Automata and logics for finitely varying functions*

Fabrice Chevalier† Deepak D'Souza‡ Raj Mohan M.§
Pavithra Prabhakar¶

Abstract

We extend some of the classical connections between automata and logic due to Büchi [B6̈0] and McNaughton and Papert [MP71], to languages of finitely varying functions or "signals". In particular we introduce a natural class of automata for generating finitely varying functions called ST-NFA's, and show that it coincides in terms of language-definability with a natural monadic second-order logic interpreted over finitely varying functions [Rab02]. We also identify a "counter-free" subclass of ST-NFA's which characterizes the first-order definable languages of finitely varying functions. Our proofs mainly factor through the classical results for word languages. These results have applications in automata characterisations for continuously interpreted real-time logics like Metric Temporal Logic (MTL) [CDP06, CDP07].

1 Introduction

The classical literature contains a rich theory connecting automata and logic over words. Büchi showed that languages definable in monadic second logic (MSO) over words are precisely the class of languages accepted by finite state automata [B6̈0]. Kamp [Kam68] showed that languages definable in Linear-Time Temporal Logic (LTL) were precisely the languages definable in the first-order (FO) fragment of Büchi's MSO. And McNaughton and Papert [MP71] showed that the class of counter-free finite state automata (where a "counter" in an automaton is a loop with at least two hops, each hop being on a common word u) define exactly the FO-definable languages. This last result factors through a characterisation due to Schutzenberger [Sch65] of the class of counter-free languages in terms of star-free regular expressions.

Our aim in this paper is to lift these connections to languages of finitely varying functions over the non-negative reals. These functions are finitely varying in that they have only a finite number of discontinuities in any bounded interval of time. Such functions, which are often called "signals" in the literature, are of interest to the computer science community as they model the behaviour of timed and hybrid systems [AD94, ACH$^+$95]. For example, non-zeno timed words [AD94] are special kinds of signals.

We first introduce a class of automata called ST-NFA's that run over signals and hence accept languages of signals. We should point out here that unlike timed automata

*This work was partly supported by Indo-French project P2R Timed-DISCOVERI.
†LSV, ENS de Cachan, France fabrice.chevalier@lsv.ens-cachan.fr
‡CSA, IISc, Bangalore, India deepakd@csa.iisc.ernet.in
§CSA, IISc, Bangalore, India raj@csa.iisc.ernet.in
¶Dept. of Comp. Sci., UIUC, USA pprabha2@uiuc.edu

we are interested in formalisms without a "metric" or operators that measure time distance. As a consequence these languages are essentially "untimed" in that they can be characterised as the set of all possible "timings" of a (regular) language of classical words. We then consider a natural monadic second-order logic introduced earlier by Rabinovich [Rab02], and called here $\mathsf{MSO}^{\mathsf{s}}$, which is interpreted over signals, and in which the second-order quantification is restricted to subsets of non-negative reals whose characteristic functions are finitely-varying. We show that the class of signal languages defined by sentences in this logic is precisely the class of signal languages defined by ST-NFA's. This gives an automata-theoretic proof of a similar result obtained in [Rab02] using logical techniques.

Next, along the lines of the Schutzenberger and McNaughton-Papert results, we identify a counter-free subclass of ST-NFA's and show that they precisely characterise the class of signal languages definable by the first-order fragment FO^{s} of $\mathsf{MSO}^{\mathsf{s}}$. The notion of a counter in an ST-NFA is similar to the classical one, except that we require the ST-NFA to be "canonical" in a certain sense. Our proof of this result factors transparently through the afore-mentioned results of Schutzenberger, McNaughton-Papert, and Kamp for word languages. The main difficulty, in a series of steps we perform, is to translate an LTL formula θ into one interpreted over signals, which accepts precisely the timings of the models of θ. As a minor by-product we re-prove the expressive completeness of LTL interpreted over signals (i.e. its expressiveness coincides with FO^{s} over signals). This result also follows from Kamp's result showing the expressive completeness of LTL over reals [Kam68]. Nonetheless, our proof gives a more accessible proof of this result, since it uses only Kamp's result for classical words, for which there are simpler proofs in the literature (see [Wil99]).

Turning now to more details on related work, as already mentioned this paper builds on the classical results due to Büchi [B̈60], Schutzenberger [Sch65], McNaughton and Papert [MP71], and Kamp [Kam68] for word languages. The work of Rabinovich contains many relevant results on signal languages. Rabinovich and Trakhtenbrot [RT97] introduce automata similar to ST-NFA's called signal acceptors. In [Rab02] Rabinovich shows how to translate an MSO sentence φ to a $\mathsf{MSO}^{\mathsf{s}}$ sentence that accepts precisely the timings of φ, and vice versa. This leads to a proof of the claim in [RT97] that signal languages definable by signal acceptors and $\mathsf{MSO}^{\mathsf{s}}$ coincide. In contrast our equivalence of ST-NFA's and $\mathsf{MSO}^{\mathsf{s}}$ uses an automata-theoretic argument similar to the proof of Büchi's result (see [Tho90]), and helps us identify the counter-free fragment. In [Rab00] Rabinovich also shows a star-free regular expression characterisation of FO^{s}-definable signal languages, along the lines of McNaughton and Papert [MP71].

Though we are mainly concerned with expressiveness in this work, there are a number of related decidability results in the literature. Shellah [She75] showed that MSO over reals with second-order quantification over arbitrary subsets of reals is undecidable. Rabin [Rab69] shows that MSO over reals, with second-order quantification over subsets which are essentially countable unions of closed sets, is decidable. The decidability of $\mathsf{MSO}^{\mathsf{s}}$ follows from this result.

Preliminary versions of the basic results in this paper appeared in [CDP06, CDP07] where they were used to obtain logical characterisations of versions of timed automata with a continuous interpretation, as well as a counter-free timed automata characterisation of several real-time temporal logics, including MTL [Koy90, AFH96], in their continuous semantics.

2 Preliminaries

For an alphabet A, we use A^* to denote the set of finite words over A. For a word w in A^*, we use $|w|$ to denote its length. The set of non-negative reals and rationals will be denoted by $\mathbb{R}_{\geq 0}$ and $\mathbb{Q}_{\geq 0}$ respectively. We will deal with intervals of non-negative reals, i.e. convex subsets of $\mathbb{R}_{\geq 0}$, and denote by $\mathcal{I}_{\mathbb{R}_{\geq 0}}$ the set of such intervals with end-points in $\mathbb{R}_{\geq 0} \cup \{\infty\}$. Two intervals I and J will be called *adjacent* if $I \cap J = \emptyset$ and $I \cup J$ is an interval.

Let A be a finite alphabet and let $f : [0, r] \to A$ be a function, where $r \in \mathbb{R}_{\geq 0}$. We use $dur(f)$ to denote the duration of f, which in this case is r. A point $t \in (0, r)$ is a point of *continuity* of f if there exists $\epsilon > 0$ such that f is constant in the interval $(t - \epsilon, t + \epsilon)$. All other points in $[0, r]$ are points of *discontinuity* of f. We say f is *finitely varying* if it has only a finite number of discontinuities in its domain. We will refer to such finitely varying functions as *signals* over A, and denote the set of signals over A by $Sig(A)$.

An *interval representation* for a signal $\sigma : [0, r] \to A$ is a sequence of the form $(a_0, I_0) \cdots (a_n, I_n)$, with $a_i \in A$ and $I_i \in \mathbb{I}_{\mathbb{R}_{\geq 0}}$, satisfying the conditions that the union of the intervals is $[0, r]$, each I_i and I_{i+1} are adjacent, and for each i, σ is constant and equal to a_i in the interval I_i. We can obtain a *canonical* interval representation for σ by putting each point of discontinuity in a singular interval by itself. Thus the above interval representation for σ is canonical if n is even, for each even i the interval I_i is singular (i.e. of the form $[t, t]$), and for no even i such that $0 < i < n$ is $a_{i-1} = a_i = a_{i+1}$.

A canonical interval representation for a function gives us a canonical way of "untiming" the signal: thus if $(a_0, I_0) \cdots (a_{2n}, I_{2n})$ is the canonical interval representation for a signal σ, then we define $untiming(\sigma)$ to be the string $a_0 \cdots a_{2n}$ in A^*. The untiming thus captures explicitly the value of the function at its points of discontinuity and the open intervals between them. Figure 1 shows a signal over the alphabet $\{a, b, c\}$, and its untiming. Note that strings which represent the untiming of a signal will always be of odd length, and for no even position i will the letters at positions $i-1$, i, and $i+1$ be the same. We refer to words in A^* which satisfy these two conditions as *canonical* words over A. We denote the set of canonical words over A by $Can(A)$.

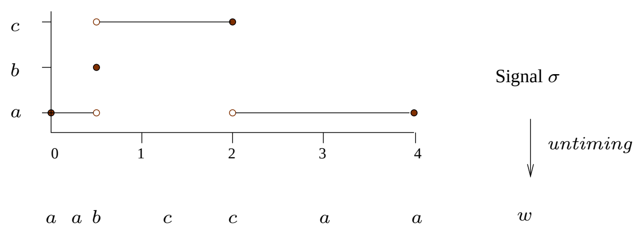


Figure 1: A signal and its untiming

A canonical word w can be "timed" to get a signal in a natural way: thus a signal σ is in $timing(w)$ if $untiming(\sigma) = w$. We extend the definition of $timing$ and $untiming$ to languages of signals and words in the expected way.

Finally, we say a subset X of $\mathbb{R}_{\geq 0}$ is *finitely varying* if its characteristic function $f_X : \mathbb{R}_{\geq 0} \to \{0, 1\}$ given by $f_X(t) = 0$ if $t \in X$ and 1 otherwise, is finitely varying (in the sense defined above) in every interval of the form $[0, r]$ with $r \in \mathbb{R}_{\geq 0}$.

3 Automata over signals – ST-NFA's

In this section we introduce a variant of classical word automata called ST-NFA's which are a convenient formalism for generating signals.

We recall that a non-deterministic finite state automaton (NFA) over an alphabet A is a structure $\mathcal{A} = (Q, S, \delta, F)$, where Q is a finite set of states, S is the set of initial states, $\delta \subseteq Q \times A \times Q$ is the transition relation, and $F \subseteq Q$ is the set of final states. A run of $\mathcal{A}$ on a word $w = a_0 \cdots a_n \in A^*$ is a sequence of states $q_0, \ldots, q_{n+1}$ such that $q_0 \in S$, and $(q_i, a_i, q_{i+1}) \in \delta$ for each $i \leq n$. The run is accepting if $q_{n+1} \in F$. The symbolic language accepted by $\mathcal{A}$, denoted $L(\mathcal{A})$, is the set of words in A^* over which $\mathcal{A}$ has an accepting run. Languages accepted by NFA's are called *regular* languages. We say the NFA $\mathcal{A}$ is *deterministic* (and call it a DFA) if the set of start states is a singleton, and the transition relation δ is a function from $Q \times A$ to Q.

A *state-transition-labeled* NFA*'s* (ST-NFA$'s$ for short) over A is a structure $\mathcal{A} = (Q, S, \delta, F, l)$ similar to an NFA over A, except that $l : Q \to A$ labels states with letters from A. As a recogniser of words, the ST-NFA $\mathcal{A}$ accepts strings of the form $A(AA)^*$. A run of $\mathcal{A}$ on a string $w = a_0 a_1 \cdots a_{2n}$ in $A(AA)^*$, is a sequence of states $q_0, \ldots, q_{n+1}$ satisfying $q_0 \in S$, $(q_i, a_{2i}, q_{i+1}) \in \delta$ for $i \in \{0, \cdots, n\}$ and $l(q_i) = a_{2i-1}$ for each $i \in \{1, \ldots, n\}$; it is accepting if $q_{n+1} \in F$. We define the word language accepted by $\mathcal{A}$, denoted $L(\mathcal{A})$, to be the set of strings $w \in A^*$ on which $\mathcal{A}$ has an accepting run.

An ST-NFA $\mathcal{A}$ also generates signals in a natural way: we begin by taking a transition emanating from the start state, emitting its label, and then spend time at the resulting state emitting its label all the while, before taking a transition again; and so on till we choose to stop at a final state. The language of signals generated by an ST-NFA $\mathcal{A}$ is defined to be $timing(L(\mathcal{A}))$, and will be denoted by $S(\mathcal{A})$.

An example ST-NFA is shown in Figure 2. We will use the convention that start states are indicated by sourceless incoming arrows, and final states are indicated by double circles. The automaton accepts the signal shown in Figure 1.

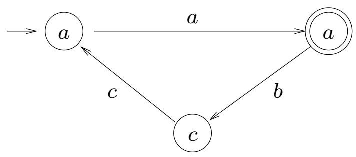


Figure 2: Example ST-NFA.

Let us call a state in an ST-NFA "originating" if it has no incoming transitions, and "terminating" if it has no outgoing transitions. We say a transition $p \xrightarrow{a} q$ in an ST-NFA $\mathcal{A} = (Q, S, \to, F, l)$ is *non-canonical* if $l(p) = l(q) = a$, *except* for the case when p is originating or q is terminating. We say the ST-NFA $\mathcal{A}$ is *canonical* if it has no non-canonical transitions. Clearly for a canonical ST-NFA $\mathcal{A}$ over A we have $L(\mathcal{A}) \subseteq Can(A)$.

We now observe the following properties of the word languages accepted by ST-NFA's.

Lemma 3.1 *1. The class of word languages accepted by* ST-NFA*'s over an alphabet A is precisely the class of regular subsets of $A(AA)^*$.*

2. The class of word languages accepted by ST-NFA*'s over an alphabet A is closed under union, intersection, and also complementation with respect to $A(AA)^*$.*

Proof

1. The word language accepted by an ST-NFA over $\mathcal{A}$ is clearly a subset of $A(AA)^*$. It is also regular, this can be observed by replacing each state q of $\mathcal{A}$ by a pair of states q_1 and q_2, with the incoming edges of q now incident on q_1 and the outgoing edges of q emanating from q_2 with an additional transition $(q_1, l(q), q_2)$ to account for the label on q. The start states are the q_2's corresponding to the start states q of $\mathcal{A}$, and the final states are those corresponding to the q_1's such that q is a final state of $\mathcal{A}$. The resulting automaton is an NFA which accepts the same word language as $\mathcal{A}$.

   Conversely, given an NFA $\mathcal{A}$ which accepts a subset of $A(AA)^*$, we can give an ST-NFA $\mathcal{A}'$ whose word language is same as that of $\mathcal{A}$. Let us assume that $\mathcal{A}$ does not contain any dead states. The states of $\mathcal{A}'$ correspond to the transitions of $\mathcal{A}$ and an initial state s. The final states are those corresponding to the transitions going into the final states of $\mathcal{A}$. Each state is labelled by the label of the transition to which it corresponds. There is a transition (e_1, a, e_2) in $\mathcal{A}'$ if there is a transition (q, a, r) in $\mathcal{A}$, and $e_1 = (p, b, q)$ and $e_2 = (r, c, t)$ for some p and t. There is also a transition (s, a, e_2) in $\mathcal{A}'$ if there is a transition (p, a, q) out of an initial state of $\mathcal{A}$ and e_2 is a transition out of q in $\mathcal{A}$. It is easy to see that the word languages of $\mathcal{A}$ and $\mathcal{A}'$ are the same.

2. Closure under union and intersection follow from above. For complementation, we observe that $A(AA)^* - L(\mathcal{A}) = (A^* - L(\mathcal{A})) \cap A(AA)^*$, which is clearly a regular subset of $A(AA)^*$. From the above characterisation, it is ST-NFA-definable.

□

Here are some properties of the signal language accepted by ST-NFA's which will be useful in the sequel.

Proposition 3.1 *1. If w and w' are canonical words over A then $w = w'$ iff $timing(w) \cap timing(w') \neq \emptyset$.*

2. $timing(Can(A)) = Sig(A)$.

3. The word language $Can(A)$ is ST-NFA-*definable.*

Proof Parts 1 and 2 follow easily from the definitions. For part 3, the required ST-NFA $\mathcal{A}_{Can}$ for the alphabet $\{a, b\}$ is shown in Figure 3 below. □

Lemma 3.2 *For every* ST-NFA $\mathcal{A}$ *over A there is a signal language equivalent* canonical ST-NFA $\mathcal{A}'$ *(i.e. $S(\mathcal{A}') = S(\mathcal{A})$).*

Proof Let $\mathcal{A} = (Q, S, \rightarrow, F, l)$. We modify $\mathcal{A}$ as follows:

1. First we transform $\mathcal{A}$ to $\mathcal{A}'$ by making the start states originating and final states terminating. This can be done by adding a new start state s' and a new final state f', and adding transitions $s' \xrightarrow{a} p$ for each transition $s \xrightarrow{a} p$ in $\mathcal{A}$ with $s \in S$, and transitions $p \xrightarrow{a} f'$ for each transition $p \xrightarrow{a} f$ in $\mathcal{A}$ with $f \in F$.

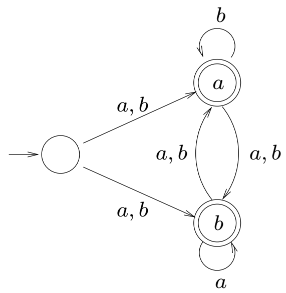


Figure 3: ST-NFA accepting the word language $Can(\{a, b\})$.

2. Next, we transform $\mathcal{A}'$ to $\mathcal{A}''$ as follows: Pick a non-canonical transition $p \xrightarrow{a} q$ and a transition $r \xrightarrow{b} p$, and add the transition $r \xrightarrow{b} q$. Repeat this till no more edges can be added.

3. Now we drop all non-canonical edges in $\mathcal{A}''$ to obtain the required canonical ST-NFA $\mathcal{B}$.

Step 1 clearly preserves the word (and hence signal) language of $\mathcal{A}$. Step 2 clearly preserves the signal language of $\mathcal{A}'$. Step 3 also preserves the signal language of $\mathcal{A}''$, since any signal σ generated by $\mathcal{A}''$ using a run of non-canonical edges can be simulated by using a single canonical edge in $\mathcal{A}''$. □

Using these observations we can now prove some closure properties of the class of ST-NFA-definable signal languages.

Lemma 3.3 *The class of signal languages definable by* ST-NFA*'s over A are closed under union, interesection, and complement.*

Proof Closure under union is direct. For complementation, let $\mathcal{A}$ be an ST-NFA over A. Let $\mathcal{A}'$ be a signal-equivalent canonical ST-NFA. Then $Sig(A) - S(\mathcal{A}) = Sig(A) - S(\mathcal{A}') = timing(Can(A) - L(\mathcal{A}')) = S(\mathcal{A}'')$ for some ST-NFA $\mathcal{A}''$ by closure properties of ST-NFA-definable word languages. The closure under intersection follows from that of union and complementation. □

4 Equivalence of ST-NFA's and $\mathsf{MSO}^{\mathsf{s}}$

In this section we introduce a natural monadic second-order logic interpreted over signals and show that the class of signal languages it defines coincides with the class of signal languages definable by ST-NFA's.

In the logics to follow we assume a countable supply of first-order variables and second-order variables. For an alphabet A, the syntax of monadic second order logic over A, denoted $\mathsf{MSO}^{\mathsf{s}}(A)$, is given by:

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

where $a \in A$, x and y are first-order variables and X is a second-order variable.

We interpret a formula φ of the logic over a signal σ in $Sig(A)$, along with an interpretation $\mathbb{I}$ with respect to σ, which assigns to each first-order variable a value in $[0, dur(\sigma)]$, and to each set variable, a *finitely-varying* subset of $[0, dur(\sigma)]$. We use $X \subseteq_{fv} Y$ to denote that X is a finitely-varying subset of Y. For an interpretation $\mathbb{I}$, we use the notation $\mathbb{I}[t/x]$ to denote the interpretation which sends x to t and agrees with $\mathbb{I}$ on all other variables. Similarly, $\mathbb{I}[B/X]$ denotes the modification of $\mathbb{I}$ which maps the set variable X to a subset B of $\mathbb{R}_{\geq 0}$, and the rest to the same as that mapped by $\mathbb{I}$. We also use the notation $[t/x]$ to denote an interpretation which sends x to t when the rest of the interpretation is irrelevant.

Given a formula $\varphi \in \mathsf{MSO}^{\mathsf{s}}(A)$, $\sigma \in Sig(A)$, and an interpretation $\mathbb{I}$ with respect to σ to the variables in φ, the satisfaction relation $\sigma, \mathbb{I} \models \varphi$, is defined inductively as:

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

For a sentence φ (a formula without free variables) in $\mathsf{MSO}^{\mathsf{s}}(A)$, the interpretation does not play any role, and we write the satisfaction relation $\sigma, \mathbb{I} \models \varphi$ as simply $\sigma \models \varphi$. We define the language of signals defined by φ to be $S(\varphi) = \{\sigma \in Sig(A) \mid \sigma \models \varphi\}$.

As an example, the formula

$$\varphi_{cont} = \exists y \exists z (y < x \wedge x < z \wedge \bigvee_{a \in A} \forall u (y < u \wedge u < z \Rightarrow Q_a(u)))$$

asserts that the point x is a point of continuity. The formula $\varphi_{disc}(x) = \neg\varphi_{cont}(x)$ asserts that x is a point of discontinuity.

We denote by $\mathsf{FO}^{\mathsf{s}}(A)$ the first-order fragment of $\mathsf{MSO}^{\mathsf{s}}(A)$ obtained by disallowing second-order quantification and atomic formulas of the form $x \in X$.

Theorem 4.1 *A signal language over an alphabet A is definable by a* $\mathsf{MSO}^{\mathsf{s}}(A)$ *sentence iff it is definable by an* ST-NFA *over A.*

We prove this theorem in the rest of this section. The proof proceeds in a similar manner to the proof of Büchi's MSO characterization of classical automata [B6̈0] (see [Tho90]).

We first show how to go from a formula in $\mathsf{MSO}^{\mathsf{s}}(A)$ to an equivalent ST-NFA over A. We will represent models of formulas with free variables in them, as functions with the interpretations built into them. We assume an ordering on the countable set of first-order variables given by $x_1, x_2, \ldots$, and similarly $X_1, X_2, \ldots$ for the set variables. For a formula φ with first-order free variables among $U = \{x_{i_1}, \cdots, x_{i_m}\}$ and second-order free variables among $V = \{X_{j_1}, \ldots, X_{j_n}\}$ (in order), we represent a signal σ and an interpretation $\mathbb{I}$ as a signal $\sigma_{\mathbb{I}}^{U,V} : [0, dur(f)] \rightarrow A \times \{0,1\}^{m+n}$ given by $\sigma_{\mathbb{I}}^{U,V}(t) = (f(t), b_1, \cdots, b_{m+n})$, where for $k \in \{1, \ldots, m\}$, $b_k = 1$ iff $\mathbb{I}(x_{i_k}) = t$, and for $k \in \{m+1, \ldots, n\}$, $b_k = 1$ iff $t \in \mathbb{I}(X_{j_{k-m}})$. Thus for a formula φ with free variables in (U, V) we have the notion of a (U, V)-model of φ.

Proposition 4.1 *Let A be a finite alphabet and let (U, V) be a set of first and second-order variables. Then we can construct an* ST-NFA $\mathcal{A}_{valid}^{U,V}$ *which accepts precisely the set of signals over $A \times \{0,1\}^{|U|+|V|}$ which represent valid (U, V)-models over A.*

Proof Figure 4 shows an ST-NFA which accepts the valid models over the alphabet $A = \{a, b\}$ and the first-order variables $\{x, y\}$. $\square$

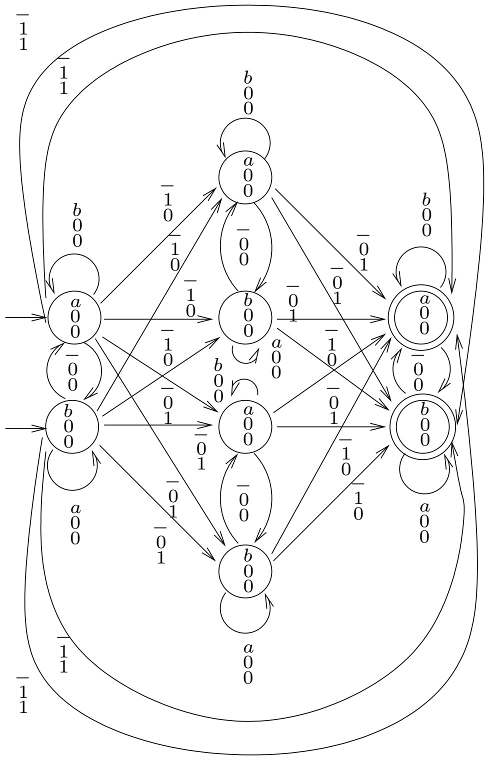


Figure 4: ST-NFA accepting valid $(\{x, y\}, \emptyset)$-models over the alphabet $\{a, b\}$
.

Proposition 4.2 *Let φ be an $\mathsf{MSO^s}(A)$ formula with free variables (U, V) and let $\mathcal{A}$ be an* ST-NFA *accepting the (U, V)-models of φ. Then for any set of variables (U', V') such that $U \subseteq U'$ and $V \subseteq V'$, we can construct an* ST-NFA $\mathcal{A}'$ *which accepts precisely the (U', V')-models of φ.*

Proof Let us consider the case when $U' = U \cup \{x\}$ and $V' = V$. Consider ST-NFA $\mathcal{A}''$ which has the same set of states as $\mathcal{A}$, with the same initial and final states. The labeling of states in $\mathcal{A}''$ is same as that of $\mathcal{A}$ except that they are extended with a 0 corresponding to x. For every transition in $\mathcal{A}$, there are two transitions in $\mathcal{A}''$ with the same start and target states, and the label extended with 0 in one and 1 in the other. The automaton $\mathcal{A}'$ is then obtained by taking the intersection of $\mathcal{A}''$ with $\mathcal{A}_{valid}^{(U',V)}$.

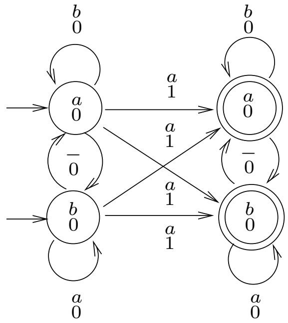
For the case when $U' = U$ and $V' = V \cup \{X\}$, we construct $\mathcal{A}''$ similar to the above, except that now corresponding to each state we have two states, one labeled with an extension of the original label with a 1 corresponding to X, and the other labeled with extension by 0. Hence corresponding to a transition in $\mathcal{A}$ there are four corresponding to two choices for the start state and two for the target states. In this case we can avoid the intersection with $\mathcal{A}_{valid}^{(U,V')}$.

This construction can be easily generalized for any U' and V'. □

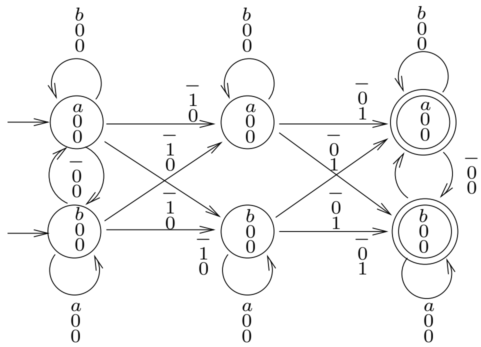
Lemma 4.1 *Let φ be an $\mathsf{MSO^s}(A)$ formula and let (U, V) be the set of free variables in it. Then we can construct an* ST-NFA $\mathcal{A}_\varphi^{U,V}$ *which accepts precisely the (U, V)-models of φ.*

Proof We construct the ST-NFA $\mathcal{A}_\varphi^{U,V}$ by induction on the structure of φ.

1. $\varphi = Q_a(x)$: The automaton $\mathcal{A}_\varphi^{\{x\}}$ is:



2. $\varphi = x < y$: The automaton $\mathcal{A}_\varphi^{\{x,y\},\emptyset}$ (assuming x occurs before y in the variable ordering) is:



3. For the case $\varphi = x \in X$, the automaton $\mathcal{A}_\varphi^{\{x\},\{X\}}$ is defined similarly.

4. $\varphi = \neg\psi$: Let $\mathcal{A}_\psi^{U,V}$ be the automaton for ψ, where (U, V) is the set of free variables in ψ. Then $\mathcal{A}_\varphi^{U,V}$ is the intersection of $\mathcal{A}_{valid}^{U,V}$ with the ST-NFA that recognizes the complement of the signal language of $\mathcal{A}_\psi^{U,V}$ (cf. Lemma 3.3).

5. $\varphi = \psi \vee \nu$: Let $\mathcal{A}_\psi^{U_1,V_1}$ be the ST-NFA for ψ, where (U_1, V_1) is the set of free variables in ψ, and let $\mathcal{A}_\nu^{U_2,V_2}$ be the ST-NFA for ν, where (U_2, V_2) is the set of free variables in ν. Let $U = U_1 \cup U_2$ and $V = (V_1 \cup V_2)$. By Prop. 4.2 we obtain ST-NFA's $\mathcal{A}_\psi^{U,V}$ and $\mathcal{A}_\nu^{U,V}$. Then $\mathcal{A}_\varphi^{U,V}$ is the ST-NFA that accepts the union of the signal languages accepted by $\mathcal{A}_\psi^{U,V}$ and $\mathcal{A}_\nu^{U,V}$.

6. $\varphi = \exists x\psi$: Let (U', V') be the set of free variables in ψ, so that $U = U' - \{x\}$ and $V = V'$. Let $\mathcal{A}_\psi^{U',V'}$ be an ST-NFA for ψ. Now we simply project away the component corresponding to x in the symbols on the transitions of $\mathcal{A}_\psi^{U',V'}$ to obtain the required ST-NFA $\mathcal{A}_\varphi^{U,V}$.

7. $\varphi = \exists X \psi$: Let (U', V') be the set of free variables in ψ, so that $U = U'$ and $V = V' - \{X\}$. Let $\mathcal{A}_\psi^{U',V'}$ be an ST-NFA for ψ. Again we simply project away the component corresponding to X in the symbols on the transitions of $\mathcal{A}_\psi^{U',V'}$ to obtain the required counter-free ST-NFA $\mathcal{A}_\varphi^{U,V}$.

□

From the above lemma it now follows that for a sentence $\varphi \in \mathsf{MSO^s}(A)$ we have an ST-NFA $\mathcal{A}_\varphi$ over A such that $S(\varphi) = S(\mathcal{A}_\varphi)$.

We now prove the converse direction of Theorem 4.1. Let $\mathcal{A} = (Q, S, \rightarrow, F, l)$ be an ST-NFA over A. Without loss of generality we assume that $\mathcal{A}$ is canonical. We give an $\mathsf{MSO^s}(A)$ sentence $\varphi_\mathcal{A}$ such that $S(\mathcal{A}) = S(\varphi_\mathcal{A})$. The sentence $\varphi_\mathcal{A}$ describes the existence of an accepting run of the automaton on a given signal. Let $\{e_i = p_i \xrightarrow{a_i} q_i \mid i = 1, \ldots, m\}$ be the set of transitions in $\mathcal{A}$. The second order variables $X_1, \cdots, X_m$ will be used to capture the points in the signal at which the transitions $e_1, \cdots, e_m$ are taken respectively. Note that since we are assuming $\mathcal{A}$ is canonical, the union of the X_i's must correspond exactly to the points of discontinuities in the given signal. We will use the abbreviation $conseq(x, y, X)$ to mean that x and y are "consecutive" points in the set X, and define it to be:

$$conseq(x, y, X) = x \in X \wedge y \in X \wedge \neg\exists z(x < z \wedge z < y \wedge z \in X).$$

We also use $\mathit{first}(x)$ as an abbreviation for $\neg\exists y(y < x)$ and $last(x)$ as an abbreviation for $\neg\exists y(x < y)$.

The formula $\varphi_\mathcal{A}$ is given below. We assume that i and j range over $0, \ldots, m$.

$$\begin{array}{ll}\exists X_1 \cdots \exists X_n \exists X (\forall x(& (x \in X \iff \bigvee_i x \in X_i) \wedge \\ & (\bigwedge_{i \neq j}(x \in X_i \Rightarrow \neg\, x \in X_j)) \wedge \\ & (x \in X \iff disc(x)) \wedge \\ & (\mathit{first}(x) \Rightarrow \bigvee_{i:\, p_i \in S} x \in X_i) \wedge \\ & (last(x) \Rightarrow \bigvee_{i:\, q_i \in F} x \in X_i) \wedge \\ & (\bigwedge_i (x \in X_i \Rightarrow (Q_{a_i}(x) \wedge ((\exists y(conseq(x, y, X))) \Rightarrow \\ & \quad \forall z((x < z \wedge z < y) \Rightarrow Q_{l(q_i)}(z)))))))).\end{array}$$

This completes the proof of Theorem 4.1.

Before we close this section we observe that the version of $\mathsf{MSO^s}$, called *weak* $\mathsf{MSO^s}$, in which we restrict the second-order quantification to *finite* subsets of the domain of the signal (rather than finitely-varying subsets) is as expressive as the version we have defined. The justification is as follows. We note that the clause $x \in X \iff disc(x)$ forces the second-order variables X and X_i's to be interpreted as *finite* subsets of the domain since the signal model has only finitely many discontinuities. Hence quantification over finite subsets suffices to capture the ST-NFA-definable signal languages. Further, signal languages definable by second-order quantification restricted to finite subsets are clearly ST-NFA-definable (by an argument similar to the one above, where we allow rows corresponding to second-order variables to have 1's only on transition (and not state) labels). Hence the expressiveness of the two variants coincide with ST-NFA-definable signal languages.

Corollary 4.1 *The class of signal languages definable in* $\mathsf{MSO^s}(A)$ *and weak* $\mathsf{MSO^s}(A)$ *coincide.*

5 Counter-free signal languages

In this section we introduce a counter-free version of signal languages which will be shown in the next section to characterize FO^{s}-definable signal languages.

We recall that a *counter* in an NFA $\mathcal{A}$ is a sequence of distinct states $q_0, \ldots, q_n$ with $n \geq 1$, along with a word $u \in A^*$, such that there is a path labeled u in $\mathcal{A}$ from q_i to q_{i+1} (for each $i \in \{0, \ldots, n-1\}$) and from q_n to q_0. An NFA is said to be *counter-free* if it does not contain a counter. A regular language is said to be *counter-free* if there exists a counter-free NFA for it.

We now define the counter-free version of ST-NFA's. A counter in an ST-NFA is similar to one in an NFA, except that by the "label" of a path in the automaton we mean the sequence of alternating transition and state labels along the path. The label of the path $q_0 \overset{a_0}{\to} q_1 \overset{a_1}{\to} \cdots q_n \overset{a_n}{\to} q_{n+1}$ is $a_0 l(q_1) a_1 \cdots l(q_n) a_n l(q_{n+1})$ (we ignore the label of the first state in the path, but count the label of the last state in the path). Thus a *counter* in an ST-NFA $\mathcal{A}$ is a sequence of distinct states $q_0, \ldots, q_n$ with $n \geq 1$, along with a word $u \in A^*$, such that there is a path labeled u in $\mathcal{A}$ from q_i to q_{i+1} (for each $i \in \{0, \ldots, n-1\}$) and from q_n to q_0. We say an ST-NFA is *counter-free* if it does not contain a counter. We say a signal language is *counter-free* if it is definable by counter-free *canonical* ST-NFA. The canonicity clause is important as otherwise we can give counter-free ST-NFA's for signal languages we would like to consider as not being counter-free. Figure 5 below shows a non-canonical counter-free ST-NFA and its equivalent canonical version which contains a counter.

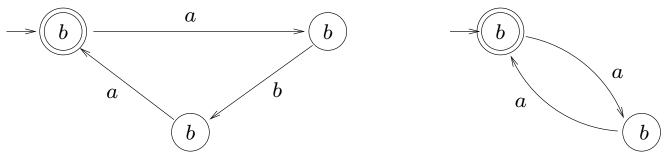


Figure 5: Example to show how canonicalization may introduce counters.

We will show in the next section that the class of first-order definable signal languages coincide with the class of counter-free signal languages. Here is a property of counter-free signal languages that we make use of there.

Lemma 5.1 *The class of counter-free signal languages over an alphabet A is closed under union, intersection, and complementation.*

Proof Let F_1 and F_2 be two counter-free signal languages over the alphabet A. Let L_1 and L_2 be the word languages of their defining canonical counter-free ST-NFA's. We can convince ourselves that $F_1 \cup F_2 = \mathit{timing}(L_1 \cup L_2)$, $F_1 \cap F_2 = \mathit{timing}(L_1 \cap L_2)$ and $\mathit{Sig}(A) - F_1 = \mathit{timing}(\mathit{Can}(A) - L_1)$. Hence to show the closure of counter-free signal languages, it suffices to show the closure of the word languages defined by canonical counter-free ST-NFA's.

We first give a characterization of the word languages of canonical counter-free ST-NFA's in terms of the finite automata which accept them. A language L is the word language of a canonical counter-free ST-NFA iff it is accepted by an NFA $\mathcal{B}$ with the following properties:

- The underlying structure of $\mathcal{B}$ is bipartite.

- The set of initial states and the set of final states belong to different partitions.
- The NFA does not contain an *even-counter* in that there is no counter with the repeating word u of the counter having an even length, i.e, $u \in (AA)^*$.
- The language accepted by $\mathcal{B}$ consists of only canonical words.

Let us denote the partitions of states in $\mathcal{B}$ as B_1 and B_2, with the initial states belonging to the former and the final states to the latter. The above observation follows easily from the constructions of Lemma 3.1.

Now we show that the class of word languages accepted by NFA's with the above properties is closed under union, intersection and complementation. The closure under union is direct. To show the closure under intersection we argue that the above properties are preserved by the standard product construction for intersection. Let $\mathcal{B}$ and $\mathcal{C}$ be two automata with the above properties. In the product automaton $\mathcal{D}$ accepting the intersection of their word languages, the only reachable states are $D_1 = B_1 \cup C_1$ and $D_2 = B_2 \cup C_2$, with D_1 and D_2 being the partitions. It remains to be shown that $\mathcal{D}$ does not have a counter of the form described above. Suppose $\mathcal{D}$ has such a counter with the sequence $(s_1, t_1) \cdots (s_n, t_n)$. Then it is not the case that all the s_i's are same and all the t_j's are same, otherwise the sequence is not a counter. Assume all the s_i's are not same. Then there is a sequence of states $s_k, s_{k+1}, \cdots s_l$ which forms a counter in B, a contradiction.

To show closure under complementation, we first determinize the automaton. The standard subset construction for determinization on an automaton $\mathcal{B}$ preserves the above properties. First the only reachable states in the determinized automaton are the subsets of B_1 and the subsets of B_2, and the subsets of B_1 form one partition and the subsets of B_2 the other. So the first two properties are satisfied. In general the subset construction does not introduce counters. To show that even-counters are not introduced, first rename the labels on the transitions going from B_2 to B_1 by their primed version (which are assumed to be disjoint from A). This ensures that there are no counters in $\mathcal{B}$ in the classical sense. Now determinize the automaton, and replace the primed symbols by their unprimed versions. If there is an even-counter in the final automaton, then the same sequence of states and the same word give a counter (in fact an even-counter) in the automaton before the replacement of the primed symbols by unprimed ones.

To complement the automaton we take the complement of the determinized automaton and take its intersection with $\mathcal{B}_{Can}$, which accepts all canonical words. To see that there exists a $\mathcal{B}_{can}$ with the above properties, it is enough to show that there is a canonical counter-free ST-NFA (S, S_0, δ, F, l) accepting $Can(A)$. It is given by $S = S_0 = F = \{s_a \mid a \in A\}$, $\delta = \{(s_a, a, s_b) \mid b \in A - \{a\}\}$ and $l(s_a) = a$. Hence the word languages with automata having the above properties are closed under the boolean operations and therefore are the class of counter-free signal languages. $\square$

6 Counter-free characterisation of FO signal languages

In this section our aim is to show that FO^{s}-definable signal languages and counter-free signal languages coincide.

Theorem 6.1 *The class of* $\mathsf{FO}^{\mathsf{s}}(A)$*-definable signal languages is precisely the class of counter-free signal languages over* A*.*

The rest of this section is devoted to a proof of this theorem. Our proof will factor through some classical results connecting counter-free languages and temporal logics. We recall briefly the logic LTL and its two interpretations, one over discrete words and the other over signals. The syntax of $\mathsf{LTL}(A)$ is given by:

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

where $a \in A$. The logic is interpreted over words in A^*, with the following semantics. Given a word $w = a_0 \cdots a_n$ in A^* and a position $i \in \{0, \ldots, n\}$, we say $w, i \models a$ iff $a_i = a$; and $w, i \models \theta U \eta$ iff there exists j such that $i < j \leq n$, $w, j \models \eta$ and for all k such that $i < k < j$, $w, k \models \theta$. The "since" operator S is defined in a symmetric way to U in the past, and the boolean operators in the usual way. We denote by $L(\theta)$ the set $\{w \in A^* \mid w, 0 \models \theta\}$.

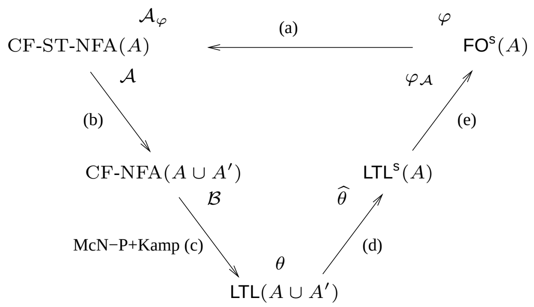
The logic LTL can also be interpreted over functions as done in [Kam68]. Here we restrict the models to finitely-varying functions in $Sig(A)$, and we denote this logic by $\mathsf{LTL}^{\mathsf{s}}(A)$. Given a signal $\sigma \in Sig(A)$, $t \in [0, dur(\sigma)]$ and $\theta \in \mathsf{LTL}^{\mathsf{s}}(A)$, the satisfaction relation $\sigma, t \models \theta$ is defined as follows:

$$\begin{array}{lll}
\sigma, t \models a & \text{iff} & \sigma(t) = a. \\
\sigma, t \models \theta U \eta & \text{iff} & \exists t' : t < t' \leq dur(\sigma),\ \sigma, t' \models \eta, \text{ and } \forall t'' : t < t'' < t', \sigma, t'' \models \theta. \\
\sigma, t \models \theta S \eta & \text{iff} & \exists t' : 0 \leq t' < t,\ \sigma, t' \models \eta, \forall t'' : t' < t'' < t,\ \sigma, t'' \models \theta.
\end{array}$$

The boolean operators are interpreted in the expected way. We set $S(\theta) = \{\sigma \in Sig(A) \mid \sigma, 0 \models \theta\}$.

As an example, the $\mathsf{LTL}^{\mathsf{s}}(A)$ formulas $\theta_{cont} = \bigvee_{a \in A}(a \wedge (aSa) \wedge (aUa))$ and $\theta_{disc} = \neg\theta_{cont}$ characterize the points of continuity and discontinuity respectively in a signal over A.

Returning now to the proof of Theorem 6.1, the route we follow is given schematically in the figure alongside.



Step (a): We show how to go from a formula in $\mathsf{FO}^{\mathsf{s}}(A)$ to a counter-free canonical ST-NFA accepting exactly its models. It can be checked that the inductive construction carried out in Section 4 for Theorem 4.1 produces a counter-free canonical ST-NFA at each step. This is true for the base cases $Q_a(x)$ and $x < y$. For the boolean operators, it follows by the closure properties of counter-free signal languages (cf. Lemma 5.1).

For the case of first-order quantification, let $\varphi = \exists x \psi$. Let $\mathcal{A}_\psi$ be a counter-free canonical ST-NFA which accepts the valid models for ψ. Without loss of generality we assume that $\mathcal{A}_\psi$ has no unreachable or dead states, and that its start and final states are respectively originating and terminating.

We now project away the x-row in transition labels of $\mathcal{A}_\psi$ to get a ST-NFA $\mathcal{A}'$ accepting the valid models of $\exists x \psi$. Now we can argue that $\mathcal{A}'$ cannot have a counter. If it did, then there are two cases: either the counter is such that no symbol in it was obtained by projecting away a '1' in the x-row, or there is a symbol in it which was obtained by projecting away a '1' in the x-row. In first case, this would mean a counter in $\mathcal{A}_\psi$ itself, contradicting the inductive assumption that $\mathcal{A}_\psi$ was counter-free. In the second it would mean $\mathcal{A}_\psi$ has a cycle containing a transition on a symbol with a '1' in the x-row, which would contradict the validity of the models generated by $\mathcal{A}_\psi$.

However, we are not yet done, as $\mathcal{A}'$ might have a non-canonical edge. Now let us canonicalize $\mathcal{A}'$ to get $\mathcal{A}''$, using the algorithm described in Section 3. Recall that the canonicalizing algorithm adds edges and finally deletes all the non-canonical edges. But it satisfies the property that the state space is the same, and every added edge from p to q has a corresponding path from p to q in $\mathcal{A}'$ which uses a non-canonical transition.

Now we claim that $\mathcal{A}''$ is counter-free (and, by construction, canonical). Suppose $\mathcal{A}''$ had a counter on states $q_0, \ldots, q_n$, on a string u. Now two possibilities exist:

- No u path in the counter uses an "added" edge. In this case this would be a counter in $\mathcal{A}'$ also, which is a contradiction.
- Some u path in the counter uses an "added" edge. So in $\mathcal{A}'$ the u-path has a corresponding u'-path which uses a non-canonical edge in $\mathcal{A}'$. But non-canonical edges in $\mathcal{A}'$ could only have come from a projection of a '1' in the x-row of a transition in $\mathcal{A}_\psi$. So the corresponding "unprojected" u'-path contains a symbol with a '1' in the x-row in $\mathcal{A}_\psi$. Once again, being part of a loop in $\mathcal{A}''$, and hence also in $\mathcal{A}_\psi$, this contradicts the validity of $\mathcal{A}_\psi$.

This completes the inductive proof of the claim that the set of signal models of a first-order formula is counter-free.

Steps (b) to (d) prove that we can go from an arbitrary counter-free canonical ST-NFA $\mathcal{A}$ over the alphabet A to a signal-language-equivalent $\mathsf{FO}^{\mathsf{s}}(A)$-sentence $\varphi_{\mathcal{A}}$.

Step (b): Let us denote by A' the alphabet $\{a' \mid a \in A\}$. For a canonical word $w = a_0 \ldots a_{2n} \in Can(A)$, we define $ann(w)$ to be the word $a'_0 a_1 a'_2 \ldots a_{2n-1} a'_{2n}$ in $(A \cup A')^*$, and extend it to work on subsets of $Can(A)$ as well. Now let $\mathcal{A}$ be a counter-free canonical ST-NFA over A. By the characterisation of counter-free languages in the proof of Lemma 5.1, there is a word-language equivalent NFA $\mathcal{B}_0$ that is bipartite and has no counters except possibly on odd-length u's. However, if we annotate the labels of the edges going from left to right (with the convention that the start states are in the left partition) by replacing each label a by a', then it is easy to see that the resulting NFA is counter-free, and accepts $ann(L(A))$. Let us call this NFA over $(A \cup A')$ as $\mathcal{B}$. Thus $L(\mathcal{B}) = ann(L(\mathcal{A}))$ and is a classical counter-free word language.

Step (c): Now by the results due to Schutzenberger [Sch65], McNaughton-Papert [MP71], and Kamp [Kam68] for classical word languages, the class of counter-free, star-free, FO-definable, and LTL-definable word languages all coincide. Thus for a counter-free NFA $\mathcal{B}$ over $(A \cup A')$ we have an $\mathsf{LTL}(A \cup A')$ formula θ which defines the same word language as $\mathcal{B}$. Thus $L(\mathcal{B}) = L(\theta)$.

Step (d): For a formula θ in $\mathsf{LTL}(A \cup A')$ such that $L(\theta) \subseteq A'(AA')^*$ and the "un-annotation" of $L(\theta)$ (i.e. $ann^{-1}(L(\theta))$ is canonical, we can construct a formula $\mathit{ltl\text{-}ltls}(\theta)$ in $\mathsf{LTL}^{\mathsf{s}}(A)$ which is such that $S(\mathit{ltl\text{-}ltls}(\theta)) = \mathit{timing}(ann^{-1}(L(\theta)))$.

We will use the abbreviation $\theta_1 U_d\, \theta_2$ to mean that at a point of discontinuity "$\theta_1 U \theta_2$" is true in an untimed sense, and define it to be $(\theta_2 U \theta_2) \vee (\theta_1 U(\theta_{disc} \wedge (\theta_2 \vee (\theta_1 \wedge (\theta_2 U \theta_2)))))$. Symmetrically we use $\theta_1 S_d\, \theta_2$ for $(\theta_2 S \theta_2) \vee (\theta_1 S(\theta_{disc} \wedge (\theta_2 \vee (\theta_1 \wedge (\theta_2 S \theta_2)))))$.

The translation *ltl-ltls* is defined as follows (we use $\widehat{\eta}$ for $\mathit{ltl\text{-}ltls}(\eta)$ in some places for brevity):

$$
\begin{array}{lcl}
\mathit{ltl\text{-}ltls}(a) & = & a \wedge \theta_{cont} (\text{where } a \in A). \\
\mathit{ltl\text{-}ltls}(a') & = & a \wedge \theta_{disc} (\text{where } a' \in A'). \\
\mathit{ltl\text{-}ltls}(\neg\theta_1) & = & \neg\widehat{\theta}_1. \\
\mathit{ltl\text{-}ltls}(\theta_1 \vee \theta_2) & = & \widehat{\theta}_1 \vee \widehat{\theta}_2. \\
\mathit{ltl\text{-}ltls}(\theta_1 U \theta_2) & = & (\theta_{disc} \Rightarrow (\widehat{\theta}_1 U_d \,\widehat{\theta}_2)) \wedge \\
 & & \quad (\theta_{cont} \Rightarrow (\theta_{cont}\, U(\theta_{disc} \wedge (\widehat{\theta}_2 \vee (\widehat{\theta}_1 \wedge (\widehat{\theta}_1 U_d \,\widehat{\theta}_2)))))). \\
\mathit{ltl\text{-}ltls}(\theta_1 S \theta_2) & = & (\theta_{disc} \Rightarrow (\widehat{\theta}_1 S_d \,\widehat{\theta}_2)) \wedge \\
 & & \quad (\theta_{cont} \Rightarrow (\theta_{cont}\, S(\theta_{disc} \wedge (\widehat{\theta}_2 \vee (\widehat{\theta}_1 \wedge (\widehat{\theta}_1 S_d \,\widehat{\theta}_2)))))).
\end{array}
$$

Lemma 6.1 *Let θ be an $\mathsf{LTL}(A \cup A')$ formula. Let w be a canonical word over A, and let $w' = ann(w)$. Let $\sigma \in timing(w)$ with the canonical interval representation $(a_0, I_0) \cdots (a_{2n}, I_{2n})$. Then for each $i \in \{0, \cdots, 2n\}$ and for all $t \in I_i$, we have $w', i \models \theta \iff \sigma, t \models \mathit{ltl\text{-}ltls}(\theta)$.*

Proof We do the proof by induction on θ. For the base case suppose $\theta = a$ with $a \in A$. Then $w', i \models a$ iff i is odd and $w'(i) = a$. This is true iff t is a point of continuity in σ and $\sigma(t) = a$. In turn this is true iff $\sigma, t \models a \wedge \theta_{cont}$.

The other base case and induction step for boolean operators are similar. We now show the induction step for $\theta = \theta_1 U \theta_2$ and omit the similar case of $\theta = \theta_1 S \theta_2$.

Left to right implication. Let $t \in I_i$ and suppose $w, i \models \theta_1 U \theta_2$. Then $\exists j > i$ such that $w, j \models \theta_2$ and $\forall i < i' < j \;\; w, i' \models \theta_1$. We note that by induction hypothesis we have that $\forall t'' \in I_j \;\; \sigma, t'' \models \widehat{\theta}_2$ and $\forall t'' \in \cup_{i' | i < i' < j} I_{i'} \;\; \sigma, t'' \models \widehat{\theta}_1$.

We distinguish two cases:

- i is even: then $\sigma, t \models \theta_{disc}$ and we have to show that $\sigma, t \models \widehat{\theta}_1 U_d \,\widehat{\theta}_2$.

  If $j = i + 1$ then $\sigma, t \models \widehat{\theta}_2 U \widehat{\theta}_2$. Otherwise, let k be the greatest even integer smaller or equal to j (this is the index corresponding to the last point of discontinuity before j). Let $t_k > t$ such that $I_k = \{t_k\}$, note that $\sigma, t_k \models \theta_{disc}$ and $\forall t < t'' < t_k \;\; \sigma, t'' \models \widehat{\theta}_1$. If $k = j$ then $\sigma, t_k \models \widehat{\theta}_2$. Otherwise $k = j - 1$ and $\sigma, t_k \models \widehat{\theta}_1 \wedge (\widehat{\theta}_2 U \widehat{\theta}_2)$. We have thus shown that in both cases $\sigma, t \models \widehat{\theta}_1 U_d \,\widehat{\theta}_2$.

- i is odd: then $\sigma, t \models \theta_{cont}$ and we have to show that $\sigma, t \models \theta_{cont}\, U\big(\theta_{disc} \wedge (\widehat{\theta}_2 \vee (\widehat{\theta}_1 \wedge (\widehat{\theta}_1 U_d \,\widehat{\theta}_2)))$.

  Let $t_{i+1} > t$ such that $I_{i+1} = \{t_{i+1}\}$. If $j = i+1$ then $\sigma, t_{i+1} \models \widehat{\theta}_2$. Otherwise $\sigma, t_{i+1} \models \widehat{\theta}_1$ and $w, i + 1 \models \theta_1 U \theta_2$. As $i + 1$ is even, by the previous case we have that $\sigma, t_{i+1} \models \widehat{\theta}_1 U_d \,\widehat{\theta}_2$. We have thus proved that $\sigma, t \models \theta_{cont}\, U\big(\theta_{disc} \wedge (\widehat{\theta}_2 \vee (\widehat{\theta}_1 \wedge (\widehat{\theta}_1 U_d \,\widehat{\theta}_2)))\big)$, t_{i+1} being a witness for the outermost until.

Right to left implication: Let $t \in I_i$ and suppose $\sigma, t \models \mathit{ltl\text{-}ltls}(\theta)$. We distinguish two cases:

- i is even: then $\sigma, t \models \theta_{disc}$ so $\sigma, t \models \widehat{\theta}_1 U_d \,\widehat{\theta}_2$. If $\sigma, t \models \widehat{\theta}_2 U \widehat{\theta}_2$ then $w, i+1 \models \theta_2$ so $w, i \models \theta_1 U \theta_2$.

Otherwise $\sigma, t \models \widehat{\theta_1} U\big(\theta_{disc} \wedge (\widehat{\theta_2} \vee \widehat{\theta_1} \wedge (\widehat{\theta_2} U \widehat{\theta_2}))\big)$, so there exists $t' > t$ such that $\sigma, t' \models \theta_{disc} \wedge (\widehat{\theta_2} \vee \widehat{\theta_1} \wedge (\widehat{\theta_2} U \widehat{\theta_2}))$ and $\forall t < t'' < t' \ \ \sigma, t'' \models \widehat{\theta_1}$. As $\sigma, t' \models \theta_{disc}$ there exists $j > i$ such that $I_j = \{t'\}$. We have that $\forall i < i' < j \ \ w, i' \models \theta_1$. If $\sigma, t' \models \widehat{\theta_2}$ then $w, j \models \theta_2$. If $\sigma, t' \models \widehat{\theta_1} \wedge (\widehat{\theta_2} U \widehat{\theta_2})$ then $w, j \models \theta_1$ and $w, j+1 \models \theta_2$. In both cases we have shown that $w, i \models \theta_1 U \theta_2$.

- i is odd: let t_{i+1} such that $I_{i+1} = \{t_{i+1}\}$. Necessarily $\sigma, t_{i+1} \models \widehat{\theta_2} \vee (\widehat{\theta_1} \wedge (\widehat{\theta_1} U_d \widehat{\theta_2}))$. If $\sigma, t_{i+1} \models \widehat{\theta_2}$ then $w, i+1 \models \theta_2$ so $w, i \models \theta_1 U \theta_2$. Otherwise $\sigma, t_{i+1} \models \widehat{\theta_1} \wedge (\widehat{\theta_1} U_d \widehat{\theta_2})$ so $w, i+1 \models \theta_1$. As $i+1$ is even and $\sigma, t_{i+1} \models \widehat{\theta_1} U_d \widehat{\theta_2}$, by the previous case, we have that $w, i+1 \models \theta_1 U \theta_2$; it follows that $w, i \models \theta_1 U \theta_2$.

□

Using the above lemma we can now show that if $\mathcal{A}$ and θ are as in the previous steps, then $S(\mathcal{A}) = S(\mathit{ltl\text{-}ltls}(\theta))$. To show $S(\mathcal{A}) \subseteq S(\mathit{ltl\text{-}ltls}(\theta))$, let $\sigma = (a_0, I_0) \cdots (a_{2n}, I_{2n}) \in S(\mathcal{A})$. Then $w = a_0 \cdots a_{2n} \in L(\mathcal{A})$ and $\sigma \in \mathit{timing}(w)$. Let $w' = \mathit{ann}(w)$. Then $w' \in L(\mathcal{B})$. Hence $w', 0 \models \theta$. By Lemma 6.1 we have that $\sigma, 0 \models \mathit{ltl\text{-}ltls}(\theta)$. Hence $\sigma \in S(\mathit{ltl\text{-}ltls}(\theta))$.

Conversely, suppose $\sigma \in S(\mathit{ltl\text{-}ltls}(\theta))$ with $\sigma = (a_0, I_0) \cdots (a_{2n}, I_{2n})$ being its canonical representation. That is $\sigma, 0 \models \mathit{ltl\text{-}ltls}(\theta)$. By Lemma 6.1 we have that $w', 0 \models \theta$, where $w = a_0 \cdots a_{2n}$ and $w' = \mathit{ann}(w)$. Hence $w' \in L(\mathcal{B})$. Hence $w \in L(\mathcal{A})$, and $\sigma \in S(\mathcal{A})$.

Step (e): A $\mathsf{LTL}^{\mathsf{s}}(A)$ formula θ can be translated to a $\mathsf{FO}^{\mathsf{s}}(A)$-formula ψ with one free variable x, such that for all $\sigma \in \mathit{Sig}(A)$, $\sigma, t \models \theta$ if and only if $\sigma, [t/x] \models \psi$. For a first-order formula φ let us denote by $\varphi[z/x]$ the formula obtained by substituting all free occurrences of x in φ by z. The translation *ltl-fo* is now given as follows:

$$
\begin{array}{lcl}
\mathit{ltl\text{-}fo}(a) & = & a(x) \\
\mathit{ltl\text{-}fo}(\theta_1 U \theta_2) & = & \exists z(x < z \wedge \mathit{ltl\text{-}fo}(\theta_2)[z/x] \wedge \forall y(x < y < z \Rightarrow \mathit{ltl\text{-}fo}(\theta_1)[y/x])) \\
\mathit{ltl\text{-}fo}(\theta_1 S \theta_2) & = & \exists z(z < x \wedge \mathit{ltl\text{-}fo}(\theta_2)[z/x] \wedge \forall y(z < y < x \Rightarrow \mathit{ltl\text{-}fo}(\theta_1)[y/x])) \\
\mathit{ltl\text{-}fo}(\neg\theta) & = & \neg(\mathit{ltl\text{-}fo}(\theta)) \\
\mathit{ltl\text{-}fo}(\theta_1 \vee \theta_2) & = & \mathit{ltl\text{-}fo}(\theta_1) \vee \mathit{ltl\text{-}fo}(\theta_2).
\end{array}
$$

We can now translate θ to the $\mathsf{FO}^{\mathsf{s}}(A)$ sentence $\psi = \forall x(\mathit{first}(x) \Rightarrow \mathit{ltl\text{-}fo}(\theta))$, so that $S(\theta) = S(\psi)$.

To summarize this direction of the proof: given a counter-free ST-NFA $\mathcal{A}$ over A by steps (b) and (c) we have an $\mathsf{LTL}(A \cup A')$ formula θ such that $\mathit{ann}(L(\mathcal{A})) = L(\theta)$. By step (d) we have $\mathsf{LTL}^{\mathsf{s}}(A)$ formula $\widehat{\theta}$ such that $S(\mathcal{A}) = S(\widehat{\theta})$. By step (e) we have an $\mathsf{FO}^{\mathsf{s}}(A)$ formula $\varphi_{\mathcal{A}}$ such that $S(\varphi_{\mathcal{A}}) = S(\widehat{\theta}) = S(\mathcal{A})$. This completes the proof of Theorem 6.1. □

7 Conclusion

In this paper we have defined a class of automata that run over signals and whose expressiveness coincides with a natural monadic second order logic. We identify a counter-free subclass of these automata that we show accept precisely the first-order definable signal languages. These results generalize some of the classical results connecting automata and logics over words, to signals.

These results were used in [CDP06, CDP07] to transparently obtain logical characterisations of timed automata with so-called "input-determined" metrics, in the continuous semantics. The first-order fragments of these logics correspond to timed temporal logics which use these metrics. By factoring through the counter-free characterisation in the present paper, we obtain counter-free timed automata characterisations for many timed temporal logics in the literature, including Metric Temporal Logic (MTL).

One of the questions which remains open is to characterise the expressiveness as well as decidability of $\mathsf{MSO}^{\mathsf{s}}$ when we allow second-order quantification over arbitrary subsets of non-negative reals (as against only finitely-varying subsets of non-negative reals), while continuing to interpret the logic over signals.

References

[ACH+95] Rajeev Alur, Costas Courcoubetis, Nicolas Halbwachs, Thomas A. Henzinger, Pei-Hsin Ho, Xavier Nicollin, Alfredo Olivero, Joseph Sifakis, and Sergio Yovine. The algorithmic analysis of hybrid systems. *Theor. Comput. Sci.*, 138(1):3–34, 1995.

[AD94] Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[AFH96] Rajeev Alur, Tomás Feder, and Thomas A. Henzinger. The Benefits of Relaxing Punctuality. *Journal of the ACM*, 43(1):116–146, 1996.

[B60̈] J.R. Büchi. On a decision method in restricted second order arithmetic. In *Z. Math. Logik Grundlag. Math.*, pages 66–92, 1960.

[CDP06] Fabrice Chevalier, Deepak D'Souza, and Pavithra Prabakhar. On continuous timed automata with input-determined guards. In Naveen Garg and S. Arun-Kumar, editors, *FSTTCS*, volume 4337 of *Lecture Notes in Computer Science*, pages 369–380, Kolkata, India, December 2006. Springer.

[CDP07] Fabrice Chevalier, Deepak D'Souza, and Pavithra Prabhakar. Counter-free input-determined timed automata. In Jean-François Raskin and P. S. Thiagarajan, editors, *FORMATS*, volume 4763 of *Lecture Notes in Computer Science*, pages 82–97. Springer, 2007.

[Kam68] Johan Anthony Willem Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, University of California, Los Angeles, California, 1968.

[Koy90] Ron Koymans. Specifying Real-Time Properties with Metric Temporal Logic. *Real-Time Systems*, 2(4):255–299, 1990.

[MP71] Robert McNaughton and Seymour Papert. *Counter-Free Automata*. MIT Press, Cambridge, MA, 1971.

[Rab69] M. O. Rabin. Decidability of second order theories and automata on infinite trees. *Trans. American Math. Monthly*, 141:1–35, 1969.

[Rab00] Alexander Rabinovich. Star free expressions over the reals. *Theor. Comput. Sci.*, 233(1-2):233–235, 2000.

[Rab02] Alexander Rabinovich. Finite variability interpretation of monadic logic of order. *Theor. Comput. Sci.*, 275(1-2):111–125, 2002.

[RT97] Alexander Rabinovich and Boris A. Trakhtenbrot. From finite automata toward hybrid systems (extended abstract). In Bogdan S. Chlebus and Ludwik Czaja, editors, *FCT*, volume 1279 of *Lecture Notes in Computer Science*, pages 411–422. Springer, 1997.

[Sch65] M. P. Schtzenberger. On finite monoids having only trivial subgroups. In *Information and Control*, pages 8:190–194, 1965.

[She75] M. Shelah. The monadic theory of order. *Annals of Mathematics*, 102:349–419, 1975.

[Tho90] Wolfgang Thomas. Automata on infinite objects. In *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B)*, pages 133–192. Elsevier, 1990.

[Wil99] Thomas Wilke. Classifying discrete temporal properties. In Christoph Meinel and Sophie Tison, editors, *STACS*, volume 1563 of *Lecture Notes in Computer Science*, pages 32–46. Springer, 1999.