ELSEVIER
Theoretical Computer Science 148 (1995) 303–324
Theoretical Computer Science

A hierarchy of temporal logics with past

F. Laroussinie, Ph. Schnoebelen *

LIFIA-IMAG, 46 Av. Félix Viallet, F-38031 Grenoble Cedex, France

Abstract

We extend the classical hierarchy of branching-time temporal logics between *UB* and *CTL** by studying which additional expressive power (if any) stems from the incorporation of past-time modalities. In addition, we propose a new temporal combinator, N for "From Now On", that brings new and interesting expressive power. In several situations, nontrivial translation algorithms exist from a temporal logic with past to a pure-future fragment. These algorithms have important practical applications, e.g., in the field of model-checking.

0. Introduction

Temporal logics have long been recognized as a very convenient formalism with which to reason about concurrent and reactive systems [8, 21]. In computer science, most theoretical studies of temporal logics only use future-time constructs. This is in contrast with the temporal logics studied by linguists, philosophers, ..., where past-time and future-time have been on an equal footing [23].

This situation is surprising because computer scientists recognize that past-time constructs can be very useful when it comes to express certain properties. For example, using "$\Box$" for "at all future moments" and "$\Diamond^{-1}$" for "at some past moment", it is easy to state that "in all cases the occurrence of a problem must have been preceded by a cause", i.e., "no problem will ever occur without a cause", which is an important safety property one often uses (under some form). One just writes:

$$\Box(\texttt{problem} \Rightarrow \Diamond^{-1}\texttt{cause}) \tag{1}$$

Finally, the usefulness of past-time constructs is most apparent in the classification of temporal properties [28, 20, 5].

However, it has been shown that formulas using past-time constructs can often be replaced by equivalent *pure-future* formulas [13, 18]. For example, (1) is equivalent to

$$\neg(\neg\texttt{cause}\ \mathsf{U}\ \texttt{problem}) \tag{2}$$

* Corresponding author Email: {fl,phs}@lifia.imag.fr.

SSDI 0304-3975(95)00035-6

which uses the "Until" construct U. (We state in the next section in which formal sense these two formulas are equivalent.) The underlying motto is that *past-time brings additional expressivity from a practical, but not from a theoretical viewpoint.* Clearly, a formulation like (1) is much more natural than the clumsier (2). This is even more obvious when one tries to express a statement like $\Box(\texttt{problem} \Rightarrow \Diamond^{-1}(\texttt{cause1} \wedge \Diamond^{-1}\texttt{cause2}))$ without past-time.

Another reason why past-time is often omitted in theoretical studies is that very efficient model-checking algorithms exist for state-based logics like *CTL* (with or without fairness) [8, 7], while it is not clear how to adapt this technology to (history-based) logics with past. Some existing results (e.g. [15]) consider model-checking for *PTL* with past, but this problem is already PSPACE-complete for pure-future *PTL* [24].

This raises the following question: "*Is it possible to combine the great convenience of past-time for specification with the efficiency of CTL model-checking for verification*?" Rather than try to adapt the existing technology to, e.g., *CTL* + Past, which we believe is a very difficult problem, we argue that a translation-based approach is feasible [16]. By only requiring the addition of a translating interface, such an approach would allow to reuse the very efficient model-checking tools that have been built after years of improvement [3]. Of course, this approach requires the use of a logic with past that can be translated into, e.g., *CTL*.

When we surveyed the available *past-elimination results* in the literature, we found:

- *PTL* + Past can be translated into *PTL* [13, 12]. This is the standard result in the field.
- The linear-time propositional μ-calculus, $L\mu$ + Past can be translated into the usual pure-future μ-calculus [26]. [1]
- *CTL** + Past can be translated into *CTL** [14]. This is a simple corollary of Gabbay's proof for *PTL*.
- $PTL \setminus \mathsf{X}$ + Past can be translated into $PTL \setminus \mathsf{X}$ [22]. This uses rewrite rules similar to Gabbay's rules.

Finally, apart from [26] all of these (and some more, e.g. [4]) are just variants of Gabbay's result for *PTL*. And they do not solve our problem. For example, if we want to add past-time constructs to a (state-based) branching-time logic-like *CTL*, the literature only tells us how to translate *CTL* + Past into *CTL**. This is not satisfactory, for we consider *CTL* precisely because it admits a very efficient model-checking procedure, while this is not the case with *CTL**. Therefore, knowing that *CTL* + Past can be translated into *CTL* would be, from a practical viewpoint, a very interesting addition to the results we mentioned.

This is exactly what we investigate in this article. We address general questions of the form "*Which past-time combinators can be added to branching-time temporal logics like CTL, ECTL, ... without compromising the possibility to translate back*

[1] In fact, [26] gives a translation from some kind of backward-and-forward Büchi automata into usual Büchi automata, so that one has to translate from the μ-calculus into Büchi automata, and *vice versa*.

into CTL, ECTL, ... ?" We consider the classical branching-time hierarchy from *UB* to *CTL** [10, 11] and systematically try to add past-time constructs.

A second motivation for this study is the introduction of a new temporal combinator, "N" for "From Now On" or "Henceforth". N is very useful in some situations where we want to restrict the scope of past-time combinators. This new combinator can also be eliminated (i.e., translated into pure-future constructs) in some situations.

Here is the plan of the article: we define *PCTL** (*CTL** + Past) in Section 1, and the relevant fragments (*PTL*, *CTL*, ...) in Section 2. Section 3 discusses and motivates *initial equivalence*, the correctness criterion we use for our expressivity problems. Then Sections 4 and 5 state fundamental expressivity results of past-time combinators in branching-time logics. The new "From Now On" combinator is motivated and introduced in Section 6 where our expressivity results are extended. Some proofs have been relegated to an Appendix when they disturb the exposition.

1. Temporal logics with Past

1.1. Syntax

We define *PCTL** (for "*CTL** with Past") as an extension of *CTL** [11] with past-time combinators. (Our definition differs slightly from the *PCTL** used in [14] as we explain later.) We assume a given set $Prop = \{a, b, \ldots, \mathtt{problem}, \mathtt{cause}, \ldots\}$ of *atomic propositions*.

Definition 1.1 (*Syntax of PCTL**). The formulas of *PCTL** are given by the following grammar

$$PCTL^* \ni f, g ::= a \mid f \wedge g \mid \neg f \mid \mathsf{E} f \mid f \,\mathsf{U}\, g \mid \mathsf{X} f \mid f \,\mathsf{S}\, g \mid \mathsf{X}^{-1} f$$

where $a \in Prop$.

Here S is the "Since" combinator, a past-time variant of U ("Until"). X^{-1} is "Previously", a past-time variant of X ("Next"). We use the standard abbreviations $\top$, $\bot$, $f \vee g$, $f \Leftrightarrow g$, ... and

$$\begin{array}{lll} \mathsf{F} f \stackrel{\text{def}}{\equiv} \top \,\mathsf{U}\, f & \mathsf{F}^{-1} f \stackrel{\text{def}}{\equiv} \top \,\mathsf{S}\, f & \mathsf{A} f \stackrel{\text{def}}{\equiv} \neg \mathsf{E} \neg f \\ & & \overset{\infty}{\mathsf{F}} f \stackrel{\text{def}}{\equiv} \mathsf{G}\mathsf{F} f \\ \mathsf{G} f \stackrel{\text{def}}{\equiv} \neg \mathsf{F} \neg f & \mathsf{G}^{-1} f \stackrel{\text{def}}{\equiv} \neg \mathsf{F}^{-1} \neg f & \overset{\infty}{\mathsf{G}} f \stackrel{\text{def}}{\equiv} \mathsf{F}\mathsf{G} f \end{array} \tag{3}$$

F and G corresponds to the $\Diamond$ and $\Box$ notations sometimes used in modal logics. In *PCTL**, (1) is written $\mathsf{G}(\mathtt{problem} \Rightarrow \mathsf{F}^{-1}\mathtt{cause})$.

Though we did not make the usual (unnecessary) distinction between "state" and "path" formulas, *PCTL** includes as fragments the *CTL* and *CTL** branching-time temporal logics, as well as the *PTL* linear-time temporal logic. In all the following, "a logic" means "a fragment of *PCTL**".

A *pure-future formula* is a formula in which no X^{-1} and S occur. Then *CTL** is the fragment of *PCTL** containing all pure-future formulas. A *state formula* is a pure-future formula that starts with a A or E quantifier. A *linear-time formula* is a formula without any E (or A) quantifier.

1.2. Semantics

Temporal logics are interpreted in Kripke structures:

Definition 1.2. A *Kripke structure* S is a tuple $S = \langle Q_S, R_S, l_S\rangle$, where $Q_S = \{p, q, \ldots\}$ is a set of *states*, $R_S \subseteq Q_S \times Q_S$ is a total[2] *accessibility relation*, and $l_S : Q_S \to 2^{Prop}$ is a labeling of the states with propositions.

A *run* in a structure S is any infinite sequence of states $q_0.q_1\ldots$ s.t. $q_i R q_{i+1}$ for $i = 0, \ldots$ We write $\Pi_S(q) = \{\pi, \ldots\}$ for the set of all runs (in S) starting from q, and $\Pi(S)$ for the set of all runs in S. For any i, $\pi(i)$ ($\stackrel{\text{def}}{=} q_i$), π^i ($\stackrel{\text{def}}{=} q_i.q_{i+1}\ldots$), and $\pi_{|i}$ ($\stackrel{\text{def}}{=} q_0.q_1\ldots q_{i-1}$) are resp. the ith state, the ith suffix and the ith prefix of π.

A *PCTL** formula expresses properties of a moment in a run. Formally, we define, for any $\pi \in \Pi(S)$ and any $n = 0, 1, 2, \ldots$ when a formula $f \in PCTL^*$ is true of run π at time n, written $\pi, n \models_S f$. We often drop the "S" subscript when it is clear from the context.

Definition 1.3 (*Semantics of PCTL**). We define $\pi, n \models_S f$ by induction on the structure of f:

$$
\begin{array}{ll}
\pi, n \models a & \text{iff } a \in l(\pi(n)),\\
\pi, n \models f \wedge g & \text{iff } \pi, n \models f \text{ and } \pi, n \models g,\\
\pi, n \models \neg f & \text{iff } \pi, n \not\models f,\\
\pi, n \models \mathsf{E} f & \text{iff there exists a } \pi' \in \Pi(S) \text{ with } \pi'_{|n} = \pi_{|n} \text{ s.t. } \pi', n \models f,\\
\pi, n \models f \,\mathsf{U}\, g & \text{iff there is a } k \geqslant n \text{ s.t. } \pi, k \models g \text{ and } \pi, i \models f \text{ for all } n \leqslant i < k,\\
\pi, n \models \mathsf{X} f & \text{iff } \pi, n+1 \models f,\\
\pi, n \models f \mathsf{S} g & \text{iff there is a } 0 \leqslant k \leqslant n \text{ s.t. } \pi, k \models g \text{ and } \pi, i \models f \text{ for all } k < i \leqslant n,\\
\pi, n \models \mathsf{X}^{-1} f & \text{iff } n > 0 \text{ and } \pi, n-1 \models f.
\end{array}
$$

Informally, $\pi(n)$ is the present state. The prefix $\pi_{|n}$ is the past and π^n is a selected future. a means "a holds now", $f \,\mathsf{U}\, g$ means "g will hold at some point in the (selected) future, and f holds in the meantime", $\mathsf{X} f$ means "f holds at the next moment", $\mathsf{X}^{-1} f$ means "f did hold at the previous moment", $f \,\mathsf{S}\, g$ means "g did hold in the past and f has been holding ever since that moment", $\mathsf{E} f$ means that "the present admits a

[2] Restricting to structures with a total accessibility relation is a technical simplification that does not change any of our expressivity results.

possible future for which f holds", $\overset{\infty}{\mathsf{F}} f$ means that "f will hold infinitely many times in the (selected) future" and $\overset{\infty}{\mathsf{G}} f$ means that "f will hold at all but finitely many times in the (selected) future".

This semantics is *Ockhamist* [23, 27] in the sense that it views the past as *fixed* (and finite) and only considers nondeterminism in the future. This is in contrast with e.g. the CTL^*+Past from [25] where it is possible to quantify over all potential ways of reaching a given state. We claim that the Ockhamist viewpoint is more suited to the specification of reactive systems behaviour, because it considers states in a computation tree, while the non-Ockhamist viewpoints consider machine states (where past is not very meaningful).

Notice also that our semantics considers a *cumulative past*, where the history of the current situation increases with time and is never forgotten. This contrasts with the definition from [14] where one has

$$\pi, n \models \mathsf{E} f \text{ iff there is a } \pi' \in \Pi(\pi(n)) \text{ s.t. } \pi', 0 \models f$$

We believe our definition is more natural and, because a non-cumulative past is sometimes handy, we present in Section 6 a larger logic, *NCTL**, which allows both viewpoints.

Now for a formula f we define derived truth concepts:

$$\begin{array}{rcll}
\pi \models_S f & \overset{\text{def}}{\Leftrightarrow} & \pi, 0 \models_S f & \text{reads "\textit{run } } \pi \text{ \textit{satisfies} } f\text{"} \\
q \models_S f & \overset{\text{def}}{\Leftrightarrow} & \pi \models_S f \text{ for all } \pi \in \Pi(q) & \text{"\textit{state} } q \text{ \textit{satisfies} } f\text{"} \\
S \models f & \overset{\text{def}}{\Leftrightarrow} & \pi \models_S f \text{ for all } \pi \in \Pi(S) & \text{"\textit{structure} } S \text{ \textit{satisfies} } f\text{"} \\
\models_g f & \overset{\text{def}}{\Leftrightarrow} & \pi, n \models_S f \text{ for all } (\pi, n) & \text{"} f \text{ \textit{is (globally) valid}"} \\
 & & \text{in all Kripke structures } S & \\
\models_i f & \overset{\text{def}}{\Leftrightarrow} & S \models f \text{ for all Kripke structures } S & \text{"} f \text{ \textit{is (initially) valid}"}
\end{array}$$

$\models_g f$ entails $\models_i f$ but the converse is not true, and in fact $\models_g f$ iff $\models_i \mathsf{G} f$. As indicated by our definition of $S \models f$, it is the "$\models_i$", so-called anchored [19], notion of validity that interests us here, as is usual in computer science [8].

The following proposition is a formal justification that an Ockhamist viewpoint is sensible.

Proposition 1.4. *Two states q, q' (in a finite Kripke Structure) satisfy the same PCTL* formulas iff they are bisimilar.*

This generalizes Theorem 3.2 of [2] where a formal definition of the well-known notion of bisimilarity can be found. The proof is an easy corollary of our Theorem 3.12. In general, this proposition does not hold for logics with a non-Ockhamist viewpoint.

Definition 1.5. (1) We say that two formulas f and g are *equivalent*, written $f \equiv g$, when for all (π, n) in all structures, $\pi, n \models f$ iff $\pi, n \models g$.

(2) We say that f and g are *initially equivalent*, written $f \equiv_i g$, when for all π in all structures, $\pi \models f$ iff $\pi \models g$.

Thus $f \equiv g$ when $\models_g f \Leftrightarrow g$, and $f \equiv_i g$ when $\models_i f \Leftrightarrow g$. Clearly, $f \equiv g$ entails $f \equiv_i g$ but the converse is not true. Here is a simple example: $\mathsf{X}^{-1}\top \equiv_i \bot$ because no run satisfies $\mathsf{X}^{-1}\top$ at its starting point. But of course $\mathsf{X}^{-1}\top \not\equiv \bot$ because for any run π with length at least 2, we have $\pi, 1 \models \mathsf{X}^{-1}\top$. Similarly, $\mathsf{G}(\mathtt{problem} \Rightarrow \mathsf{F}^{-1}\mathtt{cause})$ is initially equivalent and not globally equivalent to (2).[3]

When we use temporal logics to reason about programs, it is customary to consider initial validity as the basic concept. Specifications refer to the runs of a program, starting from some initial states. Therefore, we are content to replace a given formula f by an equivalent f', using "initial equivalence" as the relevant notion. The interest with global equivalence is that it is substitutive: if $f \equiv f'$ then f can be replaced by f' in any temporal context, yielding equivalent formulas. That is, $\equiv$ is a congruence w.r.t. all temporal combinators. On the other hand, $\equiv_i$ is only a congruence w.r.t. boolean combinators (and X^{-1} and S).

Considering initial equivalence as the correctness criterion allows to eliminate past-time combinators, according to

$$\begin{aligned} \mathsf{X}^{-1} f &\equiv_i \bot, \\ f\,\mathsf{S}\,g &\equiv_i g. \end{aligned} \tag{4}$$

but, because $\equiv_i$ is not substitutive in temporal contexts, these simplification rules cannot be used in all situations.

2. A menagerie of temporal logics

Many fragments of *CTL** have been used and investigated. In fact, *CTL** was first proposed as a logic which included all other previously proposed temporal logics. Emerson and Halpern introduced a very convenient device to denote such fragments. Following them, we write $B(C,\ldots)$ the fragment of *CTL** where $C,\ldots$ are the only allowed linear-time combinators, and where every occurrence of a linear-time combinator must be under the immediate scope of an E or A quantifier (the "*B*" is for "branching"). For example:

- $B(\mathsf{X},\mathsf{F})$ is the *UB* logic from [1]. $\mathsf{AFX}a$ is not in *UB* while $\mathsf{AFAX}a$ is.
- $B(\mathsf{X},\mathsf{U})$: This is the *CTL* logic from [6]. $\mathsf{A}[a\,\mathsf{U}\,\mathsf{EX}b]$ is in *CTL* but not in *UB* where only F and X can be used.
- $B(\mathsf{X},\mathsf{U},\overset{\infty}{\mathsf{F}})$: This is the *ECTL* ("Extended *CTL*") logic from [11]. $\mathsf{E}\overset{\infty}{\mathsf{F}}a$ is in *ECTL* but not in *CTL*.

[3] To be precise (2) is not sufficient. $\mathsf{G}(\mathtt{problem} \Rightarrow \mathsf{F}^{-1}\mathtt{cause}) \equiv_i \neg(\neg\mathtt{cause}\,\mathsf{U}\,(\mathtt{problem} \wedge \neg\mathtt{cause}))$ is correct.

$B(C,\ldots,\neg,\wedge)$ denotes a fragment enlarging $B(C,\ldots)$ inasmuch as it allows boolean combinators to appear between the linear-time combinators $C,\ldots$ and the branching-time quantifier on top of it. For example:

- $B(\mathsf{X},\mathsf{F},\neg,\wedge)$ (also called UB^+) allows $\mathsf{E}[\mathsf{F}a\wedge\mathsf{F}b \Rightarrow \mathsf{X}c]$.
- $B(\mathsf{X},\mathsf{U},\neg,\wedge)$ (also called CTL^+) allows $\mathsf{A}[(a\,\mathsf{U}\,b)\vee\mathsf{X}c]$.
- $B(\mathsf{X},\mathsf{U},\overset{\infty}{\mathsf{F}},\neg,\wedge)$ is the $ECTL^+$ logic that roughly corresponds to the CTF logic introduced in [9].

Now, because for any branching-time formulas f, g, … we have $\mathsf{E}\mathsf{X}^{-1}f \equiv \mathsf{X}^{-1}f$, $\mathsf{E}(f\mathsf{S}g) \equiv f\mathsf{S}g$, etc., we do not enforce the use of a E or A immediately on top of past-time combinators in a $B(C,\ldots)$ fragment. For example, we consider that $\mathsf{X}^{-1}\big(\mathsf{E}(\mathsf{F}^{-1}a)\mathsf{U}\,\mathsf{F}^{-1}b\big)$ is a $B(\mathsf{U},\mathsf{X}^{-1},\mathsf{S})$ formula.

This (syntactic) classification of relevant fragments of $PCTL^*$ can be linked to semantic notions through the following

Definition 2.1.

- A formula f is a *future-formula* iff the truth of f at (π,n) in S only depends on the future $\pi(n)\pi(n+1)\ldots$, i.e. if $\pi^n = \pi'^m$ implies $\pi,n \models_S f \Leftrightarrow \pi',m \models_S f$.
- f is a *present-formula* iff the truth of f at (π,n) in S only depends on the current state, i.e. if $\pi(n) = \pi'(m)$ implies $\pi,n \models_S f \Leftrightarrow \pi',m \models_S f$
- f is a *branching-time formula* iff the truth of f at (π,n) in S does not depend of the selected future, i.e. if $\pi(0)\ldots\pi(n) = \pi'(0)\ldots\pi'(m)$ implies $\pi,n \models_S f \Leftrightarrow \pi',m \models_S f$.

Clearly, any present-formula is a future-formula and a branching formula.

Proposition 2.2. (1) *Any pure-future formula is a future-formula.*
(2) *Any state formula is a present-formula.*

Proposition 2.3. *A logic of the form $B(C,\ldots,\neg,\wedge)$ only contains branching-time formulas.*

3. Compared expressivity

When we discuss comparative expressivity between two temporal logics L_1 and L_2, two notions can be used:

Definition 3.1. (1) L_1 is *less*[4] *expressive* than L_2, written $L_1 \preceq_g L_2$, if for any $f_1 \in L_1$ there is a $f_2 \in L_2$ s.t. $f_1 \equiv f_2$.

[4] or equally …

(2) L_1 is *initially less expressive* than L_2, written $L_1 \preceq_i L_2$, if for any $f_1 \in L_1$ there is a $f_2 \in L_2$ s.t. $f_1 \equiv_i f_2$.

Clearly, $L_1 \subseteq L_2$ implies $L_1 \preceq_g L_2$. Also, $L_1 \preceq_g L_2$ implies $L_1 \preceq_i L_2$. In both cases the converse is not true in general. As usual we denote by "$\prec_*$" and "$\equiv_*$" the strict ordering and the equivalence relation induced by "$\preceq_*$".

Also, for pure-future logics, both "$\preceq_g$" and "$\preceq_i$" coincide. For pure-future logics, the classical hierarchy result has been established in [10, 11]:

$$UB \prec UB^+ \prec CTL \equiv CTL^+ \prec ECTL \prec ECTL^+ \prec CTL^*$$

When logics with past-time are considered, the most relevant result is the Separation Theorem for *PTL*.

Theorem 3.2 (Gabbay [13]). *Any PPTL formula can be rewritten into an equivalent totally separated formula (that is, a boolean combination of pure-past and pure-future PPTL formulas.)*

See [12] for a proof. The immediate corollary is

Corollary 3.3. $PPTL \equiv_i PTL$.

Proof. With totally separated formulas, (4) allows to fully remove past-time constructs (modulo $\equiv_i$). □

There is a branching-time equivalent to this last result:

Theorem 3.4 (Hafer and Thomas [14]). $PCTL^* \equiv_i CTL^*$.

(The proof in [14] applies to their definition of $PCTL^*$ but it can be adapted without any difficulty to our definition.)

4. Temporal logics with X^{-1} and S

Let us write *PCTL* for "*CTL* + Past", i.e. $B(\mathsf{X}, \mathsf{U}, \mathsf{X}^{-1}, \mathsf{S})$. The question which initiated our study was "*does PCTL* $\equiv_i$ *CTL*?". The answer is unfortunately:

$$PCTL \not\preceq_i CTL$$

This section investigates why.

One problem is that the simple addition of X^{-1} gives a logic which cannot be (initially) less expressive than $ECTL^+$ which is the largest relevant fragment of CTL^* for which efficient model-checking exists.

Theorem 4.1. $ECTL^+ \not\succeq_i B(\mathsf{F}, \mathsf{X}^{-1})$.

Proof. The *CTL** formula $\mathsf{EG}(a \vee \mathsf{X}a)$ has no $ECTL^+$ equivalent [11] but it is initially equivalent to the $B(\mathsf{F}, \mathsf{X}^{-1})$ formula $\mathsf{EG}(a \vee \mathsf{X}^{-1}a \vee \neg \mathsf{X}^{-1}\top)$. □

The simple addition of S brings similar problems.

Theorem 4.2. $ECTL^+ \not\leq_i UB + \mathsf{S} \stackrel{\text{def}}{=} B(\mathsf{X}, \mathsf{F}, \mathsf{S})$.

Proof. The *CTL** formula $\mathsf{E}(s \vee a\mathsf{U}b)\mathsf{U}r$ has no $ECTL^+$ equivalent but it can be written (modulo $\equiv_i$) in $UB + \mathsf{S}$. See the Appendix. □

The F combinator of *UB* is necessary here and for example, one has

Theorem 4.3. $B(\mathsf{X}, \mathsf{X}^{-1}, \mathsf{S}) \equiv_i B(\mathsf{X})$.

Proof. See the Appendix. □

In some way, these negative results rely on the fact that our semantics considers the past as fixed so that X^{-1} and S can express properties which usually can only be expressed in a linear-time framework. However, adopting a non-Ockhamist viewpoint would make things even worse since no translation can be expected if Proposition 1.4 does not hold.

5. Temporal logics with F^{-1}

When we consider the F^{-1} past-time combinator alone, the problems we had with X^{-1} and S do not occur.

Theorem 5.1. $CTL + \mathsf{F}^{-1} \stackrel{\text{def}}{=} B(\mathsf{X}, \mathsf{U}, \mathsf{F}^{-1}) \equiv_i CTL$.

This result is fundamental. The crucial step in the proof is to establish the following lemma.

Lemma 5.2 (Separation lemma for $CTL + \mathsf{F}^{-1}$). *Any* $CTL + \mathsf{F}^{-1}$ *formula is (globally) equivalent to a* separated $CTL + \mathsf{F}^{-1}$ *formula.*

Here a (partially) *separated* formula is a formula where no past-time combinator occurs under the scope of a future-time combinator.[5] Once a $CTL + \mathsf{F}^{-1}$ formula has been separated, (4) can be applied to eliminate all past-time constructs. (See the Appendix for a proof of Lemma 5.2.)

[5] Observe that the usual notion of (totally) separated formula used for linear-time logics will not work in our branching-time framework.

The previous theorem shows how one can extend *CTL* with past-time constructs without loosing the ability to translate back into *CTL*. This must be done precisely. Adding F^{-1} will do, adding X^{-1} or S will fail. F^{-1} is a specialization of S but it is nonetheless sufficient in many situations. For example, (1) is written in $CTL + \mathsf{F}^{-1}$.

F^{-1} can be added to other logics as well. We have

Theorem 5.3. $CTL^{+} + \mathsf{F}^{-1} \stackrel{\text{def}}{=} B(\mathsf{X}, \mathsf{U}, \mathsf{F}^{-1}, \neg, \wedge) \equiv_i CTL.$

A similar result exists for $ECTL^{+}$:

Theorem 5.4. $ECTL^{+} + \mathsf{F}^{-1} \stackrel{\text{def}}{=} B(\mathsf{X}, \mathsf{U}, \stackrel{\infty}{\mathsf{F}}, \mathsf{F}^{-1}, \neg, \wedge) \equiv_i ECTL^{+}.$

Adding F^{-1} to *ECTL* increase the expressive power:

Theorem 5.5. $ECTL^{+} \succ_i ECTL + \mathsf{F}^{-1} \succ_i ECTL.$

Proof. Modulo $\equiv_i$, the $ECTL + \mathsf{F}^{-1}$ formula $\mathsf{E}\stackrel{\infty}{\mathsf{F}}(a \wedge \mathsf{G}^{-1} b)$ cannot be expressed in *ECTL*, and the $ECTL^{+}$ formula $\mathsf{E}(\stackrel{\infty}{\mathsf{F}} a \wedge \stackrel{\infty}{\mathsf{F}} b)$ cannot be expressed in $ECTL + \mathsf{F}^{-1}$. See the Appendix. □

6. A combinator for From Now On

We introduce a new unary combinator, N, for "From Now On"[6] or "Henceforth", in our $PCTL^*$ logic and write $NCTL^*$ for the logic $PCTL^* + \mathsf{N}$.

The semantics of N is given by

$$\pi, n \models \mathsf{N} f \text{ iff } \pi^n, 0 \models f$$

That is, $\mathsf{N}f$ holds if f holds when we forget the past, or if "from now on f holds". Here is an example motivating this new construct. Assume we want to state that $\mathsf{AG}(\texttt{problem} \Rightarrow \mathsf{F}^{-1}\texttt{cause})$ holds as soon as a proper reset had been done. We can write

$$\mathsf{AG}[\texttt{reset} \Rightarrow \mathsf{AG}(\texttt{problem} \Rightarrow \mathsf{F}^{-1}\texttt{cause})] \tag{5}$$

Then, if a problem occurs after a proper reset, there must have been a cause, *but the cause may have occurred before the reset.* If we want to specify that every time there is a reset, *then from now on* no problem can occur without a cause (i.e., a cause occurring after the reset), we can write:

$$\mathsf{AG}[\texttt{reset} \Rightarrow \mathsf{NAG}(\texttt{problem} \Rightarrow \mathsf{F}^{-1}\texttt{cause})] \tag{6}$$

[6] No connection with the Now of H. Kamp, Formal properties of 'now', *Theoria*, 227–263, 1971.

The difference between (5) and (6) is important. It exemplifies the interest of having N. With N we can encode the definition for E used in [14]: their $\mathsf{E}f$ is equivalent to our $\mathsf{NE}f$. Then, the *PCTL** logic of [14] can express (6). But it cannot express (directly) (5). In our experience, both constructs are useful, and that's why we propose a specific combinator.

Finally, N is very useful to characterize key semantic properties:

Proposition 6.1. *A NCTL* formula f is a future-formula iff* $f \equiv \mathsf{N}f$.

and to explain the difference between *initial* and *global* validity:

Proposition 6.2. $\models_i f$ *iff* $\models_g \mathsf{N}f$.

Basic properties of N are:

Proposition 6.3. *For all NCTL* formulas* f, g:

- $\mathsf{N}(f \wedge g) \equiv \mathsf{N}f \wedge \mathsf{N}g$ *and* $\mathsf{N}\neg f \equiv \neg \mathsf{N}f$,
- $\mathsf{NE}f \equiv \mathsf{EN}f$ *and* $\mathsf{NN}f \equiv \mathsf{N}f$,
- $\mathsf{NX}^{-1}f \equiv \bot$,
- $\mathsf{N}(f\mathsf{S}g) \equiv \mathsf{N}g$, *entailing* $\mathsf{NF}^{-1}f \equiv \mathsf{N}f$.

Formulas using N can be translated into equivalent (often longer) formulas without N:

Theorem 6.4. $NCTL^* \equiv PCTL^* \equiv_i CTL^*$.

Proof. This is a simple extension of Theorem 3.4. The new point is to eliminate N: consider $\mathsf{N}f$ with $f \in PCTL^*$ and rewrite f into a separated f'. Then $\mathsf{N}f'$ can be rewritten into some *PCTL** formula thanks for Propositions 6.2 and 6.3. □

Similarly, we can extend Theorems 5.1 and 5.4 into

Theorem 6.5. $CTL^+ + \mathsf{F}^{-1} + \mathsf{N} \stackrel{\text{def}}{=} B(\mathsf{X}, \mathsf{U}, \mathsf{F}^{-1}, \mathsf{N}, \neg, \wedge) \equiv_i CTL$.

Theorem 6.6. $ECTL^+ + \mathsf{F}^{-1} + \mathsf{N} \stackrel{\text{def}}{=} B(\mathsf{X}, \mathsf{U}, \stackrel{\infty}{\mathsf{F}}, \mathsf{F}^{-1}, \mathsf{N}, \neg, \wedge) \equiv_i ECTL^+$.

Proof. Like Theorems 5.1 and 5.4. In both cases, occurrences of N are easy to simplify because we deal with separated formulas. □

N is not only a notational facility. It sometimes (strictly) adds expressive power. For example, writing *PUB* for $UB + \mathsf{X}^{-1} + \mathsf{F}^{-1} \stackrel{\text{def}}{=} B(\mathsf{X}, \mathsf{F}, \mathsf{X}^{-1}, \mathsf{F}^{-1})$, we have $UB \prec_i PUB \prec_i$

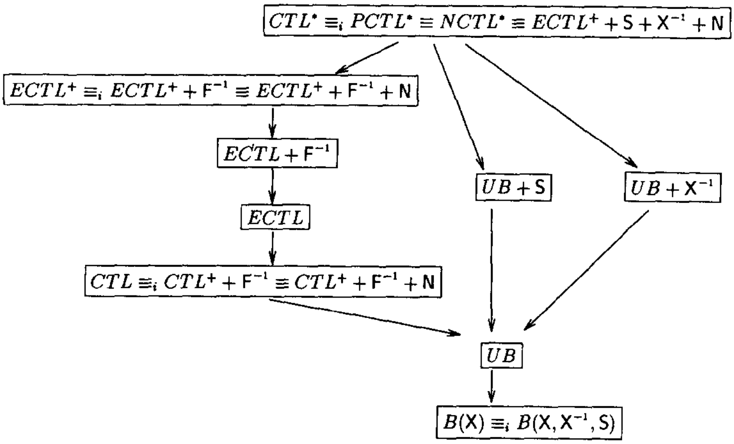

Fig. 1. A hierarchy of temporal logics with past

$PUB + \mathsf{N} \equiv_i CTL$ and $ECTL + \mathsf{F}^{-1} \prec_i ECTL + \mathsf{F}^{-1} + \mathsf{N} \prec_i ECTL^+ + \mathsf{F}^{-1} + \mathsf{N} \equiv_i ECTL^+$. Similarly, while $ECTL^+ + \mathsf{S} + \mathsf{X}^{-1} \prec_i PCTL^*$, we have

Theorem 6.7. $ECTL^+ + \mathsf{S} + \mathsf{X}^{-1} + \mathsf{N} \equiv PCTL^* \equiv_i CTL^*$.

Proof. A corollary of the Normal Form Theorem [21, p. 296]. See the Appendix. $\square$

Fig. 1 summarizes the hierarchy we established. An arrow from L_1 to L_2 means that $L_1 \succ_i L_2$. No arrow can be added because $CTL \not\preceq_i UB + \mathsf{S}$ and $CTL \not\preceq_i UB + \mathsf{X}^{-1}$. Our proof that $\mathsf{EF}(a \wedge \mathsf{E}b\mathsf{U}c \wedge \mathsf{E}b'\mathsf{U}c')$ (resp. $\mathsf{E}a\mathsf{U}b$) cannot be expressed modulo $\equiv_i$ in $UB + \mathsf{S}$ (resp. $UB + \mathsf{X}^{-1}$) is beyond the scope of this article where the emphasis is on translatability through simple rewrite rules.

7. Conclusion

In this paper, we investigated which past-time combinators can be added to which branching-time temporal logics with the conflicting aims of

- enhancing practical expressivity,
- having translation algorithms into pure-future branching-time logics, like *CTL* and $ECTL^+$.

We also proposed a new combinator, N for "From Now On" and showed how it allows simple formulations of some practical temporal properties.

In general, logics with past-time combinators can be translated into pure-future logics, provided one is willing to have CTL^* as a target. When *CTL* or $ECTL^+$ is the target, we proved that one can only add N and F^{-1} before loosing translatability.

We claim that $CTL + \mathsf{F}^{-1} + \mathsf{N}$ is convenient for specification and model-checking (through a simple translation procedure). Of course such an approach is not perfect. For example, it does not include the usual diagnostic mechanism one often finds in model-checking tools. More importantly, complexity issues sometimes make the whole scheme unapplicable (by complexity, we mean the *size* of a pure-future formula equivalent to a given formula). This problem was not investigated in this study because we conjecture that our translation from $CTL + \mathsf{F}^{-1}$ into CTL is nonelementary, exactly like Gabbay's translation for PTL+Past is likely to be. In actual practice, this potential combinatorial explosion does not occur frequently, and all formulas in the Lift example of [16] have quickly been translated automatically. This is probably because these formulas have a low modal height. However it seems difficult to pinpoint a sensible fragment of $CTL + \mathsf{F}^{-1} + \mathsf{N}$ for which no explosion will occur: the formula $\mathsf{EF}(\mathsf{F}^{-1}a_1 \wedge \cdots \wedge \mathsf{F}^{-1}a_n)$ has modal height 2, and is initially equivalent to the CTL^+ formula $\mathsf{E}(\mathsf{F}a_1 \wedge \cdots \wedge \mathsf{F}a_n)$ for which no CTL equivalent of size less than $n!$ seems to exist.

Topics deserving further studies are (among others):

Axiomatizations for temporal logics with N. Given a complete axiomatization for CTL^*, it is easy to get complete axiomatizations for $NCTL^*$ by providing axioms for the separation of formulas. But it would be interesting to study axiomatizations capturing natural way of reasoning with past-time combinators and N.

Extensions of the separation methods. The separation methods we developed for branching-time logics should be investigated in the contexts of noninterleaving temporal logics, of interval temporal logics, of real-time temporal logics, ...

Modal logics of reactive systems. We already investigated in [17] how these methods can be used for modal logics characterizing behavioral equivalences of reactive systems. This research direction has many possible prolongations.

Appendix

Proof of Theorem 4.2

Lemma A.1. $\mathsf{E}(s \vee a\mathsf{U}b)\mathsf{U}r$ *can be expressed in* $UB + \mathsf{S}$.

Proof. The idea is to state that r can be reached ($\mathsf{EF}r$) in a way where all previously encountered states satisfy s or $a\mathsf{U}b$. It is enough to ensure $s \vee a \vee b$ all along, provided that all $a \wedge \neg b \wedge \neg s$ states satisfy $a\mathsf{U}b$ (along the selected future). Then, the configuration to avoid is a past state satisfying $\neg a \wedge \neg b$ and $(\neg b)\mathsf{S}(\neg b \wedge \neg s)$. This is essentially what we express in $UB + \mathsf{S}$ through the following:

$$\begin{aligned} & r \vee \mathsf{EF}\Big(\mathsf{EX}b \wedge a\mathsf{S}\Big[r \wedge a \wedge \neg\mathsf{F}^{-1}\Big(\neg a \wedge \neg b \wedge (\neg b)\mathsf{S}(\neg b \wedge \neg s)\Big)\Big]\Big) \\ \mathsf{E}(s \vee a\mathsf{U}b)\mathsf{U}r \equiv_i & \vee \mathsf{EF}\Big(\mathsf{EX}(r \wedge b) \wedge \neg\mathsf{F}^{-1}\Big(\neg a \wedge \neg b \wedge (\neg b)\mathsf{S}(\neg b \wedge \neg s)\Big)\Big) \\ & \vee \mathsf{EF}\Big(\neg\mathsf{F}^{-1}\Big(\neg a \wedge \neg b \wedge (\neg b)\mathsf{S}(\neg b \wedge \neg s)\Big) \wedge \mathsf{EX}(r) \wedge \neg(\neg b)\mathsf{S}(\neg b \wedge \neg s)\Big) \end{aligned}$$

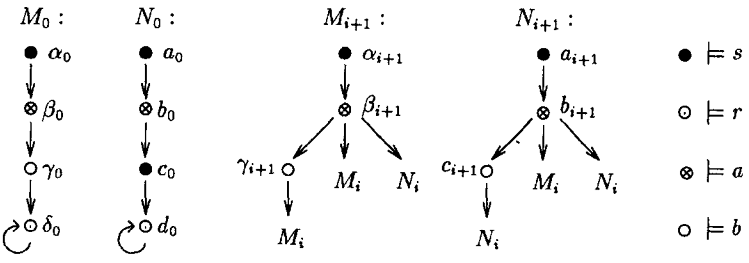

Fig. 2. A family $M_0, M_1, \ldots, N_0, N_1, \ldots$ of Kripke structures.

where the added complexity deals with various special cases, mostly regarding the position of the last required b before of after the r state. $\square$

We now need to prove that $\mathsf{E}(s \vee a\mathsf{U}b)\mathsf{U}r$ cannot be expressed in $ECTL^+$. This uses combinatorial techniques inspired from [11]. Consider the family of Kripke structures $M_0, M_1, \ldots$ and $N_0, N_1, \ldots$ given in Fig. 2.

Where the "color" indicates which propositions hold in which states. Observe that $\alpha_i \models \mathsf{E}(s \vee a\mathsf{U}b)\mathsf{U}r$ and $a_i \not\models \mathsf{E}(s \vee a\mathsf{U}b)\mathsf{U}r$ for all $i = 0, 1, \ldots$ However, writing $|f|$ for the size of a formula f, we have

Lemma A.2. *For all $f \in CTL$, and all $i \geqslant |f|$,*

$$\begin{aligned} \alpha_i \models f \quad &\textit{iff} \quad a_i \models f, \\ \beta_i \models f \quad &\textit{iff} \quad b_i \models f, \\ \gamma_i \models f \quad &\textit{iff} \quad c_i \models f. \end{aligned}$$

Proof. By structural induction on f as in [11]. $\square$

Now there remains to extend the previous lemma to cover $ECTL^+$ and not just CTL.

Lemma A.3. *For all f in $ECTL^+$, there is a f^* in CTL s.t. f and f^* are equivalent over all states of Fig. 2.*

Proof. By induction on f. The interesting case is when f has the form

$$f = \mathsf{E}(g \wedge \overset{\infty}{\mathsf{F}} \varphi_1 \wedge \ldots \wedge \overset{\infty}{\mathsf{F}} \varphi_n \wedge \overset{\infty}{\mathsf{G}} \psi)$$

with $\mathsf{E}g, \psi, \varphi_1, \ldots, \varphi_n$ in CTL^+. Because in the structures of Fig. 2 all runs eventually end up looping (on d_0 or δ_0), f is equivalent (in these models) to $\mathsf{E}(g \wedge \mathsf{FAG}(\varphi_1 \wedge \ldots \wedge \varphi_n \wedge \psi))$ which is a CTL^+ formula. There only remains to transform this into a CTL formula and we are done. $\square$

Proof of Theorem 4.3

Theorem 4.3 is proved by establishing a separation lemma for $B(\mathsf{X},\mathsf{X}^{-1},\mathsf{S})$. Afterward, it is enough to apply (4) on separated formulas. (Here we use the partial separation notion from Section 5.)

Lemma A.4 (Separation lemma for $B(\mathsf{X},\mathsf{X}^{-1},\mathsf{S})$). *Any $f \in B(\mathsf{X},\mathsf{X}^{-1},\mathsf{S})$ formula f is (globally) equivalent to a separated $f' \in B(\mathsf{X},\mathsf{X}^{-1},\mathsf{S})$.*

Proof. By structural induction. The induction step is obvious when f has the form a, $\neg g$ or $f_1 \wedge f_2$. When f is some $\mathsf{X}^{-1}g$ or $f_1 \mathsf{S} f_2$, the induction hypothesis gives us an equivalent $\mathsf{X}^{-1}g'$ or $f_1' \mathsf{S} f_2'$ which is separated. Finally, because $\mathsf{AX}g \equiv \neg\mathsf{EX}\neg g$, it is enough to only consider the case where f has the form $\mathsf{EX}g$.

Assume then that f is some $\mathsf{EX}g$. By ind. hyp., g is equivalent to a separated g'. With boolean manipulations, g' can be written as

$$g' \equiv \bigvee_i (\varphi_i^+ \wedge \psi_{i,1} \wedge \ldots)$$

where the φ_i^+'s are pure-future and the $\psi_{i,j}$'s are separated formulas of the form $\mathsf{X}^{-1}\psi$, or $\neg\mathsf{X}^{-1}\psi$, or $\psi\mathsf{S}\psi'$, or $\neg(\psi\mathsf{S}\psi')$. Using

$$\begin{aligned} f\mathsf{S}g &\equiv g \vee f \wedge \mathsf{X}^{-1}(f\mathsf{S}g) \\ \neg\mathsf{X}^{-1}f &\equiv \neg\mathsf{X}^{-1}\top \vee \mathsf{X}^{-1}\neg f \\ \mathsf{X}^{-1}f \wedge \mathsf{X}^{-1}g &\equiv \mathsf{X}^{-1}(f \wedge g) \end{aligned}$$

(and the fact that these equations respect separation), we can write g' as

$$g' \equiv \bigvee_i (\varphi_i^+ \wedge \mathsf{X}^{-1}\psi_i \; \{\wedge \neg\mathsf{X}^{-1}\top\})$$

where the $\{\ldots\}$ notation means that $\neg\mathsf{X}^{-1}\top$ may or may not appear (depending on i). Then $\mathsf{EX}g'$ is easily rewritten into a separated formula, thanks to

$$\begin{aligned} \mathsf{EX}(\textstyle\bigvee_i g_i) &\equiv \textstyle\bigvee_i \mathsf{EX}g_i \\ \mathsf{EX}(\varphi^+ \wedge \mathsf{X}^{-1}\psi) &\equiv \psi \wedge \mathsf{EX}\varphi^+ \\ \mathsf{EX}(\neg\mathsf{X}^{-1}\top \wedge \ldots) &\equiv \bot \qquad \square \end{aligned}$$

Proof of Lemma 5.2 and Theorem 5.1

The translation is done in several steps. We use contexts, i.e. $(CTL + \mathsf{F}^{-1})$ formulas with variables in them. The x in $f[x]$ can be replaced by any $(CTL + \mathsf{F}^{-1})$ formula: we write $f[g]$ for f with g in place of x. *Note that x may appear several times in $f[x]$. This is a key point in our method, used to collect copies of duplicated subformulas.*

Lemma A.5. *If $f[x]$ is a pure-future context, then $f[\mathsf{F}^{-1}x]$ is (globally) equivalent to a separated $f'[x, \mathsf{F}^{-1}x]$ with $f'[x,y]$ a pure-future context.*

Proof. By structural induction on $f[x]$. Saying that $f[x]$ is pure-future is just saying that it is in *CTL*. We spend some time considering one case in detail:

- Assume $f[x]$ is some $\mathsf{E}\varphi[x]\mathsf{U}\psi[x]$. By ind. hyp., $\varphi[\mathsf{F}^{-1}x]$ and $\psi[\mathsf{F}^{-1}x]$ are equivalent to some separated $\varphi'[x, \mathsf{F}^{-1}x]$ and $\psi'[x, \mathsf{F}^{-1}x]$. Then $f[\mathsf{F}^{-1}x] \equiv \mathsf{E}\varphi'[x, \mathsf{F}^{-1}x]\mathsf{U}\psi'[x, \mathsf{F}^{-1}x]$. In φ' and ψ', $\mathsf{F}^{-1}x$ can only appear under boolean combinators because of the separation property. We can use boolean manipulations to obtain

$$f[\mathsf{F}^{-1}x] \equiv \mathsf{E}\Big((\mathsf{F}^{-1}x \wedge \alpha) \vee (\neg \mathsf{F}^{-1}x \wedge \beta) \vee \gamma\Big)\mathsf{U}\Big((\mathsf{F}^{-1}x \wedge \alpha') \vee (\neg \mathsf{F}^{-1}x \wedge \beta') \vee \gamma'\Big) \qquad \text{(A.1)}$$

where $\alpha, \beta, \gamma, \alpha', \beta'$ and γ' are pure-future. Then we use distributivity

$$\mathsf{E}g\mathsf{U}(h \vee h') \equiv (\mathsf{E}g\mathsf{U}h) \vee (\mathsf{E}g\mathsf{U}h')$$

to further simplify (A.1). We obtain several "E_U_" formulas with at most three occurrences of $\mathsf{F}^{-1}x$. There we use the following five rewrite rules to extract $\mathsf{F}^{-1}x$ from the scope of the U:

(R1) $$\mathsf{E}\gamma\mathsf{U}(\alpha' \wedge \mathsf{F}^{-1}x) \equiv \mathsf{F}^{-1}x \wedge \mathsf{E}\gamma\mathsf{U}\alpha' \ \vee \ \mathsf{E}\gamma\mathsf{U}(x \wedge \mathsf{E}\gamma\mathsf{U}\alpha')$$

(R2) $$\mathsf{E}\gamma\mathsf{U}(\beta' \wedge \neg \mathsf{F}^{-1}x) \equiv \neg \mathsf{F}^{-1}x \wedge \mathsf{E}(\gamma \wedge \neg x)\mathsf{U}(\beta' \wedge \neg x)$$

(R3) $$\begin{aligned}&\mathsf{E}\Big((\alpha \wedge \mathsf{F}^{-1}x) \vee (\beta \wedge \neg \mathsf{F}^{-1}x) \vee \gamma\Big)\mathsf{U}\gamma' \\ &\quad \equiv \mathsf{F}^{-1}x \wedge \mathsf{E}(\alpha \vee \gamma)\mathsf{U}\gamma' \vee \neg \mathsf{F}^{-1}x \wedge \mathsf{E}\Big(\neg x \wedge (\beta \vee \gamma)\Big)\mathsf{U}\gamma' \\ &\quad \vee \neg \mathsf{F}^{-1}x \wedge \mathsf{E}\Big(\neg x \wedge (\beta \vee \gamma)\Big)\mathsf{U}\Big(x \wedge \mathsf{E}(\alpha \vee \gamma)\mathsf{U}\gamma'\Big)\end{aligned}$$

(R4) $$\begin{aligned}&\mathsf{E}\Big((\alpha \wedge \mathsf{F}^{-1}x) \vee (\beta \wedge \neg \mathsf{F}^{-1}x) \vee \gamma\Big)\mathsf{U}(\alpha' \wedge \mathsf{F}^{-1}x) \\ &\quad \equiv \mathsf{F}^{-1}x \wedge \mathsf{E}(\alpha \vee \gamma)\mathsf{U}\alpha' \vee \neg \mathsf{F}^{-1}x \wedge \mathsf{E}\Big(\neg x \wedge (\beta \vee \gamma)\Big)\mathsf{U}\Big(x \wedge \mathsf{E}(\alpha \vee \gamma)\mathsf{U}\alpha'\Big)\end{aligned}$$

(R5) $$\begin{aligned}&\mathsf{E}\Big((\alpha \wedge \mathsf{F}^{-1}x) \vee (\beta \wedge \neg \mathsf{F}^{-1}x) \vee \gamma\Big)\mathsf{U}(\beta' \wedge \neg \mathsf{F}^{-1}x) \\ &\quad \equiv \neg \mathsf{F}^{-1}x \wedge \mathsf{E}\Big(\neg x \wedge (\beta \vee \gamma)\Big)\mathsf{U}(\beta' \wedge \neg x)\end{aligned}$$

- Assume $f[x]$ is some $\mathsf{EX}\varphi[x]$. We proceed similarly. Using the ind. hyp. and distributivity

$$\mathsf{EX}(h \vee h') \equiv \mathsf{EX}h \vee \mathsf{EX}h'$$

lead to a situation where we only need the following rules:

(R6) $\mathsf{EX}(\alpha \wedge \mathsf{F}^{-1}x) \equiv \mathsf{EX}(\alpha \wedge x) \vee \mathsf{F}^{-1}x \wedge \mathsf{EX}\alpha$

(R7) $\mathsf{EX}(\alpha \wedge \neg \mathsf{F}^{-1}x) \equiv \mathsf{EX}(\alpha \wedge \neg x) \wedge \neg \mathsf{F}^{-1}x$

- Assume $f[x]$ is some $\mathsf{EG}\varphi[x]$. Then $f[\mathsf{F}^{-1}x] \equiv \mathsf{EG}\varphi'[x, \mathsf{F}^{-1}x]$. Because of the separation assumption, w.l.o.g. we can write $\mathsf{EG}\varphi'[x, \mathsf{F}^{-1}x]$ under the general form

$$f[\mathsf{F}^{-1}x] \equiv \mathsf{EG}\Big((\alpha \wedge \mathsf{F}^{-1}x) \vee (\beta \wedge \neg \mathsf{F}^{-1}x) \vee \gamma\Big) \tag{A.2}$$

Then we only need the following rule:

$$\text{(R8)}\quad \mathsf{EG}\Big((\alpha \wedge \mathsf{F}^{-1}x) \vee (\beta \wedge \neg \mathsf{F}^{-1}x) \vee \gamma\Big) \equiv \bigvee \begin{array}{l} \mathsf{F}^{-1}x \wedge \mathsf{EG}(\alpha \vee \gamma) \\ \neg \mathsf{F}^{-1}x \wedge \left(\bigvee \begin{array}{l} \mathsf{EG}\big(\neg x \wedge (\beta \vee \gamma)\big) \\ \mathsf{E}(\neg x \wedge (\beta \vee \gamma))\mathsf{U}(x \wedge \mathsf{EG} \\ (\alpha \vee \gamma)) \end{array} \right) \end{array}$$

- Finally, the other cases are obvious, or can be reduced to what we saw through $\mathsf{AX}h \equiv \neg \mathsf{EX} \neg h$ and $\mathsf{A}g\mathsf{U}h \equiv \neg \mathsf{EG} \neg h \wedge \neg\Big(\mathsf{E}\neg h \mathsf{U} \neg g \wedge \neg h\Big)$.

We let the reader check that equations (R1)–(R8) are correct.[7] □

Lemma A.6. *If $f[x_1, \ldots, x_n]$ is a pure-future context, then $f[\mathsf{F}^{-1}x_1, \ldots, \mathsf{F}^{-1}x_n]$ is equivalent to a separated $f'[x_1, \mathsf{F}^{-1}x_1, \ldots, x_n, \mathsf{F}^{-1}x_n]$ with $f'[x_1, y_1, \ldots, x_n, y_n]$ a pure-future context.*

Proof. By induction on n, using Lemma A.5. □

Lemma A.7. *If $f[x_1, \ldots, x_n]$ is a pure-future context and $\psi_1^-, \ldots, \psi_n^-$ are pure-past CTL + F^{-1} formulas (i.e. formulas with F^{-1} as the only temporal combinator), then $f[\psi_1^-, \ldots, \psi_n^-]$ is equivalent to a separated CTL + F^{-1} formula.*

Proof. By induction on the maximum number of nested F^{-1}'s in the ψ_i^-'s and using Lemma A.6. □

Lemma A.8. *If $f[x_1, \ldots, x_n]$ is a pure-future context and $\psi_1, \ldots, \psi_n$ are separated CTL + F^{-1} formulas, then $f[\psi_1, \ldots, \psi_n]$ is equivalent to a separated CTL + F^{-1} formula.*

[7] This should not be too difficult. Alternatively, all five rules (R1)–(R5) could be replaced by a single general rule for which correctness is more difficult to assert.

Proof. Because it is separated, a ψ_i has the form $g_i^-[\varphi_{i,1}^+,\ldots,\varphi_{i,m_i}^+]$ with pure-future $\varphi_{i,j}^+$'s and pure-past $g_i^-[x_1,\ldots,x_{k_i}]$'s. Applying Lemma A.7 to $f[g_1^-[x_{1,1},\ldots,x_{1,m_1}],\ldots,g_n^-[x_{n,1},\ldots,x_{n,m_n}]]$ yields a separated $f'[x_{1,1},\ldots,x_{n,m_n}]$. Then $f'[\varphi_{1,1}^+,\ldots,\varphi_{n,m_n}^+]$ is separated and equivalent to $f[\psi_1,\ldots,\psi_n]$. □

Now we can prove Lemma 5.2 by structural induction on the $CTL+\mathsf{F}^{-1}$ formula and using Lemma A.8. Then Theorem 5.1 is easy to prove: once we have a separated formula, we repeatedly use

$$\mathsf{F}^{-1}\varphi \equiv_i \varphi$$

in boolean contexts.

Proof of Theorem 5.3

We slightly generalize the proof (from [10]) that $CTL^+ \equiv CTL$ to prove that $CTL^+ + \mathsf{F}^{-1} \equiv CTL+\mathsf{F}^{-1}$. Then Theorem 5.1 concludes the proof.

Lemma A.9. *Any $CTL^+ + \mathsf{F}^{-1}$ formula f is equivalent to a $CTL+\mathsf{F}^{-1}$ formula.*

Proof. By induction on f. The only interesting case is when f has the form

$$\mathsf{E}\left(\bigwedge_i (f_i \mathsf{U} g_i) \wedge \bigwedge_i \neg(f'_i \mathsf{U} g'_i) \wedge \bigwedge_i \mathsf{X} h_i \wedge \bigwedge_i \mathsf{F}^{-1} k_i \wedge \neg \mathsf{F}^{-1} k'\right)$$

where we have

$$f \equiv \mathsf{E}\left(\bigwedge_i (f_i \mathsf{U} g_i) \wedge \bigwedge_i \neg(f'_i \mathsf{U} g'_i) \wedge \bigwedge_i \mathsf{X} h_i\right) \wedge \bigwedge_i \mathsf{F}^{-1} k_i \wedge \neg \mathsf{F}^{-1} k'$$

Then $\mathsf{E}(\bigwedge_i (f_i \mathsf{U} g_i) \wedge \bigwedge_i \neg(f'_i \mathsf{U} g'_i) \wedge \bigwedge_i \mathsf{X} h_i)$ can be transformed into a $CTL+\mathsf{F}^{-1}$ using exactly the techniques for rewriting a CTL^+ formula into a CTL formula [10]. □

Proof of Theorem 5.4

We introduce an intermediary fragment:

$$L^\infty \ni f,g ::= a \mid f \wedge g \mid \neg f \mid \mathsf{EX} f \mid \mathsf{E} f \mathsf{U} g \mid \mathsf{A} f \mathsf{U} g \mid \mathsf{F}^{-1} f \mid \mathsf{E}\left(\mathsf{G} f \wedge \bigwedge_i \overset{\infty}{\mathsf{F}} g_i\right)$$

Then the proof of Theorem 5.4 is sketched as.

$$ECTL^+ + \mathsf{F}^{-1} \equiv L^\infty \equiv L^\infty_{sep} \equiv_i ECTL^+$$

The first part is easy. Clearly, from a syntactical viewpoint $ECTL{+}\mathsf{F}^{-1} \subseteq L^{\infty} \subseteq ECTL^{+}{+}\mathsf{F}^{-1}$, but we also have $ECTL^{+} + \mathsf{F}^{-1} \preceq_g L^{\infty}$:

Lemma A.10. *Any $ECTL^{+} + \mathsf{F}^{-1}$ formula is equivalent to a formula in L^{∞}.* is easy.

Lemma A.11 (Separation lemma for L^{∞}). *Any L^{∞} is equivalent to a separated L^{∞} formula.*

Proof. We proceed as in the Separation Lemma for $CTL + \mathsf{F}^{-1}$ (Lemma 5.2). In the crucial step where we prove that any $f[\mathsf{F}^{-1}x]$ with $f[x]$ a pure-future L^{∞} context can be separated, there is one more case to consider: when $f[x]$ has the form $\mathsf{E}(\mathsf{G}\varphi\wedge \bigwedge_i \psi_i)$. This needs one more rewrite rule.

$$\mathsf{E}\left(\begin{array}{l}\mathsf{G}\left[(\mathsf{F}^{-1}x\wedge\alpha)\vee(\neg\mathsf{F}^{-1}x\wedge\alpha')\vee\gamma\right]\\ \wedge\bigwedge_i \overset{\infty}{\mathsf{F}}\left[(\mathsf{F}^{-1}x\wedge\beta_i)\vee(\neg\mathsf{F}^{-1}x\wedge\beta_i')\vee\gamma_i\right]\end{array}\right) \equiv \begin{array}{l}\mathsf{F}^{-1}x\wedge\mathsf{E}\left[\mathsf{G}(\alpha\vee\gamma)\wedge\bigwedge_i\overset{\infty}{\mathsf{F}}(\beta_i\vee\gamma_i)\right]\\ \vee\neg\mathsf{F}^{-1}x\wedge\mathsf{E}\left[\mathsf{G}(\neg x\wedge(\alpha'\vee\gamma))\right.\\ \qquad\left.\wedge\bigwedge_i\overset{\infty}{\mathsf{F}}(\beta_i'\vee\gamma_i)\right]\\ \vee\neg\mathsf{F}^{-1}x\wedge\mathsf{E}\Big(\neg x\wedge(\alpha'\vee\gamma)\Big)\\ \mathsf{U}\Big(x\wedge\mathsf{E}\big[\mathsf{G}(\alpha\vee\gamma)\wedge\bigwedge_i\overset{\infty}{\mathsf{F}}(\beta_i\vee\gamma_i)\big]\Big)\end{array} \quad \square$$

After this we can conclude immediately because the pure-future fragment of L^{∞} is $ECTL^{+}$.

Proof of Theorem 5.5

To prove that $ECTL{+}\mathsf{F}^{-1} \not\preceq_i ECTL$, first observe that $\mathsf{E}(\overset{\infty}{\mathsf{F}}\, p\wedge\mathsf{G}q) \equiv_i \mathsf{E}\,\overset{\infty}{\mathsf{F}}\,(p\wedge\mathsf{G}^{-1}q)$. Now it is enough to prove that $\mathsf{E}(\overset{\infty}{\mathsf{F}}\, p\wedge\mathsf{G}q)$ cannot be expressed in $ECTL$. For this we consider the models $M_1, M_2, \ldots, N_1, N_2, \ldots$ described in Fig. 7. (They are inspired from the proofs that $\mathsf{E}\,\overset{\infty}{\mathsf{F}}\, p$ cannot be expressed in CTL and that $\mathsf{E}(\overset{\infty}{\mathsf{F}}\, p\wedge\overset{\infty}{\mathsf{F}}\, q)$ cannot be expressed in $ECTL$ [11].)

Clearly, for any $i = 1, \ldots,$

$$a_i \models \mathsf{E}(\overset{\infty}{\mathsf{F}}\, p\wedge\mathsf{G}q) \text{ and } \alpha_i \not\models \mathsf{E}(\overset{\infty}{\mathsf{F}}\, p\wedge\mathsf{G}q)$$

However, writing $|f|$ for the size of f, we show by induction on f (left to the reader) that

$$\alpha_i \models f \text{ iff } a_i \models f,$$

$$\beta_i \models f \text{ iff } b_i \models f,$$

$$\gamma_i \models f \text{ iff } c_i \models f.$$

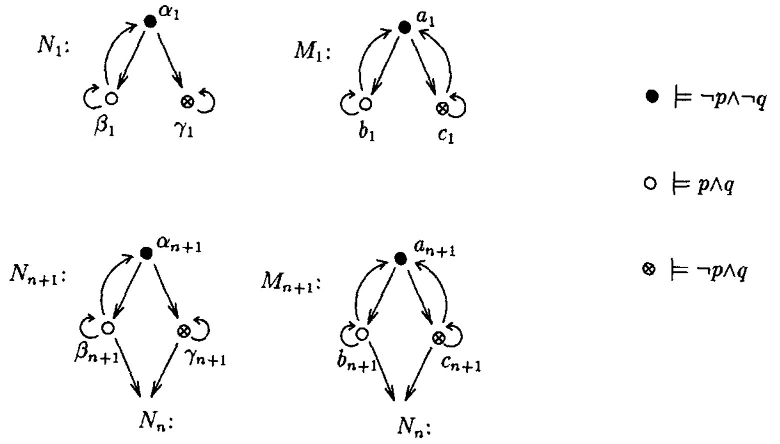

Fig. 3. $a_i, \alpha_i \models p \wedge q$, $b_i, \beta_i \models \neg p \wedge \neg q$, $c_i, \gamma_i \models \neg p \wedge q$.

for all $f \in ECTL$ and all $i \geqslant |f|$.

Then, to prove that $ECTL^+ \not\preceq_i ECTL + \mathsf{F}^{-1}$, we prove that $\mathsf{E}(\overset{\infty}{\mathsf{F}}\, p \wedge \overset{\infty}{\mathsf{F}}\, q)$ is not (initially) equivalent to any $ECTL + \mathsf{F}^{-1}$ formula. First, we observe that $\mathsf{E}(\overset{\infty}{\mathsf{F}}\, p \wedge \overset{\infty}{\mathsf{F}}\, q)$ cannot be expressed in the following pure-future fragment of $ECTL^+$:

$$L \ni f, g ::= a \mid f \wedge g \mid \neg f \mid \mathsf{EX} f \mid \mathsf{E} f \mathsf{U} g \mid \mathsf{A} f \mathsf{U} g \mid \mathsf{E}(\overset{\infty}{\mathsf{F}}\, f \wedge \mathsf{G} g)$$

Clearly $ECTL \subseteq L \subseteq ECTL^+$. The proof that $\mathsf{E}(\overset{\infty}{\mathsf{F}}\, p \wedge \overset{\infty}{\mathsf{F}}\, q)$ cannot be expressed in L can be done simply by enriching (left to the reader) the proof from [11] that $\mathsf{E}(\overset{\infty}{\mathsf{F}}\, p \wedge \overset{\infty}{\mathsf{F}}\, q)$ cannot be expressed in $ECTL$. The same models work for L as well.

Because $ECTL + \mathsf{F}^{-1} \subseteq L + \mathsf{F}^{-1}$, it is now sufficient to have a separation theorem for $L + \mathsf{F}^{-1}$, so that we shall arrive at

$$ECTL + \mathsf{F}^{-1} \subseteq L + \mathsf{F}^{-1} \equiv (L + \mathsf{F}^{-1})_{sep} \equiv_i L \prec ECTL^+$$

Lemma A.12 (Separation theorem for $L + \mathsf{F}^{-1}$). *Any $ECTL + \mathsf{F}^{-1}$ formula is equivalent to a separated $L + \mathsf{F}^{-1}$ formula.*

is a special case of Lemma A.11 and can be proved using the same transformations.

Proof of Theorem 6.7

The Normal Form Theorem [21] states that any PTL formula is initially equivalent to a formula of the form

$$\bigwedge_{i=1}^{n} (\overset{\infty}{\mathsf{F}}\, \varphi_i \vee \overset{\infty}{\mathsf{G}}\, \psi_i)$$

where the φ_i's and the ψ_i's are pure-past (linear-time) formulas.

With this, it is easy to translate any formula f in CTL^* into a globally equivalent $B(\overset{\infty}{\mathsf{F}}, \mathsf{X}^{-1}, \mathsf{S}, \wedge, \neg) + \mathsf{N}$ formula. The interesting case is when f is some $\mathsf{E}g$. Then we replace in g all subformulas of the form $\mathsf{E}h$ by new atomic propositions. This yields a PTL formula g' which can be written as some $\bigwedge_{i=1}^{n}(\overset{\infty}{\mathsf{F}}\,\varphi_i \vee \overset{\infty}{\mathsf{G}}\,\psi_i)$. Then

$$\mathsf{E}g' \equiv \mathsf{NE}\bigwedge_{i=1}^{n}(\overset{\infty}{\mathsf{F}}\,\varphi_i \vee \overset{\infty}{\mathsf{G}}\,\psi_i)$$

It now remains to replace the atomic propositions we introduced by their $B(\overset{\infty}{\mathsf{F}}, \mathsf{X}^{-1}, \mathsf{S}, \wedge, \neg) + \mathsf{N}$ equivalents, and we are done.

Acknowledgments

We would like to thank S. Pinchinat and the anonymous referees for their many helpful comments on this work.

References

[1] M. Ben-Ari, A. Pnueli and Z. Manna, The temporal logic of branching time, *Acta Inform.* **20** (1983) 207–226.

[2] M.C. Browne, E.M. Clarke and O. Grümberg, Characterizing finite Kripke structures in propositional temporal logic, *Theoret. Comput. Sci.* **59** (1988) 115–131.

[3] J.R. Burch, E.M. Clarke, K.L. McMillan, D.L. Dill and L.J. Hwang, Symbolic model checking: 10^{20} states and beyond, *Inform. and Comput.* **98** (1992) 142–170.

[4] A. Burrieza and I.P. deGuzmán. A new algebraic semantic approach and some adequate connectives for computation with temporal logic over discrete systems, *J. Appl. Non-Classical Logics* **2** (1992) 181–201.

[5] E. Chang, Z. Manna and A. Pnueli, Characterization of temporal property classes, in *Proc. 19th ICALP*, Vienna, Lecture Notes in Computer Science, Vol. 623 (Springer, Berlin, 1992) 474–486.

[6] E.M. Clarke and E.A. Emerson, Design and synthesis of synchronization skeletons using branching time temporal logic, in: *Proc. Logics of Programs Workshop*, Yorktown Heights, NY, Lecture Notes in Computer Science, Vol. 131 (Springer, Berlin, 1981) 52–71.

[7] E.M. Clarke, E.A. Emerson and A.P. Sistla, Automatic verification of finite-state concurrent systems using temporal logic specifications, *ACM Trans. Programming Languages Systems* **8** (1986) 244–263.

[8] E.A. Emerson, Temporal and modal logic, in: J. van Leeuwen, ed., *Handbook of Theoretical Computer Science*, Vol. B, ch. 16 (Elsevier, Amsterdam, 1990) 995–1072.

[9] E.A. Emerson and E.M. Clarke, Characterizing correctness properties of parallel programs using fixpoints, in: *Proc. 7th ICALP, Noordwijkerhout*, Lecture Notes in Computer Science, Vol. 85 (Springer, Berlin 1980) 169–181.

[10] E.A. Emerson and J.Y. Halpern, Decision procedures and expressiveness in the temporal logic of branching time, *J. Comput. System Sci.* **30** (1985) 1–24.

[11] E.A. Emerson and J.Y. Halpern. "Sometimes" and "Not Never" revisited: On branching versus linear time temporal logic, *J. ACM*, **33** (1986) 151–178.

[12] D. Gabbay, The declarative past and imperative future: executable temporal logic for interactive systems, in: *Proc. Temporal Logic in Specification*, Altrincham, UK, Lecture notes in Computer Science, Vol. 398 (Springer, Berlin, 1987) 409–448.

[13] D. Gabbay, A. Pnueli, S. Shelah and J. Stavi, On the temporal analysis of fairness, in: *Proc. 7th ACM Symp. Principles of Programming Languages*, Las Vegas, NV, (1980) 163–173.

[14] T. Hafer and W. Thomas, Computation tree logic CTL* and path quantifiers in the monadic theory of the binary tree, in: *Proc. 14th ICALP*, Karlsruhe, Lecture Notes in Computer Science, Vol. 267 (Springer, Berlin, 1987) 269–279.

[15] Y. Kesten, Z. Manna, H. McGuire and A. Pnueli, A decision algorithm for full propositional temporal logic, in: *Proc. CAV'93*, Elounda, Greece, Lecture Notes in Computer Science, Vol. 697 (Springer, Berlin, 1993) 97–109.

[16] F. Laroussinie, Logique Temporelle avec Passé pour la Spécification et la Vérification des Systèmes Réactifs, Thèse de Doctorat, I.N.P. de Grenoble, France, November 1994.

[17] F. Laroussinie, S. Pinchinat and Ph. Schnoebelen, Translation results for modal logics of reactive systems, in: *Proc. AMAST'93*, Enschede, NL (Springer, Berlin, 1993) 299–310.

[18] O. Lichtenstein, A. Pnueli and L. Zuck, The glory of the past, in: *Proc. Logics of Programs Workshop, Brooklyn*, Lecture Notes in Computer Science, Vol. 193 (Springer, Berlin, 1985) 196–218.

[19] Z. Manna and A. Pnueli, The anchored version of the temporal framework, in: *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, Noordwijkerhout, Lecutre Notes in Computer Science, Vol. 354 (Springer, Berlin, 1989) 201–284.

[20] Z. Manna and A. Pnueli, A hierarchy of temporal properties, in: *Proc. 9th ACM Symp. Principles of Distributed Computing*, Quebec City, Canada (1990) 377–408.

[21] Z. Manna and A. Pnueli, *The Temporal Logic of Reactive and Concurrent Systems*, Volume I:, *Specification* (Springer, Berlin 1992).

[22] L.E. Moser, P.M. Melliar-Smith, G. Kutty and Y.S. Ramakrishna, Completeness and soundness of axiomatizations for temporal logics without next, *Fundam. Inform.* (1993), to appear.

[23] A. Prior, *Past, Present, and Future* (Clarendon Press, Oxford, 1967).

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

[25] C. Stirling, Modal and temporal logics, in: S. Abramsky, D. Gabbay and T. Maibaum, eds., *Handbook of Logic in Computer Science* (Oxford Univ. Press, 1992) 477–563.

[26] M. Vardi, A temporal fixpoint calculus, in: *Proc. 15th ACM Symp. Principles of Programming Languages*, San Diego, CA, (1988) 250–259.

[27] A. Zanardo and J. Carmo, Ockhamist computational logic: Past-sensitive necessitation in CTL*, *J. Logic Comput.* **3** (1993) 249–268.

[28] L. Zuck, Past temporal logic, Ph.D. Thesis, Weizmann Institute, Rehovot, Israel, August 1986.