Expressiveness of Temporal Logic

Roberto Borelli
borelli.roberto@spes.uniud.it

Abstract

In this work, I consider the expressive power of various temporal logics. First, I recall some basic results about expressiveness of first order logic. Then I consider the case of *LTL* and I show a theorem that can be used to prove that the concept of parity is not definable in this context. I discuss a counterexample that proves that the mentioned theorem doesn't directly apply to *LTL+P* and I briefly highlight how a possible investigation may lead to a generalization of the theorem to the *LTL+P* case. Next, I relate first order definable languages with *LTL* ones and I present an extension to *LTL* which allows us to increase the expressive power and capture regular languages without changing the complexity of the decision procedure. Finally, I move to the more interesting case of interval logic. I introduce the notion of bisimulation and its use in modal logic and, in particular, I show how to apply it to prove that the logic $A\overline{A}$ is strictly more expressive than its future fragment A.

Contents

1 Preliminaries

1.1 Definitions and examples

Expressiveness is one of the most important characteristics of a logic, in combination with succinctness, and decidability. This manuscript is devoted to expressive power of some of the commonly used temporal logics. Model theory offers a way to measure expressive power [Stu00], in particular, we often deal with the following two fundamental problems:

- Which kind of structures are definable in a logic $\mathcal{L}$?
- Which pairs of structures are distinguishable by means of formulae of logic $\mathcal{L}$?

I now define more precisely what it means to define structures in a logic. Let $\mathcal{L}$ be a logic, let σ be a signature, and let C be a class of σ-structures. The class C is said to be **$\mathcal{L}$-elementary** if there exists a possibly infinite set of formulae $\tau = \{\tau_1, \tau_2, \tau_3, \dots\}$ such that for each σ-structure s it holds $s \in C \iff s \models \tau$. If the set of formulae τ is finite C is said **$\mathcal{L}$-basic elementary**. The latter definition is the most interesting one in systems' verification: if we want to check that a system s satisfies a certain property P (which characterizes a class of systems S), we want to make sure that P is finitely describable in the logic of consideration. Note that if in the logic $\mathcal{L}$ we allow the *and* operator, asking for the existence of a finite set $\tau = \{\tau_1, \dots, \tau_n\}$ is equivalent to asking for the existence of a single formula φ (which can be obtained by the conjunction of all the elements of the set).

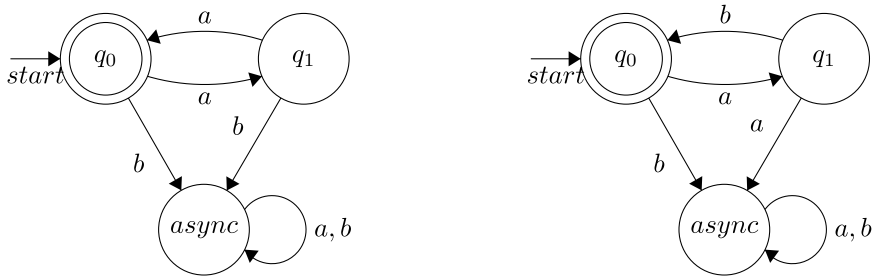
Examples with first order logic. To make definitions and the contest clearer I consider some examples in first order logic.

- The class of finite structures is not elementary, that is, we do not have a set of formulae (neither finite nor infinite) which is satisfied by all and only finite structures. This result is one of the most important applications of the compactness theorem of first order logic.
- The class C of structures with infinite domains is elementary but not basic elementary. To prove the class is elementary we consider $\tau = \left\{\exists x_1 \exists x_2 (x_1 \neq x_2), \dots, \exists x_1 \dots \exists x_n \bigwedge_{1 \leq i < j \leq n} x_i \neq x_j, \dots\right\}$ and it is immediate to see that $\mathcal{A} \models \tau$ if and only if $\mathcal{A}$ has an infinite domain. To prove that C is not basic-elementary is sufficient to observe that $\overline{C}$ is not elementary.

In dealing with finite structures the compactness theorem of first order logic is not working anymore and tools have been developed to prove inexpressiveness. In particular, I briefly recap the main steps which are required to use Ehrenfeucht-Fraïssé games to prove that the class C is not expressible in first order logic.

- For each n find two structures G_n and G'_n.
- Show with Ehrenfeucht-Fraïssé games that $G_n \equiv_n G'n$.
- Show that $G_n \in C$ but $G'_n \notin C$.
- Conclude that C cannot be characterized in first order logic.

With Ehrenfeucht-Fraïssé games we can also prove 0/1 laws and locality theorems of first order logic which are nothing but easier ways to show inexpressiveness of FO logic. Moving to the logic FO^k (which is FO with at most k different variables' names), we have a slightly different variant of Ehrenfeucht-Fraïssé games which are called Pebble games. Intuitively Pebble games allow Duplicator to win more easily, in fact, with at most k variables, formulae are less expressive and may distinguish *less* structures than FO.



(a) The minimum DFA for $(aa)^*$ is not counter-free

(b) The minimum DFA for $(ab)^*$ is counter-free

Figure 1: $(ab)^*$ is star-free while $(aa)^*$ is not.

Structure of the present work. Having recalled some basic results of first order logic I now consider the case of temporal logic. In particular, we are interested in examining which tools can be used to prove inexpressiveness of properties on a logic that allows temporal modalities. First, in section 1.2, I introduce the running example *even* and the associated results in first order logic. In section 2, I then consider the same example in *LTL* and with theorem 1 I prove that *even* is not capturable by *LTL*. I then show in section 2.4 how to extend the language and how to express *even* in this latter formalism.

Finally, in section 3, I move to propositional neighbourhood logic and I show, using bisimulation, that the logic $A\overline{A}$ is strictly more expressive than its future fragment A. It is interesting to note that in some sense, the proof of corollary 2 (for the *LTL* case) and the proof of theorem 7 (for the interval logic case), have the same structure of the schema-proof of inexpressiveness of *FO* through Ehrenfeucht-Fraïssé games as I have recalled in the current section.

1.2 Even in *FO*

One of the most symbolic results to see the limits of regular star-free languages is the concept of *even* which cannot be characterized by first order definable languages. Consider the alphabet $\Sigma = \{a, b\}$. The language $(aa)^*$ is regular but not star-free. The easiest way to see this is by noticing that the minimum DFA of the language associated with the expression $(aa)^*$ is not counter-free as can be seen in figure 1(a).

The same result can be obtained with EF games. Consider the model theoretic structure $\mathcal{A}_k = (D, +1, <, Q_a)$ with $D = Q_a = \{0, \dots, k-1\}$ associated to the word a^k. We can now associate to the language of the expression $(aa)^*$ the set of structures $EVEN_{aa} = \{\mathcal{A}_n : n = 0, 2, 4, \dots\}$. For each $k \geq 0$ duplicator has a winning strategy in the game $EF(\mathcal{A}_{2^k}, \mathcal{A}_{2^k+1})$ with k moves and hence $\mathcal{A}_{2^k} \equiv_k \mathcal{A}_{2^k+1}$. Observe that $\mathcal{A}_{2^k+1} \in EVEN_{aa}$ but $\mathcal{A}_{2^k} \notin EVEN_{aa}$ and so a *FO*-formula which is satisfied by all and only the structures in $EVEN_{aa}$ not exists.

On the contrary, the language defined by $(ab)^*$ is star-free and in fact, the minimum DFA associated with the language is counter-free as can be seen in figure 1(b). This also means that we can build a generalized regular expression with generalized star-height 0. We have:

$$(ab)^* = \overline{b\Sigma^* \cup \Sigma^* a \cup \Sigma^* aa\Sigma^* \cup \Sigma^* bb\Sigma^*} = \overline{b\overline{\emptyset} \cup \overline{\emptyset}a \cup \overline{\emptyset}aa\overline{\emptyset} \cup \overline{\emptyset}bb\overline{\emptyset}}$$

In the first rewriting we are saying that a word w which doesn't start with a b, doesn't end with an a, doesn't have two consecutive a, and doesn't have two consecutive b is a word in $(ab)^*$. In the second rewriting, we use the fact that Σ^* is the complement of the empty language.

Moving to ω-words. Considering ω-languages we have to consider a different notion of parity since we have $(pp)^\omega = (p)^\omega$ and so $(pp)^\omega$ loses the concept of *even*. First I define the notion of ω-language ω-star-free. We say that an ω-language $L \subseteq A^\omega$ is ω-star-free if it can be written as a finite union of sets $U \cdot V^\omega$ where $U, V \subseteq A^*$ are star-free and $V \cdot V \subseteq V$. Consider now the alphabet $\Sigma = \{p, \neg p\}$. The

(a) NBA for $(p \cdot (p + \neg p))^\omega$

(b) Counter-free NBA for $(p \cdot \neg p)^\omega$

Figure 2: $(p \cdot \neg p)^\omega$ is ω-star-free while $(p \cdot (p + \neg p))^\omega$ is not.

language $(p \cdot \neg p)^\omega$ is ω-star-free. In fact, we can rewrite $(p\ \neg p)^\omega = \emptyset \cdot ((p \cdot \neg p)^*)^\omega$ where $\emptyset$ is trivially star-free, $(p \cdot \neg p)^*$ is star-free as seen in the examples above[1] and it holds $(p \cdot \neg p)^* \cdot (p \cdot \neg p)^* \subseteq (p \cdot \neg p)^*$. In figure 2(b) I show a counter-free non deterministic Büchi automaton for the language $(p \cdot \neg p)^\omega$. Consider now the language $EVEN_{POSITIONS}$ defined by the ω-regular expression $(p \cdot (p + \neg p))^\omega$. As can be shown with similar arguments as in the finite setting, $EVEN_{POSITIONS}$ is not ω-star-free. In figure 2(a) I show a Büchi automata for $EVEN_{POSITIONS}$. The notion of counter-free Büchi automata and the relation with ω-star-free languages can be found in [DG08]. It is important to remark that for the finite setting, $L \subseteq \Sigma^*$ is star-free if and only if the minimum DFA is counter-free, while an ω-language $L \subseteq \Sigma^\omega$ is ω-star-free if and only if L is accepted by *some* counter-free Büchi automaton.

[1] Substitute p with the letter a and $\neg p$ with the letter b

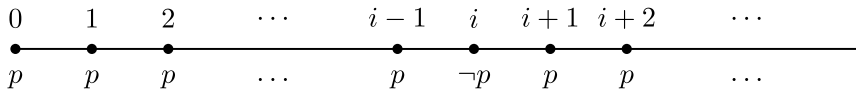


Figure 3: The sequence $p^i(\neg p)p^\omega$

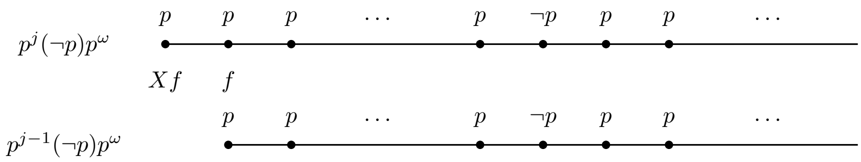


Figure 4: The following substitution holds: $eval_j(X_f) = eva_{j-1}(f)$

2 Expressive power of *LTL*

I now consider the propositional linear temporal logic. I will first examine the behavior of this logic concerning the example *even* which I considered also for first order logic in section 1.2. I then consider the relationship between *FO* and *LTL* and finally, I discuss two extensions of *LTL*, which are called *QLTL* and *ETL*.

2.1 Even in *LTL*

To extend the concept of *even* to *LTL*, consider the set of propositional letters $AP = \{p\}$, we are interested in characterizing with an *LTL* formula the set of computations on which p is true on all even states. More formally we consider the set

$$EVEN_{COMPUTATIONS}(p) = \{\sigma : \ \sigma \text{ is a computation on which } p \text{ is true on all even states}\}$$

Notice that a computation in $EVEN_{COMPUTATIONS}(p)$ can be naturally converted into an ω-word of the ω-regular (but not ω-star-free) language $EVEN_{POSITIONS}$ defined in the previous section. Through this section, I will prove that this set of computations is not expressible in *LTL*. First notice that we can build a formula like $\psi = p \wedge G(p \to X\neg p) \wedge G(\neg p \to Xp)$, but observe that ψ is satisfied only by the sequence where p is true in all even states and p is false on all odd states while in general, a computation $\sigma \in EVEN_{COMPUTATIONS}(p)$ doesn't have any constraint on p on odd states. This formula ψ is exactly the counterpart of the expression $(p \cdot \neg p)^\omega$ which I took into account in the last section and I showed that defines an ω-star-free ω-language. In general each *LTL* formula ψ which is claimed to express $EVEN_{COMPUTATIONS}(p)$, is wrong in discriminating an infinite number of models. Notice that in this case if we consider the infinite set $\tau = \{p, XXp, XXXXp, \dots\}$ is easy to see that $\sigma \models \tau \iff \sigma \in EVEN_{COMPUTATIONS}(p)$.

The following theorem is a quite general result that I will use to show the inexpressiveness of $EVEN_{COMPUTATIONS}(p)$. Before enunciating the theorem, consider the sequence $p^i(\neg p)p^\omega$ on which p is always true except on position i. Figure 3 clarifies the meaning of the sequence.

Theorem 1. *Let $f(p)$ be an LTL formula, let n denote the number of X (next) operators in f. Every sequence $p^i(\neg p)p^\omega$ where $i > n$ has the same truth value on f, or in other words*

$$p^{(n+1)}(\neg p)p^\omega \models f(p) \iff p^{(n+2)}(\neg p)p^\omega \models f(p) \iff \dots$$

Proof. Denote by $eval_i(f)$ the truth value of f on the sequence $p^i(\neg p)p^\omega$, we want to prove that $eval_{n+1}(f(p)) = eval_{n+2}(f(p)) = eval_{n+3}(f(p)) = \dots$
In other words, we want to prove that $eval_i(f(p))$ is independent of $i > n$.

Consider $j > n$, we proceed by induction on the structure of the formula $f(p)$ (we suppose the theorem holds on all the sub-formulas):

- Case $f(p) = p$. We always have $eval_i f(p) = T$.

- Case $f(p) = \neg p$. We always have $eval_i f(p) = F$.
- Cases $f(p) = f_1 \vee f_2$, $f(p) = f_1 \wedge f_2$ and $f(p) = \neg f_1$. The thesis follows immediately by the inductive hypothesis.
- Case $f(p) = Xf$. We have $eval_j(Xf) = eval_{j-1}(f)$. f contains $n-1$ next operators so we have $j-1 > n-1$ and hence by the inductive hypothesis the value of $eval_{j-1}(f)$ is independent of j. See figure 4.
- Case $f(p) = f_1\ U\ f_2$. From the following substitutions
  - $(f_1\ U\ f_2) = f_2 \vee (f_1 \wedge X(f_1\ U\ f_2))$
  - $eval_j(Xf) = eval_{j-1}(f)$

  we obtain

  $$eval_j(f_1\ U\ f_2) = eval_j(f_2) \vee (eval_j(f_1) \wedge eval_{j-1}(f_1\ U\ f_2))$$

  and *unfolding* n times

  $$\begin{aligned} eval_j(f_1\ U\ f_2) = {} & eval_j(f_2) \vee (eval_j(f_1) \wedge eval_{j-1}(f_2) \\ & \vee (eval_{j-1}(f_1) \wedge \cdots \wedge (eval_{n+1}(f_2) \vee (eval_{n+1}(f_1) \wedge eval_n(f_1\ U\ f_2)))\ldots)) \end{aligned}$$

  Both f_1 and f_2 still have n next operators so, by the inductive hypothesis

  $$\begin{aligned} eval_j(f_1) = eval_{j-1}(f_1) = \cdots = eval_{n+1}(f_1) \\ eval_j(f_2) = eval_{j-1}(f_2) = \cdots = eval_{n+1}(f_2) \end{aligned}$$

  so we have:

  $$\begin{aligned} eval_j(f_1\ U\ f_2) &= eval_{n+1}(f_2) \vee (eval_{n+1}(f_1) \wedge eval_{n+1}(f_2) \vee (eval_{n+1}(f_1) \wedge \cdots \wedge (eval_{n+1}(f_2) \\ &\qquad \vee (eval_{n+1}(f_1) \wedge eval_n(f_1\ U\ f_2)))\ldots)) \\ &= eval_{n+1}(f_2) \vee (eval_{n+1}(f_1) \wedge eval_n(f_1\ U\ f_2)) \end{aligned}$$

  and hence for each $j > n$ the value of $eval_j(f_1\ U\ f_2)$ can be determined by the evaluation of $f_1\ U\ f_2$ on the sequence $p^{n+1}(\neg p)p^{\omega}$. Symbolically:

  $$eval_{n+1}(f_1\ U\ f_2) = eval_{n+2}(f_1\ U\ f_2) = eval_{n+3}(f_1\ U\ f_2) = \ldots$$

  which is exactly what we were trying to show.

□

Corollary 1. *For each $m \geq 2$ the set $m_{COMPUTATIONS}(p)$*

$$m_{COMPUTATIONS}(p) = \{\sigma :\ \sigma \textit{ is a computation on which } p \textit{ is true at each state multiple of } m\}$$

is not expressible in LTL

Proof. Suppose that the formula $f(p)$ with n X operators captures $m_{COMPUTATIONS}(p)$ for a given m. For each k such that the multiple km of m is greater than the number of next operators in f, observe that $p^{km+1}(\neg p)p^{\omega}$ is in the set $m_{COMPUTATIONS}(p)$ while $p^{km}(\neg p)p^{\omega}$ is not. In fact, in the latter model p is always true except on positions km which is a multiple of m and by the definition of $m_{COMPUTATIONS}(p)$ we should have p true on every position multiple of m.

By the previous theorem $eval_{km+1}(f(p)) = eval_{km}(f(p))$ so the thesis follows immediately. □

Considering back the formula $\psi = p \wedge G(p \rightarrow X\neg p) \wedge G(\neg p \rightarrow Xp)$ which has two next operators we can now say that it doesn't characterize $EVEN_{COMPUTATIONS}(p)$. In fact, we have $p^4(\neg p)p^{\omega} \notin EVEN_{COMPUTATIONS}(p)$ and $p^5(\neg p)p^{\omega} \in EVEN_{COMPUTATIONS}(p)$ but $p^4(\neg p)p^{\omega} \not\models \psi$ and $p^5(\neg p)p^{\omega} \not\models \psi$.

The case of *LTL + P*. In this small paragraph, I present some considerations of mine which I didn't find in the literature. It is interesting to notice that theorem 1 talks about *LTL*, but recall that in this setting adding the past modality doesn't add expressive power[2] (even if *LTL+P* formulae may be exponentially more succinct than *LTL* ones). The question now is *Does theorem 1 holds also for LTL + P formulae?* I came up with a simple negative answer which I explain in the following. First, recall that *LTL + P* formulae are able to *count*. In particular, considering Y the *yesterday operator* and Z the *weak yesterday operator* we have:

- $\sigma, i \models Z\phi \iff i > 0 \to \sigma, i-1 \models \phi$
- $\sigma, i \models Z\, \mathit{false} \iff i = 0$
- $\sigma, i \models ZZ\, \mathit{false} \iff i \leq 1$
- $\sigma, i \models ZZZ\, \mathit{false} \iff i \leq 2$
- ...

- $\sigma, i \models Y\phi \iff i > 0 \wedge \sigma, i-1 \models \phi$
- $\sigma, i \models Y\, \mathit{true} \iff i \geq 1$
- $\sigma, i \models YY\, \mathit{true} \iff i \geq 2$
- $\sigma, i \models YYY\, \mathit{true} \iff i \geq 3$
- ...

where σ is a computation and i is a position in it. Combining these kinds of formulas we can express for example the property $i = 2$ in fact, $\sigma, i \models YY\, \mathit{true} \wedge ZZZ\, \mathit{false}$ if and only if $i = 2$.

Lemma 1. *Theorem 1 cannot be directly applied to LTL+P formulas.*

Proof. For a fixed $k > 1$, consider the *LTL+P* formula $\psi_k = pU(\neg p \wedge XGp) \wedge F(\neg p \wedge Y^k \mathit{true} \wedge Z^{k+1} \mathit{false})$. The number of next operators in ψ_k is 1 but it holds $p^k(\neg p)p^\omega \models \psi_k$ and $p^{k+1}(\neg p)p^\omega \not\models \psi_k$ contradicting theorem 1. In fact, for a computation σ it holds $\sigma \models pU(\neg p \wedge XGp)$ iff $\sigma = p^i(\neg p)p^\omega$ for some i, and, $p^i(\neg p)p^\omega \models \psi_k$ if and only if $i = k$. $\square$

Despite the fact that theorem 1 cannot be directly applied to *LTL+P* formulas, we may still wonder if a similar result can be obtained. I suspect (even though I didn't find it in the literature) that this could be the case. To obtain such a result, one may consider a translation algorithm for *LTL+P* formulas. An *efficient* algorithm to translate *LTL+P* formulae into *LTL* ones passes first through counter-free Büchi automata, second to deterministic Muller automata and finally, the translation to *LTL* is obtained. With this algorithm, the size of the translated formula can be up to triply exponential in the size of the original one. It would be of interest to know if (and how) the number of next operators in the translated *LTL* formula is related to the initial *LTL+P* formula. Forgetting about translation algorithms, it would be of interest to obtain a result similar to theorem 1 considering the semantics of past operators from the beginning, for example considering the number of *yesterday operators*. Note that, although the number of next operators of the formula ψ_k of the proof of lemma 1 is 1, ψ_k contains k *yesterday* operators and $k+1$ *weak yesterday* operators. Furthermore, if we rewrite ψ_k in terms of *yesterday* operators (using the substitution $Z\phi \equiv \neg Y \neg \phi$), in total the obtained formula contains 1 next operator and $2k+1$ yesterday operators.

2.2 LTL and *FO*

The result of section 2.1 was not a case, in fact, we have an equivalence between *LTL* and *FO* theory of discrete linear orders which was first proved by Kamp [Kam68] in his Phd dissertation. We can consider the equivalence in two theorems. The first theorem that I present will tell us how to rewrite $(LTL + P)[XU, XS]$ formulae in first order logic with at most three variables. Here we consider *LTL* with past modality where XU and XS denote the operators *Strong Strict Until* and *Strong Strict Since* respectively. The other direction from first order logic to *LTL* is more complex and involves the so-called *separation theorem* and I don't report the proof, which can be found for example in [DG08]. Note that other standard operators may be derived by formulas containing XU and XS. As an example, the next operator is defined as follows: $X\varphi = \mathit{false}\ XU\ \varphi$.

Theorem 2. $(LTL + P)[XU, XS] \subseteq FO^3[<]$

[2]This is not always the case, as we will see in section 3, there are logics in which the addition of past modalities strictly increases the expressiveness.

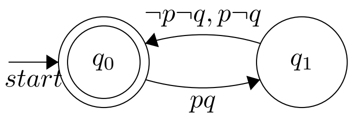


Figure 5: Counter-free NBA for the language $(pq \cdot (\neg p\neg q + p\neg q))^\omega$ over the alphabet $\Sigma = \{\neg p\neg q, \neg pq, p\neg q, pq\}$. Notice that this language is *more similar* to $(p \cdot \neg p)^\omega$ (figure 2(b)) which is ω-star-free than the language $(p \cdot (p + \neg p))^\omega$ (figure 2(a)) which is not ω-star-free.

Proof. The proof resembles the definition of the semantics of *LTL* formulae. To each *LTL* formulae φ over the set $AP = \{p_1, \ldots, p_n\}$, we associate a first order formula $\varphi(x)$ over the language $(0, <, +1, Q^{(1)}_{p_1}, \ldots, Q^{(1)}_{p_n})$ with one free variable and at most three different variables' names. We proceed by induction on the structure of the formula.

- Case $\varphi = p$ with $p \in AP$. $p(x) = Q_p(x)$.
- Propositional cases are straightforward.
- Case $\varphi = \varphi_1 XU \varphi_2$. $(\varphi_1 XU \varphi_2)(x) = \exists y\ y < x \wedge \varphi_2(y) \wedge \forall z\ x < z < y \rightarrow \varphi_1(z)$.
- Case $\varphi = \varphi_1 XS \varphi_2$. $(\varphi_1 XU \varphi_2)(x) = \exists y\ y > x \wedge \varphi_2(y) \wedge \forall z\ x > z > y \rightarrow \varphi_1(z)$.

To each *LTL* formula φ we associate the language $\mathcal{L}(\varphi) = \{w \in (2^{AP})^\omega : w, 0 \models \varphi\}$. It is immediate to see that the *LTL* formula φ defines the same language of the FO^3 formula $\varphi(x)$. $\square$

Theorem 3. $FO[<] \subseteq (LTL + P)[XU, XS]$

With theorems 2 and 3 we are essentially saying that languages capturable by *LTL* formulae are exactly star-free languages. Combing the rewriting of the theorems, given an $FO[<]$ formula we can translate it into an $(LTL + P)[XU, XS]$ formula and then back to $FO^3[<]$ and so we have the following corollary.

Corollary 2. $FO[<] = FO^3[<]$ $\square$

2.3 Express Even in *LTL*

The *even* property is very simple and so Wolper [Wol83] considered two solutions to express the desired property in *LTL*.

1. Add quantifiers to *LTL* and use the following formula.

$$\psi(p) = \exists q (q \wedge G(q \rightarrow X\neg q) \wedge G(\neg q \rightarrow Xq) \wedge G(q \rightarrow p))$$

   The existential quantification $\exists x\ \varphi$ is interpreted over a new variable not in the set of atomic propositions, namely, φ is an *LTL* formula over the set of propositions $AP \cup \{x\}$. We say that the computation σ (at position 0) over AP satisfies $\exists x\ \varphi$ if and only if there exists a computation σ' over $AP \cup \{x\}$ (called an x-variant computation of σ) which agree on all the propositions in AP and such that $\sigma', 0 \models \varphi$. The logic obtained in this way is called *QLTL* and its satisfiability problem is non-elementary [KPV01]. With the above semantics, the proposed *QLTL* formula $\psi(p)$ correctly ensures that p is true on all even states and says nothing of p on odd states. In particular, note that q is a fresh proposition not in AP and is constrained to be true on all and only even states.

2. Add the operator $even(p)$ directly in the syntax of *LTL*

Regarding point 1, one may notice that using from the beginning the set $AP = \{p, q\}$ and using a standard *LTL* formula we can write $\psi = q \wedge G(q \rightarrow X\neg q) \wedge G(\neg q \rightarrow Xq) \wedge G(q \rightarrow p)$ which correctly models the desired property of p. Apparently this may seem a contradiction with theorems 2 and 3. The point is that the language defined by ψ is a star-free one. To see this, note that we can use the alphabet $\Sigma = \{\neg p\neg q, \neg pq, p\neg q, pq\}$ and the Büchi automation (see figure 5) corresponding to ψ is counter-free. Trivially we can also build a corresponding ω-star-free expression and an *FO* formula. The *takeaway* is that before concluding immediately that a given language is star-free, one has to look carefully because, as we have seen, little variations of the same language can make the difference.

Regarding point 2, the question now is which kind of operators can we add to *LTL* to increase its expressive power while keeping its decision problem *PSPACE*-complete? It turns out that we can add an infinite number of operators, each one corresponding to a right-linear grammar. The logic obtained in this way is called **ETL** (Extended Temporal Logic). In section 2.4 I formally define *ETL* and I show how to express *even* in *ETL*.

2.4 Extended temporal logic

Consider a right linear grammar of the form $G = (V, T, P, S)$ where V is the set of n variables (non-terminals) $V = \{V_1, \ldots, V_n\}$, T is the set of m terminals $T = \{t_1, \ldots, t_m\}$ and S is the initial symbol. P is the set of right-linear productions of the form (1) $V_i \to t_{ij}$ or (2) $V_i \to t_{ij}V_{ij}$. A grammar of this type can generate either an ω-word or a finite word, and so we say that $\mathcal{L}(G) \subseteq T^\infty$. An ω-word is obtained applying productions of type (2) ω times and if the grammar contains only productions of type (2) the grammar can only generate ω-words and so $\mathcal{L}(G) \subseteq T^\omega$. To each grammar G as above, and for each non-terminal V_i we have an m-ary operator (called grammar operator) of the form $\mathcal{G}_i(f_1, \ldots, f_m)$ directly into the syntax of *ETL*.

The syntax of *ETL* is the following:

$$\phi ::= AP \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid X\phi \mid \phi\ U\ \phi \mid \mathcal{G}_i(\phi, \ldots, \phi)$$

The semantics of *ETL* is the same as the one of *LTL* for propositional and temporal cases. We have just to specify the semantics of grammar operators. Given an *LTL* structure $M = (S, x, L)$ where S is the set of states, L is the labeling function and x is a sequence of states, we say $M, x \models \mathcal{G}_i(f_1, \ldots, f_m)$ if and only if there exists a word w generated by the grammar G starting from symbol V_i such that w is of the form $w = t_{w_0}t_{w_1}t_{w_2}\ldots (1 \le w_j \le m)$ and for all $j \ge 0$ it holds $M, x^j \models f_{w_j}$. Here f_i is a temporal formula (specified as i-th argument of the grammar operator) related to the terminal t_i.

Notice that both the *next* and *until* operators can be obtained as grammar operators so we could remove them from the syntax. For the case of the next operator, consider the grammar G^X with the set of productions $P^X = \{V_0 \to t_0V_1, V_1 \to t_1\}$. The operator $\mathcal{G}_0^X(T, f)$ is equivalent to Xf. For the case of the until operator, consider the grammar G^U with the set of productions $P^U = \{V_0 \to t_0V_0, V_0 \to t_1\}$. The operator $\mathcal{G}_0^U(f_1, f_2)$ corresponds to $f_1\ U\ f_2$. In similar ways we could also define grammar operators for the *always* and *eventually* temporal operators.

Even in *ETL*. To express *even* consider the grammar $G^{even} = \{\{S, R\}, \{a, b\}, \{S \to aR, R \to bS\}, S\}$. The language generated by G^{even} is $(ab)^\omega$ and so $\mathcal{G}_S^{even}(p, \neg p \vee p)$ expresses the property $even(p)$ in fact we have:

$$\begin{aligned}
M, x \models\ & \mathcal{G}_S^{even}(p, \neg p \vee p) \\
& \iff \forall j\ ((j \text{ is even} \to M, x^j \models f_a) \wedge (j \text{ is odd} \to M, x^j \models f_b)) \\
& \iff \forall j\ ((j \text{ is even} \to M, x^j \models p) \wedge (j \text{ is odd} \to M, x^j \models \neg p \vee p)) \\
& \iff \forall j\ (j \text{ is even} \to M, x^j \models p)
\end{aligned}$$

Complexity. As we have seen, *ETL* is strictly more expressive than *LTL* but if the decision problem related to *ETL* was undecidable, then there would be no practical uses. It turns out that the addition of right-linear grammar operators to *LTL* doesn't worsen the decision problem and through this short paragraph, I'm going to recap some basic results. The proofs are not detailed since this is not the main topic of the present work.

Theorem 4. *The satisfiability problem for ETL is PSPACE-hard*

Proof sketch. We consider the problem *finite automation inequivalence* which is known to be *PSPACE*-complete. Given as input two DFAs A and B (on alphabet Σ), we construct right-linear grammars G^A and G^B which generate the same languages respectively. We consider a particular augmented alphabet $\Sigma_1 = \{v, v' : v \in \Sigma\}$ with size $2|\Sigma|$. Primed symbols in Σ_1 will mark the final character of generated words. We rewrite grammars G^A and G^B in terms of this new alphabet and we obtain G^{A_1} and G^{B_1}. To the grammars G^{A_1} and G^{B_1} correspond two grammar operators $\mathcal{G}^{A_1}$ and $\mathcal{G}^{B_1}$ with $2|\Sigma|$ arguments. From this point, we can construct an *ETL* formula using $\mathcal{G}^{A_1}$ and $\mathcal{G}^{B_1}$ which is satisfiable if and only if $\mathcal{L}(A) \neq \mathcal{L}(B)$. $\square$

We are now able to observe that if context-free grammar operators were allowed in the *ETL* syntax then *ETL* would be undecidable since we could repeat the previous proof almost without changes.

The decision procedure investigated by Wolper [Wol83] is based on the fact that *ETL* has the *small model property* that is, a formula φ of *ETL* is satisfiable if and only if has a model of size less or equal than $f(|\varphi|)$ where f is a computable function $f : \mathbb{N} \to \mathbb{N}$. The proposed decision procedure is an extension of the tableau decision procedure for *LTL* by Manna and Pnueli (1981). The key points are the following.

- The extension of the notion of *fulfillment* with respect to grammar operators.
- The decomposition of grammar operators formulae into simpler ones. I don't show too many details but I just consider the example of $\mathcal{G}_S^{even}(p, p \vee \neg p)$ discussed above which expresses the *even* property. We have that $\mathcal{G}_S^{even}(p, p \vee \neg p)$ is satisfiable iff all the formulae in at least one set of $\{\{p, X\mathcal{G}_R^{even}(p, p \vee \neg p)\}\}$ are satisfiable and in a similar way $\mathcal{G}_R^{even}(p, p \vee \neg p)$ is satisfiable iff all the formulae in at least one set of $\{\{p \vee \neg p, X\mathcal{G}_S^{even}(p, p \vee \neg p)\}\}$ are satisfiable. The negation of grammar operators (the so-called *eventualities*) is, in some sense, more problematic. We have $\neg\mathcal{G}_S^{even}(p, p \vee \neg p)$ is satisfiable iff all the formulae in at least one set of $\{\{\neg p \vee X\neg\mathcal{G}_R^{even}(p, p \vee \neg p)\}\}$ are satisfiable. To satisfy $\neg\mathcal{G}_S^{even}(p, p \vee \neg p)$ we have to check that p is false on at least one even state and through the notion of *fulfillment* we want to make sure that the check is not always postponed.

This tableau procedure is a proof that *ETL* is decidable but the procedure takes time and space exponential in the size of the formula to check for satisfiability. I conclude this presentation of *ETL* with the following theorem which can be shown by the construction of a non-deterministic version of the tableau procedure which only requires polynomial space.

Theorem 5. *The satisfiability problem for ETL is in PSPACE*

As a corollary of theorems 4 and 5, the satisfiability problem for *ETL* is *PSPACE*-complete exactly as the satisfiability problem for *LTL*.

3 Expressive power of *PNL*

In this final section, I now consider the case of propositional neighborhood logic (*PNL*) $A\overline{A}$ and its future fragment A. The main result will be that the $\langle\overline{A}\rangle$ modality is not definable in A and thus $A\overline{A}$ is strictly more expressive than A. First I define syntax and semantics of interval temporal logics, I then define the notion of bisimulation which will be used to prove inexpressiveness and finally, I move to the main result.

3.1 Halpern-Shoham's interval logic

In section 2 we considered the logic *LTL* which is based on time-points. Now we consider strict intervals as the temporal unit. Let $\mathbb{D} = (\mathcal{D}, <)$ be a linearly-ordered domain (where linear means that every two points in $\mathcal{D}$ are comparable). A strict interval on $\mathbb{D}$ is of the form $[i,j]$ where $i, j \in \mathcal{D}$ and $i < j$. On the contrary, if we allow intervals to collapse into single points, we talk about non-strict intervals. We call *interval structure* $\mathbb{I}(\mathbb{D})$ the set of intervals over $\mathbb{D}$. Standard interval structures are obtained considering $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$ with their natural orders. While between pairs of points we have three ordering relations, between intervals we have thirteen ordering relations which are called *Allen's interval relations.* The thirteen relations (except for the equality) are:

meets	$[i,j]\ R_A\ [i',j'] \iff j = i'$	is met by	$[i,j]\ R_{\overline{A}}\ [i',j'] \iff j' = i$
later	$[i,j]\ R_L\ [i',j'] \iff i' > j$	is preceded by	$[i,j]\ R_{\overline{L}}\ [i',j'] \iff i > j'$
begins	$[i,j]\ R_B\ [i',j'] \iff i = i' \wedge j' < j$	is started by	$[i,j]\ R_{\overline{B}}\ [i',j'] \iff i' = i \wedge j < j'$
ends	$[i,j]\ R_E\ [i',j'] \iff j = j' \wedge i < i'$	is finished by	$[i,j]\ R_{\overline{E}}\ [i',j'] \iff j' = j \wedge i' < i$
during	$[i,j]\ R_D\ [i',j'] \iff i < i' \wedge j' < j$	contains	$[i,j]\ R_{\overline{D}}\ [i',j'] \iff i' < i \wedge j < j'$
overlaps	$[i,j]\ R_O\ [i',j'] \iff i < i' < j < j'$	is overlapped by	$[i,j]\ R_{\overline{O}}\ [i',j'] \iff i' < i < j' < j$

These relations are mutually exclusive, namely, between two intervals it holds at maximum one relation. They are also jointly exhaustive which means that between two intervals it holds at least one relation. To each one of the twelve above relation R_X we associate a unary modal temporal operator $\langle X\rangle$ in the logic $\mathcal{HS}$ which was introduced by Halpern and Shoham. In this logic, interval structures are treated as Kripke structures and Allen's relations are the accessibility relations in them. The syntax of the Halpern-Shoham logic of Allen's interval relations is the following:

$$\phi ::= p \mid \neg\phi \mid \phi \vee \phi \mid \langle X\rangle\phi$$

where $\langle X\rangle$ is any of the described temporal modalities and $p \in AP$ with AP denoting the set of atomic propositions. Let $\mathbb{I}(\mathbb{D})$ be an interval structure and let $V : AP \to 2^{\mathbb{I}(\mathbb{D})}$ be a labeling function that associates to each atomic proposition p the set of intervals on which p holds, we call the pair $M = (\mathbb{I}(\mathbb{D}), V)$ an *interval model.* While *LTL* formulas are interpreted at time instants, in this framework formulas are evaluated at intervals. The truth value of propositions is dealt by the labeling function, propositional cases are treated in the usual way and a formula with temporal modalities is managed through the corresponding accessibility relation in the Kripke structure. Formally, let M be an interval model and let $[i,j]$ be a strict interval, the semantics of the logic $\mathcal{HS}$ is the following:

- $M, [i,j] \models p \iff [i,j] \in V(p)$
- $M, [i,j] \models \neg\phi \iff M, [i,j] \not\models \phi$
- $M, [i,j] \models \phi_1 \vee \phi_2 \iff M, [i,j] \models \phi_1$ or $M, [i,j] \models \phi_2$
- $M, [i,j] \models \langle X\rangle\phi \iff$ there exists an interval $[i',j']$ such that $[i,j]\ R_X\ [i',j']$ and $M, [i',j'] \models \phi$

To obtain fragments of $\mathcal{HS}$ we can restrict the use of temporal operators. We denote by $X_1 \dots X_k$ the fragment of $\mathcal{HS}$ which only contains operators $\langle X_1 \rangle, \dots, \langle X_k \rangle$. Formally there are $2^{12} = 4096$ different fragments of $\mathcal{HS}$. $\mathcal{HS}$ is provable to be undecidable by a reduction from the non-halting problem, moreover, undecidability dominates the scene of $\mathcal{HS}$' fragments, in particular about 90% of them are proved to be undecidable in most of the natural classes of interval structures [Del11]. Considering the strict semantics, the operators $\langle A \rangle, \langle \overline{A} \rangle, \langle B \rangle, \langle \overline{B} \rangle, \langle E \rangle, \langle \overline{E} \rangle$ are enough to express all the others, in fact, the following equivalences hold:

- $\langle D \rangle \phi \equiv \langle B \rangle \langle E \rangle \phi$
- $\langle L \rangle \phi \equiv \langle A \rangle \langle A \rangle \phi$
- $\langle O \rangle \phi \equiv \langle E \rangle \langle \overline{B} \rangle \phi$
- $\langle \overline{D} \rangle \phi \equiv \langle \overline{B} \rangle \langle \overline{E} \rangle \phi$
- $\langle \overline{L} \rangle \phi \equiv \langle \overline{A} \rangle \langle \overline{A} \rangle \phi$
- $\langle \overline{O} \rangle \phi \equiv \langle B \rangle \langle \overline{E} \rangle \phi$

Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two fragments of $\mathcal{HS}$, we say that $\mathcal{F}_1$ is *strictly less expressive than* $\mathcal{F}_2$ (denoted by $\mathcal{F}_1 \prec \mathcal{F}_2$) if all operators $\langle X \rangle$ in $\mathcal{F}_1$ are definable in $\mathcal{F}_2$ but not vice-versa. In the following, we will consider the logic $A\overline{A}$ called *propositional neighborhood logic* (*PNL*), its future fragment $\overline{A}$ called *right propositional neighborhood logic* (*RPNL*) and the logic $A\overline{L}$ which is obtained from *PNL* substituting the past modality $\langle \overline{A} \rangle$ with the past modality $\langle \overline{L} \rangle$. In particular, we will show that $A \prec A\overline{A}$ and other related results. These three logics are interesting also because they are decidable, in particular, the decision problem for $A\overline{A}$ over the classes of all linear orders, well-orders, finite linear orders and the linear order over natural numbers is *NEXPTIME*-complete. The proof is a translation from *PNL*-formulae to $FO^2[<]$-formulae recalling that the decision problem for $FO^2[<]$ with the mentioned classes of orders is *NEXPTIME*-complete. I also mention that *PNL* was the first decidable genuine interval logic discovered where *genuine* means that it cannot be reduced directly to point-based temporal logic.

3.2 Bisimulation

To prove results about expressiveness, we will use the notion of bisimulation which is widely used in many areas of computer science and in particular in the context of modal logic. I introduce the notion of bisimulation and related properties. First I move away from $\mathcal{HS}$ and I define notions regarding modal logic in general. Then I instantiate the concept of bisimulation in our interval logic of consideration.

Bisimulation in modal logic. The key property of bisimulation with respect to modal formulas is the so-called *invariance property.* First I introduce some main definitions about modal logic and then I show the invariance theorem. Let O be a set of modal operators (we denote its element by $\Delta_0, \Delta_1, \dots$), and let ρ be a function that associates to each operator its arity. The pair $\tau = (O, \rho)$ is called a *modal similar type.* A modal language built on the similarity type τ and on a set of propositions AP has the following syntax:

$$\phi ::= p \mid \neg\phi \mid \phi \vee \phi \mid \Delta(\phi_1, \dots, \phi_{\rho(\Delta)})$$

If τ is a modal similar type we call M a τ*-model* the triple $M = (W, \{R_\Delta : \Delta \in \tau\}, V)$ where W is the universe, for each n-ary operator $\Delta \in \tau$ the relation R_Δ has arity $n+1$ and $V : W \to 2^W$ is the valuation function. The notion of ϕ satisfied at state w of the model M is inductively defined. Again, propositional letters are dealt through the function V and boolean cases are managed in the obvious way. For modal operators, we have:

- $M, w \models \Delta(\phi_1, \dots, \phi_{\rho(\Delta)})$ iff for some states $v_1, \dots, v_n \in W$ such that $(w, v_1, \dots, v_n) \in R_\Delta$ we have $M, v_i \models \phi_i$ for each i

We say that two models M and M' are *modally equivalent* ($M \equiv M'$) if they have the same of τ-theory (that is, the set of τ-formulas satisfied by all states). For states w in M and w' in M' we say $M, w \equiv M', w'$ if the two sets of formulas satisfied by M, w and M', w' respectively, coincide. Given two models M and M', the relation $Z \subseteq W \times W'$ is called a *bisimulation* if (1) Z-related states

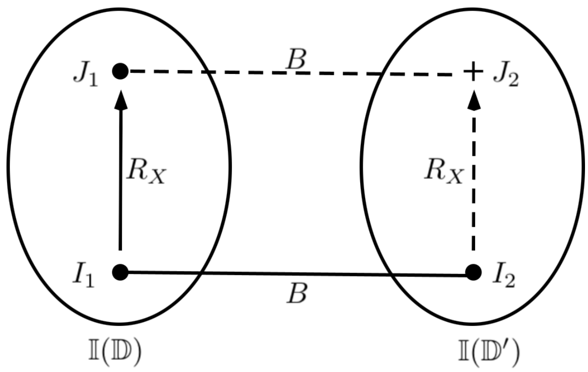


Figure 6: The forward condition of an $\mathcal{F}$-bisimulation on $M = (\mathbb{I}(\mathbb{D}), V)$ and $M' = (\mathbb{I}(\mathbb{D}'), V')$.

satisfy the same proposition letters, (2) whenever $(w, w') \in Z$ and $(w, v_1, \ldots, v_n) \in R_\Delta$ then there are $v'_1, \ldots, v'_n \in W'$ such that $(w', v'_1, \ldots, v'_n) \in R_\Delta$ and $(v_i, v'_i) \in Z$ for $1 \leq i \leq n$ and (3) whenever $(w, w') \in Z$ and $(w', v'_1, \ldots, v'_n) \in R_\Delta$ then there are $v_1, \ldots, v_n \in W$ such that $(w', v'_1, \ldots, v'_n) \in R_\Delta$ and $(v_i, v'_i) \in Z$ for $1 \leq i \leq n$. We are now ready to enunciate the main theorem which states that modal formulas are invariant under bisimulation.

Theorem 6. *Let τ be a modal similar type and let M and M' be τ-models. Then for each $w \in W$ and $w' \in W'$ we have that **if** there exists a bisimulation between M and M' which links w and w' **then** $M, w \equiv M', w'$.*

Proof. Consider the bisimulation Z between M and M' such that $(w, w') \in Z$. Suppose now $M, w \models \phi$ we have to show by induction on ϕ that $M', w' \models \phi$. I skip propositional cases which are straightforward. Consider $\phi = \Delta(\phi_1, \ldots, \phi_n)$. By definition, $M, w \models \phi$ iff there exist $v_1, \ldots, v_n \in W$ such that $(w, v_1, \ldots, v_n) \in R_\Delta$ and $M, v_i \models \phi_i$ for all i. By clause (2) of the definition of bisimulation we can now find elements $v'_1, \ldots, v'_n$ such that $(w', v'_1, \ldots, v'_n) \in R'_\Delta$ and $(v_i, v'_i) \in Z$ for all i. Combining the facts that ϕ_i is a sub-formula of ϕ, v_i is Z-related to v'_i and $M, v_i \models \phi_i$, by the inductive hypothesis we have that $M', v'_i \models \phi_i$ for all i and hence $M', w' \models \phi$. Suppose now that $M', w' \models \phi$ the proof that $M, w \models \phi$ is completely analogous and involves clause (3) of the definition of bisimulation. $\square$

The natural question is: can we invert the previous theorem? That is, if two models are modally equivalent, must they be bisimilar? This question has a negative answer, in fact, we can find two (infinite) non-bisimilar models which satisfy the same set of formulas. I don't touch more details which can be found in [BRV01].

Bisimulation in the logic $\mathcal{HS}$. Now I go back to the interval logic $\mathcal{HS}$ and I instantiate the general concept I described some lines above. Let $\mathcal{F}$ be a fragment of $\mathcal{HS}$. Consider two interval models $M = (\mathbb{I}(\mathbb{D}), V)$ and $M' = (\mathbb{I}(\mathbb{D}'), V')$, we say that a binary relation on the two interval structures $B \subseteq \mathbb{I}(\mathbb{D}) \times \mathbb{I}(\mathbb{D}')$ is an $\mathcal{F}$-bisimulation if:

1. For each pair of intervals in the relation $(I_1, I_2) \in B$ it is true that the two intervals I_1, I_2 satisfy the same propositional letters.

2. Let $\langle X \rangle$ be a temporal modality in the logic $\mathcal{F}$. For each pair of intervals in the relation $(I_1, I_2) \in B$ such that there exists another interval J_1 which satisfies $(I_1, J_1) \in R_X$ there must exist an interval J_2 which satisfies $(I_2, J_2) \in R_X$ and $(J_1, J_2) \in B$.

3. Let $\langle X \rangle$ be a temporal modality in the logic $\mathcal{F}$. For each pair of intervals in the relation $(I_1, I_2) \in B$ such that there exists another interval J_2 which satisfies $(I_2, J_2) \in R_X$ there must exist an interval J_1 which satisfies $(I_1, J_1) \in R_X$ and $(J_1, J_2) \in B$.

Condition 1 is also called *local consistency* and conditions 2 and 3 are called respectively *forward condition* and *backward condition.* In figure 6 I show the essence of the forward condition (the backward condition is completely symmetric): if we find $I_1 \in \mathbb{I}(\mathbb{D})$ and $I_2 \in \mathbb{I}(\mathbb{D}')$ which are B-related and $J_1 \in \mathbb{I}(\mathbb{D})$ such that I_1 and J_1 are R_X-related, we must find $J_2 \in \mathbb{I}(\mathbb{D}')$ such that J_1 and J_2 are B-related and I_2 and J_2 are R_X-related. If B is an $\mathcal{F}$-bisimulation on M and M', we say that intervals $I_1 \in \mathbb{I}(\mathbb{D})$ and $I_2 \in \mathbb{I}(\mathbb{D}')$ are $\mathcal{F}$-bisimilar if $(I_1, I_2) \in B$.

The key property which follows immediately from theorem 6 is that any $\mathcal{F}$-bisimulation preserves the truth of all formulas in $\mathcal{F}$. Now we can use this notion to talk about expressiveness. To prove that the modal operator $\langle X \rangle$ is not definable in $\mathcal{F}$ we just need to find two interval models M, M' and an $\mathcal{F}$-bisimulation B on them such that $M, I \models \langle X \rangle p$ and $M', J \not\models \langle X \rangle p$ where $I \in \mathbb{I}(\mathbb{D})$ and $J\mathbb{I}(\mathbb{D}')$ are two $\mathcal{F}$-bisimilar intervals, that is $(I, J) \in B$

3.3 Past operators in A

I now show how to use the previous notions to prove results about A, $A\overline{A}$, and $A\overline{L}$.

Theorem 7. *The modality $\langle \overline{L} \rangle$ is not definable in A over $\mathbb{N}$*

Proof. Consider $AP = \{p\}$. Let M and M' be two models over $\mathbb{N}$, $M = (\mathbb{I}(\mathbb{N}), V)$ and $M' = (\mathbb{I}(\mathbb{N}), V')$. Set $V(p) = V'(p) = \{[0,1], [1,2], [2,3], [3,4], \dots\} = \{[i, i+1] : i \geq 0\}$. Consider the binary relation $B \subseteq \mathbb{I}(\mathbb{N}) \times \mathbb{I}(\mathbb{N}) = B_{\text{id}} \cup B_{+1}$ where $B_{\text{id}} = \{([i,j],[i,j]) : 0 \leq i < j\}$ and $B_{+1} = \{([i,j],[i+1,j+1]) : 0 \leq i < j\}$. B_{id} is a sort of identity relation, if we fix a strict interval, it is always B_{id}-related to itself. B_{+1} is a sort of successor relation, if we fix a strict interval I, it is B_{+1}-related to the interval obtained by applying the successor function (defined on natural numbers) on the extreme points of I. Suppose that B is an A-bisimulation on M and M'. We have that $([1,2],[2,3]) \in B$ and $M', [2,3] \models \langle \overline{L} \rangle p$ in fact $M', [0,1] \models p$ since $([0,1],[2,3]) \in R_{\overline{L}}$. On the other side $M, [1,2] \not\models \langle \overline{L} \rangle p$ in fact there is no interval I which is $R_{\overline{L}}$-related to $[1,2]$ and such that $V(p) \subseteq \{p\}$. We conclude that $\langle \overline{L} \rangle$ is not definable in A. It remains to be shown that B is an A-bisimulation on M and M'. First, B-related intervals satisfy the same propositions, in fact, let $I = [i,j]$, define the function d such that $d(I) = j - i$, we have that a pair (I, J) of B-related intervals preserves the value of function d (namely $d(I) = d(J)$) and so $V(I) = \{p\} \iff V(J) = \{p\} \iff d(I) = 1 (= d(J))$ and $V(I) = \emptyset \iff V(J) = \emptyset \iff d(I) > 1 (= d(J))$. To show the forward condition, we have to consider two cases: B_{id}-related intervals and B_{+1}-related intervals. Consider $([i,j],[i,j]) \in B_{\text{id}}$, an interval in M reachable from an $\langle A \rangle$-move from $[i,j]$ is of the form $[j,k]$ but $[i,j]$ in M' is also R_A-related to $[j,k]$ in M' and $([j,k],[j,k]) \in B_{\text{id}}$. Finally consider the case $([i,j],[i+1,j+1]) \in B_{+1}$, we have that $([i,j][j,k]) \in R_A$ but we can chose the interval $[j+1,k+1]$ which satisfy $([i+1,j+1],[j+1,k+1]) \in R_A$ and $([j,k],[j+1,k+1]) \in B_{+1}$. The satisfaction of the backward condition can be done in a very similar way. □

Corollary 3. *We have that the logic RPNL is strictly less expressive than PNL and $A\overline{L}$. In symbols:*

1. $A \prec A\overline{L}$
2. $A \prec A\overline{A}$

Proof. Point 1 follows directly from the previous theorem. Point 2 follows from the previous theorem recalling that $\langle \overline{L} \rangle \phi \equiv \langle \overline{A} \rangle \langle \overline{A} \rangle \phi$. □

The proof of theorem 7 also works if $\mathbb{N}$ is replaced by $\mathbb{Z}, \mathbb{Q}$, or $\mathbb{R}$ while some work has to be done if we replace $\mathbb{N}$ by a finite prefix of it. Let $K = \{0, 1, 2, \dots, k\}$, consider $M = (\mathbb{I}(K), V)$ and $M' = (\mathbb{I}(K), V')$ with $V(p) = V'(p) = \mathbb{I}(K)$. The relation $B \subseteq \mathbb{I}(K) \times \mathbb{I}(K) = B_{\text{id(K)}} \cup \{([1,3],[2,3])\}$, where $B_{\text{id(K)}}$ is the relation B_{id} of the proof of theorem 7 restricted to K, is an A-bisimulation. Conditions for intervals in $B_{\text{id(K)}}$ can be checked exactly as done for the relation B_{id} in theorem 7 and forward/backward conditions for $([1,3],[2,3])$ are straightforward. Intervals $[1,3]$ and $[2,3]$ are A-bisimilar but $M, [1,3] \not\models \langle \overline{L} \rangle$ *true* while $M', [2,3] \models \langle \overline{L} \rangle$ *true*.

From corollary 3 we have that $A\overline{A}$ is at least as expressive as $A\overline{L}$ but we still don't know if $A\overline{A}$ and $A\overline{L}$ are equally expressive or not. The following theorem answers this question.

Theorem 8. *The modality $\langle \overline{A} \rangle$ is not definable in $A\overline{L}$ over $\mathbb{N}$*

Proof. We proceed as in the proof of theorem 7. Consider two models M and M', $M = (\mathbb{I}(\mathbb{N}), V)$ and $M' = (\mathbb{I}(\mathbb{N}), V')$ over $AP = \{p\}$. Set $V(p) = V'(p) = \mathbb{I}(\mathbb{N})$. Consider the binary relation $B \subseteq \mathbb{I}(\mathbb{N}) \times \mathbb{I}(\mathbb{N}) = B_{\text{id}} \cup \{([0,2],[1,2])\}$ where $B_{\text{id}} = \{([i,j],[i,j]) : 0 \leq i < j\}$ as in theorem 7. Suppose that B is an $A\overline{L}$-bisimulation, then we have that $[0,2]$ and $[1,2]$ are $A\overline{L}$-bisimilar and $M', [1,2] \models \langle \overline{A} \rangle$ *true* in fact, $M', [0,1] \models$ *true* and $([1,2],[0,1]) \in R_{\overline{A}}$. On the contrary, $M, [0,2] \not\models \langle \overline{A} \rangle$ *true* since if an interval I is of type $I = [0,k], k > 0$ then there is no interval J in M such that $(I, J) \in R_{\overline{A}}$. It remains to be shown that B is an $A\overline{L}$-bisimulation. The local condition is trivially satisfied since all intervals (even those not in B) satisfy the same proportions. Forward and backward conditions need to be checked both for intervals $([i,j],[i,j])$ and also for the pair $([0,2],[1,2])$. For a pair of intervals $([i,j],[i,j]) \in B_{\text{id}}$, for each interval $[h,k]$ such that $([i,j],[h,k]) \in R_{\overline{L}}$ we can choose $[h,k]$ in M' since $([i,j],[h,k]) \in R_{\overline{L}}$ and $([h,k],[h,k]) \in B_{\text{id}}$ and so the forward condition is satisfied for $R_{\overline{L}}$. The same argument holds if we replace $R_{\overline{L}}$ with R_A and also the two backward conditions are obtained in the same way. Consider now the pair $([0,2],[1,2])$. In M, pairs of intervals $([0,2],[2,j]), j > 2$ are R_A-related but observe that in M' pairs of intervals $([1,2],[2,j]), j > 2$ are also R_A-related and it holds $([2,j],[2,j]) \in B_{\text{id}}$ and so the forward condition for R_A is satisfied. We don't need to check the forward condition of $R_{\overline{L}}$ since there is no interval J such that $([0,2], J) \in R_{\overline{L}}$. Even in the case of $([0,2],[1,2])$ the two backward conditions can be checked analogously. □

Corollary 4. $A \prec A\overline{L} \prec A\overline{A}$ □

The proof of theorem 8 also works if $\mathbb{N}$ is replaced by any prefix of it. The result is still true on $\mathbb{Z}, \mathbb{Q}$, and $\mathbb{R}$ but a different bisimulation (which can be found in [DMS12]) needs to be used.

I conclude this section by enunciating (again, all details and proofs can be found in [DMS12]) some more interesting results about A and $A\overline{A}$. Considering first order formulas, it is easily provable with EF games that we cannot separate $(\mathbb{Q}, <)$ from $(\mathbb{R}, <)$. These two orders are not isomorphic: even though they are both infinite, dense, without maximum and without minimum, it is well known that the former has cardinality $\aleph_0$ and the latter has cardinality $2^{\aleph_0}$. Despite the fact that these two models are not isomorphic we don't have a first order formula that is satisfied in one model and not in the other. The question is what happens if we consider the logics *RPNL* and *PNL*. It has been shown that each $A\overline{A}$-formula which is satisfiable over $\mathbb{Q}$ is also satisfiable over $\mathbb{R}$ but not vice versa since there is an $A\overline{A}$-formula satisfied over $\mathbb{R}$ but not over $\mathbb{Q}$. The same is not true for A in fact, each A-formula satisfiable over $\mathbb{Q}$ is also satisfiable over $\mathbb{R}$ (since A is a fragment of $A\overline{A}$) but in this case also the converse direction holds.

4 Conclusions

In this work, I considered the expressive power of various temporal logics. In particular, I examined the example of *parity* in various settings. First I discussed the very well-known finite case with first order logic, star-free expression, and automata. I then moved to the infinite case, starting again from first order logic, ω-star-free expression, and Büchi automata. In section 2, I examined *parity* in *LTL* and showed how to prove that it is inexpressible. I showed two other formalisms, *QLTL* and *ETL*, which can be used to express *parity.* The final picture that I presented here is that star-free languages are expressible by counter-free automata, $FO[<]$, *LTL*, and *LTL+P*. On the other side, regular languages are capturable by Büchi automata, $MSO[<]$, *ETL* and *QLTL*. The concept of parity, both in the finite and infinite setting, is expressible by means of regular languages but not by star-free ones. Even though the languages recognized/generated by the above formalisms are the same, these formalisms all differ in terms of complexity for the decision procedure and in terms of succinctness, namely in terms of the size of the objects that we have to construct to define a given language. The logic *ETL* turns out to be practical since the complexity of the decision procedure is the same as the one for *LTL*.

In section 3, I considered the interval logic $\mathcal{HS}$ and its related fragments. First I defined the formalism based on time intervals instead of time points. Then I showed how the notion of bisimulation applies to modal logic and in particular to our case of consideration $\mathcal{HS}$. Bisimulation turns out to be a tool to prove inexpressiveness of certain properties. It's interesting to see that its use is very similar to the one of EF games. This is not a case, in fact, the concept of isomorphism and the one of bisimulation are related (even though I didn't examine their relations in the present paper). I used bisimulations to show that adding past modalities (like $\langle \overline{A} \rangle$ and $\langle \overline{L} \rangle$) to the fragment A of $\mathcal{HS}$ changes its expressive power. In particular, I showed that *RPNL* is strictly less expressive than $RPNL + \overline{L}$ which is strictly less expressive than *PNL*. These results are in contrast to what happens to *LTL*: adding past operators to *LTL* doesn't add expressiveness even though it adds succinctness. There exists an algorithm that translates $LTL + P$ formulae into *LTL* ones, but the size of the translated formula can be up to triply exponential with respect to the size of the original formula. An algorithm to translate $A\overline{A}$ formulae to A formulae simply cannot exist. To conclude the work, I briefly discussed the separation problem of $\mathbb{Q}$ and $\mathbb{R}$ in *FO* and $A\overline{A}$.

Future investigations. As pointed out in section 2, theorem 1 doesn't directly apply to the *LTL+P* case but it would be nice to have a similar result that works for formulas with past modalities. To this end, a possible investigation can proceed in two directions:

1. Take a translation algorithm from *LTL* to *LTL+P* and try to bound the number of next operators in the translated *LTL* formula.

2. Take into consideration the $LTL + P$ semantics from the beginning (forgetting about translation algorithms) and generalize the theorem, maybe considering the number of other operators like the yesterday operator.

As far as I know, such an investigation has never been taken in the literature.

5 Bibliography

[BRV01] P. Blackburn, M. d. Rijke, and Y. Venema. *Modal Logic*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2001. DOI: 10.1017/CBO9781107050884.

[Del11] D. Della Monica. "Expressiveness, decidability, and undecidability of Interval Temporal Logic". PhD thesis. University of Udine, 2011.

[DG08] V. Diekert and P. Gastin. "First-order definable languages". In: Jan. 2008, pp. 261–306.

[DMS12] D. Della Monica, A. Montanari, and P. Sala. "The Importance of the Past in Interval Temporal Logics: The Case of Propositional Neighborhood Logic". In: *Logic Programs, Norms and Action: Essays in Honor of Marek J. Sergot on the Occasion of His 60th Birthday*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 79–102. ISBN: 978-3-642-29414-3.

[Kam68] H. Kamp. "Tense Logic and the Theory of Linear Order". PhD thesis. Ucla, 1968.

[KPV01] O. Kupferman, N. Piterman, and M. Y. Vardi. "Extended Temporal Logic Revisited". In: *CONCUR 2001 — Concurrency Theory*. Ed. by K. G. Larsen and M. Nielsen. Springer Berlin Heidelberg, 2001, pp. 519–535. ISBN: 978-3-540-44685-9.

[Stu00] H. Sturm. "Elementary Classes in Basic Modal Logic". In: *Studia Logica: An International Journal for Symbolic Logic* 64.2 (2000), pp. 193–213. (Visited on 06/04/2023).

[Wol83] P. Wolper. "Temporal logic can be more expressive". In: *Information and Control* 56.1 (1983), pp. 72–99. ISSN: 0019-9958.