








# A first-order characterisation of safety and co-safety languages

Alessandro Cimatti<sup>1</sup> , Luca Geatti<sup>3</sup> , Nicola Gigante<sup>3</sup> ,  
Angelo Montanari<sup>2</sup> , and Stefano Tonetta<sup>1</sup> 

<sup>1</sup> Fondazione Bruno Kessler, Trento, Italy

{cimatti,tonettas}@fbk.eu

<sup>2</sup> University of Udine, Italy

angelo.montanari@uniud.it

<sup>3</sup> Free University of Bozen-Bolzano, Italy

{geatti,gigante}@inf.unibz.it

**Abstract.** *Linear Temporal Logic* (LTL) is one of the most popular temporal logics, that comes into play in a variety of branches of computer science. Its widespread use is also due to its strong foundational properties. One of them is Kamp’s theorem, showing that LTL and the *first-order theory of one successor* (S1S[FO]) are expressively equivalent. Safety and co-safety languages, where a finite prefix suffices to establish whether a word does not or does belong to the language, respectively, play a crucial role in lowering the complexity of problems like model checking and reactive synthesis for LTL. Safety-LTL (resp., coSafety-LTL) is a fragment of LTL where only universal (resp., existential) temporal modalities are allowed, that recognises safety (resp., co-safety) languages only. In this paper, we introduce a fragment of S1S[FO], called Safety-FO, and its dual coSafety-FO, which are *expressively complete* with regards to the LTL-definable safety languages. In particular, we prove that they respectively characterise exactly Safety-LTL and coSafety-LTL, a result that joins Kamp’s theorem, and provides a clearer view of the characterisations of (fragments of) LTL in terms of first-order languages. In addition, it gives a direct, compact, and self-contained proof that any safety language definable in LTL is definable in Safety-LTL as well. As a by-product, we obtain some interesting results on the expressive power of the *weak tomorrow* operator of Safety-LTL interpreted over finite and infinite traces.

## 1 Introduction

*Linear Temporal Logic* (LTL) is the de-facto standard logic for system specifications [14]. It is a modal logic that is usually interpreted over infinite state sequences, but the finite-trace semantics has recently gained attention as well [6, 7]. The widespread use of LTL is due to its simple syntax and semantics, and to its strong foundational properties. Among them, we would like to mention the seminal work by Kamp [10] and Gabbay *et al.* [8], on its expressive completeness, i.e., LTL-definable languages are exactly those definable in the first-order fragment of the monadic second-order theory of one successor [3] (S1S[FO] for short).

In formal verification, an important class of specifications is that of *safety languages*. They are languages of infinite words where a finite prefix suffices to tell whether a word does not belong to the language. As an example, the set of all and only those infinite sequences where some particular bad event never happens can be regarded as a safety language. In their duals, *co-safety languages* (sometimes called *guarantee languages*), a finite prefix is sufficient to tell whether a word *belongs* to the language, e.g., when some desired event is mandated to eventually happen. Safety and co-safety languages are important for verification, model-checking, monitoring, and automated synthesis because they capture a variety of real-world requirements while being much simpler to deal with algorithmically [1, 11, 20].

Safety-LTL is the fragment of LTL where only *universal* temporal modalities are allowed. Similarly, its dual coSafety-LTL is obtained by only allowing *existential* modalities. It has been proved by Chang *et al.* [5] that Safety-LTL and coSafety-LTL define exactly the safety and co-safety languages that are definable in LTL, respectively.

In this paper, we provide a novel characterization of LTL-definable safety languages, and of their duals, in terms of a fragment of S1S[FO], called Safety-FO, and its dual coSafety-FO. The presented fragments have a very natural syntax, and we prove they are *expressively complete* with regards to LTL-definable safety and co-safety languages. We prove the correspondence between coSafety-FO and coSafety-LTL, which extends naturally to their duals and can be considered as a version of Kamp’s theorem [10] specialized for safety and co-safety properties, helping to create a clearer picture of the correspondence between (fragments of) temporal and first-order logics. We exploit such a result to prove the correspondence between co-safety languages definable in LTL and coSafety-FO, thus establishing also the equivalence between the former and coSafety-LTL. This provides a proof of the fact that Safety-LTL captures exactly the set of LTL-definable safety languages [5], which can be regarded as another contribution of the paper. The interest of our proof is twofold: on the one hand, the original proof by Chang *et al.* [5] is only sketched and it relies on two non-trivial translations scattered across different sources [16, 21]; on the other hand, such an equivalence result seems not to be very much known, as some authors presented the problem as open as lately as 2017 [20].<sup>4</sup> Thus, a compact and self-contained proof of the result seems to be a useful contribution for the community. It is worth to note that both proofs build on the fact that safety/co-safety languages can be captured by formulas of the form  $G\alpha/F\alpha$  with  $\alpha$  pure-past, but after that, the two proofs significantly diverge. Finally, as a by-product of this proof, we provide some results that assess the expressive power of the *weak tomorrow* operator of Safety-LTL when interpreted over finite *vs.* infinite traces.

The paper is organized as follows. After recalling necessary background knowledge in Section 2, Section 3 introduces Safety-FO and coSafety-FO and proves their correspondence with Safety-LTL and coSafety-LTL. Then, Section 4 proves

---

<sup>4</sup> As a matter of fact, we discovered about Chang *et al.* [5] after setting up the proof shown in this paper.

their correspondence with the set of safety and co-safety languages definable in LTL, thus providing a compact and self-contained proof of the equivalence between Safety-LTL and LTL-definable safety languages. Some properties of the *weak next* operator are outlined as well. Finally, Section 5 concludes the paper with some final considerations and a discussion of future work.

## 2 Preliminaries

Let  $A$  be a finite alphabet. We denote as  $A^*$  and  $A^\omega$  the set of all finite and infinite words, respectively, over  $A$ . We let  $A^+ = A^* \setminus \{\varepsilon\}$ , where  $\varepsilon$  is the empty word. Given a word  $\sigma \in A^*$  we denote as  $|\sigma|$  the length of  $\sigma$ . For an infinite word  $\sigma \in A^\omega$ ,  $|\sigma| = \omega$ . For a (finite or infinite) word  $\sigma$ , we denote as  $\sigma_i \in A$ , for  $0 \leq i < |\sigma|$ , the letter at the  $i$ -th position of the word. With  $\sigma_{[i,j]}$ , for  $0 \leq i \leq j < |\sigma|$ , we denote the subword that goes from the  $i$ -th to the  $j$ -th letter of the word, extrema included. With  $\sigma_{[i,\infty]}$  we denote the suffix of  $\sigma$  starting from the  $i$ -th letter. Given a word  $\sigma \in A^*$  and  $\sigma' \in A^* \cup A^\omega$ , we denote the *concatenation* of the two words as  $\sigma \cdot \sigma'$ , or simply  $\sigma\sigma'$ . A *language*  $\mathcal{L}$ , either  $\mathcal{L} \subseteq A^*$  or  $\mathcal{L} \subseteq A^\omega$ , is a set of words. Given two languages  $\mathcal{L}$  and  $\mathcal{L}'$  with  $\mathcal{L} \subseteq A^*$  and either  $\mathcal{L}' \subseteq A^*$  or  $\mathcal{L}' \subseteq A^\omega$ , we define  $\mathcal{L} \cdot \mathcal{L}' = \{\sigma \cdot \sigma' \mid \sigma \in \mathcal{L} \text{ and } \sigma' \in \mathcal{L}'\}$ . For a finite word  $\sigma = \sigma_0 \dots \sigma_k$  let  $\sigma^r = \sigma_k \dots \sigma_0$  be the reverse of  $\sigma$ , and for a language of finite words  $\mathcal{L}$  let  $\mathcal{L}^r = \{\sigma^r \mid \sigma \in \mathcal{L}\}$ . We can now define *safety* and *co-safety* languages.

**Definition 1 (Safety language [11, 19]).** *Let  $\mathcal{L} \subseteq A^\omega$ . We say that  $\mathcal{L}$  is a safety language if and only if for all the words  $\sigma \in A^\omega$  it holds that, if  $\sigma \notin \mathcal{L}$ , then there exists an  $i \in \mathbb{N}$  such that, for all  $\sigma' \in A^\omega$ ,  $\sigma_{[0,i]} \cdot \sigma' \notin \mathcal{L}$ . The class of safety languages is denoted as SAFETY.*

**Definition 2 (Co-safety language [11, 19]).** *Let  $\mathcal{L} \subseteq A^\omega$ . We say that  $\mathcal{L}$  is a co-safety language if and only if for all the words  $\sigma \in A^\omega$  it holds that, if  $\sigma \in \mathcal{L}$ , then there exists an  $i \in \mathbb{N}$  such that, for all  $\sigma' \in A^\omega$ ,  $\sigma_{[0,i]} \cdot \sigma' \in \mathcal{L}$ . The class of co-safety languages is denoted as coSAFETY.*

*Linear Temporal Logic with Past (LTL+P)* is a modal logic interpreted over infinite or finite words. Given a set  $\Sigma$  of proposition variables, the syntax of an LTL formula  $\phi$  is generated by the following grammar:

$$\begin{array}{ll}
 \phi := p \mid \neg\phi_1 \mid \phi_1 \vee \phi_2 \mid \phi_1 \wedge \phi_2 & \text{Boolean connectives} \\
 \mid X\phi_1 \mid \tilde{X}\phi_1 \mid \phi_1 \mathcal{U} \phi_2 \mid \phi_1 \mathcal{R} \phi_2 & \text{future modalities} \\
 \mid Y\phi_1 \mid Z\phi_1 \mid \phi_1 \mathcal{S} \phi_2 \mid \phi_1 \mathcal{T} \phi_2 & \text{past modalities}
 \end{array}$$

where  $\phi_1$  and  $\phi_2$  are LTL+P formulas and  $p \in \Sigma$ . An LTL+P formula is a *pure future* formula if it does not make use of past modalities, and it is *pure past* if it does not make use of future modalities. We denote with LTL the set of pure future formulas, and with LTL<sub>P</sub> the set of pure past formulas. Most of the temporal operators of the language can be defined in terms of a small number

of basic ones. In particular, conjunction can be defined in terms of disjunction ( $\phi_1 \wedge \phi_2 \equiv \neg(\neg\phi_1 \vee \neg\phi_2)$ ), the *release* operator can be defined in terms of the *until* operator ( $\phi_1 \mathcal{R} \phi_2 \equiv \neg(\neg\phi_1 \mathcal{U} \neg\phi_2)$ ), and the *triggered* operator can be defined in terms of the *since* operator ( $\phi_1 \mathcal{T} \phi_2 \equiv \neg(\neg\phi_1 \mathcal{S} \neg\phi_2)$ ). Nevertheless, we consider all these connectives and operators as primitive in order to be able to put any formula in *negated normal form* (NNF), *i.e.*, a form where negations are only applied to proposition letters. Note that the syntax includes both a *tomorrow* ( $\mathbf{X}\phi$ ) and *weak tomorrow* ( $\tilde{\mathbf{X}}\phi$ ) operators, as well as a *yesterday* ( $\mathbf{Y}\phi$ ) and *weak yesterday* ( $\tilde{\mathbf{Y}}\phi$ ) operators, for the same reason. Moreover, standard shortcut operators are available such as the *eventually* ( $\mathbf{F}\phi \equiv \top \mathcal{U} \phi$ ), and *always* ( $\mathbf{G}\phi \equiv \neg \mathbf{F} \neg \phi$ ) future operators, and the *once* ( $\mathbf{O}\phi \equiv \top \mathcal{S} \phi$ ), and *historically* ( $\mathbf{H}\phi \equiv \neg \mathbf{O} \neg \phi$ ) past operators.

LTL+P is interpreted over *state sequences*, which are finite or infinite words over  $2^\Sigma$ . Given a state sequence  $\sigma \in (2^\Sigma)^+$  or  $\sigma \in (2^\Sigma)^\omega$ , the *satisfaction* of a formula  $\phi$  by  $\sigma$  at a time point  $i \geq 0$ , denoted as  $\sigma, i \models \phi$ , is defined as follows:

1.  $\sigma, i \models p$       iff  $p \in \sigma_i$ ;
2.  $\sigma, i \models \neg\phi$     iff  $\sigma, i \not\models \phi$ ;
3.  $\sigma, i \models \phi_1 \vee \phi_2$  iff  $\sigma, i \models \phi_1$  or  $\sigma, i \models \phi_2$ ;
4.  $\sigma, i \models \phi_1 \wedge \phi_2$  iff  $\sigma, i \models \phi_1$  and  $\sigma, i \models \phi_2$ ;
5.  $\sigma, i \models \mathbf{X}\phi$       iff  $i + 1 < |\sigma|$  and  $\sigma, i + 1 \models \phi$ ;
6.  $\sigma, i \models \tilde{\mathbf{X}}\phi$       iff either  $i + 1 = |\sigma|$  or  $\sigma, i + 1 \models \phi$ ;
7.  $\sigma, i \models \mathbf{Y}\phi$       iff  $i > 0$  and  $\sigma, i - 1 \models \phi$ ;
8.  $\sigma, i \models \tilde{\mathbf{Y}}\phi$       iff either  $i = 0$  or  $\sigma, i - 1 \models \phi$ ;
9.  $\sigma, i \models \phi_1 \mathcal{U} \phi_2$  iff there exists  $i \leq j < |\sigma|$  such that  $\sigma, j \models \phi_2$ ,  
and  $\sigma, k \models \phi_1$  for all  $k$ , with  $i \leq k < j$ ;
10.  $\sigma, i \models \phi_1 \mathcal{S} \phi_2$  iff there exists  $j \leq i$  such that  $\sigma, j \models \phi_2$ ,  
and  $\sigma, k \models \phi_1$  for all  $k$ , with  $j < k \leq i$ ;
11.  $\sigma, i \models \phi_1 \mathcal{R} \phi_2$  iff either  $\sigma, j \models \phi_2$  for all  $i \leq j < |\sigma|$ , or there exists  
 $k \geq i$  such that  $\sigma, k \models \phi_1$  and  
 $\sigma, j \models \phi_2$  for all  $i \leq j \leq k$ ;
12.  $\sigma, i \models \phi_1 \mathcal{T} \phi_2$  iff either  $\sigma, j \models \phi_2$  for all  $0 \leq j \leq i$ , or there exists  
 $k \leq i$  such that  $\sigma, k \models \phi_1$  and  
 $\sigma, j \models \phi_2$  for all  $i \geq j \geq k$

We say that a state sequence  $\sigma$  satisfies  $\phi$ , written  $\sigma \models \phi$ , if  $\sigma, 0 \models \phi$ . Note that, when interpreted over an infinite word, the *tomorrow* and *weak tomorrow* operators have the same semantics. The *language* of  $\phi$ , denoted as  $\mathcal{L}(\phi)$ , is the set of words  $\sigma \in (2^\Sigma)^\omega$  such that  $\sigma \models \phi$ . The *language of finite words* of  $\phi$ , denoted as  $\mathcal{L}^{<\omega}(\phi)$ , is the set of finite words  $\sigma \in (2^\Sigma)^+$  such that  $\sigma \models \phi$ . Given a logic  $\mathbf{L}$  (*e.g.*, LTL), we denote as  $\llbracket \mathbf{L} \rrbracket$  the set of languages  $\mathcal{L}$  such that there is a formula  $\phi \in \mathbf{L}$  such that  $\mathcal{L} = \mathcal{L}(\phi)$ , and we denote as  $\llbracket \mathbf{L} \rrbracket^{<\omega}$  the set of languages of finite words  $\mathcal{L}$  such that there is a formula  $\phi \in \mathbf{L}$  such that  $\mathcal{L} = \mathcal{L}^{<\omega}(\phi)$ . Note that  $\llbracket \text{LTL} \rrbracket^{<\omega}$  is usually called LTLf in the literature [6].

We now define the two fragments of LTL that are the subject of this paper.

**Definition 3 (Safety-LTL and coSafety-LTL [17]).** *The logic Safety-LTL (resp. coSafety-LTL) is the fragment of LTL where, for formulas in negated normal*

form, *only the* tomorrow, weak tomorrow *and* release (*resp.* until) *temporal operators are allowed.*

We also define the logic  $\text{coSafety-LTL}(-\tilde{X})$  as the logic  $\text{coSafety-LTL}$  devoid of the *weak tomorrow* operator (this logic will play a central role in our proofs).

In the next Section we present two fragments of the *first-order theory of one successor* [2, 3], namely  $\text{S1S[FO]}$ , or simply  $\text{FO}$  in the following. Fixed an alphabet  $\Sigma$ ,  $\text{FO}$  is a first-order language with equality over the signature  $\langle \langle, \{P\}_{p \in \Sigma} \rangle \rangle$ , and is interpreted over structures  $\mathcal{M} = \langle D^{\mathcal{M}}, <^{\mathcal{M}}, \{P^{\mathcal{M}}\}_{p \in \Sigma} \rangle$  where  $D^{\mathcal{M}}$ , for our goals, is either the set  $\mathbb{N}$  of natural numbers or a prefix  $\{0, \dots, n\}$  thereof, and  $<^{\mathcal{M}}$  is the usual ordering relation between natural numbers. Given an  $\text{FO}$  formula  $\phi(x_0, \dots, x_m)$  with  $m + 1$  free variables, the satisfaction of  $\phi$  by a first-order structure  $\mathcal{M}$  when  $x_0 = n_0, \dots, x_m = n_m$ , denoted as  $\mathcal{M}, n_0, \dots, n_m \models \phi(x_0, \dots, x_m)$ , is defined following the standard first-order semantics. State sequences over  $\Sigma$  map naturally into such structures. Given a word  $\sigma \in (2^\Sigma)^*$  or  $\sigma \in (2^\Sigma)^\omega$ , we denote as  $(\sigma)^s$  the corresponding first-order structure. Given a formula  $\phi(x)$  with exactly one free variable, the *language* of  $\phi$ , denoted as  $\mathcal{L}(\phi)$ , is the set of words  $\sigma \in (2^\Sigma)^\omega$  such that  $(\sigma)^s, 0 \models \phi$ . Similarly, the *language of finite words* of  $\phi$ , denoted as  $\mathcal{L}^{<\omega}(\phi)$ , is the set of finite words  $\sigma \in (2^\Sigma)^+$  such that  $(\sigma)^s \models \phi$ . We denote as  $\llbracket \text{FO} \rrbracket$  and  $\llbracket \text{FO} \rrbracket^{<\omega}$  the set of languages of infinite and finite words, respectively, definable by a  $\text{FO}$  formula.

Given a class of languages of finite words  $\llbracket \text{L} \rrbracket^{<\omega}$ , we denote as  $\llbracket \text{L} \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega$  the set of languages  $\llbracket \text{L} \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega = \{\mathcal{L} \cdot (2^\Sigma)^\omega \mid \mathcal{L} \in \llbracket \text{L} \rrbracket^{<\omega}\}$ . We recall now some known results.

**Proposition 1 (Kamp [10] and Gabbay [8]).**

$$\llbracket \text{LTL} \rrbracket = \llbracket \text{FO} \rrbracket \text{ and } \llbracket \text{LTL} \rrbracket^{<\omega} = \llbracket \text{FO} \rrbracket^{<\omega}.$$

Finally, we state a normal form for  $\text{LTL}$ -definable safety/co-safety languages.

**Proposition 2 (Chang *et al.* [5], Thomas [19]).** *A language  $\mathcal{L} \in \llbracket \text{LTL} \rrbracket$  is safety (*resp.* co-safety) if and only if it is the language of a formula of the form  $G\alpha$  (*resp.*  $F\alpha$ ), where  $\alpha \in \text{LTL}_P$ .*

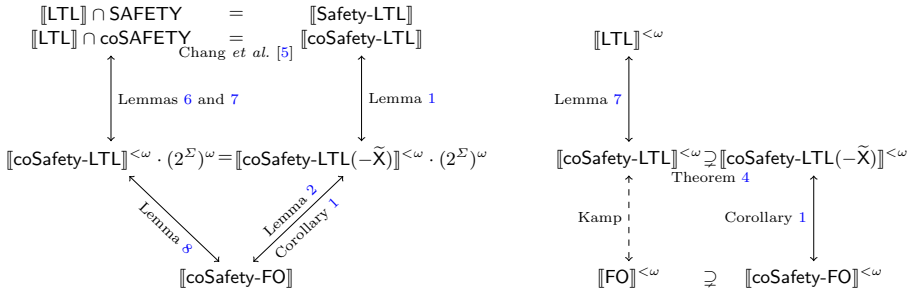
### 3 Safety-FO and coSafety-FO

In this section we introduce the core contribution of the paper, *i.e.*, two fragments of  $\text{FO}$  that precisely capture  $\text{Safety-LTL}$  and  $\text{coSafety-LTL}$ , respectively, and we prove this relationship. A summary of the results provided by the paper is given in Fig. 1.

**Definition 4 (Safety-FO).** *The logic Safety-FO is generated by the following grammar:*

$$\begin{aligned} \text{atomic} &:= x < y \mid x = y \mid x \neq y \mid P(x) \mid \neg P(x) \\ \phi &:= \text{atomic} \mid \phi_1 \vee \phi_2 \mid \phi_1 \wedge \phi_2 \mid \exists y(x < y < z \wedge \phi_1) \mid \forall y(x < y \rightarrow \phi_1) \end{aligned}$$

where  $x, y$ , and  $z$  are first-order variables,  $P$  is a unary predicate, and  $\phi_1$  and  $\phi_2$  are  $\text{Safety-FO}$  formulas.



**Fig. 1.** Summary of the results of the paper, about languages over infinite words on the left, and over finite words on the right. Solid arrows are own results. Dashed arrows are known from literature.

**Definition 5 (coSafety-FO).** *The logic coSafety-FO is generated by the following grammar:*

$$\begin{aligned} \text{atomic} &:= x < y \mid x = y \mid x \neq y \mid P(x) \mid \neg P(x) \\ \phi &:= \text{atomic} \mid \phi_1 \vee \phi_2 \mid \phi_1 \wedge \phi_2 \mid \exists y(x < y \wedge \phi_1) \mid \forall y(x < y < z \rightarrow \phi_1) \end{aligned}$$

where  $x, y,$  and  $z$  are first-order variables,  $P$  is a unary predicate, and  $\phi_1$  and  $\phi_2$  are coSafety-FO formulas.

We need to make a few observations on the syntax of the two fragments. First of all, note how any formula of Safety-FO is the negation of a formula of coSafety-FO and *vice versa*. Then, note that the two fragments are defined in *negated normal form*, i.e., negation only appears on atomic formulas. The particular kind of existential and universal quantifications allowed are the culprit of these fragments. In particular Safety-FO restricts any existentially quantified variable to be bounded between two already quantified variables. The same applies to universal quantification in coSafety-FO. Moreover Safety-FO and coSafety-FO formulas are *future formulas*, i.e., the quantifiers can only range over values *greater* than already quantified variables. These two features are essential to precisely capture Safety-LTL and coSafety-LTL. Finally, note that the comparisons in the guards of the quantifiers are strict, but non-strict comparisons can be used as well. In particular,  $\exists y(x \leq y \wedge \phi)$  can be rewritten as  $\phi[y/x] \vee \exists y(x < y \wedge \phi)$ , where  $\phi[y/x]$  is the formula obtained by replacing all occurrences of  $y$  with  $x$ . Similarly,  $\forall z(x \leq z \leq y \rightarrow \phi)$  can be rewritten as  $\phi[z/x] \wedge \phi[z/y] \wedge \forall z(x < z < y \rightarrow \phi)$ .

To prove the relationship between Safety-LTL, coSafety-LTL, and these fragments, we focus now on coSafety-FO. By duality, all the results transfer to Safety-FO. We focus on coSafety-FO because the unbounded quantification is existential, and it is easier to reason about the existence of prefixes than on all the prefixes at once. We start by observing that, since the *weak tomorrow* operator, over infinite words, coincides with the *tomorrow* operator, the following holds.

**Observation 1.**  $\llbracket \text{coSafety-LTL} \rrbracket = \llbracket \text{coSafety-LTL}(-\tilde{X}) \rrbracket$

When reasoning over finite words, the *weak tomorrow* operator plays a crucial role, since it can be used to recognize when we are at the last position of a word. In fact, the formula  $\sigma, i \models \tilde{X}\perp$  is true if and only if  $i = |\sigma| - 1$ , for any  $\sigma \in (2^\Sigma)^*$ .

Now, let us note that, thanks to the absence of the *weak tomorrow* operator, we can in some sense reduce ourselves to reasoning over finite words.

**Lemma 1.**  $\llbracket \text{coSafety-LTL}(-\tilde{X}) \rrbracket = \llbracket \text{coSafety-LTL}(-\tilde{X}) \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega$

*Proof.* We have to prove that, for each formula  $\phi \in \text{coSafety-LTL}(-\tilde{X})$ , it holds that:

$$\mathcal{L}(\phi) = \mathcal{L}^{<\omega}(\phi) \cdot (2^\Sigma)^\omega$$

We proceed by induction on the structure of  $\phi$ . For the base case, consider  $\phi \equiv p \in \Sigma$ . The case for  $\phi \equiv \neg p$  is similar. Let  $\sigma \in \mathcal{L}(p)$ . It holds that  $\sigma_0 \models p$  and  $\sigma_0 \cdot \sigma' \models p$ , for all  $\sigma' \models (2^\Sigma)^\omega$ , and in particular for  $\sigma' = \sigma_{[1,\infty)}$ . This is equivalent to say that  $\sigma \in \mathcal{L}^{<\omega}(\phi) \cdot (2^\Sigma)^\omega$ . For the inductive step:

1. Let  $\phi \equiv \phi_1 \wedge \phi_2$ . Suppose that  $\sigma \in \mathcal{L}(\phi)$ . Obviously,  $\sigma \models \phi_1$  and  $\sigma \models \phi_2$ , and therefore  $\sigma \in \mathcal{L}(\phi_1)$  and  $\sigma \in \mathcal{L}(\phi_2)$ . By the inductive hypothesis,  $\sigma \in \mathcal{L}^{<\omega}(\phi_1) \cdot (2^\Sigma)^\omega$  and  $\sigma \in \mathcal{L}^{<\omega}(\phi_2) \cdot (2^\Sigma)^\omega$ . This means that there exist two indices  $i, j \in \mathbb{N}$  such that  $\sigma_{[0,i]} \models \phi_1$  and  $\sigma_{[0,j]} \models \phi_2$ . Let  $m$  be the greatest between  $i$  and  $j$ . It holds that  $\sigma_{[0,m]} \models \phi_1 \wedge \phi_2$ . Therefore  $\sigma \in \mathcal{L}^{<\omega}(\phi_1 \wedge \phi_2) \cdot (2^\Sigma)^\omega$ .
2. Let  $\phi \equiv \phi_1 \vee \phi_2$  and let  $\sigma \in \mathcal{L}(\phi)$ . We have that  $\sigma \models \phi_1$  or  $\sigma \models \phi_2$ . Without loss of generality, we consider the case that  $\sigma \models \phi_1$  (the other case is specular). By the inductive hypothesis,  $\sigma \in \mathcal{L}^{<\omega}(\phi_1) \cdot (2^\Sigma)^\omega$ . Therefore, it also holds that  $\sigma \in \mathcal{L}^{<\omega}(\phi_1 \vee \phi_2) \cdot (2^\Sigma)^\omega$ .
3. Let  $\phi \equiv X\phi_1$  and let  $\sigma \in \mathcal{L}(X\phi_1)$ . By the semantics of the *tomorrow* operator, it holds that  $\sigma_{[1,\infty)} \models \phi_1$ . By the inductive hypothesis,  $\sigma_{[1,\infty)} \in \mathcal{L}^{<\omega}(\phi_1) \cdot (2^\Sigma)^\omega$ . This means that there exists an index  $i \geq 1$  such that  $\sigma_{[1,i]} \models \phi_1$ . Therefore, it also holds that the state sequence  $\sigma_{[0,i]} = \sigma_0 \cdot \sigma_{[1,i]}$  satisfies  $X\phi_1$  over finite words, that is,  $\sigma_{[0,i]} \models X\phi_1$ . This means that  $\sigma \in \mathcal{L}^{<\omega}(X\phi_1) \cdot (2^\Sigma)^\omega$ .
4. Let  $\phi \equiv \phi_1 \mathcal{U} \phi_2$ . Let  $\sigma \in \mathcal{L}(\phi)$ . By the semantics of the *until* operator, it holds that there exists an index  $i \in \mathbb{N}$  such that  $\sigma_{[i,\infty)} \models \phi_2$  and  $\sigma_{[j,\infty)} \models \phi_1$  for all  $0 \leq j < i$ . By the inductive hypothesis, we have that  $\sigma_{[i,\infty)} \in \mathcal{L}^{<\omega}(\phi_2) \cdot (2^\Sigma)^\omega$  and  $\sigma_{[j,\infty)} \in \mathcal{L}^{<\omega}(\phi_1) \cdot (2^\Sigma)^\omega$  for all  $0 \leq j < i$ . This means that there exists an index  $i \in \mathbb{N}$  and  $i + 1$  indices  $k_0, \dots, k_i \in \mathbb{N}$  such that  $\sigma_{[i,k_i]} \models \phi_2$  and  $\sigma_{[j,k_j]} \models \phi_1$  for all  $0 \leq j < i$ . Let  $m$  be the greatest between  $k_0, \dots, k_i$ . It holds that there exists an index  $i \in \mathbb{N}$  such that  $\sigma_{[i,m]} \models \phi_2$  and  $\sigma_{[j,m]} \models \phi_1$  for all  $0 \leq j < i$ . Therefore,  $\sigma \in \mathcal{L}^{<\omega}(\phi_1 \mathcal{U} \phi_2) \cdot (2^\Sigma)^\omega$ .

The same property applies to **coSafety-FO** as well.

**Lemma 2.**  $\llbracket \text{coSafety-FO} \rrbracket = \llbracket \text{coSafety-FO} \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega$

*Proof.* We have to prove that, for each formula  $\psi \in \text{coSafety-FO}$  with one free variable, it holds that  $\mathcal{L}(\psi) = \mathcal{L}^{<\omega}(\psi) \cdot (2^\Sigma)^\omega$ . We proceed by induction,

but with a more general statement. Let  $\phi(x_1, \dots, x_k)$  have  $k$  free variables. We prove by induction on  $\phi$  that for any infinite state sequence  $\sigma$  such that  $(\sigma)^s, n_1, \dots, n_k \models \phi(x_1, \dots, x_k)$ , there exists a prefix  $\sigma_{[0,i]}$  of  $\sigma$  such that for all  $\sigma' \in (2^\Sigma)^\omega$ ,  $(\sigma_{[0,i]}\sigma')^s, n_1, \dots, n_k \models \phi(x_1, \dots, x_k)$ . The base case considers the four kinds of atomic formulas. If  $(\sigma)^s, n_1, n_2 \models x_1 < x_2$ , then  $n_1 < n_2$  and we know that  $(\sigma_{[0,n_2]}\sigma')^s, n_1, n_2 \models x_1 < x_2$  for all  $\sigma' \in (2^\Sigma)^*$ . The case of  $x_1 = x_2$  is similar. Now, if  $(\sigma)^s, n_1 \models P(x_1)$ , then  $p \in \sigma_{n_1}$  and we know that  $(\sigma_{[0,n_1]}\sigma')^s, n_1 \models P(x_1)$  for all  $\sigma' \in (2^\Sigma)^*$ . The case for  $\neg P(x_1)$  is similar. For the inductive step:

1. if  $(\sigma)^s, n_1, \dots, n_k \models \phi_1(x_1, \dots, x_k) \wedge \phi_2(x_1, \dots, x_k)$ , by the induction hypothesis we know that there are two prefixes  $\sigma_{[0,i]}$  and  $\sigma_{[0,j]}$  such that, respectively,  $(\sigma_{[0,i]}\sigma')^s, n_1, \dots, n_k \models \phi_1(x_1, \dots, x_k)$  and  $(\sigma_{[0,j]}\sigma'')^s, n_1, \dots, n_k \models \phi_2(x_1, \dots, x_k)$ , for all  $\sigma', \sigma'' \in (2^\Sigma)^*$ . Then, supposing *w.l.o.g.* that  $i \leq j$ , we know that  $(\sigma_{[0,j]}\sigma'')^s, n_1, \dots, n_k \models \phi_1(x_1, \dots, x_k) \wedge \phi_2(x_1, \dots, x_k)$ . The case for  $\phi_1(x_1, \dots, x_k) \vee \phi_2(x_1, \dots, x_k)$  is similar.
2. If  $(\sigma)^s, n_1, \dots, n_k \models \exists x_{k+1}(x_u < x_{k+1} \wedge \phi_1(x_1, \dots, x_{k+1}))$  for some  $1 \leq u \leq k$ , then there exists an  $n_{k+1} > n_u$  such that  $(\sigma)^s, n_1, \dots, n_{k+1} \models \phi_1(x_1, \dots, x_{k+1})$ . This implies that  $(\sigma_{[0,i]}\sigma')^s, n_1, \dots, n_{k+1} \models \phi_1(x_1, \dots, x_{k+1})$  for some  $i \geq 0$  and all  $\sigma' \in (2^\Sigma)^*$ , by the induction hypothesis. It follows that  $(\sigma_{[0,i]}\sigma')^s, n_1, \dots, n_k \models \exists x_{k+1}(x_i < x_{k+1} \wedge \phi_1(x_1, \dots, x_{k+1}))$ .
3. if  $(\sigma)^s, n_1, \dots, n_k \models \forall x_{k+1}(x_u < x_{k+1} < x_v \rightarrow \phi_1(x_1, \dots, x_{k+1}))$  for some  $1 \leq u, v \leq k$ , then for all  $n_{k+1}$  with  $n_u < n_{k+1} < n_v$  it holds that  $(\sigma)^s, n_1, \dots, n_{k+1} \models \phi_1(x_1, \dots, x_{k+1})$ . Then, for the induction hypothesis, for all  $n_{k+1}$  with  $n_u < n_{k+1} < n_v$  there is a prefix  $\sigma_{[0,i_{n_{k+1}}]}$  such that  $(\sigma_{[0,i_{n_{k+1}}]}\sigma')^s, n_1, \dots, n_{k+1} \models \phi_1(x_1, \dots, x_{k+1})$  for all  $\sigma' \in (2^\Sigma)^*$ . Then, if  $n_* = \max_{n_u < n_{k+1} < n_v} (i_{n_{k+1}})$ , it holds that:

$$(\sigma_{[0,n_*]}\sigma')^s, n_1, \dots, n_k \models \forall x_{k+1}(x_u < x_{k+1} < x_v \rightarrow \phi_1(x_1, \dots, x_{k+1}))$$

Now, let  $\psi(x)$  be a **coSafety-FO** formula with exactly one free variable  $x$ . Thanks to the above induction we can conclude that each infinite state sequence  $\sigma$  such that  $(\sigma)^s, 0 \models \phi(x)$  is of the form  $\sigma_{[0,i]} \cdot \sigma'$ , where  $(\sigma_{[0,i]})^s \models \phi(x)$ , and this implies that  $\mathcal{L}(\psi) = \mathcal{L}^{<\omega}(\psi) \cdot (2^\Sigma)^\omega$ .

It is worth to note that Lemmas 1 and 2 show that **coSafety-LTL**( $-\tilde{X}$ ) and **coSafety-FO** are *insensitive to infiniteness* as defined by De Giacomo *et al.* [9].

Then, we can focus on **coSafety-LTL**( $-\tilde{X}$ ) and **coSafety-FO** on finite words. If we can prove that  $\llbracket \text{coSafety-LTL}(-\tilde{X}) \rrbracket^{<\omega} = \llbracket \text{coSafety-FO} \rrbracket^{<\omega}$ , we are done. At first, we show how to encode **coSafety-LTL**( $-\tilde{X}$ ) formulas into **coSafety-FO**.

**Lemma 3.**  $\llbracket \text{coSafety-LTL}(-\tilde{X}) \rrbracket^{<\omega} \subseteq \llbracket \text{coSafety-FO} \rrbracket^{<\omega}$

*Proof.* Let  $\mathcal{L} \in \llbracket \text{coSafety-LTL}(-\tilde{X}) \rrbracket^{<\omega}$ , and let  $\phi \in \text{coSafety-LTL}(-\tilde{X})$  such that  $\mathcal{L} = \mathcal{L}^{<\omega}(\phi)$ . By following the semantics of the operators in  $\phi$ , we can obtain an equivalent **coSafety-FO** formula  $\phi_{\text{FO}}$ . We inductively define the formula  $FO(\phi, x)$ , where  $x$  is a variable, as follows:



- $FO(p, x) = P(x)$ , for each  $p \in \Sigma$
- $FO(\neg p, x) = \neg P(x)$ , for each  $p \in \Sigma$
- $FO(\phi_1 \wedge \phi_2, x) = FO(\phi_1, x) \wedge FO(\phi_2, x)$
- $FO(\phi_1 \vee \phi_2, x) = FO(\phi_1, x) \vee FO(\phi_2, x)$
- $FO(\mathbf{X}\phi_1, x) = \exists y(x < y \wedge y = x + 1 \wedge FO(\phi_1, y))$   
 where  $y = x + 1$  can be expressed as  $\forall z(x < z < y \rightarrow \perp)$ .
- $FO(\phi_1 \mathbf{U} \phi_2, x) = \exists y(x \leq y \wedge FO(\phi_2, y) \wedge \forall z(x \leq z < y \rightarrow FO(\phi_1, z)))$

For each  $\phi \in \text{coSafety-LTL}(-\tilde{\mathbf{X}})$ , the formula  $FO(\phi, x)$  has exactly one free variable  $x$ . It is easy to see that for all finite state sequences  $\sigma \in (2^\Sigma)^*$ , it holds that  $\sigma \models \phi$  if and only if  $(\sigma)^s, 0 \models FO(\phi, x)$ , and  $FO(\phi, x) \in \text{coSafety-FO}$ . Therefore,  $\mathcal{L} \in \llbracket \text{coSafety-FO} \rrbracket^{<\omega}$ .

It is time to show the opposite direction, *i.e.*, that any  $\text{coSafety-FO}$  formula can be translated into a  $\text{coSafety-LTL}(-\tilde{\mathbf{X}})$  formula which is equivalent over finite words. To prove this fact we adapt a proof of Kamp’s theorem by Rabinovich [15]. Kamp’s theorem is one of the fundamental results about temporal logics, which states that LTL corresponds to FO in terms of expressiveness. Here, we prove a similar result in the context of co-safety languages. The proof goes by introducing a *normal form* for FO formulas, and showing that (i) any  $\text{coSafety-FO}$  formula can be translated into such normal form and (ii) any formula in normal form can be straightforwardly translated into a  $\text{coSafety-LTL}(-\tilde{\mathbf{X}})$  formula. We start by introducing such a normal form.

**Definition 6 ( $\exists\forall$ -formulas).** *An  $\exists\forall$ -formula  $\phi(z_0, \dots, z_m)$  with  $m$  free variables is a formula of this form:*

$$\begin{aligned}
 \phi(z_0, \dots, z_m) := & \exists x_0 \dots \exists x_n ( \\
 & x_0 < x_1 < \dots < x_n && \text{ordering constraints} \\
 & \wedge z_0 = x_0 \wedge \bigwedge_{k=1}^m (z_k = x_{i_k}) && \text{binding constraints} \\
 & \wedge \bigwedge_{j=0}^n \alpha_j(x_j) && \text{punctual constraints} \\
 & \wedge \bigwedge_{j=1}^n \forall y (x_{j-1} < y < x_j \rightarrow \beta_j(y)) && \text{interval constraints}
 \end{aligned}$$

where  $i_k \in \{0, \dots, n\}$  for each  $0 \leq k \leq m$ , and  $\alpha_j$  and  $\beta_j$ , for each  $1 \leq j \leq n$ , are quantifier-free formulas with exactly one free variable.

Some explanations are due. Each  $\exists\forall$ -formula states a number of requirements for its free variables and for its quantified variables. Through the binding constraints, the free variables are identified with a subset of the quantified variables in order to uniformly state the punctual and interval constraints, and the ordering constraints which sort all the variable in a total order. Note that there is no relationship between  $n$  and  $m$ : there might be more quantified variables

than free variables, or less. Note as well that the binding constraint  $z_0 = x_0$  is always present, *i.e.*, at least one free variable has to be the minimal element of the ordering. This ensures that  $\exists\forall$ -formulas are always *future* formulas.

We say that a formula of **coSafety-FO** is in *normal form* if and only if it is a disjunction of  $\exists\forall$ -formulas. To see how formulas in normal form make sense, let us immediately show how to translate them into **coSafety-LTL**( $-\tilde{X}$ ) formulas.

**Lemma 4.** *For any formula  $\phi(z) \in \text{coSafety-FO}$  in normal form, with a single free variable, there exists a formula  $\psi \in \text{coSafety-LTL}(-\tilde{X})$  such that  $\mathcal{L}^{<\omega}(\phi(z)) = \mathcal{L}^{<\omega}(\psi)$ .*

*Proof.* We show how any  $\exists\forall$ -formula is equivalent to an **coSafety-LTL**( $-\tilde{X}$ )-formula, over finite words. Since each formula in normal form is a disjunction of  $\exists\forall$ -formulas, and since **coSafety-LTL**( $-\tilde{X}$ ) is closed under disjunction, this implies the proposition. Let  $\phi(z)$  be a  $\exists\forall$ -formula with a single free variable. Having only one free variable,  $\phi(z)$  is of the form:

$$\begin{aligned} &\exists x_0 \dots \exists x_n (x_0 < \dots < x_n \wedge z = x_0 \\ &\quad \wedge \bigwedge_{j=0}^n \alpha_j(x_j) \wedge \bigwedge_{j=1}^n \forall y (x_{j-1} < y < x_j \rightarrow \beta_j(y))) \end{aligned}$$

Now, let  $A_i$  be the temporal formulas corresponding to  $\alpha_i$  and  $B_i$  be the ones corresponding to  $\beta_i$ . Recall that  $\alpha_i$  and  $\beta_i$  are quantifier free with only one free variable, hence this correspondence is trivial. Since  $z$  is the first time point of the ordering mandated by the formula, we only need future temporal operators to encode  $\phi$  into a **coSafety-LTL**( $-\tilde{X}$ ) formula  $\psi$  defined as follows:

$$\psi \equiv A_0 \wedge X(B_0 U (A_1 \wedge X(B_1 U A_2 \wedge \dots X(B_{n-1} U A_n) \dots)))$$

It can be seen that  $\sigma, k \models \psi$  if and only if  $(\sigma)^s, k \models \phi(z)$ , for each  $\sigma \in (2^\Sigma)^+$  and each  $k \geq 0$ . Thus,  $\mathcal{L}^{<\omega}(\phi(z)) = \mathcal{L}^{<\omega}(\psi)$ .

Two differences between our  $\exists\forall$ -formulas and those used by Rabinovich [15] are crucial: first, we do not have unbounded universal requirements, but all interval constraints use bounded quantifications, hence we do not need the *always* operator to encode them; second, our  $\exists\forall$ -formulas are *future* formulas, hence we only need future operators to encode them.

We now show that any **coSafety-FO** formula can be translated into normal form, that is, into a *disjunction* of  $\exists\forall$ -formulas.

**Lemma 5.** *Any **coSafety-FO** formula is equivalent to a disjunction of  $\exists\forall$ -formulas.*

*Proof.* Let  $\phi$  be a **coSafety-FO** formula. We proceed by structural induction on  $\phi$ . For the base case, for each atomic formula  $\phi(z_0, z_1)$  we provide an equivalent  $\exists\forall$ -formula  $\psi(z_0, z_1)$ :

1. if  $\phi \equiv z_0 < z_1$  then  $\psi \equiv \exists x_0 \exists x_1 (z_0 = x_0 \wedge z_1 = x_1 \wedge x_0 < x_1)$ ;
2. if  $\phi \equiv z_0 = z_1$ , then  $\psi \equiv \exists x_0 (z_0 = x_0 \wedge z_1 = x_0)$ .

3. if  $\phi \equiv z_0 \neq z_1$ , we can note that  $\phi \equiv z_0 < z_1 \vee z_1 < z_0$  and then apply Item 1;
4. If  $\phi \equiv P(z_0)$  then we define  $\psi := \exists x_0(z_0 = x_0 \wedge P(x_0))$ . Similarly if  $\phi \equiv \neg P(z_0)$ .

For the inductive step:

1. The case of a disjunction is trivial.
2. If  $\phi(z_0, \dots, z_k)$  is a conjunction, by the inductive hypothesis each conjunct is equivalent to a disjunction of  $\exists\forall$ -formulas. By distributing the conjunction over the disjunction we can reduce ourselves to the case of a conjunction  $\psi_1(z_0, \dots, z_k) \wedge \psi_2(z_0, \dots, z_k)$  of two  $\exists\forall$ -formulas. In this case we have that:

$$\begin{aligned}\psi_1 &\equiv \exists x_0 \dots \exists x_n (x_0 < \dots < x_n \wedge z_0 = x_0 \wedge \dots) \\ \psi_2 &\equiv \exists x_{n+1} \dots \exists x_m (x_{n+1} < \dots < x_m \wedge z_0 = x_{n+1} \wedge \dots)\end{aligned}$$

Since the set of quantified variables in  $\psi_1$  is disjoint from the set of quantified variables in  $\psi_2$ , we can distribute the existential quantifiers over the conjunction  $\psi_1 \wedge \psi_2$ , obtaining:

$$\begin{aligned}\psi_1 \wedge \psi_2 &\equiv \exists x_0 \dots \exists x_n \exists x_{n+1} \dots \exists x_m \\ &\quad (x_0 < \dots < x_n \wedge x_{n+1} < \dots < x_m \wedge z_0 = x_0 \wedge z_0 = x_{n+1} \wedge \dots)\end{aligned}$$

Note that we can identify  $x_0$  and  $x_{n+1}$ , obtaining:

$$\begin{aligned}\psi_1 \wedge \psi_2 &\equiv \exists x_0 \dots \exists x_n \exists x_{n+2}, \dots \exists x_m \\ &\quad (x_0 < \dots < x_n \wedge x_0 < x_{n+2} < \dots < x_m \wedge \\ &\quad z_0 = x_0 \wedge \bigwedge_{i=1}^k (z_i = x_{j_i''}) \wedge \bigwedge_{i=0, i \neq n+1}^m \alpha_i(x_i) \wedge \\ &\quad \bigwedge_{\substack{i=1, i \neq n+1 \\ i \neq n+2}}^m \forall y (x_{i-1} < y < x_i \rightarrow \beta_i(y)) \wedge \forall y (x_0 < y < x_{n+2} \rightarrow \beta_{n+2}(y)))\end{aligned}$$

Now, to turn this formula into a disjunction of  $\exists\forall$ -formulas, we consider all the possible interleavings of the variables that respect the two imposed orderings and explode the formula into a disjunction that consider each such interleaving. Let  $X = \{x_0, \dots, x_n, x_{n+2}, \dots, x_m\}$  and let  $\Pi$  be the set of all the permutations of  $X$  compatible with the orderings  $x_0 < \dots < x_n$  and  $x_0 < x_{n+1} < \dots < x_m$ . For any  $\pi \in \Pi$ ,  $\pi(0) = 0$ . Now,  $\psi_1 \wedge \psi_2$  becomes the disjunction of a set of  $\exists\forall$ -formulas  $\psi_\pi$ , for each  $\pi \in \Pi$ , defined as:

$$\begin{aligned}\psi_\pi &\equiv \exists x_{\pi(0)} \dots \exists x_{\pi(m)} \\ &\quad (x_{\pi(0)} < \dots < x_{\pi(m)} \wedge \\ &\quad z_0 = x_0 \wedge \bigwedge_{i=1}^k (z_i = x_{\pi(j_i''')}) \wedge \bigwedge_{i=0}^m \alpha_i(x_i) \wedge \\ &\quad \bigwedge_{i=0}^m \forall y (x_{\pi(i-1)} < y < x_{\pi(i)} \rightarrow \beta_i^*(y)))\end{aligned}$$

where  $\beta_i^*$  suitably combines the formulas  $\beta$  according to the interleaving of the orderings of the original variables, and is defined as follows:

$$\beta_i^* = \begin{cases} \beta_{\pi(i)} & \text{if both } \pi(i), \pi(i-1) \leq n \text{ or both } \pi(i), \pi(i-1) > n \\ \beta_{\pi(i)} \wedge \beta_{\pi(i-1)} & \text{if } \pi(i) \leq n \text{ and } \pi(i-1) > n \text{ or vice versa} \end{cases}$$

Then we have that  $\psi_1 \wedge \psi_2 \equiv \bigvee_{\pi \in \Pi} (\psi_\pi)$ , which is a disjunction of  $\exists\forall$ -formulas.

3. Let  $\phi(z_0, \dots, z_m) \equiv \exists z_{m+1} \cdot (z_i < z_{m+1} \wedge \phi_1(z_0, \dots, z_m, z_{m+1}))$ , for some  $0 \leq i \leq m$ . By the inductive hypothesis, this is equivalent to the formula  $\exists z_{m+1} (z_i < z_{m+1} \wedge \bigvee_{k=0}^j \psi_k(z_0, \dots, z_m, z_{m+1}))$ , where  $\psi_k(z_0, \dots, z_m, z_{m+1})$  is a  $\exists\forall$ -formula, for each  $0 \leq k \leq j$ , that is:

$$\exists z_{m+1} \cdot (z_i < z_{m+1} \wedge \bigvee_{k=0}^j (\exists x_0 \dots \exists x_{n_k} \psi'_k(z_0, \dots, z_{m+1}, x_0, \dots, x_{n_k})))$$

By distributing the conjunction over the disjunction, we obtain:

$$\exists z_{m+1} \cdot \left( \bigvee_{k=0}^j ((z_i < z_{m+1}) \wedge \exists x_0 \dots \exists x_{n_k} \psi'_k(z_0, \dots, z_{m+1}, x_0, \dots, x_{n_k})) \right)$$

and by distributing the existential quantifier over the disjunction, we have:

$$\bigvee_{k=0}^j (\exists z_{m+1} ((z_i < z_{m+1}) \wedge \exists x_0 \dots \exists x_{n_k} \psi'_k(z_0, \dots, z_{m+1}, x_0, \dots, x_{n_k})))$$

Since the subformula  $z_i < z_{m+1}$  does not contain the variables  $x_0, \dots, x_n$ , we can push it inside the existential quantification, obtaining:

$$\bigvee_{k=0}^j (\exists z_{m+1} \cdot \exists x_0 \dots \exists x_{n_k} \cdot ((z_i < z_{m+1}) \wedge \psi'_k(z_0, \dots, z_{m+1}, x_0, \dots, x_{n_k})))$$

Now we divide in cases:

- (a) suppose that the formula  $\psi'_k(z_0, \dots, z_{m+1}, x_0, \dots, x_{n_k})$  contains the following conjuncts:  $z_i = x_{l_i}$  and  $z_{m+1} = x_{l_{m+1}}$ , with  $l_i = l_{m+1}$ . It holds that these formulas are in contradiction with the formula  $z_i < z_{m+1}$ , that is:

$$(z_i < z_{m+1}) \wedge (z_i = x_{l_i}) \wedge (z_{m+1} = x_{l_{m+1}}) \equiv \perp$$

Therefore, the disjunct  $(z_i < z_{m+1}) \wedge \psi'_k(z_0, \dots, z_{m+1}, x_0, \dots, x_{n_k})$  is equivalent to  $\perp$ , and thus can be safely removed from the disjunction.

- (b) suppose that the formula  $\psi'_k(z_0, \dots, z_{m+1}, x_0, \dots, x_{n_k})$  contains the following conjuncts:  $z_i = x_{l_i}$ ,  $z_{m+1} = x_{l_{m+1}}$  (with  $l_i \neq l_{m+1}$ ), and  $x_{l_{m+1}} < \dots < x_{l_i}$ . As in the previous case, it holds that:

$$(z_i < z_{m+1}) \wedge (z_i = x_{l_i}) \wedge (z_{m+1} = x_{l_{m+1}}) \wedge (x_{l_{m+1}} < \dots < x_{l_i}) \equiv \perp$$

Thus, also in this case, this disjunct can be safely removed from the disjunction.

(c) otherwise, it holds that the formula  $\psi'_k(z_0, \dots, z_{m+1}, x_0, \dots, x_{n_k})$  contains the following conjuncts:  $z_i = x_{l_i}$ ,  $z_{m+1} = x_{l_{m+1}}$  (with  $l_i \neq l_{m+1}$ ), and  $x_{l_i} < \dots < x_{l_{m+1}}$ . Therefore, the subformula  $z_i < z_{m+1}$  is redundant, and can be safely removed from  $\psi'_k(z_0, \dots, z_{m+1}, x_0, \dots, x_{n_k})$ . The resulting formula is a  $\exists\forall$ -formula.

After the previous transformation, we obtain:

$$\bigvee_{k=0}^{j'} (\exists z_{m+1} \cdot \exists x_0 \dots \exists x_{n_k} \cdot \psi''_k(z_0, \dots, z_{m+1}, x_0, \dots, x_{n_k}))$$

Finally, since each formula  $\psi''_k(z_0, \dots, z_{m+1}, x_0, \dots, x_{n_k})$  contains the conjunct  $z_{m+1} = x_{l_{m+1}}$ , we can safely remove the quantifier  $\exists z_{m+1}$ . We obtain the formula:

$$\bigvee_{k=0}^{j'} (\exists x_0 \dots \exists x_{n_k} \cdot \psi''_k(z_0, \dots, z_m, x_0, \dots, x_{n_k}))$$

which is a disjunction of  $\exists\forall$ -formulas.

4. Let  $\phi(z_0, \dots, z_m) \equiv \forall z_{m+1} (z_i < z_{m+1} < z_j \rightarrow \phi_1(z_0, \dots, z_m, z_{m+1}))$ , for some  $0 \leq i, j \leq m$ . By the induction hypothesis we know that  $\phi_1$  is equivalent to a disjunction  $\bigvee_k \psi_k$  where  $\psi_k$  are  $\exists\forall$ -formulas, *i.e.*, each  $\psi_k$  is of the form:

$$\begin{aligned} \psi_k \equiv \exists x_0, \dots, x_n (x_0 < \dots < x_n \wedge z_0 = x_0 \wedge \bigwedge_{l=1}^{m+1} (z_l = x_{u_l}) \wedge \\ \bigwedge_{l=0}^n \alpha_l(x_l) \wedge \bigwedge_{l=1}^n \forall y (x_{l-1} < y < x_l \rightarrow \beta_l(y))) \end{aligned}$$

We now note that we can suppose *w.l.o.g.* that the ordering constraint and the binding constraint of  $\psi_k$  imply that  $z_i$ ,  $z_{m+1}$  and  $z_j$  are ordered consecutively, *i.e.*,  $z_i < z_{m+1} < z_j$  with no other variable in between. That is because otherwise the constraints would be in conflict with the guard of the universal quantification and the disjunct could be removed from the disjunction. Take for example a disjunct of  $\psi_k$  with an ordering constraint of the type  $z_i < z_h < z_{m+1}$ , for some  $h$ . The existence of such a  $z_h$  is not guaranteed for each  $z_{m+1}$  between  $z_i$  and  $z_j$  because when  $z_{m+1} = z_i + 1$  there is no value between  $z_i$  and  $z_i + 1$  (we are on discrete time models). That said, we can now isolate all the parts of  $\psi_k$  that talk about  $z_{m+1}$ , bringing them out of the existential quantification, obtaining  $\psi_k \equiv \theta_k \wedge \eta_k$ , where:

$$\begin{aligned} \theta_k \equiv z_i < z_{m+1} < z_j \\ \wedge \alpha(z_{m+1}) \wedge \forall y (z_i < y < z_{m+1} \rightarrow \beta(y)) \wedge \forall y (z_{m+1} < y < z_i \rightarrow \beta'(y)) \end{aligned}$$

$$\eta_k \equiv \exists x_0, \dots, x_n (x_0 < \dots < x_n \wedge z_0 = x_0 \wedge \bigwedge_{l=1}^m (z_l = x_{u_l}) \wedge \bigwedge_{\substack{l=0 \\ l \neq u_{m+1}}}^n \alpha_l(x_l) \wedge \bigwedge_{\substack{l=1 \\ l-1 \neq u_i \\ l \neq u_j}}^n \forall y (x_{l-1} < y < x_l \rightarrow \beta_l(y)))$$

Now, we have  $\phi \equiv \forall z_{m+1} (z_i < z_{m+1} < z_j \rightarrow \bigvee_k (\theta_k \wedge \eta_k))$ . We can distribute the head of the implication over the disjunction, and then over the conjunction, obtaining:

$$\phi \equiv \forall z_{m+1} (\bigvee_k ((z_i < z_{m+1} < z_j \rightarrow \theta_k) \wedge (z_i < z_{m+1} < z_j \rightarrow \eta_k)))$$

In order to simplify the exposition, we now show how to proceed in the case of two disjuncts, which is easily generalizable. So suppose we have:

$$\phi \equiv \forall z_{m+1} \left( \bigvee \left( (z_i < z_{m+1} < z_j \rightarrow \theta_1) \wedge (z_i < z_{m+1} < z_j \rightarrow \eta_1) \right) \right. \\ \left. (z_i < z_{m+1} < z_j \rightarrow \theta_2) \wedge (z_i < z_{m+1} < z_j \rightarrow \eta_2) \right)$$

Now we can a) distribute the disjunction over the conjunction (*i.e.*, convert in conjunctive normal form in the case of multiple disjuncts), b) factor out the head of the implications and c) distribute the universal quantification over the conjunction, obtaining:

$$\phi \equiv \left( \begin{array}{l} \forall z_{m+1} (z_i < z_{m+1} < z_j \rightarrow \theta_1 \vee \theta_2) \\ \wedge \forall z_{m+1} (z_i < z_{m+1} < z_j \rightarrow \theta_1 \vee \eta_2) \\ \wedge \forall z_{m+1} (z_i < z_{m+1} < z_j \rightarrow \eta_1 \vee \theta_2) \\ \wedge \forall z_{m+1} (z_i < z_{m+1} < z_j \rightarrow \eta_1 \vee \eta_2) \end{array} \right)$$

Now, note that  $\eta_1$  and  $\eta_2$  do not contain  $z_{m+1}$  as a free variable, because we factored out all the parts mentioning  $z_{m+1}$  into  $\theta_1$  and  $\theta_2$  before. Therefore we can push them out from the universal quantifications, obtaining:

$$\phi \equiv \left( \begin{array}{l} \forall z_{m+1} (z_i < z_{m+1} < z_j \rightarrow \theta_1 \vee \theta_2) \\ \wedge \forall z_{m+1} (z_i < z_{m+1} < z_j \rightarrow \theta_1) \vee \eta_2 \\ \wedge \forall z_{m+1} (z_i < z_{m+1} < z_j \rightarrow \theta_2) \vee \eta_1 \\ \wedge \neg \exists z_{m+1} (z_i < z_{m+1} < z_j) \vee \eta_1 \vee \eta_2 \end{array} \right)$$

Now, note that  $\neg \exists z_{m+1} (z_i < z_{m+1} < z_j)$  is equivalent to  $z_i = z_j \vee z_j = z_i + 1$ , which is the disjunction of two formulas that can be turned into  $\exists \forall$ -formulas. Since both  $\eta_1$  and  $\eta_2$  are already  $\exists \forall$ -formulas and since we already know how to deal with conjunctions and disjunctions of  $\exists \forall$ -formulas, it remains to show that the universal quantifications in the formula above can be turned into

$\exists\forall$ -formulas. Take  $\forall z_{m+1}(z_i < z_{m+1} < z_j \rightarrow \theta_1)$ , i.e.:

$$\forall z_{m+1} \left( \begin{array}{l} z_i < z_{m+1} < z_j \\ z_i < z_{m+1} < z_j \rightarrow \wedge \alpha(z_{m+1}) \\ \wedge \forall y(z_i < y < z_{m+1} \rightarrow \beta(y)) \\ \wedge \forall y(z_{m+1} < y < z_j \rightarrow \beta'(y)) \end{array} \right)$$

Note that the first conjunct of the consequent can be removed, since it is redundant. Now, this formula is requesting  $\beta(y)$  for all  $y$  between  $z_i$  and  $z_{m+1}$ , but with  $z_{m+1}$  that ranges between  $z_i$  and  $z_j - 1$ , hence effectively requesting  $\beta(y)$  to hold between  $z_i$  and  $z_j$ . Similarly for  $\beta'(y)$ , which has to hold for all  $y$  between  $z_i + 1$  and  $z_j$ .

Hence, it is equivalent to:

$$\begin{array}{l} z_i = z_j \\ \vee z_j = z_i + 1 \\ \vee \exists x_{i+1}(z_i < x_{i+1} \wedge x_{i+1} = z_i + 1 \wedge z_j = x_{i+1} + 1 \wedge \alpha(x_{i+1})) \\ \vee \exists x_i \exists x_{i+1} \exists x_{j-1} \exists x_j \left( \begin{array}{l} x_i < x_{i+1} < x_{j-1} < x_j \\ \wedge z_i = x_i \wedge z_j = x_j \\ \wedge \alpha(x_{i+1}) \wedge \alpha(x_{j-1}) \\ \wedge \forall y(x_i < y < x_{i+1} \rightarrow \perp) \\ \wedge \forall y(x_{j-1} < y < x_j \rightarrow \perp) \\ \wedge \forall y(x_i < y < x_{j-1} \rightarrow \alpha(y) \wedge \beta(y)) \\ \wedge \forall y(x_{i+1} < y < x_j \rightarrow \alpha(y) \wedge \beta'(y)) \end{array} \right) \end{array}$$

which is a disjunction of a  $\exists\forall$ -formula and others that can be turned into disjunctions of  $\exists\forall$ -formulas. The reasoning is at all similar for  $\forall z_{m+1}(z_i < z_{m+1} < z_j \rightarrow \theta_1 \vee \theta_2)$ .

Any **coSafety-FO** formula can be translated into a disjunction of  $\exists\forall$ -formulas by Lemma 5, and then to a **coSafety-LTL**( $-\tilde{X}$ ) formula by Lemma 4. Together with Lemma 3, we obtain the following.

**Corollary 1.**  $\llbracket \text{coSafety-FO} \rrbracket^{<\omega} = \llbracket \text{coSafety-LTL}(-\tilde{X}) \rrbracket^{<\omega}$

We are now ready to state the main result of this section.

**Theorem 1.**  $\llbracket \text{coSafety-LTL} \rrbracket = \llbracket \text{coSafety-FO} \rrbracket$

*Proof.* We know that  $\llbracket \text{coSafety-LTL} \rrbracket = \llbracket \text{coSafety-LTL}(-\tilde{X}) \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega$  by Observation 1 and Lemma 1. Since  $\llbracket \text{coSafety-LTL}(-\tilde{X}) \rrbracket^{<\omega} = \llbracket \text{coSafety-FO} \rrbracket^{<\omega}$  by Corollary 1, we have that  $\llbracket \text{coSafety-LTL}(-\tilde{X}) \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega = \llbracket \text{coSafety-FO} \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega$ . Then, by Lemma 2 we have that  $\llbracket \text{coSafety-FO} \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega = \llbracket \text{coSafety-FO} \rrbracket$ , hence  $\llbracket \text{coSafety-LTL} \rrbracket = \llbracket \text{coSafety-FO} \rrbracket$ .

**Corollary 2.**  $\llbracket \text{Safety-LTL} \rrbracket = \llbracket \text{Safety-FO} \rrbracket$

### 4 Safety-FO captures LTL-definable safety languages

In this section, we prove that  $\text{coSafety-FO}$  captures LTL-definable co-safety languages. By duality, we have that  $\text{Safety-FO}$  captures LTL-definable safety languages, and by the equivalence shown in the previous Section, this provides a novel proof of the fact that  $\text{Safety-LTL}$  captures LTL-definable safety languages. We start by characterizing co-safety languages in terms of LTL over finite words.

**Lemma 6.**  $\llbracket \text{LTL} \rrbracket \cap \text{coSAFETY} = \llbracket \text{LTL} \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega$

*Proof.* ( $\subseteq$ ) By Proposition 2 we know that each language  $\mathcal{L} \in \llbracket \text{LTL} \rrbracket \cap \text{coSAFETY}$  is definable by a formula of the form  $\text{F}\alpha$  where  $\alpha \in \text{LTL}_P$ . Hence for each  $\sigma \in \mathcal{L}$  there exists an  $n$  such that  $\sigma, n \models \alpha$ , hence  $\sigma_{[0,n]}, n \models \alpha$ . Note that  $\sigma_{[n+1,\infty]}$  is unconstrained. By replacing all the *since/yesterday/weak yesterday* operators in  $\alpha$  with *until/tomorrow/weak tomorrow* operators, we obtain an LTL formula  $\alpha^r$  such that  $(\sigma_{[0,n]})^r, 0 \models \alpha^r$  (where  $\sigma^r$  is the reverse of  $\sigma$ ). Since LTL captures star-free languages [12] and star-free languages are closed by reversal, there is also an LTL formula  $\beta$  such that  $\sigma_{[0,n]}, 0 \models \beta$ . Hence  $\mathcal{L} = \mathcal{L}^{<\omega}(\beta) \cdot (2^\Sigma)^\omega$ , and we proved that  $\llbracket \text{LTL} \rrbracket \cap \text{coSAFETY} \subseteq \llbracket \text{LTL} \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega$ .

( $\supseteq$ ) Given  $\mathcal{L} \in \llbracket \text{LTL} \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega$ , we know  $\mathcal{L} = \mathcal{L}^{<\omega}(\beta) \cdot (2^\Sigma)^\omega$  for some LTL formula  $\beta$ . Hence, for each  $\sigma \in \mathcal{L}$  there is an  $n$  such that  $\sigma_{[0,n]}, 0 \models \beta$ . Since LTL captures star-free languages and star-free languages are closed by reversal, there is an LTL formula  $\alpha^r$  such that  $(\sigma_{[0,n]})^r, 0 \models \alpha^r$ . Now, by replacing all the *until/tomorrow/weak tomorrow* operators in  $\alpha^r$  with *since/yesterday/weak yesterday* operators, we obtain an  $\text{LTL}_P$  formula  $\alpha$  such that  $\sigma_{[0,n]}, n \models \alpha$ . Hence,  $\sigma$  is such that there is an  $n$  such that  $\sigma, n \models \alpha$ , i.e.,  $\sigma \models \text{F}\alpha$ . Therefore, by Proposition 2,  $\mathcal{L} \in \llbracket \text{LTL} \rrbracket \cap \text{coSAFETY}$ , and this in turn implies that  $\llbracket \text{LTL} \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega \subseteq \llbracket \text{LTL} \rrbracket \cap \text{coSAFETY}$ .

Now, we show that, over finite words, universal temporal operators are unneeded.

**Lemma 7.**  $\llbracket \text{LTL} \rrbracket^{<\omega} = \llbracket \text{Safety-LTL} \rrbracket^{<\omega} = \llbracket \text{coSafety-LTL} \rrbracket^{<\omega}$

*Proof.* Since  $\text{Safety-LTL}$  and  $\text{coSafety-LTL}$  are fragments of LTL, we only need to show one direction, i.e., that  $\llbracket \text{LTL} \rrbracket^{<\omega} \subseteq \llbracket \text{Safety-LTL} \rrbracket^{<\omega}$  and  $\llbracket \text{LTL} \rrbracket^{<\omega} \subseteq \llbracket \text{coSafety-LTL} \rrbracket^{<\omega}$ . At first, we show that universal temporal operators are not needed over finite words. For each LTL formula  $\phi$ , we can build an equivalent  $\text{coSafety-LTL}$  formula with only existential temporal operators. The *globally* operator can be replaced by means of an *until* operator whose existential part always refers to the last position of the word. In turn, this can be done with the formula  $\tilde{\text{X}}\perp$ , which is true only at the final position:

$$\text{G}\phi \equiv \phi \mathcal{U} (\phi \wedge \tilde{\text{X}}\perp)$$

Similarly, the *release* operator can be expressed by means of a *globally* operator in disjunction with an *until* operator:

$$\phi_1 \mathcal{R} \phi_2 \equiv \text{G}\phi_2 \vee (\phi_2 \mathcal{U} (\phi_1 \wedge \phi_2)) \equiv (\phi_2 \mathcal{U} (\phi_2 \wedge \tilde{\text{X}}\perp)) \vee (\phi_2 \mathcal{U} (\phi_1 \wedge \phi_2))$$



Hence  $\llbracket \text{LTL} \rrbracket^{<\omega} = \llbracket \text{coSafety-LTL} \rrbracket^{<\omega}$ . Now, if we exploit the duality between the *eventually/until* and the *globally/release* operators, we obtain:

$$\begin{aligned} F\phi &\equiv \phi \mathcal{R} (\phi \vee \text{XT}) \\ \phi_1 \mathcal{U} \phi_2 &\equiv \phi_2 \mathcal{R} (\phi_2 \vee \text{XT}) \wedge \phi_2 \mathcal{R} (\phi_1 \vee \phi_2) \end{aligned}$$

Hence  $\llbracket \text{LTL} \rrbracket^{<\omega} = \llbracket \text{Safety-LTL} \rrbracket^{<\omega}$ .

Then, we relate coSafety-LTL on finite words and coSafety-FO.

**Lemma 8.**  $\llbracket \text{coSafety-LTL} \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega = \llbracket \text{coSafety-FO} \rrbracket$

*Proof.* ( $\subseteq$ ) We have that  $\llbracket \text{coSafety-LTL} \rrbracket^{<\omega} = \llbracket \text{LTL} \rrbracket^{<\omega}$  by Lemma 7, and this implies that  $\llbracket \text{coSafety-LTL} \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega = \llbracket \text{LTL} \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega$ , and  $\llbracket \text{coSafety-LTL} \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega = \llbracket \text{FO} \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega$  by Proposition 1. Now, let  $\phi \in \text{FO}$ , and suppose *w.l.o.g.* that  $\phi$  is in *negated normal form*. We define the formula  $\phi'(x, y)$ , where  $x$  and  $y$  are two fresh variables that do not occur in  $\phi$ , as the formula obtained from  $\phi$  by a) replacing each subformula of  $\phi$  of type  $\exists z\phi_1$  with  $\exists z(x \leq z \wedge \phi_1)$ , and b) by replacing each subformula of  $\phi$  of type  $\forall z\phi_1$  with  $\forall z(x \leq z < y \rightarrow \phi_1)$ . Now, consider the formula  $\psi \equiv \exists y(x \leq y \wedge \phi'(x, y))$ . Note that  $\psi$  is a coSafety-FO formula. When interpreted over *infinite* words, the models of  $\psi$  are exactly those containing a prefix that belongs to  $\mathcal{L}^{<\omega}(\phi)$ , with the remaining suffix unconstrained, that is  $\mathcal{L}(\psi) = \mathcal{L}^{<\omega}(\phi) \cdot (2^\Sigma)^\omega$ , hence  $\llbracket \text{FO} \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega \subseteq \llbracket \text{coSafety-FO} \rrbracket$ , and this implies that  $\llbracket \text{coSafety-LTL} \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega \subseteq \llbracket \text{coSafety-FO} \rrbracket$ .

( $\supseteq$ ) We know by Lemma 2 that  $\llbracket \text{coSafety-FO} \rrbracket = \llbracket \text{coSafety-FO} \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega$ . Since coSafety-FO formulas are also FO formulas, we have  $\llbracket \text{coSafety-FO} \rrbracket \subseteq \llbracket \text{FO} \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega$ . By Proposition 1 and Lemma 7, we obtain that  $\llbracket \text{coSafety-FO} \rrbracket \subseteq \llbracket \text{coSafety-LTL} \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega$ .

We are ready now to state the main result.

**Theorem 2.**  $\llbracket \text{LTL} \rrbracket \cap \text{coSAFETY} = \llbracket \text{coSafety-FO} \rrbracket$

*Proof.* We know that  $\llbracket \text{LTL} \rrbracket \cap \text{coSAFETY} = \llbracket \text{LTL} \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega$  by Lemma 6. Then, by Lemma 7 we know that  $\llbracket \text{LTL} \rrbracket^{<\omega} = \llbracket \text{coSafety-LTL} \rrbracket^{<\omega}$ , and this in turn implies that  $\llbracket \text{LTL} \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega = \llbracket \text{coSafety-LTL} \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega$ . Since  $\llbracket \text{coSafety-LTL} \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega = \llbracket \text{coSafety-FO} \rrbracket$  by Lemma 8, we conclude that  $\llbracket \text{LTL} \rrbracket \cap \text{coSAFETY} = \llbracket \text{coSafety-FO} \rrbracket$ .

This result together with Theorem 1 allow us to conclude the following.

**Theorem 3.**  $\llbracket \text{Safety-LTL} \rrbracket = \llbracket \text{LTL} \rrbracket \cap \text{SAFETY}$

Note that by Observation 1 and Lemma 1 on one hand, and by Lemmas 6 and 7 on the other, the question of whether  $\llbracket \text{Safety-LTL} \rrbracket = \llbracket \text{LTL} \rrbracket \cap \text{SAFETY}$  can be reduced to whether  $\llbracket \text{coSafety-LTL} \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega = \llbracket \text{coSafety-LTL}(-\tilde{X}) \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega$ . If coSafety-LTL and coSafety-LTL( $-\tilde{X}$ ) were equivalent over finite words, this would already prove Theorem 3. However, we can prove this is not the case.

**Theorem 4.**  $\llbracket \text{coSafety-LTL} \rrbracket^{<\omega} \neq \llbracket \text{coSafety-LTL}(-\tilde{X}) \rrbracket^{<\omega}$

*Proof.* Note that in  $\text{coSafety-LTL}(-\tilde{X})$  we only have existential temporal modalities and we cannot hook the final position of the word without the *weak tomorrow* operator. For these reasons, given a  $\text{coSafety-LTL}(-\tilde{X})$  formula  $\phi$ , with a simple structural induction we can prove that for each  $\sigma \in (2^\Sigma)^+$  such that  $\sigma \models \phi$ , it holds that  $\sigma\sigma' \models \phi$  for any  $\sigma' \in (2^\Sigma)^+$ , *i.e.*, all the extensions of  $\sigma$  satisfy  $\phi$  as well. This implies that  $\mathcal{L}^{<\omega}(\phi)$  is either empty (*i.e.*, if  $\phi$  is unsatisfiable) or infinite. Instead, by using the *weak tomorrow* operator to hook the last position of the word, we can describe a finite non-empty language, for example as in the formula  $\phi \equiv a \wedge X(a \wedge \tilde{X}\perp)$ . The language of  $\phi$  is  $\mathcal{L}(\phi) = \{\mathbf{aa}\}$ , including exactly one word, hence  $\mathcal{L}(\phi)$  cannot be described without the *weak tomorrow* operator.

Note that Theorem 4 does *not* contradict Theorem 3, that is, it does not imply that  $\llbracket \text{coSafety-LTL} \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega \neq \llbracket \text{coSafety-LTL}(-\tilde{X}) \rrbracket^{<\omega} \cdot (2^\Sigma)^\omega$ . For example, consider again the formula  $a \wedge X(a \wedge \tilde{X}\perp)$ . It cannot be expressed without the *weak tomorrow* operator, yet it holds that:  $\mathcal{L}^{<\omega}(a \wedge X(a \wedge \tilde{X}\perp)) \cdot (2^\Sigma)^\omega = \mathcal{L}^{<\omega}(a \wedge Xa) \cdot (2^\Sigma)^\omega$ .

## 5 Conclusions

In this paper, we gave a first-order characterization of safety and co-safety languages, by means of two fragments of first-order logic,  $\text{Safety-FO}$  and  $\text{coSafety-FO}$ . These fragments of  $\text{S1S[FO]}$  provide a very natural syntax and are *expressively complete* with regards to LTL-definable safety and co-safety languages.

The core theorem establishes a correspondence between  $\text{Safety-FO}$  (resp.,  $\text{coSafety-FO}$ ) and  $\text{Safety-LTL}$  (resp.,  $\text{coSafety-LTL}$ ), and thus it can be viewed as a special version of Kamp’s theorem for safety (resp., co-safety) properties. Thanks to these new fragments, we were able to provide a novel, compact, and self-contained proof of the fact that  $\text{Safety-LTL}$  captures LTL-definable safety languages. Such a result was previously proved by Chang *et al.* [5], but in terms of the properties of a non-trivial transformation from star-free languages to LTL by Zuck [21]. As a by-product, we provided a number of results that relate the considered languages when interpreted over finite and infinite words. In particular, we highlighted the expressive power of the *weak tomorrow* temporal modality, showing it to be essential in  $\text{coSafety-LTL}$  over finite words.

Different equivalent characterizations of LTL are known, in terms of (i) first-order logic, (ii) regular expressions, (iii) automata, and (iv) monoids (see the summary by Thomas in [19]). This work focuses on the first item, but for LTL-definable safety languages. A natural follow-up would be to investigate the other items, looking for what kind of automata (resp., regular expressions, monoids) captures exactly safety and co-safety LTL-definable languages. While on finite traces simple characterizations in terms of automata and syntactic monoids exist, the infinite-traces scenario is more complex: there exists a characterization of LTL in terms of counter-free automata [13] and the one for safety  $\omega$ -regular languages seems not to be difficult (see *e.g.*, terminal automata [4, 18]), but their combination requires to have a canonical (minimal) representation of a (Muller/Rabin/Streett) automata corresponding to any  $\omega$ -regular language.

## References

1. Biere, A., Artho, C., Schuppan, V.: Liveness checking as safety checking. *Electronic Notes in Theoretical Computer Science* **66**(2), 160–177 (2002)
2. Buchi, J.R.: Weak second-order arithmetic and finite automata. *Journal of Symbolic Logic* **28**(1) (1963)
3. Büchi, J.R.: On a decision method in restricted second order arithmetic. In: *The collected works of J. Richard Büchi*, pp. 425–435. Springer (1990)
4. Cerná, I., Pelánek, R.: Relating hierarchy of temporal properties to model checking. In: Rován, B., Vojtás, P. (eds.) *Proceedings of the 28th International Symposium on Mathematical Foundations of Computer Science 2003. Lecture Notes in Computer Science*, vol. 2747, pp. 318–327. Springer (2003). [https://doi.org/10.1007/978-3-540-45138-9\\_26](https://doi.org/10.1007/978-3-540-45138-9_26)
5. Chang, E.Y., Manna, Z., Pnueli, A.: Characterization of temporal property classes. In: Kuich, W. (ed.) *Proceedings of the 19th International Colloquium on Automata, Languages and Programming. Lecture Notes in Computer Science*, vol. 623, pp. 474–486. Springer (1992). [https://doi.org/10.1007/3-540-55719-9\\_97](https://doi.org/10.1007/3-540-55719-9_97)
6. De Giacomo, G., Vardi, M.Y.: Linear temporal logic and linear dynamic logic on finite traces. In: Rossi, F. (ed.) *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. pp. 854–860. IJCAI/AAAI (2013)
7. De Giacomo, G., Vardi, M.Y.: Synthesis for LTL and LDL on finite traces. In: Yang, Q., Wooldridge, M.J. (eds.) *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*. pp. 1558–1564. AAAI Press (2015)
8. Gabbay, D., Pnueli, A., Shelah, S., Stavi, J.: On the temporal analysis of fairness. In: *Proceedings of the 7th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*. pp. 163–173 (1980)
9. Giacomo, G.D., Masellis, R.D., Montali, M.: Reasoning on LTL on finite traces: Insensitivity to infiniteness. In: Brodley, C.E., Stone, P. (eds.) *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. pp. 1027–1033. AAAI Press (2014)
10. Kamp, J.A.W.: *Tense logic and the theory of linear order*. University of California, Los Angeles (1968)
11. Kupferman, O., Vardi, M.Y.: Model checking of safety properties. *Formal Methods in System Design* **19**(3), 291–314 (2001)
12. Lichtenstein, O., Pnueli, A., Zuck, L.: The glory of the past. In: *Workshop on Logic of Programs*. pp. 196–218. Springer (1985)
13. McNaughton, R., Papert, S.A.: *Counter-Free Automata* (MIT research monograph no. 65). The MIT Press (1971)
14. Pnueli, A.: The temporal logic of programs. In: *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*. pp. 46–57. IEEE (1977)
15. Rabinovich, A.: A Proof of Kamp’s theorem. *Logical Methods in Computer Science* **Volume 10, Issue 1** (Feb 2014). [https://doi.org/10.2168/LMCS-10\(1:14\)2014](https://doi.org/10.2168/LMCS-10(1:14)2014), <https://lmcs.episciences.org/730>
16. Sherman, R., Pnueli, A., Harel, D.: Is the interesting part of process logic uninteresting? A translation from PL to PDL. *SIAM J. Comput.* **13**(4), 825–839 (1984). <https://doi.org/10.1137/0213051>
17. Sistla, A.P.: Safety, liveness and fairness in temporal logic. *Formal Aspects of Computing* **6**(5), 495–511 (1994)
18. Strejcek, J.: *Linear temporal logic: Expressiveness and model checking*. Ph.D. thesis, Faculty of Informatics, Masaryk University in Brno (2004)

19. Thomas, W.: Safety-and liveness-properties in propositional temporal logic: characterizations and decidability. *Banach Center Publications* **1**(21), 403–417 (1988)
20. Zhu, S., Tabajara, L.M., Li, J., Pu, G., Vardi, M.Y.: A Symbolic Approach to Safety LTL Synthesis. In: Strichman, O., Tzoref-Brill, R. (eds.) *Proceedings of the 13th International Haifa Verification Conference*. *Lecture Notes in Computer Science*, vol. 10629, pp. 147–162. Springer (2017). [https://doi.org/10.1007/978-3-319-70389-3\\_10](https://doi.org/10.1007/978-3-319-70389-3_10)
21. Zuck, L.: Past temporal logic. *Weizmann Institute of Science* **67** (1986)

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

