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


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 $MTL[U_1, S_1]$ all have the same expressive power [2]. It took 15 years to show that $MTL[U_1, S_1]$ is less expressive than $TPTL$ over timed words [18] [3]. It is only in the last few years, that extensions of $MTL[U_1, S_1]$ 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 $MTL[U_1, S_1] \cap \text{I^n \text{S^n \text{T^n}}} \text{ where time interval constraints I are non-singular intervals} [1]$. They showed the decidability of this logic by reducing its formulae essentially to language equivalent non-deterministic timed automata. However,
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, \( L^{d} \), 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 \( L^{d} \) 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 languages. 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 \( \text{MTL[U]} \) (with pointwise interpretation) over finite words is decidable. The result was proved by constructing a language equivalent 1-ATA for a formula of \( \text{MTL[U]} \) [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 \( \text{MTL[U]} \) with increasing expressive power culminating in \( \text{RatMTL} \) [21]. Logic \( \text{RatMTL} \) is a powerful extension of \( \text{MTL[U]} \) 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 MSO[<] over words by adding guarded quantification with blocks of at most \( k-1 \) metric quantifiers to give a real time logic \( \text{QkMSO} \). 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 \( \text{Q2MLO} \) (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 \( \text{QkMSO} \) unlike \( \text{Q2MLO} \).

We investigate the decidability and expressive power of \( \text{QkMSO} \). 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) \( \text{QkMSO} \), b) the 1-ATA-rfl, and, c) the metric temporal logic \( \text{RatMTL} \), introduced earlier in [21], are all expressively equivalent (with effective reductions). In the process, we also prove that \( \text{QkMSO} \) has four variable property over timed words. This can be seen akin to the famous 3-variable property of \( \text{FO[<]} \) over words [10].

For our second main result, we turn to the two variable fragment \( \text{Q2MSO} \) of \( \text{QkMSO} \). For its automaton characterization, we introduce a syntactic restriction of conjunctive-disjunctiveness (C\(\oplus\)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) \( \text{Q2MSO} \), b) the C\(\oplus\)D-1-ATA-rfl, and c) the sublogic \( \text{FRatMTL} \) of \( \text{RatMTL} \) have exactly the same expressive power. Here logic \( \text{FRatMTL} \) uses only restricted version \( \text{FRat} \) of the
modality \textit{Rat} of \textit{RatMTL}. A similar modality was also defined earlier by Wilke [22] in logic \textit{EMITL}. In summary, we show that

\[
\begin{array}{ccc}
QkMSO & \equiv & 1\text{-ATA}\text{-rfl} \equiv \text{RatMTL} \\
Q2MSO & \equiv & C\oplus D\text{-}1\text{-ATA}\text{-rfl} \equiv \text{FRatMTL}
\end{array}
\]

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

2 Preliminaries

Let \(\Sigma\) be a finite set of propositions. A finite timed word over \(\Sigma\) is a tuple \(\rho = (\sigma, \tau)\), where \(\sigma\) and \(\tau\) are sequences \(\sigma_1\sigma_2\ldots \sigma_n\) and \(\tau_1\tau_2\ldots \tau_n\) respectively, with \(\sigma_i \in \Gamma = 2^\Sigma \setminus \emptyset\), and \(\tau_i \in \mathbb{R}_{\geq 0}\) for \(1 \leq i \leq n\). For all \(i \in \text{dom}(\rho)\), we have \(\tau_i \leq \tau_{i+1}\), where \(\text{dom}(\rho)\) is the set of positions \(\{1, 2, \ldots, n\}\) in the timed word. For convenience, we assume \(\tau_1 = 0\). The \(\sigma_i\)'s can be thought of as labeling positions \(i\) in \(\text{dom}(\rho)\). For example, given \(\Sigma = \{a, b, c\}\), \(\rho = ((a, c, e), 0)((a, 0.1)((b, 1.1)\) is a timed word. \(\rho\) is strictly monotonic if \(\tau_i < \tau_{i+1}\) for all \(i, i + 1 \in \text{dom}(\rho)\). Otherwise, it is weakly monotonic. The set of finite timed words over \(\Sigma\) is denoted \(T\Sigma^*\). Given \(\rho = (\sigma, \tau)\) with \(\sigma = \sigma_1 \ldots \sigma_n \in \Gamma^+\), \(\sigma\text{\text{single}}\) denotes the set of all words \(w_1w_2\ldots w_n\) where each \(w_i \in \sigma_i\). \(\rho\text{\text{single}}\) consists of all timed words \((\sigma\text{\text{single}}, \tau)\). For the \(\rho\) as above, \(\rho\text{\text{single}}\) 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 \(\Sigma\), the formulae of logic \textit{MTL} are built from: \(\varphi ::= a(e \in \Sigma) \mid \text{true} \mid \varphi \land \varphi \mid \neg \varphi \mid \varphi \cup I\nu\varphi\), where \(I \in I\nu\). For a timed word \(\rho = (\sigma, \tau) \in T\Sigma^*\), a position \(i \in \text{dom}(\rho)\), and an \textit{MTL} formula \(\varphi\), the satisfaction of \(\varphi\) at a position \(i\) of \(\rho\) is denoted \(\rho, i \models \varphi\), and is defined as follows: (i) \(\rho, i \models a \iff a \in \sigma_i\), (ii) \(\rho, i \not\models \neg \varphi \iff \rho, i \not\models \varphi\), (iii) \(\rho, i \models \varphi_1 \land \varphi_2 \iff \rho, i \models \varphi_1\) and \(\rho, i \models \varphi_2\), (iv) \(\rho, i \models \varphi_1 \cup I\nu\varphi_2 \iff \exists j > i, \rho, j \models \varphi_2, \tau_j - \tau_i \in I\), and \(\rho, k \models \varphi_1 \forall i < k < j\). The language of a \textit{MTL} formula \(\varphi\) is \(L(\varphi) = \{\rho \mid \rho, 1 \models \varphi\}\). Two formulae \(\varphi\) and \(\phi\) are said to be equivalent denoted as \(\varphi \equiv \phi\) if \(L(\varphi) = L(\phi)\). The subclass of \textit{MTL} restricting the intervals \(I\) in the until modality to non-punctual intervals is denoted \textit{MTL}. Punctual intervals like \([2,2]\) are disallowed. Note that we restrict to until-only fragment of \textit{MTL} for the sake of decidability.

\begin{itemize}
\item \textbf{Theorem 1 ([17]).} \textit{MTL} satisfiability is decidable over finite timed words with non-primitive recursive complexity.
\end{itemize}

\textbf{MTL with Rational Expressions (RatMTL)}

We first recall an extension of \textit{MTL} with rational expressions (RatMTL), introduced in [21]. The modalities in \textit{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
39:4 Logics Meet 1-Clock Alternating Timed Automata

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

Rat\(\text{MTL}\) Syntax. For a timed word \(\rho = (\sigma, \tau) \in T\Sigma^*\), a position \(i \in dom(\rho)\), a Rat\(\text{MTL}\) formula \(\varphi\) and a finite set \(S\) of subformulas of \(\varphi\), we define the satisfaction of \(\varphi\) at a position \(i\) as follows. For positions \(i < j \in dom(\rho)\), let \(\text{Seg}(\rho, S, i, j)\) denote the untimed word over \(2^S\) obtained by marking the positions \(k \in \{i + 1, \ldots, j - 1\}\) of \(\rho\) with \(\psi \in S\) if \(\rho, k \models \psi\). For a position \(i \in dom(\rho)\) and an interval \(I\), let \(\text{TSeg}(\rho, S, I, i)\) denote the untimed word over \(2^S\) obtained from \(\rho\) by marking all the positions \(k\), where \(\tau_k - \tau_i \in I\), with \(\psi \in S\) if \(\rho, k \models \psi\).

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

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

1. In [21], the URat modality was used instead of FRat; however, both have the same expressiveness. Note that \(\varphi_1\text{URat}_{I, \text{re}(S)}(\varphi_2)\) is equivalent to \(\text{FRat}_{I, \text{re}((S \cup \{\varphi_1\})^\ast)}(\varphi_2)\) where \(\text{re}'(S \cup \{\varphi_1\}) = \text{re}(S) \cap \varphi_1^\ast\).

2. The classical \(\varphi_1\text{URat}_{1}(\varphi_2)\) modality can be written in FRat\(\text{MTL}\) as \(\text{FRat}_{1, \text{re}^\ast}(\varphi_2)\). 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 Rat\(\text{MTL}\) formula is defined as follows. Let PL\(\Sigma\) be the set of propositional logic formulae over \(\Sigma\) (up to equivalence). An atomic Rat\(\text{MTL}\) formula over \(\Sigma\) is an element of PL\(\Sigma\) and has modal depth 0. A Rat\(\text{MTL}\) formula \(\varphi\) over \(\Sigma\) with a single modality (Rat or FRat) has modal depth one and has the form Rat\(\text{re}\) or FRat\(\text{I, re}\) where \(\text{re}\) is a regular expression over PL\(\Sigma\) and \(\psi \in \text{PL}_{\Sigma}\). Inductively, we define modal depth as follows: (i) \(md(\varphi \land \psi) = md(\varphi \lor \psi) = \max(md(\varphi), md(\psi))\). Similarly, \(md(\neg \varphi) = md(\varphi)\). (ii) Let \(\text{re}\) be a regular expression over the set of subformulas \(S = \{\psi_1, \ldots, \psi_k\}\).

\begin{align*}
\text{md(\text{FRat}_{I, \text{re}}(\varphi))} &= 1 + \max(\psi_1, \ldots, \psi_k, \varphi) \text{, } \\
\text{md(\text{Rat}_{I, \text{re}}(\varphi))} &= 1 + \max(\psi_1, \ldots, \psi_k).
\end{align*}

\textbf{Example 2.} Consider the formula \(\varphi = \text{Rat}_{[1, 1]}(\text{Rat}_{[0, 1]}(aa)^\ast)\). Then \(\varphi = \text{Rat}_{[1, 1]}(aa)^\ast\). The subformulæ of interest are \(S = \{\text{Rat}_{[0, 1]}(aa)^\ast, a\}\). For \(\rho = ([\{a\}, 0] ([\{a\}, 0.9]) ([\{a\}, 1]) ([\{a\}, 1.2], \rho, 3 \neq \text{re}_1\), since \([\text{TSeg}(\rho, S, (0, 1), 3)^\text{single} \cap L((aa)^\ast) = \emptyset\). 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 \(aa \in [\text{TSeg}(\rho, S, (0, 1), 3)^\text{single} \cap L((aa)^\ast).\)
2.2 MSO with guarded metric quantifiers $Q_k \text{MSO}$

Let $\rho = (\sigma, \tau)$ be a timed word over a finite alphabet $\Sigma$, as before. We define a real-time logic $Q_k \text{MSO}$ (with parameter $k \in \mathbb{N}$) which is interpreted over such words. It includes $\text{MSO}^\omega$ over words $\sigma$ 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 $Q_k \text{MSO}$ which are mutually recursively defined: these are $\text{MSO}^\omega$ formulae $\phi$ which have no real-time constraints except time constraint subformulae $\psi(t_i)$. 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 $Q_k \text{MSO}$ 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 $Q_2 \text{MLO}$, which we refer in this paper as $Q_2 \text{FO}$, 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 $\text{MSO}$ formulae $\phi$ and time constrained formulae $\psi$. 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 $\phi$ is relativized to the future of some variable, say $t_0$, called anchor variable, giving formulae of $\text{MSO}^\omega$. The syntax of $\phi$ in $\text{MSO}^\omega$ is given by: $t_0 = \tau \phi \land \phi | \exists \phi \land t_0 \land \phi | \exists T \phi \land \psi(t_p)$. Here, $\psi(t_p) \in \text{MSO}^\omega$ is a time constraint formula whose syntax and semantics are given later. A formula in $\text{MSO}^\omega$ 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 $\phi(\downarrow t_0, t_1, \ldots, t_k, T_1, \ldots, T_m)$. (The $\downarrow$ is only to indicate the anchor variable. It has no other function.) The semantics of such formulae is as usual. Given $\rho$, positions $a_0, \ldots, a_k$ in $\text{dom}(\rho)$, and sets of positions $A_1, \ldots, A_m$ with $A_i \subseteq \text{dom}(\rho)$, 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. $\rho, a_0, a_1, \ldots, a_k, A_1, \ldots, A_m \models Q \phi$ iff $a \in \sigma(a_i)$,
3. $\rho, a_0, a_1, \ldots, a_k, A_1, \ldots, A_m \models T \phi$ iff $a_i \in A_i$,
4. $\rho, a_0, a_1, \ldots, a_k, A_1, \ldots, A_m \models \exists a \phi(\downarrow t_0, t_k, \ldots, t_k, T_1, \ldots, T_m)$ iff

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

for some $a_k \geq a_0$.

The time constraint $\psi(t_0)$ has the form $Q_1 t_1 Q_2 t_2 \ldots Q_j t_j \phi(\downarrow t_0, t_1, \ldots, t_j)$ where $\phi \in \text{MSO}^\omega$ and $j < k$, the parameter of logic $Q_k \text{MSO}$. Each quantifier $Q_i t_i$ has the form $\exists \tau \in t_0 + I_i$ or $\forall \tau \in t_0 + I_i$ for a time interval $I_i$ as in MTL formulae. $Q_i$ is called a metric quantifier. The semantics of such a formula is as follows. $\rho, a_0 \models Q_1 t_1 Q_2 t_2 \ldots Q_j t_j \phi(\downarrow t_0, t_1, \ldots, t_j)$ iff for $1 \leq i \leq j$, there exist/for all $a_i$ such that $a_0 \leq a_i$ and $\tau_{a_i} \in t_{a_i} + I_i$, 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 $\psi(x) = \exists y \in x + (2, \infty) \exists z \in x + (3, \infty)(Q_b(y) \land Q_b(z))$. It can be seen that $\rho, 1 \models Q_a(x) \land \psi(x)$.

**Metric Depth.** The metric depth of a formula $\phi$ denoted $(\text{md}(\phi))$ gives the nesting depth of time constraint constructs. It is defined inductively as follows: For atomic formulae $\phi$, $\text{md}(\phi) = 0$. All the constructs of $\text{MSO}^\omega$ do not increase $\text{md}$. For example, $\text{md}(\phi \land \phi_2) = \max(\text{md}(\phi), \text{md}(\phi_2))$ and $\text{md}(\exists t_0 \phi(t))$. However, $\text{md}$ is incremented for each application of metric quantifier block. $\text{md}(Q_1 t_1 Q_2 t_2 \ldots Q_j t_j \phi) = \text{md}(\phi) + 1$. 

**CONCUR 2018**

S. N. Krishna, K. Khushraj, and P. K. Pandya 39:5
Example 4. The sentence \( \forall t_0 \forall t_1 \in t_0 + (1,2) \{ Q_a(t_1) \rightarrow (\exists t_0 \in t_1 + [1,1] Q_b(t_0)) \} \) 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, \( \exists x. \exists t. [X(t) \land \exists t' \in t + (1,2) Q_a(t')] \) is a well-formed QkMSO formula while, \( \exists x. \exists t. \exists t' \in t + (1,2) [Q_a(t') \land X(t)] \) 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 \( \Sigma \) be a finite alphabet and let \( \Gamma = 2^\Sigma \setminus \emptyset \). A 1-ATA \([17\] [16]) is a 5 tuple \( A = (\Gamma, S, s_0, F, \delta) \), 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 \( x \in J \) denotes a clock constraint where \( J \) is an interval. Let \( X \) denote a finite set of clock constraints of the form \( x \in I \). The transition function is defined as \( \delta : X \times \Gamma \rightarrow \Phi(S \cup X) \) where \( \Phi(S \cup X) \) is a set of formulae over \( S \cup X \) defined by the grammar \( \varphi ::= \top \mid \bot \mid \varphi_1 \land \varphi_2 \mid \varphi_2 \lor \varphi_2 \mid s | x \in I | x.[\varphi \lor \varphi] \) where \( s \in S \), and \( x.[\varphi \lor \varphi] \) 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 \( S \times x.\varphi \), 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:

\[
M \models_\nu s \text{ iff } (s,\nu) \in M ; M \models_\nu x \in I \text{ iff } \nu \in I ; M \models_\nu x.s \text{ iff } (s,0) \in M
\]

The conjunctions, disjunctions, \( \top \) and \( \bot \) are handled in the usual way. We say that \( M \) is a minimal model of \( \varphi \in \Phi(S \cup X) \) with respect to \( \nu \) if \( M \models_\nu \varphi \) 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 \( C = \{(s,\nu) \mid s \in S, \nu \in \mathbb{R}_{\geq 0} \} \), we denote by \( C + t \) the configuration \( \{(s,\nu + t) \mid (s,\nu) \in C \} \), obtained after a time elapse \( t \), i.e., when \( t \) is added to all the valuations in \( C \). Let \( \alpha \in \Gamma \). Given any configuration \( C_\alpha = \{(s_j,\nu_j) \mid j \in J \} \), we say that any configuration \( C_\alpha \) is a \( \alpha \)-discrete successor of \( C_\alpha \) if and only if \( C_\alpha \) can be constructed using \( C_\alpha \) by choosing a minimal model \( M_j \) of \( \delta(s_j,\nu_j) \) with respect to \( \nu_j \) for every \( j \in J \) followed by taking their union. That is, \( C_\nu = \bigcup_{j \in J} M_j \) where \( M_j \) is a minimal model of \( \delta(s_j,\nu) \) with respect to \( \nu_j \). Given a timed word \( \rho = (\alpha_0,0)(\alpha_1,t_1) \ldots (\alpha_m,t_m) \), a run associated with \( \rho \) over a given 1-ATA starts from the initial configuration \( C_0 = \{(s_0,0)\} \) and has the form \( C^0 \xrightarrow{\alpha_0} C_1 \xrightarrow{t_1=0} C_1 + t_1 \ldots C_{m-1} + (t_m - t_{m-1}) \xrightarrow{\alpha_m} C_m \) and proceeds with alternating time elapse transitions and discrete transitions reading a symbol from \( \Sigma \). A discrete transition \( C_i + (t_i - t_{i-1}) \xrightarrow{\alpha_i} C_{i+1} \) is included if and only if \( \alpha_i = \alpha \), the \( \alpha \)-discrete successor of \( C_i \) is \( \{t_i - t_{i-1}\} \). Note that there could be more than one \( \alpha \)-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 \( C \) is accepting iff for all \((s,\nu) \in C, s \in F \). Note that the empty configuration is also an accepting configuration.
We say that any timed word \( s \) is a conjunction of \( \{s_0, 0\} \) ... \( \{s_m, t_m\} \), a time delayed run associated with \( \rho \) over a given 1-ATA starts from the initial configuration \( C_0' = \{(s_0, 0)\} \) and has the form \( C_0' \overset{t_0}{\rightarrow} C_0' + t_0 - 0 \overset{t_1}{\rightarrow} C_1' + (t_2 - t_1) \cdot \ldots \cdot \overset{t_n}{\rightarrow} C_n' \), where \( C_{i-1}' + (t_i - t_{i-1}) \overset{a_i}{\rightarrow} C_i' \) is included if and only if \( C_i' \) is one of the \( a_i \)-discrete successors of \( C_{i-1}' + (t_i - t_{i-1}) \). 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 \( \rho \) is accepted in time-delayed semantics by an \( \text{ATA} \mathcal{A} = (\Gamma, S, s_0, F, \delta) \), if and only if, there exists a time-delayed run associated with \( \rho \) which ends with an accepting configuration. The set of all timed words accepted by time-delayed semantics of \( \mathcal{A} \) is \( L^\mathcal{A}(\mathcal{A}) \).

Consider the 1-ATA \( \mathcal{A} = (\Gamma, \{ s_0, s_1, s_2, s_3 \}, s_0, \{ s_2 \}, \delta_{\mathcal{A}}) \) with transitions \( \delta_{\mathcal{A}}(s_0, a) = s_1; \delta_{\mathcal{A}}(s_1, a) = \{ s_2 \} \cup \{ s_3 \}; \delta_{\mathcal{A}}(s_2, a) = s_3; \delta_{\mathcal{A}}(s_3, a) = s_4 \). Consider a timed word \( \rho = \{(a_0, 0)\} \overset{\{\{b\}, 0.5\}}{\rightarrow} \{(s_1, 0.5)\} \overset{\{s_3, 0.5\}}{\rightarrow} \{(s_3, 1.2)\} \overset{\{s_2, 1.2\}}{\rightarrow} \{(s_2, 1.2)\} \). Thus \( \rho \) 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 \( \delta(s, a) = C_1 \lor \cdots \lor C_n \) in the 1-ATA. Each \( C_i \) is a conjunction of \( a \in I \), 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 \( F_1 \) be a logic or automaton class i.e. a collection of formulae or automata describing/accepting finite timed words. For each \( \phi \in F_1 \) let \( L(F_1) \) denote the language of \( F_1 \). We define \( F_1 \subseteq_* F_2 \) 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 \( F_1 \equiv_* F_2 \), if \( F_1 \subseteq_* F_2 \) and \( F_2 \subseteq_* F_1 \).
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 $\delta(s, a) = C_1 \lor C_2 \ldots \lor C_k$ with $C_i = x \in I \land p_1 \land \ldots \land p_n \land x.q_i \land \ldots \land x.q_r$, we have (a) each $q_i \in S_r$, and (b) If $s \in P_i$, then each $p_j \in P_i - \{s_i^r\}$.  

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, $\text{Norm}(\mathcal{A})$. 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 $\text{Norm}(\mathcal{A})$ is $s_0^r$. The island $P_i$ in $\text{Norm}(\mathcal{A})$ consists of locations $s_i^r, s_i^{nr,j}$ for $1 \leq h \leq n$; entry into $P_i$ happens through the header $s_i^r$. A transition $\delta(s, a) = \varphi$ of $\mathcal{A}$ is rewritten in $\text{Norm}(\mathcal{A})$ by replacing all occurrences of locations $x.s_j$ with $x.s_i^r$ (leading into $P_j$), while occurrences of free locations $s_h$ are replaced with $s_h^{nr,j}$. The final locations of $\text{Norm}(\mathcal{A})$ 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 $\text{Norm}(\mathcal{A})$ in normal form such that $L(\mathcal{A}) = L(\text{Norm}(\mathcal{A}))$.

Example 8. The 1-ATA $\mathcal{B} = \{(a, b), \{s_0, s_1, s_2\}, \{s_1\}, \delta\}$ with transitions $\delta(s_0, b) = x.s_2, \delta(s_0, a) = (s_0 \land x.s_1), \delta(s_1, a) = (s_1 \land s_0) = \delta(s_1, b)$ and $\delta(s_2, b) = x.s_0, \delta(s_2, a) = (s_2 \land x.s_1)$ is not in normal form. Following the normalization technique, we obtain $\text{Norm}(\mathcal{B})$ with locations $S = \{s_i^r, s_i^{nr,j} \mid 0 \leq i, j \leq 2\}$ and final locations $\{s_1^r, s_1^{nr,0}, s_1^{nr,1}, s_1^{nr,2}\}$. Figure 1(right) describes $\text{Norm}(\mathcal{B})$ 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 $S_r$. 

---
Conversely, $q$ (hence that for any good region word is a word over the alphabet which will be clearer later, the region abstraction of the word defined here is by ignoring the

Theorem. The set of timed words $I$.

The 1-ATA with locations $s, p, q$ and transitions $\delta(s, a) = (x, p \land x \leq 1) \lor (q \land x = 2)$, $\delta(p, a) = x, q \land p$ and $\delta(q, a) = s \land (0 < x < 1)$ is not a-ATA-rlf, since $q$ is bound in $\delta(p, a)$ 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 $\rho = (a_0, 0)(a_1, \tau_1)(a_2, \tau_2) \ldots (a_m, \tau_m)$ be a timed word and let $i \in \text{dom}(\rho)$. Let $\text{rel}(\rho, i)$ denote the timed word obtained from $\rho$ by relativizing at position $i$; $\text{rel}(\rho, i) = (a_i, 0)(a_{i+1}, \tau_{i+1} - \tau_i)(a_{i+2}, \tau_{i+2} - \tau_i) \ldots (a_m, \tau_m - \tau_i)$. 

Region Words. Let $c_{\text{max}}$ be any non-negative integer. Let $\text{reg}_{c_{\text{max}}} = \{0, (0, 1), \ldots, c_{\text{max}}, (c_{\text{max}}, \infty)\}$ be the set of regions. Given a finite alphabet $\Sigma$, with $\Gamma = 2^\Sigma \setminus \emptyset$, a region word is a word over the alphabet $\Gamma \times \text{reg}_{c_{\text{max}}}$ called the interval alphabet. A region word $w = (a_1, I_1)(a_2, I_2) \ldots (a_m, I_m)$ 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 $\rho = (a_0, 0)(a_1, \tau_1)(a_2, \tau_2) \ldots (a_m, \tau_m)$ is consistent with a region word $(a_1, I_1)(a_2, I_2) \ldots (a_m, I_m)$ 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 $\rho$ consistent with a good region word $w$ is denoted $T_w$. Likewise, given a timed word $\rho$, $\text{reg}(\rho)$ represents the good region word $w$ such that $\rho \in T_w$.

Lemma 10 (Untiming $P$ to $A(P)$). Let $P$ be a 1-ATA over $\Gamma$ having no resets. We can construct an alternating finite automaton $A(P)$ over the interval alphabet $\Gamma \times \text{reg}_{c_{\text{max}}}$ such that for any good region word $w = (a_1, I_1) \ldots (a_n, I_n)$, $w \in L(A(P)) \Rightarrow \forall \rho \in T_w, \rho \in L^\rho(P)$. Conversely, $\rho \in L^\rho(P) \Rightarrow \text{reg}(\rho) \in L(A(P))$. Hence, $L^\rho(P) = \{\rho \mid w \in L(A(P) \land \rho \in T_w\}$.

Proof. Given a reset-free 1-ATA $P = (\Gamma, S, s_0, F, \delta)$ we construct an alternating finite automaton (AFA) $A(P) = (\Gamma \times \text{reg}_{c_{\text{max}}}, S, s_0, F, \delta')$ where $c_{\text{max}}$ is the maximum constant used in the clock constraints of $P$ and $\delta'$ is defined using $\delta$ as follows. A clock constraint
$x \in I$ is called a region constraint if $I \in \text{reg}_{\max}$. Without loss of generality, we assume that all the transitions of $P$ have the form $C_1 \lor \cdots \lor C_m$, where each clause $C_i$ has exactly one region constraint.

**Construction of $\delta'$.** Given a transition $\delta(s, a) = C_1 \lor \cdots \lor C_n$ of $P$, let $C_{t_1}, \ldots, C_{t_k} \in \{C_1, \ldots, C_n\}$ be clauses containing the region constraint $x \in I$, $I \in \text{reg}_{\max}$. We construct $\delta'(s, (a, I))$ as $C_{t_1}' \lor \cdots \lor C_{t_k}'$, where $C_{t_j}'$ is obtained by removing the conjunct $x$ from $C_{t_j}$. If there is no clause containing $x \in I$ in $\delta(s, a)$, then $\delta'(s, (a, I)) = \bot$.

1. For any timed word $\rho, \rho \in L^x(P) \Rightarrow \text{reg}(\rho) \in L(A(P))$: Let $\rho = (a_1, 0) \ldots (a_m, \tau_m)$. The time-delayed accepting run from the initial configuration $C_0 = \{(s_0, 0)\}$ to an accepting configuration $C_m$ in $P$ is as follows: $C_0 \xrightarrow{\tau_1} C_0 \xrightarrow{a_1} C_1 \xrightarrow{\tau_2} \ldots \xrightarrow{\tau_m-\tau_{m-1}} C_{m-1} \xrightarrow{\tau_m} C_m$. As $P$ is a reset-free 1-ATA, the valuation $\nu(x)$ at any point is equal to the total time elapsed till that point. Hence for all positions $j \in \text{dom}(\rho), I \in \text{reg}_{\max}, \tau_j \in I$ iff $\nu(x) \in I$. By construction of $A(P)$, for each transition $\delta(s, a_j) = (x \in I_j \land \psi) \lor C$ of $P$ we have a transition $\delta'(s, (a, I_j)) = \psi$ (wlg we assume that $C$ has no occurrence of $x \in I_j$).

The accepting run of $P$ translates into the run $D_0 \xrightarrow{(a_1, I_1)} D_1 \xrightarrow{(a_2, I_2)} \cdots \xrightarrow{(a_m, I_m)} D_m$ in $A(P)$, where $D_0 = \{s \mid (s, 0) \in C_0\}$, and $D_j = \{s \mid (s, t) \in C_j, t \in I_j\}, j \geq 1$. Since all locations in $C_m$ are accepting, $D_m$ is an accepting configuration in $A(P)$ accepting $(a_1, I_1) \ldots (a_m, I_m)$.

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

**Proof.** Let $\text{Det}[A(P)]$ be the deterministic automaton which is language equivalent to $A(P)$. Let $\delta_D$ be the transition function of $\text{Det}[A(P)]$ obtained from $\delta'$ (see lemma 10) and let $\delta_D$ be its extension to words. Let $s_0$ be the initial location of $\text{Det}[A(P)]$ and let $F$ be the set of its final locations. For any pair of locations $p, q$ of $\text{Det}[A(P)]$ and $I_1 \in \text{reg}_{\max}$, we construct a regular expression $\text{re}(p, q, I_1)$ equivalent to the language $\{w \in (\Gamma \times \{I_1\})^+ \mid \delta_D(p, w) = q\}$. Let $\text{Det}(A(P), [p, q])$ be the DFA obtained from $\text{Det}[A(P)]$ by setting the initial location to $p$ and set of final locations to $\{q\}$. Let $A(I')$ denote the DFA accepting all words $(\Gamma \times \{I\})^*$. Let $\text{Det}(A[I] = \text{Det}[A(P), [p, q]] \cap A(I')$. Then $L(\text{re}(p, q, I_1)) = L(\text{Det}[A[I]])$.

**Obtaining RatMTL formula.** Consider a sequence of locations $\text{ssseq} = q_0, q_1, q_2 \ldots q_k$ with $k = 2 \times n_{\max} + 2, q_0 = s_0$ and $q_k \in F$. Let $\text{re} = \text{re}(q_0, q_1, [0, 0]) \cap \text{re}(q_1, q_2, (1, 0)) \cap \cdots \cap \text{re}(q_{2i-1}, q_{2i}, (i, i)) \cap \text{re}(q_{2i}, q_{2i+1}, (i, i)) \cap \text{re}(q_{2i+1}, q_{2i+2}, (i, i)) \cap \cdots \cap \text{re}(q_{n_{\max} \times 2}, q_{n_{\max} \times 2}, (n_{\max}, n_{\max}))$. Define the RatMTL formula $\phi(\text{ssseq}) = \text{Rat}_{\text{ssseq}}(q_0, q_{1}, [0, 0]) \land \text{Rat}_{\text{ssseq}}(q_1, q_{2}, (0, 1)) \land \cdots \land \text{Rat}_{\text{ssseq}}(q_{n_{\max} \times 2 - 1}, q_{n_{\max} \times 2}, (n_{\max}, n_{\max}))$. Then $L(\phi(\text{ssseq})) = \{\mathcal{T}_w \mid w \in L(\text{re}(q_0, q_1, [0, 0]) \cap \text{re}(q_1, q_2, (1, 0)) \cap \cdots \cap \text{re}(q_{n_{\max} \times 2 - 1}, q_{n_{\max} \times 2}, (n_{\max}, n_{\max}))\}$. Let $\varphi^x = \bigvee_{\text{ssseq}} \phi(\text{ssseq})$. Then clearly, $L(\varphi^x) = \{\mathcal{T}_w \mid w \in L(A(P))\}$. By lemma 10, we obtain $L(\varphi^x) = L^x(P)$. 

---

2 The superscript $\text{st}$ in $\varphi^x$ 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. \( \rho, i \models \varphi \) iff \( \text{rel}(\rho, i) \in L^{\text{at}}(P) \). \( \rho, i \models \varphi \) iff \( \text{rel}(\rho, i) \in L(\varphi) \) since the truth of \( \varphi \) 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 \( A(P) \) be an AFA as constructed in lemma 10. Let \( \varphi \) be the formula constructed from \( A(P) \) as in lemma 11. We can construct a formula \( \varphi \in \text{RatMTL} \) 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 \( \subseteq \) RatMTL:** Let \( A \) 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 \( A[s_i^r] \) denote the same automaton as \( A \) 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 \( A[s_i^r] \), we construct a pair of RatMTL formulae \( \text{mtl}(A[s_i^r]) \) and \( \text{mtl}^r(A[s_i^r]) \) such that \( L(\text{mtl}(A[s_i^r])) = L(A[s_i^r]) \) and \( L(\text{mtl}^r(A[s_i^r])) = L^r(A[s_i^r]) \). Note that \( (S_r, \leq) \) 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 \( x, s_i^r \) occurring in any transition of \( A[s_i^r] \) are of lower level in the partial order \( (S_r, \leq) \). Hence, by induction hypothesis, there is a RatMTL formula \( \psi^r \) such that \( L^r(A[s_i^r]) = L(\psi^r) \). The behaviour of all the lower level \( s_i^r \) is independent of the label of the transition that calls them. In other words, the automata \( A_i[s_i^r] \) which is called at position \( i \) does not read the symbol at point \( i \). Thus any lower level automaton starting from some \( s_i^r \) 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 \( x, s_i^r \) above, which also corresponds to RatMTL formula \( \psi^r \). Let the set of such witness variables be \( \{w_1, \ldots, w_k\} \). We construct a modified automaton \( \text{A}^w[s_i^r] \) with transition function \( \delta^w \) and set of locations \( P_i \) as follows. Its alphabet is \( \Gamma \times \{0, 1\}^k \) with the \( j \)-th component giving the truth value of witness \( w_j \). Let \( \delta^w(s, a, w_1, \ldots, w_k) = \delta(s, a)[w_j/x.s_i^r] \), i.e., each occurrence of \( x, s_i^r \) is replaced by the truth value of \( w_j \) for \( 1 \leq j \leq k \). Note that \( \text{A}^w[s_i^r] \) is a reset-free 1-ATA. By lemmas 10, 11, we get a RatMTL formula \( \phi^w \) and formula \( \phi^w \) such that \( L(\text{A}^w[s_i^r]) = L(\phi^w) \) and using lemma 12 we get a language equivalent formula \( \phi^w \) over the variables \( \Gamma \cup \{w_1, \ldots, w_k\} \). Now we substitute each \( w_j \) by \( \psi^r \) and call \( \phi^w \) by \( \phi^w \) and \( \phi^w \) to obtain the required formulae \( \text{mtl}(A[s_i^r]) \) and \( \text{mtl}^r(A[s_i^r]) \), respectively. It is clear from the substitution that \( L^r(A[s_i^r]) = L(\text{mtl}^r(A[s_i^r])) \) and \( L(A[s_i^r]) = L(\text{mtl}(A[s_i^r])) \). An example illustrating this construction is in the full version.

2. **RatMTL \( \subseteq \) 1-ATA-rfl:** Consider a formula \( \psi_1 = \text{Rat}_r(\text{re}_0) \) with \( I = [l, u] \). The case of other intervals are handled similarly. For the base case, \( \psi_1 \) has modal depth 1 and has a single modality. As the formula is of modal depth 1, \( \text{re}_0 \) is an atomic regular expression over alphabet \( 2^\Gamma \). Let \( D = (\Gamma, Q, q_0, Q_f, \delta) \) be a DFA such that \( L(D) = L(\text{re}_0) \), with \( \Gamma = 2^\Gamma \setminus \emptyset \). From \( D \), we construct the 1-ATA \( \mathcal{A} = (\Gamma, Q \cup \{q_{\text{init}}, q_{\text{timecheck}}, q_f\}, Q_{\text{init}}, \{q_f\}, \delta) \) where \( q_{\text{init}}, q_{\text{timecheck}}, q_f \) are disjoint from \( Q \). The transitions \( \delta \) are as follows. Assume \( l > 0 \).

- \( \delta(q_{\text{init}}, a) = x.q_{\text{timecheck}} \), for all \( q_{\text{init}} \).
- \( \delta(q_{\text{timecheck}}, a) = [(x \geq l) \wedge \delta'(q_{\text{init}}, a)] \vee [(x > u) \wedge q_f] \) where the latter disjunct is added only when \( q_{\text{init}} \).
- \( \delta(q, a) = (x \in [l, u]) \wedge \delta'(q, a) \), for all \( q \in Q \setminus Q_f \).
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 \( q_{\text{time check}} \) is entered on the first symbol, resetting the clock; control stays in \( q_{\text{time check}} \) 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 \( q_{\text{time check}} \), we enter \( q_f \) when \( x > u \). It is clear that \( \mathcal{A} \) indeed checks that \( \text{re}_0 \) is true in the interval \([l, u]\). If \( l = 0 \), then the interval on which \( \text{re}_0 \) should hold good is \([0, u]\). In this case, if \( q_0 \) is non-final, we have the transition \( \delta(q_{\text{init}}, a) = x, \delta'(q_0, a), a \in \Gamma \) (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 \( q_{\text{time check}} \) is not used then. The case when \( \psi \) has modal depth 1 but has more than one \( \text{Rat} \) modality is dealt as follows. Firstly, if \( \psi = \neg \text{Rat}_r(\text{re}_0) \), then the result follows since 1-ATA-\( \text{Rfl} \) 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 \( \psi_1 \land \psi_2 \) of formulae, having 1-ATA-\( \text{Rfl} \) \( A_1 = (\Gamma, Q_1, q_1, F_1, \delta_1) \) and \( A_2 = (\Gamma, Q_2, q_2, F_2, \delta_2) \) such that \( L(A_1) = L(\psi_1) \) and \( L(A_2) = L(\psi_2) \), we construct \( \mathcal{A} = (\Gamma, Q_1 \cup Q_2 \cup \{q_{\text{init}}, q_{\text{init}}, F, \delta\}) \) such that \( \delta(q_{\text{init}}, a) = x, \delta_1(q_1, a) \land x, \delta_2(q_2, a) \). Clearly, \( \mathcal{A} \) is a 1-ATA-\( \text{Rfl} \) since \( A_1, A_2 \) are. It is easy to see that \( L(A) = L(A_1) \cap L(A_2) \). The case when \( \psi = \psi_1 \lor \psi_2 \) follows from the fact that we handle negation and conjunction. The case of formulae of higher modal depth \( \psi_{k+1} = \text{Rat}_r(\text{re}_k) \) is handled by substituting all lower depth formulae in \( \text{re}_k \) with witness variables, and using the inductive hypothesis that there exist 1-ATA-\( \text{Rfl} \) 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-\( \text{Rfl} \) equivalent to \( \psi_{k+1} \). Details in the full version.

4.3 RatMTL meets QkMSO

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

**Proof.**

1. QkMSO \( \subseteq \) RatMTL: Proof is by Induction on the metric depth of the formula. For the base case, consider a formula \( \psi(t_0) = Q_1t_1 \ldots Q_{k-1}t_{k-1} \psi(\downarrow t_0, t_1, \ldots, t_{k-1}) \) of metric depth one. Let \( c_{\text{max}} \) be the maximal constant used in the metric quantifiers \( Q_i \). Let \( R_j(t) \) for \( j \in \text{reg} = \{0, (0, 1), \ldots, c_{\text{max}}, (c_{\text{max}}, \infty)\} \) be fresh monadic predicates. We modify \( \psi(t_0) \) to obtain an untimed MSO formula \( \psi_{\text{reg}}(t_0) \) over the alphabet \( 2^\Sigma \times \{0, 1\}^{\text{reg}^{|}} \times \{0, 1\} \) as follows. Define \( \text{CON}(I, t_i) = \lor \{R_j(t_i) \mid j \leq I\} \). We replace every quantifier \( \exists t_i \in t_0 + I, \phi \) by \( \exists t_i \in (t_0 \leq t_i) \land \text{CON}(I, t_i) \land \phi \). Every quantifier \( \forall t_i \in t_0 \land I, \phi \) is replaced by \( \forall t_i, t_0 \leq t_i \land \text{CON}(I, t_i) \rightarrow \phi \). 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) \( \forall t, t'. [t < t' \land R_j(t) \land R_j(t')] \rightarrow j < j' \) (asserting region order). Note that these are natural properties of region abstraction of time. This gives us the formula \( \psi_{\text{reg}}(t_0) \). It has predicates \( R_j(t) \) for \( j \in \text{reg} \) and free variable \( t_0 \). Being an MSO formula, we can construct a DFA \( \mathcal{A}(\psi_{\text{reg}}(0)) \) for it over the alphabet \( 2^\Sigma \times \{0, 1\}^{\text{reg}^{|}} \). Note that we have substituted 0 for \( t_0 \). This is isomorphic to automaton over the alphabet \( 2^\Sigma \times \text{reg} \). From the construction, it is clear that \( \rho \models \psi(0) \iff \text{reg}(\rho) \in L(\mathcal{A}(\psi_{\text{reg}}(0))) \). By Lemma 12, we then obtain an equivalent RatMTL formula \( \zeta \). It is easy to see that \( L(\psi(0)) = L(\zeta) \). Because \( \psi(0) \) and \( \zeta \) 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 \( \zeta \), 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 \( \zeta_i \). Substituting \( \zeta_i \) for \( w_i \) in \( \zeta \) gives us a formula language equivalent to \( \psi(t_k) \).

2. RatMTL \( \subseteq \) QkMSO : Let \( \varphi \in \text{RatMTL} \). The proof is by induction on the modal depth of \( \varphi \). For the base case, let \( \varphi = \text{Rat}_I(\text{re}) \) where \( \text{re} \) 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 \( \sigma[x : y] \models L(\text{re}) \), 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 \( \psi(t_0) \):

\[
\exists t_{\text{first}} \in t_0 + I. \exists t_{\text{last}} \in t_0 + I. \exists t' \in t_0 + I. [(t' = t_{\text{first}} \lor t' = t_{\text{last}} \lor t_{\text{first}} < t_{\text{last}}) \land \zeta(t_{\text{first}}, t_{\text{last}})]
\]

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 \( t_0, t_{\text{first}}, t_{\text{last}}, t' \) only. Atomic and boolean constructs can be straightforwardly translated. Now let \( \varphi = \text{Rat}_I(\text{re}) \) where \( \text{re} \) 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 \( \varphi_{\text{flat}} \). 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 \( \zeta_i \) there exists a language equivalent time constrained QkMSO formula \( \kappa_i(t_j) \). 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 \( \varphi \). Note that \( \psi(t_0) \in \text{Q4MSO} \) as it only uses time constrained variables \( t_0, t_{\text{first}}, t_{\text{last}}, t' \) which are inductively reused.

### 5. \( \mathbb{C} \oplus \mathbb{D} \)-1-ATA-rfl and Logics

In this section, we show our second main result connecting logics \( \text{FRatMTL}, \text{Q2MSO} \) and a subclass of 1-ATA called conjunctive-disjunctive (abbreviated \( \mathbb{C} \oplus \mathbb{D} \)) 1-ATA with reset-free loops. Let \( A = (\Gamma, Q, \{ q_0, F, \delta \}) \) be a 1-ATA. Let \( Q_\perp = \{ x, q \mid q \in Q \} \) and let \( B(Q_\perp) ::= \text{true} \lor \text{false} [\alpha] \lor [\alpha \land q] \lor \alpha \lor q \), where \( \alpha \in Q_\perp \). \( A \) is said to be a \( \mathbb{C} \oplus \mathbb{D} \) 1-ATA if

1. \( Q \) is partitioned into \( Q_\perp \) and \( Q_\lor \).
2. Let \( q \in Q_\perp \). Transitions \( \delta(q, a) \) can be written as \( D_1 \land D_2 \land \ldots \land D_m \), where any \( D_i \) has one of the following forms. (i) \( D_i = q' \lor B(Q_\perp) \) where \( q' \in Q_\perp \), (ii) \( D_i = x \notin I \lor B(Q_\perp) \).

Each \( D_i \) has at most one free location from \( Q_\perp \), or at most one clock constraint \( x \notin I \).
3. Let \( q \in Q_\lor \). Transitions \( \delta(q, a) \) can be written as \( C_1 \lor C_2 \lor \ldots \lor C_m \), where any \( C_i \) has one of the following forms. (i) \( C_i = q' \land B(Q_\perp) \), where \( q' \in Q_\lor \), (ii) \( C_i = x \in I \land B(Q_\perp) \). Thus, each \( C_i \) has at most one free location from \( Q_\lor \), or at most one clock constraint \( x \in I \).

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

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

---

Note that \( x \notin I \) can be re-written as \( x \in [0, l] \lor x \in (u, \infty) \), where \( [0, l] \cup (u, \infty) \) 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 \( \mathbb{C} \oplus \mathbb{D} \)-1-ATA.
conjunctive-disjunctiveness is denoted $C \oplus D$-$1$-$ATA$-$rfl$. If all the intervals in a $C \oplus D$-$1$-$ATA$ are non-punctual, then it is referred to as a $np-C \oplus D$-$1$-$ATA$.

**Example 15.** We illustrate examples of $1$-$ATA$ violating the $C \oplus D$ 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 $B$ with transitions $\delta(s_0, S_a) = s_0 \lor x > 1$, $\delta(s_0, S_{\neg a}) = s_0 \land x \notin (1, \infty)$, 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 $C \oplus D$ 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 $1$-$ATA$ has an $a$.

(b) Let $S_a, S_{\neg a}$ be as above. The automaton $B$ with $\delta(s_0, S_a) = s_0 \lor s_1$, $\delta(s_2, \Gamma) = x \leq 1$, $\delta(s_0, S_{\neg a}) = s_0 \land s_2$, $\delta(s_1, \Gamma) = x > 1$ with $s_0$ being initial and none of the locations being final satisfies $rfl$ but violates $C \oplus D$. The $C \oplus D$ 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 $B$ with $\delta(s_0, \Gamma) = s_0 \lor s_1$, $\delta(s_1, \Gamma) = s_2 \land s_3$, $\delta(s_1, S_{\neg a}) = \bot$, $\delta(s_2, \Gamma) = (x \leq 1)$, $\delta(s_3, \Gamma) = s_4$, $\delta(s_4, \Gamma) = x > 1$ with $s_0$ being initial and $s_1$ being final satisfies $rfl$ but violates $C \oplus D$. The $C \oplus D$ 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 $C \oplus D$-$1$-$ATA$-$rfl$ meets $FRatMTL$

**Theorem 16.** $C \oplus D$-$1$-$ATA$-$rfl$ are expressively equivalent to $FRatMTL$.

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

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

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

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 $\delta(q, a) = C_1 \lor \cdots \lor C_m$ is such that $C_i$ is either a location or a clock constraint $x \in I_a$, from some location $q$ on an $a$, and $q$ is reachable from $q_0$. Let $rel_a$ 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 $FRat_{\Gamma_a}rel_a$ 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 $rel_{q', a}$ is the regular expression whose language is all words reaching such a $q'$, the formula $FRat_{(0, \infty)}rel_{q', a}(a \land \square \bot)$ sums up all words accepted via $q', a, q_f$. The $\square \bot$ ensures that no further symbols are read, and can be written as $\neg FRat_{(0, \infty)}(a \land \square \bot)$. 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 $C \oplus D$-$1$-$ATA$-$rfl$ $A$, we can construct $\varphi \in FRatMTL$ such that $L(\varphi) = L(A)$. The rest of the proof is very similar to Theorem 13(1) and omitted.
We have defined a new real-time logic an equivalent star-free expression. RatMTL in the automaton are reset-free self-loops. Likewise, PO fragment of full 1-ATA but our quest is for a more traditional logic. The expressive power of 3 variable extension of hybrid logic 1-logic expressively equivalent to full 1-ATA remains open. It may be noted that [5] gave an and also characterized the expressive power of its two variable fragment. The question of a satisfiability and model-checking. We have established a 4-variable property for RatMTL equivalent to a powerful extension of 1-ATA where loops do not have clock-reset. We have also shown that it is expressively tion with a block of FRatMTL Q2MSO

5.2 FRatMTL meets Q2MSO

Theorem 18. FRatMTL is expressively equivalent to Q2MSO.

Proof.

1. Q2MSO ⊆ FRatMTL : We first consider formulae of metric depth one. These have the form ψ(t0) = Q1t1ϕ(Jt0,t1) and ϕ(Jt0,t1) is an MSO formula (bound first order variables t’ in ϕ only have the comparison t’>t0, and there are no free variables other than t0, t1, and hence no metric comparison exists in ϕ). Let reϕ be the regular expression equivalent to ϕ(Jt0,t1). The presence of free variables t0, t1 implies that reϕ ie over the alphabet 2Σ×{0,1}2, where the last two bits are for t0, t1. As seen in the case of QkMSO to RatMTL, t0 is assigned the first position of reϕ since all other variables take up a position to its right. Hence reϕ can be rewritten as (2Σ,1,0)re. Since t1 is assigned a unique position, there is exactly one occurrence of a symbol of the form (2Σ,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 re(α,0,1)re, where α ∈ 2Σ, and re is re, re comprises 0 in the latter part after (α,0,1) and no more symbols after that. See the full version for higher depth case.

2. FRatMTL ⊆ Q2MSO : Proof is similar to RatMTL ⊆ 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 MTL[∪] 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□D-1-ATA are subclasses of 1-ATA and C□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□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 $C\oplus D$-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−TPTL with fixed points had the expressive power of 1-ATA.

References


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


