Simplifying Inductive Schemes in Temporal Logic

Pablo Cordero

Dept. Applied Mathematic, University of Málaga, Spain
http://webpersonal.uma.es/~pcordero/My_personal_web/Wellcome.html
pcordero@uma.es

Inmaculada Fortes

Dept. Applied Mathematic, University of Málaga, Spain
ifortes@ctima.uma.es

Inmaculada P. de Guzmán

Dept. Applied Mathematic, University of Málaga, Spain
guzman@ctima.uma.es

Sixto Sánchez

Dept. Applied Mathematic, University of Málaga, Spain
sixto@uma.es

Abstract

In propositional temporal logic, the combination of the connectives "tomorrow" and "always in the future" require the use of induction tools. In this paper, we present a classification of inductive schemes for propositional linear temporal logic that allows the detection of loops in decision procedures. In the design of automatic theorem provers, these schemes are responsible for the searching of efficient solutions for the detection and management of loops. We study which of these schemes have a good behavior in order to give a set of reduction rules that allow us to compute these schemes efficiently and, therefore, be able to eliminate these loops. These reduction laws can be applied previously and during the execution of any automatic theorem prover. All the reductions introduced in this paper can be considered a part of the process for obtaining a normal form of a given formula.

2012 ACM Subject Classification Theory of computation → Modal and temporal logics

Keywords and phrases Linear Temporal Logic, Inductive Schemes, Loop-check

Digital Object Identifier 10.4230/LIPIcs.TIME.2019.19

Funding *Pablo Cordero*: Partially supported by Project TIN2017-89023-P of the Science and Innovation Ministry of Spain, co-funded by the EU Regional Development (ERDF).
Inmaculada Fortes: Partially supported by Project PGC2018-095869-B-I00 of the Science and Innovation Ministry of Spain.
Inmaculada P. de Guzmán: Partially supported by Project TIN2017-89023-P of the Science and Innovation Ministry of Spain, co-funded by the EU Regional Development (ERDF).

1 Introduction

The notion of induction is a relevant topic in temporal logic and related areas. In fact, in [11], for the philosophical question: *What is temporal logic?* five answers are provided by Michael Fisher. One answer is that Propositional Temporal Logic characterizes simple induction. Another is that "Propositional Temporal Logic can be seen as a multi-modal logic, comprising two modalities, $\boxed{1}$ and $\boxed{*}$, which interact closely". The modal operator $\boxed{1}$ corresponds to the "next" relation and $\boxed{*}$ is a modal operator that corresponds to the "always" relation. As usual, possibility modal operators are denoted by $\langle 1 \rangle$ and $\langle * \rangle$. Thus, the *interaction* axiom between both modal operators is given by the induction axiom i.e. $\vdash \boxed{*}(\varphi \to \boxed{1}\varphi) \to (\varphi \to \boxed{*}\varphi)$.

It is well-known the mathematical induction principle can be generalized. In the same way, in Propositional Temporal Logic, there are other formulas which match with the idea of induction. Thus, for example, the formula $\boxed{*}(\langle * \rangle\varphi \to \boxed{1}\boxed{1}\varphi) \to (\boxed{1}\boxed{1}\boxed{1}\boxed{1}\varphi \to \boxed{*}\boxed{1}\boxed{1}\varphi)$ is a tautology that follows the induction idea. This kind of scheme is especially relevant in the design of automatic theorem provers for temporal logics where they are responsible for the searching of efficient solutions for the detection and management of loops (see, for example [3, 7, 12]).

The applicability of temporal logics in the field of information sciences does not need to be justified. As a simple example, an important application of temporal logics is the specification of the properties to be formally verified in a model checking [5] where the system requirements are usually expressed by means of temporal logic formulas [13]. What makes temporal logic particularly attractive is the enormous success of model checking, which, under appropriate assumptions on the system specification, can make the verification of temporal logic properties automatic.

In verification, temporal logics are usually enriched with modal connectives such as knowledge, beliefs, intentions, norms, etc. Although, the extension of results in temporal logics to these modal×temporal logics is not straightforward [16], any advance in temporal logic technics is a positive enhancement.

Induction is also a relevant issue in model checking, since it is related to the analysis of invariants and loops and, specifically, with infinite loops detection. Induction itself is relevant because reasoning about the partial correctness of programs often requires proofs by induction, in particular for reasoning about recursive functions. In [10], a proof method for inductive theorem proving is developed in rewriting a system framework. In [9], the author develops a general proof method that combines logic and induction. He provides a solid and uniform mathematical foundation to induction proof methodologies for a wide variety of formal specification frameworks and shows its applicability through several examples of formal verification proof. As it has been mentioned, induction plays a central role in temporal logic. Combining both topics, in [1], a linear temporal logic of rewriting is introduced to be used in model checking.

One of the main problems with the model checking technique is to make it as least expensive as possible. That is, a balance is needed between expressivity of the language and the cost of the logic-based methods. Numerous logics have been used for this objective [16, 2, 17]. In this paper we center on the very influential Linear Temporal Logic, considering the "always" and the "next" operators, introduced by Pnueli [19]. In [21], Sistla and Clarke studied the complexity of the satisfiability and model checking problems in this logic and, in [8], a more general study can be found regarding the complexities of temporal logic model checking.

Specifically, the considered logic in this paper is $FNext$, which is a propositional linear discrete temporal logic with three temporal connectives (using Prior's notation): $\oplus$ (tomorrow), $\mathtt{G}$ (always in the future) and $\mathtt{F}$ (some time in the future). As mentioned, it is well-known that the combination itself of the connectives $\oplus$ and $\mathtt{G}$ requires the study of induction [11].

Among the formulas which involve the induction idea, we are especially interested in those where a unique propositional symbol has incidence. In propositional logic, literals are limited to propositional symbols and their negations (aside from constants) whereas in modal/temporal logics they are extended to include modal/temporal connectives (see Section 2.2). In the medium term, our aim is to extend the notion of literal including induction schemes. There is no doubt about the relevance of literals in issues such as normal forms, automatic reasoning and the SAT problem. As an example, in [14], literals are used to find rough and fuzzy-rough set reducts. In [20] they are used to extend formal concept analysis considering negative information.

In this paper we focus on the study of induction schemes as a previous stage in the generalization of literals towards the improvement of normalization processes. To this aim, we analyze those formulas which involves the induction idea. Thus, in the same way that the interaction axiom can be extended, the pure induction (i.e. $\oplus p \wedge \mathtt{G}(p \to \oplus p) \equiv \mathtt{G}p$) can be generalized into induction schemes. That is, we consider formulas $A \wedge \mathtt{G}(B \to C)$ where A semantically implies $\mathtt{F}B$ and C implies $\mathtt{F}B$. As mentioned, we especially center on those cases in which A, B and C are formulas where a unique propositional symbol has incidence. Specifically, in this paper we characterize those formulas that can be simplified providing an equivalent without loops.

All simplifications introduced in this paper have a general purpose and can be used in deduction processes, automatic reasoning, normalization techniques, etc. Some of the given equivalences are able to simplify the formula by reducing the size and are useful in problems where cost depends on the size of the input. Moreover, although we describe our method on $FNext$, our approach can also be applied to totally expressive logic US developed by Hans Kamp [15].

The paper is organized as follows: In Section 2, the preliminary definitions and $FNext$ logic are introduced. In Section 3, the aim of the paper is specified in detail. In Section 4, G-clauses are studied and, in Section 5, inductive schemes are defined and, from among all the studied formulas, we classify them into those that can be transformed into an equivalent expression that does not involve loops and those corresponding with inductive schemes. The final section is devoted to conclusions and future works.

2 $FNext$ Logic

In this section, we introduce a temporal propositional logic, with an infinite, linear and discrete flow of time denoted by $FNext$. We develop the language, semantics, modalities and simplification laws for $FNext$.

The alphabet of the language of the logic $FNext$ consists of the following: A denumerable set, $\mathcal{V}$, of propositional variables, a set of Boolean connectives: $\{\top, \bot, \neg, \wedge, \vee, \to\}$, and a set of temporal connectives: $\{\oplus, \mathtt{F}, \mathtt{G}\}$. The language is denoted by $\mathcal{L}$ and well-formed formulas, (hereafter, wffs), in $FNext$ are generated by the construction rules of classical propositional logic, plus the following rule: *"If A is a wff, then $\oplus A$, $\mathtt{F}A$ and $\mathtt{G}A$ are wffs".*

The meaning of an expression like $\oplus A$, is : "*A will occur tomorrow*"; while the meaning of an expression like $\mathtt{F}A$ is: "*in the future, A will occur*" and of an expression like $\mathtt{G}A$ is: "*in the future, A will always occur*". We shall consider F and G as connectives of strict future. In what follows, we consider $\oplus^k A = \oplus \oplus^{k-1} A$ if $k \geq 1$ and $\oplus^0 A = A$.

2.1 Semantics for $FNext$

For the semantics of $FNext$ we consider always the temporal flow $(\mathbb{Z}, <)$ (i.e., the existence of a first instant of time is not demanded), where $<$ is the standard ordering on $\mathbb{Z}$, and interpretations, which are mappings from the language to $2^{\mathbb{Z}}$ assigning to each wff the set of instants in $\mathbb{Z}$ where such wff is true.

▶ **Definition 1.** *A* model *is a tuple* $(\mathbb{Z}, <, h)$ *where* $h \colon \mathcal{L} \longrightarrow 2^{\mathbb{Z}}$ *is a function, called* temporal interpretation, *satisfying the following conditions:*

1. $h(\top) = \mathbb{Z}$, $h(\bot) = \varnothing$,
2. $h(\neg A) = \mathbb{Z} \smallsetminus h(A)$, $h(A \wedge B) = h(A) \cap h(B)$, $h(A \vee B) = h(A) \cup h(B)$
3. $h(\oplus A) = \{t \in \mathbb{Z} \mid t + 1 \in h(A)\}$

4. $h(FA) = \{t \in \mathbb{Z} \mid (t, \infty) \cap h(A) \neq \varnothing\}$
5. $h(GA) = \{t \in \mathbb{Z} \mid (t, \infty) \subseteq h(A)\}$

where $(t, \infty) = \{z \in \mathbb{Z} \mid t < z\}$.

As usual, other classical connectives are introduced as follows:

$$A \to B =_{def} \neg A \vee B \quad \text{and} \quad A \leftrightarrow B =_{def} (A \to B) \wedge (B \to A)$$

Observe that the connective $\mathtt{F}$ can be also introduced as a definite connective like $\mathtt{F}A =_{def} \neg\mathtt{G}\neg A$. Thus, from now on, we focus our study on formulas involving $\mathtt{G}$.

Given two wffs A and B, the notions of validity, satisfiability, logical consequence and equivalence are defined in standard way: A is *valid* if, for every interpretation h, we have that $h(A) = \mathbb{Z}$, and it is denoted by $\models A$. A is *satisfiable* if an interpretation h and $t \in \mathbb{Z}$ exists, such that $t \in h(A)$. In this case, we say that A is true at t, and it is denoted by $(h, t) \models A$ and, when no confusion arises, it is denoted as $t \models A$. B is a *logical consequence* of A, denoted by $A \models B$, if for all interpretation h and all $t \in \mathbb{Z}$, $(h, t) \models A$ implies $(h, t) \models B$. Finally, A and B are *equivalent*, denoted by $A \equiv B$, if $A \models B$ and $B \models A$. That is, for all interpretation h, $h(A) = h(B)$.

In $FNext$, the propositional classical logical laws hold. Moreover, the following proposition establishes the basic laws concerning temporal connectives.

▶ **Proposition 2.** *If $A, B \in \mathcal{L}$, then the following equivalences hold:*
1. $FFA \equiv \oplus FA \equiv F \oplus A$; $GGA \equiv \oplus GA \equiv G \oplus A$,
2. *If $\gamma \in \{\oplus, F, G\}$, then $\gamma GFA \equiv GFA$ and $\gamma FGA \equiv FGA$.*

The following laws concern the interaction between classical and temporal connectives.

▶ **Proposition 3.** *If $A, B \in \mathcal{L}$, then the following equivalences hold:*
1. $\oplus\bot \equiv F\bot \equiv G\bot \equiv \bot$ *and* $\oplus\top \equiv F\top \equiv G\top \equiv \top$
2. $\neg \oplus A \equiv \oplus\neg A$; $\neg FA \equiv G\neg A$; $\neg GA \equiv F\neg A$,
3. $\oplus(A \vee B) \equiv \oplus A \vee \oplus B$; $\oplus(A \wedge B) \equiv \oplus A \wedge \oplus B$,
4. $F(A \vee B) \equiv FA \vee FB$ *and* $G(A \wedge B) \equiv GA \wedge GB$.

2.2 Modalities and literals in $FNext$

Proposition 2 leads to the definition of modalities (sequences of temporal connectives) and their canonical representation.

▶ **Definition 4.** *The set of modalities is defined as follows*

$$\mathcal{M}d = \{\Gamma = \gamma_1 \dots \gamma_n \mid n \in \mathbb{N}, \gamma_i \in \{\oplus, F, G\} \text{ for all } 1 \leq i \leq n\}$$

and the set of canonical modalities is as follows

$$\mathcal{M}d_c = \{FG, GF\} \cup \{\oplus^{k+1}, F\oplus^k, G\oplus^k \mid k \in \mathbb{N}\}$$

Thus, as a direct consequence of Proposition 2, we have the following theorem that gives meaning to the name of canonical modalities.

▶ **Theorem 5.** *Given a wff A, for any modality $\Gamma \in \mathcal{M}d$, there exists a canonical modality $\Gamma_c \in \mathcal{M}d_c$ such that $\Gamma A \equiv \Gamma_c A$.*

▶ **Example 6.** $\mathtt{F} \oplus \oplus\mathtt{F} \oplus \oplus \oplus \mathtt{F} \oplus A \equiv \mathtt{F} \oplus^8 A$ and $\mathtt{F} \oplus \mathtt{GF} \oplus \oplus\mathtt{G} \oplus A \equiv \mathtt{FG}A$, for any wff A.

▶ **Definition 7** (Literals). *Let $\mathcal{V}$ be the propositional variable set and $p \in \mathcal{V}$. Then:*

1. *A formula p or $\neg p$ will be named classical literal on p and denoted by ℓ_p. Then, $\mathcal{V}^{\pm}$ denotes the set of classical literals $\{p, \neg p \mid p \in \mathcal{V}\}$.*
2. *Given $\ell_p \in \mathcal{V}^{\pm}$, the set of* temporal literals on ℓ_p *is*

$$Lit(\ell_p) = \{\top, \bot\} \cup \{\texttt{FG}\ell_p, \texttt{GF}\ell_p\} \cup \{\oplus^k \ell_p, \texttt{F}\oplus^k \ell_p, \texttt{G}\oplus^k \ell_p \mid k \in \mathbb{N}\}$$

3. *The* temporal literal set *is:* $Lit = \bigcup_{\ell_p \in \mathcal{V}^{\pm}} Lit(\ell_p)$.

From now on, when no confusion arises, the adjective "temporal" will be omitted.

Theorem 5 ensures that, for any classical literal $\ell_p \in \mathcal{V}^{\pm}$ and any modality $\Gamma \in \mathcal{M}d$ there exists one (and only one) temporal literal $\ell \in Lit$ such that $\Gamma\ell_p \equiv \ell$. The unicity of this literal is because there does not exist two different literals that are equivalent.

On the other hand, in Subsection 2.1, we have introduced the notion of logical consequence, denoted by $\models$, which, in the set of literals, can be seen as an order relation. Moreover, the poset $(Lit, \models)$ is a lattice as the following proposition states.

▶ **Proposition 8** ([6]). *$(Lit, \models)$ is a lattice satisfying the following conditions:*

1. *$(Lit(\ell_p), \models)$ is a sublattice, for all $\ell_p \in \mathcal{V}^{\pm}$ (see Figure 1).*
2. *For all $\ell_1 \in Lit(\ell_p)$ and $\ell_2 \in Lit(\ell_q)$ with $\ell_p \neq \ell_q$.*

$$\ell_1 \models \ell_2 \quad \textit{if and only if} \quad \ell_1 = \bot \textit{ or } \ell_2 = \top$$

$$\begin{tikzcd}
\top & F\ell_p \arrow[l] & F\oplus\ell_p \arrow[l] & F\oplus^2\ell_p \arrow[l] & F\oplus^3\ell_p \arrow[l] & \cdots \arrow[l] & GF\ell_p \arrow[l] \\
\ell_p \arrow[u] & \oplus\ell_p \arrow[u] & \oplus^2\ell_p \arrow[u] & \oplus^3\ell_p \arrow[u] & \oplus^4\ell_p \arrow[u] & \cdots \arrow[u] & \\
\bot \arrow[r] \arrow[u] & G\ell_p \arrow[r] \arrow[u] & G\oplus\ell_p \arrow[r] \arrow[u] & G\oplus^2\ell_p \arrow[r] \arrow[u] & G\oplus^3\ell_p \arrow[r] \arrow[u] & \cdots \arrow[r] \arrow[u] & FG\ell_p \arrow[uu]
\end{tikzcd}$$

Figure 1 The sublattice $(Lit(\ell_p), \models)$.

2.3 Specific distribution laws

The item 2 in Proposition 2 reveals the special behavior of the modalities `FG` and `GF`: both *"absorb to any other finite sequence of temporal connectives"*. This is due to the fact that, for any interpretation, if there exists an instant in which the formula is true then it is true in every instant. This characteristic will be named using the adjective *atemporal*. Similarly, there exist formulas that can be *projected* toward the future or the past. The following definition formalize these ideas.

▶ **Definition 9.** *Let $[t, \infty) = \{t' \in \mathbb{Z} \,|\, t \leq t'\}$ and $(-\infty, t] = \{t' \in \mathbb{Z} \,|\, t' \leq t\}$. A wff A is said to be*

- projectable to the future *if $t \in h(A)$ implies $[t, \infty) \subseteq h(A)$, for all temporal interpretation h, i.e. either $h(A) = \varnothing$, or $h(A) = \mathbb{Z}$, or $h(A) = [t_0, \infty)$ for some $t_0 \in \mathbb{Z}$.*
- projectable to the past *if $t \in h(A)$ implies $(-\infty, t] \subseteq h(A)$, for all temporal interpretation h, i.e. either $h(A) = \varnothing$, or $h(A) = \mathbb{Z}$, or $h(A) = (-\infty, t_0]$ for some $t_0 \in \mathbb{Z}$.*
- atemporal *if $h(A) = \varnothing$ or $h(A) = \mathbb{Z}$, for all temporal interpretation h. So, A is atemporal if and only if A is projectable to the future and to the past.*

The following immediate results allow to characterize syntactically the formulas that are projectable: for all wffs A and B,

1. $\top$ and $\bot$ are atemporal.
2. $\mathtt{G}A$ is projectable to the future.
3. $\mathtt{F}A$ is projectable to the past.
4. If A is projectable to the future, $\oplus A$ and $\mathtt{G}A$ are also projectable to the future, $\neg A$ is projectable to the past and $\mathtt{F}A$ is atemporal.
5. If A is projectable to the past, $\oplus A$ and $\mathtt{F}A$ are also projectable to the past, $\neg A$ is projectable to the future and $\mathtt{G}A$ is atemporal.
6. If A and B are projectable to the future/past, so are $A \vee B$ and $A \wedge B$.

Proposition 3 ensures the distributivity of $\oplus$ and $\mathtt{F}$ over $\vee$, and the distributivity of $\oplus$ and $\mathtt{G}$ over $\wedge$. However, we cannot ensure neither the distributivity of $\mathtt{F}$ over $\wedge$ nor the distributivity of $\mathtt{G}$ over $\vee$, as the following example illustrates.

▶ **Example 10.** The formula $\mathtt{G}(p \vee q)$ is not equivalent to $\mathtt{G}p \vee \mathtt{G}q$ because, for example, given an interpretation such that $h(p) = \{t \in \mathbb{Z} \mid t \text{ is odd}\}$ and $h(q) = \{t \in \mathbb{Z} \mid t \text{ is even}\}$, we have that $0 \models \mathtt{G}(p \vee q)$ but $0 \not\models \mathtt{G}p \vee \mathtt{G}q$.

Counterexamples for the distributivity of $\mathtt{F}$ respect to $\wedge$ can be obtained by duality.

In the following proposition, we give the conditions in which $\mathtt{G}$ distributes respect to $\vee$. The result of the distributivity $\mathtt{F}$ respect to $\wedge$ is obtained by duality.

▶ **Proposition 11.** *Let $\{A_i \mid i \in I\}$ be a finite set of wffs in FNext. Then, the following equivalences hold:*
1. *If A_i is projectable to the future, for all $i \in I$, then $\mathtt{G}\big(\bigvee_{i\in I} A_i\big) \equiv \bigvee_{i\in I} \mathtt{G}A_i$.*
2. *If A_i is projectable to the past, for all $i \in I$, then $\mathtt{G}\big(\bigvee_{i\in I} A_i\big) \equiv \bigvee_{i\in I} \mathtt{G}A_i$.*
3. *If $J = \{i \in I \mid A_i \text{ is atemporal}\}$ then $\mathtt{G}\big(\bigvee_{i\in I} A_i\big) \equiv \bigvee_{i\in J} A_i \vee \mathtt{G}\big(\bigvee_{i\in I\smallsetminus J} A_i\big)$.*

Proof. We shall prove the item 1. The proof of item 2 can be obtained in a similar way.

Let $t \in h\big(\mathtt{G}\big(\bigvee_{i\in I} A_i\big)\big)$. Then $(t,\infty) \subseteq h(\bigvee_{i\in I} A_i) = \bigcup_{i\in I} h(A_i)$. Thus, there exists $i \in I$ such that $t+1 \in h(A_i)$. Since A_i is projectable to the future, then $(t,\infty) \subseteq h(A_i)$ and therefore $t \in h(\mathtt{G}A_i) \subseteq \bigcup_{i\in I} h(\mathtt{G}A_i)$. Thus, $t \in h\big(\bigvee_{i\in I} \mathtt{G}A_i\big)$.

Conversely, let $t \in h\big(\bigvee_{i\in I} \mathtt{G}A_i\big)$. Then $t \in \bigcup_{i\in I} h(\mathtt{G}A_i)$. Thus, there exists $i \in I$ such that $t \in h(\mathtt{G}A_i)$ and therefore $(t,\infty) \subseteq h(A_i) \subseteq \bigcup_{i\in I} h(A_i)$. Then, $t \in h\big(\mathtt{G}\big(\bigvee_{i\in I} A_i\big)\big)$.

For item 3, let $t \in h\big(\mathtt{G}\big(\bigvee_{i\in I} A_i\big)\big)$. Then, $(t,\infty) \subseteq h(\bigvee_{i\in I} A_i) = \bigcup_{i\in I} h(A_i)$. Therefore, there exists $i \in I$ such that $t+1 \in h(A_i)$. There are two cases: if $i \in I$ also belongs to J, then A_i with $i \in J$ is atemporal, and $h(A_i) = \mathbb{Z}$. Therefore, $t \in h(A_i) \subseteq \bigcup_{i\in J} h(A_i)$. Thus, $t \in h(\bigvee_{i\in J} A_i)$. The other case is if $i \in I \smallsetminus J$ then $t \in h\big(\mathtt{G}\big(\bigvee_{i\in I\smallsetminus J} A_i\big)\big)$. So, $t \in h\big(\bigvee_{i\in J} A_i \vee \mathtt{G}\big(\bigvee_{i\in I\smallsetminus J} A_i\big)\big)$.

Conversely, let $t \in h\big(\bigvee_{i\in J} A_i \vee \mathtt{G}\big(\bigvee_{i\in I\smallsetminus J} A_i\big)\big)$. Then, $t \in h\big(\bigvee_{i\in J} A_i\big)$ or $t \in h\big(\mathtt{G}\big(\bigvee_{i\in I\smallsetminus J} A_i\big)\big)$. Then, there exists $i \in J$ such that $t \in h(A_i)$. Since A_i is atemporal, $(t,\infty) \subseteq h(A_i) \subseteq \bigcup_{i\in J} h(A_i)$. Thus, $t \in h\big(\mathtt{G}\bigvee_{i\in J} A_i\big)$. Therefore, $t \in h\big(\mathtt{G}\big(\bigvee_{i\in I} A_i\big)\big)$. ◀

3 Stating the problem

Classical induction in temporal logic can be expressed in FNext by the following equivalence $\oplus A \wedge \mathtt{G}(A \to \oplus A) \equiv \mathtt{G}A$. However, induction can be easily extended by considering formulas as $\oplus^n A \wedge \mathtt{G}(A \to \oplus^m A)$. Each model of these formulas has an infinite sequence of instants in which A is true.

Following with this idea to generalize the induction, we consider formulas $A \wedge \mathtt{G}(B \to C)$ where $A \models \mathtt{F}B$ and $C \models \mathtt{F}B$. These formulas satisfy that, for every model $h \models A \wedge \mathtt{G}(B \to C)$, the set $h(C)$ is infinite (an infinite sequence of instants in which C is true exists).

Looking for a balance between the expressiveness and the complexity of management, we are interested in formulas in which A, B and C are literals. That is, we consider induction schemes like $\ell_1 \wedge \mathtt{G}(\ell_2 \to \ell_3) \equiv \ell_1 \wedge \mathtt{G}(\neg\ell_2 \vee \ell_3)$. Due to the structure of the set of literals, to ensure $\ell_1 \models \mathtt{F}\ell_2$ and $\ell_3 \models \mathtt{F}\ell_2$, it is necessary that $\ell_1, \ell_2, \ell_3 \in Lit(\ell_p)$ for a classical literal ℓ_p.

For some of these formulas, it is possible to find equivalent formulas in which temporal connectives appear only in the literals. For example,

- $\oplus^2 p \wedge \mathtt{G}(\neg p \vee \oplus p) \equiv \mathtt{G} \oplus p$.
- $\mathtt{G} \oplus^4 p \wedge \mathtt{G}(\neg \oplus^2 p \vee \oplus^3 p) \equiv \mathtt{G} \oplus^4 p \wedge (\neg \oplus^3 p \vee \oplus^4 p)$

However, there exist formulas in which it is not possible, such as $\mathtt{F} \oplus^5 p \wedge \mathtt{G}(\neg \oplus^2 p \vee \oplus^4 p)$. The aim of this paper is to distinguish those formulas that can be simplified from those that are not.

To study the induction schemes, $\ell_1 \wedge \mathtt{G}(\ell_2 \to \ell_3) \equiv \ell_1 \wedge \mathtt{G}(\neg\ell_2 \vee \ell_3)$ such that $\ell_1 \models \mathtt{F}\ell_2$ and $\ell_3 \models \mathtt{F}\ell_2$, we characterize those formulas as $\mathtt{G}(\neg\ell_2 \vee \ell_3)$ for which no simplest equivalent formulas exist. These formulas are going to be named irreducible 2-G-clauses. Observe that, in particular, we are interested in those irreducible 2-G-clauses satisfying $\ell_3 \models \mathtt{F}\ell_2$.

4 Irreducible 2-G-Clauses on p

In this section we center on the first step. That is, formulas $\mathtt{G}(\ell_1 \vee \ell_2)$ that cannot be simplified are going to be characterized. We begin by studying clauses that can be simplified to a literal (i.e. there exists a literal that is equivalent to the clause). Obviously, there are clauses where it is not possible.

It is not difficult to see that, if a clause $\ell_1 \vee \ell_2$ is equivalent to a literal ℓ, then $\sup\{\ell_1, \ell_2\} = \ell$ in the lattice $(Lit, \models)$ by definition of the order relation. For example, $\sup\{\oplus\ell_p, \mathtt{F}\oplus\ell_p\} = \mathtt{F}\ell_p$ and $\mathtt{F}\ell_p \equiv \oplus\ell_p \vee \mathtt{F} \oplus \ell_p$. However, the converse is not true, e.g. $\sup\{\oplus\ell_p, \oplus^2\ell_p\} = \mathtt{F}\ell_p$ but $\mathtt{F}\ell_p \not\equiv \oplus\ell_p \vee \oplus^2\ell_p$.

Now, we characterize the disjunctions of two literals such that there does not exist an equivalent literal.

▶ **Definition 12.** *Let $\ell_1, \ell_2 \in Lit(p) \cup Lit(\neg p)$.*

- *The wff $\ell_1 \vee \ell_2$ is named a* 2-clause on p.
- *$\ell_1 \vee \ell_2$ is said* reducible *if there exists $\ell \in Lit$ such that $\ell_1 \vee \ell_2 \equiv \ell$.*
- *$2\text{-}Cla_{irr}(p)$ is the set of irreducible 2-clauses on p.*

The following remark, which provides the cases where a 2-clause can be reduced, is simply a matter of computation from the semantics of $FNext$.

▶ Remark 13. Analyzing exhaustively all the possible pairs of literals and the lattice shown in Figure 1 it is easy to conclude that the cases in which a 2-clause is reducible are the following:

- If $\ell_1, \ell_2 \in Lit(\ell_p)$ with $\ell_1 \models \ell_2$, then $\ell_1 \vee \ell_2 \equiv \ell_2$.
- If $\ell_1 \in Lit(\ell_p)$ and $\ell_2 \in Lit(\neg\ell_p)$ with $\neg\ell_1 \models \ell_2$, then $\ell_1 \vee \ell_2 \equiv \top$.
- $\oplus^{k+1}\ell_p \vee \mathtt{F} \oplus^{k+1} \ell_p \equiv \mathtt{F} \oplus^k \ell_p$ for all $k \in \mathbb{N}$.

To give a characterization of irreducible 2-clauses on p, we introduce the following definitions

▶ **Definition 14.** *Let $\gamma_1, \ldots, \gamma_n \in \{F, G, \oplus\}$ and $\ell_p \in \{p, \neg p\}$. The opposite of the wff $\ell = \gamma_1, \ldots, \gamma_n\ell_p$ is defined as $\overline{\ell} = \overline{\gamma_1}, \ldots, \overline{\gamma_n}\overline{\ell_p}$ where: $\overline{F} = G$, $\overline{G} = F$, $\overline{\oplus} = \oplus$, $\overline{p} = \neg p$ and $\overline{\neg p} = p$.*

The following proposition follows from Remark 13

▶ **Proposition 15** (Characterization of irreducible 2-clauses on p). *The elements of* $2\text{-}Cla_{irr}(p)$ *are the following:*
1. $\oplus^n \ell_p \vee \oplus^m \ell_p$, *for all* $m, n \in \mathbb{N}$ *with* $m \neq n$ *and* $\ell_p \in \{p, \overline{p}\}$.
2. $\oplus^n \ell_p \vee \ell$, *for all* $n \in \mathbb{N}$, $\ell_p \in \{p, \overline{p}\}$ *and for all* $\ell \in \{\mathtt{F} \oplus^m \ell_p \mid m > n\} \cup \{\mathtt{GF}\ell_p, \mathtt{FG}\ell_p\} \cup \{\mathtt{G} \oplus^m \ell_p \mid m \geq n\}$
3. $\ell_1 \vee \ell_2$, *where* $\ell_1 \in Lit(\ell_p)$, $\ell_2 \in Lit(\overline{\ell_p})$ *such that* $\overline{\ell_1} \not\models \ell_2$.

We have just characterized irreducible 2-clauses on p and now we properly center on studying formulas $\mathtt{G}(\ell_1 \vee \ell_2)$ that cannot be simplified.

▶ **Definition 16.** *Let* p *be a propositional variable. Then*
- *A* 2-G-clause on p *is a wff* $\mathtt{G}A$ *where* $A \in 2\text{-}Cla_{irr}(p)$.
- *A* 2-G-clause on p, $\mathtt{G}A$, *is said to be* reducible *if there exists* $\ell \in Lit$ *such that* $\mathtt{G}A \equiv \ell$ *or there exists* $B \in 2\text{-}Cla_{irr}(p)$ *such that* $\mathtt{G}A \equiv B$.
- $2\text{-}\mathtt{G}\text{-}Cla_{irr}(p)$ *is the set of irreducible 2-G-clauses on* p.

With the idea of simplifying formulas in induction schemes, we are interested in the reduction from 2-G-clauses to 2-clauses or literals, when it is possible. Thus, the equivalences provided in Proposition 2 and Proposition 11 are to be read from left to right. For example, since $\mathtt{G}(\mathtt{F}p \vee \mathtt{F} \oplus \overline{p}) \equiv \mathtt{GF}p \vee \mathtt{GF} \oplus \overline{p}$ because of Proposition 11 (both disjuncts are past projectable) and, by Proposition 2, $\mathtt{GF} \oplus \overline{p} \equiv \mathtt{GF}\overline{p}$, then $\mathtt{G}(\mathtt{F}p \vee \mathtt{F} \oplus \overline{p})$ can be reduced to $\mathtt{GF}p \vee \mathtt{GF}\overline{p}$. Moreover, $\mathtt{GF}p \vee \mathtt{GF}\overline{p} \equiv \top$ by Remark 13

Now, to give a characterization of irreducible 2-G-clauses on p, we introduce new reduction laws for the 2-G-clauses on p. The results will be established (if it is possible) in their broader extent.

▶ **Proposition 17.** *Let* $A \in \mathcal{L}$ *and* $n, m \in \mathbb{N}$. *The following equivalences hold:*
1. $\mathtt{G}(\mathtt{F} \oplus^n A \vee \oplus^m A) \equiv \mathtt{GF}A$.
2. *If* $n > m$ *then*

$$\mathtt{G}(\mathtt{F} \oplus^n A \vee \mathtt{G} \oplus^m \neg A) \equiv \mathtt{GF}A \vee \mathtt{G} \oplus^{m+1} \neg A \equiv \mathtt{G}(\mathtt{F} \oplus^n A \vee \oplus^{m+1} \neg A)$$

Observe that, item 2 in proposition above only considers the case in which $n > m$ because otherwise $\mathtt{G}(\mathtt{F} \oplus^n A \vee \mathtt{G} \oplus^m \neg A)$ and $\mathtt{G}(\mathtt{F} \oplus^n A \vee \oplus^{m+1} \neg A)$ are not 2-G-clauses. In fact, when $n \leq m$, $\mathtt{F} \oplus^n A \vee \mathtt{G} \oplus^m \neg A \equiv \mathtt{F} \oplus^n A \vee \oplus^{m+1} \neg A \equiv \top$.

Proof. For item 1, it is a trivial task to check that $\mathtt{F} \oplus^n A \vee \oplus^m A \models \mathtt{F}A$ and therefore, by monotonicity of G, we have that $\mathtt{G}(\mathtt{F} \oplus^n A \vee \oplus^m A) \models \mathtt{GF}A$. Conversely, $\mathtt{GF}A \equiv \mathtt{GF} \oplus^n A$ by Proposition 2, and $\mathtt{GF}A \models \mathtt{G}(\mathtt{F} \oplus^n A \vee \oplus^m A)$.

For the equivalence $\mathtt{G}(\mathtt{F} \oplus^n A \vee \mathtt{G} \oplus^m \neg A) \equiv \mathtt{GF}A \vee \mathtt{G} \oplus^{m+1} \neg A$ in item 2, it is trivial that $\mathtt{GF}A \vee \mathtt{G} \oplus^{m+1} \neg A \models \mathtt{G}(\mathtt{F} \oplus^n A \vee \mathtt{G} \oplus^m \neg A)$. Conversely, assume $n > m$ and $t \in \mathtt{G}(\mathtt{F} \oplus^n A \vee \mathtt{G} \oplus^m \neg A)$. We can distinguish two cases:
- If $(t, \infty) \subseteq h(\mathtt{F} \oplus^n A)$, then $t \in h(\mathtt{GF} \oplus^n A) = h(\mathtt{GF}A) \subseteq h(\mathtt{GF}A \vee \mathtt{G} \oplus^{m+1} \neg A)$.
- Otherwise, there are instants greater than t that do not belong to $h(\mathtt{F} \oplus^n A)$. Then, let t_0 be the smallest of them, i.e. $t_0 = min(\,(t, \infty) \cap h(\mathtt{G} \oplus^m \neg A)\,)$. Thus, $t_0 + m \in h(\mathtt{G}\neg A)$ ($\dagger_2$). By definition of t_0, we have $t_0 - 1 \in h(\mathtt{F} \oplus^n A)$ ($\dagger_1$), i.e. $t_0 - 1 + n \in h(\mathtt{F}A)$, and since $n > m$ we get $t_0 - 1 + n \geq t_0 + m$ and so, we should have that ($\dagger_2$) contradicts to ($\dagger_1$), which is not possible except that $t_0 = t + 1$, in which case $t \in h(\mathtt{G} \oplus^{m+1} \neg A) \subseteq h(\mathtt{GF}A \vee \mathtt{G} \oplus^{m+1} \neg A)$.

Finally, to prove $\mathtt{GF}A \vee \mathtt{G} \oplus^{m+1} \neg A \equiv \mathtt{G}(\mathtt{F} \oplus^n A \vee \oplus^{m+1} \neg A)$ when $n > m$, we have that $\mathtt{GF}A \vee \mathtt{G} \oplus^{m+1} \neg A \models \mathtt{G}(\mathtt{F} \oplus^n A \vee \oplus^{m+1} \neg A)$ is trivial.

Conversely, suppose $t \in h(\mathtt{G}(\mathtt{F} \oplus^n A \vee \oplus^{m+1} \neg A))$ ($\dagger$). We have two cases:

- If $[t+1,\infty) \subseteq h(\oplus^{m+1}\neg A)$, then clearly we obtain $t \in h(\mathtt{GF}A \vee \mathtt{G} \oplus^{m+1} \neg A)$.
- Otherwise, $\Lambda = [t+1,\infty) \cap h(\oplus^{m+1}A) \neq \varnothing$. We shall prove that Λ is infinite, and therefore $t \in h(\mathtt{GF}A) \subseteq h(\mathtt{GF}A \vee \mathtt{G} \oplus^{m+1} \neg A)$. If Λ is finite, consider $t_0 = max\Lambda$. Then $[t_0+1,\infty) \subseteq h(\oplus^{m+1}\neg A)$ (††). On the other hand, $t_0 \in h(\oplus^{m+1}A)$, so by the hypothesis (†) we get $t_0 \in h(\mathtt{F} \oplus^n A)$, that is, $(t_0+1,\infty) \cap h(\oplus^{m+1}A) \neq \varnothing$ (since $n > m$), in contradiction with (††). ◀

In Table 1, we collect the laws previously obtained, which are denoted as $\mathtt{Red}_i$. These laws are going to be considered as rewriting rules to transform 2-G-clauses into 2-clauses and are always read from left to right.

Table 1 G-Reduction Laws.

Law	Name
$\mathtt{G}(A \vee B) \equiv \mathtt{G}A \vee \mathtt{G}B$, if A and B are past projectable, or A and B are future projectable	$\mathtt{Red}_1$
$\mathtt{G}(A \vee B) \equiv \mathtt{G}A \vee B$, if B is atemporal	$\mathtt{Red}_2$
$\mathtt{G}(\mathtt{F} \oplus^n A \vee \oplus^m A) \equiv \mathtt{GF}A$	$\mathtt{Red}_3$
$\mathtt{G}(\mathtt{F} \oplus^n A \vee \mathtt{G} \oplus^m \neg A) \equiv \mathtt{GF}A \vee \mathtt{G} \oplus^{m+1} \neg A$, if $n > m$	$\mathtt{Red}_4$
$\mathtt{G}(\mathtt{F} \oplus^n A \vee \oplus^{m+1} \neg A) \equiv \mathtt{GF}A \vee \mathtt{G} \oplus^{m+1} \neg A$, if $n > m$	$\mathtt{Red}_5$

Reductions $\mathtt{Red}_1$ and $\mathtt{Red}_2$ are due to Proposition 11. On the other hand, $\mathtt{Red}_3$, $\mathtt{Red}_4$ and $\mathtt{Red}_5$ will be applied only in 2-G-clauses and hence, reductions correspond with the equivalences given in Proposition 17.

These equivalences are those that ensure a 2-G-clause on p is irreducible only in the case that none of the reductions $\mathtt{Red}_i$ with $1 \leq i \leq 5$ is applicable to it. The following characterization theorem for irreducible 2-G-clauses on p is a consequence of this assertion.

▶ **Theorem 18** (Characterization of Irreducible 2-G-clauses on p)**.** *The elements of the set 2-G-$Cla_{irr}(p)$ are the following:*

(a) $\mathtt{G}(\oplus^n \ell_p \vee \oplus^m \ell_p)$*, if* $n \neq m$
(b) $\mathtt{G}(\oplus^n \ell_p \vee \oplus^m \overline{\ell}_p)$*, if* $n \neq m$
(c) $\mathtt{G}(\oplus^n \ell_p \vee \mathtt{G} \oplus^m \ell_p)$*, if* $n \leq m$
(d) $\mathtt{G}(\oplus^n \ell_p \vee \mathtt{G} \oplus^m \overline{\ell}_p)$

Proof. The proof is divided in two parts. In the first, we discard such 2-G-clauses on p that previous results ensure are reducible. Thus, we obtain that formulas labeled *a),b),c),d)* in the theorem are those that cannot be reduced through previous results. In the second part, we prove that they are really irreducible.

From Definition 16 and Proposition 15, a 2-G-clause on p is a formula $\mathtt{G}A$ such that A is of one of the following forms:

(i) $\oplus^n \ell_p \vee \oplus^m \ell_p$, with $m, n \in \mathbb{N}$ and $m \neq n$.
(ii) $\oplus^n \ell_p \vee \ell$, where $\ell \in \{\mathtt{F} \oplus^m \ell_p \mid m > n\} \cup \{\mathtt{GF}\ell_p, \mathtt{FG}\ell_p\} \cup \{\mathtt{G} \oplus^m \ell_p \mid m \geq n\}$
(iii) $\ell_1 \vee \ell_2$, where $\ell_1 \in Lit(\ell_p)$, $\ell_2 \in Lit(\overline{\ell_p})$ and $\overline{\ell_1} \not\models \ell_2$.

If $\mathtt{G}A$ is a 2-G-clause such that A is a clause of the type described in item i) then $\mathtt{G}A$ cannot be reduced through previous results (item *a)* of the theorem).

Now, let $\mathtt{G}A$ be 2-$\mathtt{G}$-clause such that A is a clause of the type described in item ii). Then, we discard such 2-$\mathtt{G}$-clauses on p which previous results ensured are reducible.

- if $\ell \in \{\mathtt{GF}\ell_p, \mathtt{FG}\ell_p\}$ then from Proposition 11 (reduction $\mathtt{Red}_2$), $\mathtt{G}A$ is reducible, and
- if $\ell \in \{\mathtt{F} \oplus^m \ell_p \mid m > n\}$ then from Proposition 17 (reduction $\mathtt{Red}_3$), $\mathtt{G}A$ is reducible.
- if $\ell \in \{\mathtt{G} \oplus^m \ell_p \mid m \geq n\}$ then $\mathtt{G}A$ cannot be reduced through previous results (item *c)* of the theorem).

For item iii) we have to analyze the 11 elements stated in the following table:

(1)	$\mathtt{G}(\oplus^n\ell_p \vee \oplus^m\overline{\ell}_p)$ $n \neq m$	(2)	$\mathtt{G}(\oplus^n\ell_p \vee \mathtt{G} \oplus^m \overline{\ell}_p)$	(3)	$\mathtt{G}(\oplus^n\ell_p \vee \mathtt{F} \oplus^m \overline{\ell}_p)$
(4)	$\mathtt{G}(\oplus^n\ell_p \vee \mathtt{GF}\overline{\ell}_p)$	(5)	$\mathtt{G}(\oplus^n\ell_p \vee \mathtt{FG}\overline{\ell}_p)$	(6)	$\mathtt{G}(\mathtt{G} \oplus^n \ell_p \vee \mathtt{G} \oplus^m \overline{\ell}_p)$
(7)	$\mathtt{G}(\mathtt{G} \oplus^n \ell_p \vee \mathtt{F} \oplus^m \overline{\ell}_p)$	(8)	$\mathtt{G}(\mathtt{G} \oplus^n \ell_p \vee \mathtt{GF}\overline{\ell}_p)$	(9)	$\mathtt{G}(\mathtt{G} \oplus^n \ell_p \vee \mathtt{FG}\overline{\ell}_p)$
(10)	$\mathtt{G}(\mathtt{F} \oplus^n \ell_p \vee \mathtt{F} \oplus^m \overline{\ell}_p)$	(11)	$\mathtt{G}(\mathtt{FG}\ell_p \vee \mathtt{FG}\overline{\ell}_p)$		

In the previous table, from Proposition 11 (reduction $\mathtt{Red}_1$) items (6), (8), (9), (10) and (11) are eliminated (in each of them the two literals of 2-clauses are past projectable or the two literals of 2-clauses are future projectable).

From Proposition 11 (reduction $\mathtt{Red}_2$) items (4) and (5) are eliminated because in each of them the 2-clauses have atemporal literals.

From Proposition 17 item (7) is eliminated (reduction $\mathtt{Red}_4$) and item (3) is eliminated (reduction $\mathtt{Red}_5$).

Finally, items (1) and (2) cannot be removed because these formulas cannot be reduced by previous results of the theorem: (items *b)* and *d)* respectively).

At this moment, we have proved that 2-$\mathtt{G}$-clauses on p different to those that are described in the theorem can be reduced by using previous results. Now, we focus on proving that formulas $\mathtt{G}A$ described in items *a)*, *b)*, *c)* and *d)* are really irreducible. Thus, we prove that neither $\mathtt{G}A \not\equiv \ell$ for a literal ℓ, nor $\mathtt{G}A \not\equiv \ell_1 \vee \ell_2$ for $\ell_1 \vee \ell_2 \in 2\text{-}Cla_{irr}(p)$. Note that, collecting literals and clauses, there are 40 formulas that could be equivalent to $\mathtt{G}A$. Obviously, $\top$ and $\bot$ are discarded.

First, we prove $\mathtt{G}A$ is not equivalent to any of the following formulas: $\oplus^k\ell_p$, $\mathtt{F} \oplus^k \ell_p$, $\mathtt{GF}\ell_p$. Consider $k \in \mathbb{N}$ and $r = \max\{k+1, n+1, m+1\}$. For any interpretation h such that $h(\ell_p) = \bigcup_{i \in \mathbb{N}}[2ir, 2ir+1)$, we have that $0 \in h(\oplus^k\ell_p)$, $0 \in h(\mathtt{F} \oplus^k \ell_p)$ and $0 \in h(\mathtt{GF}\ell_p)$ but $0 \notin h(\mathtt{G}A)$. Therefore, $\oplus^k\ell_p \not\models \mathtt{G}A$, $\mathtt{F} \oplus^k \ell_p \not\models \mathtt{G}A$ and $\mathtt{GF}\ell_p \not\models \mathtt{G}A$.

With a similar reasoning, it can be proved that $\oplus^k\overline{\ell_p} \not\models \mathtt{G}A$, $\mathtt{F}\oplus^k \overline{\ell_p} \not\models \mathtt{G}A$ and $\mathtt{GF}\overline{\ell_p} \not\models \mathtt{G}A$. Moreover, any 2-clause $\ell_1 \vee \ell_2$ with ℓ_1 or ℓ_2 being one of the previous cases satisfies $\ell_1 \vee \ell_2 \not\models \mathtt{G}A$.

The rest of the formulas are:

$\mathtt{G} \oplus^k \ell_p$	$\mathtt{G} \oplus^k \overline{\ell_p}$	$\mathtt{FG}\ell_p$	$\mathtt{FG}\overline{\ell_p}$
$\mathtt{G} \oplus^{k_1} \ell_p \vee \mathtt{G} \oplus^{k_2} \overline{\ell_p}$	$\mathtt{G} \oplus^k \ell_p \vee \mathtt{FG}\overline{\ell_p}$	$\mathtt{FG}\ell_p \vee \mathtt{FG}\overline{\ell_p}$	

Now we prove that none of the formulas described in the theorem is equivalent to any of them.

(a) Let us consider h with $h(\ell_p) = \mathbb{Z} \smallsetminus \{(|m-n|+1)^r \mid r \in \mathbb{N} \smallsetminus \{0\}\}$, where $|m-n|$ denotes the absolute value of $m-n$. It is just a matter of computation to check that $0 \in h(\mathtt{G}(\oplus^n\ell_p \vee \oplus^m\ell_p))$ but it is not a model for formulas in the previous table.

(b) Let us suppose, without loss of generality, that $n < m$ and consider h with $h(\ell_p) = \bigcup_{i \in \mathbb{N}}[n + 2(m-n)i, m + 2(m-n)i)$. In this case, $0 \in h(\mathtt{G}(\oplus^n\ell_p \vee \oplus^m\overline{\ell_p}))$ but is not model for formulas in the previous table.

(c) For $\mathtt{G}(\oplus^n\ell_p \vee \mathtt{G} \oplus^m \ell_p)$ when $n \leq m$, the following cases are going to be considered:

- Any interpretation h such that $h(\ell_p) = [k+1, \infty)$ satisfies $0 \in h(\mathtt{FG}\ell_p)$ and $0 \in h(\mathtt{G} \oplus^k \ell_p)$. However, $0 \notin h(\mathtt{G}(\oplus^n \ell_p \vee \mathtt{G} \oplus^m \ell_p))$ when $m < k$. Therefore, $\mathtt{FG}\ell_p \not\models \mathtt{G}(\oplus^n \ell_p \vee \mathtt{G} \oplus^m \ell_p)$ and, when $m < k$, we have that $\mathtt{G} \oplus^k \ell_p \not\models \mathtt{G}(\oplus^n \ell_p \vee \mathtt{G} \oplus^m \ell_p)$.
- For any interpretation such that $0 \in h(\mathtt{G} \oplus^k \overline{\ell_p})$ satisfies $0 \in h(\mathtt{FG}\overline{\ell_p})$, but $0 \notin h(\mathtt{G}(\oplus^n \ell_p \vee \mathtt{G} \oplus^m \ell_p))$.
- If $\{\ell_1, \ell_2\} \cap \big(\{\mathtt{FG}\ell_p, \mathtt{FG}\overline{\ell_p}\} \cup \{\mathtt{G} \oplus^k \ell_p \mid m < k\} \cup \{\mathtt{G} \oplus^k \overline{\ell_p} \mid k \in \mathbb{N}\}\big) \neq \varnothing$ then $\ell_1 \vee \ell_2 \not\models \mathtt{G}(\oplus^n \ell_p \vee \mathtt{G} \oplus^m \ell_p)$.
- Finally, $\mathtt{G} \oplus^k \ell_p$ with $k \leq m$ is the last formula to be considered. In this case, any interpretation h such that $h(\ell_p) = \mathbb{Z} \smallsetminus \{k+2\}$ satisfies that $0 \in h(\mathtt{G}(\oplus^n \ell_p \vee \mathtt{G} \oplus^m \ell_p))$ but $0 \notin h(\mathtt{G} \oplus^k \ell_p)$.

(d) With a similar reasoning to the above, it can be proved that the formula $\mathtt{G}(\oplus^n \ell_p \vee \mathtt{G} \oplus^m \overline{\ell_p})$ is not equivalent to any formula of the last table. ◀

5 Inductive Schemes

Observe that, from the kind of irreducible 2-$\mathtt{G}$-clauses $\mathtt{G}(\neg\ell_1 \vee \ell_2)$ characterized in Theorem 18, only the items b) and d) could satisfy condition $\ell_2 \models \mathtt{F}\ell_1$. The following definition sums up all the conditions step by step previously introduced relative to inductive schemes.

▶ **Definition 19.** *An* `inductive scheme on` ℓ_p *is a formula* $\ell_1 \wedge \mathtt{G}(\bar{\ell}_2 \vee \ell_3)$*, where* $\ell_1, \ell_2, \ell_3 \in Lit(\ell_p)$ *which satisfies the following three conditions:*

Ind-1: $\mathtt{G}(\bar{\ell}_2 \vee \ell_3) \in 2\text{-}\mathtt{G}\text{-}Cla_{irr}(p)$*,*

Ind-2: $\ell_1 \models \mathtt{F}\ell_2$ *and* $\ell_3 \models \mathtt{F}\ell_2$*.*

▶ **Example 20.** $\mathtt{F} \oplus^5 p \wedge \mathtt{G}(\neg \oplus^2 p \vee \oplus^4 p)$ is an inductive scheme on p.

We introduce the main result that characterises those inductive formulas that cannot be simplified. Specifically, among of the 125 kind of formulas $\ell_1 \wedge \mathtt{G}(\bar{\ell}_2 \vee \ell_3)$ (i.e., 5^3, corresponding to non-trivial possible selections ℓ_1, ℓ_2 and ℓ_3 from the types of literals $\oplus^k \ell_p$, $\mathtt{F} \oplus^k \ell_p$, $\mathtt{G} \oplus^k \ell_p$, $\mathtt{FG}\ell_p$ and $\mathtt{GF}\ell_p$), only 15 satisfy `Ind-1`. Condition `Ind-2` only imposes restrictions relative to the super-index of the $\oplus$ connectives but the number of kind of formulas remains at being 15, which are enumerated in Table 2.

6 Conclusions and future works

In the framework of the propositional linear discrete temporal logic $FNext$, we have studied those formulas with a single propositional variable that involve the well-known idea of induction. Indeed, we have classified these formulas into those that can be expressed by equivalent ones without loops and the rest (inductive formulas). The main issue of this paper has been to study the set of inductive formulas in order to determine schemes that characterize them.

The starting point is the set of expressions $\ell_1 \wedge \mathtt{G}(\bar{\ell}_2 \vee \ell_3)$ which includes 125 kinds of formulas (i.e., 5^3, corresponding to non-trivial possible selections ℓ_1, ℓ_2 and ℓ_3 from the types of literals $\oplus^k \ell_p$, $\mathtt{F} \oplus^k \ell_p$, $\mathtt{G} \oplus^k \ell_p$, $\mathtt{FG}\ell_p$ and $\mathtt{GF}\ell_p$).

In Section 4, we have characterized those formulas as $\mathtt{G}(\bar{\ell}_2 \vee \ell_3)$ for which no simplest equivalent formulas exist. These formulas have been named irreducible 2-$\mathtt{G}$-clauses. Specifically, Theorem 18 provides four schemes that cover all the irreducible 2-$\mathtt{G}$-clauses and allows us to reduce the initial number to 15 kinds of formulas.

Table 2 Formulas $\ell_1 \wedge \mathtt{G}(\overline{\ell}_2 \vee \ell_3)$ satisfying `Ind-1` and `Ind-2`.

1. $\oplus^{n_1} \ell_p \wedge \mathtt{G}(\oplus^{n_2}\overline{\ell}_p \vee \oplus^{n_3}\ell_p)$, where $n_1, n_3 > n_2$	9. $\mathtt{FG}\ell_p \wedge \mathtt{G}(\oplus^{n_2}\overline{\ell}_p \vee \mathtt{G}\oplus^{n_3} \ell_p)$
2. $\mathtt{F}\oplus^{n_1} \ell_p \wedge \mathtt{G}(\oplus^{n_2}\overline{\ell}_p \vee \oplus^{n_3}\ell_p)$, where $n_1 \geq n_2$ and $n_3 > n_2$	10. $\mathtt{G}\oplus^{n_1} \ell_p \wedge \mathtt{G}(\oplus^{n_2}\overline{\ell}_p \vee \mathtt{G}\oplus^{n_3} \ell_p)$, where $n_1, n_3 \geq n_2$
3. $\mathtt{FG}\ell_p \wedge \mathtt{G}(\oplus^{n_2}\overline{\ell}_p \vee \oplus^{n_3}\ell_p)$, where $n_3 > n_2$	11. $\oplus^{n_1} \ell_p \wedge \mathtt{G}(\mathtt{G}\oplus^{n_2} \overline{\ell}_p \vee \oplus^{n_3}\ell_p)$, where $n_1, n_3 > n_2 + 1$
4. $\mathtt{F}\oplus^{n_1} \ell_p \wedge \mathtt{G}(\oplus^{n_2}\overline{\ell}_p \vee \mathtt{G}\oplus^{n_3} \ell_p)$, where $n_1 \geq n_2$	12. $\mathtt{F}\oplus^{n_1} \ell_p \wedge \mathtt{G}(\mathtt{G}\oplus^{n_2} \overline{\ell}_p \vee \oplus^{n_3}\ell_p)$, where $n_1 > n_2$ and $n_3 \geq n_2$
5. $\mathtt{GF}\ell_p \wedge \mathtt{G}(\oplus^{n_2}\overline{\ell}_p \vee \oplus^{n_3}\ell_p)$, where $n_3 > n_2$	13. $\mathtt{GF}\ell_p \wedge \mathtt{G}(\mathtt{G}\oplus^{n_2} \overline{\ell}_p \vee \oplus^{n_3}\ell_p)$, where $n_3 > n_2 + 1$
6. $\mathtt{G}\oplus^{n_1} \ell_p \wedge \mathtt{G}(\oplus^{n_2}\overline{\ell}_p \vee \oplus^{n_3}\ell_p)$, where $n_1 \geq n_2$ and $n_3 > n_2$	14. $\mathtt{FG}\ell_p \wedge \mathtt{G}(\mathtt{G}\oplus^{n_2} \overline{\ell}_p \vee \oplus^{n_3}\ell_p)$, where $n_3 > n_2 + 1$
7. $\oplus^{n_1} \ell_p \wedge \mathtt{G}(\oplus^{n_2}\overline{\ell}_p \vee \mathtt{G}\oplus^{n_3} \ell_p)$, where $n_1 > n_2$	15. $\mathtt{G}\oplus^{n_1} \ell_p \wedge \mathtt{G}(\mathtt{G}\oplus^{n_2} \overline{\ell}_p \vee \oplus^{n_3}\ell_p)$, where $n_3 > n_2 + 1$
8. $\mathtt{GF}\ell_p \wedge \mathtt{G}(\oplus^{n_2}\overline{\ell}_p \vee \mathtt{G}\oplus^{n_3} \ell_p)$	

Not all of these formulas correspond with the idea of induction because it is necessary that $\ell_1 \models \mathtt{F}\ell_2$ and $\ell_3 \models \mathtt{F}\ell_2$ (Condition `Ind-2`). This condition only imposes restrictions relative to the super-index. Thus, the number of kinds of formulas remains at 15, which are enumerated in Table 2.

To emphasize the interest of the theoretical results of this paper, as further work, we will study how to improve the definition of Temporal Negative Normal Form for the Temporal Logic introduced in [18]. It will have a significant relevance in the future design of efficient automated theorem provers.

In the midterm, the study of inductive schemes developed in this paper could be extended to a fully expressive temporal logic such as Hans Kamp's US logic or LN logic [4].

References

1 Kyungmin Bae and José Meseguer. Model checking linear temporal logic of rewriting formulas under localized fairness. *Science of Computer Programming*, 99:193–234, 2015. `doi:10.1016/j.scico.2014.02.006`.

2 Philippe Balbiani, Andreas Herzig, François Schwarzentruber, and Nicolas Troquard. DL-PA and DCL-PC: model checking and satisfiability problem are indeed in PSPACE. *CoRR*, abs/1411.7825, 2014. `arXiv:1411.7825`.

3 Nikolaj Bjorner, Anca Browne, Eddie Chang, Michael Colón, Arjun Kapur, Zohar Manna, Henny B. Sipma, and Tomás E. Uribe. STeP: Deductive-algorithmic verification of reactive and real-time systems: Deductive-algorithmic verification of reactive and real-time systems. In Rajeev Alur and Thomas A. Henzinger, editors, *Computer Aided Verification*, volume 1102 of *Lecture Notes in Computer Science*, pages 415–418. Springer Berlin Heidelberg, 1996. `doi:10.1007/3-540-61474-5_92`.

4 Alfredo Burrieza and Inma Pérez de Guzmán. A new algebraic semantic approach and some adequate connectives for computation with temporal logic over discrete time. *Journal of Applied Non-Classical Logics*, 2(2):181–200, 1992. `doi:10.1080/11663081.1992.10510781`.

5 E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic Verification of Finite-state Concurrent Systems Using Temporal Logic Specifications. *ACM Trans. Program. Lang. Syst.*, 8(2):244–263, April 1986. `doi:10.1145/5397.5399`.

6 Pablo Cordero, Manuel Enciso, and Inma Pérez de Guzmán. Bases for closed sets of implicants and implicates in temporal logic. *Acta Inf.*, 38(9):599–619, 2002. `doi:10.1007/s00236-002-0087-2`.

7 Anatoli Degtyarev, Michael Fisher, and Boris Konev. A Simplified Clausal Resolution Procedure for Propositional Linear-Time Temporal Logic. In Uwe Egly and Chritian G. Fermüller, editors, *Automated Reasoning with Analytic Tableaux and Related Methods*, volume 2381 of *Lecture Notes in Computer Science*, pages 85–99. Springer Berlin Heidelberg, 2002. `doi:10.1007/3-540-45616-3_7`.

8 Stéphane Demri and Philippe Schnoebelen. The Complexity of Propositional Linear Temporal Logics in Simple Cases. *Information and Computation*, 174(1):84–103, 2002. `doi:10.1006/inco.2001.3094`.

9 Razvan Diaconescu. Structural induction in institutions. *Information and Computation*, 209(9):1197–1222, 2011. `doi:10.1016/j.ic.2011.06.002`.

10 Stephan Falke and Deepak Kapur. Rewriting Induction + Linear Arithmetic = Decision Procedure. In Bernhard Gramlich, Dale Miller, and Uli Sattler, editors, *Automated Reasoning*, volume 7364 of *Lecture Notes in Computer Science*, pages 241–255. Springer Berlin Heidelberg, 2012. `doi:10.1007/978-3-642-31365-3_20`.

11 Michael Fisher. *An Introduction to Practical Formal Methods Using Temporal Logic*. Wiley, 2011.

12 Michael Fisher, Clare Dixon, and Martin Peim. Clausal Temporal Resolution. *ACM Trans. Comput. Logic*, 2(1):12–56, 2001. `doi:10.1145/371282.371311`.

13 Nicoletta De Francesco, Antonella Santone, and Gigliola Vaglini. A user-friendly interface to specify temporal properties of concurrent systems. *Information Sciences*, 177(1):299–311, 2007. `doi:10.1016/j.ins.2006.03.008`.

14 Richard Jensen, Andrew Tuson, and Qiang Shen. Finding rough and fuzzy-rough set reducts with {SAT}. *Information Sciences*, 255(0):100–120, 2014. `doi:10.1016/j.ins.2013.07.033`.

15 Hans Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, Ucla, 1968.

16 Savas Konur. A survey on temporal logics for specifying and verifying real-time systems. *Frontiers of Computer Science*, 7(3):370–403, 2013. `doi:10.1007/s11704-013-2195-2`.

17 Artur Meski, Wojciech Penczek, and Grzegorz Rozenberg. Model checking temporal properties of reaction systems. *Information Sciences*, 313:22–42, 2015. `doi:10.1016/j.ins.2015.03.048`.

18 Manuel Enciso Pablo Cordero and Inmaculada P. de Guzmán. From the poset of literals to a temporal negative normal form. *Reports on Mathematical Logic*, 36:3–53, 2002.

19 Amir Pnueli. The temporal logic of programs. In *Foundations of Computer Science, 1977., 18th Annual Symposium on*, pages 46–57, 1977. `doi:10.1109/SFCS.1977.32`.

20 José Manuel Rodríguez-Jiménez, Pablo Cordero, Manuel Enciso, and Angel Mora. Negative Attributes and Implications in Formal Concept Analysis. In Fuad Aleskerov, Yong Shi, and Alexander Lepskiy, editors, *Proceedings of the Second International Conference on Information Technology and Quantitative Management, ITQM 2014, National Research University Higher School of Economics (HSE), Moscow, Russia, June 3-5, 2014*, volume 31 of *Procedia Computer Science*, pages 758–765. Elsevier, 2014. `doi:10.1016/j.procs.2014.05.325`.

21 A. P. Sistla and E. M. Clarke. The Complexity of Propositional Linear Temporal Logics. *J. ACM*, 32(3):733–749, 1985. `doi:10.1145/3828.3837`.