On ω-Automata and Temporal Logic

Preliminary Report

Shmuel Safra
Weizmann Institute of Science

Moshe Y. Vardi
IBM Almaden Research Center

Abstract

We study here the use of different representation for infinitary regular languages in extended temporal logic. We focus on three different kinds of acceptance conditions for finite automata on infinite words, due to Büchi, Streett, and Emerson and Lei (EL), and we study their computational properties. Our finding is that Büchi, Streett, and EL automata span a spectrum of succinctness. EL automata are exponentially more succinct than Büchi automata, and complementation of EL automata is doubly exponential. Streett automata are of intermediate complexity. While translating from Streett automata to Büchi automata involves an exponential blow-up, so does the translation from EL automata to Streett automata. Furthermore, even though Streett automata are exponentially more succinct than Büchi automata, complementation of Streett automata is only exponential. As a result, we show that the decision problem for ETL_{EL}, where temporal connectives are represented by EL automata, is EXPSPACE-complete, and the decision problem for ETL_S, where temporal connectives are represented by Streett automata, is PSPACE-complete.

1 Introduction

Since the proposal by Pnueli in 1977 [Pn77] to apply Temporal Logic (*TL*) to the specification and verification of concurrent programs, the role of *TL* as a feasible approach to that task has been widely accepted [E89, Ga87, Kr87]. Over the last decade an extensive research has been carried out concerning both practical and theoretical aspects of using *TL* to specify and verify concurrent programs (see survey in [Pn86]). One of the conclusions of this research is that the standard *TL*, which consists of the temporal connectives *nexttime* and *until* is not expressive enough for its task.

The first to complain about the expressive power of *TL* was Wolper [Wo83] who observed that temporal logic cannot express certain regular events (in fact, *TL* can express precisely the star-free ω-regular events [To81]). As was shown later, this makes *TL* inadequate for *compositional* verification (as opposed to *global* verification) [LPZ85]. As a remedy, Wolper suggested extending TL with temporal connectives that correspond to regular grammars. The extended TL, is called, naturally enough, ETL. It was shown later that the expressive power of ETL is sufficient and necessary to perform compositional specification and verification [LPZ85, Pn85, Pn86].

ETL was further explored by Wolper et al. in [WVS83], where the basic paradigm of ETL was made explicit: every ω-regular langauage L over an n-letter alphabet can be viewed as an n-ary temporal connective (the restriction to ω-regular language is for practical reasons). To obtain a finitary syntax, a finitary representation for ω-regular languages must be chosen. This choice is the subject matter of this paper.

As in [WVS83] we focus on representation of ω-regular languages by means of nondeterministic finite-state automata. Unlike the class of regular languages, however, there are several types of finite-state automata that defines the class of ω-regular language: Büchi automata [Bu62], Street automata [St82], which generalize Büchi automata, and Emerson-Lei (abbr. EL) automata [EL87], which generalize Streett automata.[1] While these automata have the same expressive power, they differ in their succinctness: the nonemptiness problem is NL-complete for Büchi automata [VW88], P-complete for Streett automata [EL87], and NP-complete for EL automata [EL87]. Thus, it is clear that the choice of representation has the potential of having a drastic effect on the complexity of the decision problem for ETL.

Sistla et al. [SVW87] studied ETL_B, where temporal connectives are represented by Büchi automata, and showed that the decision problem for this logic is PSPACE-complete, just like the decision problem for TL [HR83, SC85].[2] What we'd like here is to get "more bang for the buck". That is, we'd like to use the most succinct representation for temporal connectives that is compatible with retaining the polynomial-space upper bound on the complexity of the decision problem for the logic. To that end we study the computational properties of ω-automata. In particular we study the complexity of translating one representation to another and the complexity of complementation (since the class of ω-regular automata is closed under complementation). We believe that this investigation is of great interest in view of the emerging importance of ω-automata to the specification and verification of concurrent programs [AS85, AS87, CY88, MP87, Pa81, Va85, Va87, VW85, VW86].

Our finding is that Büchi, Streett, and EL automata span a spectrum of succinctness. We focus first on EL automata. We show that EL automata are exponentially more succinct than Büchi automata. Translating from the former to the latter involves an exponential blow-up. Furthermore, complementation of EL automata is doubly exponential, while complementation for Büchi automata is only exponential [Sa88, SVW87]. As a result, we show that the decision problem for ETL_{EL}, where temporal connectives are represented by EL automata, is EXPSPACE-complete. This rules out the use of EL

[1] The class of Rabin automata [Ra69] also defines the class of of ω-regular languages. This representation, however, is polynomially equivalent to the Büchi representation.

[2] ETL_B is called ETL_r in [WVS83, SVW87].

automata for the representation of temporal connectives.

The more surprising result is that Streett automata are of intermediate complexity. While translating from Streett automata to Büchi automata involves an exponential blow-up, so does the translation from EL automata to Streett automata. Furthermore, even though Streett automata are exponentially more succinct than Büchi automata, complementation of Streett automata is only exponential. This result is rather unexpected; it involves a rather complicated construction and requires a fairly deep analysis of acceptance by Streett automata. As a result, we show that the decision problem for ETL_S, where temporal connectives are represented by Streett automata, is PSPACE-complete. Since Streett automata also enable us to directly encode very powerful notions of *fairness* [FK84], one is tempted to recommend the Streett representation for ω-regular temporal connectives. We will return to this point in our concluding remarks.

2 Basic Definitions

2.1 Finite Automata

We consider words over finite alphabets. A *finite word* over an alphabet Σ is a member of Σ^*. An *infinite word* over an alphabet Σ is a member of Σ^ω. It will usually be clear from the context whether we refer to finite or infinite words.

A *table* T is a tuple (Σ, S, S_0, ρ), where Σ is a finite *alphabet*, S is a finite set of *states*, $S_0 \subseteq S$ is a set of *initial* states, and $\rho : S \times \Sigma \to 2^S$ is the *transition function*. A *run* of T over an word $w = a_0 a_1 \ldots a_{n-1}$ in Σ^* from a state s to a state t is a sequence $s_0, s_1, \ldots, s_n$ in S^* such that $s_0 = s$, $s_n = t$ and $s_{i+1} \in \rho(s_i, a_i)$ for all i, $0 \leq i < n$. A *run* of T over an word $w = a_0 a_1 \ldots$ in Σ^ω is a sequence $s_0, s_1, \ldots$ in S^ω such that $s_0 \in S_0$, and $s_{i+1} \in \rho(s_i, a_i)$ for all $i \geq 0$.

Acceptance of runs is defined in terms of their limit behavior. Let $\mathbf{x} = x_0, x_1, \ldots$ be in X^ω, for some finite set X. Then $lim(\mathbf{x})$ is the set of elements that occur in $\mathbf{x}$ infinitely often, i.e.,

$$lim(\mathbf{x}) = \{x \mid (\forall i \geq 0)(\exists j \geq i) \text{ such that } x_j = x\}.$$

An *acceptance condition* for a table $T = (\Sigma, S, S_0, \rho)$ is a collection $C \subseteq 2^S$ of sets of states. A set S' is *accepting* with respect to C if it is in C. A run $\mathbf{s} = s_0, s_1, \ldots$ is *accepting* with respect to an acceptance condition C if $lim(\mathbf{s})$ is accepting with respect to C.

A *Muller* automaton A is a pair (T, C) consisting of a table $T = (\Sigma, S, S_0, \rho)$ and an acceptance condition $C \subseteq 2^S$ for T. A word $w \in \Sigma^\omega$ is *accepted* by A if T has a run $\mathbf{s}$ on w that is accepting with respect to C. $L_\omega(A)$ is the set of all words in Σ^ω accepted by A. An automaton A is *nonempty* if $L_\omega(A)$ is nonempty. A language $L \subseteq \Sigma^\omega$ is ω-regular if $L = L_\omega(A)$ for some Muller automaton A.

We can specify acceptance conditions for a table $T = (\Sigma, S, S_0, \rho)$ more succinctly by means of *acceptance formulas*, which are propositional formulas constructed using the states in S as propositional variables. Let S' be a subset of S, then $\chi(S')$ is the characteristic function of S', which can be viewed as a truth assignment on S. A set S' is accepting with respect to an acceptance formula f if $\chi(S')$ satisfies f.

Example 2.1: Let $S = \{p, q, r\}$, and let f be the acceptance formula $\neg p \wedge (q \vee r)$. Then, the accepting set with respect to f are $\{q\}$, $\{r\}$, and $\{q, r\}$. That is, a run is accepting with respect to f if p occurs only finite often in the run and either q or r occurs infinitely often in the run. ∎

An *Emerson-Lei (EL) automaton* A is a pair (T, f) consisting of a table T and an acceptance formula f for T. *Streett automata* and *Büchi automata* are obtained by restricting the class of acceptance formulas. A *Büchi formula* is simply a disjunction of propositional variables. Equivalently, a *Büchi condition* is a set F of states, and a set S' is accepting if $S' \cap F \neq \emptyset$. A *Streett formula* is of the form $\bigwedge_{i=1}^{k}(\bigvee L_i \rightarrow \bigvee U_i)$, $k \geq 0$, where L_i and U_i are sets of states. Equivalently, a *Streett condition* is a collection of pairs of sets of states, and a set S' is accepting if $S' \cap L_i \neq \emptyset$ entails $S' \cap U_i \neq \emptyset$ for every pair (L_i, U_i). It is easy to see that Büchi condition is a special case of Streett condition.

The interest in Büchi and Streett conditions stems from the fact that they can be viewed as *normal forms*, they both define the class of ω-regular languages. More formally, for every ω-regular language L there exists a Büchi automaton A such that $L = L_\omega(A)$ [Mc66]. Furthermore, for every ω-regular language L there exists a *deterministic* Streett automaton A' such that $L = L_\omega(A')$ [Mc66, St82] (note that deterministic Büchi automata are weaker than nondeterministic Büchi automata [Ch74]). Streett condition is also of special interest because it is essentially the condition of *generalized fairness* defined in [FK84] (see also [Fr86]).

2.2 Extended Temporal Logic

We consider a propositional temporal logic where the temporal connectives are defined by ω-regular languages [WVS83, Wo83, SVW87]. The formulas built from a set *Prop* of atomic propositions. The set of formulas is defined inductively as follows:

- every proposition $p \in Prop$ is a formula.
- if ψ_1 and ψ_2 are formulas, then $\neg\psi_1$ and $\psi_1 \wedge \psi_2$ are formulas.
- If Σ is the alphabet $\{a_1, \ldots, a_n\}$, $L \subseteq \Sigma^\omega$ is an ω-regular language, and $\psi_1, \ldots, \psi_n$ are formulas, then $L(\psi_1, \ldots, \psi_n)$ is a formula.

The actual syntax for the logic depends on the representation of ω-regular languages. If we represent ω-regular languages by Büchi automata, we get ETL_B (called ETL_r in

[WVS83, SVW87]). Analogously, if we represent ω-regular languages by Streett automata (resp., EL automata) we get ETL_S (resp., ETL_{EL}).[3]

We interpret ETL formulas over *computations*, which are infinite sequences of truth assignments, i.e., functions $\pi : \omega \to 2^{Prop}$ that assign truth values to the atomic propositions in each state. We now define *satisfaction* of formulas. Satisfaction of a formula ψ at a position i in a computation π is denoted $\pi, i \models f$.

- for an atomic proposition p, we have $\pi, i \models p$ iff $p \in \pi(i)$.
- $\pi, i \models \psi_1 \wedge \psi_2$ iff $\pi, i \models \psi_1$ and $\pi, i \models \psi_2$.
- $\pi, i \models \neg\psi$ iff $\pi, i \not\models \psi$.
- $\pi, i \models L(\psi_1, \ldots, \psi_n)$, where $L \subseteq \Sigma^\omega$, for $\Sigma = \{a_1, \ldots, a_n\}$, if there is a word $w = a_{j_0} a_{j_1} \ldots$ in L such that for all $k \geq 0$ we have that $\pi, i + k \models \psi_{j_k}$.

We say that a computation π satisfies f, denoted $\pi \models f$, if $\pi, 0 \models f$. Every formula φ defines a set of computations $Mod(\varphi) = \{\pi \mid \pi \models \varphi\}$, i.e., the set of computations that satisfy φ. We will say that a formula is *satisfiable* if $Mod(\varphi)$ is nonempty. The *satisfiability problem* is to determine, given a formula φ, whether φ is satisfiable.

3 Translations among the Models

We know that Büchi, Streett, and EL automata all have the same expressive power; they all define the class of ω-regular languages. Thus, for every EL automaton A there exists an *equivalent* Büchi automaton A', that is, $L_\omega(A) = L_\omega(A')$. What we are interested here is in the size of A' relative to the size of A, where the size of an automaton is the length of its description in some standard encoding.

Proposition 3.1: *Let A be an EL automaton of size n. Then there is an equivalent Büchi automaton A' of size $2^{O(n)}$.*

Proof: The proof is constructive. Let $A = (\Sigma, S, S_0, \rho, f)$ be an EL automaton of size n. Let $\bigvee_{i=1}^k f_i$ be the disjunctive normal form of f, where clearly $0 \leq k \leq 2^n$. it is easy to see that $L_\omega(A) = \bigcup_{i=1}^k L_\omega(A_i)$, where $A_i = (\Sigma, S, S_0, \rho, f_i)$, for $1 \leq i \leq k$. Now we construct for each A_i an equivalent Büchi automaton A'_i of size $O(n^2)$. Finally, we take the disjoint union of the A'_i's to obtain A'. ∎

Since a Büchi automaton is in particular also a Streett automaton and Streett automaton is in particular also an EL automaton, Proposition 3.1 also gives us exponential translations from EL automata to Streett automata and from Streett automata to Büchi automata. We now show that these translations are inherently exponential.

[3]We note that Wolper et al. considered also the logics ETL_f and ETL_l where not all ω-regular connectives are allowed. They showed that these logics are nevertheless as expressive as ETL_B [WVS83].

Proposition 3.2: *For every $n > 0$ there is a language L_n that can be accepted by a (deterministic) Streett automaton of size $O(n)$ but cannot be accepted by any Büchi automaton with less than 2^n states.*

Proof: Let Σ be the ternary alphabet $\{0, 1, 2\}$. Every word w in Σ^ω can be viewed as a word in $(\Sigma^n)^\omega$, that is, as an infinite sequence of n-vectors over Σ. Let $u = \langle a_0, \ldots, a_{n-1} \rangle \in \Sigma^n$. We say that i is *0-activated* (resp., *1-activated*) in u, for $0 \leq i < n$, if $a_i = 0$ (resp., $a_i = 1$). Let $w \in \Sigma^\omega$, then $w = u_0 u_1 \ldots$, where $u_j \in \Sigma^n$ for all $j \geq 0$. We say that i is 0-activated (resp., 1-activated) in w, for $0 \leq i < n$, if i is 0-activated (resp., 1-activated) in u_j for infinitely many j's. L_n is the set of words in Σ^ω with a symmetric activity record. Formally,

$$L_n = \{w \in \Sigma^\omega \mid i \text{ is 0-activated in } w \text{ iff } i \text{ is 1-activated in } w \text{ for } 0 \leq i < n\}.$$

L_n is accepted by a (deterministic) Streett automaton with $O(n)$ states and an acceptance formula of length $O(n)$. Let $A_n = (\Sigma, S, S_0, \rho, f)$, where $S = \{0, ..., n-1\} \times \{0, 1, 2\}$, $S_0 = \{(0, 0)\}$, $\rho((i, k), l) = (i + 1 \bmod n, l)$ for $0 \leq i < n$ and $0 \leq k, l \leq 2$, and f is the formula

$$\bigwedge_{i=0}^{n-1} ((i, 0) \rightarrow (i, 1)) \wedge ((i, 1) \rightarrow (i, 0)).$$

The reader can verify that $L_n = L_\omega(A_n)$. On the other hand, in the full paper we shall show that L_n is not accepted by any Büchi automaton with less than 2^n states. The proof depends crucially on the fact that for any Büchi acceptance condition the set of accepting sets is closed under extension. ∎

Proposition 3.3: *For every $n > 0$ there is a language L_n that can be accepted by a (deterministic) EL automaton of size $O(n)$ but cannot be accepted by any Streett automaton with less than 2^n states.*

Proof: Let Σ be the binary alphabet $\{0, 1\}$. Every word w in Σ^ω can be viewed as a word in $(\Sigma^n)^\omega$, that is, as an infinite sequence of n-bit vectors. We take L_n to consist of the sequences that are almost everywhere identical. Formally,

$$L_n = \{w \mid \exists u \in (\Sigma^n)^* \text{ and } v \in \Sigma^n \text{ such that } w = uv^\omega\}.$$

L_n is accepted by a (deterministic) EL automaton with $O(n)$ states and an acceptance formula of length $O(n)$. Let $A_n = (\Sigma, S, S_0, \rho, f)$, where $S = \{0, ..., n-1\} \times \{0, 1\}$, $S_0 = \{(0, 0)\}$, $\rho((i, k), l) = (i + 1 \bmod n, l)$ for $0 \leq i < n$ and $0 \leq k, l \leq 1$, and f is the formula

$$\bigwedge_{i=1}^{n} \neg((i, 0) \wedge (i, 1)).$$

The reader can verify that $L_n = L_\omega(A_n)$. On the other hand, in the full paper we shall show that L_n is not accepted by any Streett automaton with less than 2^n states. The proof depends crucially on the fact that for any Streett acceptance condition the set of accepting sets is closed under union. ∎

The bottom line of the results in this section is that EL automata are exponentially more succinct than Streett automata, and the latter are exponentially more succinct than Büchi automata. The two exponentials, however, do not combine, and EL automata are only exponentially more succinct than Büchi automata. Interestingly, the difference in the complexity of the *nonemptiness problem* (i.e., checking whether a given automaton is nonempty) is less substantial; the nonemptiness problem is NL-complete for Büchi automata [VW88], P-complete for Streett automata [EL87], and NP-complete for EL automata [EL87].

4 Complementation

Since the set of *ETL* formulas is closed under negation, it is clear that we have to deal with the complementation of ω-regular languages. Büchi has shown that the class of ω-regular languages is closed under complementation [Bu62]. That is, if $L \subseteq \Sigma^\omega$ is an ω-regular language, then so is the complementary language $\Sigma^\omega - L$. Constructive proofs of this property were given in [Mc66, Sa88, SVW87] in the context of Büchi automata. The best bound was given in [Sa88]:

Proposition 4.1: *Let A be an n-state Büchi automaton over an alphabet Σ. Then there is a Büchi automaton A' with $2^{O(n \log n)}$ states such that $L_\omega(A') = \Sigma^\omega - L_\omega(A)$.*

The above upper bound was proven optimal by Michel [Mi88].

A related issue is that of *universality*. An automaton A over an alphabet Σ is universal if $L_\omega(A) = \Sigma^\omega$. The *universality problem*, checking whether a given automaton is universal, is related to complementation, since $L_\omega(A) = \Sigma^\omega$ precisely when $\Sigma^\omega - L_\omega(A) = \emptyset$. The universality problem for Büchi automata was studied in [SVW87]:

Proposition 4.2: *The universality problem for Büchi automata is PSPACE-complete.*

Here we examine the issues of complementation and universality in the context of Büchi, Streett, and EL automata. Our first result is that complementation of Büchi automata is exponential even if one is willing to use an EL automata for the complementary language.

Proposition 4.3: *There is an alphabet Σ such that for every $n > 0$ there is a language L_n with the property that L_n can be accepted by a Büchi automaton of size $O(n)$ but $\Sigma^\omega - L_n$ cannot be accepted by any EL automaton with less than 2^n states.*

Proof: Let Σ be the alphabet $\{0,1,2\}$, and let $\Delta = \{0,1\}$. A word $u \in \Delta^n$ can be viewed as an n-bit number. Let $w = u_1 2 u_2 2 \ldots$, where $u_j \in \Delta^n$ for all $j \geq 1$ (i,e $w \in (\Delta^n 2)^\omega$). Then w is a sequence of n-bit numbers separated by 2's. We say that w is an n-*counter* if u_{j+1} is the successor of u_j (modulo 2^n) for all $j \geq 1$. We take $\Sigma^\omega - L_n$ to consist of all n-counters. Formally,

$$L_n = \{w \in \Sigma^\omega \mid w \text{ is not an } n - \text{counter}\}.$$

In the full paper we shall show that L_n is accepted by a Büchi automaton of size $O(n)$, but $\Sigma^\omega - L_n$ is not accepted by any EL automaton with less than 2^n states. ∎

We now turn to complementation of EL automata. A straightforward approach would be to combine Propositions 3.1 and 4.1. That is, to complement an EL automaton we first convert it to an equivalent Büchi automaton, which we then complement.

Proposition 4.4: *Let A be a EL automaton of size $O(n)$ over an alphabet Σ. Then there is a Büchi automaton A' with $2^{2^{O(n)}}$ states such that $L_\omega(A') = \Sigma^\omega - L_\omega(A)$.*

Now we know that both the translation from EL automata to Büchi automata and the complementation of Büchi automata are inherently exponential, by Propositions 3.2 and 4.3, but it does not follow from this that the complementation of EL automata is inherently doubly exponential. It is conceivable that EL automata can be complemented exponentially by some direct construction. Nevertheless, the following proposition shows that this is not the case.

Proposition 4.5: *There is an alphabet Σ such that for every $n > 0$ there is a language L_n with the property that L_n can be accepted by an EL automaton of size $O(n)$ but $\Sigma^\omega - L_n$ cannot be accepted by any EL automaton with less than 2^{2^n} states.*

Proof: Let Σ be the alphabet $\{0,1,2,3\}$, and let $\Delta = \{0,1\}$. A word $u \in \Delta^n$ can be viewed as an n-bit number. Consider now a word $v \in (\Delta^n 2 \Delta)^{2^n}$, i.e., $v = u_1 2 i_1 \ldots u_{2^n} 2 i_{2^n}$. The sequence $i_1 \ldots i_{2^n}$ can be viewed an a 2^n-bit number. Suppose now that u_1 is 0^n and u_{j+1} is the successor of u_j for $1 \leq j < 2^n$. Then each u_j can be viewed as an index to the bit i_j. Thus, we call v an *indexed* 2^n-bit number. We say that an indexed 2^n-bit number $v' = u_1 2 i'_1 \ldots u_{2^n} 2 i'_{2^n}$ is the successor of v (modulo 2^{2^n}) if the sequence $i'_1 \ldots i'_{2^n}$ is the successor of the sequence $i'_1 \ldots i'_{2^n}$ (modulo 2^{2^n}).

Let $w = v_1 3 v_2 3 \ldots$, where v_j is an indexed 2^n-bit counter for all $j \geq 1$, i.e., w is a sequence of indexed 2^n-bit numbers separated by 3's. We say that w is an 2^n-*counter* if there is some j_0 such that u_{j+1} is the successor of u_j (modulo 2^{2^n}) for all $j \geq j_0$. (Notice that, unlike the definition of n-counters, we require proper counting "behavior" only in the limit). We take $\Sigma^\omega - L_n$ to consist of all 2^n-counters. Formally,

$$L_n = \{w \in \Sigma^\omega \mid w \text{ is not a } 2^n - \text{counter}\}.$$

In the full paper we shall show that L_n is accepted by an EL automaton of size $O(n)$, but $\Sigma^\omega - L_n$ is not accepted by any EL automaton with less than 2^{2^n} states. ∎

The ideas underlying Propositions 4.4 and 4.5 can be extended to deal with for the universality problem for EL automata.

Proposition 4.6: *The universality problem for EL automata is EXPSPACE-complete.*

Finally, we turn to Streett automata. Clearly, they can be complemented by a doubly exponential construction, but the lower bound of Proposition 4.5 seem to need the full power of EL automata. It turns out that there is an exponential complementation for Streett automata. This is a fairly deep result to which we dedicate the rest of this section.

For a word w we denote by $w[i, j]$ the word $w_i, w_{i+1}, ..., w_{j-1}$. We say that a run $\mathbf{s} = s_0, s_1, \ldots, s_n$ *visits* a state s if there exists an i such that $s_i = s$.

Let $T = (\Sigma, S, S_0, \rho)$ be a table. We extend ρ to take as arguments sets of states and finite words as follows: $\rho(V, \lambda) = V$, and $\rho(V, wa) = \{t \mid t \in \rho(s, a) \text{ for some } s \in \rho(V, w)\}$.

A *history graph* $G = \langle V, E\rangle$ for T is an edge-labelled directed graph over a subset of the states $V \subseteq S$, where each edge is labelled by a collection of sets of states, $E: V \times V \to 2^{2^S}$. A history graph is intended to describe all possible runs of T over some finite words. A history graph $G = \langle V, E\rangle$ *corresponds* to a finite word $w \in \Sigma^*$ if $\rho(V, w) = V$ and $H \in E(s, t)$ if and only if there is a T-run from s to t over w visiting exactly the set H. A history graph $G = \langle V, E\rangle$ is *idempotent* if for any (not necessarily distinct) states $r, s, t \in V$, if $H_1 \in E(r, s)$ and $H_2 \in E(s, t)$ then $H_1 \cup H_2 \in E(r, t)$.

History graphs can also describe runs over infinite words. A history graph $G = \langle V, E\rangle$ *corresponds* to an infinite word $w \in \Sigma^\omega$ if there is an increasing sequence $0 \le i_0 < i_1 < \ldots$ such that $\rho(S_0, w[0, i_0]) = V$ and G corresponds to $w[i_j, i_{j+1}]$ for all $j \ge 0$.

Lemma 4.7: *For each word w there exists an idempotent history graph that corresponds to w.*

Proof: Given a word $w \in \Sigma^\omega$, we assign to each pair of integers (i, j), $i < j$, a triple $\langle V_i, V_j, E_{ij}\rangle$ such that $V_i = \rho(S_0, w[0, i])$, $V_j = \rho(S_0, w[0, j])$, and $E_{ij}: V_i \times V_j \to 2^{2^S}$ is such that $H \in E(s, t)$ iff there is a T-run from s to t over $w[i, j]$ visiting exactly the set H. By the Infinite Ramsey Theorem, there exists a history graph, $G = \langle V, E\rangle$, and an infinite set $X \subseteq \mathcal{N}$ such that for every $i \in X$, $\rho(S_0, w[0, i]) = V$, and for every $i, j \in X$, $i < j$, we have that G corresponds to $w[i, j]$. Thus, G corresponds to w, and is idempotent. ∎

Now let A be an EL automaton (T, f). A history graph $G = \langle V, E\rangle$ for T is *nonaccepting* with respect to f if for each $s \in V$, and for each history set $H \in E(s, s)$, H is a nonaccepting.

Lemma 4.8: *A word w is not accepted by A iff there exists a nonaccepting idempotent history graph that corresponds to w.*

Proof: Note that there may be different history graphs that correspond to a word w. Nevertheless, we can show that they are all either accepting or nonaccepting.

Assume there exists a nonaccepting idempotent history graph $G = \langle V, E \rangle$ that corresponds to w via the sequence $i_0, i_1, \ldots$. Consider any T-run $\mathbf{s}$ over w such that $lim(\mathbf{s}) = H$. Since V is finite, there is a state $s \in V$ such that $s = s_{i_j}$ for infinitely many j's. By the idempotence condition $H \in E(s, s)$, and because G is nonaccepting, H is nonaccepting.

In the other direction, assume that there is no nonaccepting idempotent history graph that corresponds to w. Then, by Lemma 4.7, there is an accepting idempotent history graph $G = \langle V, E \rangle$ corresponding to w. That is, there is some $s \in V$ and an accepting set $H \in E(s, s)$. That means that $s \in \rho(S_0, w[0, i_0])$, and for every $j \geq 0$, there is a T-run from s to s over $w[i_j, i_{j+1}]$ visiting exactly the set H. It follows that there is an accepting T-run over w. ∎

We could use history graphs for complementation, but unfortunately history graphs are of exponential size, which means that a construction that uses history graphs directly would be doubly exponential. What we need is to encode the information captured in a history graph in a more efficient way. This is possible if A is a Streett automaton, which means that f is a Streett formula $\bigwedge_{i=1}^{k} (\bigvee L_i \rightarrow \bigvee U_i)$. Let $\Delta = \{1, \ldots, k\}$. A *pair assignment* for a set V of states is a function $\alpha : V \rightarrow \Delta^*$ that assigns to every state in V a sequence of *distinct* indices of pairs. Note that the length of $\alpha(s)$ is bounded by k for all $s \in V$. The pair assignment α is a pair assignment for a history graph $G = \langle V, E \rangle$ if α is an assignment on V and for each $s \in V$, where $\alpha(s) = i_1, \ldots, i_l$, and every $H \in E(s, s)$ there exists a j, $1 \leq j \leq l$ such that

- $H \cap L_{i_j} \neq \emptyset$ and
- $H \cap U_{i_{j'}} = \emptyset$ for every $j' \leq j$.

Lemma 4.9: *An idempotent history graph G is nonaccepting if and only if it has a pair assignment.*

Proof: The condition is clearly sufficient. To prove necessity, suppose that $G = \langle V, E \rangle$ is an idempotent nonaccepting history graph and $s \in V$. We define a sequence of $i_1, \ldots, i_k$ of pair indices, a sequence $H_1, \ldots, H_k$ of elements of $E(s, s)$, and sequences $\mathbf{H}_0, \ldots, \mathbf{H}_k$ and $\mathbf{H}'_0, \ldots, \mathbf{H}'_k$ of subsets of $E(s, s)$ such that

1. If $H \in \mathbf{H}_{i_j}$, $1 \leq j \leq l$, then $H \cap L_{i_j} \neq \emptyset$ and $H \cap U_{i_{j'}} = \emptyset$ for $1 \leq j' \leq j$.
2. If $H \in \mathbf{H}'_{i_j}$, $1 \leq j \leq l$, then $H \cap L_{i_{j'}} = \emptyset$ and $H \cap U_{i_{j'}} = \emptyset$ for $1 \leq j' \leq j$.
3. For $1 \leq j \leq l$, we have that $\mathbf{H}_j = \{H \in \mathbf{H}'_{j-1} \mid H \cap L_{i_j} \neq \emptyset\}$.
4. For $0 \leq j \leq l$, we have that $\mathbf{H}'_j = E(s, s) - \bigcup_{j=0}^{l} \mathbf{H}_j$.

5. For $0 \leq j \leq l$, we have that $\mathbf{H}'_j$ is closed under union.

We first take $\mathbf{H}_0$ to be the empty set and we take $\mathbf{H}'_0$ to be $E(s,s)$. It is easy to verify that the inductive properties hold.

Inductively, suppose that we have already defined $i_1, \ldots, i_l$, $H_1, \ldots, H_l$, $\mathbf{H}_0, \ldots, \mathbf{H}_l$, and $\mathbf{H}'_0, \ldots, \mathbf{H}'_l$, $l \geq 0$.

Consider now the collection $\mathbf{H}'_l$. If $\mathbf{H}' = \emptyset$, then $E(s,s) = \bigcup_{j=1}^{l} \mathbf{H}_j$, so we can take $\alpha(s)$ to be $i_1, \ldots, i_l$, and we are done. Assume now that $\mathbf{H}'_l \neq \emptyset$. We know that $\mathbf{H}'_l$ is closed under union. Let $H_{l+1} = \bigcup_{H \in \mathbf{H}'_l} H$. By induction, $H_{l+1} \in \mathbf{H}'_l \subseteq E(s,s)$, so H_{l+1} is nonaccepting. It follows that for some index j we have that $H_{l+1} \cap L_j \neq \emptyset$ and $H_{l+1} \cap U_j = \emptyset$. Let $i_{l+1} = j$, let $\mathbf{H}_{l+1} = \{H \in \mathbf{H}'_l \mid H \cap L_j \neq \emptyset\}$, and let $\mathbf{H}'_{l+1} = \mathbf{H}'_l - \mathbf{H}_{l+1}$. We leave it to the reader to verify that the inductive properties holds. ∎

We are now in position to define a more compact version of the history graph. The idea is to replace the sets labelling the edges by indices of pairs with which these sets "interact". Let $\Delta' = \{1, \ldots, k, k+1\}$. A *pair graph* $G = \langle V, F \rangle$ for A with respect to a pair assignment α on V is is a labelled directed graph over the set $V \subseteq S$, where $F : V^3 \rightarrow 2^{\Delta'^2}$ labels every triple of states with a set of pairs of indices such that if $(i,j) \in F(r,s,t)$ then $1 \leq i, j \leq \mid \alpha(t) \mid +1$. Put differently, the label of every edge is a mapping from V to $2^{\Delta'^2}$. Note that the size of the pair graph is bounded by n^5, where n is the size of A.

A pair graph $G = \langle V, F \rangle$ with respect to a pair assigment α *corresponds* to a finite word $w \in \Sigma^*$ if $\rho(V, w) = V$ and $(i,j) \in F(r,s,t)$ where $\alpha(t) = l_1, \ldots, l_m$ if and only if there is an T-run from r to s over w that

- visits no state in $L_{l_{i'}}$ for $i' < i$, and visits no state in $U_{l_{j'}}$ for $j' < j$,
- visits a state in L_{l_i}, if $i \leq m$, and visit a state in U_{l_j}, if $j \leq m$.

A pair graph $G = \langle V, F \rangle$ with respect to a pair assigment α *corresponds* to an infinite word $w \in \Sigma^\omega$ if there is an increasing sequence $0 \leq i_0 < i_1 < \ldots$ such that $\rho(S_0, w[0, i_0]) = V$ and G corresponds to $w[i_j, i_{j+1}]$ for all $j \geq 0$.

We now have to redefine idempotence and nonacceptance for pair graphs. A pair graph $G = \langle V, F \rangle$ with respect to a pair assignment α is *idempotent* if for any (not necessarily distinct) states $r, s, t, u \in V$, if $(i_1, j_1) \in F(r,s,u)$ and $(i_2, j_2) \in F(s,t,u)$, then $(\min(i_1, i_2), \min(j_1, j_2)) \in F(r,t,u)$. A pair graph $G = \langle V, F \rangle$ with respect to a pair assignment α is *nonaccepting* if for each $s \in V$ and index pair $(i,j) \in F(s,s,s)$, we have that $i < j$.

Lemma 4.10: *A word w is not accepted by A if and only if there exists a pair assignment α and nonaccepting idempotent pair graph with respect to α that corresponds to w.*

Proof: If w is not accepted by A, then, by Lemma 4.8, there is a nonaccepting idempotent history graph $G = \langle V, E \rangle$ that corresponds to w. By Lemma 4.9, this history graph has a pair assignment α. We now define a pair graph $G' = \langle V, F \rangle$ with respect to α. Let $r, s, t \in V$, and let $\alpha(t) = l_1, \ldots, l_k$. Then we have that $(i, j) \in F(r, s, t)$ if there is a set $H \in E(r, s)$ such that

- $H \cap L_{l_{i'}} = \emptyset$ for $i' < i$ and $H \cap U_{l_{j'}} = \emptyset$ for $j' < j$,
- $H \cap L_{l_i} \neq \emptyset$, if $i \leq m$, and $H \cap U_{l_j}$, if $j \leq m$.

It can be shown that G' is idempotent, nonaccepting, and corresponds to w.

Assume that there exist a pair assignment α and a nonaccepting idempotent pair graph $G = \langle V, F \rangle$ with respect to α such that G corresponds to w via the sequence $m_0, m_1, \ldots$. Consider any T-run $\mathbf{s}$ over w such that $lim(\mathbf{s}) = H$. Since V is finite, there is a state $s \in V$ such that $s = s_{m_j}$ for infinitely many j's. Let $\alpha(s) = l_1, \ldots, l_k$. By idempotence, there is a pair $(i, j) \in F(s, s, s)$ such that

- $H \cap L_{l_{i'}} = \emptyset$ for $i' < i$ and $H \cap U_{l_{j'}} = \emptyset$ for $j' < j$,
- $H \cap L_{l_i} \neq \emptyset$, if $i \leq m$, and $H \cap U_{l_j}$, if $j \leq m$.

Since G is nonaccepting, we have that $i < j$. It follows that H is nonaccepting. ∎

We are now in position to provide an exponential complementation construction for Streett automata.

Theorem 4.11: *Let A be a Streett automaton of size n over an alphabet Σ, then is a Büchi automaton A' with 2^{n^5} states such that $L_\omega(A') = \Sigma^\omega - L_\omega(A)$.*

Proof: A' first guesses a pair assignment α and a nonaccepting idempotent pair graph with respect to α. It then reads w and nondeterministically verifies that the pair graph corresponds to w. ∎

Michel's $2^{O(n \log n)}$ lower bound for complementation of Buchi automata [Mi88] can be extended to Streett automata. This leaves a marked gap between the upper and the lower bounds.

Finally, we use our complementation construction to deal with universality of Streett automata.

Proposition 4.12: *The universality problem for Streett automata is PSPACE-complete.*

5 Extended Temporal Logic

Having studied the computational properties of Büchi, Streett, and EL automata, we turn our attention back to *ETL*. As we have mentioned earlier, the satisfiability problem for ETL_B is PSPACE-complete. The question is whether we can use more succinct representation for the temporal connectives while retaining the polynomial-space upper bound.

Our approach is based on the automata-theoretic paradigm expounded in [WVS83, SVW87]. The basis of that paradigm is that computations can be viewed as words over the alphabet 2^{Prop} and *ETL* formulas always define ω-regular sets of computations. Thus, with every *ETL* formula φ we can associated a Büchi automaton A_φ such that $Mod(\varphi) = L_\omega(A_\varphi)$, i.e., the set of computations defined by φ is precisely the language accepted by A_φ. To check whether φ is satisfiable we just have to check whether A_φ is nonempty.[4] The crucial question is how big is the automaton A_φ; since negation corresponds to complementation and complementation involves at least an exponential blow up, one might suspect that the size of A_φ might be nonelementary in the length of φ. Fortunately, for ETL_B formulas, it was shown in [SVW87] that the size A_φ is exponential in the length of φ, from which a polynomial-space decision procedure is obtained. There are two major components to that construction. The first is running unboundedly many automata in parallel under central control, and the second is using the power of nondeterminism so complementation has to be applied only once.

We now turn our attention to EL automata.

Proposition 5.1: *Let φ be an ETL_{EL} formula of length n. Then there is a Büchi automaton A_φ of size $2^{2^{O(n)}}$ such that $Mod(\varphi) = L_\omega(A_\varphi)$. Furthermore, for every $n > 0$ there is an ETL_{EL} formula φ_n of length $O(n)$ such that $Mod(\varphi)$ is not accepted by any EL automaton with less than 2^{2^n}.*

Proof: The upper bound follows from Proposition 3.1 and the lower bound follows from Proposition 4.5. ∎

Using Propositions 4.6 and 5.1 we can charaterize the complexity of ETL_{EL}.

Theorem 5.2: *The satisfiability problem for ETL_{EL} is EXPSPACE-complete.*

The good news is that ETL_S is more tractable.

Proposition 5.3: *Let φ be an ETL_S formula of length n. Then there is a Büchi automaton A_φ of size $2^{O(n^5)}$ such that $Mod(\varphi) = L_\omega(A_\varphi)$. Furthermore, for every $n > 0$ there is an ETL_S formula φ_n of length $O(n)$ such that $Mod(\varphi)$ is not accepted by any EL automaton with less than 2^n.*

[4]The translation of a formula φ to an equivalent automaton A_φ has several other applications to program synthesis and verification. See [AS85, AS87, CY88, MP87, Va85, Va87, VW85, VW86].

Proof: The lower bound follows from Proposition 4.3. The upper bound requires an extension of the ideas in [SVW87]. For complementation we use the construction described in Section 4. In addition we have to show that we can run Streett automata in parallel and convert the result to a Büchi automaton with a single exponential blow-up. ∎

We finally get the desired complexity result by combining Propositions 5.3 and 4.12.

Theorem 5.4: *The satisfiability problem for ETL_S is PSPACE-complete.*

6 Concluding Remarks

Our investigation yielded the surprising result that even though Streett automata are exponentially more succinct that Büchi automata, one can use them to represent ω-regular languages in *ETL* while retaining the polynomial-space complexity of the decision procedure. We should note, however, that the decision procedure for ETL_B runs in time $2^{O(n \log n)}$ [Sa88, SVW87], while our decision procedure for ETL_S runs in time $2^{O(n^5)}$. The practical difference between these running times is significant enough to put a question mark on the practicality of ETL_S.

In conclusion we note that there is another yardstick of complexity for ω-automata that we have not considered here and that is the complexity of determinization. While it is known that Büchi automata can be determinized with an exponential blow-up [Sa88], it follows from Proposition 4.5 that determinization of EL automata requires a doubly exponential blow-up. The complexity of determinizing Streett automata is currently unknown.

Acknowledgements. We'd like to thank A. Pnueli and R. Heiman for very helpful discussions. In particular, A. Pnueli helped us to significantly simplify the complementation construction for Streett automata.

References

[AS85] Alpern, B., Schneider, F.B.: *Verifying temporal properties without using temporal logic*. Technical Report TR–85–723, Cornell University, Dec. 1985.

[AS87] Alpern, B., Schneider, F.B.: Proving Boolean combinations of deterministic properties. *Proc. 2nd IEEE Symp. on Logic in Computer Science*, Ithaca, 1987, pp. 131–137.

[Bu62] Büchi, J.R.: On a decision method in restricted second–order arithmetics. *Proc. Int'l Congr. on Logic, Method. and Phil. of Sci. 1960*, Stanford University Press, 1962, pp. 1–12.

[Ch74] Choueka, Y.: Theories of automata on ω–tapes – a simplified approach. *J. Computer and System Science* 8(1974), pp. 117–141.

[CY88] Courcoubetis, C., Yannakakis, M.: Verifying temporal properties of finite-state probabilistic programs. *Proc. 29th IEEE Symp. on Foundation of Computer Science*, Oct. 1988, pp. 338–345.

[EL87] Emerson, E.A., C.L. Lei: Modalities for model checking - branching time strikes back.

[E89] Emerson, E.A.: Temporal and Modal Logic. *Handbook of Theoretical Computer Science*, eds. J. van Leeuwen et. al., North-Holland Pub., in press, 1987.

[FK84] Francez, N., Kozen, D.: Generalized fair termination. *Proc. 11th ACM Symp. on Principles of Programming Languages*, Salt Lake City, 1984.

[Fr86] Francez, N.: *Fairness.* Text and Monographs in Computer Science, Springer–Verlag, 1986.

[Ga87] Galton, A.: *Temporal logics and their applications.* Academic Press, 1987.

[HR83] Halpern, J.Y., Reif, J.H.: The propositional dynamic logic of deterministic well-structured programs. *Theoretical Computer Science* 27(1983), pp. 127–165.

[Kr87] Kröger, F.: *Temporal logics of programs.* EATCS Monographs on Theoretical Computer Science, Springer–Verlag, 1987.

[LPZ85] Lichtenstein, O., Pnueli, A., Zuck L.: The glory of the past. *Proc. Workshop on Logics of Programs*, Brooklyn, Springer-Verlag, Lecture Notes in Computer Science 193, 1985, pp. 97–107.

[Mc66] McNaughton, R.: Testing and generating infinite sequences by a finite automaton. *Information and Control* 9(1966), pp. 521–530.

[Mi88] Michel, M.: Complementation is more difficult with automata on infinite words. Manuscript, 1988.

[MP87] Manna, Z., Pnueli, A.: Specification and verification of concurrent programs by ∀–automata. *Proc. 14 ACM Symp. on Principles of Programming Languages*, Munich, Jan. 1987, pp. 1–12.

[Pa81] Park, D.: Concurrency and automata on infinite sequences. *Proc. 5th GI Conf. on Theoretical Computer Science*, 1981, Springer–Verlag, LNCS 104, pp. 167–183.

[Pn77] Pnueli, A.: The temporal logic of programs. *Proc. 18th Symp. on Foundations of Computer Science*, 1977, pp. 46–57.

[Pn85] Pnueli, A: In transition from global to modular reasoning about programs. *Logics and Models of Concurrent Systems*, ed. K.R. Apt, Springer-Verlag, 1985, pp. 123–144.

[Pn86] Pnueli, A: Applications of temporal logic to the specification and verification of reactive systems - a survey of current trends. *Current Trends in Concurrency*, eds. J.W. de Bakker et al., Springer-Verlag, Lecture Notes in Computer Science 224, 1986, pp. 510–584.

[Ra69] Rabin, M.O.: Decidability of second-order theories and automata on infinite trees. *Trans. AMS* 141(1969), pp. 1–35.

[Sa88] Safra, S.: On the complexity of ω-automata. *Proc. 29th IEEE Symp. on Foundation of Computer Science*, Oct. 1988, pp. 319–327.

[SC85] Sistla, A.P., Clarke, E.M.: The complexity of propositional linear temporal logics. *J. ACM* 32(1985), pp. 733–749.

[St82] Streett, R.S.: Propositional dynamic logic of looping and converse is elementarily decidable. *Information and Control*, 54(1982), pp. 121–141.

[SVW87] Sistla, A.P., Vardi, M.Y., Wolper, P.: The complementation problem for Büchi automata with applications to temporal logic. Theoretical Computer Science 49(1987), pp. 217-237.

[To81] Thomas, W.: A combinatorial approach to the theory of ω-automata. *Information and Control* 48(1981), pp. 261–283.

[Va85] Vardi, M.Y.: Automatic verification of probabilistic concurrent finite-state programs. *Proc. 26th IEEE Symp. on Foundations of Computer Science*, Portland, 1985, pp. 327–338.

[Va87] Vardi, M.Y.: Verification of concurrent programs — the automata–theoretic framework. *Proc. 2nd IEEE Symp. on Logic in Computer Science*, Ithaca, 1987, pp. 167–176.

[VW85] Vardi, M.Y., Wolper, P.: Applications of temporal logic - an automata-theoretic perspective. Stanford University, CSLI, 1985.

[VW86] Vardi, M.Y., Wolper, P.: An automata-theoretic approach to automatic program verification. *Proc. IEEE Symp. on Logic in Computer Science*, Cambridge, 1986, pp. 332–344.

[VW88] Vardi, M.Y., Wolper, P.: Reasoning about infinite computation paths. IBM Research Report RJ 6209, April 1988.

[Wo83] Wolper, P.: Temporal logic can be more expressive. *Information and Control* 56(1983), pp. 72–99.

[WVS83] Wolper, P.L., Vardi, M.Y., Sistla, A.P.: Reasoning about infinite computation paths. *Proc. 24th IEEE Symp. on Foundation of Computer Science*, Tuscon, 1983, pp. 185–194.