of a system. The formulas in the logic are interpreted over (finite or infinite) state sequences and can be used to specify that a property is "eventually" ($\Diamond$) true or "always" true ($\Box$) along a sequence. For example, $\Box(start \Rightarrow \Diamond heat)$ says that along any execution of the oven if it is in a state labelled *start*, then it will eventually reach a state labelled *heat*.

The third and final step is to check if the model meets the specification. This problem is known as model-checking problem and can be phrased as "Given a finite state transition system A and a temporal logic formula φ, does every behavior of the system satisfy the property specified by the formula?", or equivalently, "Is $L(A)$, the language accepted by A, contained in $L(\varphi)$, the language defined by φ?". The model-checking problem is usually solved by first constructing a FSA $A_{\neg\varphi}$ which accepts all the undesirable behaviors specified by φ, and then checking if "$L(A) \cap L(A_{\neg\varphi}) = \emptyset$?". The construction of $A_{\neg\varphi}$ can be done using the Vardi-Wolper construction [26]. Since the emptiness of FSA's can be checked algorithmically, model-checking can be done effectively.

Now we turn to the model-checking of *real-time* properties of reactive systems, which unlike the above are quantitative rather than qualitative with respect to time, in the sense that they depend not just on the order in which the events occur but also on the exact times at which they occur. An example of a real-time property is the bounded response time property which states

that every event p is followed by an event q within 5 time units.

The above property cannot be verified in the temporal framework that we discussed. Thus we view the behavior of a system in a quantitative way as a sequence of actions along with their time-stamps. We call these sequences of actions and time-stamp pairs, *timed words*. We now need to extend transition systems so that they model this quantitative behavior of reactive systems. One such formalism is the Alur-Dill timed automata [1], which extend finite automata with finite sets of clocks, and annotate the transitions with resets of clocks and guards based on them. All clocks are initially set to 0, they proceed at the same rate and measure the amount of time that has elapsed since they were started or reset. A timed word is accepted by a timed automaton if there exists a run of the automaton over the word which essentially consists of starting off in a start state and alternately, spending some time in a state, and taking a transition whose guards are satisfied by the current values of the clocks. When a transition is taken the clocks specified to be reset are set to 0.

For specifying real-time properties many extensions of LTL have been proposed [15, 4] of which *Metric Temporal Logic* (MTL) is one of the popular ones. MTL was introduced in [15] and replaces the unrestricted temporal operators of LTL by time-constrained versions, which index the temporal operators with intervals specifying the time within which the arguments need to be satisfied. For example, $\Diamond_{[1,2]}a$ says that there is an a within 1 to 2 time units from now. The above bounded response time property can be written as $\Box(p \Rightarrow \Diamond_{[0,5]}q)$.

There have been two ways in which these formulas have been interpreted over timed words in literature, which have come to be known as the "pointwise" and "continuous" semantics. In the pointwise semantics, the temporal operators quantify over a countable set of points which correspond to the times of occurrences of "actions", whereas in the continuous semantics they quantify over "all" real values. For example, consider the timed word which consists of an event a at time 1 and an event b at time 3. Then the formula $\Diamond\Diamond_{[1,1]}b$ is not true when interpreted at time 0, since there is no action point in the future from which at time distance 1 in the future there is a b. However the formula is satisfied in the continuous interpretation as the quantification is over all real values, and hence we can choose the time instant 2 for the satisfaction of the subformula $\Diamond_{[1,1]}b$.

Model checking real-time properties is harder owing to the undecidability of the "satisfiability" problem of several timed logics. The satisfiability problem of a logic is to check, given a formula, if there exists a model which satisfies it. The satisfiability problem of MTL in the continuous semantics, and in the pointwise semantics over infinite models is undecidable [3, 20].

However some amount of model-checking can still be done. In [3], it is shown that MTL is decidable over timed words in which the time stamps associated with the events are chosen from a discrete time domain, like the set of natural numbers. It is shown in [19] that the logic is decidable in the pointwise semantics over finite models, and model-checking is possible with respect to Alur-Dill timed automata. In case of the continuous interpretation a restriction of the logic which disallows singular interval indices for its temporal operators, called *Metric Interval Temporal Logic* (MITL) [2], is decidable.

1.2 Theoretical underpinnings of LTL

Till now we have discussed the practical applications of the various logic and automata based formalisms. The practical aspects are complemented by a rich theory which connects them, at least in the untimed setting. We present here some theoretical results on the expressiveness of LTL and FSA's. Informally the expressiveness of a formalism is the set of properties that can be specified using it.

It is observed that every property that can be expressed using LTL can also be expressed using FSA's [27]. In fact, the set of properties that can be expressed using FSA's is a strict superset of that using LTL. To be precise, there is a difference of one order in their logical characterizations, in that, there exists a natural first-order logic which characterizes LTL, whose extension to monadic second-order quantification characterizes the FSA's [14, 12, 6]. The existence of the natural logic characterizations establishes the expressive completeness of LTL, and is a strong endorsement of the regularity of FSA's.

There is a more direct relation between the expressiveness of FSA's and LTL. LTL corresponds to a subclass of FSA's called counter-free automata, which are the FSA's which lack a "counter" [18]. A counter is a structure in an automaton which corresponds to a sequence of states (where the first and the last states are the same) such that there is a way to proceed from one state to the next reading the same sequence of symbols. The languages accepted by counter-free automata are termed counter-free languages and those by FSA's are called regular languages. Thus a counter-free language can also be defined as a regular language such that there do not exist finite words u, v and w, where uv^iw is in the language for infinitely many i's and is not in the language for infinitely many i's.

The other issue about the expressiveness of LTL is its comparison with that of the logic obtained by adding past operators. It is shown that the extension of LTL with past temporal operators does not add any expressive

power to the logic [14, 12].

1.3 Contributions of this thesis

In this thesis we study the expressiveness of *Metric Temporal Logic* (MTL). The work in this thesis is motivated by an intension to answer the following questions, which form the counterparts in the timed setting, of the results in the last section for the untimed setting.

- Is there a characterization of the timed languages definable in MTL?
- What is the relative expressiveness of MTL with respect to its various syntactic and semantic variants?
- Is MTL expressively complete?

We give an overview of the work done towards solving the above problems and discuss some related work.

1.3.1 Counter-freeness

We prove a necessary counter-freeness property for MTL, analogous to the classical language theoretic property for classical LTL. We show that for a given MTL formula φ, there cannot exist finite timed words μ, τ and ν, such that for infinitely many i's, $\mu\tau^i\nu$ is a model of φ, and for infinitely many i's, $\mu\tau^i\nu$ is not a model of φ.

We first define a notion of "ultimate satisfiability" of a "periodic sequence" of finite timed words, where a periodic sequence consists of all words of the form $\mu\tau^i\nu$, for some μ, τ and ν. We show that given an MTL formula and a periodic sequence, there exists a point in the sequence after which either every timed word satisfies the formula, or every timed word does not satisfy the formula. Our counter-freeness result follows from this.

As an application of this result, we are able deduce that the language L_{even_b} which consists of timed words having even number of b's is not expressible by MTL (it is not expressible by LTL as it is not counter-free). These results also help us in extending some of the inexpressibility results of [5] for the case of infinite words to that of finite models.

The above results are true for the pointwise semantics and we show similar results for the continuous semantics which take into account the "granularity" of the formulas.

1.3.2 Relative expressiveness

The next part of the thesis is devoted to the comparison of the expressiveness of the logic along two axes. Along one axis, we consider the syntactic variants of MTL obtained by adding the past operators S ("since") and S_I (interval constrained "since") into the logic. These operators can refer to the past and are symmetrical to the operators U and U_I, respectively. For example, in the pointwise semantics, the formula $aS_{[1,2]}b$ when interpreted at time-point 5 in a timed word is true if there is a b in the interval $[3, 4]$ and the only actions between that point and the time point 5 are a's. We will refer to MTL with S and S_I as MTL_S and MTL_{S_I}, respectively. Along another axis, we consider the pointwise and continuous interpretations of these logics. We add the superscripts pw and c to denote the pointwise and continuous versions of the logics, respectively. We interpret the logics over both finite and infinite models.

It is seen easily that for each of these variants the continuous version is at least as expressive as the pointwise version, as one can characterize the action points in the continuous semantics, and hence mimic the pointwise interpretation. There have also been some strict containment results. In [5], it is shown that the language L_{2b}, which consists of timed words in which there are at least two occurrences of b's in the interval $(0, 1)$, is not expressible by MTL in the pointwise semantics but is expressible by MTL in the continuous semantics, and also by MTL_S in the pointwise semantics. It is also shown that the language L_{last_a}, which consists of timed words in which there is an action at time 1 which is preceded by an a, is not expressible by MTL in the continuous semantics but is expressible by MTL_S in the continuous semantics. However these results hold for the case of infinite words and do not extend readily to the case of finite words. The proofs exploit the fact that the models are infinite by using the property that the futures of two distinct points in the constructed models are the same (which is never true for any finite model).

Figure 1.2 shows the known relative expressiveness results before the work in this thesis. The solid arrows denote "strict containment", the dashed arrows represent "containment", the dashed line says that "relative expressiveness in not known", and the absence of an arrow (or transitive arrow) or a line indicates "incomparable".

The first result we show is the strict containment of MTL in the pointwise semantics in MTL in the continuous semantics over finite timed words. We do this by showing that the language L_{ni} (for "no insertions") over the alphabet $\{a, b\}$, consisting of timed words in which for every two consecutive a's the time period between them translated by one time unit does not contain any

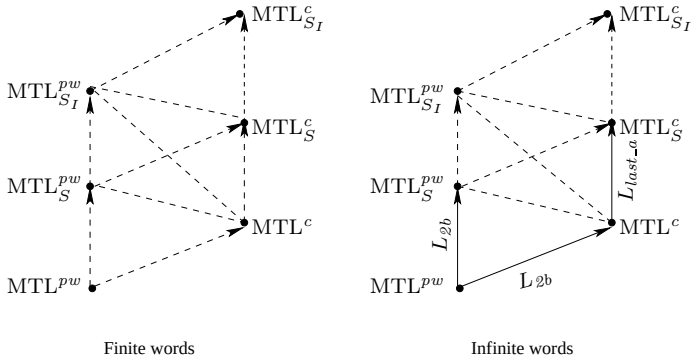


Figure 1.2: Relative expressiveness before the work in this thesis.

events, is expressible in the continuous semantics, but its expressibility in the pointwise semantics would render the logic undecidable, contradicting the decidability result in [19]. This is a novel technique which can be used to show inexpressibility results in other timed settings. For example, in [8] the technique is employed to deduce that Alur-Dill timed automata and 1-clock alternating timed automata form orthogonal classes of timed languages.

Further we also obtain the strict containment for finite words by extending the results in [5] using the notion of "ultimate satisfiability". We show that L_{2b} and L_{last_a} are not expressible by MTL in the pointwise and continuous semantics, respectively, thereby making the diagrams above, the same for both finite and infinite words.

Next we show that each of the continuous versions of the logic is strictly more expressive than its pointwise counterpart. We do so by showing that the language L_{2ins}, which consists of timed words which contain two consecutive a's such that the time period between them when translated by one time unit contains two a's, is not expressible by MTL_{S_I} (and hence not expressible by MTL_S and MTL) in the pointwise semantics, but is expressible by MTL (and hence by MTL_S and MTL_{S_I}) in the continuous semantics.

Finally we show that the language L_{em} (for "exact match"), which consists of timed words such that for every a in the interval $(0, 1)$ there is an a in the interval $(1, 2)$ at distance 1 from it, and vice versa, is expressible by MTL_{S_I} in the pointwise semantics but not by MTL_S in the pointwise semantics. This result holds for both finite and infinite words.

Figure 1.3 summarizes the relative expressiveness of the various versions of MTL after the work in this thesis. We note that it is still open whether $\mathrm{MTL}^c_{S_I}$ is strictly more expressive than MTL^c_S and whether $\mathrm{MTL}^{pw}_{S_I}$ is contained in or incomparable with MTL^c_S.

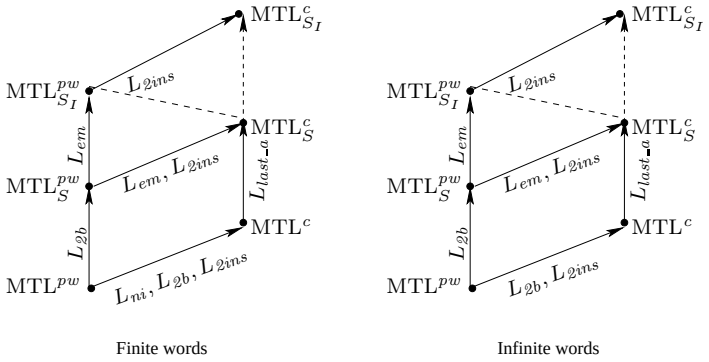


Figure 1.3: Relative expressiveness after the work in this thesis.

1.3.3 Expressive completeness

The last part of the thesis deals with showing the expressive completeness of MTL in the continuous semantics. Towards this we define a class of continuous timed automata based on "input-determined" operators, and show that MTL is expressively complete with respect to these automata for a suitably chosen set of operators.

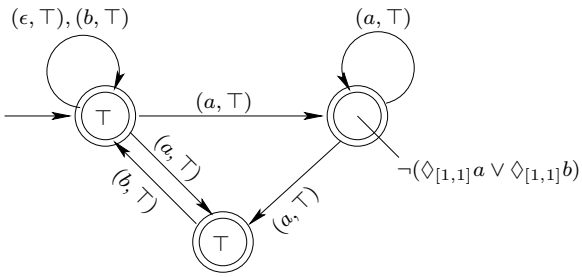
In the literature, several variants of Alur-Dill automata based on "input-determined" guards have been proposed [2, 25, 10, 7]. Unlike the explicit clock based guards of timed automata, an input-determined guard is based on a distance operator whose value is completely determined by the input timed word and a time point in it. This property leads to robust logical properties including closure under complementation which timed automata lack. A good example of an input-determined operator is the event-recording operator $\triangleleft_a$ of [2] which measures the distance to the last time an event a occurred. The "eventual" operator $\Diamond_a$ [9, 7] inspired by MTL is an input-determined operator which measures the time to "some" future occurrence of an a event.

Once again these operators can be employed in automata and logical formalisms in a pointwise or continuous manner. In the pointwise semantics, the work in [9] provides a general framework for showing determinizability, closure properties, and monadic second-order (MSO) logic characterizations, for classes of timed automata based on input-determined operators, called input-determined automata (IDA's). It also identifies natural timed temporal logics based on these operators which are expressively complete with respect to the corresponding automata classes.

In this thesis we show a similar general framework for the continuous semantics. Thus we first define an appropriate "continuous" version of these

automata called continuous input-determined automata (CIDA's) which are parameterized by a set of input-determined operators. These CIDA's extend IDA's by allowing epsilon-transitions and state invariants.

For example the diagram below gives a *CIDA* based on the operator $\Diamond$ which recognizes the language L_{ni}.



We show that *CIDA*'s are determinizable and are closed under boolean operations. They also admit logical characterizations via natural MSO logics based on the input-determined operators, and interpreted over continuous time. Further, the continuous version of the natural timed temporal logics based on these operators are shown to be expressively complete, in that they correspond to the first-order fragments of the associated MSO logics. These results generalize to the corresponding *recursive* formalisms where the input-determined operators take as arguments logical formulas or "floating" automata, as originally used in the work of [13].

As a corollary, we obtain an expressive completeness result for MTL in the continuous semantics. MTL_S can be viewed as the recursive timed temporal logic based on the operator $\Diamond$ and MTL_{S_I} that based on the operators $\Diamond$ and $\overleftarrow{\Diamond}$, and hence correspond to the first-order fragment of recursive CIDA's and the MSO based on the corresponding operators. We summarize the expressive completeness of MTL_S^c in Figure 1.4.

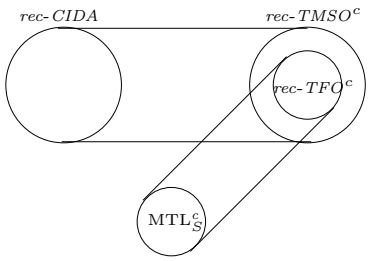


Figure 1.4: Expressive completeness of MTL_S^c

Moreover, this framework can be used as a general technique for showing such results for any class of automata and logics based on input-determined operators. In particular, the results of [13] for the class of recursive event clock automata (ECA's), (which correspond in our framework to the class of recursive CIDA's based on the operators $\triangleright$ and $\triangleleft$), pertaining to the MSO characterization via the logic MinMaxML and the expressive completeness of recursive Event Clock Temporal Logic (ECTL), can be deduced in a straight forward manner from our results.

The techniques used to prove our results are similar to [9] in that we also make use of the notion of *proper* alphabets. These alphabets help in determinizing CIDA's and showing closure properties. For the MSO characterization we use proper alphabets to translate formulas into a continuous version of Büchi's MSO logic, which preserves, in a sense, the original models of the formula. Now we need to make use of the fact that the "untiming" of continuous MSO formulas is regular in order to obtain a CIDA for the original MSO formula. We give an automata-theoretic proof of this result which was independently proved by Rabinovich in [22] using a translation to classical MSO. For the expressive completeness result concerning our timed temporal logics we factor through the well-known result of Kamp for classical LTL [14].

The technique used in [13, 24] for event clock automata is similar in that they factor through Kamp's theorem to prove their expressive completeness result. However the MSO characterization is obtained differently by showing that quantified ECTL is expressively equivalent to recursive ECA's.

1.4 Organization of the thesis

This thesis is organized as follows. In chapter 2 we start by defining the notations and preliminaries. We give the syntax of MTL, and define its pointwise and continuous interpretations. We then introduce the syntactic variants of MTL which include the past operators, and show the containment of the pointwise version in the continuous version.

In chapter 3 we show the strict containment of MTL^{pw} in MTL^{c} using a technique which exploits the decidability of a logic to extract an inexpressible language. We show that the language L_{ni} is not expressible by MTL^{pw}.

Chapters 4 and 5 give a necessary property of the timed languages definable by MTL in the pointwise and continuous semantics, respectively. In chapter 4 we prove a necessary "counter-freeness" property of MTL in the pointwise semantics, and give an application of the result in extending the proof of inexpressibility of L_{2b} by MTL^{pw} for infinite words to that of finite

words. Chapter 5 is more or less similarly organized. We first show the "counter-freeness" of MTL^c and then an application of it in extending the proof of inexpressibility of L_{last_a} to finite words.

In chapter 6 we elicit languages not expressible by MTL with past operators. We show that L_{2ins} is not expressible by $\mathrm{MTL}_{S_I}^{pw}$ over both finite and infinite words. We also sketch the proof of inexpressibility of L_{em} by MTL_{S}^{pw} which is along the lines of that for L_{2ins}.

We summarize the relative expressiveness results in chapter 7. We give a Venn diagram and a Hasse diagram which depict the results of the previous chapters.

Chapters 8 to 11 are devoted to the proof of expressive completeness of MTL with past operators in the continuous semantics. In chapter 8 we introduce the continuous input-determined automata (*CIDA*'s) and show that they are closed under boolean operations and are determinizable.

In chapter 9 we give a logical characterization for *CIDA*'s. Thus we define the continuous timed monadic second order logic (TMSO^c) based on a set of input-determined operators and show that they characterize *CIDA*'s. As an auxiliary result we obtain that the untimings of function languages definable in a continuous version of Büchi's MSO logic are regular.

In chapter 10 we define the recursive versions of *CIDA*'s and TMSO^c, called *rec-CIDA*'s and *rec-TMSO*^{c}, respectively and show that *rec-CIDA*'s are characterized by *rec-TMSO*^{c}.

In chapter 11 we show the expressive completeness of MTL_{S}^{c} and $\mathrm{MTL}_{S_I}^{c}$. We first define the timed temporal logics based on input-determined operators, and their recursive versions. We show their equivalence to first-order fragments of TMSO^c and *rec-TMSO*^{c}, respectively. We then infer the expressive completeness of MTL_{S}^{c} and $\mathrm{MTL}_{S_I}^{c}$ by showing that they are equivalent to the recursive timed temporal logics based on $\Diamond$ and $\Diamondsuit$ operators.

In chapter 12, we conclude by summarizing the results, and put forward some open problems.

Chapter 2

Metric Temporal Logic and its variants

In this chapter we formally define *Metric Temporal Logic* (MTL) and its two interpretations over timed words, namely, pointwise and continuous. We then explain the variants of the logic obtained by adding past operators. Finally we show that for each of these variants the pointwise version is contained in the continuous version.

2.1 Preliminaries

We begin with some preliminary definitions. As usual, A^* and A^ω will denote the set of finite words and infinite words over an alphabet A, respectively. For a finite word $w = a_1 \cdots a_n$ we use $|w|$ to denote the length of w (in this case n). We make use of the standard notations for regular expressions with '$\cdot$' for concatenation and '*' for Kleene closure. Given finite words u and v, we denote the concatenation of u followed by v as $u \cdot v$, or just uv. We use u^i to denote the concatenation of u with itself i times, and u^ω to denote the infinite word obtained by repeated concatenation of u. We extend these notations to subsets of A^* in the standard way.

We denote the set of non-negative and positive real numbers by $\mathbb{R}_{\geq 0}$ and $\mathbb{R}_{>0}$ respectively, the set of positive rational numbers by $\mathbb{Q}_{>0}$, and the set of non-negative integers by $\mathbb{N}$. We use $\mathcal{I}_{\mathbb{R}_{\geq 0}}$ to denote the set of intervals, where an interval is defined as a convex subset of $\mathbb{R}_{\geq 0}$. We use $\mathcal{I}_{\mathbb{Q}}$ to denote the set of intervals whose end-points are either rationals or ∞. We say that two intervals I and J are *adjacent* if $I \cap J = \emptyset$ and $I \cup J$ is an interval.

We now define finite and infinite timed words which are sequences of action and time pairs. An *infinite timed word* α over an alphabet Σ is an

element of $(\Sigma \times \mathbb{R}_{\geq 0})^\omega$ of the form $(a_1, t_1)(a_2, t_2) \cdots$ satisfying:

- (Strict-monotonicity) $t_1 < t_2 < \cdots$.
- (Progressiveness) For every $t \in \mathbb{R}_{\geq 0}$, there exists $i \in \mathbb{N}$ such that $t_i > t$.

Wherever convenient we will also denote the timed word α above as a sequence of delay and action pairs, $(d_1, a_1)(d_2, a_2) \cdots$, where for each i, $d_i = t_i - t_{i-1}$. Here and elsewhere, we use the convention that t_0 denotes the time point 0.

A *finite timed word* over Σ is an element of $(\Sigma \times \mathbb{R}_{\geq 0})^*$ which satisfies the strict monotonicity condition above. Given $\sigma = (a_1, t_1)(a_2, t_2) \cdots (a_n, t_n)$, we use $\mathit{length}(\sigma)$ to denote the time of the last action, namely t_n. The delay representation for the above finite timed word σ is $(d_1, a_1) \cdots (d_n, a_n)$ where for each i, $d_i = t_i - t_{i-1}$. Given finite timed words σ and ρ, the delay representation for the concatenation of σ followed by ρ is the concatenation of the delay representation of σ followed by the delay representation of ρ. We will use $T\Sigma^*$ for the set of all finite timed words over Σ, and $T\Sigma^\omega$ for the set of all infinite timed words over Σ.

2.2 Syntax and semantics of MTL

We start by defining the syntax of MTL. The formulas of MTL over an alphabet Σ are built up from symbols in Σ by boolean connectives and a time-constrained version of the *until* operator U. The formulas of MTL over an alphabet Σ are inductively defined as follows:

$$\varphi ::= a \mid \neg\varphi \mid (\varphi \vee \varphi) \mid (\varphi U_I \varphi),$$

where $a \in \Sigma$ and $I \in \mathcal{I}_{\mathbb{Q}}$. We also use other derived boolean connectives like $\wedge$ and $\Rightarrow$, where $\varphi_1 \wedge \varphi_2$ is $\neg(\neg\varphi_1 \vee \neg\varphi_2)$ and $\varphi_1 \Rightarrow \varphi_2$ is $\neg\varphi_1 \vee \varphi_2$. We use $\top$ for $\varphi \vee \neg\varphi$ and $\bot$ for $\neg\top$.

We first define the *pointwise* semantics for MTL over finite words. Given an MTL formula φ, a finite timed word $\sigma = (a_1, t_1)(a_2, t_2) \cdots (a_n, t_n)$ and a position $i \in \{0, \ldots, n\}$ denoting the leftmost time point 0 or one of the action points $t_1, t_2, \cdots, t_n$, the satisfaction relation $\sigma, i \models_{pw} \varphi$ (read "σ at position i satisfies φ in the pointwise semantics") is inductively defined as:

$$\begin{array}{lll}
\sigma, i \models_{pw} a & \text{iff} & a_i = a. \\
\sigma, i \models_{pw} \neg\varphi & \text{iff} & \sigma, i \not\models_{pw} \varphi. \\
\sigma, i \models_{pw} \varphi_1 \vee \varphi_2 & \text{iff} & \sigma, i \models_{pw} \varphi_1 \text{ or } \sigma, i \models_{pw} \varphi_2. \\
\sigma, i \models_{pw} \varphi_1 U_I \varphi_2 & \text{iff} & \exists j \text{ such that } i \leq j \leq |\sigma|, t_j - t_i \in I \text{ and } \sigma, j \models_{pw} \varphi_2, \\
 & & \quad \text{and } \forall k \text{ such that } i < k < j,\ \sigma, k \models_{pw} \varphi_1.
\end{array}$$

The language of finite timed words defined by an MTL formula φ in the pointwise semantics is given by $L^{pw}(\varphi) = \{\, \sigma \in T\Sigma^* \mid \sigma, 0 \models_{pw} \varphi\}$. We will use MTL^{pw} to denote the pointwise interpretation of this logic.

We define the derived operators $\Diamond_I$, $\Box_I$, U, $\Diamond$ and $\Box$ (defined in the expected manner), as follows. $\Diamond_I\varphi$, $\Box_I\varphi$, $\varphi_1 U \varphi_2$, $\Diamond\varphi$ and $\Box\varphi$ are syntactic abbreviations for $\top U_I \varphi$, $\neg\Diamond_I\neg\varphi$, $\varphi_1 U_{(0,\infty)}\varphi_2$, $\top U\varphi$ and $\neg\Diamond\neg\varphi$, respectively.

The strict until U is the standard operator used in the continuous semantics. But the usual operators used for the pointwise version are the "next" operator O and a non-strict "until" U^n, similar to classical LTL, with the semantics:

$$\begin{array}{lll} \sigma, i \models_{pw} O\varphi & \text{iff} & i < |\sigma| \text{ and } \sigma, i+1 \models_{pw} \varphi. \\ \sigma, i \models_{pw} \varphi_1 U^n_I \varphi_2 & \text{iff} & \exists j \text{ such that } i \le j \le |\sigma|, t_j - t_i \in I, \text{ and } \sigma, j \models_{pw} \varphi_2, \\ & & \text{and } \forall k \text{ such that } i \le k < j,\ \sigma, k \models_{pw} \varphi_1. \end{array}$$

We use the strict until to have a common syntax for both the interpretations. However it can be seen that this change in syntax does not affect the expressiveness of the logic in the pointwise semantics. In one direction the operators O and U^n can be expressed using the U operator as below:

$$\begin{array}{lcll} O\varphi & \equiv & \bot U_{(0,\infty)}\varphi. & \\ \varphi_1 U^n_I \varphi_2 & \equiv & \left\{ \begin{array}{ll} \varphi_1 \wedge (\varphi_1 U_I \varphi_2) & \text{if } \ 0 \notin I. \\ \varphi_2 \vee (\varphi_1 \wedge (\varphi_1 U_I \varphi_2)) & \text{if } \ 0 \in I. \end{array} \right. \end{array}$$

Let $\Diamond^n_I\varphi$ denote $\top U^n_I \varphi$, and $\Box^n_I\varphi$ denote $\neg\Diamond^n_I\neg\varphi$. We can then express the operator U_I in terms of the O and U^n_I operators, as follows:

$$\varphi_1 U_I \varphi_2 \equiv \left\{ \begin{array}{ll} \Diamond^n_I\varphi_2 \wedge \Box^n_{(0,a)}(\varphi_1 U^n_{[0,\infty)}(\varphi_1 \wedge O\varphi_2)) & \text{if } I = [a,b), a > 0. \\ \Diamond^n_I\varphi_2 \wedge \Box^n_{(0,a]}(\varphi_1 U^n_{[0,\infty)}(\varphi_1 \wedge O\varphi_2)) & \text{if } I = (a,b), a > 0. \\ \Diamond^n_I\varphi_2 \wedge (\varphi_2 \vee O(\varphi_1 U^n_{[0,\infty)}\varphi_2)) & \text{if } I = [0,b). \\ \Diamond^n_I\varphi_2 \wedge O(\varphi_1 U^n_{[0,\infty)}\varphi_2) & \text{if } I = (0,b). \end{array} \right.$$

We now give below a couple of examples of MTL formulas along with their pointwise interpretations. Consider the timed word $\sigma = (a, 1.3)(a, 2.5)(a, 3.7)$ $(b, 5.4)(c, 6.5)(b, 7.7)(b, 9)$ depicted below. We interpret the formulas at 0.

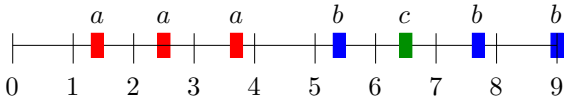


Figure 2.1: A timed word

1. $aU_{[5,6]}b$: The timed word in the above figure satisfies this formula, since b is true at some time in the interval $[5, 6]$ and a is true at all the action points before that.

2. $aU_{[6,7]}(a \vee b)$: This formula is not satisfied by the above timed word, since there is neither an occurrence of a nor that of b anywhere in the interval $[6, 7]$.

3. $\Diamond_{(0,1)}\Diamond_{[1,1,]}a$: This formula is not satisfied by the timed word since there is no action point in the interval $(0, 1)$, and hence no action point in the interval $(0, 1)$ where the subformula $\Diamond_{[1,1]}a$ can be satisfied.

We now turn to the continuous semantics. Given an MTL formula φ, a finite timed word $\sigma = (a_1, t_1)(a_2, t_2)\cdots(a_n, t_n)$ and a time $t \in \mathbb{R}_{\geq 0}$, such that $0 \leq t \leq \mathit{length}(\sigma)$, the satisfaction relation $\sigma, t \models_c \varphi$ (read "σ at time t satisfies φ in the continuous semantics") is inductively defined as follows:

$$
\begin{array}{lll}
\sigma, t \models_c a & \text{iff} & \exists i \,\text{such that}\, t_i = t \,\text{and}\, a_i = a. \\
\sigma, t \models_c \neg\varphi & \text{iff} & \sigma, t \not\models_c \varphi. \\
\sigma, t \models_c \varphi_1 \vee \varphi_2 & \text{iff} & \sigma, t \models_c \varphi_1 \,\text{or}\, \sigma, t \models_c \varphi_2. \\
\sigma, t \models_c \varphi_1 U_I \varphi_2 & \text{iff} & \exists t' \,\text{such that}\, t \leq t' \leq \mathit{length}(\sigma), t' - t \in I \,\text{and}\, \sigma, t' \models_c \varphi_2, \\
 & & \quad \text{and } \forall t'' \text{ such that } t < t'' < t',\ \sigma, t'' \models_c \varphi_1.
\end{array}
$$

The language of finite timed words defined by an MTL formula φ in the continuous semantics is given by $L^c(\varphi) = \{\, \sigma \in T\Sigma^* \mid \sigma, 0 \models_c \varphi\}$. We will use MTL^c to denote this continuous interpretation of the MTL formulas.

Below are some examples of MTL formulas and their interpretations in the continuous semantics. As before we interpret the formulas at time 0.

1. $aU_{[5,6]}b$: The timed word in Figure 2.1 does not satisfy this formula. In fact there is no model which satisfies this formula, since it requires that b is true at some point in the interval $[5, 6]$ and a is true at all the previous points (not just the action points).

   However the formula $(\neg(\bigvee_{d\in\Sigma} d) \vee a)U_{[5,6]}b$, which defines in the continuous semantics the language defined by $aU_{[5,6]}b$ in the pointwise semantics, is satisfied by the above timed word as it asks for the satisfaction of b at some point in $[5, 6]$ and the satisfaction of a at only all the previous action points.

2. $\Diamond_{(0,1)}\Diamond_{[1,1]}b$: The timed word in Figure 2.1 satisfies this formula in continuous semantics, since there is the point 0.3 in the interval $(0, 1)$ such that there is a b at distance one from this point in future. Recall that this timed word does not satisfy the formula in the pointwise semantics.

2.3 MTL with past operators

In this section we introduce some syntactic extensions of MTL. We denote by MTL_{S_I} the logic which contains in addition to U_I, a past operator S_I which is a time-constrained version of the "since" operator of LTL. Its semantics is symmetric to that of U_I and refers to the past instead of the future. We give below its semantics under both the pointwise and continuous interpretations:

$$\begin{array}{lll} \sigma, i \models_{pw} \varphi_1 S_I \varphi_2 & \text{iff} & \exists j \,\text{such that}\, 0 \leq j \leq i,\ t_i - t_j \in I \,\text{and}\, \sigma, j \models_{pw} \varphi_2, \\ & & \text{and } \forall k \text{ such that } j < k < i,\ \sigma, k \models_{pw} \varphi_1. \\ \sigma, t \models_{c} \varphi_1 S_I \varphi_2 & \text{iff} & \exists t' \,\text{such that}\, 0 \leq t' \leq t,\ t - t' \in I \,\text{and}\, \sigma, t' \models_{c} \varphi_2, \\ & & \text{and } \forall t'' \text{ such that } t' < t'' < t,\ \sigma, t'' \models_{c} \varphi_1. \end{array}$$

As before, we denote the pointwise and continuous interpretations of MTL_{S_I} as $\mathrm{MTL}_{S_I}^{pw}$ and $\mathrm{MTL}_{S_I}^{c}$, respectively. For φ in MTL_{S_I}, we use $L^{pw}(\varphi)$ and $L^{c}(\varphi)$ to denote the timed languages defined by φ in the pointwise and continuous semantics, respectively.

We define the derived operators $\Diamond\!\!\!\!-_I$, $\boxminus_I$, S, $\Diamond\!\!\!\!-$ and $\boxminus$ as follows. $\Diamond\!\!\!\!-_I\varphi$, $\boxminus_I\varphi$, $\varphi_1 S \varphi_2$, $\Diamond\!\!\!\!-\varphi$ and $\boxminus\varphi$ are syntactic abbreviations for $\top S_I \varphi$, $\neg\Diamond\!\!\!\!-_I\neg\varphi$, $\varphi_1 S_{(0,\infty)}\varphi_2$, $\top S \varphi$ and $\neg\Diamond\neg\varphi$, respectively.

Some examples of MTL_{S_I} formulas are given below:

- $\Box_{(1,2)}(a \Rightarrow \Diamond\!\!\!\!-_{[1,1]} a)$: The formula in the pointwise semantics defines the language which consists of timed words in which for every a in the interval $(1, 2)$, there is an a at distance 1 in the past.
- $\Diamond_{[2,2]}((\neg b) S a)$: Over the alphabet $\{a, b\}$, this formula in the continuous semantics accepts the timed language which consists of timed words whose last symbol in the interval $(0, 2)$ is an a.

Another extension of the logic MTL, denoted by MTL_S, is obtained by adding the derived operator S to MTL. Since this logic allows a restricted use of the S_I operator, every MTL_S formula is an MTL_{S_I} formula. We denote by MTL_S^{pw} and MTL_S^{c}, its pointwise and continuous interpretations.

2.4 Continuous as expressive as pointwise

In this section we show that MTL^{pw} is at least as expressive as MTL^{c}, MTL_S^{pw} at least as MTL_S^{c} and $\mathrm{MTL}_{S_I}^{pw}$ at least as $\mathrm{MTL}_{S_I}^{c}$. As usual by the expressiveness of a logic we mean the class of languages definable by the formulas of the logic.

We give a mapping *pw-c*, which maps an MTL_{S_I} formula φ to a formula φ' in MTL_{S_I} such that φ' in the continuous semantics defines the language defined by φ in the pointwise semantics. The formula φ_{action} is a shorthand for $\bigvee_{a\in\Sigma} a$ and captures the notion of an action point in a timed word.

$$
\begin{array}{lcl}
pw\text{-}c(a) & = & a \text{ where } a \in \Sigma. \\
pw\text{-}c(\neg\varphi_1) & = & \neg pw\text{-}c(\varphi_1). \\
pw\text{-}c(\varphi_1 \vee \varphi_2) & = & pw\text{-}c(\varphi_1) \vee pw\text{-}c(\varphi_2). \\
pw\text{-}c(\varphi_1 U_I \varphi_2) & = & (\neg\varphi_{action} \vee pw\text{-}c(\varphi_1)) U_I (\varphi_{action} \wedge pw\text{-}c(\varphi_2)). \\
pw\text{-}c(\varphi_1 S_I \varphi_2) & = & (\neg\varphi_{action} \vee pw\text{-}c(\varphi_1)) S_I (\varphi_{action} \wedge pw\text{-}c(\varphi_2)).
\end{array}
$$

Lemma 2.1 *Let φ be an MTL_{S_I} formula over an alphabet Σ. Then $L^{pw}(\varphi) = L^{c}(pw\text{-}c(\varphi))$.*

Proof Proof follows easily by induction on the structure of φ. □

The containment of MTL^{pw} in MTL^{c} and MTL_S^{pw} in MTL_S^{c}, follows from the fact that *pw-c* maps an MTL formula to an MTL formula, and an MTL_S formula to an MTL_S formula.

Theorem 2.1 MTL^{pw} *is contained in* MTL^{c}, MTL_S^{pw} *in* MTL_S^{c}, *and* $\mathrm{MTL}_{S_I}^{pw}$ *in* $\mathrm{MTL}_{S_I}^{c}$. □

We can similarly define the semantics of the above logics for infinite words. The only change would be to replace $length(\sigma)$ and $|\sigma|$ by ∞. It is not difficult to see that the above result holds for infinite words also.

Chapter 3

Decidability and inexpressibility

In the last chapter, we saw that MTL in the continuous semantics is at least as expressive as MTL in the pointwise semantics, as we can characterize the action points in the continuous semantics and mimic the pointwise interpretation. Intuitively, one would also expect the continuous version to be strictly more expressive given its ability to quantify over arbitrary time points. In this chapter we confirm this intuition for the case of finite models by exhibiting a language of finite timed words which is definable in the continuous semantics but *not* in the pointwise semantics.

The language we consider is L_{ni} (for "no insertions") over the alphabet $\{a, b\}$, which consists of finite timed words in which for any two consecutive a's the time period between them translated by one time unit contains no events. Below is a timed word in L_{ni}. This language can easily be seen to

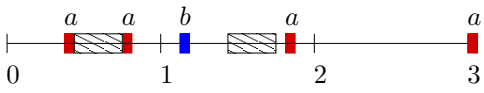


Figure 3.1: A timed word in L_{ni}.

be expressible in the continuous semantics. However we argue that it is not expressible in the pointwise semantics, since it would essentially lead to the undecidability of the pointwise version of the logic, which would contradict the result of [19]. Our aim is to prove the following theorem:

Theorem 3.1 *The timed language L_{ni} is not expressible by* MTL^{pw} *over finite timed words.* □

In the rest of the chapter we will prove this claim. We begin with a sketch of the proof in the next section.

3.1 Sketch of the proof

We will prove the inexpressibility of the language L_{ni} by a contradiction. Suppose that there exists an MTL formula φ_{ni} which describes the language L_{ni} in the pointwise semantics. We argue that the assumption that the MTL formula φ_{ni} exists, leads to the undecidability of the satisfiability problem for MTL^{pw}. We do so by giving a reduction from the *halting problem* of a 2-counter machine on an empty tape to the *satisfiability problem* of MTL^{pw}. The halting problem of a 2-counter machine is: "Given a deterministic 2-counter machine, does it have a halting computation?" and the satisfiability problem of MTL^{pw} is: "Given an MTL formula, does there exist a finite model which satisfies the formula in the pointwise semantics?". Since the halting problem of a 2-counter machine is undecidable, the reduction implies that the satisfiability problem of MTL^{pw} is also undecidable. This would contradict the result of [19], where it is shown that the satisfiability problem of MTL^{pw} is decidable. It therefore follows that L_{ni} in not expressible by MTL in the pointwise semantics.

We now informally describe the reduction of the halting problem to the satisfiability problem. The reduction is along the lines of [3, 1, 2], though the encoding is adapted to enable us to isolate the language above as being inexpressible.

Recall that a 2-counter machine M has 2 counters C and D and a sequence of n instructions, the last of which is a halt instruction. The rest of the instructions are increment (or decrement) of a counter together with a jump to another location, or a jump conditional on the value of a counter being 0. A *configuration* of M is represented by a triple $\langle i, c, d\rangle$, where $i \in \{1, \cdots, n\}$ is the location counter, and c and d are the values of the two counters C and D, respectively. A *computation* of M is a finite sequence of configurations where successive configurations respect preceding instructions. A computation is *halting* if its last configuration points to the halt instruction.

The reduction is done in two steps. Let M be a 2-counter machine as above.

1. We encode the computations of M as finite timed words over a suitable alphabet. The encoding is such that we can associate a non-empty set of timed words with every computation. Let us denote by L_M the language of timed words which correspond to halting computations.

2. We then give a formula φ_M which defines the language L_M (assuming φ_{ni} exists).

So M has a halting computation if and only if L_M is not empty, which in turn is true if and only if φ_M is satisfiable.

In the next section we describe the language L_M in detail, and in the following section we give the construction of the formula φ_M.

3.2 Formal description of L_M

In this section we describe the language L_M consisting of timed words encoding halting computations of a 2-counter machine M.

Let M be a 2-counter machine with n instructions. We first explain how the computations of M are encoded as timed words. The computations of M are encoded using timed words over the alphabet $\Sigma_M = \{b_1, b_2, \cdots, b_n, a_1, a_2\}$. We encode a configuration $\langle i, c, d\rangle$ as a sequence $b_i a_1^c a_2^d$ and a computation as a concatenation of such sequences. Moreover, we associate a unit time interval with each configuration with the j-th configuration being encoded in the interval $[j, j+1)$. In any such interval, the corresponding b occurs at time j, all the a_1's occur in the interval $[j+0.25, j+0.5)$ with the first a_1 (if any) occurring at $j+0.25$ and all the a_2's occur in the interval $[j+0.5, j+1)$ with the first a_2 (if any) occurring at $j+0.5$. We further require that the corresponding a_1's in two such consecutive intervals are a distance one apart, and we have a similar requirement for a_2. The second last requirement is a deviation from the reductions in the literature. It is used to ensure that, along with the property L_{ni}, we have no "illegal" insertions of a_1's and a_2's in succeeding configurations.

For example, consider a 2-counter machine whose instruction set is given below:

1. Increment C and jump to 3.
2. Increment D and jump to 4.
3. Increment D and jump to 2.
4. Halt.

The timed word in Figure 3.2 encodes the only halting computation of the counter machine, which is $\langle b_1, 0, 0\rangle\langle b_3, 1, 0\rangle\langle b_2, 1, 1\rangle\langle b_4, 1, 2\rangle$. Note that there are other timed words which encode the same computation, since the last a_2 could be placed anywhere in the interval $(4.5, 5)$.

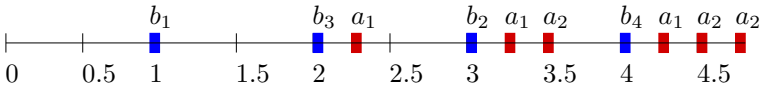


Figure 3.2: A timed word encoding halting computation

Before we move on to a detailed description of L_M we formally define the language L_{ni}. To give a language-theoretic definition of L_{ni}, let us first define the language $L_{ni}^{\Sigma,c}$ which is parameterised by an alphabet Σ and an action $c \in \Sigma$. The language $L_{ni}^{\Sigma,c}$ consists of timed words $\sigma \in T\Sigma^*$ of the form $(a_1, t_1) \cdots (a_n, t_n)$, satisfying the condition that whenever $a_i = a_{i+1} = c$ for some i, for no j does t_j belong to the interval $(t_i + 1, t_{i+1} + 1)$. The language L_{ni} can now be defined as $L_{ni}^{\{a,b\},a}$.

We now define the language L_M over Σ_M as the intersection of the languages, $L_1, L_2, \cdots, L_7$, described below. Let $\sigma = (a_1, t_1) \cdots (a_l, t_l)$ be a timed word over Σ_M and let $w = a_1 \cdots a_l$. Let $B = \{b_1, \cdots, b_n\}$.

1. L_1: σ is a timed word in L_1 if $w = b_{i_1} a_1^{c_1} a_2^{d_1} b_{i_2} a_1^{c_2} a_2^{d_2} \cdots b_{i_k} a_1^{c_k} a_2^{d_k}$, where $1 \leq i_j \leq n$, $c_j \geq 0$ and $d_j \geq 0$, for all $j \in \{1, \cdots, k\}$, and $i_1 = 1$, $c_1 = 0$ and $d_1 = 0$. This is to ensure that the untiming of the timed word is a concatenation of valid encodings of configurations, starting with the encoding of the initial configuration $\langle 1, 0, 0 \rangle$.

2. L_2: It consists of timed words in which the j-th element from B occurs at integer time j.

3. L_3: It consists of timed words in which for every b_i which occurs at time j, all a_1's in the interval $(j, j+1)$ occur in the interval $[j+0.25, j+0.5)$ and all a_2's in the interval $(j, j+1)$ occur in the interval $[j+0.5, j+1)$. The first a_1 in the interval (if any) occurs at time $j + 0.25$ and the first a_2 in the interval (if any) occurs at time $j + 0.5$.

4. L_4: It is the intersection of the two languages $L_{ni}^{\Sigma_M,a_1}$ and $L_{ni}^{\Sigma_M,a_2}$. It consists of timed words in which the time period between any two consecutive a_1's or any two consecutive a_2's when translated by 1 time unit does not contain any actions.

5. L_5: It consists of timed words in which for every b_i at time j which is eventually followed by some $b_{i'}$, there is an a_1 at time $t+1$ corresponding to every a_1 at time t in the interval $(j, j+1)$ which is immediately followed by an a_1. There is a similar condition on the times of occurrences of a_2.

6. L_6: We define L_6 as the intersection of the languages, $L_6^1, \cdots, L_6^n$, where the conditions on the timed words in L_6^i partially help to ensure that the i-th instruction is executed correctly. However the conditions on the timed words in L_6^i depend on the type of the i-th instruction. Hence we consider the following cases:

   (a) Suppose that the i-th instruction is "If $C = 0$, then jump to p, else jump to q". A timed word is in L_6^i if for every b_i at time j in the timed word,

   - if there is no a_1 in the interval $(j, j+1)$, then there is no a_1 in the interval $(j+1, j+2)$, and
   - if there is an a_1 in the interval $(j, j+1)$, then there is an a_1 at time $t+1$ which is not immediately followed by an a_1, for every a_1 at time t in the interval $(j, j+1)$ which is not immediately followed by an a_1.

   The above condition helps to ensure that the number of a_1's in the intervals, $(j, j+1)$ and $(j+1, j+2)$, is the same. We have a similar condition on a_2. Further, if there is no a_1 in the interval $(j, j+1)$, then there is a b_p at $j+1$, otherwise there is a b_q at $j+1$.

   (b) Suppose that the i-th instruction is "Increment C and jump to p". A timed word is in L_6^i if for every b_i at time j in the timed word,

   - if there is no a_1 in the interval $(j, j+1)$, then there is an a_1 at time $j+1.25$ which is not immediately followed by an a_1, and
   - if there is an a_1 in the interval $(j, j+1)$, then for every a_1 at time t in the interval $(j, j+1)$ which is not immediately followed by an a_1, there is an a_1 at time $t+1$ which is immediately followed by an a_1, which in turn is not immediately followed by an a_1.

   Along with this, we need to ensure that the number of a_2's in the intervals, $(j, j+1)$ and $(j+1, j+2)$, is the same. This can be done as in the previous case. We also require that there is a b_p at time $j+1$.

   (c) Suppose that the i-th instruction is "Decrement C and jump to p". A timed word is in L_6^i if for every b_i at time j in the timed word,

   - if there is no a_1 in the interval $(j, j+1)$, then there is no a_1 in the interval $(j+1, j+2)$,

- if every a_1 in the interval $(j, j+1)$ is not followed by an a_1 immediately, then there is no a_1 in the interval $(j+1, j+2)$, and
- if there is an a_1 in the interval $(j, j+1)$ which is immediately followed by an a_1, then for every a_1 at time t in $(j, j+1)$ which is not immediately followed by an a_1, there is no a_1 in the interval $[t+1, t+1.5)$.

Along with this, we need to ensure that the number of a_2's in the intervals, $(j, j+1)$ and $(j+1, j+2)$, is the same. This can be done as in the previous case. We also require that there is a b_p at time $j+1$.

We get the corresponding languages for the operations on counter D by exchanging the roles of a_1 and a_2 in the above languages.

(d) Suppose that the i-th instruction is a halt instruction. Then L_6^i consists of timed words in which for every b_i at time j there are no symbols in the interval $[j+1, \infty)$.

7. L_7: It consists of timed words in which there is an occurrence of b_n. This ensures that every computation has a configuration which points to the halt instruction.

It is easy to see that the timed language L_M consists of exactly those timed words which correspond to the encodings of halting computations. Therefore we have the following lemma:

Lemma 3.1 *The timed language L_M is non-empty if and only if M has a halting computation.* □

3.3 Construction of φ_M

In this section we give the formula φ_M which defines the language L_M assuming that there exists a formula which defines L_{ni}.

Let φ_{ni} be an MTL formula over $\{a, b\}$ which describes the language L_{ni} in the pointwise semantics. Then the formula $\varphi_{ni}[c/a, (\bigvee_{d \in \Sigma - \{c\}} d)/b]$ obtained from φ_{ni} by replacing every occurrence of a by c, and b by $\bigvee_{d \in \Sigma - \{c\}} d$, can be seen to define the language $L_{ni}^{\Sigma, c}$. Let us call this formula $\varphi_{ni}^{\Sigma, c}$.

We now proceed to give the formula φ_M over Σ_M which describes the language L_M. We use B as a shorthand for $\bigvee_{i \in \{1, \cdots, n\}} b_i$. As usual, φ_{action} is

a shorthand for $\bigvee_{a\in\Sigma} a$. The formula φ_M is the conjunction of the formulas, $\varphi_1, \cdots, \varphi_7$, where each φ_i corresponds to the language L_i.

- φ_1:
$$\Box(a_2 \Rightarrow \neg O a_1) \wedge O(b_1 \wedge \neg O a_1 \wedge \neg O a_2).$$

- φ_2:
$$\Diamond_{[1,1]} B \wedge \Box_{(0,1)} \neg B \wedge \Box((B \wedge \Diamond_{(0,\infty)} B) \Rightarrow (\Diamond_{[1,1]} B \wedge \Box_{(0,1)} \neg B)).$$

- φ_3:
$$\Box(B \Rightarrow (\Box_{[0,0.25)} \neg(a_1 \vee a_2) \wedge \Box_{[0.25,0.5)} \neg a_2 \wedge \Box_{[0.5,1)} \neg a_1 \\ \wedge(\Diamond_{(0,1)} a_1 \Rightarrow \Diamond_{[0.25,0.25]} a_1) \wedge (\Diamond_{(0,1)} a_2 \Rightarrow \Diamond_{[0.5,0.5]} a_2))).$$

- φ_4:
$$\varphi_{ni}^{\Sigma_M, a_1} \wedge \varphi_{ni}^{\Sigma_M, a_2}.$$

- φ_5:
$$\Box((B \wedge \Diamond_{(0,\infty)} B) \Rightarrow (\Box_{(0,1)}((a_1 \wedge O a_1) \Rightarrow (\Diamond_{[1,1]} a_1)) \wedge \\ \Box_{(0,1)}((a_2 \wedge O a_2) \Rightarrow (\Diamond_{[1,1]} a_2)))).$$

- φ_6:
$$\varphi_6 = \bigwedge_{i\in\{1,\cdots,n\}} \varphi_6^i,$$

  where φ_6^i defines the language L_6^i.

  Let $\psi_{a_1} = (\Box_{(0,1)} \neg a_1 \Rightarrow \Box_{(1,2)} \neg a_1) \wedge \Box_{(0,1)}((a_1 \wedge \neg O a_1) \Rightarrow (\Diamond_{[1,1]}(a_1 \wedge \neg O a_1)))$ and $\psi_{a_2} = (\Box_{(0,1)} \neg a_2 \Rightarrow \Box_{(1,2)} \neg a_2) \wedge \Box_{(0,1)}((a_2 \wedge \neg O a_2) \Rightarrow (\Diamond_{[1,1]}(a_2 \wedge \neg O a_2)))$.

  ψ_{a_1} along with φ_5 verifies that the number of a_1's in two consecutive encodings of configurations is the same. ψ_{a_2} along with φ_5 verifies the number of a_2's in two consecutive encodings of configurations is the same.

  The construction of the formula φ_6^i depends on the i-th instruction:

  - Suppose that the i-th instruction is: "If $C = 0$, then jump to p, otherwise jump to q". Then φ_6^i is given by:
$$\Box(b_i \Rightarrow (\psi_{a_1} \wedge \psi_{a_2} \wedge (\Box_{(0,1)} \neg a_1 \Rightarrow \Diamond_{[1,1]} b_p) \wedge (\Diamond_{(0,1)} a_1 \Rightarrow \Diamond_{[1,1]} b_q))).$$

– Suppose that the i-th instruction is: "Increment C and jump to p". Then φ_6^i is given by:

$$\Box(b_i \Rightarrow ((\Box_{(0,1)}\neg a_1 \Rightarrow \Diamond_{[1.25,1.25]}(a_1 \wedge \neg O a_1)) \wedge \\ \Box_{(0,1)}((a_1 \wedge \neg O a_1) \Rightarrow \Diamond_{[1,1]}(a_1 \wedge O a_1 \wedge \neg O O a_1)) \wedge \psi_{a_2} \wedge \Diamond_{[1,1]} b_p)).$$

– Suppose that the i-th instruction is: "Decrement C and jump to p". Then φ_6^i is given by:

$$\Box(b_i \Rightarrow ((\Box_{(0,1)}\neg a_1 \Rightarrow \Box_{(1,2)}\neg a_1) \\ \wedge(\Box_{(0,1)}(a_1 \Rightarrow \neg O a_1) \Rightarrow \Box_{(1,2)}\neg a_1) \\ \wedge(\Diamond_{(0,1)}(a_1 \wedge O a_1) \Rightarrow \Box_{(0,1)}((a_1 \wedge \neg O a_1) \Rightarrow \Box_{[1,1.5)}\neg a_1)) \\ \wedge \psi_{a_2} \wedge \Diamond_{[1,1]} b_p).$$

Similarly, we can write formulas for the above operations on counter D by exchanging a_1's and a_2's in the above formulas.

– Suppose that the i-th instruction is a halt instruction.

$$\Box(b_i \Rightarrow \Box_{[1,\infty)}\neg \varphi_{action}).$$

- $\varphi_7 : \Diamond b_n$.

It is not difficult to verify that each formula φ_i defines exactly the language L_i. Therefore $L^{pw}(\varphi_M) = L_M$. Hence we have the following lemma:

Lemma 3.2 *$L^{pw}(\varphi_M)$ is empty if and only if L_M is empty.* □

3.4 Putting it all together

Now we can give a turing machine which maps an instance of the halting problem to that of the satisfiability problem. It takes as input a 2-counter machine M, guesses the formula φ_{ni}, and outputs φ_M using the construction of the last section. From Lemma 3.1 and Lemma 3.2 it follows that M has a halting computation exactly when φ_M is satisfiable. This implies the undecidability of the satisfiability problem of MTL^{pw} which contradicts the result of [19]. Hence our assumption that φ_{ni} exists is incorrect. This proves Theorem 3.1.

But L_{ni} is expressible by the following formula in the continuous semantics.

$$\neg\Diamond(a \wedge \neg\varphi_{action} U(\neg\varphi_{action} \wedge \Diamond_{[1,1]}(a \vee b) \wedge (\neg\varphi_{action} U a))),$$

Therefore we have the following strict containment result.

Theorem 3.2 MTL^{pw} *is strictly less expressive than* MTL^{c} *over finite timed words.* □

We note that we can use the above construction to give a proof of the undecidability of the satisfiability problem of MTL^{c}. We can give MTL^{c} formulas defining L_1, L_2, L_3, L_5, L_6 and L_7 by applying the function *pw-c* on φ_1, φ_2, φ_3, φ_5, φ_6 and φ_7, respectively. Also we can give MTL^{c} formula for $L_{ni}^{\Sigma,c}$ by replacing the a's in the above formula for L_{ni} by c's, and b by $\bigvee_{\Sigma-\{c\}} c$. Thus we can give a formula which defines L_4. Therefore we have the following theorem.

Theorem 3.3 *The satisfiability problem of* MTL^{c} *is undecidable over finite timed words.* □

3.5 Discussion

In this chapter we have addressed the problem of comparing the expressiveness of MTL in the pointwise and continuous semantics. We show that the continuous version is strictly more expressive than the pointwise one. This result holds for the case when models are *finite* timed words.

It is appropriate to point out here that the fact that one version of the logic is decidable while the other isn't, does not necessarily mean that one is less expressive than the other. It is crucial to be able to identify a property which is in a sense *independent* of the problem instance being translated. In fact, such a property may not exist in general. This is borne out by the fact that there are logics [17] with two equally expressive versions, one of which is decidable and the other isn't.

Apart from the above result, our technique of using decidability to show inexpressibility can be used in other timed settings. For instance, in the technical report [8], we use a similar argument to show that L_{ni} is not expressible by 1-clock alternating timed automata (1-clock ATA's). We argue that the expressibility would contradict the decidability of their emptiness problem [19, 16]. Here the technique is used to show the orthogonality of the class of timed languages accepted by 1-clock ATA's and Alur-Dill timed automata.

However the technique used in this chapter is not applicable to differentiate the two semantics when we look at infinite words, since both versions of the logic are undecidable over infinite words [3, 20]. Moreover it would be interesting to have an expressiveness result that arises out of a characterization of the languages definable in the two semantics.

Chapter 4

Counter-freeness of MTL^{pw}

In this chapter, we elicit a necessary counter-freeness property of the timed languages over finite words expressible in MTL^{pw}. We define a notion of satisfiability on a sequence of finite timed words, which we call "ultimate satisfiability", and then show that every MTL^{pw} formula either "ultimately satisfies" or "ultimately does not satisfy" a "periodic sequence". Also, we give an application of the above satisfiability result in carrying over an inexpressibility result for infinite models to the case of finite models.

In section 4.1, we prove the result on ultimate satisfiability. In section 4.2, we use this result to show that the timed language $L_{\mathbb{2}b}$ consisting of timed words which contain two b's in the interval $(0, 1)$ is not expressible by MTL^{pw} over finite timed words.

4.1 Ultimate satisfiability of MTL^{pw}

In this section we show that an MTL formula in the pointwise semantics is either ultimately satisfied or ultimately not satisfied over a periodic sequence of timed words, leading to a "counter-freeness" property of MTL.

We first define the notion of when a formula is *ultimately satisfied* or *ultimately not satisfied* over a sequence of finite timed words. Let $\langle \sigma_i \rangle$ be a sequence of finite timed words $\sigma_0, \sigma_1, \cdots$. Given a $j \in \mathbb{N}$ and $\varphi \in \mathrm{MTL}$, we say that $\langle \sigma_i \rangle$ at j *ultimately satisfies* φ, denoted $\langle \sigma_i \rangle, j \models_{us} \varphi$, iff $\exists k \in \mathbb{N} : \forall k' \geq k, \sigma_{k'}, j \models_{pw} \varphi$. We say that $\langle \sigma_i \rangle$ at j *ultimately does not satisfy* φ, denoted $\langle \sigma_i \rangle, j \models_{un} \varphi$, iff $\exists k \in \mathbb{N} : \forall k' \geq k, \sigma_{k'}, j \models_{pw} \neg\varphi$. We refer to the least such k in either case above as the *stability point* of φ at j in $\langle \sigma_i \rangle$. The following proposition follows from the above definitions.

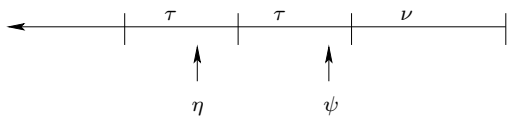
Proposition 4.1 *Let $\langle \sigma_i \rangle$ be a sequence of finite timed words. Let φ be an* MTL *formula and let $j \in \mathbb{N}$. If $\langle \sigma_i \rangle, j \models_{us} \varphi$, then it is not the case that*

$\langle\sigma_i\rangle, j \models_{un} \varphi$, *and if* $\langle\sigma_i\rangle, j \models_{un} \varphi$, *then it is not the case that* $\langle\sigma_i\rangle, j \models_{us} \varphi$. □

However the other directions of the above implications are not true, as seen in the following example. Consider the sequence $\langle\sigma_i\rangle$ given by $\sigma_0 = (1,a)$, $\sigma_1 = (1,a)(1,b)$, $\sigma_2 = (1,a)(1,b)(1,a)$, etc. Then the formula $\Diamond(a \wedge \Box\bot)$, which says that the last action of the timed word is an a, is neither ultimately satisfied nor ultimately not satisfied in $\langle\sigma_i\rangle$ at 0. But the following theorem says that a "periodic" sequence of timed words at a position j either ultimately satisfies a given MTL formula or ultimately does not satisfy it. A sequence $\langle\sigma_i\rangle$ of finite timed words is said to be *periodic* if there exist finite timed words μ, τ and ν, where $|\tau| > 0$, such that $\sigma_i = \mu\tau^i\nu$ for all $i \in \mathbb{N}$.

Theorem 4.1 *Let* $\langle\sigma_i\rangle$ *be a periodic sequence of finite timed words. Let* φ *be an* MTL *formula and let* $j \in \mathbb{N}$. *Then either* $\langle\sigma_i\rangle, j \models_{us} \varphi$ *or* $\langle\sigma_i\rangle, j \models_{un} \varphi$.

Proof Sketch The proof uses an induction on the structure of the formula. Though the boolean cases are handled in a straightforward way, the case of U_I requires careful argument, since $\eta U_I \psi$ is ultimately satisfied at a point does not imply that ψ is ultimately satisfied at some point in the interval I and η is ultimately satisfied at all the intermediate points (as expected). This was exactly the problem with the sequence we gave as a counter-example for this theorem to hold for general sequences. Here we make use of the periodicity of the words and argue that though ψ is not ultimately satisfied at some point in the future, $\eta U_I \psi$ is either ultimately satisfied or ultimately not satisfied at the current point. And the idea behind the argument is depicted in the diagram below. Let the sequence be $\langle\sigma_i\rangle$ where $\sigma_i = \mu\tau^i\nu$.



If ψ is not ultimately satisfied at some point in the future then there is a leftmost point in the above diagram where it is satisfied. If all points to the left of this satisfy η, then it can be seen that $\eta U_I \psi$ is ultimately satisfied in the sequence $\mu\tau^i\nu$. Otherwise it is seen that $\eta U_I \psi$ is ultimately not satisfied in the sequence.

Detailed Proof

Since $\langle\sigma_i\rangle$ is a periodic sequence, there exist timed words $\mu = (d_1, a_1)\cdots(d_l, a_l)$,

$\tau = (e_1, b_1) \cdots (e_m, b_m)$ and $\nu = (f_1, c_1) \cdots (f_n, c_n)$, such that $\sigma_i = \mu\tau^i\nu$. Let $\mu\tau^\omega = (a'_1, t_1)(a'_2, t_2) \cdots$. We use induction on the structure of φ to prove the theorem.

Case $\varphi = a$:

If $a'_j = a$, then clearly $\langle\sigma_i\rangle, j \models_{us} \varphi$, otherwise $\langle\sigma_i\rangle, j \models_{un} \varphi$.

Case $\varphi = \neg\psi$:

If $\langle\sigma_i\rangle, j \models_{us} \psi$, then $\langle\sigma_i\rangle, j \models_{un} \varphi$. Otherwise, by induction hypothesis, $\langle\sigma_i\rangle, j \models_{un} \psi$ and hence $\langle\sigma_i\rangle, j \models_{us} \varphi$.

Case $\varphi = \eta \vee \psi$:

Suppose $\langle\sigma_i\rangle, j \models_{us} \eta$ or $\langle\sigma_i\rangle, j \models_{us} \psi$. Let k be the maximum of the stability points of η and ψ at j in $\langle\sigma_i\rangle$. For all $k' \geq k, \sigma_{k'}, j \models_{pw} \eta$ or for all $k' \geq k, \sigma_{k'}, j \models_{pw} \psi$, and hence for all $k' \geq k, \sigma_{k'}, j \models_{pw} \eta \vee \psi$. Therefore, $\langle\sigma_i\rangle, j \models_{us} \eta \vee \psi$.

Otherwise, it is not the case that $\langle\sigma_i\rangle, j \models_{us} \eta$ and it is not the case that $\langle\sigma_i\rangle, j \models_{us} \psi$. By induction hypothesis, $\langle\sigma_i\rangle, j \models_{un} \eta$ and $\langle\sigma_i\rangle, j \models_{un} \psi$. So, $\langle\sigma_i\rangle, j \models_{us} \neg\eta$ and $\langle\sigma_i\rangle, j \models_{us} \neg\psi$. Let k be the maximum of the stability points of $\neg\eta$ and $\neg\psi$ at j in $\langle\sigma_i\rangle$. Then for all $k' \geq k, \sigma_{k'}, j \models_{pw} \neg\eta$ and for all $k' \geq k, \sigma_{k'}, j \models_{pw} \neg\psi$, and hence for all $k' \geq k, \sigma_{k'}, j \models_{pw} \neg\eta \wedge \neg\psi$. Therefore, $\langle\sigma_i\rangle, j \models_{us} \neg\eta \wedge \neg\psi$ and hence $\langle\sigma_i\rangle, j \models_{us} \neg(\eta \vee \psi)$. So, $\langle\sigma_i\rangle, j \models_{un} \eta \vee \psi$.

Case $\varphi = \eta U_I \psi$:

We consider two cases, one in which there exists $j' \geq j$ such that $t_{j'} - t_j \in I$ and $\langle\sigma_i\rangle, j' \models_{us} \psi$, and the other in which the above condition does not hold.

- Suppose there exists $j' \geq j$ such that $t_{j'} - t_j \in I$ and $\langle\sigma_i\rangle, j' \models_{us} \psi$. Let j_s be the smallest such j'.
  - Now suppose for all k such that $j < k < j_s$, $\langle\sigma_i\rangle, k \models_{us} \eta$. Let n_k be the stability point of η at k for each of the k's above, and n_{j_s} that of ψ at j_s. Let n' be the maximum of all n_k's and n_{j_s}. It can be seen that for all $n'' \geq n', \sigma_{n''}, j \models_{pw} \eta U_I \psi$. Hence $\langle\sigma_i\rangle, j \models_{us} \varphi$.
  - Otherwise there exists k such that $j < k < j_s$ and $\langle\sigma_i\rangle, k \models_{un} \eta$. Let m_k be the stability point of η at k. For each k' such that $j < k' < j_s$ and $t_{k'} - t_j \in I$, $\langle\sigma_i\rangle, k' \models_{un} \psi$ (because we chose j_s to be the smallest). Let $n_{k'}$ be the stability point of each k' above. Take n' to be the maximum of m_k and the $n_{k'}$'s. It can now be seen that for all $n'' \geq n'$, $\sigma_{n''}, j \not\models_{pw} \eta U_I \psi$. Hence $\langle\sigma_i\rangle, j \models_{un} \varphi$.
- Now turning to the second case, suppose that for all $j' \geq j$ such that $t_{j'} - t_j \in I$, it is not the case that $\langle\sigma_i\rangle, j' \models_{us} \psi$. Then by induction hypothesis, we have $\langle\sigma_i\rangle, j' \models_{un} \psi$, for all $j' : j' \geq j$ and $t_{j'} - t_j \in I$.

– Suppose I is bounded. If there is no j' such that $t_{j'} - t_j \in I$, then it is easy to see that $\langle \sigma_i \rangle, j \models_{un} \eta U_I \psi$. Otherwise, since I is bounded, there exist a finite number of j''s which satisfy $t_{j'} - t_j \in I$ and $\langle \sigma_i \rangle, j' \models_{un} \psi$. Let $n_{j'}$ be the stability point of ψ at each of these j''s. Take n' to be the maximum of all the $n_{j'}$'s. It can then be seen that for all $n'' \geq n'$, $\sigma_{n''}, j \not\models_{pw} \eta U_I \psi$. Hence $\langle \sigma_i \rangle, j \models_{un} \varphi$.

– Suppose I is unbounded. Let $S = \{s_1, s_2, \cdots, s_m\}$ be the suffixes of τ in the order of decreasing length. Thus $s_i = (e_i, b_i) \cdots (e_m, b_m)$. Let $W = \{w_1, w_2, \cdots, w_n\}$ be the suffixes of ν in the order of decreasing length. Let $X = W \cup (S \cdot \tau^* \cdot \nu)$. (We note that we can arrange the timed words in X in the increasing order of length such that the difference in lengths of the adjacent words in this sequence is one and that the succeeding string in the sequence is a prefix of the present. The sequence is $w_n, w_{n-1}, \cdots, w_1, s_m\nu, s_{m-1}\nu, \cdots, s_1\nu$, $s_m\tau\nu, \cdots, s_1\tau\nu, s_m\tau^2\nu, \cdots, s_1\tau^2\nu$, and so on.)

  We now claim that ψ is satisfied at 1 for only finitely many timed words from X. Otherwise ψ is satisfied at 1 by all timed words in $W \cup S \cdot \tau^* \cdot \nu$ whose length is greater than, say l' and hence by all timed words whose length is greater than l' in $s_i \cdot \tau^* \cdot \nu$ for some i. It is now easy to see that $\langle \sigma_i \rangle, l + i \models_{us} \psi$ (recall that l is the length of μ), and therefore $\langle \sigma_i \rangle, l + i + cm \models_{us} \psi$, for every $c \in \mathbb{N}$ (recall that m is the length of τ). Since I is unbounded there exists $j' \geq j$ such that $t_{j'} - t_j \in I$ and $\langle \sigma_i \rangle, j' \models_{us} \psi$. This is a contradiction to our assumption that $\langle \sigma_i \rangle, j' \models_{un} \psi$, for all $j' : j' \geq j$ and $t_{j'} - t_j \in I$.

  Further, for every $j'' > j$ such that $t_{j''} - t_i \in I$ and $j'' < |\mu|$, $\langle \sigma_i \rangle, j'' \models_{un} \psi$ (by the assumption of the present case). Let $n_{j''}$ be the stability point of the j'''s (which are finite in number). Let n_m be the maximum of $n_{j''}$'s.

  Suppose there is no timed word in X which satisfies ψ at 1. It can be seen that for all $n'' \geq n_m$, $\sigma_{n''}, j \not\models_{pw} \eta U_I \psi$ as we will not be able to find a point in I for the satisfaction of ψ. Hence $\langle \sigma_i \rangle, j \models_{un} \eta U_I \psi$.

  We now consider the case where there exists a timed word in X which satisfies ψ at 1. Since we proved that such timed words are finite in number, let l' be the length of the largest of them.

  Suppose that there exists a timed word in X whose length is greater than l' and which does not satisfy η at 1. Let the length of one such timed word be l''. Let n' be a number which is greater than or equal to the n_m, and which satisfies $|\sigma_{n'}| > max(j, |\mu|) + l''$.

Now for all $n'' \geq n'$, $\sigma_{n''}, j \not\models_{pw} \eta U_I \psi$ since the smallest $j' \geq j$ where ψ is satisfied is $|\sigma_{n''}| - l'$ but before that there is the point $|\sigma_{n''}| - l''$ where η is not satisfied. Hence $\langle \sigma_i \rangle, j \models_{un} \varphi$.

Suppose that all timed words in X whose length is greater than l' satisfy η at 1. Now if there exists $j < k \leq |\mu|$ such that $\langle \sigma_i \rangle, k \models_{un} \eta$, then let n' be such that it is larger than n_m and the stability point of η at k. It can be seen that for all $n'' \geq n'$, $\sigma_{n''}, j \not\models_{pw} \eta U_I \psi$. Hence, $\langle \sigma_i \rangle, j \models_{un} \varphi$. Otherwise for every $j < k \leq |\mu|$, $\langle \sigma_i \rangle, k \models_{us} \eta$. Take n' to be greater than the maximum of the stability point of η at k's and such that $|\sigma_{n'}| > j + n_I + l'$, where n_I is such that $t_{j+n_I} - t_j \in I$. We can now see that for all $n'' \geq n'$, $\sigma_{n''}, j \models_{pw} \eta U_I \psi$ and hence $\langle \sigma_i \rangle, j \models_{us} \varphi$.

□

It is well known that linear-time temporal logic (LTL) and counter-free languages [14, 18] are expressively equivalent. We recall that a *counter* in a deterministic finite automaton is a finite sequence of states $q_0 q_1 \cdots q_n$ such that $n > 1$, $q_0 = q_n$ and there exists a non-empty finite word v such that every q_i on reading v reaches q_{i+1} for $i = 0, \cdots, n-1$. A counter-free language is a regular language whose minimal DFA does not contain any counters. It is not difficult to see that the following is an equivalent characterization of counter-free languages. A regular language L is a counter-free language iff there do not exist finite words u, v and w, where $|v| > 0$, such that $uv^i w \in L$ for infinitely many i's and $uv^i w \notin L$ for infinitely many i's.

We show a similar necessary property for timed languages defined by MTL^{pw} formulas. Let us call a timed language L *counter-free* if there do not exist finite timed words μ, τ and ν, where $|\tau| > 0$, such that $\mu \tau^i \nu \in L$ for infinitely many i's and $\mu \tau^i \nu \notin L$ for infinitely many i's. The following theorem follows from the ultimate satisfiability result for MTL^{pw}.

Theorem 4.2 *Every timed language of finite words definable in* MTL^{pw} *is counter-free.*

Proof Suppose that a timed language L is definable in MTL^{pw} by a formula φ, but is not counter-free. Then there exist finite timed words μ, τ and ν, where $|\tau| > 0$, such that $\mu \tau^i \nu \in L$ for infinitely many i's and $\mu \tau^i \nu \notin L$ for infinitely many i's. The periodic sequence $\langle \sigma_i \rangle$ where $\sigma_i = \mu \tau^i \nu$, is neither ultimately satisfied by φ nor ultimately not satisfied by it which is a contradiction to Theorem 4.1. □

The above theorem, for example, implies that the language L_{even_b}, which consists of timed words in which the number of b's is even, is not expressible in MTL^{pw}. Taking $\mu = \nu = \epsilon$ and $\tau = (1, b)$ implies that L_{even_b} is not counter-free. By Theorem 4.2, L_{even_b} is not definable in MTL^{pw}.

4.2 Inexpressibility of L_{2b} by MTL^{pw}

In this section we show that the language L_{2b}, consisting of timed words over the alphabet $\Sigma = \{b\}$ which contain two b's in the interval $(0, 1)$, is not definable in MTL^{pw} over finite models. We extend a proof of the same over infinite words by using the ideas and results from the previous section.

We first present a simplified version of the proof in [5] which shows that a language similar to L_{2b} is not in MTL^{pw} over infinite timed words. We define L_{2b} formally for infinite words as, $L_{2b} = \{\alpha \in T\Sigma^\omega \mid \alpha = (a_1, t_1)(a_2, t_2)\cdots$ such that $\exists i, j \in \mathbb{N} : 0 < t_i < t_j < 1, a_i = a_j = b\}$.

We briefly explain the idea of the proof. Suppose that we want to show that a language L is not expressible by MTL^{pw}. We enumerate two families of models parameterized by a rational number p, such that one is contained in L where as the other is disjoint from it, as seen in Figure 4.1. We then show that

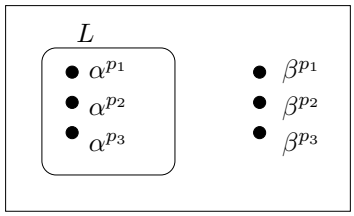


Figure 4.1: Idea of the proof of inexpressibility of L_{2b}.

no MTL formula with "granularity" p distinguishes between the two models corresponding to p in either of the families, implying the inexpressibility.

Let $p \in \mathbb{Q}_{>0}$ such that $p = 1/k$ for some $k \in \mathbb{N}$. The two models α^p and β^p are constructed as follows and are depicted in Figure 4.2. $\alpha^p = (1-3p/4, b)(p/2, b)(p/2, b)\cdots$ and $\beta^p = (1-p/4, b)(p/2, b)(p/2, b)\cdots$ in their delay representations. Clearly, $\alpha^p \in L_{2b}$ and $\beta^p \notin L_{2b}$.

We say that an MTL formula φ has a *granularity* q if all the end-points of the intervals occurring in the formula are either integral multiples of q, or ∞. We denote by MTL(q) the set of MTL formulas having granularity q. Note that a formula in MTL(q) is also in MTL$(q/2)$.

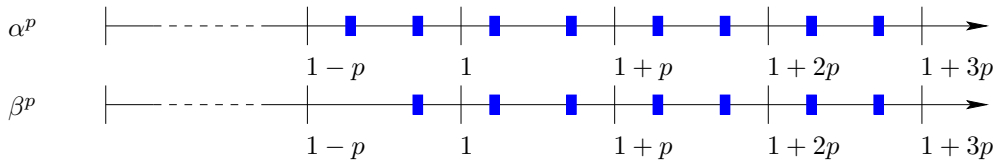


Figure 4.2: Models for showing inexpressibility of L_{2b}.

We show below that no MTL(p) formula can distinguish between α^p and β^p. The following proposition will be helpful in the proof.

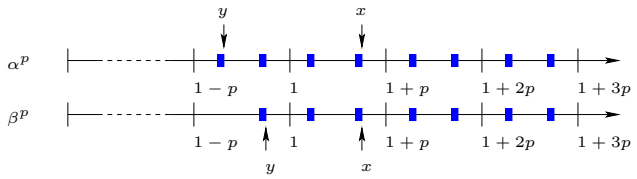
Proposition 4.2 *Let $\varphi \in$ MTL and let $i, j \in \mathbb{N}$. Then for every $i > 0$ and $j > 0$,*

1. *$\alpha^p, i \models_{pw} \varphi$ iff $\alpha^p, j \models_{pw} \varphi$,*
2. *$\beta^p, i \models_{pw} \varphi$ iff $\beta^p, j \models_{pw} \varphi$ and*
3. *$\alpha^p, i \models_{pw} \varphi$ iff $\beta^p, j \models_{pw} \varphi$.*

Proof The proof follows from the fact that the satisfiability of an MTL formula at any point in a timed word depends only on the future of the timed word with respect to that point. □

Lemma 4.1 *Let $\varphi \in$ MTL(p). Then $\alpha^p, 0 \models_{pw} \varphi$ iff $\beta^p, 0 \models_{pw} \varphi$.*

Proof Sketch We intend to argue that no MTL formula with granularity p distinguishes the two models at time 0. The only formulas which seem to be able to distinguish them are of the form $\eta U_I \psi$. If $\eta U_I \psi$ is satisfied at 0 in the first model, then ψ is satisfied at some point in the future and η is satisfied at all points between the two. If the point is not the first point then we can choose a point in the second model such that the times associated with the two points are the same, for satisfying ψ as shown in the diagram below by the arrows labelled x. Note that the futures of the models at these two points are the same. If it is the first point then the corresponding point in the second model would be the first point as shown by the arrows labelled y.



Then it is easy to see that $\eta U_I \psi$ is true at 0 in the second model.

Detailed Proof

Let $\alpha^p = (b, t_1)(b, t_2)\cdots$, where $t_1 = 1 - 3p/4$ and $t_i = t_1 + (i-1)p/2$ for $i > 1$, and $\beta^p = (b, t'_1)(b, t'_2)\cdots$, where $t'_1 = 1 - p/4$ and $t'_i = t'_1 + (i-1)p/2$ for $i > 1$. We note that $t_{i+1} = t'_i$ for $i > 0$.

Proof is by an induction on the structure of φ. The lemma is trivially true for the atomic case and boolean combinations of formulas. Let us consider the case when $\varphi = \eta U_I \psi$. We can assume that $0 \notin I$.

($\Rightarrow$)

$\alpha^p, 0 \models_{pw} \eta U_I \psi \Rightarrow \exists i \geq 0 : t_i \in I, \alpha^p, i \models_{pw} \psi$ and $\forall j : 0 < j < i, \alpha^p, j \models_{pw} \eta$.

Case $i > 1$:

$\exists i - 1 \geq 0 : t'_{i-1} \in I$(since $t_i = t'_{i-1}$), $\beta^p, i-1 \models_{pw} \psi$ (from Proposition 4.2), and $\forall j : 0 < j < i-1, \beta^p, j \models_{pw} \eta$ (from Proposition 4.2 using $\alpha^p, 1 \models_{pw} \eta$). Hence $\beta^p, 0 \models_{pw} \eta U_I \psi$.

Case $i = 1$:

$\exists i \geq 0 : t'_i \in I$ (since $t_1 \in I \Rightarrow (1-p, 1) \subseteq I$), $\beta^p, i \models_{pw} \psi$ (from Proposition 4.2), and $\forall j : 0 < j < i, \beta^p, j \models_{pw} \eta$ (in fact there is no such j). Hence $\beta^p, 0 \models_{pw} \eta U_I \psi$.

($\Leftarrow$)

$\beta^p, 0 \models_{pw} \eta U_I \psi \Rightarrow \exists i \geq 0 : t'_i \in I, \beta^p, i \models_{pw} \psi$ and $\forall j : 0 < j < i, \beta^p, j \models_{pw} \eta$.

Case $i > 1$:

$\exists i+1 \geq 0 : t_{i+1} \in I$(since $t'_i = t_{i+1}$), $\alpha^p, i+1 \models_{pw} \psi$ (from Proposition 4.2) and $\forall j : 0 < j < i+1, \alpha^p, j \models_{pw} \eta$ (from Proposition 4.2 using $\beta^p, 1 \models_{pw} \eta$). Hence $\alpha^p, 0 \models_{pw} \eta U_I \psi$.

Case $i = 1$:

$\exists i \geq 0 : t_i \in I$ (since $t'_1 \in I \Rightarrow (1-p, 1) \subseteq I$), $\alpha^p, i \models_{pw} \psi$ (from Proposition 4.2) and $\forall j : 0 < j < i, \alpha^p, j \models_{pw} \eta$ (in fact there is no such j). Hence $\alpha^p, 0 \models_{pw} \eta U_I \psi$. □

Theorem 4.3 ([5]) *The timed language L_{2b} is expressible by* MTL$_S^{pw}$ *and* MTL^c*, but not by* MTL^{pw} *over infinite timed words.*

Proof Suppose that L_{2b} is definable in MTL^{pw} by a formula φ. Let $p = 1/k$, where k'/k is the product of the rational interval end-points in φ. Clearly φ is in MTL(p). But from Lemma 4.1 it follows that either both α^p and β^p are satisfied by φ or both are not satisfied by it. This is a contradiction to

the fact that α^p is in L_{2b} whereas β^p is not. Hence L_{2b} is not expressible in MTL^{pw} over infinite timed words.

However the disjunction of the following formulas expresses L_{2b} in the continuous semantics:

1. $\Diamond_{(0,0.5]} b \wedge \Diamond_{(0.5,1)} b$: Includes all timed words in which there is a b in the interval $(0, 0.5]$ and one in the interval $(0.5, 1)$.
2. $\Diamond_{(0,0.5]}(b \wedge \Diamond_{(0,0.5)} b)$: Includes all timed words in which there are two b's in the interval $(0, 0.5]$ (and some more which are in L_{2b}).
3. $\Diamond_{(0,0.5)}(\Diamond_{[0.5,0.5]} b \wedge \Diamond_{(0,0.5)} b)$: Includes all timed words in which there are two b's in the interval $(0.5, 1)$ (and some more which are in L_{2b}).

Also, the MTL$_S$ formula $\Diamond_{(0,1)}(b \wedge \Diamond b)$ defines L_{2b} in the pointwise semantics. □

We now intend to extend the proof for finite words. A natural approach might be to replace every infinite model in the two families by a finite timed word and emulate the above proof. We see that even Proposition 4.2 fails to hold, since the futures at two distinct points cannot be the same for any finite timed word. Instead, we tackle the above problem by replacing every infinite timed word by an infinite sequence of finite timed words, and then using the notion of ultimate satisfiability, which behaves over these sequences of finite words, the way satisfiability behaved for infinite words.

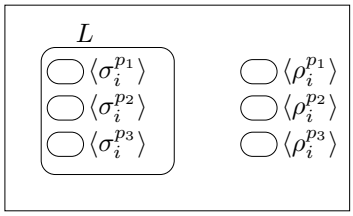


Figure 4.3: Idea behind the extension of proof to infinite words.

Let $p \in \mathbb{Q}_{>0}$ such that $p = 1/k$ for some $k \in \mathbb{N}$. We define two sequences of finite timed words, $\langle \sigma_i^p \rangle$ and $\langle \rho_i^p \rangle$, as $\sigma_i^p = \mu_1 \tau^i$ and $\rho_i^p = \mu_2 \tau^i$, where $\mu_1 = (1 - 3p/4, b)(p/2, b)$, $\mu_2 = (1 - p/4, b)$ and $\tau = (p/2, b)$. One can see that $\langle \sigma_i^p \rangle$ is contained in L_{2b} whereas $\langle \rho_i^p \rangle$ is disjoint from it.

We now show that a formula φ in MTL(p) is ultimately satisfied at 0 by $\langle \sigma_i^p \rangle$ iff it is ultimately satisfied at 0 by $\langle \rho_i^p \rangle$. We see that the claims which were true for the infinite case continue to hold for finite case with the notion of ultimate satisfiability.

Proposition 4.3 *Let $\varphi \in \mathrm{MTL}(p)$ and let $i, j \in \mathbb{N}$. Then for every $i > 0$ and $j > 0$,*

1. *$\langle \sigma_i^p \rangle, i \models_{us} \varphi$ iff $\langle \sigma_i^p \rangle, j \models_{us} \varphi$,*
2. *$\langle \rho_i^p \rangle, i \models_{us} \varphi$ iff $\langle \rho_i^p \rangle, j \models_{us} \varphi$ and*
3. *$\langle \sigma_i^p \rangle, i \models_{us} \varphi$ iff $\langle \rho_i^p \rangle, j \models_{us} \varphi$.*

Proof The proof is based on the following fact. Suppose $i < j$. Let k be the satisfiability point of φ at i in $\langle \sigma_i^p \rangle$. Then $k + j - i$ is the satisfiability point of φ at j in $\langle \sigma_i^p \rangle$. □

Lemma 4.2 *Let $\varphi \in \mathrm{MTL}(p)$. Then $\langle \sigma_i^p \rangle, 0 \models_{us} \varphi$ iff $\langle \rho_i^p \rangle, 0 \models_{us} \varphi$.*

Proof Sketch We essentially follow the proof for the infinite case. Suppose that $\eta U_I \psi$ is ultimately satisfied at 0 in one model. If it is the case that ψ is ultimately satisfied at some point in the future in interval I and η is satisfied at all the intermediate points, then we choose the corresponding points in the other model for satisfying ψ as done for the infinite case. And Proposition 4.3 is the counterpart of the fact that the futures at these two points were the same. In case where ψ is not ultimately satisfied at any point in interval I we make use of the periodicity as in the proof of Theorem 4.1 to argue that $\eta U_I \psi$ is satisfied in the other model.

Detailed Proof

Note that $\langle \sigma_i^p \rangle$ and $\langle \rho_i^p \rangle$ are periodic, since $\sigma_i^p = \mu_1 \tau^i$ and $\rho_i^p = \mu_2 \tau^i$, where $\mu_1 = (1 - 3p/4, b)(p/2, b)$, $\mu_2 = (1 - p/4, b)$ and $\tau = (p/2, b)$. Let $\mu_1 \tau^\omega = (a_1, t_1)(a_2, t_2) \cdots$ and $\mu_2 \tau^\omega = (a_1, t_1')(a_2, t_2') \cdots$. We use induction on the structure of φ for the proof.

Case φ is atomic:

Clearly $\langle \sigma_i^p \rangle, 0 \models_{us} \varphi$ iff $\langle \rho_i^p \rangle, 0 \models_{us} \varphi$.

Case $\varphi = \neg\psi$:

If $\langle \sigma_i^p \rangle, 0 \models_{us} \neg\psi$, then $\langle \sigma_i^p \rangle, 0 \models_{un} \psi$. Now it is not the case that $\langle \rho_i^p \rangle, 0 \models_{us} \psi$, since otherwise it would imply that $\langle \sigma_i^p \rangle, 0 \models_{us} \psi$ (by induction hypothesis). Hence $\langle \rho_i^p \rangle, 0 \models_{un} \psi$ (from Theorem 4.1), implying $\langle \rho_i^p \rangle, 0 \models_{us} \neg\psi$. The other direction is similar.

Case $\varphi = \eta \vee \psi$:

Suppose $\langle \sigma_i^p \rangle, 0 \models_{us} \eta \vee \psi$. Then it can not be the case that $\langle \sigma_i^p \rangle, 0 \models_{un} \eta$ and $\langle \sigma_i^p \rangle, 0 \models_{un} \psi$, since it would imply that $\langle \sigma_i^p \rangle, 0 \models_{un} \eta \vee \psi$. Hence from Theorem 4.1, $\langle \sigma_i^p \rangle, 0 \models_{us} \eta$ or $\langle \sigma_i^p \rangle, 0 \models_{us} \psi$. From the induction hypothesis, it follows that $\langle \rho_i^p \rangle, 0 \models_{us} \eta$ or $\langle \rho_i^p \rangle, 0 \models_{us} \psi$. Hence $\langle \rho_i^p \rangle, 0 \models_{us} \eta \vee \psi$. Similarly, $\langle \rho_i^p \rangle, 0 \models_{us} \eta \vee \psi$ implies $\langle \sigma_i^p \rangle, 0 \models_{us} \eta \vee \psi$.
Case $\varphi = \eta U_I \psi$: Suppose that $\langle \sigma_i^p \rangle, 0 \models_{us} \eta U_I \psi$. Then there are two cases - there exists $j > 0$ such that $t_j \in I$ and $\langle \sigma_i^p \rangle, j \models_{us} \psi$, or there is no such j.

- Suppose there exists j such that $j > 0$, $t_j \in I$ and $\langle \sigma_i^p \rangle, j \models_{us} \psi$. Let j_s be the smallest such j. Then it can not be the case that there exists k such that $0 < k < j_s$ and $\langle \sigma_i^p \rangle, k \models_{un} \eta$, since otherwise $\langle \sigma_i^p \rangle, 0 \models_{un} \eta U_I \psi$. Hence for all k such that $0 < k < j_s$, $\langle \sigma_i^p \rangle, k \models_{us} \eta$.

  Case $j_s > 1$: There exists $j_s - 1 > 0$ such that $t'_{j_s-1} \in I$ (since $t'_{j_s-1} = t_{j_s}$), $\langle \rho_i^p \rangle, j_s - 1 \models_{us} \psi$ (from Proposition 4.3) and for all k : $0 < k < j_s - 1$, $\langle \rho_i^p \rangle, k \models_{us} \eta$ (since $\langle \sigma_i^p \rangle, 1 \models_{us} \eta$ and from Proposition 4.3). Hence $\langle \rho_i^p \rangle, 0 \models_{us} \eta U_I \psi$.

  Case $j_s = 1$: There exists $j_s > 0$ such that $t'_{j_s} \in I$ (since $t_{j_s} \in I \Rightarrow (1-p, 1) \subseteq I$), $\langle \rho_i^p \rangle, j_s \models_{us} \psi$ (from Proposition 4.3) and for all $k : 0 < k < j_s$ (in fact no such k exists), $\langle \rho_i^p \rangle, k \models_{us} \eta$. Hence $\langle \rho_i^p \rangle, 0 \models_{us} \eta U_I \psi$.

- Now turning to the other case, suppose that there does not exist a $j > 0$ such that $t_j \in I$ and $\langle \sigma_i^p \rangle, j \models_{us} \psi$. Then I is not bounded, since otherwise $\langle \sigma_i^p \rangle, 0 \models_{un} \eta U_I \psi$. Further, it can not be the case that ψ is satisfied at 0 for infinitely many words from S, where $S = (0, b)(p/2, b)^*$. Because if it were, then $\langle \sigma_i^p \rangle, 1 \models_{us} \psi$ and hence $\langle \sigma_i^p \rangle, k \models_{us} \psi$ for every $k > 0$, which would contradict the non-existence of a j such that $t_j \in I$ and $\langle \sigma_i^p \rangle, j \models_{us} \psi$. Hence ψ is satisfied at 0 for finitely many timed words from S. Let $(0, b)(p/2, b)^l$ be the largest word which satisfies φ at 0. Every word in S which is longer than this, should satisfy η at 0, since otherwise $\langle \sigma_i^p \rangle, 0 \models_{un} \varphi$. But then there exists an m such that for all $m' > m$, $\rho_{m'}, 0 \models_{pw} \varphi$. Hence $\langle \rho_i^p \rangle, 0 \models_{us} \varphi$.

Note that in the case where ψ is ultimately satisfied at some point in the interval, we mimic the proof for the infinite models and in the case where ψ is not ultimately satisfied at any point in the interval, we follow the proof of Theorem 4.1. Now the proof for the other direction can be written down similarly. □

Theorem 4.4 *The timed language L_{2b} is expressible by MTL^c and MTL_S^{pw} but not by MTL^{pw} over finite timed words.*

Proof Suppose there exists a formula φ which defines L_{2b} in the pointwise semantics. Then $\varphi \in \text{MTL}(p)$ for some p. Since φ defines L_{2b} it is satisfied by all timed words in $\langle \sigma_i^p \rangle$. So φ is ultimately satisfied at 0 in $\langle \sigma_i^p \rangle$ and hence by Lemma 4.2 is ultimately satisfied at 0 in $\langle \rho_i^p \rangle$. This is a contradiction to the fact that none of the timed words in $\langle \rho_i^p \rangle$ are in L_{2b}. Therefore no MTL formula defines L_{2b} in the pointwise semantics over finite timed words.

However the formulas given in the proof of Theorem 4.3 for defining L_{2b} define the same over finite timed words in the continuous semantics. □

4.3 Conclusion

In this chapter we showed that a periodic sequence of finite timed words either ultimately satisfies an MTL formula or ultimately does not satisfy it. This implies that one cannot expect a formula to be satisfied by alternate timed words of a periodic sequence. Further, the above result gives us a necessary counter-freeness property of the timed languages definable by MTL formulas in the pointwise semantics.

We also made use of this property in showing the inexpressibility of L_{2b} by MTL^{pw}. But since it is expressible by MTL^c, this gives us an alternate proof of the strict containment of MTL^{pw} in MTL^c for finite words. However it is a totally different technique, and unlike the earlier proof is independent of the decidability of the logic.

Our results in this chapter are for the pointwise semantics. In the next chapter we address the problem of a necessary property of timed languages definable in MTL^c.

Chapter 5

Counter-freeness of MTL^c

In this chapter, we show a necessary "counter-freeness" property of timed languages definable in MTL^c. Our approach is similar to that of the previous chapter in the sense that we show the ultimate satisfiability of MTL^c and deduce the counter-freeness property from it. As before we show an application of the result in extending an inexpressibility result for the case of infinite words to that of finite words.

In section 5.1, we prove the continuous counterpart of the ultimate satisfiability for MTL. In section 5.2, we show that the language L_{last_a} consisting of finite timed words containing an action at time 1 which is immediately preceded by an action a, is not expressible in MTL^c.

5.1 Ultimate satisfiability of MTL^c

In this section, we show that an MTL^c formula with granularity p is either ultimately satisfied or ultimately not satisfied by a "p-periodic" sequence of finite timed words.

A periodic sequence of finite timed words $\langle \sigma_i \rangle$ has *period* p if there exist finite timed words μ, τ and ν, such that for each i, $\sigma_i = \mu\tau^i\nu$ and $length(\tau) = p$. Note that every periodic sequence has a unique period.

We now proceed to define the notion of ultimate satisfiability for the continuous semantics. Given a sequence $\langle \sigma_i \rangle$ of finite timed words, $t \in \mathbb{R}_{\geq 0}$ and $\varphi \in \mathrm{MTL}$, we say that $\langle \sigma_i \rangle$ at t *ultimately satisfies* φ in the continuous semantics, denoted $\langle \sigma_i \rangle, t \models^c_{us} \varphi$, iff $\exists j : \forall k \geq j, \sigma_k, t \models_c \varphi$. And we say that $\langle \sigma_i \rangle$ at t *ultimately does not satisfy* φ in the continuous semantics, denoted $\langle \sigma_i \rangle, t \models^c_{un} \varphi$, iff $\exists j : \forall k \geq j, \sigma_k, t \models_c \neg\varphi$.

As in the pointwise case, the following proposition follows from the above definitions.

Proposition 5.1 *Let $\langle\sigma_i\rangle$ be a sequence of finite timed words. Let φ be an* MTL *formula and let $j \in \mathbb{N}$. If $\langle\sigma_i\rangle, j \models^c_{us} \varphi$, then it is not the case that $\langle\sigma_i\rangle, j \models^c_{un} \varphi$, and if $\langle\sigma_i\rangle, j \models^c_{un} \varphi$, then it is not the case that $\langle\sigma_i\rangle, j \models^c_{us} \varphi$.* □

In the proof of the ultimate satisfiability result for the pointwise case, we extensively use the argument that if a formula is ultimately satisfied at all points in a bounded interval then there exists a point in the periodic sequence, corresponding to the maximum of the stability points of the formula at these finite number of points, after which all timed words in the sequence satisfy the formula at all points in the interval. However the argument fails to hold for the continuous semantics since there are infinitely many time points even in a bounded interval. Towards tackling this problem, we define a *canonical* set of time points in a timed word such that the satisfiability of a formula is invariant between two consecutive points in the set. So given a finite timed word $\sigma = (a_1, t_1)\cdots(a_n, t_n)$ and a $p \in \mathbb{Q}_{>0}$, we define the set of *canonical points* in σ with respect to p to be the set containing 0 and $\{t \mid \exists i, c \in \mathbb{N} : t = t_i - cp\}$. Since this set is finite, we can arrange the time points in it in increasing order to get the sequence $r_0 r_1 \cdots r_m$ which we call the *canonical sequence* of σ with respect to p. We mention some of the immediate properties of a canonical sequence which we will use later.

Proposition 5.2 *Let σ be a finite timed word and $p \in \mathbb{Q}_{>0}$. Let $r_0 r_1 \cdots r_m$ be the canonical sequence of σ with respect to p. Then*

1. *For each $i \in \{0, \cdots, m-1\}$ σ does not contain any action in the interval (r_i, r_{i+1}).*
2. *Let $t, t' \in [0, \mathit{length}(\sigma)]$ with $t < t'$, such that (t, t') does not contain any r_i. Then for every $c \in \mathbb{N}$, the interval $(t + cp, t' + cp)$ also does not contain any r_i.*

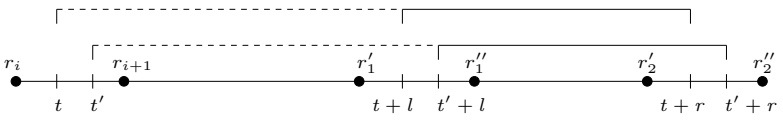
Lemma 5.1 *Let σ be a finite timed word and $p \in \mathbb{Q}_{>0}$. Let $r_0 r_1 \cdots r_m$ be the canonical sequence of σ with respect to p. Let $\varphi \in$ MTL(p). Then for each $i \in \{0, \cdots, m-1\}$ and for all $t, t' \in (r_i, r_{i+1})$, $\sigma, t \models_c \varphi$ iff $\sigma, t' \models_c \varphi$.*

Proof Proof is by an induction on the structure of φ. For the cases when φ is atomic or boolean combinations of formulas, the proof is straightforward. Let us consider the case when $\varphi = \eta U_I \psi$. Without loss of generality, we can us assume $t < t'$.

Suppose I is an open interval, i.e $I = (l, r)$, where $l = cp$ and $r = c'p$ or ∞, for some $c, c' \in \mathbb{N}$.

(⇒) Suppose $\sigma, t \models_c \varphi$. Then there exists a point t_s such that $t_s \in t + I$ and ψ is satisfied at t_s, and η is satisfied at all points in the interval (t, t_s).
We consider the following cases which depend on the interval in which t_s lies.
Case $t_s \in ((t + I) \cap (t' + I))$**:**

We can choose t_s itself to satisfy ψ, and clearly $\sigma, t' \models_c \eta U_I \psi$.



Case $t_s \in ((t + I) - (t' + I))$**:**

It is in the interval $(t+l, t'+l]$ as seen in the figure above. We can deduce from Proposition 5.2 and the fact that $t_s < \mathit{length}(\sigma)$, that there exist r'_1 and r''_1 which are adjacent elements in the canonical sequence such that $[t+l, t'+l]$ is contained in (r'_1, r''_1). From the fact that t_s is in the interval $(t + l, t' + l]$ and was the point chosen for the satisfaction of ψ, it follows that there also exists a point in the same interval before t_s which satisfies η. Now we obtain from the induction hypothesis that every point in (r'_1, r''_1) satisfy both η and ψ. Therefore there exists a point in the interval $t'+I$ before r''_1 which satisfies ψ. Now we can use this point for the satisfaction of ψ and deduce that $\eta U_I \psi$ is true at t'.
(⇐) In the other direction, let $\sigma, t' \models_c \varphi$. Then there exists t_s which is in the interval $t' + I$ such that it satisfies ψ and all points in the interval (t', t_s) satisfy η. Now there exists a point in the interval (r_i, r_{i+1}) which satisfies η (recall that the points t and t' are in the interval (r_i, r_{i+1})). By induction hypothesis we have that every point in the interval satisfies η (*). Again we consider the following cases depending on where t_s lies:
Case $t_s \in ((t + I) \cap (t' + I))$**:**

As before we can choose t_s to satisfy ψ and using (*) we can infer that $\eta U_I \psi$ is satisfied at t'.
Case $t_s \in ((t' + I) - (t + I))$**:**

Then clearly I is bounded. By an argument similar to the earlier one, we can deduce that there exist adjacent elements r'_2 and r''_2 in the canonical sequence such that both t and t' lie between them. The rest of the argument is similar to that in the other direction.

The proof for the case when I is singular is also similar. □

With each finite word σ we associate a sequence of delays which specifies the delays between the consecutive canonical points in the canonical sequence. So given a canonical sequence $r_0 r_1 \cdots r_m$ of σ with respect to p,

we call the sequence of delays $D = e_1 e_2 \cdots e_m$ an *invariant delay sequence* of σ with respect to p if each $e_i = r_i - r_{i-1}$. Given any subword of σ, $(d_i, a_i) \cdots (d_j, a_j)$, we can associate a delay sequence with it in a natural way which is given by $e_{i'} \cdots e_{j'}$, where i' and j' are such that $\sum_{k=1}^{i-1} d_k = \sum_{k=1}^{i'-1} e_k$ and $\sum_{k=i}^{j} d_k = \sum_{k=i'}^{j'} e_k$.

Proposition 5.3 *Let $\sigma = \mu\tau\nu$ be a finite timed word such that length$(\tau) = p$ and $p \in \mathbb{Q}_{>0}$. Let $D = D_1 D_2 D_3$ be the invariant delay sequence of σ with respect to p where D_1, D_2 and D_3 are the delay sequences corresponding to the subwords μ, τ and ν. Then for any j, the invariant delay sequence of $\mu\tau^j\nu$ with respect to p is $D_1 (D_2)^j D_3$.*

Proof Let $r = r^1 r^2 r^3$ be the canonical sequence of σ w.r.t p such that r^1, r^2 and r^3 correspond to D_1, D_2 and D_3. Let $r^1 = r_0 r_1 \cdots r_{n_1}$, $r^2 = r_{n_1+1} r_{n_1+2} \cdots r_{n_1+n_2}$ and $r^3 = r_{n_1+n_2+1} r_{n_1+n_2+2} \cdots\ r_{n_1+n_2+n_3}$. One can see that the canonical sequence of $\mu\tau^j\nu$ w.r.t p is:

$$
\begin{array}{llll}
r_0 & r_1 & \cdots & r_{n_1} \\
r_{n_1+1} & r_{n_1+2} & \cdots & r_{n_1+n_2} \\
r_{n_1+1} + p & r_{n_1+2} + p & \cdots & r_{n_1+n_2} + p \\
r_{n_1+1} + 2p & r_{n_1+2} + 2p & \cdots & r_{n_1+n_2} + 2p \\
 & \vdots & & \\
r_{n_1+1} + (j-1)p & r_{n_1+2} + (j-1)p & \cdots & r_{n_1+n_2} + (j-1)p \\
r_{n_1+n_2+1} + (j-1)p & r_{n_1+n_2+2} + (j-1)p & \cdots & r_{n_1+n_2+n_3} + (j-1)p
\end{array}
$$

Hence the invariant delay sequence associated with $\mu\tau^j\nu$ is $D_1 (D_2)^j D_3$. □

To mimic the proof of the pointwise case we define intervals in a timed word in which the satisfaction of formulas is invariant. Moreover we require this breaking up of the timed word into intervals to be consistent in some sense over the timed words in a periodic sequence. Hence we introduce the following definitions.

Given a canonical sequence $r_0 r_1 \cdots r_m$ of σ with respect to p, we define the *invariant interval sequence* of σ with respect to p to be $J = J_0 J_1 \cdots J_{2m}$ where $J_{2i} = [r_i, r_i]$ and $J_{2i+1} = (r_i, r_{i+1})$. It follows from Lemma 5.1 that the satisfiability of an MTL(p) formula is invariant in σ over each interval J_i.

Given a delay sequence $D = d_1 \cdots d_m$, we can associate an interval sequence $J = J_0 J_1 \cdots J_{2m}$ with it such that $J_0 = [0, 0]$, $J_{2i} = [t, t]$, where $t = \sum_{j=1}^{i} d_j$ and $J_{2i+1} = (t_1, t_2)$, where $t_1 = \sum_{j=1}^{i} d_j$ and $t_2 = \sum_{j=1}^{i+1} d_j$. Note that the interval sequence associated with an invariant delay sequence is the invariant interval sequence.

Lemma 5.2 *Let $\sigma = \mu\tau\nu$ be a finite timed word such that length$(\tau) = p$, where $p \in \mathbb{Q}_{>0}$. Let $D = D_1D_2D_3$ be the invariant delay sequence of σ with respect to p, where D_1, D_2 and D_3 are the delay sequences corresponding to the subwords μ, τ and ν. Let $\langle\sigma_i\rangle$ be the periodic sequence of finite timed words given by $\sigma_i = \mu\tau^i\nu$. Let $J = J_0J_1\cdots$ be the interval sequence corresponding to the delay sequence $D_1(D_2)^\omega$. Then for all $t \in J_j$ and $\varphi \in$ MTL(p),*

1. *if $\langle\sigma_i\rangle, t \models^c_{us} \varphi$ then there exists n_j such that for all $n \geq n_j$ and $t' \in J_j$, $\sigma_n, t' \models_c \varphi$ and*
2. *if $\langle\sigma_i\rangle, t \models^c_{un} \varphi$ then there exists n_j such that for all $n \geq n_j$ and $t' \in J_j$, $\sigma_n, t' \not\models_c \varphi$.*

Proof It follows from Proposition 5.3 that the invariant delay sequence associated with $\mu\tau^i\nu$ is $D_1(D_2)^iD_3$. Hence the invariant interval sequences associated with $\mu\tau\nu$, $\mu\tau\tau\nu$, $\mu\tau\tau\tau\nu$,$\cdots$ are of the form $K_1K_2K_3'$, $K_1K_2K_3K_4'$, $K_1K_2K_3K_4K_5'$,$\cdots$ where K_i's are themselves interval sequences. So given any J_j it belongs to some K_k and hence there exists $k-1$ such that for all $k' \geq k-1$, the satisfiability of $\varphi \in$ MTL(p) is invariant over the interval J_j in σ'_k. If $\langle\sigma_i\rangle, t \models^c_{us} \varphi$, then there exists m such that for all $m' \geq m$, $\sigma_{m'}, t \models_c \varphi$. Taking $n = max(k, m)$, we have that for all $n' \geq n$, for all $t' \in J_j$, $\sigma_{n'}, t' \models_c \varphi$. We can similarly prove the other claim. □

We call the n_j above, the *stability point* of φ at J_j in $\langle\sigma_i\rangle$.

Proposition 5.4 *Let $\langle\sigma_i\rangle$ be the periodic sequence given by $\sigma_i = \mu\tau^i\nu$ and length$(\tau) = p$, where $p \in \mathbb{Q}_{>0}$. Let $t \in \mathbb{R}_{\geq 0}$ such that $t >$ length(μ). Let $c \in \mathbb{N}$ and let $\varphi \in$ MTL. Then $\langle\sigma_i\rangle, t \models^c_{us} \varphi$ iff $\langle\sigma_i\rangle, t + cp \models^c_{us} \varphi$.*

Proof If σ_k satisfies φ at t, then σ_{k+c} satisfies φ at $t+cp$ because the futures of the two timed words from the corresponding points are the same. Hence if $\langle\sigma_i\rangle, t \models^c_{us} \varphi$ then $\langle\sigma_i\rangle, t + cp \models^c_{us} \varphi$. The proof of the other direction is similar. □

Theorem 5.1 *Let $\langle\sigma_i\rangle$ be a periodic sequence with period p, where $p \in \mathbb{Q}_{>0}$. Let φ be an* MTL(p) *formula and let $t \in \mathbb{R}_{\geq 0}$. Then either $\langle\sigma_i\rangle, t \models^c_{us} \varphi$ or $\langle\sigma_i\rangle, t \models^c_{un} \varphi$.*

Proof Sketch The proof follows that for the pointwise case. Since the ultimate satisfiability of a formula is invariant within the intervals of an invariant interval sequence, and there exists an n_j for each interval J_j as given

by Lemma 5.2, we consider each of these intervals as one entity (comparable to a point in the pointwise case). For example if ψ is not ultimately satisfied at all time t in a bounded interval I, then we can assume that there exists a point in the sequence after which ψ is not satisfied at any point in I.

Detailed Proof

Since $\langle\sigma_i\rangle$ is periodic with period p, there exist finite timed words μ, τ and ν such that for each i, $\sigma_i = \mu\tau^i\nu$ and $length(\tau) = p$. Let $\rho = \mu\tau^\omega = (a_0, t_0)(a_1, t_1)\cdots$.

We now use induction on the structure of φ. For the atomic case and boolean combinations of formulas, the proof is similar to that for the pointwise case. Hence we consider the following case.

Case $\varphi = \eta U_I \psi$:

We assume $0 \notin I$. Let $D = D_1D_2D_3$ be the invariant delay sequence of $\mu\tau\nu$, where D_1, D_2 and D_3 correspond to μ, τ and ν respectively. Let $\langle J_i\rangle = J_0J_1J_2\cdots$ be the interval sequence associated with $D_1(D_2)^\omega$. We define $between(i, j)$ to be the indices of intervals between the i-th and j-th intervals. Hence $between(i, j) = \{i + 1, \cdots, j - 1\} \cup S_1 \cup S_2$ where $S_1 = \{i\}$ if i is odd, $\emptyset$ otherwise and $S_2 = \{j\}$ if j is odd, $\emptyset$ otherwise.

We consider two cases, one in which there exists $t' \geq t$ such that $t' - t \in I$ and $\langle\sigma_i\rangle, t' \models^c_{us} \psi$, and the other in which the above condition does not hold.

- Let $t \in J_j$. In the first case, there exists j' such that $t' \in J_{j'}$, $t' \geq t$, $t' - t \in I$ and $\langle\sigma_i\rangle, t' \models^c_{us} \psi$. Let j_s be the smallest such j'.
  - Now suppose for all $k \in between(j, j_s)$, $\langle\sigma_i\rangle, t'' \models^c_{us} \eta$ for some $t'' \in J_k$. Let n_k be the stability point of η at J_k. Let m be the stability point of ψ at J_{j_s}. Let n' be the maximum of all n_k's and m. It can be seen that for all $n'' \geq n'$, $\sigma_{n''}, t \models_c \eta U_I \psi$. Hence $\langle\sigma_i\rangle, t \models^c_{us} \varphi$.
  - Otherwise there exists $k \in between(j, j_s)$ such that for all $t'' \in J_k$ it is not the case that $\langle\sigma_i\rangle, t'' \models^c_{us} \eta$. By induction hypothesis, then for all $t'' \in J_k$, $\langle\sigma_i\rangle, t'' \models^c_{un} \eta$. Let m_k be the stability point of η at J_k. Let m be the stability point of ψ at J_{j_s}. For every j'' such that $j'' \leq j_s$ and $J_{j''} \cap t + I \neq \emptyset$, let $n_{j''}$ be the stability point of ψ at $J_{j''}$. Let n' be the maximum of m, $n_{j''}$'s and m_k. Then for all $n'' \geq n'$, $\sigma_{n''}, t \not\models_c \eta U_I \psi$. Hence $\langle\sigma_i\rangle, t \models^c_{un} \varphi$.
- Now turning to the second case, suppose that for all $t' \geq t$ such that

$t' - t \in I$, it is not the case that $\langle \sigma_i \rangle, t' \models^c_{us} \psi$. Then, by induction hypothesis, for all $t' \geq t$ such that $t' - t \in I$, $\langle \sigma_i \rangle, t' \models^c_{un} \psi$.

- Suppose I is bounded. If there is no t' such that $t' - t \in I$, then it is easy to see that $\langle \sigma_i \rangle, t \models^c_{un} \eta U_I \psi$. Otherwise there exist finite (non-zero) number of j''s such that $J_{j'} \cap t + I \neq \emptyset$. For each such j', let $n_{j'}$ be the stability point of ψ at $J_{j'}$. Let n' be the maximum of $n_{j'}$'s. For all $n'' \geq n'$, $\sigma_{n''}, t \not\models_c \eta U_I \psi$. Hence $\langle \sigma_i \rangle, t \models^c_{un} \varphi$.
- Suppose I is unbounded. Let $D_1 = d_1 \cdots d_l$, $D_2 = e_1 \cdots e_m$ and $D_3 = f_1 \cdots f_n$, where $D = D_1 D_2 D_3$ is the invariant delay sequence of $\mu \tau \nu$ with respect to p, and D_1, D_2 and D_3 correspond to μ, τ and ν, respectively. Let $\tau = (d'_1, a'_1) \cdots (d'_{m'}, a'_{m'})$ and $\nu = (d''_1, a''_1) \cdots (d''_{n''}, a_{n''})$. Given any suffix $D_s = e_i \cdots e_m$ of D_2, we can associate with it a suffix $\tau_s = (d', a'_{i'})(d'_{i'+1}, a'_{i'+1}) \cdots (d'_{m'}, a'_{m'})$ of τ such that $\sum_{k=i}^{m} e_k = d' + \sum_{k=i'+1}^{m'} d'_k$. Similarly we can associate suffixes of the timed word ν with suffixes of the delay sequence D_3.

  Let $S = \{s_1, s_2, \cdots, s_m\}$ be the suffixes of τ, where s_i corresponds to $e_i \cdots e_m$. Similarly let $W = \{w_1, w_2, \cdots, w_n\}$ be the suffixes of ν. Let $X = W \cup (S \cdot \tau^* \cdot \nu)$. It can be seen that for any timed word $\tau_1 = (g_0, b_0)(g_1, b_1) \cdots (g_{l'}, b_{l'})$ in X, the satisfiability of an MTL(p) formula is invariant in the interval $(0, g_0)$. We call g_0 the *first delay* of τ_1. Hence we say that a timed word τ_1 in X satisfies at point, an MTL(p) formula φ_1, if $\tau_1, 0 \models_c \varphi_1$ and τ_1 satisfies in interval, an MTL(p) formula φ_1, if $\tau_1, t' \models_c \varphi_1$ for all $t' \in (0, g_0)$ (or equivalently some $t' \in (0, g_0)$).

  We now claim that only finitely many timed words from X satisfy ψ at point or in interval. Otherwise infinitely many timed words from $W \cup (S \cdot \tau^* \cdot \nu)$ would satisfy ψ at point or in interval and hence infinitely many from $(\{s_i\} \cdot \tau^* \cdot \nu)$ would satisfy ψ at point or in interval, for some i. It they satisfy at point then $\langle \sigma_i \rangle, t' \models^c_{us} \psi$ where $t' = \mathit{length}(\mu) + \sum_{k=1}^{i-1} d'_k$. Otherwise they satisfy in interval and hence $\langle \sigma_i \rangle, t' \models^c_{us} \psi$ for all $t' \in \mathit{length}(\mu) + \sum_{k=1}^{i-1} d'_k + (0, d'_i)$. Hence by Proposition 5.4, $\langle \sigma_i \rangle, t' + cp \models^c_{us} \psi$ for all $c \in \mathbb{N}$, where $p = \mathit{length}(\tau)$. Therefore there exists $t' \in t + I$ (since I is unbounded) such that $t' - t \in I$ and $\langle \sigma_i \rangle, t' \models^c_{us} \psi$. This is a contradiction. So, only finitely many timed words from X satisfy ψ at point or in interval.

  ψ is ultimately not satisfied at every $t' \in t + I \cap (t, \mathit{length}(\mu))$. Since this interval is bounded there are only finitely many j'''s

such that $J_{j''}$ has a non-empty intersection with this interval. Let $n_{j''}$ be the stability point of ψ at $J_{j''}$.

Suppose there exists no timed word in X which satisfies ψ at point or in interval. Then let n' be the maximum of all $n_{j''}$'s. For all $n'' \geq n'$, $\sigma_{n''}, t \not\models_c \eta U_I \psi$. Hence $\langle \sigma_i \rangle, t \models^c_{un} \varphi$.

Suppose there exists a timed word in X which satisfies ψ at point or in interval. Since we proved that such timed words are finite in number, let $l' = \mathit{length}(\tau_1)$ where $\tau_1 \in X$ is such that it satisfies ψ at point or in interval, and for all $\tau_2 \in X$ such that $\mathit{length}(\tau_2) > \mathit{length}(\tau_1)$, τ_2 does not satisfy ψ at point and τ_2 does not satisfy ψ in interval.

Suppose τ_1 satisfies ψ at point. Suppose there exists $\tau_3 \in X$ such that $\mathit{length}(\tau_3) > \mathit{length}(\tau_1)$, and τ_3 does not satisfy η at point or τ_3 does not satisfy η in interval. Then let n' be such that $\mathit{length}(\sigma_{n'}) > max(\mathit{length}(\mu), t) + \mathit{length}(\tau_3)$. It can now be argued that for all $n'' \geq n$, $\sigma_{n''}, t \not\models_c \eta U_I \psi$. Hence $\langle \sigma_i \rangle, t \models^c_{un} \varphi$.

Suppose τ_1 does not satisfy ψ at point. Then τ_1 satisfies ψ in interval. Now, suppose there exists $\tau_3 \in X$ such that $\mathit{length}(\tau_3) \geq \mathit{length}(\tau_1)$, and τ_3 does not satisfy η at point or τ_3 does not satisfy η in interval. We can then use an argument similar to the previous case.

Suppose that the above two conditions do not hold. Then if there is a point t'' in the interval $(t, \mathit{length}(\mu))$ such that $\langle \sigma_i \rangle, t'' \models^c_{un} \eta$, then $\langle \sigma_i \rangle, t \models^c_{us} \eta U_I \psi$. Otherwise $\langle \sigma_i \rangle, t \models^c_{us} \eta U_I \psi$.

□

The above theorem gives us a counter-freeness result for the continuous case. Given a $p \in \mathbb{R}_{>0}$, we call a timed language L, *p-counter-free*, if there do not exist finite timed words μ, τ and ν such that $\mathit{length}(\tau) = p$ and there exist infinitely many i's for which $\mu\tau^i\nu \in L$ and infinitely many of them for which $\mu\tau^i\nu \notin L$. Below is the result for the continuous semantics.

Theorem 5.2 *Let $p \in \mathbb{Q}_{>0}$. Then every timed language of finite words definable by an* MTL(p) *formula in the continuous semantics is p-counter-free.* □

5.2 L_{last_a} not expressible by MTL^c

In this section we show that the language L_{last_a} is not expressible by any MTL formula in the continuous semantics. L_{last_a} consists of timed words over $\{a, b\}$ that contain an action at time 1, which is immediately preceded by an action a. We sketch a proof of the above claim for the case of infinite words, which is along the lines of [5], and then show how it can be extended for finite words.

The idea of the proof is to enumerate two families of timed words parameterized by a rational number p and a natural number n, such that one is contained in the language while the other is completely outside the language. Then we show that no MTL formula with granularity p and level n can distinguish between the models in the two families corresponding to p and n, which would lead to its inexpressibility.

Let $p = 1/q$, where $q \in \mathbb{N}$ and $q > 0$, and let $n \in \mathbb{N}$. We give the two infinite timed words $\alpha^{p,n}$ and $\beta^{p,n}$, depicted below. Let $d = p/(n+4)$. Then

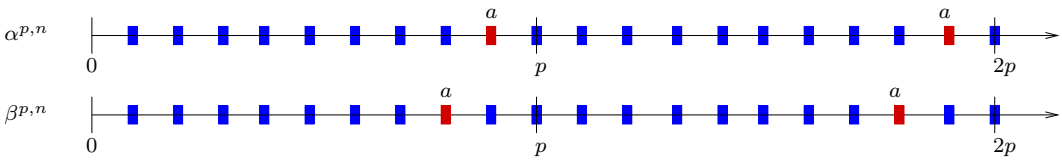
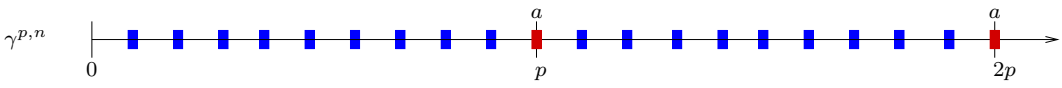


Figure 5.1: Models for showing inexpressibility of L_{last_a}.

$\alpha^{p,n}$ is given by $(c_1, d)(c_2, 2d)\cdots$ where $c_k = a$ if $k \mod (n+4) = n+3$, $c_k = b$ otherwise. And $\beta^{p,n}$ is given by $(c_1, d)(c_2, 2d)\cdots$ where $c_k = a$ if $k \mod (n+4) = n+2$, $c_k = b$ otherwise. It can be seen that $\alpha^{p,n}$ is in L_{last_a}, whereas $\beta^{p,n}$ is not in L_{last_a}. Let us denote by MTL(p, n), the set of MTL formulas with granularity p, and with an U nesting depth of less than or equal to n. We now show that no MTL(p, n) formula can distinguish between $\alpha^{p,n}$ and $\beta^{p,n}$ in the continuous semantics.

Let us consider the following infinite model $\gamma^{p,n}$ which is given as (c_1, d) $(c_2, 2d)\cdots$ where $c_k = a$ if $k \mod (n+4) = 0$, $c_k = b$ otherwise.



Proposition 5.5 *Let $\varphi \in$ MTL(p, n), $c \in \mathbb{N}$ and $t \in \mathbb{R}_{\geq 0}$. Then*

1. *$\alpha^{p,n}, t \models_c \varphi$ iff $\alpha^{p,n}, t + cp \models_c \varphi$,*

2. $\beta^{p,n}, t \models_c \varphi$ *iff* $\beta^{p,n}, t + cp \models_c \varphi$, *and*

3. $\gamma^{p,n}, t \models_c \varphi$ *iff* $\gamma^{p,n}, t + cp \models_c \varphi$.

Proof The suffixes of the infinite timed words with respect to t and $t + cp$ are the same. □

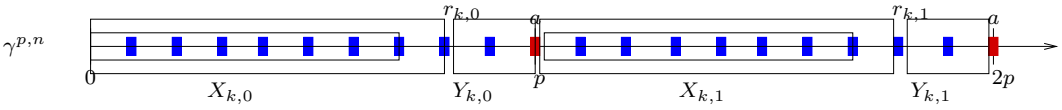
Lemma 5.3 *Let* $k \in \mathbb{N}$ *and* $0 \leq k \leq n$, *and let* $\varphi \in \text{MTL}(p, k)$. *Let* $i, j \in \{1, \cdots, n + 3 - k\}$ *and let* $m \geq 0$. *Then*

1. $\gamma^{p,n}, (m(n + 4) + i)d \models_c \varphi$ *iff* $\gamma^{p,n}, (m(n + 4) + j)d \models_c \varphi$ *and*

2. *for all* $t_1, t_2 \in (0, d)$, $\gamma^{p,n}, (m(n+4)+i)d-t_1 \models_c \varphi$ *iff* $\gamma^{p,n}, (m(n+4)+j)d - t_2 \models_c \varphi$.

Proof Sketch We prove inductively that a formula of U nesting depth k cannot distinguish between the first k action points or the time points between them, in each p interval. The intuition behind this is that, to distinguish between the points, a formula will have to count the number of b's before the next a, and this requires a formula with a nesting depth equal to that of the number of b's.

Detailed Proof



We note that the action points in $\gamma^{p,n}$ are the only canonical points with respect to p. So, the satisfiability of an MTL(p) formula is invariant in the corresponding interval. Hence we will use $\gamma^{p,n}, pt(i) \models_c \varphi$ to denote $\gamma^{p,n}, id \models_c \varphi$ and $\gamma^{p,n}, int(i) \models_c \varphi$ to denote $\gamma^{p,n}, id - t \models_c \varphi$ for all $t \in (0, d)$.

We define $X_{k,m} = \{m(n + 4) + 1, \cdots, m(n + 4) + (n + 3 - k)\}$ and $Y_{k,m} = \{m(n+4)+(n+3-k+1), \cdots, m(n+4)+(n+4)\}$. Let $\gamma^{p,n} = (a_1, t_1)(a_2, t_2) \cdots$. We first use induction on k and then on the structure of φ. When $k = 0$, φ is a boolean combination of atomic formulas and hence the lemma is trivially true. Let us assume that that the lemma holds for k. We now use induction on the structure of φ. The atomic case and boolean combinations of formulas

are straightforward. Let $\varphi = \eta U_I \psi$. We assume $I = [cp, cp]$ or $I = (cp, r)$, where $r = c'p$ or ∞ and $c, c' \in \mathbb{N}$.

Case $I = [cp, cp]$: Let $i, j \in X_{k+1,m}$. If $\gamma^{p,n}, pt(i) \models_c \eta U_I \psi$, then $\gamma^{p,n}, pt(i+c(n+4)) \models_c \psi$, and $\gamma^{p,n}, pt(i') \models_c \eta$ for all $i < i' < i+c(n+4)$ and $\gamma^{p,n}, int(i') \models_c \eta$ for all $i < i' \leq i+c(n+4)$. Let $r_{k,m} = m(n+4)+n+3-k$ be the rightmost point in $X_{k,m}$. Since $i < r_{k,m} < i+c(n+4)$, $\gamma^{p,n}, pt(r_{k,m}) \models_c \eta$ and $\gamma^{p,n}, int(r_{k,m}) \models_c \eta$. By induction hypothesis, $\gamma^{p,n}, pt(j') \models_c \eta$ and $\gamma^{p,n}, int(j') \models_c \eta$ for all $j' \in X_{k,m}$ and hence for all $j' \in X_{k,m'}$ (by Proposition 5.5) for all $m' \in \mathbb{N}$. $\gamma^{p,n}, pt(j+c(n+4)) \models_c \psi$ (by induction hypothesis), $\gamma^{p,n}, pt(j') \models_c \eta$ and $\gamma^{p,n}, int(j') \models_c \eta$ for all $j < j' \leq r_{k,m}$, and for all j' such that $j' \in X_{k,m+c}$ and $j' < j + c(n+4)$ (from what we showed above), and $\gamma^{p,n}, pt(j') \models_c \eta$ and $\gamma^{p,n}, int(j') \models_c \eta$ for all $j' \in \{r_{k,m}+1, \cdots, m+c(n+4)\}$ (because i' ranged over these points). Hence $\gamma^{p,n}, pt(j) \models_c \eta U_I \psi$. Similar, we can argue for the case when $\gamma^{p,n}, int(i) \models_c \eta U_I \psi$.

Case $I = (cp, r)$: Let $i, j \in X_{k+1,m}$. Suppose $\gamma^{p,n}, pt(i) \models_c \eta U_I \psi$. Suppose $\gamma^{p,n}, pt(i'') \models_c \psi$ for some $t_{i''} - t_i \in I$ such that $\gamma^{p,n}, pt(i') \models_c \eta$ for all $i < i' < i''$ and $\gamma^{p,n}, int(i) \models_c \eta$ for all $i < i' \leq i''$.

If $i'' \in Y_{k,m'}$ for some $m' \in \mathbb{N}$, then $t_{i''} - t_j \in I$. By an argument similar to the previous case we can argue that $\gamma^{p,n}, pt(j') \models_c \eta$ for all $j < j' < i''$ and $\gamma^{p,n}, int(j') \models_c \eta$ for all $j < j' \leq i''$. Hence $\gamma^{p,n}, pt(j) \models_c \eta U_I \psi$.

Suppose $i'' \in X_{k,m'}$ and $i'' > i + (m'-m)(n+4)$. Let r be the rightmost point in $X_{k,m'}$. $\gamma^{p,n}, pt(r) \models_c \psi$ by induction hypothesis and $t_r - t_j \in I$. η is true at all the intermediate points by an argument similar to the previous case. Hence $\gamma^{p,n}, pt(j) \models_c \eta U_I \psi$.

Suppose $i'' \in X_{k,m'}$ and $i'' < i + (m'-m)(n+4)$. Let l be the leftmost point in $X_{k,m'}$. If j is not the leftmost point in $X_{k,m}$, then $t_l - t_j \in I$ and by induction hypothesis, $\gamma^{p,n}, pt(l) \models_c \psi$. Otherwise, let $l' = l - (n+3)$. $t_{l'} - t_j \in I$ and $\gamma^{p,n}, pt(l) \models_c \psi$ by induction hypothesis and Proposition 5.5. And η is true at all the intermediate points. Hence $\gamma^{p,n}, pt(j) \models_c \eta U_I \psi$.

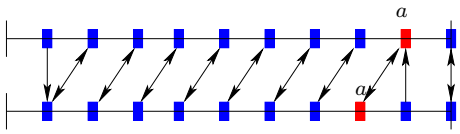
If we take ψ to be satisfied is at some $int(i'')$ or assert $\eta U_I \psi$ at some $int(i)$, then the argument is similar. □

Corollary 5.1 *Let $p = 1/q$, where $q \in \mathbb{N}$ and $q > 0$, and let $n \in \mathbb{N}$. Let $\varphi \in \mathrm{MTL}(p, n)$ and let $m \geq 0$. Then*

1. *$\gamma^{p,n}, (m(n+4)+1)d \models_c \varphi$ iff $\gamma^{p,n}, (m(n+4)+2)d \models_c \varphi$ iff $\gamma^{p,n}, (m(n+4)+3)d \models_c \varphi$ and*
2. *for all $t_1, t_2, t_3 \in (0, d)$, $\gamma^{p,n}, (m(n+4)+1)d - t_1 \models_c \varphi$ iff $\gamma^{p,n}, (m(n+4)+2)d - t_2 \models_c \varphi$ iff $\gamma^{p,n}, (m(n+4)+3)d - t_3 \models_c \varphi$.* □

Lemma 5.4 *Let* $\varphi \in \mathrm{MTL}(p,n)$. *Then* $\alpha^{p,n}, 0 \models_c \varphi$ *iff* $\beta^{p,n}, 0 \models_c \varphi$.

Proof Sketch We prove by induction on the structure of φ that no MTL formula with granularity p distinguishes the two models. Below is the diagram which show the points which we choose for the satisfaction of the formula ψ in one model corresponding to the one in the other model, for the case of $\eta U_I \psi$.



It can be seen that these points can be chosen because they lie in the same interval, and ψ is true at the corresponding points either because the futures of the two points are the same or because of Lemma 5.3.

Detailed Proof

Let $\alpha^{p,n} = (a_1,t_1)(a_2,t_2)\cdots$ and $\beta^{p,n} = (b_1,t'_1)(b_2,t'_2)\cdots$. Proof is by induction on the structure of φ. The atomic case and boolean combinations of formulas are straightforward. Let $\varphi = \eta U_I \psi$.
$\Rightarrow$ Suppose $\alpha^{p,n}, 0 \models_c \eta U_I \psi$. Consider the case when there exists j such that $\alpha^{p,n}, jd \models_c \psi$, where $t_j \in I$, and for all $0 < t < jd$, $\alpha^{p,n}, t \models_c \eta$.

If $j \mod (n+4)$ is 0, then let $i = j$, if $j \mod (n+4) > 1$, then let $i = j-1$ and if $j \mod (n+4) = 1$, then let $i = j$. It can be seen that $t'_i \in I$. Now $\beta^{p,n}, i \models_c \psi$ from Proposition 5.5 and corollary 5.1. Since we have chosen i to be less than or equal to j, η is true at all points between 0 and id in $\beta^{p,n}$ from Proposition 5.5. Hence $\beta^{p,n}, 0 \models_c \eta U_I \psi$.
$\Leftarrow$ Suppose $\beta^{p,n}, 0 \models_c \eta U_I \psi$. Suppose there exists j such that $\beta^{p,n}, jd \models_c \psi$, where $t'_j \in I$, and for all $0 < t < jd$, $\beta^{p,n}, t \models_c \eta$.

If $j \mod (n+4)$ is 0, then let $i = j$, if $j \mod (n+4) < (n+3)$ and $j \neq 1$, then let $i = j+1$, if $j = 1$, then $i = 1$ and if $j \mod (n+4) = (n+3)$, then let $i = j-(n+2)$. It is not difficult to see that $t_i \in I$. $\alpha^{p,n}, i \models_c \psi$ from Proposition 5.5 and corollary 5.1. η is true at all points in (d, id) from Proposition 5.5. If $j \neq 1$, then $\beta, t \models_c \eta$ for all $t \in (0,d]$. Hence from Proposition 5.5 and corollary 5.1, $\alpha, t \models_c \eta$ for all $t \in (0,d]$. However if $j = 1$, then $i = 1$ and $\alpha, t \models_c \eta$ for all $t \in (0,d)$ by a similar argument.

If ψ is satisfied at some $jd+t$, where $t \in (0,d)$, then we choose $id+t$ as the point in the other model where ψ is satisfied, and the rest of the argument is the same. $\square$

Theorem 5.3 ([5]) *The timed language L_{last_a} is expressible by* MTL_S^{pw} *but not by* MTL^c *over infinite timed words.*

Proof Given an MTL^c formula, it belongs to $\mathrm{MTL}(p,n)$ for some p and n. However it cannot then distinguish between $\alpha^{p,n}$ and $\beta^{p,n}$. Hence no MTL^c formula can express L_{last_a}. But the following MTL_S formula defines the same in the pointwise semantics: $\Diamond_{[1,1]}(\neg\varphi_{action} S a)$.

□

Theorem 5.4 *The timed language L_{last_a} is expressible by* MTL^c *but not by* MTL_S^{pw} *over finite timed words.*

Proof We use the notion of ultimate satisfiability to extend the above proofs for the case of finite words. We replace the infinite word $\alpha^{p,n}$, $\beta^{p,n}$ and $\gamma^{p,n}$ by the sequences of finite timed words, $\langle\sigma_i^{p,n}\rangle$, $\langle\rho_i^{p,n}\rangle$ and $\langle\kappa_i^{p,n}\rangle$. For each i, $\sigma_i^{p,n} = \mu_1\tau^i$, $\rho_i^{p,n} = \mu_2\tau^i$ and $\kappa_i^{p,n} = \tau^{i+1}$ where $\mu_1 = (b,d)(b,2d)\cdots(b,(n+2)d)(a,(n+3)d)$, $\mu_2 = (b,d)(b,2d)\cdots(b,(n+1)d)(a,(n+2)d)$ and $\tau = (b,d)(b,2d)\cdots(b,(n+3)d)(a,(n+4)d)$.

We note that with the replacement of the infinite models by the above sequences of finite timed words, and $\models_c$ by $\models_{us}^c$, the above propositions, lemmas and theorems continue to hold. The proofs of Lemma 5.3 and Lemma 5.4 follow that of Theorem 5.1 for the atomic and boolean combinations of formulas. For the case where $\varphi = \eta U_I \psi$, they mimic the proofs for infinite case, if ψ is ultimately satisfied at some point in the interval I, otherwise they follow that of Theorem 5.1 for the same case. □

5.3 Conclusion

In this chapter we showed that an MTL formula with granularity p is either ultimately satisfied or ultimately not satisfied by a periodic sequence with period p. We deduce from it that the languages over finite timed words definable by MTL^c formulas with granularity p are p-counter-free. We note here that the results we have regarding counter-freeness for the continuous case is weaker than that for the pointwise case. Finally we show how these results can be used to extend to the case of finite words the proof of the inexpressibility of L_{last_a} by MTL^c over infinite timed words [5].

Chapter 6

Languages inexpressible with past

In the previous chapters we exhibited certain languages which were not expressible by MTL in pointwise and continuous semantics. In this chapter we will elicit languages not expressible by MTL with S and S_I in the pointwise semantics.

In section 6.1, we show that the language L_{2ins} is not expressible by $\mathrm{MTL}_{S_I}^{pw}$ over both finite and infinite words. In section 6.2, we show that the language L_{em} is not expressible by MTL_{S}^{pw}.

6.1 L_{2ins} not expressible by $\mathrm{MTL}_{S_I}^{pw}$

In this section we show that the language L_{2ins} (for "two insertions") is not expressible by $\mathrm{MTL}_{S_I}^{pw}$ which would imply its inexpressibility by MTL_{S}^{pw} and MTL^{pw}. However this language is seen to be expressible by MTL^{c} and hence by MTL_{S}^{c} and $\mathrm{MTL}_{S_I}^{c}$. This leads to the strict containment of the pointwise versions of the logics in their corresponding continuous versions.

We first show the result for finite words and then sketch how it can be extended for infinite words. L_{2ins} is the timed language over $\Sigma = \{a, b\}$ in which every timed word in the language contains two consecutive a's such that there exist two distinct time points between their times of occurrences, at distance one in the future from each of which there is an a. Formally, $L_{2ins} = \{\sigma \in T\Sigma^* \mid \sigma = (a_1, t_1) \cdots (a_n, t_n), \exists i, j, k \in \mathbb{N} : a_i = a_{i+1} = a, t_j, t_k \in (t_i + 1, t_{i+1} + 1), j \neq k \text{ and } a_j = a_k = a\}$.

Let $p = 1/q$, where $q \in \mathbb{N}$ and $q > 0$, and let $n \in \mathbb{N}$. Let $d = p/(2n + 3)$. We give the two models $\sigma^{p,n}$ and $\rho^{p,n}$ which are as defined below. $\sigma^{p,n}$ is given by $(a, 1-p+d/2)(a, 1-p+3d/2)\cdots(a, 1-p/2-d)(a, 1-p/2)(a, 1-$

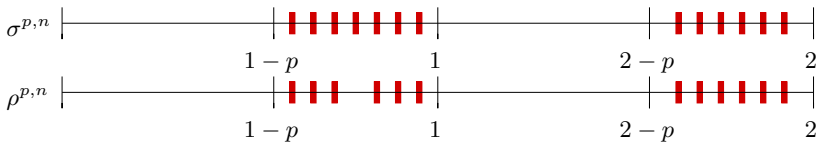
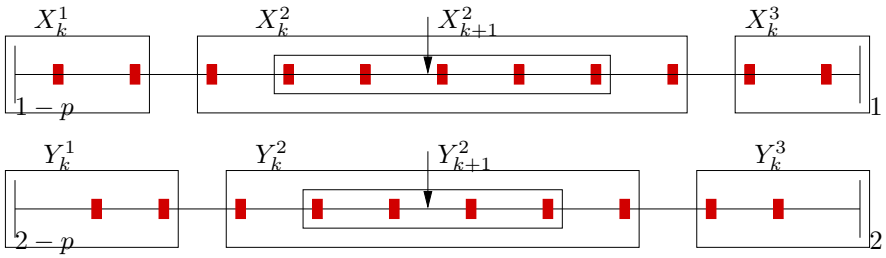


Figure 6.1: Models for showing inexpressibility of L_{2ins}.

$p/2+d)\cdots(a,1-d/2)(a,2-p+d)(a,2-p+2d)\cdots(a,2-d)$.
And $\rho^{p,n}$ is given by $(a,1-p+d/2)(a,1-p+3d/2)\cdots(a,1-p/2-d)(a,1-p/2+d)\cdots(a,1-d/2)(a,2-p+d)(a,2-p+2d)\cdots(a,2-d)$.
Clearly $\sigma^{p,n} \notin L_{2ins}$ and $\rho^{p,n} \in L_{2ins}$. We use the following lemmas to show that no MTL_{S_I} formula can define L_{2ins} in the pointwise semantics. We use $\mathrm{MTL}_{S_I}(p,n)$ to denote the MTL_{S_I} formulas with granularity p, and nesting depth of U and S, n.

Lemma 6.1 *Let $k \in \mathbb{N}$ and $0 \leq k \leq n$. Let $X_k^2 = \{k+1,\cdots,2n+3-k\}$ and $Y_k^2 = \{2n+3+(k+1),\cdots,2n+3+(2n+2-k)\}$. Let $\varphi \in \mathrm{MTL}_{S_I}(p,k)$. Then for all $i,j \in X_k^2$, $\sigma^{p,n},i \models_{pw} \varphi$ iff $\sigma^{p,n},j \models_{pw} \varphi$ and for all $i,j \in Y_k^2$, $\sigma^{p,n},i \models_{pw} \varphi$ iff $\sigma^{p,n},j \models_{pw} \varphi$.*

Proof



Let $X_k^1 = \{1,\cdots,k\}$ and $X_k^3 = \{2n+3-k+1,\cdots,2n+3\}$. Let $Y_k^1 = \{2n+3+1,\cdots,2n+3+k\}$ and $Y_k^3 = \{2n+3+(2n+2-k+1),\cdots,2n+3+(2n+2)\}$.

Let $\sigma^{p,n} = (a_1,t_1)\cdots(a_{4n+5},t_{4n+5})$. Proof is by induction on k and then on the structure of φ. If $k=0$ then φ is a boolean combination of atomic formulas and the lemma is trivially true.

Let us assume that the lemma is true for k. We will prove by induction on the structure of φ that the lemma holds for $k+1$. The case when φ is atomic or of the form $\neg\psi$ or $\eta \vee \psi$ is straightforward.

Case $\varphi = \eta U_I \psi$: Assume $0 \notin I$. Suppose $i,j \in X_{k+1}^2$. $\sigma^{p,n},i \models_{pw} \eta U_I \psi \Rightarrow \exists i' > i : t_{i'} - t_i \in I, \sigma^{p,n},i' \models_{pw} \psi, \forall i'' : i < i'' < i', \sigma^{p,n},i'' \models_{pw} \eta$.

1. $i' \in X_k^3$:

   (a) $(0,p) \subseteq I$. $t_{i'} - t_j \in (0,p)$, hence $t_{i'} - t_j \in I$.

   (b) Since i is in X_{k+1}^2 and i' is greater than every number in X_k^2, the rightmost position in X_k^2, r satisfies η. Hence every position in X_k^2 satisfies η (by induction hypothesis). For all j'', $j < j'' \leq r$, $\sigma^{p,n}, j'' \models_{pw} \eta$.

   (c) For j'', $r < j'' < i'$, $\sigma^{p,n}, j'' \models_{pw} \eta$ (since $i < r$, $r < i'$ and for all i'', $i < i'' < i'$, $\sigma^{p,n}, i'' \models_{pw} \eta$).

   Therefore $\exists i' : i' > j, t_{i'} - t_j \in I, \sigma^{p,n}, i' \models_{pw} \psi, \forall j'' : j < j'' < i', \sigma^{p,n}, j'' \models_{pw} \eta$, and hence $\sigma^{p,n}, j \models_{pw} \varphi$.

2. $i' \in Y_k^3$: The argument is similar to the above except that $1(a)$ needs to be replaced by: $(1, 1+p) \subseteq I$. $t_{i'} - t_j \in (1, 1+p)$, hence $t_{i'} - t_j \in I$.

3. $i' \in Y_k^1$: The argument is similar to 1 except that $1(a)$ needs to be replaced by: $(1-p, 1) \subseteq I$. $t_{i'} - t_j \in (1-p, 1)$, hence $t_{i'} - t_j \in I$.

4. $i' \in Y_k^2$: Suppose $t_{i'} - t_i \in (1-p, 1)$.

   (a) $(1-p, 1) \subseteq I$. Let l' be the leftmost point in Y_k^2. Since $t_{l'} - t_j \in (1-p, 1)$, $t_{l'} - t_j \in I$.

   (b) $i' \in Y_k^2$ and $\sigma^{p,n}, i' \models_{pw} \psi$. By induction hypothesis $\sigma^{p,n}, l' \models_{pw} \psi$ since $l' \in Y_k^2$.

   (c) Since i is in X_{k+1}^2 and i' in Y_k^2, the rightmost position in X_k^2, r satisfies η. Hence every position in X_k^2 satisfies η. For all j'', $j < j'' \leq r$, $\sigma^{p,n}, j'' \models_{pw} \eta$.

   (d) For j'', $r < j'' < l'$, $\sigma^{p,n}, j'' \models_{pw} \eta$ (since $i < r$, $r < l'$ and $l' < i'$, and for all i'', $i < i'' < i'$, $\sigma^{p,n}, i'' \models_{pw} \eta$).

   Therefore $\exists l' : l' > j, t_{l'} - t_j \in I, \sigma^{p,n}, l' \models_{pw} \psi, \forall j'' : j < j'' < l', \sigma^{p,n}, j'' \models_{pw} \eta$, and hence $\sigma^{p,n}, j \models_{pw} \varphi$.

   Suppose $t_{i'} - t_i \in (1, 1+p)$.

   (a) $(1, 1+p) \subseteq I$. Let r' be the rightmost point in Y_k^2. Since $t_{r'} - t_j \in (1, 1+p)$, $t_{r'} - t_j \in I$.

   (b) $i' \in Y_k^2$ and $\sigma^{p,n}, i' \models_{pw} \psi$. By induction hypothesis $\sigma^{p,n}, r' \models_{pw} \psi$ since $r' \in Y_k^2$.

(c) Since i is in X^2_{k+1} and i' in Y^2_k, the rightmost position in X^2_k, r satisfies η. Hence every position in X^2_k satisfies η. For all j'', $j < j'' \leq r$, $\sigma^{p,n}, j'' \models_{pw} \eta$.

(d) For j'', $r < j'' < l'$, $\sigma^{p,n}, j'' \models_{pw} \eta$ (since $i < r$, $r < l'$ and $l' \leq i'$, and for all i'', $i < i'' < i'$, $\sigma^{p,n}, i'' \models_{pw} \eta$).

(e) $l' < i'$ since $t_{i'} - t_i \in (1, 1+p)$ and hence $t_{i'} - t_i > 1$. $\sigma^{p,n}, l' \models_{pw} \eta$ since $i < r$, $r < l'$ and $l' < i'$, and for all i'', $i < i'' < i'$, $\sigma^{p,n}, i'' \models_{pw} \eta$. Hence η is satisfied at every position in Y^2_k. So, for $l' \leq j'' < r'$, $\sigma^{p,n}, j'' \models_{pw} \eta$.

Therefore $\exists r' : r' > j, t_{r'} - t_j \in I, \sigma^{p,n}, r' \models_{pw} \psi, \forall j'' : j < j'' < r', \sigma^{p,n}, j'' \models_{pw} \eta$, and hence $\sigma^{p,n}, j \models_{pw} \varphi$.

5. $i' \in X^2_k$: $(0,p) \subseteq I$. Since $t_{j+1} - t_j \in (0,p)$, $t_{j+1} - t_j \in I$. Since $\sigma^{p,n}, i' \models_{pw} \psi$, every position in X^2_k satisfies ψ. Since $j+1$ is in X^2_k, $\sigma^{p,n}, j+1 \models_{pw} \psi$. So, $\exists j+1 : t_{j+1} - t_j \in I, \sigma^{p,n}, j+1 \models_{pw} \psi, \forall j'' : j < j'' < j+1$ (in fact no such i'' exists) $\sigma^{p,n}, j'' \models_{pw} \eta$. Hence $\sigma^{p,n}, j \models_{pw} \varphi$.

Suppose $i, j \in Y^2_{k+1}$. $\sigma^{p,n}, i \models_{pw} \eta U_I \psi \Rightarrow \exists i' > i : t_{i'} - t_i \in I, \sigma^{p,n}, i' \models_{pw} \psi, \forall j'' : i < j'' < i', \sigma^{p,n}, j'' \models_{pw} \eta$.

1. $i' \in Y^3_k$:

(a) $(0,p) \subseteq I$. $t_{i'} - t_j \in (0,p)$, hence $t_{i'} - t_j \in I$.

(b) Since i is in Y^2_{k+1} and i' is greater than every number in Y^2_k, the rightmost position in Y^2_k, r' satisfies η. Hence every position in Y^2_k satisfies η (by induction hypothesis). For all j'', $j < j'' \leq r'$, $\sigma^{p,n}, j'' \models_{pw} \eta$.

(c) For j'', $r' < j'' < i'$, $\sigma^{p,n}, j'' \models_{pw} \eta$ (since $i < r'$, $r' < i'$ and for all i'', $i < i'' < i'$, $\sigma^{p,n}, i'' \models_{pw} \eta$).

Therefore $\exists i' : i' > j, t_{i'} - t_j \in I, \sigma^{p,n}, i' \models_{pw} \psi, \forall j'' : j < j'' < i', \sigma^{p,n}, j'' \models_{pw} \eta$, and hence $\sigma^{p,n}, j \models_{pw} \varphi$.

2. $i' \in Y^2_k$: $(0,p) \subseteq I$. Since $t_{j+1} - t_j \in (0,p)$, $t_{j+1} - t_j \in I$. Since $\sigma^{p,n}, i' \models_{pw} \psi$, every position in Y^2_k satisfies ψ. Since $j+1$ is in Y^2_k, $\sigma^{p,n}, j+1 \models_{pw} \psi$. So, $\exists j+1 : t_{j+1} - t_j \in I, \sigma^{p,n}, j+1 \models_{pw} \psi, \forall j'' : j < j'' < j+1$ (in fact no such j'' exists) $\sigma^{p,n}, j'' \models_{pw} \eta$. Hence $\sigma^{p,n}, j \models_{pw} \varphi$.

Case $\varphi = \eta S_I \psi$: Assume that $0 \notin I$. Suppose $i, j \in X^2_{k+1}$. $\sigma^{p,n}, i \models_{pw} \eta S_I \psi \Rightarrow \exists 0 \leq i' < i : t_i - t_{i'} \in I, \sigma^{p,n}, i' \models_{pw} \psi, \forall i'' : i' < i'' < i, \sigma^{p,n}, i'' \models_{pw} \eta$.

1. $i' \in X_k^1$:

   (a) $(0,p) \subseteq I$. $t_j - t_{i'} \in (0,p)$, hence $t_j - t_{i'} \in I$.

   (b) Since i is in X_{k+1}^2 and i' in X_k^1, the leftmost position in X_k^2, l satisfies η. Hence every position in X_k^2 satisfies η (by induction hypothesis). For all j'', $l \leq j'' < j$, $\sigma^{p,n}, j'' \models_{pw} \eta$.

   (c) For j'', $i' < j'' < l$, $\sigma^{p,n}, j'' \models_{pw} \eta$ (since $i' < l$, $l < i$ and for all i'', $i' < i'' < i$, $\sigma^{p,n}, i'' \models_{pw} \eta$).

   Therefore $\exists i' : 0 \leq i' < j, t_j - t_{i'} \in I, \sigma^{p,n}, i' \models_{pw} \psi, \forall j'' : i' < j'' < j, \sigma^{p,n}, j'' \models_{pw} \eta$, and hence $\sigma^{p,n}, j \models_{pw} \varphi$.

2. $i' \in X_k^2$: $(0,p) \subseteq I$. Since $t_j - t_{j-1} \in (0,p)$, $t_j - t_{j-1} \in I$. Since $\sigma^{p,n}, i' \models_{pw} \psi$, every position in X_k^2 satisfies ψ. Since $j-1$ is in X_k^2, $\sigma^{p,n}, j-1 \models_{pw} \psi$. So, $\exists j-1 : t_j - t_{j-1} \in I, \sigma^{p,n}, j-1 \models_{pw} \psi, \forall j'' : j-1 < j'' < j$ (in fact no such j'' exists) $\sigma^{p,n}, j'' \models_{pw} \eta$. Hence $\sigma^{p,n}, j \models_{pw} \varphi$.

3. $i' = 0$.

   (a) $(1-p,1) \subseteq I$. $t_j - t_0 \in (1-p,1)$, hence $t_j - t_0 \in I$.

   (b) Since i is in X_{k+1}^2 and $i' = 0$, the leftmost position in X_k^2, l satisfies η. Hence every position in X_k^2 satisfies η (by induction hypothesis). For all j'', $l \leq j'' < j$, $\sigma^{p,n}, j'' \models_{pw} \eta$.

   (c) For j'', $0 < j'' < l$, $\sigma^{p,n}, j'' \models_{pw} \eta$ (since $i' = 0$, $0 < l$, $l < i$ and for all i'', $i' < i'' < i$, $\sigma^{p,n}, i'' \models_{pw} \eta$).

   Therefore $\exists i' : 0 \leq i' < j, t_j - t_{i'} \in I, \sigma^{p,n}, i' \models_{pw} \psi, \forall j'' : i' < j'' < j, \sigma^{p,n}, j'' \models_{pw} \eta$, and hence $\sigma^{p,n}, j \models_{pw} \varphi$.

Suppose $i, j \in Y_{k+1}^2$. $\sigma^{p,n}, i \models_{pw} \eta S_I \psi \Rightarrow \exists 0 \leq i' < i : t_i - t_{i'} \in I, \sigma^{p,n}, i' \models_{pw} \psi, \forall i'' : i' < i'' < i, \sigma^{p,n}, i'' \models_{pw} \eta$.

1. $i' \in Y_k^1$:

   (a) $(0,p) \subseteq I$. $t_j - t_{i'} \in (0,p)$, hence $t_j - t_{i'} \in I$.

   (b) Since i is in Y_{k+1}^2 and i' is smaller than every number in Y_k^2, the leftmost position in Y_k^2, l' satisfies η. Hence every position in Y_k^2 satisfies η (by induction hypothesis). For all j'', $l' \leq j'' < j$, $\sigma^{p,n}, j'' \models_{pw} \eta$.

   (c) For j'', $i' < j'' < l'$, $\sigma^{p,n}, j'' \models_{pw} \eta$ (since $i' < l'$, $l' < i$ and for all i'', $i' < i'' < i$, $\sigma^{p,n}, i'' \models_{pw} \eta$).

Therefore $\exists i' : 0 < i' < j, t_j - t_{i'} \in I, \sigma^{p,n}, i' \models_{pw} \psi, \forall j'' : i' < j'' < j, \sigma^{p,n}, j'' \models_{pw} \eta$, and hence $\sigma^{p,n}, j \models_{pw} \varphi$.

2. $i' \in X_k^1$: The argument is similar to the above except that $1(a)$ needs to be replaced by: $(1, 1+p) \subseteq I$. $t_j - t_{i'} \in (1, 1+p)$, hence $t_j - t_{i'} \in I$.

3. $i' \in X_k^3$: The argument is similar to 1 except that $1(a)$ needs to be replaced by: $(1-p, 1) \subseteq I$. $t_j - t_{i'} \in (1-p, 1)$, hence $t_j - t_{i'} \in I$.

4. $i' \in X_k^2$: Suppose $t_i - t_{i'} \in (1-p, 1)$.

   (a) $(1-p, 1) \subseteq I$. Let r be the rightmost point in X_k^2. Since $t_j - t_r \in (1-p, 1)$, $t_j - t_r \in I$.

   (b) $i' \in X_k^2$ and $\sigma^{p,n}, i' \models_{pw} \psi$. By induction hypothesis $\sigma^{p,n}, r \models_{pw} \psi$ since $r \in X_k^2$.

   (c) Since i is in Y_{k+1}^2 and i' in X_k^2, the leftmost position in Y_k^2, l' satisfies η. Hence every position in Y_k^2 satisfies η. For all j'', $l' \leq j'' < j$, $\sigma^{p,n}, j'' \models_{pw} \eta$.

   (d) For j'', $r < j'' < l'$, $\sigma^{p,n}, j'' \models_{pw} \eta$ (since $i' \leq r$, $r < l'$, $l' < i$ and for all i'', $i' < i'' < i$, $\sigma^{p,n}, i'' \models_{pw} \eta$).

   Therefore $\exists r : 0 < r < j, t_j - t_r \in I, \sigma^{p,n}, r \models_{pw} \psi, \forall j'' : r < j'' < j, \sigma^{p,n}, j'' \models_{pw} \eta$, and hence $\sigma^{p,n}, j \models_{pw} \varphi$.

   Suppose $t_i - t_{i'} \in (1, 1+p)$.

   (a) $(1, 1+p) \subseteq I$. Let l be the leftmost point in X_k^2. Since $t_j - t_l \in (1, 1+p)$, $t_j - t_l \in I$.

   (b) $i' \in X_k^2$ and $\sigma^{p,n}, i' \models_{pw} \psi$. By induction hypothesis $\sigma^{p,n}, l \models_{pw} \psi$ since $l \in X_k^2$.

   (c) Since i is in Y_{k+1}^2 and i' in X_k^2, the leftmost position in Y_k^2, l' satisfies η. Hence every position in Y_k^2 satisfies η. For all j'', $l' \leq j'' < j$, $\sigma^{p,n}, j'' \models_{pw} \eta$.

   (d) For j'', $r < j'' < l'$, $\sigma^{p,n}, j'' \models_{pw} \eta$ (since $i' < r$, $r < l'$ and $l' < i$ and for all i'', $i' < i'' < i$, $\sigma^{p,n}, i'' \models_{pw} \eta$).

   (e) $i' < r$ since $t_i - t_{i'} \in (1, 1+p)$ and hence $t_i - t_{i'} > 1$. $\sigma^{p,n}, r \models_{pw} \eta$ since $i' < r$, $r < l'$ and $l' < i$ and for all i'', $i' < i'' < i$, $\sigma^{p,n}, i'' \models_{pw} \eta$. Hence η is satisfied at every position in X_k^2. So, for $l < j'' \leq r$, $\sigma^{p,n}, j'' \models_{pw} \eta$.

   Therefore $\exists l : 0 < l < j, t_j - t_l \in I, \sigma^{p,n}, l \models_{pw} \psi, \forall j'' : l < j'' < j, \sigma^{p,n}, j'' \models_{pw} \eta$, and hence $\sigma^{p,n}, j \models_{pw} \varphi$.

5. $i' \in Y_k^2$: $(0,p) \subseteq I$. Since $t_j - t_{j-1} \in (0,p)$, $t_j - t_{j-1} \in I$. Since $\sigma^{p,n}, i' \models_{pw} \psi$, every position in Y_k^2 satisfies ψ. Since $j-1$ is in Y_k^2, $\sigma^{p,n}, j-1 \models_{pw} \psi$. So, $\exists j-1 : t_j - t_{j-1} \in I, \sigma^{p,n}, j-1 \models_{pw} \psi, \forall j'' : 0 < j-1 < j'' < j$ (in fact no such j'' exists) $\sigma^{p,n}, j'' \models_{pw} \eta$. Hence $\sigma^{p,n}, j \models_{pw} \varphi$.

6. $i' = 0$.

   (a) $(2-p,2) \subseteq I$. $t_j - t_{i'} \in (2-p,2)$, hence $t_j - t_{i'} \in I$.

   (b) Since i is in Y_{k+1}^2 and $i' = 0$ the leftmost position in Y_k^2, l' satisfies η. Hence every position in Y_k^2 satisfies η (by induction hypothesis). For all j'', $l' \leq j'' < j$, $\sigma^{p,n}, j'' \models_{pw} \eta$.

   (c) For j'', $i' < j'' < l'$, $\sigma^{p,n}, j'' \models_{pw} \eta$ (since $i' < l'$, $l' < i$ and for all i'', $i' < i'' < i$, $\sigma^{p,n}, i'' \models_{pw} \eta$).

   Therefore $\exists i' : 0 < i' < j, t_j - t_{i'} \in I, \sigma^{p,n}, i' \models_{pw} \psi, \forall j'' : i' < j'' < j, \sigma^{p,n}, j'' \models_{pw} \eta$, and hence $\sigma^{p,n}, j \models_{pw} \varphi$.

This completes the proof of Lemma 6.1. □

We define a partial function $h : \mathbb{N} \to \mathbb{N}$ which is defined for all $i \in \mathbb{N}$ except for $n+2$. $h(i) = i$ if $i < n+2$ and $h(i) = i-1$ if $i > n+2$. $h(i)$ is the position in $\rho^{p,n}$ corresponding to the position i in $\sigma^{p,n}$ in the sense that the time of the $h(i)$-th action in $\rho^{p,n}$ is the same as that of the i-th action in $\sigma^{p,n}$ (hence it is not defined for $n+2$).

Lemma 6.2 *Let $k \in \mathbb{N}$, $0 \leq k \leq n$ and let $\varphi \in \mathrm{MTL}_{S_I}(p,k)$. For all $i, j \in X_k^2 - \{n+2\}$, $\rho^{p,n}, h(i) \models_{pw} \varphi$ iff $\rho^{p,n}, h(j) \models_{pw} \varphi$ and for all $i, j \in Y_k^2$, $\rho^{p,n}, h(i) \models_{pw} \varphi$ iff $\rho^{p,n}, h(j) \models_{pw} \varphi$.*

Proof Proof is similar to that of the previous lemma. □

Corollary 6.1 *Let $\varphi \in \mathrm{MTL}_{S_I}(p,n)$. Then*

1. *$\sigma^{p,n}, n+1 \models_{pw} \varphi$ iff $\sigma^{p,n}, n+2 \models_{pw} \varphi$ iff $\sigma^{p,n}, n+3 \models_{pw} \varphi$, and*
2. *$\rho^{p,n}, h(n+1) \models_{pw} \varphi$ iff $\rho^{p,n}, h(n+3) \models_{pw} \varphi$.* □

Lemma 6.3 *For any $\varphi \in \mathrm{MTL}_{S_I}(p,n)$ and $i \in \mathbb{N}$, where $i \neq n+2$, $\sigma^{p,n}, i \models_{pw} \varphi$ iff $\rho^{p,n}, h(i) \models_{pw} \varphi$.*

Proof Let $\sigma^{p,n} = (a_1, t_1) \cdots (a_{4n+5}, t_{4n+5})$ and $\rho^{p,n} = (a'_1, t'_1) \cdots (a'_{4n+4}, t'_{4n+4})$. Proof is by induction on the structure of φ. If φ is atomic or of the form $\neg\psi$ or $\eta \vee \psi$, then it is straight forward. Let us look at the case when $\varphi = \eta U_I \psi$ (assume $0 \notin I$).

$\sigma^{p,n}, i \models_{pw} \eta U_I \psi$ iff $\exists j \geq i : t_j - t_i \in I, \sigma^{p,n}, j \models_{pw} \psi, \forall j' : i < j' < j, \sigma^{p,n}, j' \models_{pw} \eta$.

Suppose $j \neq n+2$. $h(j) \geq h(i)$ (definition of h, $i, j \neq n+2$ and $j \geq i$). For all j'': $h(i) < j'' < h(j)$, $\rho^{p,n}, j'' \models_{pw} \eta$ (since $j'' = h(j')$ for some $i < j' < j$ and from induction hypothesis). $\rho^{p,n}, h(j) \models_{pw} \psi$ (by induction hypothesis). $t'_{h(j)} - t'_{h(i)} \in I$ since $t'_{h(j)} = t_j$ and $t'_{h(i)} = t_i$. Hence $\exists h(j) : h(j) \geq h(i), t'_{h(j)} - t'_{h(i)} \in I, \rho^{p,n}, h(j) \models_{pw} \psi, \forall j'' : h(i) < j'' < h(j), \rho^{p,n}, j'' \models_{pw} \eta$. $\rho^{p,n}, h(i) \models_{pw} \varphi$.

Suppose $j = n+2$ and $0 < i < n+2$. Then $0 < t_j - t_i < p$ and $(0, p) \subseteq I$. $t_{j+1} - t_i \in (0, p)$ and hence $t_{j+1} - t_i \in I$. $\sigma^{p,n}, j+1 \models_{pw} \psi$ (by corollary 6.1) $\exists t'_{h(j+1)} : t'_{h(j+1)} - t'_{h(i)} \in I, \rho^{p,n}, h(j+1) \models_{pw} \psi, \forall j' : h(i) < j' < h(j+1), \rho^{p,n}, j' \models_{pw} \eta$ (since every j' is $h(j'')$ for some $i < j'' < j$). Hence $\rho^{p,n}, h(i) \models_{pw} \varphi$.

If $j = n+2$ and $i = 0$, then the argument is the same as above except that now $t_j - t_i \in (1-p, 1)$.

In the other direction, $\rho^{p,n}, h(i) \models_{pw} \eta U_I \psi$ iff $\exists h(j) \geq h(i) : t'_{h(j)} - t'_{h(i)} \in I, \rho^{p,n}, h(j) \models_{pw} \psi, \forall h(i) < j' < h(j) : \rho^{p,n}, j' \models_{pw} \eta$.

Suppose $j \neq n+3$. Then $\forall i < j'' < j, \sigma^{p,n}, j' \models_{pw} \eta$ since either $n+2 \notin \{i+1, \cdots, j-1\}$ or $n+3 \in \{i+1, \cdots, j-1\}$ (in which case $\sigma^{p,n}, n+3 \models_{pw} \eta$ and hence by corollary 6.1 $\sigma^{p,n}, n+2 \models_{pw} \eta$). Hence $\exists j \geq i : t_j - t_i \in I, \sigma^{p,n}, j \models_{pw} \psi, \forall i < j'' < j'' \sigma^{p,n}, j'' \models_{pw} \eta$. Hence $\sigma^{p,n}, i \models_{pw} \eta$.

If $j = n+3$ and $0 < i < n+2$, then $t_{n+3} - t_i \in (0, p)$ and $(0, p) \subseteq I$. $t_{n+2} - t_i \in (0, p)$ and hence $t_{n+2} - t_i \in I$. So, $\exists n+2 : t_{n+2} - t_i \in I, \sigma^{p,n}, n+2 \models_{pw} \psi$ (by corollary 6.1), $\forall i < j'' < n+2, \sigma^{p,n}, j'' \models_{pw} \eta$ (by induction hypothesis). Hence $\sigma^{p,n}, i \models_{pw} \varphi$.

If $j = n+3$ and $i = 0$, then the argument is similar except that $t_{n+3} - t_i \in (1-p, 1)$.

Let us now look at $\varphi = \eta S_I \psi$ (where $0 \notin I$). In one direction if $\sigma^{p,n}, i \models_{pw} \eta S_I \psi$, then there is some $j < i$ such that $t_i - t_j \in I$ and all $j' : j < j' < i$ satisfy η.

If $j \neq n+2$, then there exists $h(j)$ such that $\rho^{p,n}, h(j) \models_{pw} \psi$ and for all $j'' : h(j) < j'' < h(i)$ $\rho^{p,n}, j'' \models_{pw} \eta$. Hence $\rho^{p,n}, i \models_{pw} \varphi$.

Suppose $j = n+2$. I is not singular since there is no $i' > j$ such that $t'_i - t_j = cp$ where $c \in \mathbb{N}$. Hence either $j' = n+1$ or $n+3$ satisfies $t_i - t_{j'} \in I$. So, $\exists h(j') : t'_{h(i)} - t'_{h(j')} \in I, \rho^{p,n}, h(j') \models_{pw} \psi, \forall j'' : j' < j'' < i, \rho^{p,n}, h(j'') \models_{pw} \eta$. Hence $\rho^{p,n}, h(i) \models_{pw} \varphi$.

In the other direction, $\rho^{p,n}, h(i) \models_{pw} \eta S_I \psi$ iff $\exists h(j) : j < i, t'_{h(i)} - t'_{h(j)} \in I, \rho^{p,n}, h(j) \models_{pw} \psi, \forall j' : h(j) < j' < h(i), \rho^{p,n}, j' \models_{pw} \eta$.

Suppose $j \neq n+1$. Then either $n+2 \notin \{j+1, \cdots, i-1\}$ or $n+1 \in \{j+1, \cdots, i-1\}$ (in which case $\sigma^{p,n}, n+2 \models_{pw} \eta$ and hence by corollary 6.1 $\sigma^{p,n}, n+1 \models_{pw} \eta$). $\exists j : j < i, t_i - t_j \in I, \sigma^{p,n}, j \models_{pw} \psi, \forall j'' : j < j'' < i, \sigma^{p,n}, j'' \models_{pw} \eta$. Hence $\sigma^{p,n}, i \models_{pw} \varphi$.

Suppose $j = n+1$ and $i \neq n+3$. Then $n+2, n+3 \in \{j+1, \cdots, i-1\}$. $\sigma^{p,n}, n+3 \models_{pw} \eta$. Hence by corollary 6.1, $\sigma^{p,n}, n+2 \models_{pw} \eta$. $\exists j : j < i, t_i - t_j \in I, \sigma^{p,n}, j \models_{pw} \psi, \forall j'' : j < j'' < i, \sigma^{p,n}, j'' \models_{pw} \eta$. Hence $\sigma^{p,n}, i \models_{pw} \varphi$.

Suppose $j = n+1$ and $i = n+3$. $t_i - t_j \in (0, p)$ and hence $(0, p) \subseteq I$. $\exists j+1 : t_i - t_{j+1} \in I, \sigma^{p,n}, j+1 \models_{pw} \psi$ (by corollary 6.1), $\forall j'' : j+1 < j'' < i$ (in fact there exists no such j''), $\sigma^{p,n}, j'' \models_{pw} \eta$. Hence $\sigma^{p,n}, i \models_{pw} \varphi$. □

Corollary 6.2 *For any $\varphi \in \mathrm{MTL}_{S_I}(p, n)$, $\sigma^{p,n}, 0 \models_{pw} \varphi$ iff $\rho^{p,n}, 0 \models_{pw} \varphi$.* □

Theorem 6.1 *The timed language L_{2ins} is expressible by MTL^c but not by $\mathrm{MTL}_{S_I}^{pw}$ over finite timed words.*

Proof Suppose that there exists an MTL_{S_I} formula φ which defines L_{2ins} in the pointwise semantics. $\varphi \in \mathrm{MTL}_{S_I}(p, n)$ for some p and n. Hence it would not distinguish between $\sigma^{p,n}$ and $\rho^{p,n}$, whereas exactly one of them is in L_{2ins}, which is a contradiction. However the following MTL formula in the continuous semantics defines L_{2ins}:
$\Diamond(a \wedge \neg\varphi_{action} U(\neg\varphi_{action} \wedge \Diamond_{[1,1]} a \wedge \neg\varphi_{action} U_{(0,\infty)}(\neg\varphi_{action} \wedge \Diamond_{[1,1]} a \wedge \neg\varphi_{action} U a)))$.
□

Corollary 6.3 MTL^{pw} *is strictly contained in* MTL^c, MTL_S^{pw} *is strictly contained in* MTL_S^c *and* $\mathrm{MTL}_{S_I}^{pw}$ *is strictly contained in* $\mathrm{MTL}_{S_I}^c$ *over finite timed words.* □

Theorem 6.2 *The timed language L_{2ins} is expressible by MTL^c but not by $\mathrm{MTL}_{S_I}^{pw}$ over infinite timed words.*

Proof We extend the above results for the case of infinite words by replacing the finite models above, by infinite models which are similar to their counterparts in the interval $[0, 2]$ but contain a b at every integer time greater than 2. We replace $\sigma^{p,n}$ and $\rho^{p,n}$ by the infinite words $\alpha^{p,n}$ and $\beta^{p,n}$, respectively.

$\alpha^{p,n} = (a, 1-p+d/2)(a, 1-p+3d/2)\cdots(a, 1-p/2-d)(a, 1-p/2)(a, 1-p/2+d)\cdots(a, 1-d/2)(a, 2-p+d)(a, 2-p+2d)\cdots(a, 2-d)(b, 3)(b, 4)\cdots$.
$\beta^{p,n} = (a, 1-p+d/2)(a, 1-p+3d/2)\cdots(a, 1-p/2-d)(a, 1-p/2+d)\cdots(a, 1-d/2)(a, 2-p+d)(a, 2-p+2d)\cdots(a, 2-d)(b, 3)(b, 4)\cdots$.

In the proof of Lemma 6.1 we need to consider some extra cases corresponding to the satisfaction of ψ at one of these newly introduced points. The argument for the case where $\varphi = \eta S_I \psi$ is similar because no new action points are introduced in the interval $[0, 2]$. In the case of $\varphi = \eta U_I \psi$ and $i \in X^2_{k+1}$, if i' is such that $t_{i'} = m$, $m > 2$ and $m \in \mathbb{N}$ (at some position corresponding to a b), then the argument is similar to 1 except that we need to replace $1(a)$ by: $(m-1-p, m-1) \subseteq I$, $t_{i'} - t_j \in (m-1-p, m-1)$, hence $t_{i'} - t_j \in I$.

In the case when $\varphi = \eta U_I \psi$ and $i \in Y^2_{k+1}$, if i' is such that $t_{i'} = m$, $m > 2$ and $m \in \mathbb{N}$), then the argument is similar to 1, except that $1(a)$ needs to be replaced by: $(m-2-p, m-2) \subseteq I$, $t_{i'} - t_j \in (m-2-p, m-2)$, hence $t_{i'} - t_j \in I$.

For Theorem 6.3, the same proof goes through. Hence L_{2ins} can not be expressed by any MTL_{S_I} formula in the pointwise semantics over infinite words. But the formula which expressed L_{2ins} over finite words also expresses the same over finite words. □

Corollary 6.4 MTL^{pw} *is strictly contained in* MTL^{c}, MTL^{pw}_{S} *is strictly contained in* MTL^{c}_{S} *and* $\mathrm{MTL}^{pw}_{S_I}$ *is strictly contained in* $\mathrm{MTL}^{c}_{S_I}$ *over infinite timed words.* □

6.2 L_{em} not expressible by MTL^{pw}_{S}

In this section, we show that the language L_{em} is not expressible by MTL_S in the pointwise semantics over finite and infinite timed words. L_{em} (for "exact match") is the timed language over $\{a, b\}$ which consists of timed words in which, for every b in the interval $(0, 1)$, there is a b in the future which is at time distance 1 from it, and for every b in the interval $(1, 2)$, there is a b in the past which is at time distance 1 from it. The proof is similar to the one for inexpressibility of L_{2ins} by $\mathrm{MTL}^{pw}_{S_I}$ and we sketch it below.

Let $p = 1/q$, where $q \in \mathbb{N}$ and $q > 0$, and let $n \in \mathbb{N}$. We give two finite models $\sigma^{p,n}$ and $\rho^{p,n}$ such that $\sigma^{p,n}$ is in L_{em} and $\rho^{p,n}$ is not in L_{em}. Further we prove that no $\mathrm{MTL}_S(p, n)$ formula can distinguish between $\sigma^{p,n}$ and $\rho^{p,n}$ in the pointwise semantics, where $\mathrm{MTL}_S(p, n)$ formula is an MTL_S formula

with granularity p and an U and S nesting depth of n. Let $x = 1 - p$,

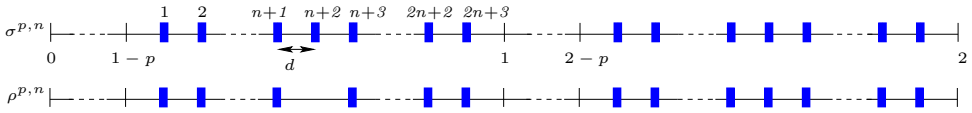
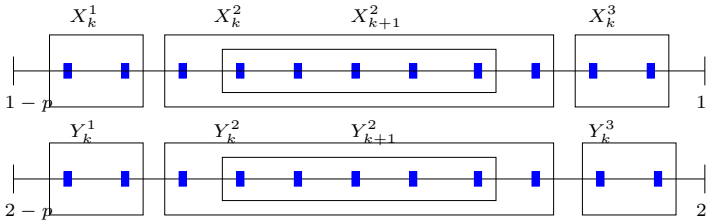


Figure 6.2: Models for showing inexpressibility of L_{em}.

$y = 2 - p$ and $d = p/(2n + 4)$.
Then $\sigma^{p,n} = (b, x+d)(b, x+2d)\cdots(b, x+(n+1)d)(b, x+(n+2)d)(b, x+(n+3)d)\cdots(b, x+(2n+2)d)(b, x+(2n+3)d)(b, y+d)(b, y+2d)\cdots(b, y+(n+1)d)(b, y+(n+2)d)(b, y+(n+3)d)\cdots(b, y+(2n+2)d)(b, y+(2n+3)d)$.
And $\rho^{p,n} = (b, x+d)(b, x+2d)\cdots(b, x+(n+1)d)(b, x+(n+3)d)\cdots(b, x+(2n+2)d)(b, x+(2n+3)d)(b, y+d)(b, y+2d)\cdots(b, y+(n+1)d)(b, y+(n+2)d)(b, y+(n+3)d)\cdots(b, y+(2n+2)d)(b, y+(2n+3)d)$.



Lemma 6.4 *Let $k \in \mathbb{N}$, $0 \leq k \leq n$ and let $\varphi \in \mathrm{MTL}_S(p, k)$. Let $X_k^2 = \{k+1, \cdots, 2n+3-k\}$ and $Y_k^2 = \{(2n+3)+k+1, \cdots, (2n+3)+2n+3-k\}$. Then for all $i, j \in X_k^2$, $\sigma^{p,n}, i \models_{pw} \varphi$ iff $\sigma^{p,n}, j \models_{pw} \varphi$ and for all $i, j \in Y_k^2$, $\sigma^{p,n}, i \models_{pw} \varphi$ iff $\sigma^{p,n}, j \models_{pw} \varphi$.* □

Let h be the function defined in the last section.

Lemma 6.5 *Let $k \in \mathbb{N}$, $0 \leq k \leq n$ and let $\varphi \in \mathrm{MTL}_S(p, k)$. Then for all $i, j \in X_k^2 - \{n+2\}$, $\rho^{p,n}, h(i) \models_{pw} \varphi$ iff $\rho^{p,n}, h(j) \models_{pw} \varphi$ and for all $i, j \in Y_k^2$, $\rho^{p,n}, h(i) \models_{pw} \varphi$ iff $\rho^{p,n}, h(j) \models_{pw} \varphi$.* □

Corollary 6.5 *Let $\varphi \in \mathrm{MTL}_S(p, n)$. Then $\sigma^{p,n}, n+1 \models_{pw} \varphi$ iff $\sigma^{p,n}, n+2 \models_{pw} \varphi$ iff $\sigma^{p,n}, n+3 \models_{pw} \varphi$. Similarly, $\rho^{p,n}, h(n+1) \models_{pw} \varphi$ iff $\rho^{p,n}, h(n+3) \models_{pw} \varphi$.* □

Lemma 6.6 *For any* $\varphi \in \mathrm{MTL}_S(p,n)$ *and* $i \in \mathbb{N}$*, where* $i \neq n{+}2$*,* $\sigma^{p,n}, i \models_{pw} \varphi$ *iff* $\rho^{p,n}, h(i) \models_{pw} \varphi$*.* □

Corollary 6.6 *For any* $\varphi \in \mathrm{MTL}_S(p,n)$*,* $\sigma^{p,n}, 0 \models_{pw} \varphi$ *iff* $\rho^{p,n}, 0 \models_{pw} \varphi$*.* □

Theorem 6.3 *The timed language* L_{em} *is expressible by* MTL^c *and* $\mathrm{MTL}^{pw}_{S_I}$ *but not by* MTL^{pw}_{S} *over finite timed words.*

Proof It follows from corollary 6.6 that L_{em} is not expressible in MTL^{pw}_{S}. However the MTL_{S_I} formula, $\Box_{(0,1)}(b \Rightarrow \Diamond_{[1,1]} b) \wedge \Box_{(1,2)}(b \Rightarrow \Diamond\!\!\!\!-_{[1,1]} b)$, in the pointwise semantics and the MTL formula $\Box_{(0,1)}(b \leftrightarrow \Diamond_{[1,1]} b)$, in the continuous semantics express L_{em}. □

The result can be extended for the case of infinite words in a manner similar to that done in the previous section.

Theorem 6.4 *The timed language* L_{em} *is expressible by* MTL^c *and* $\mathrm{MTL}^{pw}_{S_I}$ *but not by* MTL^{pw}_{S} *over infinite timed words.* □

6.3 Conclusion

In this chapter we exhibited languages not expressible by MTL with S and S_I operators in the pointwise semantics. The inexpressibility results imply that even with past operators MTL in the pointwise semantics does not attain the power of MTL without the past operators in the continuous semantics. The results also lead to the strict containment of the various extensions of MTL with past operators in the pointwise semantics, in their corresponding continuous versions. We also obtain the result that MTL_S is strictly contained in MTL_{S_I} in the pointwise semantics. These results hold for both finite and infinite words. We note that we do not yet know if MTL_S is strictly contained MTL_{S_I} in the continuous semantics for either finite or infinite words.

Chapter 7

An overview of relative expressiveness

In this chapter we compile the relative expressiveness results which can be obtained from the results in the previous chapters.

7.1 Venn diagram for relative expressiveness

The Venn diagram below depicts the sets MTL^{pw}, MTL_S^{pw} and $\mathrm{MTL}_{S_I}^{pw}$, and MTL^{c}, MTL_S^{c} and $\mathrm{MTL}_{S_I}^{c}$, as one contained in the next. It also shows that MTL^{pw}, MTL_S^{pw} and $\mathrm{MTL}_{S_I}^{pw}$ are contained in MTL^{c}, MTL_S^{c} and $\mathrm{MTL}_{S_I}^{c}$, respectively.

We describe below languages over the alphabet $\Sigma = \{a, b, c, d\}$ which are present in various regions of the diagram. Since the diagram is the same for both finite and infinite words, we present just one.

1. L_a is the timed language over Σ whose timed words contain at least one a. The MTL formula $\top U_{[0,\infty)} a$ expresses this language in both the pointwise and continuous semantics.

2. $L_{\mathbb{2}b}$ is the language over Σ which consists of timed words over $\{b\}$ which contain at least 2 b's in the interval $(0, 1)$. From Theorem 4.3 and Theorem 4.4, we have that $L_{\mathbb{2}b}$ is in $(\mathrm{MTL}^{c} \cap \mathrm{MTL}_S^{pw}) - \mathrm{MTL}^{pw}$.

3. L_{em} is the language over Σ which consists of timed words over $\{a, b\}$ in which the number of b's in the interval $(0, 1)$ is the same as that in the interval $(1, 2)$, and corresponding to every b in the interval $(0, 1)$, there is a b in the interval $(1, 2)$ which is exactly at time distance 1 from it. Theorem 6.3 and Theorem 6.4 show that L_{em} is in $(\mathrm{MTL}^{c} \cap \mathrm{MTL}_{S_I}^{pw}) - \mathrm{MTL}_S^{pw}$.

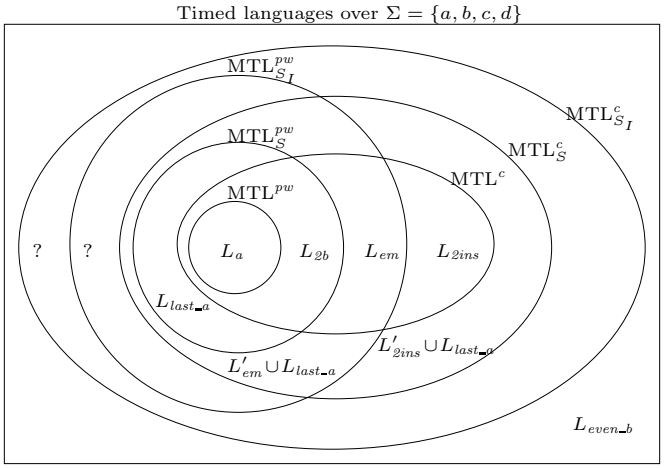


Figure 7.1: Venn diagram for relative expressiveness.

4. L_{2ins} is the language over Σ which consists of timed words over $\{a, b\}$ which contain two consecutive a's such that there are two distinct points between there time of occurrence at distance 1 from each of which there is an a. From Theorem 6.1 and Theorem 6.2, it follows that L_{2ins} is in $\mathrm{MTL}^{c} - \mathrm{MTL}^{pw}_{S_I}$.

5. L_{last_a} is the language over Σ which consists of timed words over $\{a, b\}$ in which there is a symbol at time 1 which is immediately preceded by an a. Theorem 5.3 and Theorem 5.4 show that L_{last_a} is in $\mathrm{MTL}^{pw}_{S} - \mathrm{MTL}^{c}$.

6. L'_{em} is the language over Σ which consists of timed words obtained from those of L_{em} by replacing a's and b's by c's and d's, respectively. Note that L'_{em} and L_{last_a} are disjoints sets. Now we can argue that $L'_{em} \cup L_{last\_a}$ is not expressible by MTL^{pw}_{S}, because if it were, then the conjunction of the formula expressing it and the negation of that expressing L_{last_a} (which we know is expressible by MTL^{pw}_{S}), expresses L'_{em} which is a contradiction to the fact that L'_{em} is not expressible by MTL^{pw}_{S} (since L_{em} is not expressible by it). Similarly $L'_{em} \cup L_{last\_a}$ is not expressible by MTL^{c}. However, it is expressible by both MTL^{c}_{S} and $\mathrm{MTL}^{pw}_{S_I}$ since both L'_{em} and L_{last_a} are expressible by either of them.

7. L'_{2ins} is the language over Σ which consists of timed words obtained from those of L_{2ins} by replacing a's and b's in them by c's and d's. By an argument similar to the above, we can show that $L'_{2ins} \cup L_{last\_a}$ is in $\mathrm{MTL}^{c}_{S} - (\mathrm{MTL}^{pw}_{S_I} \cup \mathrm{MTL}^{c})$.

8. Finally, the language L_{even_b} over Σ which consists of timed words containing even number of b's can be shown to be not expressible by $\mathrm{MTL}^c_{S_I}$.

9. It is not known if the regions $\mathrm{MTL}^{pw}_{S_I} - \mathrm{MTL}^c_S$ and $\mathrm{MTL}^c_{S_I} - (\mathrm{MTL}^{pw}_{S_I} \cup \mathrm{MTL}^c_S)$ are empty.

7.2 Hasse diagram for relative expressiveness

We conclude the part on relative expressiveness by giving the Hasse diagram showing the relative expressiveness of MTL and its variants. We use solid arrows for strict containment, dotted arrows when strict containment is not known but containment is known, dotted lines when the relative expressiveness of the logics is not known, and the absence of an arrow (or transitive arrow) or a line indicates that the expressiveness of the logics is incomparable.

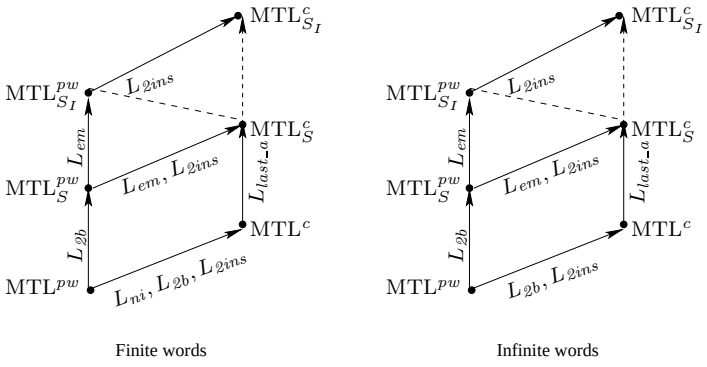


Figure 7.2: Hasse diagram for relative expressiveness

Chapter 8

Continuous input-determined automata

The rest of the chapters in the thesis are devoted to showing the expressive completeness of MTL_S and MTL_{S_I} in the continuous semantics. In the pointwise semantics, the expressive completeness of these two logics has been shown in [9]. A generalized class of timed automata based on input-determined operators, called input-determined automata (IDA's), is considered. The IDA's are shown to be closed under boolean operations. A logical characterization of these automata is given in terms of an MSO logic based on input-determined operators. It is also shown that the first-order fragments of these logics correspond to natural timed temporal logics based on the corresponding input-determined operators and interpreted in a pointwise way. Further, recursive versions of these logics and automata are defined and shown to admit a similar characterization. Finally, the expressive completeness of $\mathrm{MTL}_{S_I}^{pw}$ is shown by showing their equivalence to a recursive timed temporal logic based on the operators $\Diamond$ and $\Diamond\!\!\!\!-$.

Similarly, MTL_S is shown to be expressively complete with respect to the IDA's based on the operator $\Diamond$. These were termed Eventual Timed Automata (ETA's) in [7] as they are based on the eventual operator $\Diamond$. We now discuss a few possible extensions of ETA's through which we intend to show the expressive completeness of MTL_S^c, and also provide some motivation for the choice of the automata we propose.

First we consider the extension of ETA's which allow epsilon-transitions. We can show that these automata cannot express L_{ni} using the technique in chapter 3, since their emptiness problem can be reduced to that of ETA's, and can be decided [7]. However they can express the complement of L_{ni} which is not expressible by ETA's. Thus though they are strictly more expressive than ETA's they are not closed under complementation which is an important

property we expect from our automata. Further, L_{ni} can be expressed by a "non-recursive" fragment of MTL in the continuous semantics, which is the set of MTL formulas we want the automata to at least contain.

The other possible extension is to add invariants to the states but disallow epsilon-transitions. In this framework we do not see a straight-forward way to express the complement of L_{ni}, as intuitively the automaton will need to make an epsilon-transition to guarantee the existence of an a at distance one from some point between two consecutive a actions.

Thus we propose the *Continuous Eventual Timed Automata* (CETA's) which allow epsilon-transitions as well as contain invariants on the states. These automata can express both L_{ni} and its complement. Further these automata are closed under complementation, as will be seen later.

We choose to prove our results for a general class of automata parameterized by a set of input-determined operators. This approach will be useful when we handle the "recursive" case of our formalisms. After proving our results in this general framework, the required results for CETA's and MTL will follow as corollaries. We call this general class of automata *continuous input-determined automata* (*CIDA*'s). In this chapter we show that CIDA's form a robust class of automata which are determinizable and closed under boolean operations. We will deal with only finite words, but the techniques extend to infinite words as well.

8.1 Preliminaries

A *finite state automaton (FSA)* $\mathcal{A}$ over a finite alphabet A is a structure $\mathcal{A} = (Q, s, \delta, F)$, where Q is a finite set of states, s is the initial state, $\delta \subseteq Q \times A \times Q$ is the set of transitions, and $F \subseteq Q$ is the set of final states. A run ρ of $\mathcal{A}$ on a word $w = a_1 \cdots a_n \in A^*$ is a mapping from $\{0, \cdots, n\} \to Q$ such that $\rho(0) = s$ and $(\rho(i), a_{i+1}, \rho(i+1)) \in \delta$ for each $i < n$. The run is accepting if $\rho(n) \in F$. The language accepted by $\mathcal{A}$, denoted $L_{sym}(\mathcal{A})$, is the set of words in A^* over which $\mathcal{A}$ has an accepting run.

Let A be an alphabet and let $f : [0, r] \to A$ be a function, where $r \in \mathbb{R}_{\geq 0}$. We denote r by $length(f)$. We call f a *finitely varying* function over A, if there exist a word $a_0 a_1 \cdots a_{2n}$ in A^*, and an interval sequence $I_0 I_1 \cdots I_{2n}$, such that $0 \in I_0$, I_i and I_{i+1} are adjacent for each i, I_i is singular if i is even, and for all $t \in [0, r]$, $f(t) = a_i$ if $t \in I_i$. We then call $(a_0, I_0) \cdots (a_{2n}, I_{2n})$ an *interval representation* of f. We define $func(A)$ to be the set of all finitely varying functions over A. An interval representation $(b_0, I_0) \cdots (b_{2n}, I_{2n})$ of f is called *canonical*, if there does not exist an i such that $0 < i < n$ and $b_{2i-1} = b_{2i} = b_{2i+1}$. Note that every finitely varying function has a canonical

interval representation.

Let $f \in \mathit{func}(A)$ and let $(a_0, I_0) \cdots (a_{2n}, I_{2n})$ be its canonical interval representation. We denote the *untiming* of the function as a sequence which captures explicitly the points of discontinuities and the intervals between them. The untiming of the above f, denoted $\mathit{untiming}(f)$, is defined as $a_0 \cdots a_{2n}$. Given a word w in A^*, we define its *timing* to be a set of functions: $\mathit{timing}(w) = \emptyset$ if $|w|$ is even, otherwise $f \in \mathit{timing}(w)$ if $w = a_0 a_1 \cdots a_{2n}$ and $(a_0, I_0)(a_1, I_1) \cdots (a_{2n}, I_{2n})$ is an interval representation of f. We can extend the definitions of *timing* and *untiming* to languages of functions in the expected way.

We define an *input-determined operator* Δ over an alphabet Σ as a partial function from $(T\Sigma^* \times \mathbb{R}_{\geq 0})$ to $2^{\mathbb{R}_{\geq 0}}$, which is defined for all pairs (σ, t), where $t \in [0, \mathit{length}(\sigma)]$. Given a set of input-determined operators Op, we define the set of guards over Op, denoted by $\mathcal{G}(Op)$, inductively as $g ::= \top \mid \Delta^I \mid \neg g \mid g \vee g \mid g \wedge g$, where $\Delta \in Op$ and $I \in \mathcal{I}_{\mathbb{Q}}$. Guards of the form Δ^I are called atomic. Given a timed word σ, we define the satisfiability of a guard g at time $t \in [0, \mathit{length}(\sigma)]$, denoted $\sigma, t \models g$, as $\sigma, t \models \Delta^I$ iff $\Delta(\sigma, t) \cap I \neq \emptyset$, and in the usual way for boolean operators. For example Δ_r, which maps (σ, t) to $\{1\}$ if t is rational, and to $\{0\}$ otherwise, is an input-determined operator. Other examples include the eventual operator $\Diamond_a$, inspired by MTL, which maps (σ, t) to the set of distances to time points in σ after t at which an event a occurs, and the event-recording operator $\triangleleft_a$ which maps (σ, t) to the set containing the distance to the time point which corresponds to the last occurrence of the event a before time t.

We call an input-determined operator Δ over Σ *finitely varying* if for all $\sigma \in T\Sigma^*$ and $I \in \mathcal{I}_{\mathbb{Q}}$, the function $f_{\sigma,\Delta^I} : [0, \mathit{length}(\sigma)] \to \{0, 1\}$ defined as, $f_{\sigma,\Delta^I}(t)$ is 1 if $\sigma, t \models \Delta^I$, and 0 otherwise, is finitely varying. In the examples above, the operators $\Diamond_a$ and $\triangleleft_a$ are finitely varying, whereas Δ_r is not.

Let Σ be an alphabet and Op be a set of input determined operators over Σ. A *symbolic alphabet* based on (Σ, Op) is of the form (Γ_1, Γ_2) where Γ_1 is a finite subset of $(\Sigma \cup \{\epsilon\}) \times \mathcal{G}(Op)$ and Γ_2 is a finite subset of $\mathcal{G}(Op)$. We define the set of timed words over Σ associated with a function f in $\mathit{func}(\Gamma_1 \cup \Gamma_2)$, denoted $\mathit{tw}(f)$, as follows. If $\mathit{untiming}(f) \notin \Gamma_1 \cdot (\Gamma_2 \cdot \Gamma_1)^*$, then $\mathit{tw}(f) = \emptyset$. Otherwise, a timed word $\sigma = (a_1, t_1) \cdots (a_n, t_n)$ is in $\mathit{tw}(f)$, if

1. $\mathit{length}(\sigma) = \mathit{length}(f)$,
2. for each $i \in \{1, \cdots, n\}$, $f(t_i) = (a_i, g_i)$ for some g_i such that $\sigma, t_i \models g_i$, and
3. for all other $t \in [0, \mathit{length}(\sigma)]$, $f(t)$ should be of the form (ϵ, g_t) or g_t, with $\sigma, t \models g_t$.

Note that for any f, $tw(f)$ is either a singleton set or an empty set. We extend the definition of tw to a set of finitely varying functions over $\Gamma_1 \cup \Gamma_2$.

Example 8.1 *Let* $\Sigma = \{a\}$ *and* $Op = \{\Diamond_a\}$. *Then* (Γ_1, Γ_2) *is a symbolic alphabet, where* $\Gamma_1 = \{(a, \top), (\epsilon, \Diamond_a^{[1,1]})\}$ *and* $\Gamma_2 = \{\top\}$.

The finitely varying function f_1 *whose canonical interval representation is* $((a, \top), [0, 0])(\top, (0, 1))((\epsilon, \Diamond_a^{[1,1]}), [1, 1])(\top, (1, 2))((a, \top), [2, 2])$, *is depicted below. It can be seen that the timed word* $(a, 0)(a, 2)$ *belongs to* $tw(f_1)$.

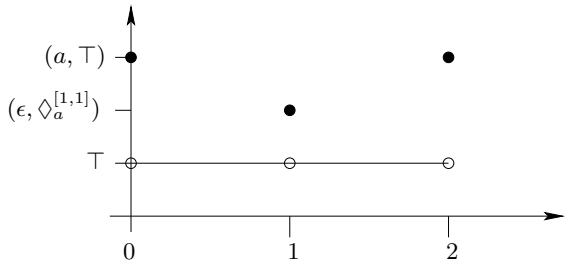


Figure 8.1: Finitely varying function f_1.

However the finitely varying function f_2 *whose canonical interval representation is* $((a, \top), [0, 0])(\top, (0, 1))((\epsilon, \neg\Diamond_a^{[1,1]}), [1, 1])(\top, (1, 2))((a, \top), [2, 2])$, *does not have any timed word associated with it.*

8.2 Definition

In this section we define *CIDA*'s and the languages accepted by them.

Let Σ be an alphabet and Op be a set of input determined operators based on Σ. A *Continuous Input Determined Automaton* (*CIDA*) $\mathcal{A}$ over (Σ, Op) is a structure (Q, s, δ, F, inv) on a symbolic alphabet (Γ_1, Γ_2) over (Σ, Op), where

- Q is a finite set of states,
- $s \in Q$ is the start state,
- $\delta \subseteq Q \times \Gamma_1 \times Q$ is the transition relation,
- $inv : Q \to \Gamma_2$ is the labelling function for the states, and
- $F \subseteq Q$ is the set of accepting states.

We now define the *symbolic language* accepted by the *CIDA* $\mathcal{A}$. Let $\gamma \in \Gamma_1 \cdot (\Gamma_2 \cdot \Gamma_1)^*$ and let $\gamma = \hat{a}_0 \hat{a}_1 \cdots \hat{a}_{2n}$. Let $N = \{0, \cdots, n+1\}$. A run of $\mathcal{A}$ over γ is a map $\rho : N \to Q$ such that

- $\rho(0) = s$,
- $(\rho(i), \hat{a}_{2i}, \rho(i+1)) \in \delta$ for $i = 0, \cdots, n$, and
- $inv(\rho(i)) = \hat{a}_{2i-1}$ for all $1 \leq i \leq n$.

We say ρ is accepting if $\rho(n+1) \in F$. The symbolic language defined by $\mathcal{A}$, denoted $L_{sym}(\mathcal{A})$, is the set of words in $\Gamma_1 \cdot (\Gamma_2 \cdot \Gamma_1)^*$ over which $\mathcal{A}$ has an accepting run. We define the language of functions accepted by the *CIDA* $\mathcal{A}$, denoted $F(\mathcal{A})$, as $timing(L_{sym}(\mathcal{A}))$. The timed language of the *CIDA* $\mathcal{A}$, denoted $L(\mathcal{A})$, is defined as $tw(F(\mathcal{A}))$.

We give below a concrete example of a *CIDA*, which we call *Continuous Eventual Timed Automata* (*CETA*). A *CETA* over an alphabet Σ is a *CIDA* over (Σ, Op), where $Op = \{\Diamond_a \mid a \in \Sigma\}$ is the set of eventual operators based on Σ. The diagram below gives the *CETA* $\mathcal{A}$ over $\{a, b\}$ which recognizes the language L_{ni}, that is, $L(\mathcal{A}) = L_{ni}$.

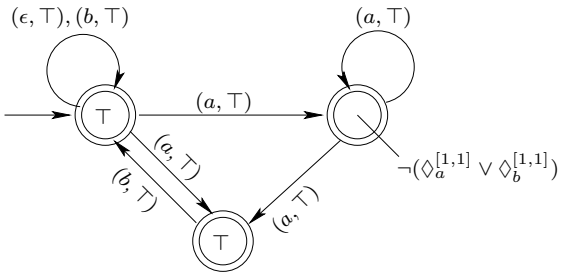


Figure 8.2: CETA for L_{ni}.

8.3 Closure properties and determinization

In this section we show that *CIDA*'s over finitely varying input-determined operators are closed under boolean operations and are determinizable. Towards this end it is useful to define the notion of a "proper" *CIDA*.

Let us fix an alphabet Σ and a set of finitely varying operators Op over Σ. We first define a *proper symbolic alphabet*. A proper symbolic alphabet based on (Σ, Op) is of the form (Γ_1, Γ_2), where $\Gamma_1 = (\Sigma \times \{\epsilon\}) \times 2^G$ and $\Gamma_2 = 2^G$ for some finite set G of atomic guards based on Op. An element of a proper

symbolic alphabet is thus of the form (c, h) or h, where h is a finite subset of atomic guards, and is meant to represent the exact set of guards in G satisfied at a point. For a function $f \in \mathit{func}(\Gamma_1 \cup \Gamma_2)$, we can associate a set of timed words, also denoted $\mathit{tw}(f)$, in a similar way to symbolic alphabets, by viewing each proper guard h as the conjunction $\bigwedge_{h' \in h} h' \wedge \bigwedge_{h' \in G-h} \neg h'$. We now define a *proper CIDA*. A proper *CIDA* over (Σ, Op) is a structure $(Q, s, \delta, F, \mathit{inv})$ on a proper symbolic alphabet (Γ_1, Γ_2) over (Σ, Op), where Q, s, δ, F and inv are defined as for *CIDA*'s. The symbolic languages and the timed languages accepted by them can now be defined similar to *CIDA*'s.

We call a word $w = a_0 \cdots a_n$ over a proper symbolic alphabet *fully canonical* if n is even and for no even i is $a_{i-1} = a_i = a_{i+1}$. We call a proper *CIDA* *fully canonical* if its symbolic language consists of fully canonical symbolic words.

We first prove the following lemmas which connect *CIDA*'s, proper *CIDA*'s and fully canonical proper *CIDA*'s, and then make use of them to show closure and determinizability.

Lemma 8.1 *CIDA's over* (Σ, Op) *and proper CIDA's over* (Σ, Op) *define the same class of timed languages.*

Proof Given a proper *CIDA* over a proper symbolic alphabet (Γ_1, Γ_2) based on a set of atomic guards G, we can give a *CIDA* which accepts the same timed language by replacing every subset h of G in the symbols by a guard $\bigwedge_{h' \in h} h' \wedge \bigwedge_{h' \in G-h} \neg h'$.

In the other direction let $\mathcal{A}$ be a *CIDA* over a symbolic alphabet (Γ_1, Γ_2). Every guard g in $\mathcal{G}(Op)$ can be written in a disjunctive normal form (d.n.f.) $(c_1 \vee \cdots \vee c_k)$ where each c_i is a conjunction of atomic guards or their negations. Now we say that a set of atomic guards h is consistent with a guard g if it is consistent with at least one of the conjuncts in its d.n.f., and it is consistent with a conjunct if it contains all atomic guards which occur in the conjunct in a non-negated form and does not contain any atomic guards which occur in the conjunct in the negated form.

We split each state into a set of states each of which corresponds to a set of guards which are consistent with the invariant on the original state. Each of these states is then labelled by the set of guards to which it corresponds. Further there is a transition between two states if there were a transition between the original two states, and the set of guards on the transition is consistent with the guard on the original transition. We also add epsilon-transitions between the states which correspond to the same original state such that the set of atomic guards on them is consistent with the invariant on the original state. We give the formal construction below.

Let $\mathcal{A} = (Q, s, \delta, F, inv)$. The corresponding proper *CIDA* $\mathcal{A}'$ is given by $\mathcal{A}' = (Q', s', \delta', F', inv')$, where $Q' = \{s'\} \cup \{(q,h) \mid q \in Q, h \subseteq G, h \text{ is consistent with } inv(q)\}$, $\delta' = \{(s', (c, h_1), (q, h_2)) \mid (s, (c, g), q) \in \delta, h_1 \text{ is consistent with } g\} \cup \{((p, h_1), (c, h_2), (q, h_3)) \mid (p, (c, g), q) \in \delta, h_2 \text{ is consistent with } g\} \cup \{((p, h_1), (\epsilon, h_2), (p, h_3)) \mid h_2 \text{ is consistent with } inv(p)\}$, $F' = \{(q,h) \in Q' \mid q \in F\}$, and $inv'(s') = \top$ and $inv'((q,h)) = h$. □

Lemma 8.2 *Proper CIDA's over* (Σ, Op) *and fully canonical proper CIDA's over* (Σ, Op) *define the same class of timed languages.*

Proof Given a proper *CIDA* $\mathcal{A}$ based on a proper symbolic alphabet (Γ_1, Γ_2) we give a fully canonical proper *CIDA* $\mathcal{A}'$ based on the same proper symbolic alphabet such that $L(\mathcal{A}) = L(\mathcal{A}')$.

Let $\mathcal{A} = (Q, s, \delta, F, inv)$. We first compute the smallest set δ' such that $\delta \subseteq \delta'$, and if $(q, (\epsilon, h), r) \in \delta'$ such that $inv(q) = inv(r) = h$, then for all $(p, \hat{a}, q) \in \delta'$, $(p, \hat{a}, r) \in \delta'$. This set is computable since one can add transitions one at a time until no more can be added, and this process terminates since the total number of transitions that can be added is finite. We now argue that adding such a transition does not change the language of the proper *CIDA*. Suppose γ is a new word in its symbolic language, then it has a run over γ which takes the newly added transition zero or more times. Then there is a run in the original automaton over γ' got by replacing every $h_1(c, h_2)h_3$ corresponding to the newly added transition by $h_1(c, h_2)h_3(\epsilon, h_3)h_3$. But it is easy to see that the $tw(timing(\gamma)) = tw(timing(\gamma'))$. Hence the newly added word does not add any more timed words to the language of the proper *CIDA*.

We now remove the transitions of the form $(p, (\epsilon, h), q)$ where $inv(p) = inv(q) = h$. We can argue similarly that the removal of a transition leaves the language of the proper *CIDA* unchanged. Since no word in the symbolic language of the final automaton has $h(\epsilon, h)h$ as a substring, it can be seen that the resulting *CIDA* accepts only fully canonical words over the proper symbolic alphabet. □

We define a *CIDA-language* L over (Σ, Op) to be a regular subset of $\Gamma_1 \cdot (\Gamma_2 \cdot \Gamma_1)^*$ for some proper symbolic alphabet (Γ_1, Γ_2) over (Σ, Op).

Proposition 8.1 *Let* (Γ_1, Γ_2) *be a proper symbolic alphabet over* (Σ, Op)*, and let* L *be a language over* $(\Gamma_1 \cup \Gamma_2)$*. Then* L *is the symbolic language of a proper CIDA iff* L *is a CIDA-language.* □

Lemma 8.3 *Let* (Γ_1, Γ_2) *be a proper symbolic alphabet over* (Σ, Op)*. Given a timed word* σ *over* Σ*, there exists a unique fully canonical word* γ *over* (Γ_1, Γ_2) *such that* $\sigma \in tw(timing(\gamma))$*.* □

CIDA's are determinizable in the sense that we can give a timed language equivalent fully canonical proper *CIDA* $\mathcal{A}'$ which is deterministic in the sense that for every timed word σ accepted by it, there is a unique symbolic word γ in $L_{sym}(\mathcal{A}')$ such that $\sigma \in tw(timing(\gamma)))$.

Theorem 8.1 *The class of CIDA's over* (Σ, Op) *is closed under union, intersection and complementation.*

Proof From Lemma 8.1, it follows that it is enough to show that the class of proper *CIDA*'s over (Σ, Op) is closed under boolean operations.

The union of proper *CIDA*'s is equivalent to the union of their symbolic languages. Since their symbolic languages are *CIDA*-languages and *CIDA*-languages are closed under union, proper *CIDA*'s are themselves closed under union.

Now turning to complementation, using Lemma 8.2 we can give an equivalent fully canonical proper *CIDA* $\mathcal{A}'$ for a given proper *CIDA* $\mathcal{A}$. From Lemma 8.3, we have that the set of timed words associated with two distinct fully canonical words over a proper symbolic alphabet is disjoint. Hence we can complement the timed language of $\mathcal{A}'$ by complementing its symbolic language with respect to the set of all fully canonical proper words (which is a *CIDA*-language). Hence we have a *CIDA* accepting the complement of $L(\mathcal{A})$. □

Chapter 9

A logical characterization of *CIDA*'s

In this chapter we define an MSO logic based on a set of input-determined operators. We show that these logics characterize *CIDA*'s. In the proof we make use of the fact that the untiming of a language definable by an MSO formula interpreted over finitely varying functions is regular. Hence we will provide a detailed proof of the result in section 9.2, which was also independently proved by Rabinovich [22].

9.1 Continuous timed monadic second order logic

In this section we define the syntax and semantics of continuous timed monadic second order logic based on a set of input-determined operators.

Let Σ be an alphabet and Op be a set of input-determined operators over Σ. We define the syntax of *continuous timed monadic second order logic* over (Σ, Op), denoted $\mathrm{TMSO}^c(\Sigma, Op)$, as follows. The formulas of $\mathrm{TMSO}^c(\Sigma, Op)$ are inductively defined as:

$$\varphi ::= Q_a(x) \mid \Delta^I(x) \mid x \in X \mid x < y \mid \neg\varphi \mid (\varphi \vee \varphi) \mid \exists x\varphi \mid \exists X\varphi,$$

where $a \in \Sigma$, $\Delta \in Op$, $I \in \mathcal{I}_{\mathbb{Q}}$, and x and X are first and second order variables. We use the convention that the small letters are first order variables and capital letters are second order variables.

We interpret the logic over timed words in $T\Sigma^*$. An interpretation $\mathbb{I}$ with respect to a timed word σ maps a first order variable x to $t \in [0, \mathit{length}(\sigma)]$ and a second order variable X to $B \subseteq_{\mathit{fin}} [0, \mathit{length}(\sigma)]$ ($\subseteq_{\mathit{fin}}$ denotes finite subset). The relation $<$ is interpreted as the usual ordering of $\mathbb{R}_{\geq 0}$.

For an interpretation $\mathbb{I}$, we use the notation $\mathbb{I}[t/x]$ to denote the interpretation which sends x to t and agrees with $\mathbb{I}$ on all other variables. Similarly, $\mathbb{I}[B/X]$ denotes the modification of $\mathbb{I}$ which maps the set variable X to B and the rest to the same as that by $\mathbb{I}$. We also use the notation $[t/x]$ to denote an interpretation which sends x to t when the rest of the interpretation is irrelevant.

Given a formula $\varphi \in \mathrm{TMSO}^c(\Sigma, Op)$, a timed word $\sigma = (a_1, t_1) \cdots (a_n, t_n)$ in $T\Sigma^*$, and an interpretation $\mathbb{I}$ to the variables in φ with respect to σ, we define the satisfaction relation $\sigma, \mathbb{I} \models \varphi$, inductively as follows:

$$
\begin{array}{lll}
\sigma, \mathbb{I} \models \mathrm{Q}_a(x) & \text{iff} & \exists i : t_i = \mathbb{I}(x) \text{ and } a_i = a. \\
\sigma, \mathbb{I} \models \Delta^I(x) & \text{iff} & \Delta(\sigma, \mathbb{I}(x)) \cap I \neq \emptyset. \\
\sigma, \mathbb{I} \models x \in X & \text{iff} & \mathbb{I}(x) \in \mathbb{I}(X). \\
\sigma, \mathbb{I} \models x < y & \text{iff} & \mathbb{I}(x) < \mathbb{I}(y). \\
\sigma, \mathbb{I} \models \neg\varphi & \text{iff} & \sigma, \mathbb{I} \not\models \varphi. \\
\sigma, \mathbb{I} \models \varphi_1 \vee \varphi_2 & \text{iff} & \sigma, \mathbb{I} \models \varphi_1 \text{ or } \sigma, \mathbb{I} \models \varphi_2. \\
\sigma, \mathbb{I} \models \exists x \varphi & \text{iff} & \exists t \in [0, \mathit{length}(\sigma)],\ \sigma, \mathbb{I}[t/x] \models \varphi. \\
\sigma, \mathbb{I} \models \exists X \varphi & \text{iff} & \exists B \subseteq_{\mathit{fin}} [0, \mathit{length}(\sigma)] : \sigma, \mathbb{I}[B/X] \models \varphi.
\end{array}
$$

For a sentence, a formula without free variables, the interpretation does not play any role. Hence, for a sentence φ in $\mathrm{TMSO}^c(\Sigma, Op)$, we set the timed language defined by φ to be $L(\varphi) = \{\sigma \in T\Sigma^* \mid \sigma \models \varphi\}$.

Example 9.1 *$\exists x(Q_a(x) \wedge \Diamond_a^{[1,1]}(x))$ is a sentence which defines the language in which every timed word contains two a's which are distance 1 apart.*

Example 9.2 *The following sentence defines L_{ni}.*
$\neg(\exists x \exists y \exists z(Q_a(x) \wedge Q_a(y) \wedge (x < z \wedge z < y) \wedge \Diamond_a^{[1,1]}(z)))$.

In the following theorem we claim that TMSO^c characterizes *CIDA*'s.

Theorem 9.1 *Let Σ be a finite alphabet and Op be a set of finitely varying input-determined operators based on Σ. Let L be a timed language over Σ. Then L is accepted by a CIDA over (Σ, Op) iff it is definable by a $\mathrm{TMSO}^c(\Sigma, Op)$ sentence.* □

The rest of the chapter is devoted to a proof of the above theorem. We first prove a result for MSO^c which we will use in the proof.

9.2 Continuous monadic second order logic

In this section we define a continuous version of Büchi's monadic second-order logic which is interpreted over finitely varying functions. We show that the untiming of a function language defined by a sentence in the logic is regular. We note that a similar result is due to Rabinovich [22], though the techniques are different. Our approach is automata-theoretic, whereas in [22] regularity is shown by a reduction to classical MSO.

We recall that for an alphabet A, Büchi's monadic second order logic (denoted here by $\mathrm{MSO}^c(A)$) is given as follows:

$$\varphi ::= Q_a(x) \mid x \in X \mid x < y \mid \neg\varphi \mid (\varphi \vee \varphi) \mid \exists x\varphi \mid \exists X\varphi,$$

where $a \in A$, and x and X belong to the countable sets of first-order and second-order variables, respectively.

We interpret a formula of the logic over a finitely varying function f in $\mathit{func}(A)$, along with an interpretation $\mathbb{I}$ with respect to f, which assigns to a first order variable x, a value in $[0, \mathit{length}(f)]$, and to a set variable X, a finite subset of $[0, \mathit{length}(f)]$.

We now define the semantics of $\mathrm{MSO}^c(A)$. Given a formula $\varphi \in \mathrm{MSO}^c(A)$, $f \in \mathit{func}(A)$ and an interpretation $\mathbb{I}$ with respect to f to the variables in φ, the satisfaction relation $f, \mathbb{I} \models \varphi$, is defined inductively as:

$$\begin{array}{lll}
f, \mathbb{I} \models \mathrm{Q}_a(x) & \text{iff} & f(\mathbb{I}(x)) = a. \\
f, \mathbb{I} \models x \in X & \text{iff} & \mathbb{I}(x) \in \mathbb{I}(X). \\
f, \mathbb{I} \models x < y & \text{iff} & \mathbb{I}(x) < \mathbb{I}(y). \\
f, \mathbb{I} \models \neg\varphi & \text{iff} & f, \mathbb{I} \not\models \varphi. \\
f, \mathbb{I} \models \varphi_1 \vee \varphi_2 & \text{iff} & f, \mathbb{I} \models \varphi_1 \text{ or } f, \mathbb{I} \models \varphi_2. \\
f, \mathbb{I} \models \exists x\varphi & \text{iff} & \exists t \in [0, \mathit{length}(f)] : \ f, \mathbb{I}[t/x] \models \varphi. \\
f, \mathbb{I} \models \exists X\varphi & \text{iff} & \exists B \subseteq_{\mathit{fin}} [0, \mathit{length}(f)] : \ f, \mathbb{I}[B/X] \models \varphi.
\end{array}$$

For a sentence φ in $\mathrm{MSO}^c(A)$, we set the language defined by φ to be $F(\varphi) = \{f \in \mathit{func}(A) \mid f \models \varphi\}$.

The following theorem relates MSO^c and *FSA*'s.

Theorem 9.2 *Given a sentence φ in $\mathrm{MSO}^c(A)$, we can give a finite state automaton $\mathcal{A}_\varphi$ such that $F(\varphi) = \mathit{timing}(L_{\mathit{sym}}(\mathcal{A}_\varphi))$.* □

In the rest of the section we present a proof of the above theorem.

We will represent models for formulas with free variables in them, as functions with the interpretations built into them. We assume an ordering on the countable sets of first-order and second-order variables, given by

$x_1, x_2, \cdots$ and $X_1, X_2, \cdots$. For a formula φ with free variables among $X = \{x_{i_1}, \cdots, x_{i_m}\}$ and $Y = \{X_{j_1}, \cdots, X_{j_n}\}$ (in order), we represent a function f and an interpretation $\mathbb{I}$ as a function $f_{\mathbb{I}}^{X,Y} : [0, \mathit{length}(f)] \to A \times \{0,1\}^{m+n}$ given by $f_{\mathbb{I}}^{X,Y}(t) = (f(t), b_1, \cdots, b_m, c_1, \cdots, c_n)$, where $b_k = 1$ iff $\mathbb{I}(x_{i_k}) = t$ and $c_k = 1$ iff $t \in \mathbb{I}(X_{j_k})$. Thus for a formula φ with free variables in (X, Y) we have a notion of (X, Y)-models of φ.

We will prove the following lemma which will help us in the proof of the theorem.

Lemma 9.1 *Let φ be a $\mathrm{MSO}^c(A)$ formula with free variables in $X = \{x_{i_1}, \cdots, x_{i_m}\}$ and $Y = \{X_{j_1}, \cdots, X_{j_n}\}$. Let X' and Y' be finite sets of first and second order variables which are supersets of X and Y respectively. Suppose that we have an automaton $\mathcal{A}$ accepting the untiming of the (X, Y)-models of φ. Then we can construct an automaton A' accepting precisely the untiming of (X', Y')-models of φ.*

Proof Let $\mathcal{A} = (Q, s, \delta, F)$. We first construct an automaton $\mathcal{A}_1$, which consists of two copies of the states of the above automaton with transitions only between states of different copies, and accepts the set of odd length words accepted by the original automaton (recall that a word which is the untiming of some f is of odd length). $\mathcal{A}_1 = (Q_1, s_1, \delta_1, F_1)$, where $Q_1 = Q \times \{0,1\}$, $s_1 = (s, 0)$, $\delta_1 = \{((p,0), a, (q,1)) \mid (p, a, q) \in \delta\} \cup \{((p,1), a, (q,0)) \mid (p, a, q) \in \delta\}$ and $F_1 = F \times \{1\}$.

Next we give an automaton $\mathcal{A}_2$ such that if $w = a_0 b_1 a_1 \cdots b_l a_l$ is accepted by $\mathcal{A}_1$ then a w' obtained by replacing every b_i by some $b_i^{k_i}$, where k_i is odd, is accepted by $\mathcal{A}_2$. Let $\delta_o = \{((p,1), a, (q,0)) \in \delta_1\}$. $\mathcal{A}_2 = (Q_2, s_2, \delta_2, F_2)$, where $Q_2 = Q_1 \cup (\delta_o \times \{0,1\})$, $s_2 = s_1$, $\delta_2 = \delta_1 \cup \{(p, a, (e,0)) \mid e = (p, a, q), e \in \delta_o\} \cup \{((e,1), a, q) \mid e = (p, a, q), e \in \delta_o\} \cup \{((e,0), a, (e,1)) \mid e = (p, a, q), e \in \delta_o\} \cup \{((e,1), a, (e,0)) \mid e = (p, a, q), e \in \delta_o\}$, and $F_2 = F_1$.

We give here the automaton $\mathcal{A}'$ for the case when $X' = X$ and $Y' = \{X_{j_1}, \cdots, X_{j_n}, X_{j_{n+1}}\}$. However, we note that it can be easily extended to any sets X' and Y'. We construct the automaton $\mathcal{A}'$ by replacing every transition labelled by a symbol from $A \times \{0,1\}^{m+n}$ in $\mathcal{A}_2$ by transitions corresponding to every possible extension of the symbol. $\mathcal{A}' = (Q', s', \delta', F')$ where $\delta' = \{(p, (i_1, \cdots, i_n, j_1, \cdots, j_m, k), q) \mid (p, (i_1, \cdots, i_m, j_1, \cdots, j_n), q) \in \delta_2, k \in \{0,1\}\}$, $Q' = Q$, $s' = s$, and $F' = F$.

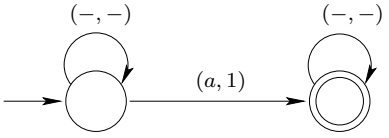
Finally we intersect the automaton with the automata $\mathcal{A}_{\mathit{canon}}$ and $\mathcal{A}_{\mathit{valid}}$, where $\mathcal{A}_{\mathit{canon}}$ consists of words which are canonical, and $\mathcal{A}_{\mathit{valid}}$ consists of words which have for every first-order variable exactly one symbol which has a 1 in the component corresponding to it, and which satisfy the condition that a symbol has a 1 in some component only if it occurs at an even position.

□

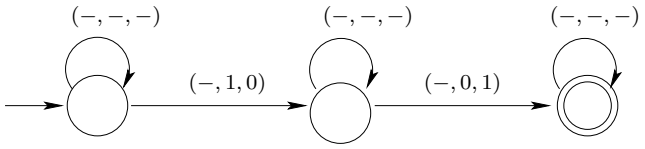
We now give the automaton accepting the untiming of a function language of a sentence. Given a formula $\varphi \in \mathrm{MSO}^c(A)$, we inductively construct the automaton $\mathcal{A}_\varphi^{X,Y}$, where X and Y contain exactly the free variables in φ, such that its symbolic language consists of the untimings of the (X,Y)-models of φ. We further argue that $f, \mathbb{I} \models \varphi$ iff $\mathit{untiming}(f_{\mathbb{I}}^{X,Y}) \in \mathcal{A}_\varphi^{X,Y}$.

Let $\mathcal{A}_{canon}^{X,Y}$ be the automaton which accepts canonical words over $A \times \{0,1\}^{|X|+|Y|}$. Let $\mathcal{A}_{valid}^{X,Y}$ be the automaton which accepts words over $A \times \{0,1\}^{|X|+|Y|}$ which have for every first-order variable in X exactly one symbol which has a 1 in the component corresponding to it, and which satisfy the condition that a symbol has a 1 in some component only if it occurs at an even position.

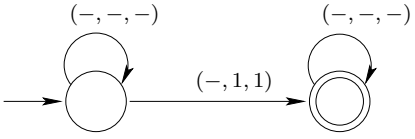
1. $\varphi = Q_a(x)$: The automaton $\mathcal{A}_\varphi^{X,Y}$ is the intersection of $\mathcal{A}_{canon}^{X,Y}$ and $\mathcal{A}_{valid}^{X,Y}$ with:



2. $\varphi = x_1 < x_2$: The automaton $\mathcal{A}_\varphi^{X,Y}$ is the intersection of $\mathcal{A}_{canon}^{X,Y}$ and $\mathcal{A}_{valid}^{X,Y}$ with:



3. $\varphi = x \in X$: The automaton $\mathcal{A}_\varphi^{X,Y}$ is the intersection of $\mathcal{A}_{canon}^{X,Y}$ and $\mathcal{A}_{valid}^{X,Y}$ with:



4. $\varphi = \neg\eta$: Let $\mathcal{A}_\eta^{X,Y}$ be the automaton for η. Then $\mathcal{A}_\varphi^{X,Y}$ is the intersection of $\mathcal{A}_{canon}^{X,Y}$ and $\mathcal{A}_{valid}^{X,Y}$ with complement of $\mathcal{A}_\eta^{X,Y}$.

5. $\varphi = \eta \vee \psi$: Let $\mathcal{A}_\eta^{X',Y'}$ be the automaton for η and let $\mathcal{A}_\psi^{X'',Y''}$ be the automaton for ψ. Let (X, Y) be the free variables in $\eta \vee \psi$. We can use the construction in Lemma 9.1 to get the automata $\mathcal{A}_\eta^{X,Y}$ and $\mathcal{A}_\psi^{X,Y}$, which accept the untimings of the (X, Y)-models of η and ψ respectively. Then $\mathcal{A}_\varphi^{X,Y}$ is the intersection of $\mathcal{A}_{canon}^{X,Y}$ and $\mathcal{A}_{valid}^{X,Y}$ with the union of $\mathcal{A}_\eta^{X,Y}$ and $\mathcal{A}_\psi^{X,Y}$.

6. $\varphi = \exists x\eta$: Let $\mathcal{A}_\eta^{X',Y}$ be the automaton for η, where $X' = (x, x_1, \cdots, x_n)$. Let $X = (x_1, \cdots, x_n)$.

   Let w be a word of odd length over A and w' be in A^*. Then $w' = \mathit{canonical}(w)$ if $w' = a_0b_1a_1 \cdots b_na_n$ such that for no i is $b_i = a_i = b_{i+1}$, and w can be obtained from w' by replacing each b_i by $b_i^{k_i}$ for some odd k_i. Given a word w over $A \times \{0,1\}^{n+m+1}$, we define $\mathit{proj}(w)$ to be a word over $A \times \{0,1\}^{n+m}$, which projects away its x-component.

   Proposition 9.1 *Let $f \in \mathit{func}(A)$ and $\mathbb{I}$ be an interpretation with respect to f. Then $\mathit{untiming}(f_{\mathbb{I}}^{X,Y}) = \mathit{canonical}(\mathit{proj}(\mathit{untiming}(f_{\mathbb{I}}^{X',Y})))$.* □

   We first project away the x-components of the labels on the transitions in the automaton. We then use the construction in Lemma 9.1 used for the construction of $\mathcal{A}_1$ to obtain the automaton $\mathcal{A}'$. We now add transitions $((p,1), a, (q,0))$ if there exist transitions $((p,1), a, (r,0))$, $((r,0), a, (t,1))$ and $((t,1), a, (q,0))$, repeatedly as done in Lemma 8.2 to get $\mathcal{A}''$. We intersect it with $\mathcal{A}_{canon}^{X,Y}$ and $\mathcal{A}_{valid}^{X,Y}$ to get $\mathcal{A}_\varphi^{X,Y}$.

7. $\varphi = \exists Z\eta$: The construction is similar to that of the previous case.

9.3 Proof of MSO characterization of *CIDA*'s

In this section we complete the proof of Theorem 9.1.

In the forward direction, we show that the class of languages defined by *CIDA*'s over (Σ, Op) is a subset of the class of languages defined by $\mathrm{TMSO}^c(\Sigma, Op)$ sentences.

Let $\mathcal{A} = (Q, s, \delta, F, inv)$ be a *CIDA* over (Σ, Op). We give a $\mathrm{TMSO}^c(\Sigma, Op)$ sentence $\varphi_\mathcal{A}$ such that $L(\mathcal{A}) = L(\varphi_\mathcal{A})$. The formula essentially checks for the existence of a valid run of $\mathcal{A}$ over a timed word. Let $\delta = \{e_1, \cdots, e_m\}$ be the set of transitions. We define the set $\mathit{consec} \subseteq \delta \times \delta$, where $(e, e') \in \mathit{consec}$

if and only if there exists q such that $e = (p, \hat{a}, q)$ and $e' = (q, \hat{b}, r)$. We define below some abbreviations which will help us in the construction of the formula $\varphi_{\mathcal{A}}$.

- $\mathit{first}(x) = \neg\exists y(y < x)$,
- $\mathit{last}(x) = \neg\exists y(x < y)$,
- $\mathit{next}(x, y, X) = x \in X \wedge y \in X \wedge \neg\exists w(x < w \wedge w < y \wedge w \in X)$, and
- $\mathit{between}(x, y, z) = x < y \wedge y < z$.

We use $\mathit{action}(x)$ for $\bigvee_{a\in\Sigma} Q_a(x)$. Given a guard g over Op, we will use $g(x)$ to denote the TMSO^c formula obtained by replacing every Δ^I in g by $\Delta^I(x)$. The second order variables $X_{e_1}, \cdots, X_{e_m}$ are used to capture the points in the timed words which correspond to the transitions $e_1, \cdots, e_m$, respectively, and X to capture their union. $\varphi_{\mathcal{A}}$ is given by: $\exists X \exists X_{e_1} \cdots \exists X_{e_m}(\varphi_1 \wedge \varphi_2 \wedge \varphi_3 \wedge \varphi_4 \wedge \varphi_5 \wedge \varphi_6 \wedge \varphi_7 \wedge \varphi_8)$, where:

1. φ_1 says that X is the union of $X_{e_1}, \cdots, X_{e_m}$:
$$\forall x((\bigvee_{e\in\delta} x \in X_e) \Leftrightarrow x \in X).$$
2. φ_2 guarantees that the sets $X_{e_1}, \cdots, X_{e_m}$ are disjoint:
$$\forall x \bigwedge_{i,j\in\{1,\cdots,m\}, i\neq j} (x \in X_{e_i} \Rightarrow x \notin X_{e_j}).$$
3. φ_3 ensures that the time instant 0 belongs to the set corresponding to some transition from the start state:
$$\forall x(\mathit{first}(x) \Rightarrow \bigvee_{(s,\hat{a},q)\in\delta} x \in X_{(s,\hat{a},q)}).$$
4. φ_4 ensures that the time of the last action of the timed word is in a set corresponding to some transition into the final state:
$$\forall x(\mathit{last}(x) \Rightarrow \bigvee_{(p,\hat{a},q)\in\delta, q\in F} x \in X_{(p,\hat{a},q)}).$$
5. φ_5 says that any two consecutive points in the set X belong to consecutive transitions:
$$\forall x \forall y(\mathit{next}(x, y, X) \Rightarrow \bigvee_{(e,e')\in \mathit{consec}} (x \in X_e \wedge y \in X_{e'})).$$

6. φ_6 says that if a point in a timed word corresponds to a transition labelled (a, g) where $a \in \Sigma$, then the timed word at that point contains an a and satisfies the guard g:

$$\forall x \bigwedge_{(p,(a,g),q)\in\delta} (x \in X_{(p,(a,g),q)} \Rightarrow (Q_a(x) \wedge g(x))).$$

7. φ_7 says that if a point in a timed word corresponds to a transition labelled (ϵ, g), then the timed word at that point does not contain an action and satisfies the guard g:

$$\forall x \bigwedge_{(p,(\epsilon,g),q)\in\delta} (x \in X_{(p,(\epsilon,g),q)} \Rightarrow (\neg action(x) \wedge g(x))).$$

8. φ_8 says that all points which lie between the points corresponding to two consecutive transition in the run of a timed word should not contain any action and should satisfy the guard on the corresponding state:

$$\forall x \forall y \forall z((next(y, z) \wedge between(y, x, z)) \Rightarrow$$

$$(\bigwedge_{(p,\hat{a},q)\in\delta} (y \in X_{(p,\hat{a},q)} \Rightarrow (\neg action(x) \wedge [inv(q)](x))))).$$

In the other direction we reduce a TMSO^c formula to an MSO^c formula, and then factor through Theorem 9.2 to get an *FSA* for the untiming of its language, from which we construct the required *CIDA*.

Let $\sigma \in T\Sigma^*$ and $\sigma = (a_1, t_1) \cdots (a_n, t_n)$. Let G be a finite set of atomic guards over Op. We define the function $f^G_\sigma : [0, length(\sigma)] \to 2^G$ as: for all t, $f^G_\sigma(t) = \{h' \mid h' \in G, \sigma, t \models h'\}$. Let $(h_0, I_0)(h_1, I_1) \cdots (h_{2m}, I_{2m})$ be the canonical interval representation of f^G_σ (recall the operators are finitely varying). Let (Γ_1, Γ_2) be the proper symbolic alphabet based on G and let $\Gamma = \Gamma_1 \cup \Gamma_2$. We define the function $f^\Gamma_\sigma : [0, length(\sigma)] \to \Gamma$ as follows. Let $t \in [0, length(\sigma)]$ and let $t \in I_j$. If $t = t_i$ for some $i \in \{i, \cdots, n\}$, then $f^\Gamma_\sigma(t) = (a_i, h_j)$, otherwise $f^\Gamma_\sigma(t) = (\epsilon, h_j)$ if j is even, and $f^\Gamma_\sigma(t) = h_j$ if j is odd. Note that $\sigma \in tw(f^\Gamma_\sigma)$.

Let $\varphi \in$ TMSO$^c(\Sigma, Op)$ and let $G = \{\Delta^I \mid \Delta^I(x)$ is a subformula of $\varphi\}$. Let (Γ_1, Γ_2) be the proper alphabet over (Σ, Op) based on G, and let $\Gamma = \Gamma_1 \cup \Gamma_2$. We now give the function *tmso-mso*, which maps a TMSO$^c(\Sigma, Op)$ formula φ to an MSO$^c(\Gamma)$ formula. We define *tmso-mso*, inductively as

follows:

$$
\begin{array}{lcl}
\mathit{tmso\text{-}mso}(Q_a(x)) & = & \bigvee_{(a,h)\in\Gamma} Q_{(a,h)}(x). \\
\mathit{tmso\text{-}mso}(\Delta^I(x)) & = & \bigvee_{(c,h)\in\Gamma,\Delta^I\in h} Q_{(c,h)}(x) \vee \bigvee_{h\in\Gamma,\Delta^I\in h} Q_h(x). \\
\mathit{tmso\text{-}mso}(x \in X) & = & x \in X. \\
\mathit{tmso\text{-}mso}(x < y) & = & x < y. \\
\mathit{tmso\text{-}mso}(\neg\psi) & = & \neg \mathit{tmso\text{-}mso}(\psi). \\
\mathit{tmso\text{-}mso}(\eta \vee \psi) & = & \mathit{tmso\text{-}mso}(\eta) \vee \mathit{tmso\text{-}mso}(\psi). \\
\mathit{tmso\text{-}mso}(\exists x\, \psi) & = & \exists x \mathit{tmso\text{-}mso}(\psi). \\
\mathit{tmso\text{-}mso}(\exists X\, \psi) & = & \exists X \mathit{tmso\text{-}mso}(\psi).
\end{array}
$$

Lemma 9.2 *Let $\sigma \in T\Sigma^*$ and $\mathbb{I}$ be an interpretation to the variables in φ with respect to σ. Let $f \in \mathit{func}(\Gamma)$ such that $\sigma \in \mathit{tw}(f)$. Then $\sigma, \mathbb{I} \models \varphi$ iff $f, \mathbb{I} \models \mathit{tmso\text{-}mso}(\varphi)$.*

Proof If $\sigma, \mathbb{I} \models Q_a(x)$, then there exists i such that $t_i = \mathbb{I}(x)$ and $a_i = a$. Since σ is in $\mathit{tw}(f)$, we have that $f(t_i) = (a_i, h)$ for some set of atomic guards h. But then $f(\mathbb{I}(x)) = (a, h)$ and hence $f, \mathbb{I} \models Q_{(a,h)}(x)$.

Now suppose $\sigma, \mathbb{I} \models \Delta^I(x)$. Then $\Delta(\sigma, \mathbb{I}(x)) \cap I \neq \emptyset$. Therefore $\sigma, \mathbb{I}(x) \models \Delta^I$, and it follows that $f(\mathbb{I}(x)) = (a, h)$ or (ϵ, h) or h where $\Delta^I \in h$. □

Lemma 9.3 *Let φ be a sentence in* $\mathrm{TMSO}^c(\Sigma, Op)$. *Then we have $L(\varphi) = \mathit{tw}(F(\mathit{tmso\text{-}mso}(\varphi)))$.*

Proof Let $\mathit{tmso\text{-}mso}(\varphi) = \tilde{\varphi}$. Suppose $\sigma \in \mathit{tw}(F(\tilde{\varphi}))$. Then there exists f such that $\sigma \in \mathit{tw}(f)$ and $f \in F(\tilde{\varphi})$. Hence $f \models \tilde{\varphi}$, which implies that $\sigma \models \varphi$ from Lemma 9.2. Therefore $\sigma \in L(\varphi)$.

Suppose $\sigma \in L(\varphi)$. Then $\sigma \models \varphi$. Since $\sigma \in \mathit{tw}(f^\Gamma_\sigma)$, $f^\Gamma_\sigma \models \tilde{\varphi}$. Hence $f^\Gamma_\sigma \in F(\tilde{\varphi})$. Therefore, $\sigma \in \mathit{tw}(F(\tilde{\varphi}))$. □

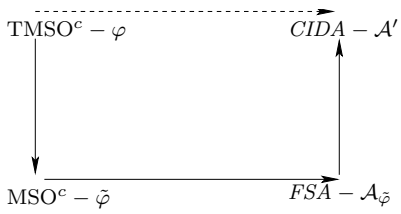


Figure 9.1: Route for going from TMSO^c to *CIDA*.

We can now complete the proof of Theorem 9.1 by taking the route in Figure 9.1. From Lemma 9.3, we have $L(\varphi) = tw(F(\tilde{\varphi}))$, where $\tilde{\varphi} = tmso\text{-}mso(\varphi)$. There exists an *FSA* $\mathcal{A}_{\tilde{\varphi}}$ such that $timing(L_{sym}(\mathcal{A}_{\tilde{\varphi}})) = F(\tilde{\varphi})$ by Theorem 9.2. Hence $L(\varphi) = tw(timing(L_{sym}(\mathcal{A}_{\tilde{\varphi}})))$. Now $L_{sym}(\mathcal{A}_{\tilde{\varphi}}) \cap \Gamma_1 \cdot (\Gamma_2 \cdot \Gamma_1)^*$ is a *CIDA*-language. Hence we have a *CIDA* $\mathcal{B}$ with the same symbolic language. Further, since the symbolic words not in $\Gamma_1 \cdot (\Gamma_2 \cdot \Gamma_1)^*$ do not contribute any timed words, we have that $tw(timing(L_{sym}(\mathcal{B}))) = tw(timing(L_{sym}(\mathcal{A}_{\tilde{\varphi}})))$. Therefore $L(\mathcal{B}) = tw(timing(L_{sym}(\mathcal{B}))) = tw(timing(L_{sym}(\mathcal{A}_{\tilde{\varphi}}))) = L(\varphi)$ and hence $\mathcal{B}$ is our required *CIDA*.

Chapter 10

A logical characterization of *rec-CIDA*'s

In this chapter we introduce the recursive versions of *CIDA*'s and TMSO^c, and show that the recursive logics characterize the recursive automata.

10.1 Recursive continuous input-determined automata

We now consider "recursive" *CIDA*'s. The main motivation is to increase the expressive power of our automata, as well as to characterize the expressiveness of recursive temporal logics which occur naturally in the real-time settings.

We define a *recursive* input-determined operator Δ over an alphabet Σ as a partial function from $(2^{\mathbb{R}_{\geq 0}} \times T\Sigma^* \times \mathbb{R}_{\geq 0})$ to $2^{\mathbb{R}_{\geq 0}}$, which is defined for tuples (X, σ, t) where $X \subseteq \mathbb{R}_{\geq 0}$, $\sigma \in T\Sigma^*$ and $t \in [0, \mathit{length}(\sigma)]$. Given a recursive operator Δ and a set $X \subseteq \mathbb{R}_{\geq 0}$, we denote by Δ_X, the operator whose semantics is given by $\Delta_X(\sigma, t) = \Delta(X, \sigma, t)$. We call a set $X \subseteq \mathbb{R}_{\geq 0}$ finitely varying if its characteristic function in the interval $[0, r]$ is finitely varying for all r. The characteristic function of X in an interval $[0, r]$, denoted f_X^r is given by $f_X^r(t) = 1$ iff $t \in X$. We call a recursive operator Δ *finitely varying* if for every finitely varying set X, Δ_X is a finitely varying operator.

Given a timed word σ in $T\Sigma^*$ and a $t \in [0, \mathit{length}(\sigma)]$ we call the pair (σ, t) a *floating timed word* over Σ. A floating timed language is then a set of floating timed words. We will use the notation Σ' for $(\Sigma \cup \{\epsilon\}) \times \{0, 1\}$. Given $\sigma' \in T\Sigma'^*$, we denote by σ the timed word obtained from σ' by projecting away the $\{0, 1\}$ component from each pair and then dropping any ϵ's in the resulting word. A timed word σ' over the alphabet Σ' which contains exactly

one symbol from $(\Sigma \cup \{\epsilon\}) \times \{1\}$, and whose last symbol is from $\Sigma \times \{0,1\}$, defines the floating timed word (σ, t) where t is the time of the unique action which has a 1-extension. We use *fw* to denote the (partial) map which given a timed word σ' over Σ' returns (σ, t), and extend it to apply to timed languages over Σ' in a natural way.

Let Σ be an alphabet and *Op* be a set of input determined operators. Given $\Delta \in Op$, we use the notation Δ' for the operator over Σ' with the semantics $\Delta'(\sigma', t) = \Delta(\sigma, t)$. We use the notation Op' to denote the set $\{\Delta' \mid \Delta \in Op\}$. We now define a *floating CIDA* over (Σ, Op) to be a *CIDA* over (Σ', Op'). We define the floating language of a floating *CIDA* $\mathcal{B}$, denoted $L^{fl}(\mathcal{B})$, as $fw(L(\mathcal{B}))$.

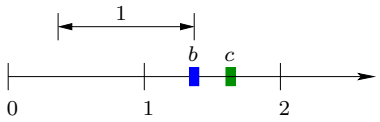
We define recursive continuous input determined automata (*rec-CIDA*'s) and floating recursive continuous input determined automata (*frec-CIDA*'s) over an alphabet Σ and a set of recursive operators *Rop* based on Σ. These are, respectively, the union of level i *rec-CIDA*'s and level i *frec-CIDA*'s for $i \in \mathbb{N}$, which are defined inductively as follows:

- A level 0 *rec-CIDA* $\mathcal{A}$ is a *CIDA* over Σ that uses only the guard $\top$. It accepts the timed language $L(\mathcal{A})$. A level 0 *frec-CIDA* $\mathcal{B}$ is a floating *CIDA* over Σ which uses only the guard $\top$. It accepts the floating language $L^{fl}(\mathcal{B})$.

- Let C be a finite collection of *frec-CIDA*'s of level i or less over (Σ, Rop). Let *Op* be the set of operators $\{\Delta_{\mathcal{B}} \mid \Delta \in Rop, \mathcal{B} \in C\}$, where the semantics of each $\Delta_{\mathcal{B}}$ is defined as follows. Let $pos(\sigma, \mathcal{B}) = \{t \in [0, length(\sigma)] \mid (\sigma, t) \in L^{fl}(\mathcal{B})\}$. Then $\Delta_{\mathcal{B}}(\sigma, t) = \Delta(pos(\sigma, \mathcal{B}), \sigma, t)$. We say that an operator $\Delta_{\mathcal{B}}$ is of level j if $\mathcal{B}$ is a level j *frec-CIDA*.

  A level $i+1$ *rec-CIDA*(Σ, Rop) is a *CIDA*(Σ, Op) which uses at least one operator of level i. And a level $i+1$ *frec-CIDA*(Σ, Rop) is a floating *CIDA*(Σ, Op) which uses at least one operator of level i.

Example 10.1 *Figure 10.1 give a rec-CIDA based on the recursive operator $\Diamond$, which accepts the timed language over $\{b, c\}$ consisting of timed words which contain a point corresponding to no action such that there is a b at distance one from it which is immediately followed by a c. Given a set X, $\Diamond_X$ is the input-determined operator which maps (σ, t) to the set of distances to the points in X which are after t. For example the following timed word is accepted by the rec-CIDA.*



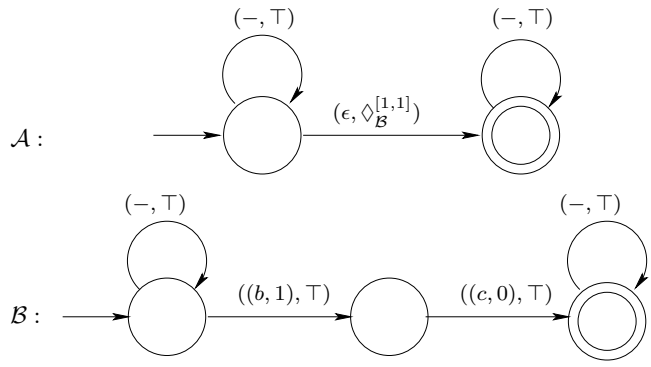


Figure 10.1: Example of a *rec-CIDA*

The frec-CIDA $\mathcal{B}$ accepts the floating timed language consisting of floating words (σ, t) such that the timed word σ contains a b at time t and is immediately followed by a c. The rec-CIDA $\mathcal{A}$ then accepts a timed word σ if it contains no action at some time t and the floating timed word $(\sigma, t+1)$ is accepted by $\mathcal{B}$.

10.2 Recursive timed monadic second order logic

In this section we introduce the recursive version of TMSO^c. The logic is parameterized by an alphabet Σ and a set of recursive operators *Rop*, and denoted by *rec-TMSO*$^c(\Sigma, \mathit{Rop})$. Given an alphabet Σ and a set of recursive operators *Rop*, the set of formulas of *rec-TMSO*$^c(\Sigma, \mathit{Rop})$ are defined inductively as:

$$\varphi ::= Q_a(x) \mid \Delta_\psi^I(x) \mid x < y \mid x \in X \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x\varphi \mid \exists X\varphi,$$

where $a \in \Sigma$, $\Delta \in \mathit{Rop}$, $I \in \mathcal{I}_{\mathbb{Q}}$ and ψ is a *rec-TMSO*^{c} formula with a single free variable z.

The logic is interpreted over timed words in $T\Sigma^*$. If φ contains no predicates of the form "$\Delta_\psi^I(x)$", then $\sigma, \mathbb{I} \models \varphi$ is defined as for TMSO^c. Inductively we assume that $\sigma, \mathbb{I} \models \psi$ is defined where ψ has a single free variable z. Let $\mathit{pos}(\sigma, \psi) = \{t \mid \sigma, [t/z] \models \psi\}$ be the set of interpretations of z which make ψ true in σ. We then consider Δ_ψ as an input-determined operator with the semantics $\Delta_\psi(\sigma, t) = \Delta(\mathit{pos}(\sigma, \psi), \sigma, t)$. The rest of the interpretation is similar to TMSO^c. We note that each *rec-TMSO*$^c(\Sigma, \mathit{Rop})$ formula can be

viewed as a TMSO$^c(\Sigma, Op)$ formula, where Op is the set of Δ_ψ's which have a *top-level* occurrence, i.e., they are not in the scope of any other Δ operator.

We now define the level of a *rec-TMSO*$^c(\Sigma, Rop)$ formula.

- A level 0 *rec-TMSO*$^c(\Sigma, Rop)$ formula is a formula in TMSO$^c(\Sigma, \emptyset)$.
- Let C be the set of *rec-TMSO*$^c(\Sigma, Rop)$ formulas of level i or less with exactly one free variable z. Then a level $i+1$ *rec-TMSO*$^c(\Sigma, Rop)$ formula is a formula in TMSO$^c(\Sigma, Op)$, where $Op = \{\Delta_\psi \mid \Delta \in Rop, \psi \in C\}$, which uses at least one level i operator, i.e., operator of the form Δ_ψ where ψ is a level i formula.

Note that a variable z which is free in ψ is not free in the formula $\Delta^I_\psi(x)$. A *rec-TMSO*$^c(\Sigma, Rop)$ sentence φ defines the language $L(\varphi) = \{\sigma \in T\Sigma^* \mid \sigma \models \varphi\}$. A *rec-TMSO*$^c(\Sigma, Rop)$ formula ψ with one free variable z defines the floating language $L^{fl}(\psi) = \{(\sigma, t) \mid \sigma, [t/z] \models \psi\}$.

Example 10.2 *We give a rec-TMSO^c sentence based on the recursive input-determined operator $\Diamond$ which defines the timed language in Example 10.1. The formula ψ with one free variable z defines the floating timed language accepted by $\mathcal{B}$. And the sentence φ defines the timed language of $\mathcal{A}$. ψ is given by $Q_b(z) \wedge \exists x(z < x \wedge Q_c(x) \wedge \neg\exists y(z < y \wedge y < x \wedge Q_b(y)))$ and φ is given by $\exists x(\neg Q_b(x) \wedge \neg Q_c(x) \wedge \Diamond^{[1,1]}_\psi(x))$.*

10.3 MSO characterization of *rec-CIDA*

In this section we show that *rec-TMSO^c* characterizes *rec-CIDA*'s. We first show the equivalence of floating *CIDA*'s and TMSO^c formulas with one free variable. Also, we show that TMSO^c formulas with one free variable are finitely varying. We then use the above results in showing the MSO characterization of *rec-CIDA*'s.

10.3.1 Relating floating *CIDA*'s and TMSO^c

We first show the equivalence of the class of floating timed languages accepted by floating automata and that definable by TMSO^c formulas with one free variable.

Lemma 10.1 *Let L be a floating timed language over an alphabet Σ. Let Op be a set of finitely varying operators based on Σ. Then $L = L^{fl}(\mathcal{B})$ for some floating-CIDA $\mathcal{B}$ over (Σ, Op) iff $L = L^{fl}(\psi)$ for some* TMSO$^c(\Sigma, Op)$ *formula ψ with one free variable.*

Proof ($\Rightarrow$) Let $\mathcal{B} = (Q, s, \delta, F, inv)$ be a floating *CIDA* over (Σ, Op) such that $L = L^{fl}(\mathcal{B})$. Then $\mathcal{B}$ is a *CIDA* over (Σ', Op'). Let $\delta = \{e_1, \cdots, e_m\}$. We now give the TMSO$^c(\Sigma, Op)$ formula $\psi_{\mathcal{B}}$ with one free variable z such that $L^{fl}(\mathcal{B}) = L^{fl}(\psi_{\mathcal{B}})$. The construction of the formula is similar to that of Theorem 9.1 except for the following changes. Let $g(x)$ be the TMSO^c formula obtained from g by replacing each Δ'^I in it by $\Delta^I(x)$. We replace φ_6 and φ_7 by the following and take the conjunction of the resulting formula with φ_9.

$$\varphi_6 : \forall x \bigwedge_{(p,((a,i),g),q)\in\delta} (x \in X_{(p,((a,i),g),q)} \Rightarrow (Q_a(x) \wedge g(x))).$$

$$\varphi_7 : \forall x \bigwedge_{(p,(\epsilon,g),q)\in\delta} (x \in X_{(p,(\epsilon,g),q)} \Rightarrow (\neg action(x) \wedge g(x))) \wedge$$

$$\forall x \bigwedge_{(p,((\epsilon,i),g),q)\in\delta} (x \in X_{(p,((\epsilon,i),g),q)} \Rightarrow (\neg action(x) \wedge g(x))).$$

$$\varphi_9 : \forall x((\bigvee_{(p,((c,1),g),q)\in\delta} x \in X_{(p,((c,1),g),q)}) \Leftrightarrow x = z).$$

($\Leftarrow$) In the other direction, let ψ be a formula in TMSO$^c(\Sigma, Op)$ with one free variable z. Let (Γ_1, Γ_2) be the proper symbolic alphabet over (Σ, Op) based on the set of atomic guards G in ψ. Let $\Gamma = \Gamma_1 \cup \Gamma_2$. We obtain from Lemma 9.2 that $\sigma, [t/z] \models \psi$ iff $f, [t/z] \models tmso\text{-}mso(\psi)$, where $\sigma \in T\Sigma^*$, $f \in func(\Gamma)$ and $\sigma \in tw(f)$. We can now use the construction in Theorem 9.2 to build the automaton $\mathcal{A}_{\psi}^{X,Y}$, where $X = \{z\}$ and $Y = \emptyset$, such that it contains the untimings of $f_{[t/z]}^{X,Y}$, where $f, [t/z] \models \psi$. The automaton is over the alphabet $\Gamma \times \{0, 1\}$, and every symbol in it is of the form $((c, h), i)$ or (h, i), where $c \in \Sigma \cup \{\epsilon\}$, $h \subseteq G$ and $i \in \{0, 1\}$. But it can be seen that words which are the untiming of some $f_{\mathbb{I}}^{X,Y}$ start and end with symbols of the form $((c, h), i)$ and alternate between those of the form $((c, h), i)$ and $(h, 0)$. Hence we intersect the *FSA* with an automaton which removes such words. Note that this does not alter the floating timed language obtained by first timing the words, then interpreting them as pairs (f, t) and finally extracting the floating timed words. Therefore it accepts the floating timed language of ψ. Next we replace in the resulting automaton every $((c, h), i)$ by $((c, i), h'')$, and (h, i) (note that we have removed (h, i)'s where i is 1) by h'', where h'' is obtained from h by replacing every operator Δ by Δ'. Since this is a *CIDA* language over (Σ', Op'), we can give a proper *CIDA*(Σ', Op') whose symbolic language is the same. Hence we can give a *CIDA*(Σ', Op') whose timed language is the same. This can be viewed as a floating *CIDA*(Σ, Op).

We can now argue that the floating timed language of this floating *CIDA* is same as that of ψ. □

10.3.2 Finite variability of TMSO^c

We now prove the finite variability of TMSO^c formulas. We call a formula ψ in $\mathrm{TMSO}^c(\Sigma, Op)$ with one free variable finitely varying if for every $\sigma \in T\Sigma^*$, $pos(\sigma, \psi)$ is a finitely varying set. If we can show that ψ is finitely varying, then the operator Δ_ψ, where Δ is a finitely varying recursive operator, would be finitely varying.

Lemma 10.2 *Let Σ be an alphabet and Op be a set of finitely varying operators. Every formula $\psi \in \mathrm{TMSO}^c(\Sigma, Op)$ with a single free variable z is finitely varying.*

Proof Let $\sigma \in T\Sigma^*$ and let $X = pos(\sigma, \psi)$. To show the finite variability of ψ, we need to show that $f_X^{length(\sigma)}$, the characteristic function of X in the interval $[0, length(\sigma)]$ is finitely varying. We will show that the interval sequences of the canonical interval representations of the functions $f_X^{length(\sigma)}$ and f_σ^Γ are the same, where $\Gamma = \Gamma_1 \cup \Gamma_2$ and (Γ_1, Γ_2) is the proper symbolic alphabet over (Σ, Op) based on G, the set of atomic guards in ψ. Let the canonical interval representation of f_σ^Γ be $(a_0, I_0) \cdots (a_{2n}, I_{2n})$. We will show that if $t_1, t_2 \in I_i$ then $t_1 \in pos(\sigma, \psi)$ iff $t_2 \in pos(\sigma, \psi)$. This is because $t_1 \in pos(\sigma, \psi)$ iff $\sigma, [t_1/z] \models \psi$ iff $f_\sigma^\Gamma, [t_1/z] \models tmso\text{-}mso(\psi)$ iff $untiming(f^\Gamma_{\sigma\,[t_1/z]}) \in \mathcal{A}_{tmso\text{-}mso(\psi)}$ iff $untiming(f^\Gamma_{\sigma\,[t_2/z]}) \in \mathcal{A}_{tmso\text{-}mso(\psi)}$ (since $untiming(f^\Gamma_{\sigma\,[t_1/z]}) = untiming(f^\Gamma_{\sigma\,[t_2/z]})$) iff $f_\sigma^\Gamma, [t_2/z] \models tmso\text{-}mso(\psi)$ iff σ, $[t_2/z] \models \psi$ iff $t_2 \in pos(\sigma, \psi)$. □

10.3.3 *rec-TMSO*^{c} characterizes *rec-CIDA*'s

The following theorem states that *rec-TMSO*^{c} characterizes *rec-CIDA*'s. We also obtain from the proof of the theorem that there is a level by level correspondence between the automata and logics.

Theorem 10.1 *Let Σ be a finite alphabet and let Rop be a set of finitely varying recursive operators over Σ. Then $L \subseteq T\Sigma^*$ is accepted by a rec-CIDA over (Σ, Rop) iff L is definable by a rec-TMSO$^c(\Sigma, Rop)$ sentence.*

Proof We will show that for each i, the class of *rec-CIDA*'s of level i correspond to the sentences of *rec-TMSO*^{c} of level i. We will use induction on the level of automata and formulas to argue that

1. $L \subseteq T\Sigma^*$ is accepted by a level i *rec-CIDA*(Σ, Rop) iff L is definable by a level i *rec-TMSO*$^c(\Sigma, Rop)$ sentence φ.

2. A floating timed language L over Σ is accepted by a level i *frec-CIDA* over (Σ, Rop) iff L is definable by a level i *rec-TMSO*$^c(\Sigma, Rop)$ formula ψ with one free variable.

3. Level i *rec-TMSO*$^c(\Sigma, Rop)$ formulas with one free variable and level i *frec-CIDA*(Σ, Rop)'s are finitely varying. A $\mathcal{B} \in$ *frec-CIDA*(Σ, Rop) is called finitely varying if $pos(\sigma, \mathcal{B})$ is finitely varying for every $\sigma \in T\Sigma^*$.

For the base case of the first part of the claim, we consider level 0 automata and formulas. They correspond to $CIDA(\Sigma, \emptyset)$'s and $\mathrm{TMSO}^c(\Sigma, \emptyset)$ sentences, which by Theorem 9.1 are expressively equivalent.

For the second part of the claim we consider level 0 *frec-CIDA*(Σ, Rop)'s and *rec-TMSO*$^c(\Sigma, Rop)$ formulas with one free variable, which correspond to floating $CIDA(\Sigma, \emptyset)$'s and $\mathrm{TMSO}^c(\Sigma, \emptyset)$ formulas with one free variable, which are expressively equivalent by Lemma 10.1.

The level 0 *rec-TMSO*$^c(\Sigma, Rop)$ formulas with one free variable are finitely varying by Lemma 10.2, and hence the level 0 *frec-CIDA*(Σ, Rop)'s are finitely varying.

Turning now to the induction step, let $\mathcal{A}$ be a level $i+1$ *rec-CIDA* over (Σ, Rop). Then it is a $CIDA(\Sigma, Op)$ for some set of level i or less operators Op, with at least one level i operator. The operators in Op are finitely varying, since they are formed from a finitely varying recursive operator and a level i or less *frec-CIDA* which by induction hypothesis is finitely varying. From Theorem 9.1, we can get an equivalent $\mathrm{TMSO}^c(\Sigma, Op)$ sentence. For each $\Delta_{\mathcal{B}} \in Op$, $\mathcal{B}$ is a level j *rec-CIDA*(Σ, Rop) for some $j < i+1$. Hence by induction hypothesis there exists a level j *rec-TMSO*$^c(\Sigma, Op)$ formula with one free variable such that $L^{fl}(\psi) = L^{fl}(\mathcal{B})$. Hence $pos(\sigma, \psi) = pos(\sigma, \mathcal{B})$. Therefore the semantics of the operators $\Delta_{\mathcal{B}}$ and Δ_ψ are the same, and hence we can replace $\Delta_{\mathcal{B}}$ by Δ_ψ to get a level $i+1$ *rec-TMSO*$^c(\Sigma, Op)$ sentence.

Conversely, let φ be a level $i+1$ sentence in *rec-TMSO*$^c(\Sigma, Rop)$. It is then a $\mathrm{TMSO}^c(\Sigma, Op)$ formula for some set of level i or less operators Op. The operators are finitely varying by induction hypothesis. By Theorem 9.1, we can get an equivalent $CIDA(\Sigma, Op)$. Now we can replace each Δ_ψ in Op by $\Delta_{\mathcal{B}}$, where the floating timed languages of ψ and $\mathcal{B}$ are same, which exists by Lemma 10.1.

The arguments for inductive part of the second claim are similar.

Now turning to the third claim, let ψ be a level $i+1$ *rec-TMSO*$^c(\Sigma, Rop)$ formula with one free variable. It is then a TMSO$^c(\Sigma, Op)$ formula with one free variable for some set of operators Op formed by finitely varying recursive operators and level i or less *rec-TMSO*^{c} formulas. Hence by induction hypothesis, the operators in Op are finitely varying. From Lemma 10.2, the formula ψ itself in finitely varying. Further, for every level $i+1$ *frec-CIDA*(Σ, Rop) we can give a floating timed language equivalent level $i+1$ *rec-TMSO*$^c(\Sigma, Rop)$ formula with one free variable. Hence level $i+1$ *frec-CIDA*(Σ, Rop) are finitely varying. □

Chapter 11

Expressive completeness of MTL^c_S and $\mathrm{MTL}^c_{S_I}$

In this chapter we show the expressive completeness of linear-time temporal logics based on input determined operators, and their recursive versions. From this we will be able to infer the expressive completeness of MTL_S and MTL_{S_I} in the continuous semantics.

11.1 Continuous timed linear temporal logic

In this section we identify a natural, expressively complete, timed linear temporal logic based on a set of input-determined operators. The logic is denoted $\mathrm{TLTL}^c(\Sigma, Op)$, parameterized by the alphabet Σ and the set of input-determined operators Op over Σ. The formulas of TLTL^c are given by:

$$\theta ::= a \mid \Delta^I \mid (\theta U \theta) \mid (\theta S \theta) \mid \neg\theta \mid (\theta \vee \theta),$$

where $a \in \Sigma$, $\Delta \in Op$ and $I \in \mathcal{I}_{\mathbb{Q}}$. We interpret $\mathrm{TLTL}^c(\Sigma, Op)$ formulas over timed words over Σ. Let φ be a $\mathrm{TLTL}^c(\Sigma, Op)$ formula. Let $\sigma \in \mathrm{T}\Sigma^*$, with $\sigma = (a_1, t_1)\cdots(a_n, t_n)$, and let $t \in [0, \mathit{length}(\sigma)]$. Then the satisfaction relation $\sigma, t \models \varphi$ is given by:

$$\begin{array}{lll}
\sigma, t \models a & \text{iff} & \exists i : t_i = t, a_i = a. \\
\sigma, t \models \Delta^I & \text{iff} & \Delta(\sigma, t) \cap I \neq \emptyset. \\
\sigma, t \models \theta U \eta & \text{iff} & \exists t' : t < t' \leq \mathit{length}(\sigma), \sigma, t' \models \eta, \forall t'' : t < t'' < t', \sigma, t'' \models \theta. \\
\sigma, t \models \theta S \eta & \text{iff} & \exists t' : 0 \leq t' < t, \sigma, t' \models \eta, \text{ and } \forall t'' : t' < t'' < t, \sigma, t'' \models \theta. \\
\sigma, t \models \neg\theta & \text{iff} & \sigma, t \not\models \theta. \\
\sigma, t \models (\theta \vee \eta) & \text{iff} & \sigma, t \models \theta \text{ or } \sigma, t \models \eta.
\end{array}$$

The language defined by a $\mathrm{TLTL}^c(\Sigma, Op)$ formula θ is given by $L(\theta) = \{\sigma \in T\Sigma^* \mid \sigma, 0 \models \theta\}$.

11.2 TFO^c characterizes TLTL^c

In this section we show that TLTL^c is expressively equivalent to the first-order fragment of TMSO^c. Let us denote by $\mathrm{TFO}^c(\Sigma, Op)$ the first order fragment of $\mathrm{TMSO}^c(\Sigma, Op)$ (i,e, the fragment we get by disallowing quantification over set variables). The logics TLTL^c and TFO^c are expressively equivalent in the following sense:

Theorem 11.1 *Let Σ be an alphabet and Op be a set of finitely varying input-determined operators over Σ. A timed language $L \subseteq T\Sigma^*$ is definable by a $\mathrm{TLTL}^c(\Sigma, Op)$ formula θ iff it is definable by a sentence φ in $\mathrm{TFO}^c(\Sigma, Op)$.*

Proof ($\Rightarrow$) Given a $\mathrm{TLTL}^c(\Sigma, Op)$ formula θ we can associate with it a $\mathrm{TFO}^c(\Sigma, Op)$ formula φ with a single free variable z, such that $\sigma, t \models \theta$ iff $\sigma, [t/z] \models \varphi$. This can be done in a straightforward inductive manner as follows. For the atomic formulas a and Δ^I, we can take φ to be $Q_a(z)$ and $\Delta^I(z)$ respectively. In the inductive step, assuming we have already translated θ_1 and θ_2 into η and ψ respectively, such that they do not use the variables x and y, we can translate $\theta_1 U \theta_2$ into

$$\exists x(z < x \wedge \psi[x/z] \wedge \forall y((z < y \wedge y < x) \Rightarrow \eta[y/z])).$$

Here $\psi[x/z]$ denotes the standard substitution of the free variable z by x in ψ. We can similarly write the translations for S and the boolean operators. It is not difficult to verify that if φ is the above translation of θ then $\sigma, t \models \theta$ iff $\sigma, [t/z] \models \varphi$. It also follows that $\sigma, 0 \models \theta$ iff σ satisfies the sentence φ_0 given by $\forall z(\mathit{first}(z) \Rightarrow \varphi)$. Hence we have that $L(\theta) = L(\varphi_0)$.

($\Leftarrow$) Let φ be a $\mathrm{TFO}^c(\Sigma, Op)$ sentence. Let (Γ_1, Γ_2) be the proper symbolic alphabet over (Σ, Op) based on $G = \{\Delta^I \mid \Delta^I(x) \text{ is a subformula of } \varphi\}$. Let $\Gamma = \Gamma_1 \cup \Gamma_2$ and let $\sigma \in T\Sigma^*$.

We first give a translation from an $\mathrm{LTL}^c(\Gamma)$ formula θ to a $\mathrm{TLTL}^c(\Sigma, Op)$ formula $\mathit{ltl\text{-}tltl}(\theta)$ such that σ satisfies $\mathit{ltl\text{-}tltl}(\theta)$ iff f_σ^Γ satisfies θ. In the translations of (ϵ, h) and h, we make sure that (ϵ, h) corresponds to a point of discontinuity whereas h does not.

Let $a \in \Sigma$, $h \subseteq G$ and $(a,h) \in \Gamma$. Let $g_h^G = \bigwedge_{h' \in h} h' \wedge \bigwedge_{h' \in G-h} \neg h'$.

$$\begin{array}{lcl}
\mathit{ltl\text{-}tltl}((a,h)) & = & a \wedge g_h^G. \\
\mathit{ltl\text{-}tltl}((\epsilon,h)) & = & \neg \bigvee_{a \in \Sigma} a \wedge g_h^G \wedge \neg(g_h^G S g_h^G \wedge g_h^G U g_h^G). \\
\mathit{ltl\text{-}tltl}(h) & = & \neg \bigvee_{a \in \Sigma} a \wedge g_h^G \wedge (g_h^G S g_h^G \wedge g_h^G U g_h^G). \\
\mathit{ltl\text{-}tltl}(\theta U \eta) & = & \mathit{ltl\text{-}tltl}(\theta) U \mathit{ltl\text{-}tltl}(\eta). \\
\mathit{ltl\text{-}tltl}(\theta S \eta) & = & \mathit{ltl\text{-}tltl}(\theta) S \mathit{ltl\text{-}tltl}(\eta). \\
\mathit{ltl\text{-}tltl}(\neg\theta) & = & \neg \mathit{ltl\text{-}tltl}(\theta). \\
\mathit{ltl\text{-}tltl}(\theta \vee \eta) & = & \mathit{ltl\text{-}tltl}(\theta) \vee \mathit{ltl\text{-}tltl}(\eta).
\end{array}$$

Lemma 11.1 *Let $\sigma \in T\Sigma^*$ and let $\theta \in \mathrm{LTL}^c(\Gamma)$. Then $f_\sigma^\Gamma, t \models \theta$ iff $\sigma, t \models \mathit{ltl\text{-}tltl}(\theta)$.* □

We complete the proof by factoring through Kamp's result for classical LTL^c. Recall that the syntax of LTL$^c(A)$ is given by:

$$\theta ::= a \mid (\theta U \theta) \mid (\theta S \theta) \mid \neg\theta \mid (\theta \vee \theta),$$

where $a \in A$. The logic is interpreted over finitely varying functions $f \in \mathit{func}(A)$. Given $t \in [0, \mathit{length}(f)]$ and $\theta \in \mathrm{LTL}^c(A)$, the satisfaction relation $f, t \models \theta$ is inductively defined as:

$$\begin{array}{lll}
f,t \models a & \text{iff} & f(t) = a. \\
f,t \models \theta U \eta & \text{iff} & \exists t' : t < t' \leq \mathit{length}(f), f, t' \models \eta, \forall t'' : t < t'' < t', f, t'' \models \theta. \\
f,t \models \theta S \eta & \text{iff} & \exists t' : 0 \leq t' < t, f, t' \models \eta, \forall t'' : t' < t'' < t, f, t'' \models \theta. \\
f,t \models \neg\theta & \text{iff} & f,t \not\models \theta. \\
f,t \models (\theta \vee \eta) & \text{iff} & f,t \models \theta \text{ or } f,t \models \eta.
\end{array}$$

The language defined by an LTL$^c(A)$ formula θ is given by $F(\theta) = \{f \in \mathit{func}(A) |\ f, 0 \models \theta\}$. Let FO$^c(A)$ denote the first order fragment of MSO$^c(A)$. Then the result due to Kamp [14] states that:

Theorem 11.2 ([14]) *A language of finitely varying functions over A is definable by an* LTL$^c(A)$ *formula θ iff it is definable by a* FO$^c(A)$ *sentence φ.* □

We now put everything together, following the route in Figure 11.1. The function *tmso-mso* maps a TFO$^c(\Sigma, Op)$ formula to an FO$^c(\Gamma)$ formula. And it follows from Lemma 9.2 that $f_\sigma^\Gamma \models \mathit{tmso\text{-}mso}(\varphi)$ iff $\sigma \models \varphi$.

By Kamp's result, there exists a mapping *fo-ltl* which maps a FO$^c(\Gamma)$ formula to an equivalent LTL$^c(\Gamma)$ formula. Hence $F(\mathit{tmso\text{-}mso}(\varphi)) = F(\mathit{fo\text{-}ltl}(\mathit{tmso\text{-}mso}(\varphi)))$ and therefore $f_\sigma^\Gamma \models \mathit{tmso\text{-}mso}(\varphi)$ iff $f_\sigma^\Gamma \models \mathit{fo\text{-}ltl}(\mathit{tmso\text{-}mso}(\varphi))$.

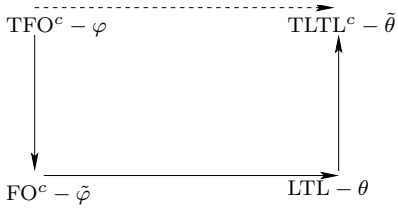


Figure 11.1: Route taken to go from TFO^c to TLTL^c.

Finally we have from Lemma 11.1 that $\sigma \models \mathit{ltl\text{-}tltl}(\mathit{fo\text{-}ltl}(\mathit{tmso\text{-}mso}(\varphi)))$ iff $f_\sigma^\Gamma \models \mathit{fo\text{-}ltl}(\mathit{tmso\text{-}mso}(\varphi))$.

Thus we have $\sigma \models \varphi$ iff $f_\sigma^\Gamma \models \mathit{tmso\text{-}mso}(\varphi)$ iff $f_\sigma^\Gamma \models \mathit{fo\text{-}ltl}(\mathit{tmso\text{-}mso}(\varphi))$ iff $\sigma \models \mathit{ltl\text{-}tltl}(\mathit{fo\text{-}ltl}(\mathit{tmso\text{-}mso}(\varphi)))$. This proves Theorem 11.1, since $L(\varphi) = L(\mathit{ltl\text{-}tltl}(\mathit{fo\text{-}ltl}(\mathit{tmso\text{-}mso}(\varphi))))$. □

11.3 *rec-TFO^c* characterizes *rec-TLTL^c*

We now define a recursive timed temporal logic along the lines of [9] and show that it is expressively complete. It is similar to the logic TLTL^c, and is parameterized by an alphabet Σ and a set of recursive input determined operators *Rop*, and is denoted *rec-TLTL^c*(Σ, Rop). The syntax of the logic is given by

$$\theta ::= a \mid \Delta_\theta^I \mid (\theta U \theta) \mid (\theta S \theta) \mid \neg\theta \mid (\theta \vee \theta),$$

where $a \in \Sigma$, $\Delta \in \mathit{Rop}$ and $I \in \mathcal{I}_{\mathbb{Q}}$.

The logic is interpreted over timed words in a manner similar to TLTL^c. The predicate Δ_θ^I is interpreted as follows. If θ does not use a Δ predicate, then the satisfaction relation $\sigma, t \models \theta$ is defined as for TLTL^c. Inductively assuming the semantics of *rec-TLTL^c*(Σ, Rop) formula θ has been defined, and setting $\mathit{pos}(\sigma, \theta) = \{t \in \mathbb{R}_{\geq 0} \mid \sigma, t \models \theta\}$, the operator Δ_θ is interpreted as an input determined operator with the semantics, $\Delta_\theta(\sigma, t) = \Delta(\mathit{pos}(\sigma, \theta), \sigma, t)$. The satisfaction relation $\sigma, t \models \Delta_\theta^I$ is then defined as for TLTL^c.

Once again, since Δ_θ behaves like an input determined operator, each formula in *rec-TLTL^c*(Σ, Rop) is also a TLTL$^c(\Sigma, \mathit{Op})$ formula, for an appropriately chosen set of input determined operators *Op*, containing operators of the form Δ_θ. We define the level of a TLTL$^c(\Sigma, \mathit{Op})$ formula in the expected way. A *rec-TLTL^c* formula naturally defines both a timed language $L(\theta) = \{\sigma \in T\Sigma^* \mid \sigma, 0 \models \theta\}$ and a floating language $L^{fl}(\theta) = \{(\sigma, t) \mid \sigma, t \models \theta\}$.

Let us use *rec-TFO*$^c(\Sigma, Rop)$ to denote the first order fragment of the logic *rec-TMSO*$^c(\Sigma, Rop)$. Then we have the following expressiveness result:

Theorem 11.3 *Let Σ be an alphabet and Rop be a set of finitely varying recursive operators. Then rec-TLTL*$^c(\Sigma, Rop)$ *is expressively equivalent to rec-TFO*$^c(\Sigma, Rop)$.

Proof We show by induction on i that

1. A timed language $L \subseteq T\Sigma^*$ is definable by a level i *rec-TLTL*$^c(\Sigma, Rop)$ formula iff it is definable by a level i *rec-TFO*$^c(\Sigma, Rop)$ sentence.

2. A floating timed language over Σ is definable by a *rec-TLTL*$^c(\Sigma, Rop)$ formula of level i iff it is definable by a level i *rec-TFO*$^c(\Sigma, Rop)$ formula with one free variable.

3. Level i *rec-TLTL*$^c(\Sigma, Rop)$ formulas and level i *rec-TFO*$^c(\Sigma, Rop)$ formulas with one free variable are finitely varying. A *rec-TLTL*$^c(\Sigma, Rop)$ formula θ is finitely varying if $pos(\sigma, \theta)$ is finitely varying for every σ.

We need to make use of the following result due to Kamp:

Theorem 11.4 ([14]) *For any* FO$^c(A)$ *formula ψ with one free variable z, there is an* LTL$^c(A)$ *formula θ such that for each $f \in func(A)$ and $t \in [0, length(f)]$, $f, [t/z] \models \psi$ iff $f, t \models \theta$.*

The proof is now similar to Theorem 10.1 and uses Theorem 11.1. □

11.4 Expressive completeness of MTL

In this section we show that MTL^c_S and MTL$^c_{S_I}$ are expressively complete with respect to *CIDA*'s based on a suitably defined set of operators.

We first identify recursive timed temporal logics corresponding to MTL$_S$ and MTL^c_S. We define below the semantics of the recursive operators $\Diamond$ and $\Diamondsuit$.

$$\begin{aligned} \Diamond(X, \sigma, t) &= \{t' - t \mid t' \geq t, t' \in X\}. \\ \Diamondsuit(X, \sigma, t) &= \{t - t' \mid t' \leq t, t' \in X\}. \end{aligned}$$

Lemma 11.2 MTL^c_S *over Σ and rec-TLTL*$^c(\Sigma, \{\Diamond\})$ *define the same class of timed languages.*

Proof We first observe that $\mathrm{MTL}_S^c(\Sigma)$ is expressively equivalent to its sublogic $\mathrm{MTL}_S^{c\,\Diamond}(\Sigma)$ in which the modality U_I is replaced by the modalities U and $\Diamond_I$. We give below the translation for U_I. We use $\rangle$ to denote) or].

$$\theta U_I \eta = \begin{cases} \Diamond_I \eta \wedge \neg\Diamond_{(0,a)}\neg\theta \wedge \Diamond_{[a,a]}(\eta \vee (\theta \wedge (\theta U \eta))) & \text{if } I = [a,b\rangle, a > 0 \\ \Diamond_I \eta \wedge \neg\Diamond_{(0,a]}\neg\theta \wedge \Diamond_{[a,a]}(\theta U \eta) & \text{if } I = (a,b\rangle, a > 0 \\ \Diamond_I \eta \wedge (\eta \vee (\theta U \eta)) & \text{if } I = [0,b\rangle \\ \Diamond_I \eta \wedge (\theta U \eta) & \text{if } I = (0,b\rangle \end{cases}$$

We define the level of an $\mathrm{MTL}_S^{c\,\Diamond}$ formula as the depth of the $\Diamond$ nesting in the formula. For example $\Diamond(a \wedge \Diamond b \wedge \Diamond\Diamond c)$ has depth 3. It can now be shown that the class of timed languages defined by level i $\mathrm{MTL}_S^{c\,\Diamond}$ formulas over Σ and level i *rec-TLTL*$^c(\Sigma, \{\Diamond\})$ formulas are equivalent. The semantics of $\Diamond_\eta^I$ and $\Diamond_I \eta'$ are the same, where η is an $\mathrm{MTL}_S^{c\,\Diamond}$ formula and η' is an equivalent *rec-TLTL*^{c} formula of the same level. □

We can similarly see that:

Lemma 11.3 $\mathrm{MTL}_{S_I}^c$ *over* Σ *and rec-TLTL*$^c(\Sigma, \{\Diamond, \Diamondsuit\})$ *define the same class of timed languages.* □

We now prove that the recursive operators $\Diamond$ and $\Diamondsuit$ are finitely varying.

Lemma 11.4 *The recursive input-determined operators* $\Diamond$ *and* $\Diamondsuit$ *are finitely varying.*

Proof Let X be a finitely varying set, and let $I \in \mathcal{I}_{\mathbb{Q}}$ and $I = \langle l, r\rangle$. Let σ be a timed word. We will show that $\Diamond_X$ is a finitely varying operator. Consider the set $X' = \{t - l \,|\, t \in X\} \cup \{t - r \,|\, t \in X\} \cup X$. X' is also a finitely varying set. We can now argue in a straightforward manner that between two consecutive points in X' the satisfiability of $\Diamond_X^I$ does not change in σ. For the proof for finite variability of $\Diamondsuit$ we need to consider the set $X'' = \{t + l \,|\, t \in X\} \cup \{t + r \,|\, t \in X\} \cup X$. □

The following theorem states the expressive completeness of MTL_S^c and $\mathrm{MTL}_{S_I}^c$, which follows immediately from Theorem 11.3 and Lemmas 11.3 and 11.4.

Theorem 11.5 $\mathrm{MTL}_S^c(\Sigma)$ *is expressively equivalent to rec-TFO*$^c(\Sigma, \{\Diamond\})$ *and* $\mathrm{MTL}_{S_I}^c(\Sigma)$ *is expressively equivalent to rec-TFO*$^c(\Sigma, \{\Diamond, \Diamondsuit\})$. □

Chapter 12

Conclusion

In this chapter we summarize the work in the thesis and put forward some open problems which need to be explored.

12.1 Summary

In this thesis we have addressed the issues concerning the expressiveness of Metric Temporal Logic. We considered two traditional interpretation of MTL over timed words, namely, pointwise and continuous interpretations. First we gave a characterization of the languages definable in the logic for both pointwise and continuous semantics, in terms of a necessary counter-freeness property comparable to that of LTL. We also applied this property to derive some inexpressibility results.

The second problem we addressed was that of relative expressiveness. We considered some natural syntactic extension of MTL with past temporal operators and compared their relative expressiveness in both pointwise and continuous semantics. We summarize the results below.

- The continuous versions of the logic are strictly more expressive than the corresponding pointwise versions.
- In both the interpretations of the logic, the addition of past operators makes it strictly more expressive.
- Further, the addition of the time constrained since S_I makes it more expressive than the addition of the unconstrained since S, in the pointwise semantics.

The above results hold for interpretations over both finite models and infinite models.

The last problem considered in this thesis was that of the expressive completeness of MTL in the continuous semantics. Towards this a class of timed-automata based on input-determined operators called *CIDA*'s was defined. These were shown to be characterized by timed monadic second order logics TMSO^c based on input-determined operators. A class of natural timed temporal logics TLTL^c based on these operators was introduced, and was shown to be equivalent to the first-order fragment of TMSO^c. Then the recursive versions of these logics and automata were considered, and were shown to have a similar relation. Finally variants of MTL were shown to be expressively complete by showing their correspondence to recursive timed temporal logics. The following is a summary of the results:

- MTL^c_S was shown to be expressively complete with respect to *rec-CIDA*'s based on the operator $\Diamond$.
- MTL$^c_{S_I}$ was shown to be expressively complete with respect to the automata and logics based on the operators $\Diamond$ and $\Diamond\!\!\!\!-$.

12.2 Future work

In this section we shed some light on the work that remains to be done. With regard to counter-freeness our results are only for MTL without the past operators. It would be interesting to explore if a similar characterization exists for these logics with past operators.

Some relative expressiveness questions are still open. We intend to examine if adding S_I gives more power to the logic than adding S, in the continuous semantics.

Turning to the expressive completeness results we would like to give a direct relation between the automata and timed temporal logics along the lines of classical LTL and counter-free automata. The answer to this question might be to identify the notion of "counter-freeness" in the automata which might correspond to these logics.

Bibliography

[1] Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[2] Rajeev Alur, Tomás Feder, and Thomas A. Henzinger. The Benefits of Relaxing Punctuality. *Journal of the ACM*, 43(1):116–146, 1996.

[3] Rajeev Alur and Thomas A. Henzinger. Real-time Logics: Complexity and Expressiveness. In *LICS*, pages 390–401. IEEE Computer Society, 1990.

[4] Rajeev Alur and Thomas A. Henzinger. A Really Temporal Logic. *Journal of the ACM*, 41(1):181–204, 1994.

[5] Patricia Bouyer, Fabrice Chevalier, and Nicolas Markey. On the Expressiveness of TPTL and MTL. In Ramanujam and Sen [23], pages 432–443.

[6] J.R. Büchi. On a decision method in restricted second order arithmetic. In *Z. Math. Logik Grundlag. Math.*, pages 66–92, 1960.

[7] Deepak D'Souza and M. Raj Mohan. Eventual Timed Automata. In Ramanujam and Sen [23], pages 322–334.

[8] Deepak D'Souza and Pavithra Prabhakar. On the expressiveness of MTL in the pointwise and continuous semantics. Technical Report IISc-CSA-TR-2005-7, Indian Institute of Science, Bangalore 560012, India, May 2005. URL: `http://archive.csa.iisc.ernet.in/TR/2005/7/`.

[9] Deepak D'Souza and Nicolas Tabareau. On timed automata with input-determined guards. In Yassine Lakhnech and Sergio Yovine, editors, *FORMATS/FTRTFT*, volume 3253 of *Lecture Notes in Computer Science*, pages 68–83. Springer, 2004.

[10] Deepak D'Souza and P. S. Thiagarajan. Product interval automata: A subclass of timed automata. In C. Pandu Rangan, Venkatesh Raman, and R. Ramanujam, editors, *FSTTCS*, volume 1738 of *Lecture Notes in Computer Science*, pages 60–71. Springer, 1999.

[11] Orna Grumberg Edmund M. Clarke, Jr. and Doron A. Peled. *Model Checking*. MIT Press, 2000.

[12] Dov M. Gabbay, Amir Pnueli, Saharon Shelah, and Jonathan Stavi. On the temporal basis of fairness. In *POPL*, pages 163–173, 1980.

[13] Thomas A. Henzinger, Jean-Francois Raskin, and Pierre-Yves Schobbens. The Regular Real-Time Languages. In Kim Guldstrand Larsen, Sven Skyum, and Glynn Winskel, editors, *ICALP*, volume 1443 of *Lecture Notes in Computer Science*, pages 580–591. Springer, 1998.

[14] Johan Anthony Willem Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, University of California, Los Angeles, California, 1968.

[15] Ron Koymans. Specifying Real-Time Properties with Metric Temporal Logic. *Real-Time Systems*, 2(4):255–299, 1990.

[16] Slawomir Lasota and Igor Walukiewicz. Alternating Timed Automata. In Vladimiro Sassone, editor, *FoSSaCS*, volume 3441 of *Lecture Notes in Computer Science*, pages 250–265. Springer, 2005.

[17] Étienne Lozes. Adjuncts elimination in the static ambient logic. *Electr. Notes Theor. Comput. Sci.*, 96:51–72, 2004.

[18] Robert McNaughton and Seymour A. Papert. *Counter-Free Automata (M.I.T. research monograph no. 65)*. The MIT Press, 1971.

[19] Joël Ouaknine and James Worrell. On the decidability of metric temporal logic. In *LICS*, pages 188–197. IEEE Computer Society, 2005.

[20] Joël Ouaknine and James Worrell. On metric temporal logic and faulty turing machines. In Luca Aceto and Anna Ingólfsdóttir, editors, *FoSSaCS*, volume 3921 of *Lecture Notes in Computer Science*, pages 217–230. Springer, 2006.

[21] A. Pnueli. The temporal logic of programs. Technical report, Jerusalem, Israel, Israel, 1997.

[22] Alexander Moshe Rabinovich. Finite variability interpretation of monadic logic of order. *Theor. Comput. Sci.*, 275(1-2):111–125, 2002.

[23] R. Ramanujam and Sandeep Sen, editors. *FSTTCS 2005: Foundations of Software Technology and Theoretical Computer Science, 25th International Conference, Hyderabad, India, December 15-18, 2005, Proceedings*, volume 3821 of *Lecture Notes in Computer Science*. Springer, 2005.

[24] J. F. Raskin. *Logics, Automata and Classical Theories for Deciding Real-Time*. PhD thesis, FUNDP, Belgium, 1999.

[25] Jean-Francois Raskin and Pierre-Yves Schobbens. State clock logic: A decidable real-time logic. In *HART*, pages 33–47, 1997.

[26] Moshe Y. Vardi and Pierre Wolper. An automata-theoretic approach to automatic program verification (preliminary report). In *LICS*, pages 332–344. IEEE Computer Society, 1986.

[27] Pierre Wolper, Moshe Y. Vardi, and A. Prasad Sistla. Reasoning about infinite computation paths (extended abstract). In *FOCS*, pages 185–194. IEEE, 1983.