Logics Meet 1-Clock Alternating Timed Automata

Shankara Narayanan Krishna
Department of Computer Science & Engineering IIT Bombay, India
krishnas@cse.iitb.ac.in

Khushraj Madnani
Department of Computer Science & Engineering IIT Bombay, India
khushraj@cse.iitb.ac.in

Paritosh K. Pandya
School of Technology and Computer Science, TIFR, India
pandya@tcs.tifr.res.in

Abstract

This paper investigates a decidable and highly expressive real time logic QkMSO which is obtained by extending MSO[<] with guarded quantification using block of less than k metric quantifiers. The resulting logic is shown to be expressively equivalent to 1-clock ATA where loops are without clock resets, as well as, RatMTL, a powerful extension of MTL[U_I] with regular expressions. We also establish 4-variable property for QkMSO and characterize the expressive power of its 2-variable fragment. Thus, the paper presents progress towards expressively complete logics for 1-clock ATA.

2012 ACM Subject Classification Theory of computation → Modal and temporal logics

Keywords and phrases Metric Temporal Logic, Alternating Timed Automata, MSO, Regular Expressions, Expressive Completeness

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2018.39

Related Version A full version of the paper is available at [15], `https://arxiv.org/abs/1802.02514`.

1 Introduction

Since the inception of real-time logics and timed automata, the question of finding expressive timed logics which are also decidable has been a prominent concern. Both classical first-order/monadic second-order logics as well as temporal logics were extended with metric constraints. Expressive power of a logic is typically measured by comparing it with respect to other well established logics and automata as exemplified by the celebrated Büchi and Kamp theorems [11, 13]. In real-time scenario, Alur and Henzinger in 1990 asked whether First order logic of Distance FO[$<, +$], a hybrid logic TPTL[U, S], and the metric temporal logic MTL[$\mathsf{U_I}, \mathsf{S_I}$] all have the same expressive power [2]. It took 15 years to show that MTL[$\mathsf{U_I}, \mathsf{S_I}$] is less expressive than TPTL over timed words [18] [3]. It is only in the last few years, that extensions of MTL[$\mathsf{U_I}, \mathsf{S_I}$] which are expressively complete for FO[$<, +$] have been found [9, 8]. Unfortunately all these logics have undecidable satisfiability.

For establishing the decidability of satisfiability of a logic, often an effective reduction to some form of automata with decidable non-emptiness is used. Alur and Henzinger in 1991 came up with the sub-logic MITL which is MTL[$\mathsf{U_I}, \mathsf{S_I}$] where time interval constraints I are non-singular intervals [1]. They showed the decidability of this logic by reducing its formulae essentially to language equivalent non-deterministic timed automata. However,

29th International Conference on Concurrency Theory (CONCUR 2018).
Editors: Sven Schewe and Lijun Zhang; Article No. 39; pp. 39:1–39:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

the logic was expressively weak as compared to timed automata . Looking for a more expressive but decidable logic, Wilke in a seminal paper introduced the *Monadic Second Order logic of relative distance*, $\mathcal{L}\overleftrightarrow{\mathsf{d}}$, and showed that this had exactly the expressive power of non-deterministic timed automata [22]. Moreover, the logic had decidable satisfiability. Wilke also showed that $\mathcal{L}\overleftrightarrow{\mathsf{d}}$ subsumed the expressive power of temporal logic EMITL, which was MITL extended with a finite automaton modality. Unfortunately, for any such logic, the validity and model checking (against timed automata) are necessarily undecidable as non-deterministic timed automata have undecidable universality/language inclusion. In another important paper, Henzinger, Raskin and Schobbens related *Event Clock Logic* ECL extended with *automata modality* to recursive event clock automata [6]. Attempts in these works have been to match the expressive power of timed automata and to have decidable satisfiability. The alternative is to try to match expressive power of some decidable and boolean closed class of timed automata. 1-clock Alternating Timed Automata (1-ATA) over finite words are perhaps the largest boolean closed class of timed languages for which emptiness is known to be decidable [17], [16]. Utilizing this fact, Ouaknine and Worrell showed in their seminal work that satisfiability as well as model checking of $\mathsf{MTL}[\mathsf{U_I}]$ (with pointwise interpretation) over finite words is decidable. The result was proved by constructing a language equivalent 1-ATA for a formula of $\mathsf{MTL}[\mathsf{U_I}]$ [17]. Unfortunately, the logic turns out to have much less expressive power than 1-ATA. In a series of papers [20], [12] we have investigated decidable extensions of $\mathsf{MTL}[\mathsf{U_I}]$ with increasing expressive power culminating in RatMTL [21]. Logic RatMTL is a powerful extension of $\mathsf{MTL}[\mathsf{U_I}]$ allowing counting and regularity constraints. But in retrospect it also turns out to be less expressive than 1-ATA.

The quest for expressively complete real-time logics matching the power of 1-ATA has remained open for over 13 years. In this paper, we give a partial solution to this problem. We show expressive completeness of some classical and metric temporal logics for natural subclasses of 1-ATA. We define an extension of Monadic Second Order Logic $\mathsf{MSO}[<]$ over words by adding guarded quantification with blocks of at most $k-1$ metric quantifiers to give a real time logic QkMSO. An essential syntactic restriction is that no free second order variable occurs in the scope of a metric quantifier, and a metric quantifier block results into a formula with only one free variable. In this, we have been inspired by the logic Q2MLO (over continuous time) defined by Hirshfeld and Rabinovich [7] as well as Hunter [8]. A carefully defined syntax gives us a logic which allows only future time properties to be stated. Note that punctual constraints are permitted in QkMSO unlike Q2MLO.

We investigate the decidability and expressive power of QkMSO. Firstly, we define a subclass of 1-ATA called 1-ATA with reset-free loops (1-ATA-rfl). In these automata, there is no cycle involving clock reset. Thus, on any run each transition with reset occurs at most once. As our first main result, we show that a) QkMSO, b) the 1-ATA-rfl, and, c) the metric temporal logic RatMTL, introduced earlier in [21], are all expressively equivalent (with effective reductions). In the process, we also prove that QkMSO has four variable property over timed words. This can be seen akin to the famous 3-variable property of $\mathsf{FO}[<]$ over words [10].

For our second main result, we turn to the two variable fragment Q2MSO of QkMSO. For its automaton characterization, we introduce a syntactic restriction of conjunctive-disjunctiveness (C⊕D) in 1-ATA. Here, each ATA thread is either in conjunctive mode or in disjunctive mode at a time, and it can switch modes only on a reset transition. We show that a) Q2MSO, b) the C⊕D-1-ATA-rfl, and c) the sublogic FRatMTL of RatMTL have exactly the same expressive power. Here logic FRatMTL uses only restricted version FRat of the modality Rat of RatMTL. A similar modality was also defined earlier by Wilke [22] in logic EMITL. In summary, we show that

QkMSO	$\equiv$	1−ATA−rfl	$\equiv$	RatMTL
Q2MSO	$\equiv$	C$\oplus$D−1−ATA−rfl	$\equiv$	FRatMTL

These results make QkMSO to be amongst the highly expressive logics with decidable satisfiability and model checking problems.

2 Preliminaries

Let Σ be a finite set of propositions. A finite timed word over Σ is a tuple $\rho = (\sigma, \tau)$, where σ and τ are sequences $\sigma_1\sigma_2 \dots \sigma_n$ and $\tau_1\tau_2 \dots \tau_n$ respectively, with $\sigma_i \in \Gamma = 2^{\Sigma} \backslash \emptyset$, and $\tau_i \in \mathbb{R}_{\geq 0}$ for $1 \leq i \leq n$. For all $i \in dom(\rho)$, we have $\tau_i \leq \tau_{i+1}$, where $dom(\rho)$ is the set of positions $\{1, 2, \dots, n\}$ in the timed word. For convenience, we assume $\tau_1 = 0$. The σ_i's can be thought of as labeling positions i in $dom(\rho)$. For example, given $\Sigma = \{a, b, c\}$, $\rho = (\{a, c\}, 0)(\{a\}, 0.7)(\{b\}, 1.1)$ is a timed word. ρ is strictly monotonic iff $\tau_i < \tau_{i+1}$ for all $i, i+1 \in dom(\rho)$. Otherwise, it is weakly monotonic. The set of finite timed words over Σ is denoted $T\Sigma^*$. Given $\rho = (\sigma, \tau)$ with $\sigma = \sigma_1 \dots \sigma_n \in \Gamma^+$, σ^{single} denotes the set of all words $w_1 w_2 \dots w_n$ where each $w_i \in \sigma_i$. ρ^{single} consists of all timed words $(\sigma^{\mathsf{single}}, \tau)$. For the ρ as above, ρ^{single} consists of timed words $(\{a\}, 0)(\{a\}, 0.7)(\{b\}, 1.1)$ and $(\{c\}, 0)(\{a\}, 0.7)(\{b\}, 1.1)$.

2.1 Temporal Logics

In this section, we define preliminaries pertaining to the temporal logics studied in the paper. Let $I\nu$ be a set of open, half-open or closed time intervals. The end points of these intervals are in $\mathbb{N} \cup \{0, \infty\}$. Examples of such intervals are $[1, 3), [2, 2], [2, \infty)$. For a time stamp $\tau \in \mathbb{R}_{\geq 0}$ and an interval $\langle a, b\rangle$, where $\langle$ is left-open or left-closed and $\rangle$ is right-open or right-closed, $\tau + \langle a, b\rangle$ represents the interval $\langle \tau + a, \tau + b\rangle$.

Metric Temporal Logic (MTL [14]). Given a finite alphabet Σ, the formulae of logic MTL are built from Σ using boolean connectives and time constrained version of the until modality U as follows: $\varphi ::= a (\in \Sigma) \;|true\; |\varphi \wedge \varphi \mid \neg\varphi \mid \varphi \,\mathsf{U}_I \varphi$, where $I \in I\nu$. For a timed word $\rho = (\sigma, \tau) \in T\Sigma^*$, a position $i \in dom(\rho)$, and an MTL formula φ, the satisfaction of φ at a position i of ρ is denoted $\rho, i \models \varphi$, and is defined as follows: (i) $\rho, i \models a \leftrightarrow a \in \sigma_i$, (ii) $\rho, i \models \neg\varphi \leftrightarrow \rho, i \nvDash \varphi$, (iii) $\rho, i \models \varphi_1 \wedge \varphi_2 \leftrightarrow \rho, i \models \varphi_1$ and $\rho, i \models \varphi_2$, (iv) $\rho, i \models \varphi_1 \,\mathsf{U}_I \varphi_2 \leftrightarrow \exists j > i,\ \rho, j \models \varphi_2, \tau_j - \tau_i \in I$, and $\rho, k \models \varphi_1 \;\forall\; i < k < j$. The language of a MTL formula φ is $L(\varphi) = \{\rho \mid \rho, 1 \models \varphi\}$. Two formulae φ and ϕ are said to be equivalent denoted as $\varphi \equiv \phi$ iff $L(\varphi) = L(\phi)$. The subclass of MTL restricting the intervals I in the until modality to non-punctual intervals is denoted MITL. Punctual intervals like [2,2] are disallowed. Note that we restrict to until-only fragment of MTL for the sake of decidability.

▶ **Theorem 1** ([17]). MTL *satisfiability is decidable over finite timed words with non-primitive recursive complexity.*

MTL with Rational Expressions (RatMTL)

We first recall an extension of MTL with rational expressions (RatMTL), introduced in [21]. The modalities in RatMTL assert the truth of a rational expression (over subformulae) within a particular time interval with respect to the present point. For example, the formula

$\mathsf{Rat}_I(\varphi_1.\varphi_2)^+$ when evaluated at a point i, asserts the existence of $2k$ points with time stamps $\tau_{j+1} < \tau_{j+2} < \cdots < \tau_{j+2k}$, $k > 0$, such that τ_{j+1} and τ_{j+2k} are the first and last time stamps in $\tau_i + I$, respectively. φ_1 evaluates to true at τ_{j+2l+1}, and φ_2 evaluates to true at τ_{j+2l+2}, for all $0 \leq l < k$.

RatMTL Syntax. Formulae of RatMTL are built from a finite alphabet Σ as:
$\varphi ::= a(\in \Sigma) \mid true \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathsf{Rat}_I\mathsf{re(S)} \mid \mathsf{FRat}_{I,\mathsf{re(S)}}\varphi$, where $I \in I\nu$ and S is a finite set of RatMTL subformulae, and re(S) is defined as a rational expression over S. $\mathsf{re(S)} ::= \epsilon \mid \varphi(\in \mathsf{S}) \mid \mathsf{re(S).re(S)} \mid \mathsf{re(S)}+\mathsf{re(S)} \mid [\mathsf{re(S)}]^*$. Thus, RatMTL is MTL extended with modalities URat and Rat (RatMTL=MTL+Rat+FRat). An *atomic* rational expression re is any well-formed formula $\varphi \in$ RatMTL.

RatMTL Semantics. For a timed word $\rho = (\sigma, \tau) \in T\Sigma^*$, a position $i \in dom(\rho)$, a RatMTL formula φ, and a finite set S of subformulae of φ, we define the satisfaction of φ at a position i as follows. For positions $i < j \in dom(\rho)$, let $\mathsf{Seg}(\rho, \mathsf{S}, i, j)$ denote the untimed word over 2^{S} obtained by marking the positions $k \in \{i+1, \dots, j-1\}$ of ρ with $\psi \in \mathsf{S}$ iff $\rho, k \models \psi$. For a position $i \in dom(\rho)$ and an interval I, let $\mathsf{TSeg}(\rho, \mathsf{S}, I, i)$ denote the untimed word over 2^{S} obtained from ρ by marking all the positions k, where $\tau_k - \tau_i \in I$, with $\psi \in \mathsf{S}$ iff $\rho, k \models \psi$.

- $\rho, i \models \mathsf{FRat}_{I,\mathsf{re(S)}}\varphi \leftrightarrow \exists j>i,\ \rho, j \models \varphi, \tau_j - \tau_i \in I$ and, $[\mathsf{Seg}(\rho, \mathsf{S}, i, j)]^{\mathsf{single}} \cap L(\mathsf{re(S)}) \neq \emptyset$, where $L(\mathsf{re(S)})$ is the language of the rational expression re formed over the set S.
- $\rho, i \models \mathsf{Rat}_I\mathsf{re} \leftrightarrow [\mathsf{TSeg}(\rho, S, I, i)]^{\mathsf{single}} \cap L(\mathsf{re(S)}) \neq \emptyset$.

The language accepted by a RatMTL formula φ is given by $L(\varphi) = \{\rho \mid \rho, 1 \models \varphi\}$. The subclass of RatMTL using only the FRat modality is denoted FRatMTL (FRatMTL=MTL+FRat). If we stick to non-punctual intervals, the subclass obtained is FRatMITL (FRatMITL=MITL+FRat). Some remarks are in order.

1. In [21], the URat modality was used instead of FRat; however, both have the same expressiveness. Note that $\varphi_1\mathsf{URat}_{I,\mathsf{re(S)}}\varphi_2$ is equivalent to $\mathsf{FRat}_{I,\mathsf{re}'(\mathsf{S}\cup\{\varphi_1\})}\varphi_2$ where $\mathsf{re}'(\mathsf{S} \cup \{\varphi_1\}) = \mathsf{re(S)} \cap \varphi_1^*$.
2. The classical $\varphi_1\ \mathsf{U}_I\varphi_2$ modality can be written in FRatMTL as $\mathsf{FRat}_{I,\varphi_1^*}\varphi_2$. Also, it can be shown (see [21]) that the URat(and thus FRat) modality can be expressed using the Rat modality.

Modal depth. The modal depth (md) of a RatMTL formula is defined as follows. Let PL_Σ be the set of propositional logic formulae over Σ (up to equivalence). An atomic RatMTL formula over Σ is an element of PL_Σ and has modal depth 0. A RatMTL formula φ over Σ having a single modality (Rat or FRat) has modal depth one and has the form $\mathsf{Rat}_I\mathsf{re}$ or $\mathsf{FRat}_{I,\mathsf{re}}\psi$ where re is a regular expression over PL_Σ and $\psi \in \mathsf{PL}_\Sigma$. Inductively, we define modal depth as follows: (i) $\mathsf{md}(\varphi \wedge \psi) = \mathsf{md}(\varphi \vee \psi) = \max[\mathsf{md}(\varphi), \mathsf{md}(\psi)]$. Similarly, $\mathsf{md}(\neg\varphi) = \mathsf{md}(\varphi)$. (ii) Let re be a regular expression over the set of subformulae $\mathsf{S} = \{\psi_1, \dots, \psi_k\}$.
$\mathsf{md}(\mathsf{FRat}_{I,\mathsf{re}}(\varphi))$=1+ $\max(\psi_1, \dots, \psi_k, \varphi)$. Similarly $\mathsf{md}(\mathsf{Rat}_I(\mathsf{re}))$=1+ $\max(\psi_1, \dots, \psi_k)$.

▶ **Example 2.** Consider the formula $\varphi = \mathsf{Rat}_{[1,1]}(\mathsf{Rat}_{(0,1)}(aa)^*)$. Then $\varphi = \mathsf{Rat}_{[1,1]}\mathsf{re}_1$ where $\mathsf{re}_1 = \mathsf{Rat}_{(0,1)}(aa)^*$. The subformulae of interest are $\mathsf{S} = \{\mathsf{Rat}_{(0,1)}(aa)^*, a\}$. For ρ=$(\{a\}, 0)$ $(\{a, b\}, 0.9)$ $(\{a\}, 1)(\{a\}, 1.2)$, $\rho, 3 \not\models \mathsf{re}_1$, since $[\mathsf{TSeg}(\rho, \mathsf{S}, (0, 1), 3)]^{\mathsf{single}}$ has only the word a and hence $[\mathsf{TSeg}(\rho, S, (0, 1), 3)]^{\mathsf{single}} \cap L((aa)^*) = \emptyset$. Hence, $\rho, 1 \not\models \varphi$. On the other hand, for $\rho = (\{a\}, 0)(\{a\}, 0.3)$ $(\{a\}, 1)(\{a\}, 1.1)$ $(\{a\}, 1.8)$, $\rho, 1 \models \varphi$, since $aa \in [\mathsf{TSeg}(\rho, S, (0, 1), 3)]^{\mathsf{single}} \cap L((aa)^*)$.

2.2 MSO with guarded metric quantifiers QkMSO

Let $\rho = (\sigma, \tau)$ be a timed word over a finite alphabet Σ, as before. We define a real-time logic QkMSO (with parameter $k \in \mathbb{N}$) which is interpreted over such words. It includes MSO[<] over words σ relativized to specify only future properties. This is extended with a notion of time constraint formula $\psi(t_i)$. All variables in our logic range over positions in the timed word and not over time stamps. There are two sorts of formulae in QkMSO which are mutually recursively defined : these are MSO^{t_0} formulae ϕ which have no real-time constraints except time constraint subformulae $\psi(t_p)$. These subformulae $(\psi(t_p))$ have only one free variable t_p, which is a first order variable. Such a time constraint formula $\psi(t_p)$ consists of a block of real-time constrained quantification applied to a QkMSO formula with no free second order variables. This form of real time constraints in first order logic was pioneered by Hirshfeld and Rabinovich [7] in their logic Q2MLO, which we refer in this paper as Q2FO, and later used by Hunter [8]. Let $t_0, t_1, \ldots$ be first order variables and $T_0, T_1, \ldots$ the monadic second-order variables. We have a two sorted logic consisting of MSO formulae ϕ and time constrained formulae ψ. Let $a \in \Sigma$, and let t_i range over first order variables, while T_i range over second order variables. Each quantified first order variable in ϕ is relativized to the future of some variable, say t_0, called anchor variable, giving formulae of MSO^{t_0}. The syntax of $\phi \in \mathsf{MSO}^{t_0}$ is given by: $t_p{=}t_q \mid t_p{<}t_q \mid Q_a(t_p) \mid T_j(t_i) \mid \phi{\wedge}\phi \mid \neg\phi \mid \exists t'.t'{>}t_0 \wedge \phi \mid \exists T_i \phi \mid \psi(\mathbf{t_p})$.
Here, $\psi(t_p) \in \mathsf{MSO}^{t_p}$ is a time constraint formula whose syntax and semantics are given little later. A formula in MSO^{t_0} with first order free variables $t_0, t_1, \ldots t_k$ and second-order free variables $T_1, \ldots, T_m$ and which is relativized to the future of t_0 is denoted $\phi(\downarrow t_0, \ldots t_k, T_1, \ldots, T_m)$. (The $\downarrow$ is only to indicate the anchor variable. It has no other function.) The semantics of such formulae is as usual. Given ρ, positions $a_0, \ldots a_k$ in $dom(\rho)$, and sets of positions $A_1, \ldots, A_m$ with $A_i \subseteq dom(\rho)$,we define

$$\rho, (a_0, a_1, \ldots, a_k, A_1, \ldots, A_m) \models \phi(\downarrow t_0, t_1, \ldots t_k, T_1, \ldots, T_m)$$

inductively, as usual.
1. $(\rho, a_0, a_1, \ldots, a_k, A_1, \ldots, A_m) \models t_i{<}t_j$ iff $a_i{<}a_j$,
2. $(\rho, a_0, a_1, \ldots, a_k, A_1, \ldots, A_m) \models Q_a(t_i)$ iff $a{\in}\sigma(a_i)$,
3. $(\rho, a_0, a_1, \ldots, a_k, A_1, \ldots, A_m) \models T_j(t_i)$ iff $a_i{\in}A_j$,
4. $(\rho, a_0, a_1, \ldots, a_k, A_1, \ldots, A_m) \models \exists t_k\ t_0{<}t_k{\wedge}\phi(\downarrow t_0, \ldots t_k, T_1, \ldots, T_m)$ iff $(\rho, a_0, \ldots, a'_k, A_1, \ldots, A_m) \models \phi(\downarrow t_0, \ldots t_k, , T_1, \ldots, T_m)$ for some $a'_k \geq a_0$.

The **time constraint** $\psi(t_0)$ has the form $\mathcal{Q}_1 t_1 \mathcal{Q}_2 t_2 \ldots \mathcal{Q}_j t_j\ \phi(\downarrow t_0, t_1, \ldots t_j)$ where $\phi \in \mathsf{MSO}^{t_0}$ and $\mathbf{j < k}$, the parameter of logic QkMSO. Each quantifier $\mathcal{Q}_i t_i$ has the form $\overline{\exists} t_i \in t_0{+}I_i$ or $\overline{\forall} t_i \in t_0{+}I_i$ for a time interval I_i as in MTL formulae. $\mathcal{Q}_i$ is called a metric quantifier. The semantics of such a formula is as follows. $(\rho, a_0) \models \mathcal{Q}_1 t_1 \mathcal{Q}_2 t_2 \ldots \mathcal{Q}_j t_j\ \phi(\downarrow t_0, t_1, \ldots t_j)$ iff for $1 \leq i \leq j$, there exist/for all a_i such that $a_0 \leq a_i$ and $\tau_{a_i} \in \tau_{a_0} + I_i$, we have $(\rho, a_0, a_1 \ldots a_j) \models \phi(\downarrow t_0, t_1, \ldots t_j)$. Note that each time constraint formula has exactly one free variable. Variables $t_0, t_1, \ldots, t_j$ are called time constrained in $\psi(t_0)$.

▶ **Example 3.** Let $\rho{=}(\{a\}, 0)\ (\{b\}, 2.1)\ (\{a, b\}, 2.75)\ (\{b\}, 3.1)$ be a timed word. Consider the time constraint $\psi(x) = \overline{\exists} y \in x + (2, \infty) \overline{\exists} z \in x + (3, \infty)(Q_b(y) \wedge Q_b(z))$. It can be seen that $\rho, 1 {\models} Q_a(x) \wedge \psi(x)$.

Metric Depth. The *metric depth* of a formula φ denoted $(\mathsf{md}(\varphi))$ gives the nesting depth of time constraint constructs. It is defined inductively as follows: For atomic formulae φ, $\mathsf{md}(\varphi) = 0$. All the constructs of MSO^{t_i} do not increase md. For example, $\mathsf{md}[\varphi_1 \wedge \varphi_2] = max(\mathsf{md}[\varphi], \mathsf{md}[\varphi_2])$ and $\mathsf{md}[\exists t.\varphi(t)]$. However, md is incremented for each application of metric quantifier block. $\mathsf{md}[\mathcal{Q}_1 t_1 \mathcal{Q}_2 t_2 \ldots \mathcal{Q}_j t_j \phi] = \mathsf{md}[\phi] + 1$.

▶ **Example 4.** The sentence $\forall t_0\ \overline{\forall} t_1 \in t_0+(1,2)\ \{Q_a(t_1) \rightarrow (\overline{\exists} t_0 \in t_1+[1,1]\ Q_b(t_0))\}$ accepts all timed words such that for each a which is at distance (1,2) from some time stamp t, there is a b at distance 1 from it. This sentence has metric depth two with time constrained variables t_0, t_1.

Note that QkMSO is not closed under second order quantification: arbitrary use of second order quantification is not allowed, and its syntactic usage as explained above is restricted to prevent a second order free variable from occurring in the scope of the real-time constraint (similar to [19], [6] and [22]). For example, $\exists X.\exists t.[X(t) \wedge \overline{\exists} t' {\in} t{+}(1,2) Q_a(t')]$ is a well-formed QkMSO formula while, $\exists X.\exists t.\overline{\exists} t' {\in} t{+}(1,2)[Q_a(t') \wedge X(t)]$ is not, since X is freely used within the scope of the metric quantifier.

Special Cases of QkMSO. The case when $k = 2$ gives logic Q2MSO. The absence of second order variables and second order quantifiers gives logics QkFO and Q2FO. The formulae in example 4 is a Q2FO formulae. Note that our Q2FO is the pointwise counterpart of logic Q2MLO studied in [7] in the continuous semantics.

2.3 1-clock Alternating Timed Automata (1-ATA)

Let Σ be a finite alphabet and let $\Gamma = 2^{\Sigma} \backslash \emptyset$. A 1-ATA [17] [16] is a 5 tuple $\mathcal{A} = (\Gamma, S, s_0, F, \delta)$, where S is a finite set of locations, $s_0 \in S$ is the initial location and $F \subseteq S$ is the set of final locations. Let x denote the clock variable in the 1-ATA, and $x \in I$ denotes a clock constraint where I is an interval. Let X denote a finite set of clock constraints of the form $x \in I$. The transition function is defined as $\delta : S \times \Gamma \rightarrow \Phi(S \cup X)$ where $\Phi(S \cup X)$ is a set of formulae over $S \cup X$ defined by the grammar $\varphi ::= \top | \bot | \varphi_1 \wedge \varphi_2 | \varphi_1 \vee \varphi_2 | s | x \in I | x.\varphi$ where $s \in S$, and $x.\varphi$ is a binding construct resetting clock x to 0. A state of 1-ATA is defined as a pair of a location and a clock valuation. In other words, a state of a 1-ATA is an element of $S \times \mathbb{R}_{\geq 0}$, where S is a set of locations. A set of states $M \subseteq S \times \mathbb{R}_{\geq 0}$ and a clock valuation $\nu \in \mathbb{R}_{\geq 0}$ defines a boolean valuation for $\Phi(S \cup X)$ as follows:

$M \models_\nu s$ iff $(s, \nu) \in M$; $M \models_\nu x \in I$ iff $\nu \in I$; $M \models_\nu x.s$ iff $(s, 0) \in M$

The conjunctions, disjunctions, $\top$ and $\bot$ are are handled in the usual way. We say that M is a minimal model of $\varphi \in \Phi(S \cup X)$ with respect to ν iff $M \models_\nu \varphi$ and no proper subset M' of M is such that $M' \models_\nu \varphi$.

A configuration of a 1-ATA is a set of states. Given a configuration $\mathcal{C} = \{(s, \nu) \mid s \in S, \nu \in \mathbb{R}_{\geq 0}\}$, we denote by $\mathcal{C}{+}t$ the configuration $\{(s, \nu{+}t) \mid (s, \nu) \in \mathcal{C}\}$, obtained after a time elapse t, i.e., when t is added to all the valuations in $\mathcal{C}$. Let $\alpha \in \Gamma$. Given any configuration $\mathcal{C}_x = \{(s_j, \nu_j) | j \in J\}$, we say that any configuration $\mathcal{C}_y$ is a 'α-discrete successor' of $\mathcal{C}_x$ if and only if $\mathcal{C}_y$ can be constructed using $\mathcal{C}_x$ by choosing a minimal model M_j of $\delta(s_j, g)$ with respect to ν_j for every $j \in J$ followed by taking their union. That is, $\mathcal{C}_y = \bigcup_{j \in J} M_j$, where M_j is a minimal model of $\delta(s_j, g)$ with respect to ν_j. Given a timed word $\rho = (\alpha_0, 0)(\alpha_1, t_1) \dots (\alpha_m, t_m)$, a run associated with ρ over a given 1-ATA starts from the initial configuration $\mathcal{C}_0 = \{(s_0, 0)\}$ and has the form $\mathcal{C}_0 \xrightarrow{g_0} \mathcal{C}_1 \xrightarrow{t_1 - 0} \mathcal{C}_1 + t_1 \dots \mathcal{C}_{m-1} + (t_m - t_{m-1}) \xrightarrow{g_m} \mathcal{C}_m$ and proceeds with alternating time elapse transitions and discrete transitions reading a symbol from Σ. A discrete transition $\mathcal{C}_i + (t_i - t_{i-1}) \xrightarrow{\alpha_i} \mathcal{C}_{i+1}$ is included if and only if $\mathcal{C}_{i+1}$ is a α_i-discrete successor of $\mathcal{C}_i + (t_i - t_{i-1})$. Note that there could be more than one α_i-discrete successors and hence more than one run could be associated with a timed word. Note that if at any point in the run there is no discrete successor for a discrete transition, the automaton gets stuck and that run will not be associated with the word. A configuration $\mathcal{C}$ is accepting iff for all $(s, \nu) \in \mathcal{C}$, $s \in F$. Note that the empty configuration is also an accepting configuration.

The language accepted by a 1-ATA $\mathcal{A}$, denoted $L(\mathcal{A})$ is the set of all timed words ρ such that, there exists a run associated with ρ which ends at an accepting configuration.

▶ **Example 5.** Let $\Gamma=2^{\{a,b\}} \setminus \emptyset$. Consider the 1-ATA $\mathcal{A}=(\Gamma, \{t_0,t_1,t_2\}, t_0, \{t_0,t_2\}, \delta_{\mathcal{A}})$ with transitions $\delta_{\mathcal{A}}(t_0,\{b\})=t_0$, $\delta_{\mathcal{A}}(t_0,\{a\})=(t_0 \wedge x.t_1) \vee t_2$, $\delta_{\mathcal{A}}(t_1,\{a\})=(t_1 \wedge x{<}1) \vee (x{>}1)= \delta_{\mathcal{A}}(t_1,\{b\})$, and $\delta_{\mathcal{A}}(t_2,\{b\})=t_2, \delta_{\mathcal{A}}(t_2,\{a\})=\bot$, $\delta_{\mathcal{A}}(t,\{a,b\})=\bot$ for $t \in \{t_0,t_1,t_2\}$. The automaton accepts all strings where $\{a,b\}$ does not occur, and every non-last $\{a\}$ has no symbols at distance 1 from it, and has some symbol at distance > 1 from it.

1-ATA are closed under boolean operations using well-known constructions. One additional operation which is repeatedly used in this paper can be designated as $A_1[tr \wedge A_2]$. Here A_1, A_2 are 1-ATA over the same alphabet and $tr = \delta_1(s,a) = \phi$ is a transition of A_1. The aim is to conjunctively start the automaton A_2 on taking transition tr. Formally, transition tr is replaced by $tr' = \phi \wedge \delta_2(s^2_{init}, a)$ where s^2_{init} is the start state of A_2. Semantically, runs of A_2 must now commence with a time delay after tr. The delayed run defined below defines such an execution of A_2.

Time-Delayed Runs & Acceptance. Given a timed word $\rho = (\alpha_0,0)(\alpha_1,t_1)\dots(\alpha_m,t_m)$, a time delayed run associated with ρ over a given 1-ATA starts from the initial configuration $\mathcal{C}'_0 = \{(s_0,0)\}$ and has the form $\mathcal{C}'_0 \xrightarrow{t_1} \mathcal{C}'_0 + t_1 - 0 \xrightarrow{\alpha_1} \mathcal{C}'_1 \xrightarrow{t_2-t_1} \mathcal{C}'_1 + (t_2-t_1) \cdots \xrightarrow{\alpha_m} \mathcal{C}'_m$, where $\mathcal{C}'_{i-1} + (t_i - t_{i-1}) \xrightarrow{\alpha_i} \mathcal{C}'_i$ is included if and only if $\mathcal{C}'_i$ is one of the α_i-discrete successors of $\mathcal{C}'_{i-1} + (t_i - t_{i-1})$. Note that the only difference between a run and a time-delayed run is that the time delayed run starts with the time elapse between the first and the second symbol of the timed word and ignores the first symbol.

We say that any timed word ρ is accepted in time-delayed semantics by an ATA $\mathcal{A} = (\Gamma, S, s_0, F, \delta)$, if and only if, there exists a time-delayed run associated with ρ which ends with an accepting configuration. The set of all timed words accepted by time -delayed semantics of $\mathcal{A}$ is $L^{\mathsf{st}}(\mathcal{A})$.

▶ **Example 6.** Let $\Gamma = 2^{\{a,b\}} \setminus \emptyset$. Let α and α_b be any symbol in Γ and $\Gamma \setminus \{\{a,\}\}$,respectively. Consider the 1-ATA $\mathcal{A}=(\Gamma, \{s_0,s_1,s_2,s_3\}, s_0, \{s_2\}, \delta_{\mathcal{A}})$ with transitions $\delta_{\mathcal{A}}(s_0,\alpha)=s_1$; $\delta_{\mathcal{A}}(s_1,\alpha_b)=[s_2 \wedge x \in (1,2)] \vee [s_3 \wedge x \in (0,1)]$; $\delta_{\mathcal{A}}(s_1,\{a\})=s_3$; $\delta_{\mathcal{A}}(s_2,\alpha)=s_2$; $\delta_{\mathcal{A}}(s_3,\alpha)=s_3$. Consider a timed word $\rho = (\{a\},0)(\{b\},0.5)(\{b\},1.2)$. The run associated with ρ is : $\{(s_0,0)\} \xrightarrow{\{a\}} \{(s_1,0)\} \xrightarrow{0.5} \{(s_1,0.5)\} \xrightarrow{\{b\}} \{(s_3,0.5)\} \xrightarrow{0.7} \{(s_3,1.2)\} \xrightarrow{b} \{(s_3,1.2)\}$ Time delayed run associated with the same word is: $\{(s_0,0)\} \xrightarrow{0.5} \{(s_0,0.5)\} \xrightarrow{\{b\}} \{(s_1,0.5)\} \xrightarrow{0.7} \{(s_2,1.2)\} \xrightarrow{b} \{(s_2,1.2)\}$. Thus ρ is not accepted by the automaton in usual semantics, but accepted in time-delayed semantics.

We will define some terms which will be used in sections 4, 5. Consider a transition $\delta(s,a) = C_1 \vee \cdots \vee C_n$ in the 1-ATA. Each C_i is a conjunction of $x \in I$, locations p and $x.p$. We say that p is *free* in C_i if there is an occurrence of p in C_i and no occurrences of $x.p$ in C_i; if C_i has an $x.p$, then we say that p is *bound* in C_i. We say that p is bound in $\delta(s,a)$ if it is bound in some C_i.

Expressive Completeness and Equivalence. Let F_i be a logic or automaton class i.e. a collection of formulae or automata describing/accepting finite timed words. For each $\phi \in F_i$ let $L(F_i)$ denote the language of F_i. We define $F_1 \subseteq_e F_2$ if for each $\phi \in F_1$ there exists $\psi \in F_2$ such that $L(\phi) = L(\psi)$. Then, we say that F_2 is expressively complete for F_1. We also say that F_1 and F_2 are expressively equivalent, denoted $F_1 \equiv_e F_2$, iff $F_1 \subseteq_e F_2$ and $F_2 \subseteq_e F_1$.

3 A Normal Form for 1-ATA

In this section, we establish a normal form for 1-ATA, which plays a crucial role in the rest of the paper. Let $\mathcal{A} = (\Gamma, S, s_0, F, \delta)$ be a 1-ATA. $\mathcal{A}$ is said to be in normal form if and only if

- The set of locations S is partitioned into two sets S_r and S_{nr}. The initial state $s_0 \in S_r$.
- The locations of S are partitioned into $P_1, \ldots, P_k$ satisfying the following: Each P_i has a unique header location $s_i^r \in S_r$. Also, $P_i - \{s_i^r\} \subseteq S_{nr}$. Moreover, for any transition of $\mathcal{A}$ of the form $\delta(s, a) = C_1 \vee C_2 \ldots \vee C_k$ with $C_i = x \in I \wedge p_1 \wedge \ldots \wedge p_m \wedge x.q_i \wedge \ldots \wedge x.q_r$ we have (a) each $q_i \in S_r$, and (b) If $s \in P_i$ then each $p_j \in P_i - \{s_i^r\}$. [1]

We refer to each partition P_j as an *island* of locations. Each island has a unique header (obtained on reset) location s_i^r. All transitions into P_j occur only to this unique header location, and only with reset of clock x. Moreover, all non-reset transitions stay in the same island until a clock is reset, at which point, the control extends to the header location of same or another island (this behaviour can be seen on each path of the run tree).

Establishing the Normal Form. The main result of this section is that every 1-ATA $\mathcal{A}$ can be normalized, obtaining a language equivalent 1-ATA, $\mathsf{Norm}(\mathcal{A})$. The key idea behind this is to duplicate locations of $\mathcal{A}$ such that the conditions of normalization are satisfied. Let the set of locations of $\mathcal{A}$ be $S = \{s_1, \ldots, s_n\}$. For each location $s_i, 1 \leq i \leq n$, create a reset copy denoted s_i^r as well as n non-reset copies $s_i^{nr,j}, 1 \leq j \leq n$. The superscript r on a location represents that all incoming transitions to it are on a clock reset, while superscripts nr, j represent that all incoming transitions to that location are on non-reset and it belongs to island P_j. If s_0 is the initial location of $\mathcal{A}$, then the initial location of $\mathsf{Norm}(\mathcal{A})$ is s_0^r. The island P_i in $\mathsf{Norm}(\mathcal{A})$ consists of locations $s_i^r, s_h^{nr,i}$ for $1 \leq h \leq n$; entry into P_i happens through the header s_i^r. A transition $\delta(s_i, a) = \varphi$ of $\mathcal{A}$ is rewritten in $\mathsf{Norm}(\mathcal{A})$ by replacing all occurrences of locations $x.s_j$ with $x.s_j^r$ (leading into P_j), while occurrences of free locations s_h are replaced with $s_h^{nr,i}$. The final locations of $\mathsf{Norm}(\mathcal{A})$ are $s_i^r, s_i^{nr,j}$ for $1 \leq j \leq n$ whenever s_i is a final location in $\mathcal{A}$. The full version gives a formal proof for the following straightforward lemma. Thanks to lemma 7, we assume without loss of generality, in the rest of the paper, that 1-ATA are in normal form.

▶ **Lemma 7.** *Given a 1-ATA $\mathcal{A}$, one can construct a 1-ATA $\mathsf{Norm}(\mathcal{A})$ in normal form such that $L(\mathcal{A}) = L(\mathsf{Norm}(\mathcal{A}))$.*

▶ **Example 8.** The 1-ATA $\mathcal{B} = (\{a, b\}, \{s_0, s_1, s_2\}, s_0, \{s_1\}, \delta)$ with transitions $\delta(s_0, b) = x.s_2, \delta(s_0, a) = (s_0 \wedge x.s_1)$, $\delta(s_1, a) = (s_1 \wedge s_0) = \delta(s_1, b)$ and $\delta(s_2, b) = x.s_0, \delta(s_2, a) = (s_2 \wedge x.s_1)$ is not in normal form. Following the normalization technique, we obtain $\mathsf{Norm}(\mathcal{B})$ with locations $S = \{s_i^r, s_j^{nr,i} \mid 0 \leq i, j \leq 2\}$ and final locations $\{s_1^r, s_1^{nr,0}, s_1^{nr,1}, s_1^{nr,2}\}$. Figure 1(right) describes $\mathsf{Norm}(\mathcal{B})$ consisting of islands P_0, P_1 and P_2.

4 1-ATA-rfl and Logics

In this section, we show the first of our expressive equivalence results connecting logics RatMTL, QkMSO and a subclass of 1-ATA called 1-ATA with reset-free loops (1-ATA-rfl).We first introduce 1-ATA-rfl. A 1-ATA $\mathcal{A}$ (in normal form) is said to be a 1-ATA-rfl if it satisfies

---

[1] If the transitions of the 1-ATA are presented in CNF, the equivalent restriction is that each free location in the transition should be in $P_i - \{s_i^r\}$ while each bound location should be in S_r.

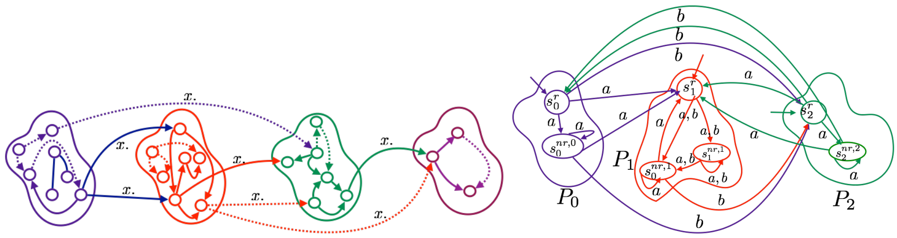

Figure 1 left: strict island hopping. right: $\mathsf{Norm}(\mathcal{B})$ corresponding to $\mathcal{B}$ in Example 8.

the following: There is a partial order $(S_r, \preceq)$ on the header locations (equivalently, islands P_i). Moreover, for any location $p \in P_i$ and a location q, if $x.q$ occurs in $\delta(p, a)$ for any a (hence $q = s^r_j$) then $s^r_j \prec s^r_i$. Thus, islands (which are only connected by reset transitions) form a DAG, and every reset transition goes to a lower level island. See Figure 1 (left), where we call this *strict island hopping*. The colored islands are all disjoint. On non-reset transitions, control stays in the same island; on resets, it may expand to another island. From this finite control, you cannot go back to the island where you started from. This provides a partial order between islands due to resets (the name rfl comes from here). Semantically, this means that on any branch of a run tree, a reset transition occurs at most once.

▶ **Example 9.** The 1-ATA with locations s, p, q and transitions $\delta(s, \alpha)=(x.p \wedge x \leq 1) \vee (q \wedge x=2)$, $\delta(p, \alpha)=x.q \wedge p$ and $\delta(q, a)=s \wedge (0<x<1)$ is not 1-ATA-rfl, since q is bound in $\delta(p, \alpha)$ and starting from q, we can reach $x.p$ via s.

4.1 Useful Lemmas

In this section, we introduce some notations and prove lemmas (Lemma 10, Lemma 11) which will be used several times in the paper. Let $\rho = (a_0, 0)(a_1, \tau_1)(a_2, \tau_2) \dots (a_m, \tau_m)$ be a timed word and let $i \in dom(\rho)$. Let $\mathsf{rel}(\rho, i)$ denote the timed word obtained from ρ by relativizing at position i; $\mathsf{rel}(\rho, i) = (a_i, 0)(a_{i+1}, \tau_{i+1} - \tau_i)(a_{i+2}, \tau_{i+2} - \tau_i) \dots (a_m, \tau_m - \tau_i)$.

Region Words. Let c_{max} be any non-negative integer. Let $reg_{c_{max}} = \{0, (0, 1), \dots, c_{max}, (c_{max}, \infty)\}$ be the set of regions. Given a finite alphabet Σ, with $\Gamma = 2^{\Sigma} \backslash \emptyset$, a *region word* is a word over the alphabet $\Gamma \times reg_{c_{max}}$ called the *interval* alphabet. A region word $w = (a_1, I_1)\ (a_2, I_2)\ \dots (a_m, I_m)$ is *good* iff $I_j \leq I_k \Leftrightarrow j < k$. Here, $I_j \leq I_k$ represents that either $I_j = I_k$ or the upper bound of I_j is at most the lower bound of I_k. The timed word $\rho = (a_0, 0)(a_1, \tau_1)(a_2, \tau_2) \dots (a_m, \tau_m)$ is consistent with a region word $(a_1, I_1)(a_2, I_2) \dots (a_m, I_m)$ iff $\tau_j \in I_j$ for all $0 < j \leq m$. Note that for technical reasons, which will be clearer later, the region abstraction of the word defined here is by ignoring the first point. The set of timed words ρ consistent with a good region word w is denoted $\mathcal{T}_w$. Likewise, given a timed word ρ, $reg(\rho)$ represents the good region word w such that $\rho \in \mathcal{T}_w$.

▶ **Lemma 10** (Untiming P to $\mathcal{A}(P)$)**.** *Let P be a 1-ATA over Γ having no resets. We can construct an alternating finite automaton $\mathcal{A}(P)$ over the interval alphabet $\Gamma \times reg_{c_{max}}$ such that for any good region word $w=(a_1, I_1) \dots, (a_n, I_n)$, $w \in L(\mathcal{A}(P)) \Rightarrow \forall \rho \in \mathcal{T}_w, \rho \in L^{\mathsf{st}}(P)$. Conversely, $\rho \in L^{\mathsf{st}}(P) \Rightarrow reg(\rho) \in L(\mathcal{A}(P))$. Hence, $L^{\mathsf{st}}(P)=\{\rho \mid w \in L(A(P) \wedge \rho \in \mathcal{T}_w\}$.*

Proof. Given a reset-free 1-ATA $P = (\Gamma, S, s_0, F, \delta)$ we construct an alternating finite automaton (AFA) $\mathcal{A}(P) = (\Gamma \times reg_{c_{max}}, S, s_0, F, \delta')$ where c_{max} is the maximum constant used in the clock constraints of P and δ' is defined using δ as follows. A clock constraint

$x \in I$ is called a *region constraint* if $I \in reg_{c_{max}}$. Without loss of generality, we assume that all the transitions of P have the form $C_1 \vee \cdots \vee C_m$, where each clause C_i has exactly one region constraint.

Construction of δ'. Given a transition $\delta(s,a) = C_1 \vee \cdots \vee C_n$ of P, let $C_{I_1}, \ldots, C_{I_k} \in \{C_1, \ldots, C_n\}$ be clauses containing the region constraint $x \in I$, $I \in reg_{c_{max}}$. We construct $\delta'(s,(a,I))$ as $C'_{I_1} \vee \cdots \vee C'_{I_k}$ where C'_{I_j} is obtained by removing the conjunct $x \in I$ from C_{I_j}. If there is no clause containing $x \in I$ in $\delta(s,a)$, then $\delta'(s,(a,I)) = \bot$.

1. For any timed word ρ, $\rho \in L^{\mathsf{st}}(P) \Rightarrow reg(\rho) \in L(\mathcal{A}(P))$: Let $\rho = (a_1, 0)\ldots(a_m, \tau_m)$. The time-delayed accepting run from the initial configuration $\mathcal{C}_0 = \{(s_0, 0)\}$ to an accepting configuration $\mathcal{C}_m$ in P is as follows: $\mathcal{C}_0 \xrightarrow{\tau_1} \mathcal{C}_0 + \tau_1 \xrightarrow{a_1} \mathcal{C}_1 \xrightarrow{\tau_2-\tau_1} \cdots \xrightarrow{\tau_m-\tau_{m-1}} \mathcal{C}_{m-1} + (\tau_m - \tau_{m-1}) \xrightarrow{a_m} \mathcal{C}_m$. As P is a reset-free 1-ATA, the valuation $\nu(x)$ at any point is equal to the total time elapse till that point. Hence for all positions $j \in dom(\rho)$, $I \in reg_{c_{max}}$, $\tau_j \in I$ iff $\nu(x) \in I$. By construction of $\mathcal{A}(P)$, for each transition $\delta(s, a_j) = (x \in I_j \wedge \psi) \vee C$ of P we have a transition $\delta'(s,(a_j, I_j)) = \psi$ (wlg we assume that C has no occurrence of $x \in I_j$). The accepting run of P translates into the run $\mathcal{D}_0 \xrightarrow{(a_1,I_1)} \mathcal{D}_1 \xrightarrow{(a_2,I_2)} \mathcal{D}_2 \cdots \xrightarrow{(a_m,I_m)} \mathcal{D}_m$ in $\mathcal{A}(P)$, where $\mathcal{D}_0 = \{s \mid (s,0) \in \mathcal{C}_0\}$, and $\mathcal{D}_j = \{s \mid (s,t) \in \mathcal{C}_j, t \in I_j\}$, $j \geq 1$. Since all locations in $\mathcal{C}_m$ are accepting, $\mathcal{D}_m$ is an accepting configuration in $\mathcal{A}(P)$ accepting $(a_1, I_1)\ldots(a_m, I_m)$.
2. For any good word w, $w \in L(\mathcal{A}(P)) \Rightarrow \rho \in L^{\mathsf{st}}(P)$ for all $\rho \in \mathcal{T}_w$. The argument is similar to the previous bullet. Full proof can be found in the full version. ◀

▶ **Lemma 11** ($\mathcal{A}(P)$ to RatMTL)**.** *Let $\mathcal{A}(P)$ be an AFA over the interval alphabet $\Gamma \times reg_{c_{max}}$ constructed from a reset-free 1-ATA P as in lemma 10. We can construct a* RatMTL *formula φ^{st} such that for any good word w, $w \in L(\mathcal{A}(P)) \Rightarrow \rho \in L(\varphi^{\mathsf{st}})$ for all $\rho \in \mathcal{T}_w$, and, for any timed word ρ, $\rho \in L(\varphi^{\mathsf{st}}) \Rightarrow reg(\rho) \in L(\mathcal{A}(P))$. $L(\varphi^{\mathsf{st}}) = \{\mathcal{T}_w \mid w \in L(\mathcal{A}(P))\}$. Hence, by lemma 10, $L(\varphi^{\mathsf{st}}) = L^{\mathsf{st}}(P)$.*

Proof. Let $Det[\mathcal{A}(P)]$ be the deterministic automaton which is language equivalent to $\mathcal{A}(P)$. Let δ_D be the transition function of $Det[\mathcal{A}(P)]$ obtained from δ' (see lemma 10) and let $\hat{\delta}_D$ be its extension to words. Let s_0 be the initial location of $Det[\mathcal{A}(P)]$ and let F be the set of its final locations. For any pair of locations p, q of $Det[\mathcal{A}(P)]$ and $I_i \in reg_{c_{max}}$, we construct a regular expression $\mathsf{re}(p,q,I_i)$ equivalent to the language $\{w \in (\Gamma \times \{I_i\})^+ \mid \hat{\delta}_D(p,w) = q\}$.

Construction of $\mathsf{re}(p,q,I_i)$. Let $Det[\mathcal{A}(P)[p,q]]$ be the DFA obtained from $Det[\mathcal{A}(P)]$ by setting the initial location to p and set of final locations to $\{q\}$. Let $\mathcal{A}(I_i^*)$ denote the DFA accepting all words $(\Gamma \times \{I_i\})^*$. Let $Det[\mathcal{A}_i] = Det[\mathcal{A}(P)[p,q]] \cap \mathcal{A}(I_i^*)$. Then $L(\mathsf{re}(p,q,I_i)) = L(Det[\mathcal{A}_i])$.

Obtaining RatMTL formula. Consider a sequence of locations $\mathsf{sseq}{=}q_0, q_1, q_2 \ldots q_k$ with $k{=}2*c_{max}+2$, $q_0{=}s_0$ and $q_k{\in}F$. Let $\mathsf{re}{=}\mathsf{re}(q_0,q_1,[0,0]).\mathsf{re}(q_1,q_2,(0,1)). \quad \cdots .\mathsf{re}(q_{k-1},q_k,(c_{max},\infty))$ where $\mathsf{re}(q_{2i-1},q_{2i},(i-1,i)) \subseteq (\Gamma \times (i-1,i))^*$ and $\mathsf{re}(q_{2i},q_{2i+1},[i,i]) \subseteq (\Gamma \times [i,i])^*$, and $L(\mathsf{re}) \subseteq L(Det\mathcal{A}(P))$. Define the RatMTL formula $\phi(\mathsf{sseq}) = \mathsf{Rat}_{[0,0]}\mathsf{re}(q_0,q_1,[0,0]) \wedge \mathsf{Rat}_{(0,1)}\mathsf{re}(q_1,q_2,(0,1)) \wedge \cdots \wedge \mathsf{Rat}_{(c_{max},\infty)}[\mathsf{re}(q_{k-1},q_k,(c_{max},\infty))]$. Then $L(\phi(\mathsf{sseq})){=}\{\mathcal{T}_w \mid w \in L(\mathsf{re}(q_0,q_1,[0,0]).\mathsf{re}(q_1,q_2,(0,1))\cdots\mathsf{re}(q_{k-1},q_k,(c_{max},\infty))\}$. Let $\varphi^{\mathsf{st}} = \vee_{\mathsf{sseq}}\, \phi(\mathsf{sseq})$[2]. Then clearly, $L(\varphi^{\mathsf{st}}) = \{\mathcal{T}_w \mid w \in L(\mathcal{A}(P))\}$. By lemma 10, we obtain $L(\varphi^{\mathsf{st}}) = L^{\mathsf{st}}(P)$. ◀

[2] The superscript st in φ^{st} represents that it is a strict future formulae. The truth of the formula at any point i is only dependent on points $j > i$

Remark. $\rho, i \models \varphi^{\mathsf{st}}$ iff $\mathsf{rel}(\rho, i) \in L^{\mathsf{st}}(P)$. $\rho, i \models \varphi^{\mathsf{st}}$ iff $\mathsf{rel}(\rho, i) \in L(\varphi^{\mathsf{st}})$ since the truth of φ^{st} only depends on the truth value of the propositions at the points strictly greater than i and the relative time difference with respect to i. The proof of Lemma 12 is in the full version.

▶ **Lemma 12.** *Let $P = (\Gamma, S, s_0, F, \delta)$ be any reset free 1-ATA and $\mathcal{A}(P)$ be an AFA as constructed in lemma 10.Let φ^{st} be the formula constructed from $\mathcal{A}(P)$ as in lemma 11.We can construct a formula $\varphi \in$ RatMTL such that $L(\varphi) = L(P)$.*

4.2 1-ATA-rfl meets RatMTL

▶ **Theorem 13.** *1-ATA-*rfl *are expressively equivalent to* RatMTL.

Proof.

1. 1-ATA-rfl $\subseteq_e$ RatMTL: Let $\mathcal{A}$ be a 1-ATA-rfl in normal form. For each location $s_i^r \in S_r$ (which is the header of partition P_i) let $\mathcal{A}[s_i^r]$ denote the same automaton as $\mathcal{A}$ except that the initial location is changed to s_i^r, the header location of P_i. We can also delete all islands higher than P_i as their locations are not reachable. For each such automaton $\mathcal{A}[s_i^r]$, we construct a pair of RatMTL formulae $\mathsf{mtl}(\mathcal{A}[s_i^r])$ and $\mathsf{mtl}^{\mathsf{st}}(\mathcal{A}[s_i^r])$ such that $L(\mathsf{mtl}(\mathcal{A}[s_i^r])) = L(\mathcal{A}[s_i^r])$ and $L(\mathsf{mtl}^{\mathsf{st}}(\mathcal{A}[s_i^r])) = L^{\mathsf{st}}(\mathcal{A}[s_i^r])$. Note that $(S_r, \preceq)$ is a partial order. The construction and proof of equivalence are by complete induction on the level of the header location s_i^r of island P_i in the partial order. All $x.s_j^r$ occurring in any transition of $\mathcal{A}[s_i^r]$ are of lower level in the partial order $(S_r, \preceq)$. Hence, by induction hypothesis, there is a RatMTL formula $\psi_j^{\mathsf{st}} = \mathsf{mtl}^{\mathsf{st}}(\mathcal{A}[s_j^r])$ such that $L^{\mathsf{st}}(\mathcal{A}[s_j^r]) = L(\psi_j^{\mathsf{st}})$. The behaviour of all the lower level s_j^r is independent of the label of the transition that calls them. In other words, the automata $\mathcal{A}_i[s_j^r]$ which is called at position i does not read the symbol at point i. Thus any lower level automaton starting from some s_j^r called at a point i restricts the behaviour of the points strictly in the future of i, fixing the anchor (the point with respect to which the time differences are checked) at i. Let w_j be a fresh witness variable for each $x.s_j^r$ above, which also corresponds to RatMTL formula ψ_j^{st}. Let the set of such witness variables be $\{\mathsf{w}_1, \ldots, \mathsf{w}_k\}$. We construct a modified automaton $\mathcal{A}^{\mathsf{w}}[s_i^r]$ with transition function δ' and set of locations P_i as follows. Its alphabet is $\Gamma \times \{0,1\}^k$ with the jth component giving the truth value of witness w_j. Let $\delta'(s, a, \mathsf{w}_1, \ldots, \mathsf{w}_k) = \delta(s, a)[\mathsf{w}_j / x.s_j^r]$, i.e., each occurrence of $x.s_j^r$ is replaced by the truth value of w_j for $1 \leq j \leq k$.Note that $\mathcal{A}^{\mathsf{w}}[s_i^r]$ is a reset-free 1-ATA. By lemmas 10, 11, we get a RatMTL formula $\phi^{\mathsf{st},\mathsf{w}}$ such that $L^{\mathsf{st}}(\mathcal{A}^{\mathsf{w}}[s_i^r]) = L(\phi^{\mathsf{st},\mathsf{w}})$ and using lemma 12 we get a language equivalent formula ϕ^{w} over the variables $\Sigma \cup \{\mathsf{w}_1, \ldots, \mathsf{w}_k\}$. Now we substitute each w_j by ψ_j^{st} (and hence $\neg \mathsf{w}_j$ by $\neg \psi_j^{\mathsf{st}}$) in ϕ^{w} and $\phi^{\mathsf{st},\mathsf{w}}$ to obtain the required formulae $\mathsf{mtl}(\mathcal{A}[s_i^r])$ and $\mathsf{mtl}^{\mathsf{st}}(\mathcal{A}[s_i^r])$, respectively. It is clear from the substitution that $L^{\mathsf{st}}(\mathcal{A}[s_i^r]) = L(\mathsf{mtl}^{\mathsf{st}}(\mathcal{A}[s_i^r]))$ and $L(\mathcal{A}[s_i^r]) = L(\mathsf{mtl}(\mathcal{A}[s_i^r]))$. An example illustrating this construction is in the full version
2. RatMTL $\subseteq_e$ 1-ATA-rfl: Consider a formula $\psi_1 = \mathsf{Rat}_I(\mathsf{re}_0)$ with $I = [l, u)$. The case of other intervals are handled similarly. For the base case, ψ_1 has modal depth 1 and has a single modality. As the formula is of modal depth 1, re_0 is an atomic regular expression over alphabet 2^Σ. Let $D = (\Gamma, Q, q_0, Q_f, \delta')$ be a DFA such that $L(D) = L(\mathsf{re}_0)$, with $\Gamma = 2^\Sigma \backslash \emptyset$. From D, we construct the 1-ATA $\mathcal{A}$=$(\Gamma, Q \cup \{q_{init}, q_{\mathsf{timecheck}}, q_f\}, q_{init}, \{q_f\}, \delta)$ where $q_{init}, q_{\mathsf{timecheck}}, q_f$ are disjoint from Q. The transitions δ are as follows. Assume $l > 0$.
   - $\delta(q_{init}, a) = x.q_{\mathsf{timecheck}}, a \in \Gamma$,
   - $\delta(q_{\mathsf{timecheck}}, a) = [(x \geq l \wedge \delta'(q_0, a) \vee (q_{\mathsf{timecheck}})] \vee [x > u \wedge q_f]$ where the latter disjunct is added only when $q_0 \in Q_f$,
   - $\delta(q, a) = (x \in [l, u)) \wedge \delta'(q, a))$, for all $q \in Q \setminus Q_f$,

- $\delta(q, a) = (x \in [l, u) \wedge \delta'(q, a)) \vee (x > u \wedge q_f)$, for all $q \in Q_f$, $\delta(q_f, a) = q_f$.

It is easy to see that $\mathcal{A}$ has the reset-free loop condition since q_f is the only location entered on resets, and control stays in q_f once it enters q_f. The correctness of $\mathcal{A}$ is easy to establish, the location $q_{\mathsf{timecheck}}$ is entered on the first symbol, resetting the clock; control stays in $q_{\mathsf{timecheck}}$ as long as $x < l$, and when $x \geq l$, the DFA is started. As long as $x \in [l, u)$, we simulate the DFA. If $x > u$ and we are in a final location of the DFA, the control switches to the final location q_f of $\mathcal{A}$. If q_0 is itself a final location of the DFA, then from $q_{\mathsf{timecheck}}$, we enter q_f when $x > u$. It is clear that $\mathcal{A}$ indeed checks that re_0 is true in the interval $[l, u)$. If $l = 0$, then the interval on which re_0 should hold good is $[0, u)$. In this case, if q_0 is non-final, we have the transition $\delta(q_{init}, a) = x.\delta'(q_0, a), a \in \Gamma$ (since our timed words start at time stamp 0, the first symbol is read at time 0, so $x.\delta'(q_0, a)$ preserves the value of x after the transition $\delta'(q_0, a)$). The location $q_{\mathsf{timecheck}}$ is not used then. The case when ψ_1 has modal depth 1 but has more than one Rat modality is dealt as follows. Firstly, if $\psi_1 = \neg\mathsf{Rat}_I(\mathsf{re}_0)$, then the result follows since 1-ATA-rfl are closed under complementation (the fact that the resets are loop-free on a run does not change when one complements). For the case when we have a conjunction $\psi_1 \wedge \psi_2$ of formulae, having 1-ATA-rfl $\mathcal{A}_1 = (\Gamma, Q_1, q_1, F_1, \delta_1)$ and $\mathcal{A}_2 = (\Gamma, Q_2, q_2, F_2, \delta_2)$ such that $L(\mathcal{A}_1) = L(\psi_1)$ and $L(\mathcal{A}_2) = L(\psi_2)$, we construct $\mathcal{A} = (\Gamma, Q_1 \cup Q_2 \cup \{q_{init}\}, q_{init}, F, \delta)$ such that $\delta(q_{init}, a) = x.\delta_1(q_1, a) \wedge x.\delta_2(q_2, a)$. Clearly, $\mathcal{A}$ is a 1-ATA-rfl since $\mathcal{A}_1, \mathcal{A}_2$ are. It is easy to see that $L(\mathcal{A}) = L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$. The case when $\psi = \psi_1 \vee \psi_2$ follows from the fact that we handle negation and conjunction. The case of formulae of higher modal depth $\psi_{k+1} = \mathsf{Rat}_I(\mathsf{re}_k)$ is handled by substituting all lower depth formulae in re_k with witness variables, and using the inductive hypothesis that there exist 1-ATA-rfl equivalent to these. The main argument is then to show that on plugging-in these automata corresponding to the witnesses, we obtain a 1-ATA-rfl equivalent to ψ_{k+1}. Details in the full version. ◀

4.3 RatMTL meets QkMSO

▶ **Theorem 14.** QkMSO *is expressively equivalent to* RatMTL.

Proof.

1. QkMSO $\subseteq_e$ RatMTL: Proof is by Induction on the metric depth of the formula. For the base case,consider a formula $\psi(t_0) = \mathcal{Q}_1 t_1 \ldots \mathcal{Q}_{k-1} t_{k-1} \varphi(\downarrow t_0, t_1, \ldots, t_{k-1})$ of metric depth one.Let c_{max} be the maximal constant used in the metric quantifiers $\mathcal{Q}_i$. Let $R_j(t)$ for j in reg=$\{0, (0,1)$,1,..., c_{max}, $(c_{max}, \infty)\}$ be fresh monadic predicates. We modify $\psi(t_0)$ to obtain an untimed MSO formula $\psi_{rg}(t_0)$ over the alphabet $2^{\Sigma} \times \{0,1\}^{|reg|} \times \{0,1\}$ as follows. Define $CON(I_i, t_i) = \vee\{R_j(t_i) \mid j \subseteq I_i\}$. We replace every quantifier $\overline{\exists} t_i {\in} t_0 + I_i\ \phi$ by $\exists t_i (t_0 {\leq} t_i) {\wedge} CON(I_i, t_i) {\wedge} \phi$. Every quantifier $\forall t_i {\in} t_0 + I_i\ \phi$ is replaced by $\forall t_i.(t_0 {\leq} t_i {\wedge} CON(I_i, t_i) {\rightarrow} \phi)$. To the resulting MSO formula we add a conjunct WELLREGION that states that (a) exactly one $R_j(t)$ holds at any t, and (b) $\forall t, t'$. $[t{<}t' \wedge R_j(t) \wedge R_{j'}(t')] \rightarrow j \leq j'$ (asserting region order). Note that these are natural properties of region abstraction of time. This gives us the formula $\psi_{rg}(t_0)$. It has predicates $R_j(t)$ for $j \in reg$ and free variable t_0. Being an MSO formula, we can construct a DFA $\mathcal{A}(\psi_{rg}(0))$ for it over the alphabet $2^{\Sigma} \times \{0,1\}^{|reg|}$. Note that we have substituted 0 for t_0. This is isomorphic to automaton over the alphabet $2^{\Sigma} \times reg$. From the construction, it is clear that $\rho \models \psi(0)$ iff $reg(\rho) \in L(\mathcal{A}(\psi_{rg}(0)))$. By Lemma 12, we then obtain an equivalent RatMTL formula ζ. It is easy to see that $L(\psi(0)) = L(\zeta)$. Because $\psi(0)$ and ζ are purely future time formulae, this gives $\rho, i \models \psi(t_0)$ iff $\rho, i \models \zeta$. For the induction step, consider a metric depth $n + 1$ formula $\psi(t_0)$. We can replace every time constraint sub-formula

$\psi_i(t_k)$ occurring in it by a witness monadic predicate $w_i(t_k)$. This gives a metric depth 1 formula and we can obtain a RatMTL formula, say ζ, over variables $\Sigma \cup \{w_i\}$ exactly as in the base step. Notice that each $\psi_i(t_k)$ was a formula of modal depth n or less. Hence by induction hypothesis we have an equivalent RatMTL formula ζ_i. Substituting ζ_i for w_i in ζ gives us a formula language equivalent to $\psi(t_0)$.

2. RatMTL $\subseteq_e$ QkMSO : Let $\varphi \in$ RatMTL. The proof is by induction on the modal depth of φ. For the base case, let $\varphi = \mathsf{Rat}_I(\mathsf{re})$ where re is a regular expression over propositions. Let $\zeta(x,y)$ be an MSO formula with the property that $\sigma, i, j \models \zeta(x,y)$ iff $\sigma[x:y] \in L(re)$, where $\sigma[x:y]$ denotes the sub-string $\sigma(x+1)\ldots\sigma(y)$. Since MSO has exactly the expressive power of regular languages, such a formula can always be constructed. Consider $\psi(t_0)$: $\overline{\exists} t_{first} {\in} t_0{+}I.\ \overline{\exists} t_{last} {\in} t_0{+}I. \overline{\forall} t' {\in} t_0{+}I.[(t'{=}t_{first} \vee t'{=}t_{last} \vee t_{first}{<}t'{<}t_{last}) \wedge \zeta(t_{first}, t_{last})]$ Then, it is clear that $\rho, i \models \varphi$ iff $\rho, i \models \psi(t_0)$. Note that the time constraint formula $\psi(t_0)$ is actually a formula of QkMSO with $k = 4$ using time constrained variables $t_0, t_{first}, t_{last}, t'$ only. Atomic and boolean constructs can be straightforwardly translated. Now let $\varphi = \mathsf{Rat}_I(\mathsf{re})$ where re is over a set of subformulae S. For each $\zeta_i \in S$, substitute it by a witness proposition w_i to get a formula φ_{flat}. This is a modal depth 1 formula and we can construct a language equivalent formula of QkMSO, say $\Xi(t_0)$ over alphabet $\Sigma \cup \{w_i\}$. By induction hypothesis, for each ζ_i there exists a language equivalent time constrained QkMSO formula $\kappa_i(t_0)$. Now substitute $\kappa_i(t_j)$ for each occurrence of $w_i(t_j)$ in $\Xi(t_0)$ to get a formula $\psi(t_0)$. Then $\psi(t_0)$ is language equivalent to φ. Note that $\psi(t_0) {\in}$Q4MSO as it only uses time constrained variables $t_0, t_{first}, t_{last}, t'$ which are inductively reused. ◀

5 C⊕D-1-ATA-rfl and Logics

In this section, we show our second main result connecting logics FRatMTL, Q2MSO and a subclass of 1-ATA called *conjunctive-disjunctive* (abbreviated C⊕D) 1-ATA with reset-free loops. Let $\mathcal{A} = (\Gamma, Q, q_0, F, \delta)$ be a 1-ATA. Let $Q_x = \{x.q \mid q \in Q\}$ and let $\mathcal{B}(Q_x) ::= \mathsf{true}|\mathsf{false}|\alpha|\alpha \wedge \alpha|\alpha \vee \alpha$, where $\alpha \in Q_x$. $\mathcal{A}$ is said to be a C⊕D 1-ATA if

1. Q is partitioned into $Q_\wedge$ and $Q_\vee$,
2. Let $q \in Q_\wedge$. Transitions $\delta(q,a)$ can be written as $D_1 \wedge D_2 \wedge \ldots \wedge D_m$, where any D_i has one the following forms. (i) $D_i = q' \vee \mathcal{B}(Q_x)$ where $q' \in Q_\wedge$, (ii) $D_i = x \notin I \vee \mathcal{B}(Q_x)$.[3] Each D_i has at most one free location from $Q_\wedge$, or at most one clock constraint $x \notin I$.
3. Let $q \in Q_\vee$. Transitions $\delta(q,a)$ can be written as $C_1 \vee C_2 \vee \ldots \vee C_m$ where any C_i has one of the following forms. (i) $C_i = q' \wedge \mathcal{B}(Q_x)$, where $q' \in Q_\vee$, (ii) $C_i = x \in I \wedge \mathcal{B}(Q_x)$. Thus, each C_i has at most one free location from $Q_\vee$, or at most one clock constraint $x \in I$.

In other words, all the non-reset transitions coming out of $q \in Q_\wedge$ will go to all the locations in $Q' \subseteq Q_\wedge$ conjunctively asserting some time constraint. Similarly, all the non-reset transitions coming out of $q \in Q_\vee$ will non-deterministically go to one of the locations in $Q_\vee$, without checking any time constraint.

Remark. If any C⊕D-1-ATA A is normalized to Norm(A), then any island of Norm(A) is exclusively made of locations from $Q_\wedge$ or $Q_\vee$ (not both). Note that if we delete all the reset transitions from any island of Norm(A), all the transitions within the island will be either conjunctive or disjunctive. Thus, the name C⊕D is based on the fact that each island of Norm$(\mathcal{A})$ is either conjunctive or disjunctive. A 1-ATA which has both reset-free loops and

[3] Note that $x \notin I$ can be re-written as $x \in [0,l\rangle \vee x \in \langle u, \infty)$, where $[0,l\rangle \cup \langle u,\infty)$ is the complement of the interval I with lower and upper bounds l, u respectively. We restrict the specification of time intervals in this form to make sure that, using conjunctions and non-punctual intervals, one cannot express punctual intervals. This is used to define non-punctual C⊕D-1-ATA.

conjunctive-disjunctiveness is denoted C⊕D-1-ATA-rfl. If all the intervals in a C⊕D-1-ATA are non-punctual, then it is referred to as a np-C⊕D-1-ATA.

▶ **Example 15.** We illustrate examples of 1-ATA violating the C⊕D condition.

(a) For $a \in \Sigma$, let S_a and $S_{\neg a}$ denote any set containing a and not containing a, respectively. Consider the automaton $\mathcal{B}$ with transitions $\delta(s_0, S_a){=}s_0 \vee x > 1$, $\delta(s_0, S_{\neg a}){=}s_0 \wedge x \notin (1, \infty)$, where s_0 is the only location, which is non-final. The only way to accept a word is by reaching an empty configuration. The C⊕D condition is violated due to the combination of having a free location and a clock constraint simultaneously in a clause irrespective of $s_0 \in Q_\vee$ or $s_0 \in Q_\wedge$. This accepts the set of all words where the first symbol in $(1, \infty)$ has an a.

(b) Let S_a, $S_{\neg a}$ be as above. The automaton $\mathcal{B}$ with $\delta(s_0, S_a){=}s_0 \vee s_1$, $\delta(s_2, \Gamma) = x \leq 1$, $\delta(s_0, S_{\neg a}){=}s_0 \wedge s_2$, $\delta(s_1, \Gamma){=}x > 1$ with s_0 being initial and none of the locations being final satisfies rfl but violates C⊕D. The C⊕D condition is violated since a clause contains more than one free location irrespective of $s_0, s_1, s_2 \in Q_\vee$ or $s_0, s_1, s_2 \in Q_\wedge$. This accepts the language of all words where the last symbol in $(0, 1)$ has an a.

(c) The automaton $\mathcal{B}$ with $\delta(s_0, \Gamma){=}s_0{\vee}s_1$, $\delta(s_1, S_a) = s_2{\wedge}s_3$, $\delta(s_1, S_{\neg a}) = \bot$, $\delta(s_2, \Gamma){=}(x \leq 1)$, $\delta(s_3, \Gamma){=}s_4$, $\delta(s_4, \Gamma){=}x > 1$ with s_0 being initial and s_1 being final satisfies rfl but violates C⊕D. The C⊕D condition is violated since the automata switches between conjunctive and disjunctive locations without any reset. Note that $s_0{\in}Q_\vee$ while $s_1{\in}Q_\wedge$. The language accepted is all words where the second last symbol in $(0, 1)$ has an a.

5.1 C⊕D-1-ATA-rfl meets FRatMTL

▶ **Theorem 16.** C⊕D-*1-ATA*-rfl *are expressively equivalent to* FRatMTL*.*

Proof. We only detail the containment C⊕D-1-ATA-rfl $\subseteq_e$ FRatMTL; the converse direction is almost identical to the proof of RatMTL $\subseteq_e$1-ATA−rfl and is provided in the full version.

1. C⊕D-1-ATA-rfl $\subseteq_e$ FRatMTL : The first thing is to convert C⊕D 1-ATA with no resets to FRat formula of modal depth 1 as in Lemma 17.

   ▶ **Lemma 17.** *Given a* C⊕D *1-ATA* $\mathcal{A}$ *over* Σ *with no resets, we can construct a* FRat *formula* φ *such that for any timed word* $\rho = (a_1, \tau_1) \ldots (a_m, \tau_m)$*,* $\rho, i \models \varphi^{\mathsf{st}}$ *iff* $\mathcal{A}$ *accepts* $\mathsf{rel}(\rho, i)$ *in time-delayed semantics.* $L(\varphi^{\mathsf{st}}) = L^{\mathsf{st}}(\mathcal{A})$*.*

   Assuming $q_0 \in Q_\vee$, the key idea is to check how an accepting configuration is reached. The reset-freeness ensures that any transition $\delta(q, a) = C_1 \vee \cdots \vee C_m$ is such that C_i is either a location or a clock constraint $x \in I$. Assume time-delayed acceptance happens through an empty configuration via a clock constraint $x \in I_a$, from some location q on an a, and q is reachable from q_0. Let re_{I_a} be the regular expression whose language is the set of all such words reaching some q, from where acceptance happens via interval I_a on an a. The formula $\mathsf{FRat}_{I_a, \mathsf{re}_{I_a}} a$ sums up all such words. Disjuncting over all possible intervals and symbols, we have the result. The second case is when a final state q_f is reached from some q' reachable from q_0. If $\mathsf{re}_{q_f, a}$ is the regular expression whose language is all words reaching such a q', the formula $\mathsf{FRat}_{[0,\infty)}, \mathsf{re}_{q_f, a}(a \wedge \Box\bot)$ sums up all words accepted via q', a, q_f.The $\Box\bot$ ensures that no further symbols are read, and can be written as $\neg\mathsf{FRat}_{[0,\infty), \Sigma^*}\top$. Disjuncting over all possible final states q_f and $a \in \Sigma$ gives us the formula. The case when $q_0 \in Q_\wedge$ is handled by negating the automaton, obtaining $q_0 \in Q_\vee$ and negating the resulting formula. Details in the full version Like lemma 12, given any C⊕D-1-ATA-rfl $\mathcal{A}$, we can construct $\varphi \in$ FRatMTL such that $L(\varphi) = L(\mathcal{A})$. The rest of the proof is very similar to Theorem 13(1) and omitted.

2. $\mathsf{FRatMTL} \subseteq_e \mathsf{C\oplus D\text{-}1\text{-}ATA\text{-}rfl}$: This is almost identical to the proof of Theorem 13(2), and is provided in the full version for completeness. ◀

5.2 FRatMTL meets Q2MSO

▶ **Theorem 18.** FRatMTL *is expressively equivalent to* Q2MSO.

Proof.

1. $\mathsf{Q2MSO} \subseteq_e \mathsf{FRatMTL}$: We first consider formulae of metric depth one. These have the form $\psi(t_0) = \mathcal{Q}_1 t_1 \varphi(\downarrow t_0, t_1)$ and $\varphi(\downarrow t_0, t_1)$ is an MSO formula (bound first order variables t' in φ only have the comparison $t' > t_0$, and there are no free variables other than t_0, t_1, and hence no metric comparison exists in φ). Let re_φ be the regular expression equivalent to $\varphi(\downarrow t_0, t_1)$. The presence of free variables t_0, t_1 implies that re_φ ie over the alphabet $2^\Sigma \times \{0,1\}^2$, where the last two bits are for t_0, t_1. As seen in the case of QkMSO to RatMTL, t_0 is assigned the first position of re_φ since all other variables take up a position to its right. Hence re_φ can be rewritten as $(2^\Sigma, 1, 0)\mathsf{re}'$. Since t_1 is assigned a unique position, there is exactly one occurrence of a symbol of the form $(2^\Sigma, 0, 1)$ in re'. Using (Lemma 7, page 16) [4], we can write re' as a finite union of disjoint expressions each of the form $\mathsf{re}_\ell(\alpha, 0, 1)\mathsf{re}_r$ where $\alpha \in 2^\Sigma$, and $\mathsf{re}_\ell, \mathsf{re}_r \subseteq [(2^\Sigma, 0, 0)]^*$. $\varphi(\downarrow t_0, t_1)$ is thus equivalent to having a symbol $(\alpha, 0, 1)$ at a time point $t \in t_0 + I$, and $(2^\Sigma, 0, 1)\mathsf{re}_\ell$ holds till t, and beyond t, re_r holds. This is captured by the formula $\mathsf{FRat}_{I,\mathsf{re}'}[\bigvee_{\alpha \in 2^\Sigma}(\alpha, 0, 1) \wedge \mathsf{FRat}_{(0,\infty),\mathsf{re}_r}\Box\bot]$. Here, $\mathsf{re}' = (2^\Sigma, 0, 1)\mathsf{re}_\ell$, and the $\Box\bot$ symbolizes the fact that we see re_r in the latter part after $(\alpha, 0, 1)$ and no more symbols after that. See the full version for higher depth case.
2. $\mathsf{FRatMTL} \subseteq_e \mathsf{Q2MSO}$: Proof is similar to $\mathsf{RatMTL} \subseteq_e \mathsf{QkMSO}$ and is in the full version. ◀

6 Discussion

We have defined a new real-time logic QkMSO by extending MSO[<] with guarded quantification with a block of $k-1$ metric quantifiers. We have shown that it is expressively equivalent to 1-ATA where loops do not have clock-reset. We have also shown that it is expressively equivalent to a powerful extension of $\mathsf{MTL}[\mathsf{U_I}]$ called RatMTL. This makes QkMSO as well as RatMTL to be amongst highly expressive, real-time logics (future only) with decidable satisfiability and model-checking. We have established a 4-variable property for QkMSO and also characterized the expressive power of its two variable fragment. The question of a logic expressively equivalent to full 1-ATA remains open. It may be noted that [5] gave an extension of hybrid logic 1-TPTL with fixed point operators to get the expressive power of full 1-ATA but our quest is for a more traditional logic. The expressive power of 3 variable fragment of QkMSO also remains unexplored.

We briefly discuss some special cases and extensions of our results. PO-1-ATA [21] and PO$-$C$\oplus$D-1-ATA are subclasses of 1-ATA and C$\oplus$D-1-ATA respectively, where the only loops in the automaton are reset-free self-loops. Likewise, SfrMTL and FSfrMTL are subclasses of RatMTL and FRatMTL respectively, where the regular expression used in the modality has an equivalent star-free expression.

1. Thanks to theorems 13, 14 and 16, we can obtain the expressive equivalence of (i) QkFO, SfrMTL and PO-1-ATA, and (ii) Q2FO, FSfrMTL and PO$-$C$\oplus$D-1-ATA, In the absence of second order quantification, the regular expressions have a star-free equivalent, and the respective automata are aperiodic.

2. The emptiness problem for non punctual C⊕D-1-ATA is elementarily decidable. The proof is via a reduction to FRatMITL with least fix points [15]. To the best of our knowledge, this is the largest known subclass of 1-ATA which is elementarily decidable.
3. Lastly, to obtain a logical equivalence with 1-ATA, the logic RatMTL with least fixpoint operators suffices [15]. Previously, Haase et al [5] showed that $1 - \mathsf{TPTL}$ with fixed points had the expressive power of 1-ATA.

References

1 Rajeev Alur, Tomás Feder, and Thomas A. Henzinger. The benefits of relaxing punctuality. In *Proceedings of the Tenth Annual ACM Symposium on Principles of Distributed Computing, Montreal, Quebec, Canada, August 19-21, 1991*, pages 139–152, 1991. doi:10.1145/112600.112613.

2 Rajeev Alur and Thomas A. Henzinger. Real-time logics: Complexity and expressiveness. In *Proceedings of the Fifth Annual Symposium on Logic in Computer Science (LICS '90), Philadelphia, Pennsylvania, USA, June 4-7, 1990*, pages 390–401, 1990. doi:10.1109/LICS.1990.113764.

3 Patricia Bouyer, Fabrice Chevalier, and Nicolas Markey. On the expressiveness of TPTL and MTL. In *FSTTCS 2005: Foundations of Software Technology and Theoretical Computer Science, 25th International Conference, Hyderabad, India, December 15-18, 2005, Proceedings*, pages 432–443, 2005. doi:10.1007/11590156_35.

4 J. Engelfriet and H. Hoogeboom. MSO definable string transductions and two-way finite-state transducers. *ACM Trans. Comput. Log.*, 2(2):216–254, 2001.

5 C. Haase, J. Ouaknine, and J. Worrell. On process-algebraic extensions of metric temporal logic. In *Reflections on the Work of C. A. R. Hoare.*, pages 283–300. 2010.

6 Thomas A. Henzinger, Jean-François Raskin, and Pierre-Yves Schobbens. The regular real-time languages. In *Automata, Languages and Programming, 25th International Colloquium, ICALP'98, Aalborg, Denmark, July 13-17, 1998, Proceedings*, pages 580–591, 1998. doi:10.1007/BFb0055086.

7 Y. Hirshfeld and A. Rabinovich. An expressive temporal logic for real time. In *MFCS*, pages 492–504, 2006.

8 P. Hunter. When is metric temporal logic expressively complete? In *CSL*, pages 380–394, 2013.

9 P. Hunter, J. Ouaknine, and J. Worrell. Expressive completeness for metric temporal logic. In *LICS*, pages 349–357, 2013.

10 Neil Immerman and Dexter Kozen. Definability with bounded number of bound variables. *Inf. Comput.*, 83(2):121–139, 1989. doi:10.1016/0890-5401(89)90055-2.

11 J.R.Büchi. On a decision method in restricted second-order arithmetic. In *Proceedings of the 1960 Congress on Logic, Methdology and Philosophy of Science, Stanford Univeristy Press, Stanford*, 1962.

12 S. N. Krishna K. Madnani and P. K. Pandya. Partially punctual metric temporal logic is decidable. In *TIME*, pages 174–183, 2014.

13 Hans Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, Ucla, 1968.

14 Ron Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2(4):255–299, 1990. doi:10.1007/BF01995674.

15 Shankara Narayanan Krishna, Khushraj Madnani, and Paritosh K. Pandya. Büchi-kamp theorems for 1-clock ATA. *CoRR*, abs/1802.02514, 2018. arXiv:1802.02514.

16 S. Lasota and I. Walukiewicz. Alternating timed automata. *ACM Trans. Comput. Log.*, 9(2):10:1–10:27, 2008. doi:10.1145/1342991.1342994.

17 J. Ouaknine and J. Worrell. On the decidability of metric temporal logic. In *LICS*, pages 188–197, 2005.

18 Paritosh K. Pandya and Simoni S. Shah. On expressive powers of timed logics: Comparing boundedness, non-punctuality, and deterministic freezing. In *CONCUR 2011 - Concurrency Theory - 22nd International Conference, CONCUR 2011, Aachen, Germany, September 6-9, 2011. Proceedings*, pages 60–75, 2011. doi:10.1007/978-3-642-23217-6_5.

19 Jean Francois Raskin. *Logics, Automata and Classical Theories for Deciding Real Time*. PhD thesis, Universite de Namur, 1999.

20 S.Krishna, K. Madnani, and P. K. Pandya. Metric temporal logic with counting. In *FoSSaCS*, pages 335–352, 2016.

21 P. K. Pandya S.Krishna, K. Madnani. Making metric temporal logic rational. In *MFCS*, 2017.

22 T. Wilke. Specifying timed state sequences in powerful decidable logics and timed automata. In *FTRTFT*, pages 694–715, 1994.