Automata Construction for PSL

Doron Bustan[1] Dana Fisman[2*] John Havlicek[1]

dbustan@freescale.com dana.fisman@weizmann.ac.il john.havlicek@freescale.com

[1] Freescale Semiconductor, Inc.

[2] Weizmann Institute of Science, IBM Haifa Research Lab

May 10, 2005

1 Introduction

The language PSL [Acc04] is a temporal logic standardized by the Accellera standards organization and currently undergoing the process of becoming an IEEE standard. The core of PSL, denoted here LTL_WR, is an extension of the linear temporal logic LTL. The extension takes two orthogonal directions. In one direction the logic is interpreted over finite, possibly truncated, as well as infinite words. Truncated words are words that are finite, but not necessarily maximal. Reasoning over truncated words (as well as maximal words) is important for incomplete verification methods such as simulation and bounded model checking as well as for supporting abort/reset operators [EFH$^+$03]. In another direction, new basic formulas and operators are added to the language. The new basic formulas are weak and strong regular expressions [EFH05], and the new operators are suffix conjunction/implication that combine regular expressions with other formulas.

In this document we provide automata construction for LTL_WR. We show that for every LTL_WR formula φ there exists a Büchi automaton whose size is exponential in the size of φ. In addition, we classify the complexity of model checking simple properties of the regular expression layer. The suggested constructions can be used in the process of model checking PSL properties using the automata-theoretic approach.

A construction for a logic extending LTL with suffix conjuction/implication operators, interpreted over infinite words appears in [Fla04]. A construction for a logic extending LTL with suffix conjunction/implication operators, interpreted over finite/infinite maximal words appears in [Kar03]. A construction for reasoning over truncated words, via reset

*The work of this author was supported in part by the European Commission (FP6 STREP PROSYD contract no. 507219) and carried out at the John von-Neumann Minerva Center for the Verification of Reactive Systems.

operators appears in [ABKV03]. Our contribution is twofold. First, we are the first to show a construction for weak regular expressions. Second, we are the first to show a complete construction for LTL_WR, synthesizing previous results into a cohesive whole.

2 Preliminaries

2.1 The Logic

Below we provide a formal definition of the temporal logic LTL_WR, an extension of LTL with weak and strong REs and a suffix conjunction/implication operator, interpreted over finite (possibly truncated) as well as infinite words.

Notations

We denote a letter from a given alphabet Σ by ℓ and an empty, finite, or infinite word from Σ by u, v, or w (possibly with subscripts). The *concatenation* of u and v is denoted by uv. If u is infinite, then $uv = u$. The empty word is denoted by ϵ, so that $w\epsilon = \epsilon w = w$. If $w = uv$ we say that u is a *prefix* of w, denoted $u \preceq w$, that v is a *suffix* of w, and that w is an *extension* of u, denoted $w \succeq u$.

We denote the *length* of word v as $|v|$. The empty word ϵ has length 0, a finite word $v = (\ell_0\ell_1\ell_2\cdots\ell_n)$ has length $n+1$, and an infinite word has length ω. We use i, j, and k to denote non-negative integers. For $i < |v|$ we use v^i to denote the $(i+1)^{st}$ letter of v (since counting of letters starts at zero), and we denote by $v^{i\cdot\cdot}$ the suffix of v starting at v^i. When $i \leq j \leq |v|$, we denote by $v^{i..j}$ the finite sequence of letters starting from v^i and ending in v^j. That is, $v^{i..j} = v^i v^{i+1}\cdots v^j$.

We denote by ℓ^k the word of length k, each letter of which is ℓ, and by ℓ^ω the infinite-length word, each letter of which is ℓ. We use Σ^* to denote the set of all finite words over Σ, and Σ^ω to denote the set of all infinite words over Σ. We use Σ^∞ to denote the set of all finite and infinite words over Σ.

An *unlabeled tree* is a prefix-closed subset of $\mathbb{N}^*$. Elements of the tree are referred to as *nodes*. For a node $t \in \mathbb{N}^*$ we refer to $|t|$ as the *depth* of t. The node ϵ is called the *root*. For nodes t_1 and t_2 such that $t_1 \prec t_2$ we say that t_2 is a *descendant* of t_1. If t_2 is a descendant of t_1 and $|t_2| = |t_1| + 1$ we say that t_2 is a *child* of t_1. A Σ-*labeled tree* τ is a pair $\langle \mathtt{T}, \mathtt{t}\rangle$ where $\mathtt{T}$ is an unlabeled tree and $\mathtt{t} : \mathtt{T} \to \Sigma$ maps nodes of $\mathtt{T}$ to symbols in Σ.

Syntax

The logic LTL_WR is defined over boolean expressions as well as regular expressions. We assume a given set $\mathbb{B}$ of boolean expressions defined over a set $\mathbb{P}$ of atomic propositions. Below we define Regular Expressions (REs) using $\cdot$, $\cup$, and * for concatenation, union and Kleene-closure, respectively.

Definition 1 (REs)

- *Every boolean expression* $b \in \mathbb{B}$ *is an* RE.
- *If* r, r_1 *and* r_2 *are* REs*, then the following are* REs*:* • $r_1 \cdot r_2$ • $r_1 \cup r_2$ • r^*

Definition 2 (LTL_WR **formulas**)

- *If* b *is a boolean expression then* $b!$ *is an* LTL *formula.*
- *If* φ *and* ψ *are* LTL_WR *formulas, and* r *is an* RE*, then the following are* LTL_WR *formulas:*

• $\neg\varphi$ • $\varphi \wedge \psi$ • $\mathsf{X!}\ \varphi$ • $[\varphi\ \mathsf{U}\ \psi]$ • $r \diamondsuit\!\!\rightarrow \varphi$ • r

Additional operators are defined as syntactic sugaring of the above operators:

• $\varphi \vee \psi \stackrel{\text{def}}{=} \neg(\neg\varphi \wedge \neg\psi)$ • $[\varphi\ \mathsf{W}\ \psi] \stackrel{\text{def}}{=} \neg[\neg\psi \mathsf{U} (\neg\varphi \wedge \neg\psi)]$ • $\mathsf{X}\ \varphi \stackrel{\text{def}}{=} \neg(\mathsf{X!}\ \neg\varphi)$
• $r! \stackrel{\text{def}}{=} r \diamondsuit\!\!\rightarrow true$ • $r \mapsto \varphi \stackrel{\text{def}}{=} \neg(r \diamondsuit\!\!\rightarrow \neg\varphi)$ • $b \stackrel{\text{def}}{=} \neg(\neg b)!$

We refer to the operators $r!$ and r as *strong* and *weak* regular expressions, respectively. We refer to the $\diamondsuit\!\!\rightarrow$ operator as the *suffix conjunction operator*, since $r \diamondsuit\!\!\rightarrow \varphi$ (read r "suffix-and" φ) demands that there exist a non-empty prefix of the path tightly satisfying r *and* that the suffix starting at the last letter of the prefix satisfy φ. We refer to its dual, the $\mapsto$ operator, as the *suffix implication operator*, since $r \mapsto \varphi$ (read r "suffix-implies" φ) requires that *if* there exists a non-empty prefix of the path tightly satisfying r *then* the suffix starting at the last letter of the prefix should satisfy φ. The operator $\diamondsuit\!\!\rightarrow$ exists in the temporal logic ForSpec [AFF+02] under the names follows_by and seq. The operator $\mapsto$ exists in the temporal logic Sugar [BBL98] under the notation $r(\varphi)$. These operators are essentially the "diamond" and "box" modalities of PDL [FL79], respectively.

Semantics

The semantics of LTL_WR is defined with respect to a non-empty set of atomic propositions $\mathbb{P}$, a set of boolean expressions $\mathbb{B}$ over $\mathbb{P}$, which we identify with $2^{2^{\mathbb{P}}}$, and the set of regular expressions over $\mathbb{B}$.

Although we are interested in the set of words over $2^{\mathbb{P}}$ which satisfy a given LTL_WR formula, it is convenient to define the semantics of LTL_WR over an enhanced set of words. Let Σ denote the alphabet $2^{\mathbb{P}} \cup \{\top, \bot\}$. The semantics of LTL_WR is defined with respect to words over Σ. Note, however, that since computations of systems are words over $2^{\mathbb{P}}$, our interest is still focused on such words. We refer to words over $2^{\mathbb{P}}$ as *natural* words. We use $\overline{w}$ to denote the word obtained by replacing every $\top$ with a $\bot$ and vice versa. We call $\overline{w}$ the *dual* of w. For a set of word W we use $\overline{W}$ to denote the set $\{\overline{w} \mid w \in W\}$.

The semantics of LTL_WR is defined inductively with respect to finite/infinite (possibly empty) words over Σ, using as the base case the semantics of boolean expressions over letters in Σ and of regular expressions over finite words over Σ. We use $\models_{\text{PSL}}$, $\models\!\!\models_{\text{PSL}}$ and $\models\!\!\models\!\!\models_{\text{PSL}}$ to denote boolean satisfaction, tight satisfaction, and formula satisfaction, respectively.

For a boolean expression $b \in \mathbb{B}$ and a letter $\ell \in \Sigma$ we define the boolean satisfaction relation $\models_{\text{PSL}}$ as follows. Let $b \in \mathbb{B}$. For $\ell \in 2^{\mathbb{P}}$, we define $\ell \models_{\text{PSL}} b \iff \ell \in b$. We define $\top \models_{\text{PSL}} b$ and $\bot \not\models_{\text{PSL}} b$.

Definition 3 (RE Tight Satisfaction) *Let v denote a finite (possibly empty) word over Σ; b denote a boolean expression; and r, r_1, and r_2 denote* REs. *The notation $v \models\!\!\models_{\text{PSL}} r$ means that v tightly satisfies r. The relation $\models\!\!\models_{\text{PSL}}$ is defined as follows:*

- $v \models\!\!\models_{\text{PSL}} b \iff |v| = 1$ *and* $v^0 \models_{\text{PSL}} b$
- $v \models\!\!\models_{\text{PSL}} r_1 \cdot r_2 \iff \exists v_1, v_2$ *s.t.* $v = v_1 v_2$ *and* $v_1 \models\!\!\models_{\text{PSL}} r_1$ *and* $v_2 \models\!\!\models_{\text{PSL}} r_2$
- $v \models\!\!\models_{\text{PSL}} r_1 \cup r_2 \iff v \models\!\!\models_{\text{PSL}} r_1$ *or* $v \models\!\!\models_{\text{PSL}} r_2$
- $v \models\!\!\models_{\text{PSL}} r^* \iff$ *either* $v = \epsilon$ *or* $\exists v_1, v_2$ *s.t.* $v_1 \neq \epsilon$, $v = v_1 v_2$, $v_1 \models\!\!\models_{\text{PSL}} r$ *and* $v_2 \models\!\!\models_{\text{PSL}} r^*$

For a regular expression r, we use $\mathbb{S}(r)$ to denote the set $\{w \in \Sigma^* \mid w \models\!\!\models_{\text{PSL}} r\}$

Definition 4 (Formula Satisfaction) *Let v denote a word over Σ; b a boolean expression; r an* RE*; and φ and ψ* LTL_WR *formulas. The notation $v \models\!\!\models\!\!\models_{\text{PSL}} \varphi$ means that v satisfies φ. The relation $\models\!\!\models\!\!\models_{\text{PSL}}$ is defined as follows:*[1]

1. $v \models\!\!\models\!\!\models_{\text{PSL}} b! \iff |v| > 0$ *and* $v^0 \models_{\text{PSL}} b$
2. $v \models\!\!\models\!\!\models_{\text{PSL}} \neg\varphi \iff \overline{v} \not\models\!\!\models\!\!\models_{\text{PSL}} \varphi$
3. $v \models\!\!\models\!\!\models_{\text{PSL}} \varphi \wedge \psi \iff v \models\!\!\models\!\!\models_{\text{PSL}} \varphi$ *and* $v \models\!\!\models\!\!\models_{\text{PSL}} \psi$
4. $v \models\!\!\models\!\!\models_{\text{PSL}} \mathsf{X!}\,\varphi \iff |v| > 1$ *and* $v^{1..} \models\!\!\models\!\!\models_{\text{PSL}} \varphi$
5. $v \models\!\!\models\!\!\models_{\text{PSL}} [\varphi \mathsf{U} \psi] \iff \exists\ 0 \leq k < |v|$ *s.t.* $v^{k..} \models\!\!\models\!\!\models_{\text{PSL}} \psi$ *and* $\forall\ 0 \leq j < k,\ \ v^{j..} \models\!\!\models\!\!\models_{\text{PSL}} \varphi$
6. $v \models\!\!\models\!\!\models_{\text{PSL}} r \diamondsuit\!\!\rightarrow \psi \iff \exists\ 0 \leq j < |v|$ *s.t.* $v^{0..j} \models\!\!\models_{\text{PSL}} r$ *and* $v^{j..} \models\!\!\models\!\!\models_{\text{PSL}} \psi$
7. $v \models\!\!\models\!\!\models_{\text{PSL}} r \iff \forall$ *finite* $u \preceq v,\ \ u\top^\omega \models\!\!\models\!\!\models_{\text{PSL}} r!$

For an LTL_WR formula φ, we use $[\![\varphi]\!]$ to denote the set $\{w \in \Sigma^\infty \mid w \models\!\!\models\!\!\models_{\text{PSL}} \varphi\}$.

[1] The semantics of the LTL operators are the standard ones. The operators of LTL are all the operators in definition 2 except for $\diamondsuit\!\!\rightarrow$ and r.

2.2 Automata-Theoretic Approach

We recall some basic notions from automata theory. Since LTL_WR is interpreted over *finite* as well as infinite words, we use a version of Büchi automata that accepts both finite and infinite words.

Definition 5 (Büchi automaton) *A Büchi automaton over finite/infinite words is a tuple* $B = (\Sigma, S, I, \rho, F, A)$ *where*

- Σ *is a finite nonempty* alphabet
- S *is a finite nonempty set of* states
- $I \subseteq S$ *is a nonempty set of* initial *states*
- $\rho : S \times \Sigma \to 2^S$ *is a* transition *function*
- $F \subseteq S$ *is a set of* final *states*
- $A \subseteq S$ *is a set of* accepting *states*

We often regard ρ as a relation $\rho \subseteq S \times \Sigma \times S$. A *run* σ of B on an infinite word $w = \ell_0\ell_1 \ldots$ is an infinite sequence $s_0, s_1, \ldots$ where $s_0 \in I$ and $s_{i+1} \in \rho(s_i, \ell_i)$ for all $i \geq 0$. A *run* σ of B on a finite word $w = \ell_0\ell_1 \ldots \ell_n$ is a finite sequence $s_0, s_1, \ldots, s_{n+1}$ where $s_0 \in I$ and either $w = \epsilon$ or $s_{i+1} \in \rho(s_i, \ell_i)$ for all $0 \leq i \leq n$. We define by $lim(\sigma)$ the set $\{s \mid s = s_i$ for infinitely many i's $\}$. An infinite run is *accepting* if there is some accepting state that repeats infinitely often, i.e. if $lim(\sigma) \cap A \neq \emptyset$. A finite run is *accepting* if the last state is a final state, i.e. if $s_{|w|} \in F$. The word w is *accepted* by B if there is an accepting run of B on w. We denote by $\mathcal{L}(B)$ the set of words w such that there exists an accepting run of B on w. For $s \in S$ we use B^s to denote the automaton $(\Sigma, S, \{s\}, \rho, F, A)$.

Definition 6 (Alternating Büchi automaton) *An Alternating Büchi automaton over finite/infinite words is a tuple* $B = (\Sigma, S, \{s_0\}, \rho, F, A)$ *where*

- Σ *is a finite nonempty* alphabet
- S *is a finite nonempty set of* states
- $s^0 \in S$ *is the* initial *state.*
- $\rho : S \times \Sigma \to \mathcal{B}^+(S)$ *is a* transition *function where* $\mathcal{B}^+(S)$ *is the set of boolean formulas obtained by application of* $\wedge$ *and* $\vee$ *to element in* S. [2]
- $F \subseteq S$ *is a set of* final *states*

[2] We assume *true* and *false* are in $\mathcal{B}^+(s)$.

- $A \subseteq S$ *is a set of* accepting *states*

A *run* of B on an infinite word $w = \ell_0\ell_1 \ldots$ is a (possibly infinite) S-labeled tree $\tau = \langle \mathtt{T}, \mathtt{t} \rangle$ such that $\mathtt{t}(\epsilon) = s^0$ and for every node $t \in \tau$, t has at most $|S|$ children and, if $|t| = i$ and $\mathtt{t}(t) = s$, then the children of t satisfy $\rho(s, \ell_i)$ (i.e. if $t_1, \ldots, t_k$ are t's children then $\{\mathtt{t}(t_1), \ldots, \mathtt{t}(t_k)\}$ satisfies $\rho(s, \ell_i)$). A *run* of B on a finite word $w = \ell_0\ell_1 \ldots \ell_n$ is a finite S-labeled tree $\tau = \langle \mathtt{T}, \mathtt{t} \rangle$ such that $\mathtt{t}(\epsilon) = s^0$ and for every node $t \in \tau$, t has at most $|S|$ children and, if $|t| = i < n$ and $\mathtt{t}(t) = s$, then the children of t satisfy $\rho(s, \ell_i)$. If w is infinite, then a run tree τ is *accepting* if every branch of infinite depth has infinitely many labels in A. If w is finite, then a run tree τ is *accepting* if all nodes at depth $|w|$ are labeled by states in F. The word w is *accepted* by B if there is an accepting run tree of B on w. We denote by $\mathcal{L}(B)$ the set of words w such that there exists an accepting run tree of B on w.

Proposition 2.1 *Let $\mathcal{A}$ be an alternating Büchi automaton on finite/infinite words with n states. There exists a Büchi automaton $\mathcal{B}$ on finite/infinite words with $2^{O(n)}$ states that accepts the same language.*

Proof: The proof follows the construction of Miyano and Hayashi [MH84] for the same proposition restricted to infinite words. Let $B = \langle \Sigma, S, \{s_0\}, \delta, F, A \rangle$ be an alternating Büchi automaton on finite/infinite words. We define the Büchi automaton (on finite/infinite words) $B_n = \langle \Sigma, S_n, \{(\{s_0\}, \{s_0\} \setminus A)\}, \delta_n, F_n, A_n \rangle$ as follows:

- $S_n \subseteq 2^S \times 2^S$ is the set of consistent pairs with respect to B, where a pair $(Q, P) \in 2^S \times 2^S$ is said to be consistent with respect to B if $P \subseteq Q \setminus A$.
- (Q', P') is in $\delta_n((Q, P), \sigma)$ iff $Q' \models \bigwedge_{s \in Q} \delta(s, \sigma)$ and either:
  - $P = \emptyset$ and $P' = Q' \setminus A$ or
  - $P \neq \emptyset$ and there exists a set $Y \subseteq Q'$ such that $Y \models \bigwedge_{s \in P} \delta(s, \sigma)$, and $P' = Y \setminus A$.
- $F_n = \{(Q, P) \in S_n \mid Q \subseteq F\}$.
- $A_n = \{(Q, P) \in S_n \mid P = \emptyset\}$.

Following [MH84], we have that for every infinite word w, $w \in \mathcal{L}(B)$ iff $w \in \mathcal{L}(B_n)$. Thus, we need to prove the following lemma:

Lemma 2.2 *Let w be a finite word. Then $w \in \mathcal{L}(B)$ iff $w \in \mathcal{L}(B_n)$.*

Proof: First direction: Let w be a finite word in $\mathcal{L}(B)$. We prove that $w \in \mathcal{L}(B_n)$. Let $\tau = \langle T, \mathtt{t} \rangle$ be an accepting running tree of B on w. We define a running trace $(Q_0, P_0), (Q_1, P_1), \ldots (Q_{|w|}, P_{|w|})$ such that $Q_i = \{s \mid \mathtt{t}(t) = s$ for some $t \in \tau$ with $|t| = i\}$, and $P_i = Q_i \setminus A$. We prove that this trace is an accepting running trace of B_n on w.

– Since $\mathtt{t}(\epsilon) = s_0$, we have that $(Q_0, P_0) = (\{s_0\}, \{s_0\} \setminus A)$ which is the initial state of B_n.

– Let $s \in Q_i$ for $i < |w|$. Then, there exists a node t of τ such that $\mathtt{t}(t) = s$ and $|t| = i$. Let $X = \{\mathtt{t}(t') \mid t' \text{ is a child of } t\}$. Then, $X \| \models \delta(s, w^i)$ and $X \subseteq Q_{i+1}$, thus $Q_{i+1} \| \models \delta(s, w^i)$. This implies that for every $s \in Q_i$ we have that $Q_{i+1} \| \models \delta(s, w^i)$. Since $P_i \subseteq Q_i$, we can always select $Y = Q_{i+1}$ and $P_{i+1} = Y \setminus A$. Then for every $s \in P_i$, $Y \| \models \delta(w, w^i)$. Thus, we have that $(Q_{i+1}, P_{i+1}) \in \delta_n((Q_i, P_i))$.

– Since τ is accepting, every branch of τ of depth $|w|$ ends in a node t such that $\mathtt{t}(t) \in F$. This implies that every state $s \in Q_{|w|}$ is in F. Thus, $(Q_{|w|}, P_{|w|}) \in F_n$.

Second direction: Let $(Q_0, P_0), (Q_1, P_1), \ldots (Q_{|w|}, P_{|w|})$ be an accepting running trace of B_n on w. We define a tree $\tau = \langle T, \mathtt{t} \rangle$ such that $\mathtt{t}(\epsilon) = s_0$ and every node t such that $|t| = i < |w|$ has $|Q_{i+1}|$ children such that for every $s \in Q_{i+1}$ the node t has a child t' with $\mathtt{t}(t') = s$. We prove that τ is an accepting tree of B on w.

– s_0 is the initial state of B.

– Let t be a node of τ such that $|t| = i$ for $i < |w|$. Let $\mathtt{t}(t) = s$. Then, $s \in Q_i$, and the definition of δ_n implies that $Q_{i+1} \| \models \delta(s, w^i)$. Since $Q_{i+1} = \{\mathtt{t}(t') \mid t' \text{ is a child of } t\}$, we have that $\{\mathtt{t}(t') \mid t' \text{ is a child of } t\} \| \models \delta(\mathtt{t}(t), w^i)$.

– Since $Q_{|w|} \subseteq F_n$, we have that every branch of τ of depth $|w|$ ends in a node t such that $\mathtt{t}(t) \in F$.

□

This completes the proof of the proposition. □

As a measure of efficiency we use the classification of Büchi automata as suggested by Muller et al. [MSS88] and adopted by [BRS99]. The classification uses the notion of a *weak* Büchi automaton and a *terminal* Büchi automaton.[3] As stated by the claim below, weak and terminal Büchi automata are especially efficient to model check.

Definition 7 (weak, terminal (see e.g [BRS99])) *Let $B = (\Sigma, S, I, \rho, F, A)$ be a Büchi automaton.*

- *B is a* weak Büchi automaton *if there exists a partition of the set of states S into components S_i and a partial ordering $\leq$ on these sets, such that*

  1. *for each $s_i \in S_i$, $s_j \in S_j$, if $\exists \ell \in \Sigma$ s.t. $s_j \in \rho(s_i, \ell)$ then $S_j \leq S_i$, and*
  2. *for each S_i, either $S_i \cap A = \emptyset$, in which case S_i is a rejecting component, or $S_i \subseteq A$, in which case S_i is an accepting component.*

[3] Note that the term *weak* used for a Büchi automaton has nothing to do with the term *weak* used for REs. We apologize for the potential confusion.

- *B is a* terminal Büchi automaton *if it is a weak Büchi automaton such that the components of the partition contained in A are maximal elements of the partial order.*

Claim 2.3 (see e.g [BRS99]) *Let $B = (\Sigma, S, I, \rho, F, A)$ be a Büchi automaton.*

- *The language of a weak Büchi automaton is empty iff $\mathsf{AG}\,\mathsf{AF}\,\neg A$*
- *The language of a terminal Büchi automaton is empty iff $\mathsf{AG}\,\neg A$*

3 Constructions and Proofs

Below we provide an automata-theoretic approach to the verification problem of LTL_WR formulas.

3.1 Automata construction for LTL over finite as well as infinite words

Since we are interested in finite as well as infinite words, we check first that the well-known singly-exponential complexity for constructing a Büchi automaton for the ω-language of an LTL formula still applies when the automaton runs on both finite and infinite words.

We follow the construction of an alternating Büchi automaton for an LTL formula φ as given in [Var95, Theorem 22]. Our construction differs from that of [Var95] in several ways.

1. Our automaton is designed to run on both finite and infinite words, while that of [Var95] is only for infinite words. Some care is needed to handle the nexttime operators on finite words since the semantics looks ahead in order to know whether the condition $|w| > 1$ is satisfied. Our construction uses a simple subautomaton to distinguish nonempty from empty words.

2. The construction of [Var95] introduces an automaton state for each subformula and its negation and relies on a dualization operator in the definition of the transition relation. We achieve a similar effect by including only subformulas and using a negation parity bit in the state. The negation parity of a subformula indicates the number of enclosing negations, modulo 2. Our construction gives the dual forms of the transition relation explicitly rather than relying on the dualizing operator.

3. Our construction works over the extended alphabet $\Sigma = 2^{\mathbb{P}} \cup \{\top, \bot\}$. This is accomplished simply by using the dual $\bar{\ell}$ of the input letter ℓ when testing boolean satisfaction for the transition relation on a boolean subformula with negation parity 1.

Proposition 3.1 *Given an* LTL *formula φ, one can build an alternating Büchi automaton $B_\varphi = (\Sigma, S, I, \rho, F, A)$ where $|S|$ is in $O(|\varphi|)$ and $\mathcal{L}(B_\varphi)$ is exactly the set of (finite and infinite) words satisfying the formula φ.*

Proof: Let φ be an LTL formula. Let S be the set of pairs (p, ψ) where either ψ is a subformula of φ or $\psi \in \{\text{TRUE}, \text{FALSE}, \text{NONEMPTY}\}$, and $p \in \{0, 1\}$ denotes the negation parity, with $p = 1$ indicating an odd number of enclosing negations. The initial state is $(0, \varphi)$.

The set A of accepting states is produced by the following rules:

1. A contains all states of the form:
   $(1, \text{NONEMPTY})$, $(0, \text{TRUE})$, $(1, \text{FALSE})$ $(1, b!)$, $(1, \mathsf{X}!\psi)$, $(1, [\psi \mathsf{U} \vartheta])$.
2. $(p, \neg\psi) \in A$ iff $(1 - p, \psi) \in A$.
3. $(0, \psi \wedge \vartheta) \in A$ iff both $(0, \psi) \in A$ and $(0, \vartheta) \in A$.
   $(1, \psi \wedge \vartheta) \in A$ iff either $(1, \psi) \in A$ or $(1, \vartheta) \in A$.

The set F of final states is equal to A.

The transition relation ρ is defined as follows:

- $\rho((p, \text{NONEMPTY}), \ell) = (p, \text{TRUE})$
- $\rho((p, \text{TRUE}), \ell) = (p, \text{TRUE})$
  $\rho((p, \text{FALSE}), \ell) = (p, \text{FALSE})$
- $\rho((0, b!), \ell) = (0, \ell \Vdash_{\text{PSL}} b)$
  $\rho((1, b!), \ell) = (1, \overline{\ell} \Vdash_{\text{PSL}} b)$
- $\rho((0, \psi \wedge \vartheta), \ell) = \rho((0, \psi), \ell) \wedge \rho((0, \vartheta), \ell)$
  $\rho((1, \psi \wedge \vartheta), \ell) = \rho((1, \psi), \ell) \vee \rho((1, \vartheta), \ell)$
- $\rho((0, \neg\psi), \ell) = \rho((1, \psi), \ell)$
  $\rho((1, \neg\psi), \ell) = \rho((0, \psi), \ell)$
- $\rho((0, \mathsf{X}!\psi), \ell) = (0, \text{NONEMPTY}) \wedge (0, \psi)$
  $\rho((1, \mathsf{X}!\psi), \ell) = (1, \text{NONEMPTY}) \vee (1, \psi)$
- $\rho((0, [\psi \mathsf{U} \vartheta]), \ell) = \rho((0, \vartheta), \ell) \vee (\rho((0, \psi), \ell) \wedge (0, [\psi \mathsf{U} \vartheta]))$
  $\rho((1, [\psi \mathsf{U} \vartheta]), \ell) = \rho((1, \vartheta), \ell) \wedge (\rho((1, \psi), \ell) \vee (1, [\psi \mathsf{U} \vartheta]))$

Lemma 3.2 *If ψ is a subformula of φ, let $A^{(p,\psi)}$ denote the subautomaton of this construction obtained by taking (p, ψ) as initial state.*

1. $\mathcal{L}(A^{(0,\psi)}) = [\![\psi]\!]$.
2. $\mathcal{L}(A^{(1,\psi)}) = \Sigma^\infty - \overline{[\![\psi]\!]} = [\![\neg\psi]\!]$.

Proof: By induction over the subformulas of φ. Note that for any $L \subseteq \Sigma^\infty$,

$$\Sigma^\infty - \overline{L} = \overline{\Sigma^\infty - L}$$

Note also that for any ψ,

$$\Sigma^\infty - \overline{[\![\psi]\!]} = [\![\neg\psi]\!]$$

Let v, w denote elements of Σ^∞.

- $\mathcal{L}(A^{(0,\text{NONEMPTY})}) = \{v : |v| > 0\}$.
  $\mathcal{L}(A^{(1,\text{NONEMPTY})}) = \{\epsilon\}$.
- $\mathcal{L}(A^{(0,b!)})$
  $= $ [$(0, b!)$ is not accepting]
  $\{\ell w : \ell \models_{\text{PSL}} b\}$
  $= [\![b!]\!]$
  $\mathcal{L}(A^{(1,b!)})$
  $= $ [$(1, b!)$ is accepting]
  $\{\epsilon\} \cup \{\ell w : \overline{\ell} \not\models_{\text{PSL}} b\}$
  $= \Sigma^\infty - \overline{[\![b!]\!]}$
- $[\![\neg\psi]\!]$
  $= \Sigma^\infty - \overline{[\![\psi]\!]}$
  $= $ [induction]
  $\mathcal{L}(A^{(1,\psi)})$
  $= \mathcal{L}(A^{(0,\neg\psi)})$
  $\mathcal{L}(A^{(1,\neg\psi)})$
  $= \mathcal{L}(A^{(0,\psi)})$
  $= $ [induction]
  $[\![\psi]\!]$
  $= \Sigma^\infty - \overline{[\![\neg\psi]\!]}$
- $[\![\psi \wedge \vartheta]\!]$
  $= [\![\psi]\!] \cap [\![\vartheta]\!]$
  $= $ [induction]
  $\mathcal{L}(A^{(0,\psi)}) \cap \mathcal{L}(A^{(0,\vartheta)})$
  $= \mathcal{L}(A^{(0,\psi\wedge\vartheta)})$
  $\mathcal{L}(A^{(1,\psi\wedge\vartheta)})$
  $= \mathcal{L}(A^{(1,\psi)}) \cup \mathcal{L}(A^{(1,\vartheta)})$

$= $ [induction]
$(\Sigma^\infty - \overline{[\![\psi]\!]}) \cup (\Sigma^\infty - \overline{[\![\vartheta]\!]})$
$= \Sigma^\infty - (\overline{[\![\psi]\!]} \cap \overline{[\![\vartheta]\!]})$
$= \Sigma^\infty - (\overline{[\![\psi \wedge \vartheta]\!]})$

– $\mathcal{L}(A^{(0,\mathsf{X}!\psi)})$

$= [(0, \mathsf{X}!\psi)$ is not accepting; $\mathcal{L}(A^{(0,\text{NONEMPTY})}) = \{v : |v| > 0\}]$
$\{\ell w : |w| > 0$ and $w \in \mathcal{L}(A^{(0,\psi)})\}$
$=$ [induction]
$\{\ell w : |w| > 0$ and $w \in [\![\psi]\!]\}$
$= \{v : |v| > 1$ and $v^{1\cdot\cdot} \models_{\text{PSL}} \psi\}$
$= [\![\mathsf{X}!\psi]\!]$

$\mathcal{L}(A^{(1,\mathsf{X}!\psi)})$

$= [(1, \mathsf{X}!\psi)$ is accepting; $\mathcal{L}(A^{(1,\text{NONEMPTY})}) = \{\epsilon\}]$
$\{v : |v| \leq 1$ or $v^{1\cdot\cdot} \in \mathcal{L}(A^{(1,\psi)})\}$
$=$ [induction]
$\{v : |v| \leq 1$ or $v^{1\cdot\cdot} \in [\![\neg\psi]\!]\}$
$= \{v : |v| \leq 1$ or $\overline{v}^{1\cdot\cdot} \not\models_{\text{PSL}} \psi\}$
$= \Sigma^\infty - \overline{\{v : |v| > 1 \text{ and } v^{1\cdot\cdot} \models_{\text{PSL}} \psi\}}$
$= \Sigma^\infty - \overline{[\![\mathsf{X}!\psi]\!]}$

– Let $v \in \mathcal{L}(A^{(0,[\psi\mathsf{U}\vartheta])})$. Suppose $v \notin \mathcal{L}(A^{(0,\vartheta)})$. Then $v \in \mathcal{L}(A^{(0,\psi)})$ and, since $(0, [\psi\mathsf{U}\vartheta])$ is not accepting, $|v| > 0$ and $v^{1\cdot\cdot} \in \mathcal{L}(A^{(0,[\psi\mathsf{U}\vartheta])})$. Thus,
$v \in \mathcal{L}(A^{(0,[\psi\mathsf{U}\vartheta])})$
$\iff v \in \mathcal{L}(A^{(0,\vartheta)})$ or $(v \in \mathcal{L}(A^{(0,\psi)})$ and $|v| > 0$ and $v^{1\cdot\cdot} \in \mathcal{L}(A^{(0,[\psi\mathsf{U}\vartheta])}))$
$\iff$ [induction]
$v \models_{\text{PSL}} \vartheta$ or $(v \models_{\text{PSL}} \psi$ and $|v| > 0$ and $v^{1\cdot\cdot} \in \mathcal{L}(A^{(0,[\psi\mathsf{U}\vartheta])}))$

If there does not exist $0 \leq k < |v|$ such that $v^{k\cdot\cdot} \models_{\text{PSL}} \vartheta$, then, since $(0, [\psi\mathsf{U}\vartheta])$ is not accepting, there is no accepting run of $A^{(0,[\psi\mathsf{U}\vartheta])}$ on v. Therefore,
$v \in \mathcal{L}(A^{(0,[\psi\mathsf{U}\vartheta])})$
$\iff$ there exists $0 \leq k < |v|$ such that $v^{k\cdot\cdot} \models_{\text{PSL}} \vartheta$ and for all $0 \leq j < k$, $v^{j\cdot\cdot} \models_{\text{PSL}} \psi$
$\iff v \models_{\text{PSL}} [\psi\mathsf{U}\vartheta]$

$v \in \mathcal{L}(A^{(1,[\psi\mathsf{U}\vartheta])})$
$\iff v \in \mathcal{L}(A^{(1,\vartheta)})$ and $(v \in \mathcal{L}(A^{(1,\psi)})$ or $v \in \mathcal{L}(A^{(1,[\psi\mathsf{U}\vartheta])}))$
$\iff$ [induction]
$v \models_{\text{PSL}} \neg\vartheta$ and $(v \models_{\text{PSL}} \neg\psi$ or $v \in \mathcal{L}(A^{(1,[\psi\mathsf{U}\vartheta])}))$

Since $(1, [\psi \mathsf{U} \vartheta])$ is accepting, there need not exist $0 \leq k < |v|$ such that $v^{k..} \in [\![\neg\psi]\!]$. Thus,
$v \in \mathcal{L}(A^{(1,[\psi \mathsf{U} \vartheta])})$
$\iff$ $v \models_{\text{PSL}} \neg\vartheta \mathsf{W} (\neg\psi \wedge \neg\vartheta)$
$\iff$ [duality of U, W over Σ]
$v \models_{\text{PSL}} \neg[\psi \mathsf{U} \vartheta]$

□

This completes the proof of the proposition.

□

Corollary 3.3 *Given an* LTL *formula φ, one can build a Büchi automaton $B_\varphi = (\Sigma, S, I, \rho, F, A)$ where $|S|$ is in $2^{O(|\varphi|)}$ and $\mathcal{L}(B_\varphi)$ is exactly the set of (finite and infinite) words satisfying the formula φ.*

Proof: Follows directly from Propositions 3.1 and 2.1. □

3.2 Automata Construction for LTL_WR

In this section we extend the construction of automata for LTL (over finite as well as infinite words) given in the previous section to all of LTL_WR. Recall from Definition 2 that the LTL_WR formulas are generated from the LTL formulas by adding the basic form r, where r is a regular expression, and the inductive form $r \diamondsuit\!\!\rightarrow \varphi$, where r is a regular expression and φ is an LTL_WR formula. Thus, the construction of Proposition 3.1 needs to be extended to handle each of these forms under both negation parities.

First we construct a non-deterministic finite automata on finite words (NFA) that recognizes tight satisfaction ($\models_{\text{PSL}}$) and a Büchi automata that recognizes weak RE satisfaction. Then we provide the overall construction for LTL_WR.

Proposition 3.4 *Let r be an* RE. *There exists an* NFA *N_r with $O(|r|)$ states such that*

$$w \models_{\text{PSL}} r \iff w \in \mathcal{L}(N_r)$$

Proof: We inductively define an NFA N_r for r as follows:

- Base: For a Boolean expression b we construct the automaton $N_b = (\Sigma, \{0,1\}, \{0\}, \rho_b, \{1\})$ such that for $l \in \Sigma$ we have $\rho(0, l) = \{1\}$ iff $l \models b$.
- Induction: assume that for the regular expressions r_1 and r_2 we constructed the NFAs $N_1 = (\Sigma, S_1, I_1, \rho_1, F_1)$ and $N_2 = (\Sigma, S_2, I_2, \rho_2, F_2)$ respectively.

– The NFA $N.$ for $r_1 \cdot r_2$ is $N. = (\Sigma, S_1 \cup S_2, I_1, \rho_1 \cup \rho_2 \cup \{(s_1, l, s_2) | s_1 \in F_1 \wedge s_2 \in \rho_2(q, l) \text{ for } q \in I_2\}, \hat{F})$. Where $\hat{F} = F_2$ if $I_2 \cap F_2 = \emptyset$ and $F_1 \cup F_2$ otherwise.

– The NFA $N_\cup$ for $r_1 \cup r_2$ is $N_\cup = (\Sigma, S_1 \cup S_2 \cup \{s_0\}, \{s_0\}, \rho_1 \cup \rho_2 \cup \{(s_0, l, s_1) | s_1 \in \rho_1(q, l) \text{ for } q \in I_1\} \cup \{(s_0, l, s_2) | s_2 \in \rho_2(q, l) \text{ for } q \in I_2\}, \hat{F})$. Where $\hat{F} = F_1 \cup F_2$ if $I_2 \cap F_2 = \emptyset$ and $I_1 \cap F_1 = \emptyset$ and $F_1 \cup F_2 \cup \{s_0\}$ otherwise.

– The NFA N_* for r_1^* is $N_* = (\Sigma, S_1 \cup \{s_0\}, \{s_0\}, \rho_1 \cup \{(s_0, l, s_1) | s_1 \in \rho_1(q, l) \text{ and } q \in I_1\} \cup \{(s_1, l, s_0) | \rho_1(s_1, l) \cap F_1 \neq \emptyset\}, F_1 \cup \{s_0\})$.

Let w be a finite word. We show that

$$w \in \mathcal{L}(N_r) \quad \text{iff} \quad w \models\!\!\!\models_{\text{PSL}} r$$

by induction on the structure of r.

- Base: Let $r = b$. Note that 0 is not accepting and that there are no outgoing transitions from 1, thus if N_b accepts v, then $|v| = 1$. Then, N_b accepts a word v iff $|v| = 1$ and $1 \in \rho(0, v^0)$ iff $|v| = 1$ and $v^0 \models b$ iff $v \models\!\!\!\models_{\text{PSL}} b$.

- Induction: Assume that for REs r_1 and r_2 we proved that $w \in \mathcal{L}(N_1) \Leftrightarrow w \models\!\!\!\models_{\text{PSL}} r_1$ and $w \in \mathcal{L}(N_2) \Leftrightarrow w \models\!\!\!\models_{\text{PSL}} r_2$.

  – For $r_1 \cdot r_2$ we distinguish between two cases:

  1. The case where $\epsilon \models\!\!\!\models_{\text{PSL}} r_2$. In this case, $N.$ for $r_1 \cdot r_2$ is $N. = (\Sigma, S_1 \cup S_2, I_1, \rho_1 \cup \rho_2 \cup \{(s_1, l, s_2) | s_1 \in F_1 \wedge s_2 \in \rho_2(q, l) \text{ for } q \in I_2\}, F_1 \cup F_2)$. Then, $N.$ accepts a word w iff
  it has a running trace $s_0, s_1, \ldots s_k$ $(k = |w|)$ over w such that $s_k \in F_1 \cup F_2$ iff either N_1 accepts w or there exists $j < k$ such that $s_j \in F_1$ and $s_{j+1} \in \rho_2(q, w^j)$ for $q \in I_2$ iff
  either N_1 accepts w or there exists $j < k$ such that N_1 accepts $w^{0..j}$ and N_2 accepts $w^{j+1..k}$ iff
  (induction) either $w \models\!\!\!\models_{\text{PSL}} r_1$ or there exists $j < k$ such that $w^{0..j} \models\!\!\!\models_{\text{PSL}} r_1$ and $w^{j+1..k} \models\!\!\!\models_{\text{PSL}} r_2$ iff
  $w \models\!\!\!\models_{\text{PSL}} r_1 \cdot r_2$.
  2. The case where $\epsilon \not\models\!\!\!\models_{\text{PSL}} r_2$. In this case, $N.$ for $r_1 \cdot r_2$ is $N. = (\Sigma, S_1 \cup S_2, I_1, \rho_1 \cup \rho_2 \cup \{(s_1, l, s_2) | s_1 \in F_1 \wedge s_2 \in \rho_2(q, l) \text{ for } q \in I_2\}, F_2)$. Then, $N.$ accepts a word w iff
  it has a running trace $s_0, s_1, \ldots s_k$ $(k = |w|)$ over w such that $s_k \in F_2$ iff
  there exists $j < k$ such that $s_j \in F_1$ and $s_{j+1} \in \rho_2(q, w^j)$ for $q \in I_2$ iff
  there exists $j < k$ such that N_1 accepts $w^{0..j}$ and N_2 accepts $w^{j+1..k}$ iff (induction)
  there exists $j < k$ such that $w^{0..j} \models\!\!\!\models_{\text{PSL}} r_1$ and $w^{j+1..k} \models\!\!\!\models_{\text{PSL}} r_2$ iff
  $w \models\!\!\!\models_{\text{PSL}} r_1 \cdot r_2$.

– For $r_1 \cup r_2$ we have that the NFA $N_\cup$ for $r_1 \cup r_2$ is $N_\cup = (\Sigma, S_1 \cup S_2 \cup \{s_0\}, \{s_0\}, \rho_1 \cup \rho_2 \cup \{(s_0, l, s_1) | s_1 \in \rho_1(q, l) \text{ for } q \in I_1\} \cup \{(s_0, l, s_2) | s_2 \in \rho_2(q, l) \text{ for } q \in I_2\}, \hat{F})$. Where $\hat{F} = F_1 \cup F_2 \cup \{s_0\}$ if $I_2 \cap F_2 \neq \emptyset$ or $I_1 \cap F_1 \neq \emptyset$ and $\hat{F} = F_1 \cup F_2$ otherwise.
$N_\cup$ accepts ϵ iff
$s_0 \in \hat{F}$ iff
$I_1 \cap F_1 \neq \emptyset$ or $I_2 \cap F_2 \neq \emptyset$ iff
N_1 accepts ϵ or N_2 accepts ϵ iff (induction) $\epsilon \models_{\text{PSL}} r_1$ or $\epsilon \models_{\text{PSL}} r_2$ iff $\epsilon \models_{\text{PSL}} r_1 \cup r2$.
Let w be a non empty word. $N_\cup$ accepts w iff
$N_\cup$ has a running trace that $s_0, s_1, \ldots, s_k$ (for $k = |w|$) such that s_0 is the initial state and $s_k \in F_1 \cup F_2$ iff
either $s_1 \in \rho(q, w^0)$ for $q \in I_1$ and $s_1, s_2, \ldots, s_k$ is a running trace of N_1 or $s_1 \in \rho(q, w^0)$ for $q \in I_2$ and $s_1, s_2, \ldots, s_k$ is a running trace of N_2 iff
N_1 accepts w or N_2 accepts w iff (induction)
$w \models_{\text{PSL}} r_1$ or $w \models_{\text{PSL}} r_2$ iff $w \models_{\text{PSL}} r_1 \cup r2$.

– For r_1^* we have that the NFA N_* for r_1^* is $N_* = (\Sigma, S_1 \cup \{s_0\}, \{s_0\}, \rho_1 \cup \{(s_0, l, s_1) | s_1 \in \rho_1(q, l) \text{ and } q \in I_1\} \cup \{(s_1, l, s_0) | \rho_1(s_1, l) \cap F_1 \neq \emptyset\}, F_1 \cup \{s_0\})$.
First note that $s_0 \in F_1$, thus N_* accepts ϵ. This comply with $\epsilon \models_{\text{PSL}} r_1^*$.
Let w be a non empty word. First direction, assume that N_* accepts w. Let $s_0, s_1, \ldots s_k$ be an accepting run trace of N_* on w. Let $j_0, j_1, \ldots$ be the sequence of all indices such that $s_{j_i} = s_0$. Then for every j_i in the set we have that $s_{j_i+1} \in \rho_1(q, w^{j_i})$ for $q \in I_1$ and that $p \in \rho_1(s_{j_i-1}, w^{j_i-1})$ for $p \in F_1$. Thus, for every j_i excepts for the last, we have that $q, s_{j_i+1}, \ldots s_{j_{i+1}-1}, p$ is accepting run of N_1 on $w^{j_i..j_{i+1}-1}$, and that for the last j_i, we have that $q, s_{j_i+1}, \ldots s_{k+1}$ is accepting run of N_1 on $w^{j_i..k}$. This implies that (induction) for every j_i excepts for the last, we have that $w^{j_i..j_{i+1}-1} \models_{\text{PSL}} r1$, and that for the last j_i, we have that $w^{j_i..k} \models_{\text{PSL}} r1$. Thus $w \models_{\text{PSL}} r_1^*$.
Second direction: Note that for every accepting run $q, s_1, \ldots, s_{k-1}, p$ of N_1 on a non empty word w, there exists an accepting run $s_0, s_1, \ldots s_{k-1}, s_0$ of N_* on w. Assume that $w \models_{\text{PSL}} r_1^*$. Then, there are $w_0, w_1, \ldots, w_k$ such that $w_0 \cdot w_1 \cdot \ldots \cdot w_k = w$ and for every $0 \leq j \leq k$ we have $w_j \models_{\text{PSL}} r_1$. Thus (induction), for every $0 \leq j \leq k$ we have that N_1 accepts w_j. Then, for every $0 \leq j \leq k$ we have N_* has a running trace on w_j that starts and ends at s_0. This implies that N_* accepts w.

□

Proposition 3.5 *Let r be an* RE. *There exists a Büchi automaton B_r with $O(|r|)$ states such that B_r has a trapping non-accepting state q_{bad} and for every word w over Σ,*

1. *there exists an accepting run of B_r on w iff $w \models_{\text{PSL}} r$*
2. *every run of B_r on w reaches q_{bad} iff $w \not\models_{\text{PSL}} r$*

Proof:

We build the Büchi automaton from the NFA for r constructed in Proposition 3.4. Before we provide the actual construction, we prove the following Lemma.

Lemma 3.6 *Let r be an RE and let N_r be the automaton constructed for r in Proposition 3.4. Then, for every $s \in S$ there exists $k \geq 0$ and a running trace $s, s_1, \ldots s_k$ of N_r on $\top^k$ such that $s_k \in F$.*

Proof: By induction.

- Base: $r = b$. The state 1 is in F, and $(0, \top, 1) \in \rho$.
- Induction: Assume that the NFAs N_1 and N_2 for r_1 and r_2, respectively, satisfy the lemma.
  - $r = r_1 \cdot r_2$. If $s \in S_2$, then by induction there exists $k \geq 0$ and a running trace s to F_2 on $\top^k$. Otherwise, $s \in S_1$. By induction, there is a running trace on $\top^j$ from s to F_1. From the transition relation, there exists $j \geq 0$ and a running trace on $\top^j$ from s to a state in S_2. By induction, there exists $k \geq 0$ and a running trace on $\top^k$ from that state of S_2 to F_2.
  - $r = r_1 \cup r_2$. If $s \in S_i$, then by induction there exists $k \geq 0$ and a running trace on $\top^k$ from s to F_i. Otherwise, $s = s_0$, and there is a transition on $\top$ from s_0 to each of S_1 and S_2.
  - $r = r_1^*$. If $s \in S_1$, then by induction there exists $k \geq 0$ and a running trace on $\top^k$ from s to F_1. Otherwise, $s = s_0$, which is itself accepting.

□

Let $N_r = (\Sigma, S, I, \rho, F)$ be the NFA for RE r, constructed as in Proposition 3.4. Let $B_r = (\Sigma, S \cup \{q_{bad}\}, I, \rho', S, S)$ where

$$\rho' = \begin{array}{l} \rho \cup \{(s, \ell, s) \mid \ell \in \Sigma,\ s \in F\} \cup \{(q_{bad}, \ell, q_{bad}) \mid \ell \in \Sigma\} \cup \\ \{(s, \ell, q_{bad}) \mid \ell \in \Sigma,\ s \notin F \text{ and } \forall s' \in S :\ (s, \ell, s') \notin \rho\} \end{array}$$

Observation 3.7 *Let w be a word over Σ. If there exists a running trace of B_r on w that contains a state in F, then B_r accepts w.*

Lemma 3.8 *A (finite or infinite) word w is accepted by B_r iff for every finite $v \preceq w$ there exists a prefix $u \preceq v \cdot \top^\omega$, such that $u \not\models_{\text{PSL}} r$.*

Proof: First direction: Let w be a word that is accepted by B_r. We prove that for every finite $v \preceq w$ there exists a prefix $u \preceq v \cdot \top^\omega$ such that $u \not\models_{\text{PSL}} r$. Let $s_0, s_1, \ldots$ be an accepting running trace of B_r on w. We distinguish between two cases:

1. The trace $s_0, s_1, \ldots$ contains a state in F (in N_r). In this case let k be the minimal number such that $s_k \in F$. The definition of ρ implies that $s_0, s_1, \ldots, s_k$ is a trace in N_r, thus $w^{0..k-1} \in \mathcal{L}(N_r)$. Proposition 3.4 implies that $w^{0..k-1} \models\!\!\!\models_{\text{PSL}} r$. Let $v \preceq w$ be a prefix of w. We distinguish between two cases:
   - If $w^{0..k-1} \preceq v$, then $w^{0..k-1} \preceq v \cdot \top^\omega$ and $w^{0..k-1} \models\!\!\!\models_{\text{PSL}} r$.
   - $v \prec w^{0..k-1}$. Let $j = |v|$, then $s_0, s_1, \ldots, s_j$ is a trace of N_r over v. Lemma 3.7 implies that there exists a trace $s_j, s_{j+1}, \ldots s_i$ of N_r on $\top^{i-j}$ such that $s_i \in F$, thus, $v \cdot \top^{i-j} \in \mathcal{L}(N_r)$. Then $v \cdot \top^{i-j} \models\!\!\!\models_{\text{PSL}} r$ and $v \cdot \top^{i-j} \preceq v \cdot \top^\omega$.
2. The trace $s_0, s_1, \ldots$ does not contain a state in F (in N_r). Then the definition of ρ implies that $s_0, s_1, \ldots$ is a trace in N_r. Let $v \preceq w$ be a finite prefix of w of length k. Then, $s_0, s_1, \ldots s_{k+1}$ is a run of N_r on v. Lemma 3.7 implies that there exists a trace $s_k, s_{k+1}, \ldots s_i$ of N_r on $\top^{i-k}$ such that $s_i \in F$, thus, $v \cdot \top^{i-k} \in \mathcal{L}(N_r)$. Then $v \cdot \top^{i-k} \models\!\!\!\models_{\text{PSL}} r$ and $v \cdot \top^{i-k} \preceq v \cdot \top^\omega$.

Second direction: Let w be a word that is not accepted by B_r. We prove that there exists a finite $v \preceq w$ such that for every prefix $u \preceq v \cdot \top^\omega$, we have that $u \not\models\!\!\!\models_{\text{PSL}} r$. Since w is not accepted by B_r, and since q_{bad} is the only non accepting and non final state, there exists a prefix $v \preceq w$ such that all running trace of B_r on v are trapped in q_{bad}. In addition, by Observation 3.7 above there is no running trace of B_r on $u \preceq v$ that contains a state in F. The definition of ρ implies that there is no running trace of N_r on v, and that there is no running trace of N_r on $u \preceq v$ that contains a state in F. Since there is no running trace of N_r on v, for every $v \preceq u \preceq v \cdot \top^\omega$ we have that u is not accepted by N_r and thus $u \not\models\!\!\!\models_{\text{PSL}} r$. Since there is no running trace of N_r on $u \preceq v$ that contains a state in F, we have that for every $u \preceq v$, u is not accepted by N_r, and thus $u \not\models\!\!\!\models_{\text{PSL}} r$. □

Lemma 3.8 shows that $w \models r$ iff there exists an accepting run of B_r on w. That is, $w \not\models\!\!\!\models_{\text{PSL}} r$ iff every run of B_r on w is not accepting.
Assume w is infinite. Then $w \not\models\!\!\!\models_{\text{PSL}} r$

iff every run of B_r on w does not visit S infinitely often

iff [since $S_r \setminus S = \{q_{bad}\}$] every run of B_r on w visits q_{bad} infinitely often

iff [since q_{bad} is trapping] every run of B_r on w visits q_{bad}.

Assume w is finite. Then $w \not\models\!\!\!\models_{\text{PSL}} r$

iff every run of B_r on w does not terminate in a state in S

iff [since $S_r \setminus S = \{q_{bad}\}$] every run of B_r on w terminates in S

Thus $w \not\models_{\text{PSL}} r$ iff every run of B_r on w reaches state q_{bad}. This completes the proof of Proposition 3.5. □

Below we extend the construction of the alternating automaton given in the previous section for LTL (over finite as well as infinite words) to LTL_WR. The idea is that for RE-based formulas $r \diamondsuit\!\!\rightarrow \psi$ and r the automaton mimics the automaton for r. Recall that $r \diamondsuit\!\!\rightarrow \psi$ states that *there exists* a prefix tightly satisfying r and the suffix beginning at the end of this prefix satisfies ψ, while $r \mapsto \varphi$ states that *for each* prefix tightly satisfying r, the suffix beginning at the end of the prefix satisfies φ. Thus, for "$r \diamondsuit\!\!\rightarrow \varphi$" states when the parity bit is 0 the automaton mimics a move to some next state in the NFA of r (so as to require one match for r) and when the parity bit is 1 the automaton mimics moves to all next states in the NFA of r (in order to traverse all matches for r). Then, when the automaton reaches a final state in N_r it continues to state "ψ" in order to require satisfaction of ψ. For a weak RE r the automaton mimics the Büchi automaton B_r of Proposition 3.5. When the parity bit is 0, the automaton traverses all states, to check that all prefixes are "potentially accepting" (i.e., have an extension with $\top$s that is accepting). When the parity bit is 1, the automaton traverses some state to check for some bad prefix.

To make this idea work we need to consider a new type of formula involving automata. For these we introduce the following logic.

Definition 8 (LTL_WRA formulas)

- *If b is a boolean expression then $b!$ is an* LTL_WRA *formula.*
- *If φ and ψ are* LTL_WRA *formulas, then the following are* LTL_WRA *formulas:*

$$\bullet\ \neg\varphi \qquad \bullet\ \varphi \wedge \psi \qquad \bullet\ X!\,\varphi \qquad \bullet\ [\varphi\ U\ \psi]$$

- *If φ is an* LTL_WRA *formula $N = (\Sigma, S, I, \rho, F)$ is an* NFA *such that I is a singleton, then $\langle N, \varphi\rangle$ is an* LTL_WRA *formula.*
- *If $B = (\Sigma, S, I, \rho, F, A)$ is a Büchi automaton such that I is a singleton then $\langle B\rangle$ is an* LTL_WRA *formula.*

The semantics of LTL_WRA formulas is defined as follows:

Definition 9 *Let v denote a word over Σ; b a boolean expression; r an* RE*; and φ and ψ* LTL_WRA *formulas. The notation $v \models_{\text{AUT}} \varphi$ means that v satisfies φ. The relation $\models_{\text{AUT}}$ is defined as follows:*[4]

1. $v \models_{\text{AUT}} b! \iff |v| > 0$ *and* $v^0 \models_{\text{PSL}} b$
2. $v \models_{\text{AUT}} \neg\varphi \iff \overline{v} \not\models_{\text{AUT}} \varphi$

[4]The semantics of the LTL operators are the standard ones.

3. $v \models_{\text{AUT}} \varphi \wedge \psi \iff v \models_{\text{AUT}} \varphi$ *and* $v \models_{\text{AUT}} \psi$

4. $v \models_{\text{AUT}} \mathsf{X!}\,\varphi \iff |v| > 1$ *and* $v^{1..} \models_{\text{AUT}} \varphi$

5. $v \models_{\text{AUT}} [\varphi \mathsf{U} \psi] \iff \exists 0 \leq k < |v|$ *s.t.* $v^{k..} \models_{\text{AUT}} \psi$ *and* $\forall 0 \leq j < k,\ \ v^{j..} \models_{\text{AUT}} \varphi$

6. $v \models_{\text{AUT}} \langle N, \psi \rangle \iff \exists 0 \leq j < |v|$ *s.t.* $v^{0..j} \in \mathcal{L}(N)$ *and* $v^{j..} \models_{\text{AUT}} \psi$

7. $v \models_{\text{AUT}} \langle B \rangle \iff v \in \mathcal{L}(B)$

Proposition 3.9 *Let φ be a formula of* LTL_WR. *Let Φ be the formula of* LTL_WRA *obtained from φ by replacing sub-formulas of the form $r \diamondsuit\!\!\rightarrow \psi$ with $\langle N_r, \psi \rangle$ where $N_r = (\Sigma, S_r, I_r, \rho_r, F_r)$ is the* NFA *from Proposition 3.4; and formulas of the form r with $\langle B_r \rangle$ where $B_r = (\Sigma, S_r \cup \{q_{bad}\}, I_r, \rho'_r, S_r, S_r)$ is the Büchi automaton defined in Proposition 3.5 and q_{bad} is the trapping non-accepting state. Then*

$$[\![\varphi]\!] = [\![\Phi]\!]$$

Proof: The proof is by induction on the structure of the formula. The cases for booleans and the LTL operators are immediate since the semantics for those are the same.

- $v \models_{\text{PSL}} r \diamondsuit\!\!\rightarrow \psi$

  $\iff \exists\ 0 \leq j < |v|$ s.t. $v^{0..j} \not\models_{\text{PSL}} r$ and $v^{j..} \models_{\text{PSL}} \psi$

  $\iff$ [by Proposition 3.4]
  $\exists\ 0 \leq j < |v|$ s.t. $v^{0..j} \in \mathcal{L}(N_r)$ and $v^{j..} \models_{\text{PSL}} \psi$

  $\iff v \models_{\text{AUT}} \langle N_r, \psi \rangle$

- $v \models_{\text{PSL}} r$

  $\iff$ [by Proposition 3.5]
  $v \in \mathcal{L}(B_r)$

  $\iff v \models_{\text{AUT}} \langle B_r \rangle$

□

Proposition 3.9 allows to use the construction of a Büchi automaton for an LTL_WRA formula Φ to obtain a construction for an LTL_WR formula φ for which $[\![\varphi]\!] = [\![\Phi]\!]$.

Proposition 3.10 *Given an* LTL_WRA *formula φ, there exists an alternating Büchi automaton $B_\varphi = (\Sigma, S, I, \rho, F, A)$ where $|S|$ is in $O(|\varphi|)$ and $\mathcal{L}(B_\varphi)$ is exactly the set of (finite and infinite) words satisfying the formula φ.*

Proof: Let φ be a formula of LTL_WRA. Let S be the set of pairs (p, ψ) where $p \in \{0, 1\}$ and ψ satisfies at least one of the following:

- ψ is a subformula of φ.
- $\psi \in \{\text{TRUE}, \text{FALSE}, \text{NONEMPTY}\}$.
- ψ is of the form $\langle N^q, \vartheta\rangle$, where $\langle N, \vartheta\rangle$ is a subformula of φ and q is a state of the NFA N.
- ψ is of the form $\langle B^q\rangle$, where $\langle B\rangle$ is a subformula of φ and q is a state of the Büchi automaton B.

Note that if q is the initial state of N, then $N^q = N$, and similarly with B.

The initial state is $(0, \varphi)$.

The set A of accepting states is produced by the following rules:

1. A contains all states of the form: $(1, \text{NONEMPTY})$, $(0, \text{TRUE})$, $(1, \text{FALSE})$ $(1, b!)$, $(1, \mathsf{X}!\psi)$, $(1, [\psi \mathsf{U} \vartheta])$, $(1, \langle N^q, \psi\rangle)$, $(0, \langle B^q\rangle)$, $(1, \langle B^{q_{bad}}\rangle)$.
   In $(0, \langle B^q\rangle)$ it is understood that q is not the trapping state of B, while in $(1, \langle B^{q_{bad}}\rangle)$ it is understood that q_{bad} is the trapping state of B.
2. $(p, \neg\psi) \in A$ iff $(1 - p, \psi) \in A$.
3. $(0, \psi \wedge \vartheta) \in A$ iff both $(0, \psi) \in A$ and $(0, \vartheta) \in A$.
   $(1, \psi \wedge \vartheta) \in A$ iff either $(1, \psi) \in A$ or $(1, \vartheta) \in A$.

The set F of final states is equal to A.

The transition relation ρ is defined as follows, where $N^q = (\Sigma, S_N, \{q\}, \rho_N, F_N)$ and $B^q = (\Sigma, S_B \cup q_{bad}, \{q\}, \rho_B, S_B)$:

- $\rho((p, \text{NONEMPTY}), \ell) = (p, \text{TRUE})$
- $\rho((p, \text{TRUE}), \ell) = (p, \text{TRUE})$
  $\rho((p, \text{FALSE}), \ell) = (p, \text{FALSE})$
- $\rho((0, b!), \ell) = (0, \ell \Vdash_{\text{PSL}} b)$
  $\rho((1, b!), \ell) = (1, \overline{\ell} \Vdash_{\text{PSL}} b)$
- $\rho((0, \psi \wedge \vartheta), \ell) = \rho((0, \psi), \ell) \wedge \rho((0, \vartheta), \ell)$
  $\rho((1, \psi \wedge \vartheta), \ell) = \rho((1, \psi), \ell) \vee \rho((1, \vartheta), \ell)$
- $\rho((0, \neg\psi), \ell) = \rho((1, \psi), \ell)$
  $\rho((1, \neg\psi), \ell) = \rho((0, \psi), \ell)$
- $\rho((0, \mathsf{X}!\psi), \ell) = (0, \text{NONEMPTY}) \wedge (0, \psi)$
  $\rho((1, \mathsf{X}!\psi), \ell) = (1, \text{NONEMPTY}) \vee (1, \psi)$
- $\rho((0, [\psi \mathsf{U} \vartheta]), \ell) = \rho((0, \vartheta), \ell) \vee (\rho((0, \psi), \ell) \wedge (0, [\psi \mathsf{U} \vartheta]))$
  $\rho((1, [\psi \mathsf{U} \vartheta]), \ell) = \rho((1, \vartheta), \ell) \wedge (\rho((1, \psi), \ell) \vee (1, [\psi \mathsf{U} \vartheta]))$

- If $\rho_N(q,\ell) \cap F_N$ is non-empty, then

$$\rho((0,\langle N^q,\psi\rangle),\ell) = \bigvee_{q' \in \rho_N(q,\ell)} (0,\langle N^{q'},\psi\rangle) \vee \rho((0,\psi),\ell)$$

Otherwise

$$\rho((0,\langle N^q,\psi\rangle),\ell) = \bigvee_{q' \in \rho_N(q,\ell)} (0,\langle N^{q'},\psi\rangle)$$

If $\rho_N(q,\bar{\ell}) \cap F_N$ is non-empty, then

$$\rho((1,\langle N^q,\psi\rangle),\ell) = \bigwedge_{q' \in \rho_N(q,\bar{\ell})} (1,\langle N^{q'},\psi\rangle) \wedge \rho((1,\psi),\ell)$$

Otherwise

$$\rho((1,\langle N^q,\psi\rangle),\ell) = \bigwedge_{q' \in \rho_N(q,\bar{\ell})} (1,\langle N^{q'},\psi\rangle)$$

- $\rho((0,\langle B^q\rangle),\ell) = \bigvee_{q' \in \rho_B(q,\ell)} (0,\langle B^{q'}\rangle)$
  $\rho((1,\langle B^q\rangle),\ell) = \bigwedge_{q' \in \rho_B(q,\bar{\ell})} (1,\langle B^{q'}\rangle)$

Lemma 3.11 *Let φ be an* LTL_WRA *formula. If ψ is a subformula of φ, let $A^{(p,\psi)}$ denote the subautomaton of this construction obtained by taking (p,ψ) as initial state. Then*

1. $\mathcal{L}(A^{(0,\varphi)}) = [\![\varphi]\!]$.
2. $\mathcal{L}(A^{(1,\varphi)}) = \Sigma^\infty - \overline{[\![\varphi]\!]} = [\![\neg\varphi]\!]$.

Proof: The proof is by induction on the structure of the formula. The cases for booleans and the LTL operators are the same as in the proof of Lemma 3.2.

– We show that $\mathcal{L}(A^{(0,\langle N^q,\psi\rangle)}) = [\![\langle N^q,\psi\rangle]\!]$ and $\mathcal{L}(A^{(1,\langle N^q,\psi\rangle)}) = [\![\neg\langle N^q,\psi\rangle]\!]$

- We show that $v \in [\![\langle N^q,\psi\rangle]\!]$ iff $v \in \mathcal{L}(A^{(0,\langle N^q,\psi\rangle)})$.

  Suppose that $v \in [\![\langle N^q,\psi\rangle]\!]$. Then $\exists 0 \leq j < |v|$ s.t. $v^{0..j} \in \mathcal{L}(N^q)$ and $v^{j..} \in [\![\psi]\!]$. Since $v^{0..j} \in \mathcal{L}(N^q)$, there exists an accepting run

  $$q, q_0, ..., q_j \in F_N$$

  of N^q on $v^{0..j}$. By induction, $\mathcal{L}(A^{(0,\psi)}) = [\![\psi]\!]$, so there exists an accepting run tree t of $A^{(0,\psi)}$ on $v^{j..}$

From the transition relation,

$$\langle N^q, \psi\rangle, \langle N^{q_0}, \psi\rangle, ..., \langle N^{q_{j-1}}, \psi\rangle$$

is a run of $A^{(0,\langle N^q,\psi\rangle)}$ on $v^{0..j-1}$. Also, since $q_j \in F_N$, it follows from the transition relation that each of the successors of the root of the run tree t on letter v^j is also a successor of $\langle N^{q_{j-1}}, \psi\rangle$ on the letter v^j. Let t' be obtained from t by replacing the root with $\langle N^{q_{j-1}}, \psi\rangle$. Since t,t' differ only on the root node and since $v^{j\cdot\cdot}$ is non-empty, it follows that t' is an accepting run tree of $A^{(0,\langle N^{q_{j-1}},\psi\rangle)}$ on $v^{j\cdot\cdot}$. This shows that $v \in \mathcal{L}(A^{(0,\langle N^q,\psi\rangle)})$.

Suppose now that $v \in \mathcal{L}(A^{(0,\langle N^q,\psi\rangle)})$. Then there exists an accepting run tree t of $A^{(0,\langle N^q,\psi\rangle)}$ on v. Since $(0, \langle N^q, \psi\rangle)$ is not accepting, v is non-empty. From the transition relation, t has a branch labeled

$$(0, \langle N^q, \psi\rangle), (0, \langle N^{q_0}, \psi\rangle), \ldots$$

Since none of the states $(0, \langle N^{q0}, \psi\rangle)$ is accepting, there must exist $0 \leq j < |v|$ such that

$$(0, \langle N^q, \psi\rangle), (0, \langle N^{q_0}, \psi\rangle), \ldots (0, \langle N^{q_{j-1}}, \psi\rangle)$$

labels a branch $t, t_0, \ldots t_{j-1}$ of t, and $q, q_0, \ldots, q_{j-1}$ is a run of N^q on $v^{0..j-1}$, and $\rho_N(q_{j-1}, v^j) \cap F_N \neq \emptyset$, and the set of children of t_{j-1} is labeled by the set $\rho((0, \psi), v^j)$. Then $q, q_0, \ldots, q_j$ is a run of N^q on $v^{0..j}$, and so $v^{0..j} \in \mathcal{L}(N^q)$. Let $t' = t_{j,1}$.
Since t is accepting, t' must be an accepting run of $A^{(0,\langle N^{q_{j-1}},\psi\rangle)}$ on $v^{j\cdot\cdot}$. Let t'' be the tree obtained from t' by switching the root to $(0, \psi)$. From the transition relation, t'' is a run tree of $A^{(0,\psi)}$ on $v^{j\cdot\cdot}$. Since $v^{j\cdot\cdot}$ is non-empty, it follows that t'' is accepting. Therefore, $v^{j\cdot\cdot} \in \mathcal{L}(A^{(0,\psi)})$. By induction, $v^{j\cdot\cdot} \in [\![\psi]\!]$. Thus, $v \in [\![\langle N^q, \psi\rangle]\!]$.

- We show that $v \in [\![\neg\langle N^q, \psi\rangle]\!]$ iff $v \in \mathcal{L}(A^{(1,\langle N^q,\psi\rangle)})$.

  $v \in [\![\neg\langle N^q, \psi\rangle]\!]$ iff $\bar{v} \notin [\![\langle N^q, \psi\rangle]\!]$ iff $\forall 0 \leq j < |v|$ s.t. $\bar{v}^{0..j} \in \mathcal{L}(N^q)$, $\bar{v}^{j\cdot\cdot} \notin [\![\psi]\!]$ iff [induction]

  $$(\star): \quad \forall 0 \leq j < |v| \text{ s.t. } \bar{v}^{0..j} \in \mathcal{L}(N^q),\ v^{j\cdot\cdot} \in \mathcal{L}(A^{(1,\psi)})$$

  Suppose that $(\star)$ holds. We construct an accepting run tree t of $A^{(1,\langle N^q,\psi\rangle)}$ on v. Let t_0 be a run tree of $A^{(1,\langle N^q,\psi\rangle)}$ on v. Let t_N be the rooted subtree of t_0 consisting of all states of the form $(1, \langle N^{q'}, \psi\rangle)$ (that is, trim every branch of t_0 at the first node which is not of the form $(1, \langle N^{q'}, \psi\rangle)$). Since the states of this form are all accepting, every branch of this subtree is accepting. Consider a node $n = (1, \langle N^{q'}, \psi\rangle)$ of t_N such that n has a nonempty set M_n of successors in $t_0 - t_N$. From the transition relation, it follows that there exists $0 \leq j < |v|$ and a run $q, q_0, ..., q_{j-1} = q'$ of N^q on $\bar{v}^{0..j-1}$ and there

exists $q_j \in \rho_N(q', \bar{v}^j) \cap F_N$. Then $q, q_0, \ldots, q_j$ is an accepting run of N^q on $\bar{v}^{0..j}$, so by $(\star)$, $v^{j..} \in \mathcal{L}(A^{(1,\psi)})$. Let t_n be an accepting run tree of $A^{(1,\psi)}$ on $v^{j..}$. Replace M_n in t_0 by the set of successors of the root of t_n. As a result, any branch of t_0 passing through a successor of n in $t_0 - t_N$ is now converted to an accepting branch. The procedure is repeated for all such nodes n to yield the accepting run tree t.

Suppose that $v \in \mathcal{L}(A^{(1,\langle N^q,\psi\rangle)})$. Let t be an accepting run tree of $A^{(1,\langle N^q,\psi\rangle)}$ on v. Let t_N be the rooted subtree of t consisting of all states of the form $(1, \langle N^{q'}, \psi\rangle)$ (that is, trim every branch of t at the first node which is not of the form $(1, \langle N^{q'}, \psi\rangle)$). Suppose that there exists $0 \leq j < |v|$ s.t. $\bar{v}^{0..j} \in \mathcal{L}(N^q)$. Let $q, q_0, \ldots, q_j$ be an accepting run of N^q on $\bar{v}^{0..j}$. From the transition relation,

$$(1, \langle N^q, \psi\rangle), (1, \langle N^{q_0}, \psi\rangle), \ldots, (1, \langle N^{q_{j-1}}, \psi\rangle)$$

is a prefix of a branch of t_N, and the successors of $(1, \langle N^{q_{j-1}}, \psi\rangle)$ in $t - t_N$ are successors of $(1, \psi)$ on v^j.
Let t' be the tree obtained from the subtree of t rooted at $(1, \langle N^{q_{j-1}}, \psi\rangle)$ by eliminating the successors of $(1, \langle N^{q_{j-1}}, \psi\rangle)$ that are in t_N and replacing the root by $(1, \psi)$. It follows that t' is an accepting run tree of $A^{(1,\psi)}$ on $v^{j..}$, and so $v^{j..} \in \mathcal{L}(A^{(1,\psi)})$. This proves that $(\star)$ holds.

– We show that $\mathcal{L}(A^{(0,\langle B^q\rangle)}) = [\![\langle B^q\rangle]\!]$ and $\mathcal{L}(A^{(1,\langle B^q\rangle)}) = [\![\neg\langle B^q\rangle]\!]$.
Let w be a finite/infinite word.
Recall that q' is an accepting state of $\langle B^q\rangle$
iff $q' \neq q_{bad}$. Also, $(0, \langle B^{q'}\rangle)$ is an accepting state of $A^{(0,\langle B^q\rangle)}$
iff $q' \neq q_{bad}$.
Thus, by the construction,
$w \in \mathcal{L}(A^{(0,\langle B^q\rangle)})$
iff there exists a run of B^q on w that does not visit q_{bad}
iff $w \in \mathcal{L}(B^q)$
iff $w \in [\![\langle B^q\rangle]\!]$
Similarly, $(1, \langle B^{q'}\rangle)$ is an accepting state of $A^{(1,\langle B^q\rangle)}$ iff $q' = q_{bad}$. Thus,
$w \in \mathcal{L}(A^{(1,\langle B^q\rangle)})$
iff every run of B^q on $\bar{w}$ reaches q_{bad}
iff $\bar{w} \notin \mathcal{L}(B^q)$
iff $\bar{w} \notin [\![\langle B^q\rangle]\!]$
iff $w \in [\![\neg\langle B^q\rangle]\!]$

□

This completes the proof of the Proposition. □

Corollary 3.12 *Given an* LTL_WRA *formula φ, there exists a Büchi automaton $B_\varphi = (\Sigma, S, I, \rho, F, A)$ where $|S|$ is in $2^{O(|\varphi|)}$ and $\mathcal{L}(B_\varphi)$ is exactly the set of (finite and infinite) words satisfying the formula φ.*

Proof: Follows directly from Propositions 3.10 and 2.1. □

Corollary 3.13 *Given an* LTL_WR *formula φ, there exists a Büchi automaton $B_\varphi = (\Sigma, S, I, \rho, F, A)$ where $|S|$ is in $2^{O(|\varphi|)}$ and $\mathcal{L}(B_\varphi)$ is exactly the set of (finite and infinite) words satisfying the formula φ.*

Proof: Follows directly from Corollary 3.12 and Proposition 3.9. □

Definition 10 *We say that a language is ∞-regular iff it is a union of a regular language and an ω-regular language.*

Proposition 3.14 LTL_WR *is as expressive as ∞-regular languages.*

Proof: Clearly a langauge accepted by a Büchi automaton on finite/infinite words is an ∞-regular language. Thus, Corollary 3.13 implies that LTL_WR formulas are not more expressive than ∞-regular expressions (since for every formula φ in LTL_WR there exists a Büchi automaton B_φ on finite/infinite words such that $[\![\varphi]\!] = \mathcal{L}(B_\varphi)$).

A proof that any ω-regular language, can be expressed as an LTL_WR formula appears in [AFF+02], where the suffix implication operator ($\mapsto$) is denoted triggers, and the suffix conjunction operator ($\diamondsuit\!\!\rightarrow$) is denoted follows_by. Note, that the language of an LTL_WR formula interpreted over finite and infinite words may not be the same as its language when interpreted over infinite words alone. For example, when the semantics is interpreted over infinite words alone, we have that $[\![b \mathsf{W} \mathit{false}]\!] = \{\ell \mid \ell \models_{\text{PSL}} b\}^\omega$, but when the semantics is interpreted over finite words as well, we have $[\![b \mathsf{W} \mathit{false}]\!] = \{\ell \mid \ell \models_{\text{PSL}} b\}^\omega \cup \{\ell \mid \ell \models_{\text{PSL}} b\}^*$.

A regular language (over finite words) L can be expressed in LTL_WR by means of the formula $(r_L \diamondsuit\!\!\rightarrow \mathsf{X}\, \mathit{false})$ where r_L is a regular expression accepting the same language as L. An omega regular language (over infinite words) L can expressed in LTL_WR by means of the formula $\psi_L \wedge ((\mathsf{X}!\mathit{true})\ \mathsf{W}\ \mathit{false})$ where ψ_L is an LTL_WR formula accepting the language L when interpreted over infinite words only.

Thus, for every ∞-regular language L we can construct an LTL_WR formula ψ_{fin} such that $[\![\psi_{fin}]\!] = \{w \mid w \in L \text{ and } w \text{ is finite}\}$ and a formula ψ_{inf} such that $[\![\psi_{inf}]\!] = \{w \mid w \in L \text{ and } w \text{ is infinite}\}$. Thus, the formula $\psi_{fin} \vee \psi_{inf}$ accepts exactly the set of (finite and infinite) words in L.

□

3.3 Classification of Automata for RE-based LTL_WR-formulas

Proposition 3.15 *Let r be an* RE*, and φ an* LTL_WR *formula. If there exists a weak (terminal) Büchi automaton for $\neg\varphi$ with state set S, then there exists a weak (terminal) Büchi automaton for $\neg(r \mapsto \varphi)$ with at most $O(|r| + |S|)$ states.*

Proof: Let r be an RE and φ be an LTL_WR formula. We want to build a Büchi automaton for the negation of $r \mapsto \varphi$. Note that by the semantics of suffix implication $[\![r \mapsto \varphi]\!] = [\![r' \mapsto \varphi']\!]$ where $\mathbb{S}(r') = \mathbb{S}(r) \setminus \{\epsilon\}$ and $[\![\varphi']\!] = [\![\varphi]\!] \setminus \{\epsilon\}$. We therefore assume without loss of generality that $\epsilon \notin \mathbb{S}(r)$ and that $\epsilon \notin [\![\varphi]\!]$

Let $N_r = (\Sigma, S_r, I_r, \rho_r, F_r)$ be the NFA constructed for r as in Proposition 3.4. We build from it the NFA $N'_r = (\Sigma, S'_r, I_r, \rho'_r, F'_r)$ where $S'_r = S_r \cup \{q_{fin}\}$, $F'_r = \{q_{fin}\}$ and $\rho'_r = \rho_r \cup \{(q, \ell, q_{fin}) \mid \exists q_f \in F_r \text{ s.t. } (q, \ell, q_f) \in \rho_r\}$. Clearly, N'_r is a linear NFA in $|r|$ accepting the words tightly satisfying r such that there are no states in $I_r \cap F'_r$.

Let $B_\varphi = (\Sigma, S_\varphi, I_\varphi, \rho_\varphi, F_\varphi, A_\varphi)$ be the Büchi automaton accepting language $[\![\neg\varphi]\!]$. We can assume the construction of B_φ is such that I_φ is disjoint from F_φ and that $(s, \ell, s') \in \rho_\varphi$ implies $s' \notin I_\varphi$.

We build the Büchi automaton $B = (\Sigma, S, I, \rho, F, A)$ for $\neg(r \mapsto \varphi)$ as follows: $S = (S'_r \setminus F'_r) \cup (S_\varphi \setminus I_\varphi)$; $I = I_r$; $F = F_\varphi$; $A = A_\varphi$; and

$$\rho = \begin{array}{l} (\rho_r \setminus \{(s, \ell, t) \mid t \in F'_r\}) \ \cup \ (\rho_\varphi \setminus \{(s^0, \ell, s') \mid s^0 \in I_\varphi\}) \cup \\ \{(s, \ell, s') \mid \exists t \in F'_r \text{ s.t. } (s, \ell, t) \in \rho_r \text{ and } \exists s^0 \in I_\varphi \text{ s.t. } (s^0, \ell, s') \in \rho_\varphi)\} \end{array}$$

First we show that $\mathcal{L}(B) = [\![\neg(r \mapsto \varphi)]\!]$. Then we argue that if B_φ is weak (terminal) then so is B.

1. $v \in [\![\neg(r \mapsto \varphi)]\!]$

   $\iff v \models_{\text{PSL}} \neg(r \mapsto \varphi)$

   $\iff v \models_{\text{PSL}} r \diamondsuit\!\!\rightarrow \neg\varphi$

   $\iff \exists j < |v| \text{ s.t. } v^{0..j} \models_{\text{PSL}} r \text{ and } v^{j..} \models_{\text{PSL}} \neg\varphi$

   $\iff$ there exists a run $s_0, s_1, \ldots, s_{j+1}$ of N'_r on $v^{0..j}$ such that $s_{j+1} \in F'_r$ and there exists an accepting (finite or infinite) run $s'_0, s'_1, s'_2 \ldots$ of B_φ on $v^{j..}$ such that $s'_0 \in I_\varphi$

   $\iff$ [$s_j \xrightarrow{v^j} s'_1 \in \rho$ iff $\exists s_{j+1} \in F'_r$ s.t. $s_j \xrightarrow{v^j} s_{j+1} \in \rho_r$ and $\exists s'_0 \in I_\varphi$ s.t. $s'_0 \xrightarrow{v^j} s'_1 \in \rho_\varphi$] there exists an accepting (finite or infinite) run $s_0, s_1, \ldots, s_j, s'_1, s'_2 \ldots$ of B on v

   $\iff v \in \mathcal{L}(B)$

2. Let $S_1, \ldots, S_k$ be the partition of states and $\leq$ the partial order that prove B_φ is weak(terminal). Let S'_i be $S_i \setminus I_\varphi$ for $1 \leq i \leq k$. Let $S'_r = S_r \setminus F'_r$. Then B is proved weak (terminal) by the partition $S'_r, S'_1, \ldots, S'_k$ together with the partial order $S'_r \leq S'_i$, and $S'_i \leq S'_j$ iff $S_i \leq S_j$.

□

Proposition 3.16 *Let r be an* RE. *Then, there exists a universal alternating Büchi automaton for $\neg r$ with state complexity in $O(|r|)$, and there exists a terminal existential Büchi automaton for $\neg r$ with state complexity in $2^{O(|r|)}$.*

Proof: Let $B_r = (\Sigma, S \cup \{q_{bad}\}, I, \rho, S, S)$ be the existential Büchi automaton from Proposition 3.5. Recall that B_r has $O(|r|)$ states and for every word w over Σ,

1. there exists an accepting run of B_r on w iff $w \models_{\text{PSL}} r$
2. every run of B_r on w reaches q_{bad} iff $w \not\models_{\text{PSL}} r$

Let B be the universal Büchi automaton $(\Sigma, S \cup \{q_{bad}\}, I, \rho, \{q_{bad}\}, \{q_{bad}\})$. Clearly B accepts a word w iff $w \not\models_{\text{PSL}} r$. Let $\overline{B}$ be the universal Büchi automaton $(\Sigma, S \cup \{q_{bad}\}, I, \overline{\rho}, \{q_{bad}\}, \{q_{bad}\})$ where

$$\overline{\rho} = \{(s_1, \ell, s_2) \mid \ell \in \Sigma,\ (s_1, \overline{\ell}, s_2) \in \rho\}.$$

Clearly, $w \in \mathcal{L}(B)$ iff $\overline{w} \in \mathcal{L}(\overline{B})$. Therefore $w \in \mathcal{L}(B)$ iff $\overline{w} \not\models_{\text{PSL}} r$ iff $w \models_{\text{PSL}} \neg r$.

The universal Büchi automaton $\overline{B}$ can be translated into an ordinary existential Büchi automaton B' with state set S' such that $|S'|$ is in $2^{O(|r|)}$ [MH84]. This translation can be done in a way that preserves q_{bad} as the unique trapping accepting state of B'. Then it is clear that B' is terminal by partitioning according to $S \setminus \{q_{bad}\} < \{q_{bad}\}$

□

Proposition 3.17 *Let r be an* RE. *Then, there exists a weak Büchi automaton for $\neg r!$ whose state complexity is in $O(|r|)$.*

Proof: Let r be an RE. Let $N = (\Sigma, S, I, \rho, F)$ be an NFA accepting the non-empty words tightly satisfying r, with state complexity linear in $|r|$ (for example, N is the NFA of Proposition 3.4). We build from it the NFA $N' = (\Sigma, S', I, \rho', F')$ where $S' = S \cup \{q_{fin}\}$, $F' = \{q_{fin}\}$ and $\rho' = \rho \cup \{(q, \ell, q_{fin}) \mid \exists q_f \in F \text{ s.t. } (q, \ell, q_f) \in \rho\}$. Clearly, N' is a linear NFA in $|r|$ accepting the non-empty words tightly satisfying r such that there are no transitions out of F'. It follows that in any accepting run, F' cannot be reached prior to the end of the run (i.e., if state $s_f \in F'$ is reached during an accepting run, then s_f is the last state of the run).

We build a Büchi automaton $B'' = (\Sigma, S', I, \rho'', F'', A'')$ as follows: $A'' = F'' = S' \setminus F'$; and $\rho'' = \rho' \cup \{(s, \ell, s) \mid s \in F',\ \ell \in \Sigma\}$. Note that the states of F' in B are non-accepting and trapping. Finally, we build a Büchi automaton $B = (\Sigma, S', I, \overline{\rho}, F'', A'')$ for $\neg r!$ as follows:

$$\overline{\rho} = \{(s_1, \ell, s_2) \mid \ell \in \Sigma,\ (s_1, \overline{\ell}, s_2) \in \rho''\}.$$

Clearly $v \in \mathcal{L}(B)$ if and only if $\overline{v} \in \mathcal{L}(B'')$.

First we show that $\mathcal{L}(B) = [\![\neg(r!)]\!]$. Then we argue that B is weak.

1. $v \in [\![\neg(r!)]\!]$

$\iff v \models_{\text{PSL}} \neg r!$

$\iff \overline{v} \not\models_{\text{PSL}} r!$

$\iff \nexists k < |v|$ s.t. $\overline{v}^{0..k} \models_{\text{PSL}} r$

$\iff \forall k < |v|, \overline{v}^{0..k} \not\models_{\text{PSL}} r$

$\iff$ [since N' recognizes non-empty matches of r]
$\forall k < |v|$ there exists no run of N' on $\overline{v}^{0..k}$ terminating in a state in F'

$\iff$ [by the properties of N']
$\forall k < |v|$ there exists no run of N' on $\overline{v}^{0..k}$ visiting a state in F'

$\iff$ [by the definition of B'']
$\forall k < |v|$ there exists no run of B'' on $\overline{v}^{0..k}$ visiting a state in F'

$\iff$ [there are two options]
Either v is finite and any run of B'' on $\overline{v}$ does not terminate in a state in F'
Or v is infinite and any run of B'' on $\overline{v}$ never visits a state in F'

$\iff \overline{v} \in \mathcal{L}(B'')$

$\iff v \in \mathcal{L}(B)$

2. Let the partition and partial order of the states of B be $S' \setminus F' < F'$. Since $F'' = A'' = S' \setminus F'$ and F' is trapping, it follows that B is weak.

□

Acknowledgment

We would like to thank Cindy Eisner for many stimulating discussions. The second author would also like to thank Nir Piterman and Sitvanit Ruah for interesting discussions on some of the underlying issues.

References

[ABKV03] Roy Armoni, Doron Bustan, Orna Kupferman, and Moshe Y. Vardi. Aborts vs resets in linear temporal logic. In *Proc. 9th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, 2003.

[Acc04] Accellera Organization, Inc. Formal semantics of Accellera property specification language. In *Appendix B of http://www.eda.org/vfv/docs/PSL-v1.1.pdf*, pages 109–119, January 2004.

[AFF+02] R. Armoni, L. Fix, A. Flaisher, R. Gerth, B. Ginsburg, T. Kanza, A. Landver, S. Mador-Haim, E. Singerman, A. Tiemeyer, M. Y. Vardi, and Y. Zbar. The ForSpec temporal logic: A new temporal property-specification language. In *TACAS*, pages 296–211, 2002.

[BBL98] I. Beer, S. Ben-David, and A. Landver. On-the-fly model checking of RCTL formulas. In *Proc.* 10^{th} *International Conference on Computer Aided Verification (CAV'98)*, LNCS 1427, pages 184–194. Springer-Verlag, 1998.

[BRS99] R. Bloem, K. Ravi, and F. Somenzi. Efficient decision procedures for model checking of linear time logic properties. In *Proc.* 11^{th} *International Conference on Computer Aided Verification (CAV)*, LNCS 1633, pages 222–235. Springer-Verlag, 1999.

[EFH+03] C. Eisner, D. Fisman, J. Havlicek, Y. Lustig, A. McIsaac, and D. Van Campenhout. Reasoning with temporal logic on truncated paths. In *The 15th international conference on computer aided verification (CAV)*, LNCS 2725, pages 27–40. Springer-Verlag, July 2003.

[EFH05] Cindy Eisner, Dana Fisman, and John Havlicek. A topological characterization of weakness. In Proc24th Annual ACM SIGACT-SIGOPS Symposium on Principles Of Distributed Computing (PODC05), Las-Vegas, Nevada, USA, July 2005.

[FL79] M. J. Fischer and R. E. Ladner. Propositional dynamic logic of regular programs. In *J. Comput. Syst. Sci.*, pages 18(2), 194–211, 1979.

[Fla04] Alon Flaisher. Enhanced vacuity detection in linear temporal logic. Master's thesis, Israel Institute of Technology (The Technion), Haifa, Israel, August 2004.

[Kar03] Christian Josef Kargl. A sugar translator. Master's thesis, Institut für Softwaretechnologie, Technische Univesität Graz, Graz, Austria, December 2003.

[MH84] S. Miyano and T. Hayashi. Alternating finite automata on ω-words. *Theoretical Computer Science*, 32:321–330, 1984.

[MSS88] D. E. Muller, A. Saoudi, and P. E. Schupp. Weak alternating automata give a simple explanation of why most temporal and dynamic logics are decidable in exponential time. In *Proceedings of 3rd IEEE Symposium on Logic in Computer Science*, pages 422–427, 1988.

[Var95] Moshe Y. Vardi. An automata-theoretic approach to linear temporal logic. In *Banff Higher Order Workshop*, pages 238–266, 1995.