A Point-Interval Logic for Temporal Knowledge Representation

Manuel Enciso, Inmaculada P. de Guzmán, and Carlos Rossi

E.T.S.I. Informática. Universidad de Málaga
Campus de Teatinos. 29071 Málaga, Spain
enciso@lcc.uma.es {guzman,rossi}@ctima.uma.es

Abstract. This paper may be classified within the area of the representation of temporal knowledge and, specifically, within its specification and processing. We present a point and interval logic, LNint-e, which integrates approaches that traditionally appear as conflicting in literature: LNint-e permits point and interval expressions, absolute and relative treatment of time and, notably, the distinction between classes and executions of events. Unlike temporal modal logics existing in bibliography, LNint-e allows to express connectives related to any number of intervals as Moszkowski's *chop* connective. Besides, LNint-e uses the *topological semantics* to avoid the use of first order logic as meta-theory.
We introduce LNint-e to be not only a powerful tool for specification, but also to be a preliminary step in the design of new temporal reasoning methods. In order to do that, as a remarkable contribution of this work, we define reduction transformations that allow us to reduce, in linear time, the complexity of LNint-e formulae.
Finally, we want to mention that the usefulness of LNint-e has been verified by its successful application to dynamic behaviour modelling, specially in the areas of user interfaces and UML statechart diagrams.

1 Introduction

The representation of temporal knowledge is an area of a great importance, contrasted by its many applications in Artificial Intelligence. This paper deals with logic-based time representation. Among the applications which have caused the theoretical development included in this paper, we place particular emphasis on two of them [1] for which it has already shown its usefulness: the modelling of the dynamic behavior of user interfaces [3] and systems [4] (by means of statechart diagrams). In this kind of diagrams there appear different elements (each one of them with different temporal connotations):

– *States*, which occur during a period of time. They are characterized by some properties that are true at every instant of the period.

[1] When we look for a temporal representation language for these applications, we find in the literature some approaches that may be classified in three large groups: modal temporal logics, reified logics and temporal arguments logics. Our work is situated among the first ones, because they are more natural and expressive than others.

– *Transitions*, which are activated by instantaneous events. Statechart events can be described relatively or absolutely (dates).
– *Operations*, that must be executed by the system. Operations are executed *completely* at one period of time.

When we analyze the existing works about temporal logic, we detect artificially opposing views, as the selection of the temporal primitives (points or intervals) or the treatment of time (absolute or relative). Our approach, consolidated in [2], combines the precedent ones.

In this paper we present the LNint-e logic, a point modal logic extended to express interval situations,[2] and with the chance of treating time either in an absolute and in a relative way. A differentiating characteristic of this logic is its topological semantics, strongly based on the flow of time and which avoid the use of the first order logic as meta-theory. LNint-e is based on LN point logic [11]. LNint-e has fully expressive power as regards point expressions and at least the same expressive power as Allen's [1] and Halpern and Shoham's [5] logics as regards interval expressions. LNint-e can be considered as an improvement of LNint logic [2] with the addition of the following characteristics:

– LNint expressivity is increased, differentiating between *classes* and *executions* of events. This technique has been previously used by several authors [1, 9, 6]) and it permits us to express temporal situations of abutment and overlapping of executions of the same event.
– We provide interval temporal connectives with an homogeneous structure, whichever be their type.
– Both syntax and semantics are remarkably simplified.

The expressive power of LNint-e lead us to an unavoidable complexity in the initial description of the language. However, as we will see, we can palliate this problem by means of efficient transformations of simplification that allow to ensure its usefulness in computing applications. Particularly, we will highlight those transformations which allows the elimination of some binary temporal connectives (those which entail a greater difficulty of processing by any prover or deductive mechanism). These transformations have also a particular interest in order to obtain a definition of the normal form for LNint-e formulas [12], with the corresponding advantages on specification tasks.

In the following section we present the syntax of LNint-e and in section 3, its semantics. Section 4 illustrates the expressive ability of LNint-e with regard to different kinds of expressions: absolute or relative, point or interval expressions, etc. In section 5 we show the transformations of simplification and, finally, in section 6 we include an illustrative example of the good characteristics of LNint-e for the representation of temporal knowledge.

[2] This approach makes that both point expressions and interval expressions be evaluated at an instant, the *current instant*. This way, the concept of *current interval*, common in interval modal logics and difficult to treat computationally, is abandoned.

2 Syntax of LNint-e

At the syntactic level, we begin by considering a component to collect assertions about points and dates. We emphasise that point expressions can be affirmed over intervals which we call *hereditary* interval expressions [3].

To complement the treatment of interval expressions, we need another component to collect events. We use atomic events in a similar sense to that used by Allen [1], i.e., expressions about intervals which are not true at the subintervals nor, more specifically, at the points of the interval over which the expression is affirmed [4]. Event expressions [5] will be called *non-hereditary* interval expressions. In LNint-e, we distinguish between event *classes* (or event types) and event *executions* (or tokens) that represent each of the concrete occurrences of an event class [6]. Therefore,

The alphabet de LNint-e includes symbols to represent each of these kinds of components. Specifically:

- An enumerable set $\Omega_{ins} = \{p, q, \ldots, p_n, q_n, \ldots\}$ of point atoms.
- A set $\underline{\mathbb{Z}} = \{\underline{m} \mid m \in \mathbb{Z}\}$ of *date* atoms, used to name known instants.
- An enumerable set $\Omega_{ev} = \{\alpha, \beta, \ldots, \alpha_n, \beta_n, \ldots\}$ of *event class* atoms. To distinguish the executions of an event class, the alphabet includes:
  - A set $Labels = \{l, l_1, l_2, \ldots, l_n, \ldots\}$ of *labels*.
  - The symbol *, which acts as a wild card for event execution labels.
- The symbols $\top$ and $\bot$, to denote truth and falsity, respectively.
- The symbols $\uparrow, \downarrow y^{\longrightarrow}$.
- The symbols of boolean connectives $\neg, \wedge$ and $\vee$.
- The symbols of binary temporal connectives of points $\preccurlyeq$ and $\succcurlyeq$.

2.1 Language of LNint-e

In order to have a uniform treatment of every kind of atoms, we characterize the events by means of their starting, ending and course instants.

Let $EventExecs = \{\alpha_l \mid \alpha \in \Omega_{ev}, l \in Labels\}$ be the set of all event executions. For each $\alpha_l \in EventExecs$ we use the following notation:

- $\uparrow\alpha_l$ is read as *this is the starting instant of execution l of event α.*
- $\downarrow\alpha_l$ is read as *this is the ending instant of execution l of event α.*
- $\overrightarrow{\alpha_l}$ is read as *this is an instant at which execution l of event α is in process.*

[3] As *states* in an statechart diagram

[4] As *operations* in an statechart diagram.

[5] The LNint-e term "event" cannot be confused with the statechart term *event*. We follow the notation used in bibliography, where the same term is used for these different concepts.

[6] We emphasise that a given execution of an event class will be true (at most) at an unique interval of the flow of time.

Moreover, LNint-e language provide expressions which refer to an *anonymous execution* [7] of an event class. Given an event class, $\alpha \in \Omega_{ev}$, the notation for the anonymous execution thereof is α_*.

Definition 1. *The language $\mathcal{LN}int\text{-}e$ is the inductive closure generated over the base set*

$$\Omega_{ins}^{ev\text{-}e} = \Omega_{ins} \cup \underline{\mathbb{Z}} \cup \{\top, \bot\} \cup \{\uparrow\alpha_l, \downarrow\alpha_l, \overrightarrow{\alpha_l} \mid \alpha_l \in EventExecs\} \cup \{\uparrow\alpha_*, \downarrow\alpha_*, \overrightarrow{\alpha_*} \mid \alpha \in \Omega_{ev}\}$$

by the monary connective $\neg$ and the binary connectives $\wedge, \vee, \preccurlyeq$ and $\succcurlyeq$.

So, $\Omega_{ins}^{ev\text{-}e}$ is the set of atomic expressions that make sense declared on instants.

3 Semantics of LNint-e

The semantics of LNint-e are an extension of the topological semantics introduced in [11] for LN logic. The extension carried out is not trivial: the philosophy of the topological semantics, originally created for the management of instants, has been adapted to manage intervals, keeping its good properties.

We consider $(\mathbb{Z}, \leq)$ as the flow of time. The key concepts in our semantics, denoted m_{tA}^+ and m_{tA}^-, are defined as follows:

Definition 2. *If A is a wff of LNint-e and $t \in \mathbb{Z}$, we define:*

$$m_{tA}^+ = min\{t' \in \mathbb{Z} \mid t' > t \text{ and } A \text{ is true at } t'\}$$
$$m_{tA}^- = max\{t' \in \mathbb{Z} \mid t' < t \text{ and } A \text{ is true at } t'\}$$

In other words, m_{tA}^+ is the first instant after t in which A will be true, and m_{tA}^- is the last instant before t in which A was true. We agree that $min\, \varnothing = +\infty$ and $max\, \varnothing = -\infty$. Likewise, we extend the order of $\mathbb{Z}$ over the set $\{-\infty, +\infty\} \cup \mathbb{Z}$ in the usual manner. We denote $Int(\mathbb{Z}) = \{[t_1, t_2] \mid t_1, t_2 \in \mathbb{Z}, t_1 < t_2\}$ as the set of closed finite intervals of $\mathbb{Z}$ with different start and end points.

Definition 3. *A* **temporal interpretation** *for LNint-e is a tern $\mathcal{I} = \langle H_{ev}, H_{exec}, h\rangle$ where*

- *H_{ev}: $\Omega_{ev} \longrightarrow 2^{Int(\mathbb{Z})}$, where $H_{ev}(\alpha)$ is the set of all the intervals at which the executions of the class event α take place.*
- *H_{exec}: $EventExecs \longrightarrow Int(\mathbb{Z})$ associates each event execution α_l with the only interval of $Int(\mathbb{Z})$ where α_l holds. This function satisfies the following conditions:*
  1. *$H_{exec}(\alpha_l) \in H_{ev}(\alpha)$ for all $\alpha_l \in EventExecs$*
  2. *For all $\alpha \in \Omega_{ev}$, and any pair of labels, $l, l' \in Labels$, $l \neq l'$ if $H_{exec}(\alpha_l) = [t_1, t_2]$ and $H_{exec}(\alpha_{l'}) = [t_1', t_2']$ then $t_2 - t_1 = t_2' - t_1'$.*

[7] Whose particular label is either unknown or not interesting to be specified.

– $h\colon \Omega_{ins}^{ev\text{-}e} \longrightarrow 2^{\mathbb{Z}}$ *associates each atom of* $\Omega_{ins}^{ev\text{-}e}$ *with a subset of* $\mathbb{Z}$ *that satisfies the following conditions:*

1. $h(\bot) = \varnothing, \quad h(\top) = \mathbb{Z}$ *and* $h(\underline{t}) = \{t\}$, *for all* $\underline{t} \in \underline{\mathbb{Z}}$
2. *For all* $\alpha_l \in EventExecs$, *if* $H_{exec}(\alpha_l) = [t_1, t_2]$, *then* $h(\uparrow\alpha_l) = \{t_1\}$, $h(\downarrow\alpha_l) = \{t_2\}$ *and* $h(\overrightarrow{\alpha_l}) = (t_1, t_2)$.
3. *For all* $\alpha \in \Omega_{ev}$, *we have*
   - 3.1 $h(\uparrow\alpha_*) = \{t \mid t \in h(\uparrow\alpha_l), l \in Labels\}$
   - 3.2 $h(\downarrow\alpha_*) = \{t \mid t \in h(\downarrow\alpha_l), l \in Labels\}$
   - 3.3 $h(\overrightarrow{\alpha_*}) = \{t \mid t \in h(\overrightarrow{\alpha_l}), l \in Labels\}$

The extension of $\mathcal{I} = \langle H_{ev}, H_{exec}, h\rangle$ to $\mathcal{LN}int\text{-}e$, also denoted $\mathcal{I}$, only requires the extension of h to $\mathcal{LN}int\text{-}e$ as follows: if $A, B \in \mathcal{LN}int\text{-}e$

$$h(\neg A) = \mathbb{Z} \setminus h(A); \quad h(A \vee B) = h(A) \cup h(B); \quad h(A \wedge B) = h(A) \cap h(B)$$
$$h(A \preccurlyeq B) = \{t \in \mathbb{Z} \mid m^+_{tA} < +\infty \text{ and } m^+_{tA} \leq m^+_{tB}\}$$
$$h(A \succcurlyeq B) = \{t \in \mathbb{Z} \mid m^-_{tA} > -\infty \text{ and } m^-_{tA} \geq m^-_{tB}\}$$

where $m^+_{tA} = \min\big((t, +\infty) \cap h(A)\big)$ and $m^-_{tA} = \max\big(-\infty, t) \cap h(A)\big)$

Definition 4. *Let* A *be a wff of* $\mathcal{LN}int\text{-}e$, A *is said to be* **satisfiable** *if there exists* $t \in \mathbb{Z}$ *and a temporal interpretation* $\mathcal{I} = \langle H_{ev}, H_{exec}, h\rangle$ *such that* $t \in h(A)$. A *is said to be* **valid in a temporal interpretation** $\mathcal{I} = \langle H_{ev}, H_{exec}, h\rangle$, *denoted* $\models_{\mathcal{I}} A$, *if* $h(A) = \mathbb{Z}$. A *is said to be* **valid**, *denoted* $\models A$, *if* $\models_{\mathcal{I}} A$ *for all temporal interpretation* $\mathcal{I}$. *Two wffs* A *and* B *of* $\mathcal{LN}int\text{-}e$ *are* **equivalent**, *denoted* $A \equiv B$, *if* $h(A) = h(B)$, *for all temporal interpretation* $\mathcal{I} = \langle H_{ev}, H_{exec}, h\rangle$.

4 Expressivity of LNint-e

H. Kamp's [7] ensures that US logic is fully expressive. LNint-e has the same expressive power since the connectives U [8] and S [9] are definable in LNint-e in the following way:

$$U(A, B) \stackrel{def}{=} A \preccurlyeq \neg B \quad \text{and} \quad S(A, B) \stackrel{def}{=} A \succcurlyeq \neg B$$

We will distinguish, both for point expressions and interval expressions, two kinds of connectives: those that represent an *absolute* treatment of time (relating statements with dates or concrete positions along the line of time) and those that represent a *relative* treatment (analyzing the mutual position of pairs of statements, independently of their absolute temporal position).

From now on, we will refer only to the future connectives. The past connectives are obtained using the *mirror* rules as usual.

[8] $U(A, B)$ is read as *sometime in the future* A *will occur and* B *will be true from now to* A.

[9] $S(A, B)$ is read as *sometime in the past* A *occurred and* B *was true since* A.

4.1 Specifying with LNint-e

LNint-e point temporal connectives ($\preccurlyeq$ and $\succcurlyeq$) describe future and past non-strict precedence and, following Kröger [8] notation, they are *strong* connectives[10].

The language of LNint-e allows us to define in a convenient way (in terms relative to computing) the following types of connectives:

Strict precedence and simultaneity connectives:

- Strong connectives of strict precedence (denoted $\prec$ and $\succ$):
  $A \prec B$ is read as *in the future A, and the next occurrence of A will be before the next occurrence of B*. Therefore, its semantics is the following:
  $$h(A \prec B) = \{t \in \mathbb{Z} \mid m^+_{tA} < m^+_{tB}\}$$
  .
- Simultaneity connectives (denoted $\approx^+$ and $\approx^-$):
  $A \approx^+ B$ is read as *sometime in the future A and B, and the next occurrence of A will be simultaneous with the next occurrence of B.* Therefore, its semantics is
  $$h(A \approx^+ B) = \{t \in \mathbb{Z} \mid \ m^+_{tA} = m^+_{tB} < +\infty\}$$

It is a trivial task to prove that the formal LNint-e definitions of the above connectives are the following[11]:

$$A \approx^+ B \stackrel{def}{=} A \preccurlyeq B \wedge B \preccurlyeq A$$
$$A \prec B \ \stackrel{def}{=} A \preccurlyeq B \wedge \neg(B \preccurlyeq A)$$

Besides, the fully expressive power of LNint-e allows us to define the following temporal connectives:

Standard temporal connectives:

- FA is read as *A will be true sometime in the future.* $FA \stackrel{def}{=} A \preccurlyeq \bot$
- $\oplus A$ is read as *A will be true tomorrow.* $\oplus A \stackrel{def}{=} A \preccurlyeq \top$
- GA is read as *A will be true always in the future.* $GA \stackrel{def}{=} \neg F \neg A$

Kröger connectives :

- $A\,atnext\,B$ is read as *A will be true at the next instant of time that B occurs.*
  $A\,atnext\,B \stackrel{def}{=} B \approx^+ (A \wedge B)$
- $A\,atlast\,B$ is read as *A was true at the last instant of time that B occurred.*
  $A\,atlast\,B \stackrel{def}{=} B \approx^- (A \wedge B)$

[10] The connective $\preccurlyeq$ (resp. $\succcurlyeq$) is strong if $A \preccurlyeq B$ (resp. $A \succcurlyeq B$) implies *sometimes in the future (resp. past) A.*

[11] Moreover, we have that the the sets of connectives $\{\preccurlyeq, \succcurlyeq\}$, $\{\prec, \succ\}$ and $\{\approx^+, \approx^-\}$ are equivalent [11].

Connectives for absolute temporal treatment:

- The point connective: $A\,at\,\underline{m} =_{def} (A \wedge \underline{m}) \vee A\,atnext\,\underline{m} \vee A\,atlast\,\underline{m}$, which allows us to know if A is true at a particular instant, independently of the instant from which we speak. $A\ at\ \underline{m}$ will be true at every instant of the flow of time or at none of them. Thus, its semantics are: $h(A\,at\,\underline{m}) = \mathbb{Z}$ if $m \in h(A)$ and $h(A\,at\,\underline{m}) = \varnothing$ in another case.
- The event connective for concrete executions

$$\alpha_l\ at_{ev}\ [\underline{m}, \underline{n}] \stackrel{def}{=} (\uparrow\alpha_l \wedge \downarrow\alpha_l \approx^+ \underline{n})\ at\ \underline{m}$$

  where $\alpha_l \in EventExecs$ and $m < n$. $\alpha_l\ at_{ev}\ [\underline{m}, \underline{n}]$ is read as: *the execution α_l occurs exactly at the interval $[m, n]$.*
- The event connective for anonymous executions

$$\alpha_*\ at_{ev}\ [\underline{m}, \underline{n}] \stackrel{def}{=} (\uparrow\alpha_*\ \wedge\ \neg\overrightarrow{\alpha_*} \wedge\ \downarrow\alpha_* \approx^+ \underline{n})\ at\ \underline{m} \wedge\ \neg\overrightarrow{\alpha_*}\ at\ \underline{n}$$

  where $\alpha \in \Omega_{ev}$ and $m < n$ [12].
  The connective $\alpha_*\ at_{ev}\ [\underline{m}, \underline{n}]$ is read: *en $[m, n]$ an execution of the event α is carried out, and this one does not overlap any other execution of α.*

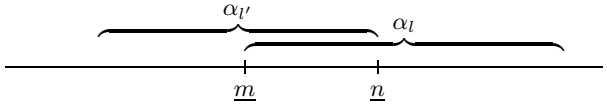
Connectives for relative temporal event treatment:

All interval modal logics included in bibliography establish the reference of a current interval from which formulas are evaluated. In LNint-e we consider the current instant and we define the current interval as an interval to which the current instant belongs. As it is shown in [1], a number of thirteen temporal relations can occur between two intervals. In LNint-e it suffices to define seven of them as primitives (an exhaustive description can be seen in [12]), among which we remark the following. If $\alpha_l, \beta_{l'} \in EventExecs$,

- $eq^+(\alpha_l, \beta_{l'})$ holds at an instant t if and only if α_l and $\beta_{l'}$ are true [13] at an interval $[t_1, t_2]$, where $t < t_1 < t_2$. The definition in LNint-e is:

$$eq^+(\alpha_l, \beta_{l'}) \stackrel{def}{=} \uparrow\alpha_l \approx^+ \uparrow\beta_{l'}\ \wedge\ \downarrow\alpha_l \approx^+ \downarrow\beta_{l'}$$

[12] The modification included in the precedent definition imposes that either $\underline{m}$ nor $\underline{n}$ are an instant of course of no execution of α (this exigency is not contemplated in the definition of connective for concrete executions previously defined). This way, we avoid that $\alpha_*\,at_{ev}\,[\underline{m}, \underline{n}]$ be true in situations like the one represented in the following figure:



[13] Notice that we must not refer to the *next* occurence of an event, because event executions take place only once.

- $ab^+(\alpha_l, \beta_{l'})$ holds at an instant t if and only if α_l is true at an interval $[t_1, t_2]$ and $\beta_{l'}$ is true at an interval $[t_2, t_3]$ adjacent to $[t_1, t_2]$, where $t < t_1 < t_2 < t_3$. The definition in LNint-e is:

$$ab^+(\alpha_l, \beta_{l'}) \stackrel{def}{=} F \uparrow\alpha_l \wedge \downarrow\alpha_l \approx^+ \uparrow\beta_{l'}$$

Analogously, it is possible to define the corresponding connectives for anonymous executions.

Hereditary Expressions:

We can extend the concepts of start, course and end point to point assertions by defining for all $A \in \mathcal{L}_{ins}$:

$\uparrow A =_{def} \ominus\neg A \wedge A$; $\downarrow A =_{def} A \wedge \oplus\neg A$ and $\overrightarrow{A}^+ =_{def} A \wedge \oplus A$

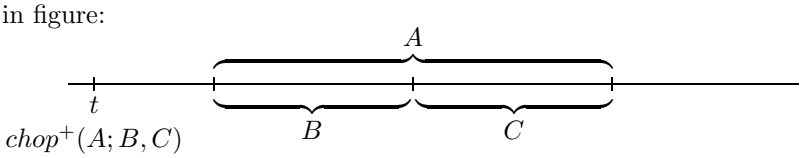
Thus, we might define the following connective:

$$Int^+(A) \stackrel{def}{=} (\oplus A \rightarrow \neg A) \wedge (A \approx^+ \overrightarrow{A}^+) \wedge F \downarrow A$$

$Int^+(A)$ is read as *in the future A will occur, and the next time that A occurs, A holds at a finite closed interval.*

The operator *chop* is definable in LNint-e.

A remarkable property of LNint is that, unlike Halpern and Shoham's logic and other interval modal logics, we can define the operator *chop* introduced by Moszkowski in [10]. This way, it is possible to express relations among three intervals. As it is well-known, the connective *chop* represents the temporal relation shown in figure:



Its definition in LNint is as follows: Given three (point) formulas $A, B, C \in \mathcal{L}_{ins}$, we define:

$chop^+(A; B, C) =_{def}$

$$Int^+(A) \wedge Int^+(B) \wedge Int^+(C) \wedge (A \approx^+ B) \wedge (\downarrow A \approx^+ \downarrow C) \wedge ab^+(B, C)$$

5 Working with LNint-e. Reduction transformations

The expressive power of LNint-e, analyzed in previous section, of theoretical interest by itself, does not guarantee its usefulness in computing as, though it seems to be optimum for specification tasks (because of the transparency of its semantical reading), the management of the introduced binary temporal connectives contributes a great complexity for computing. This is the reason why, as we said in the introduction, our following aim in view of the applications was

to have at our disposal reduction transformations for the wffs of LNint-e. The results, as we will see, are remarkable and ratify the usefulness of our contribution. Transformations now introduced are applied to formulas with nestings of monary and binary temporal connectives. Concretely, these transformations reduce binary connectives in the following two types of formulas:

i) Formulas in which one of the arguments of a binary temporal connective is under the scope of a monary temporal connective.
ii) Formulas in which the argument of a monary temporal connective is based on a binary temporal connective

Proposition 1. *Let A, B be wffs of LNint-e, then:*

(a) $FA \prec B \equiv F \oplus A \wedge \oplus\neg B$; $\quad FA \preccurlyeq B \equiv F \oplus A$.
(b) $A \prec FB \equiv FA \wedge G \oplus \neg B$; $\quad A \preccurlyeq FB \equiv (G \oplus \neg B \vee \oplus A) \wedge FA$.
(c) $GA \prec B \equiv F(GA \wedge \neg B)$; $\quad GA \preccurlyeq B \equiv FGA \wedge G[\neg B, A]$.
(d) $A \prec GB \equiv F[A, \neg B]$; $\quad A \preccurlyeq GB \equiv \oplus A \vee F[A, \ominus\neg B]$.
(e) Let $\gamma \in \{FG, GF\}$, then:
$\gamma A \preccurlyeq B \equiv \gamma A$; $\quad \gamma A \prec B \equiv \gamma A \wedge \oplus\neg B$;
$A \prec \gamma B \equiv \bar{\gamma}\neg B \wedge FA$; $\quad A \preccurlyeq \gamma B \equiv (\bar{\gamma}\neg B \vee \oplus A) \wedge FA$.

Proof. We prove some of the items. The rest of proofs are similar:

(a) For all temporal interpretation h, we have that $t \in h(FA \preccurlyeq B)$ if and only if $m^+_{tFA} \leq m^+_{tB}$ and $m^+_{tFA} < +\infty$. This situation is fulfilled if and only if $m^+_{tFA} = t + 1$, i.e. $t \in h(F \oplus A)$ and, therefore, $FA \preccurlyeq B \equiv F \oplus A$.
(e) For all temporal interpretation h, we have that $t \in h(FGA \preccurlyeq B)$ if and only if $m^+_{tFGA} \leq m^+_{tB}$ and $m^+_{tFGA} < +\infty$. This situation is fulfilled if and only if $m^+_{tFGA} = t + 1$, i.e. $t \in h(FGA)$. Therefore, we conclude $\gamma A \preccurlyeq B \equiv \gamma A$.

Proposition 2. *Let A, B be wffs of LNint-e, then:*

(a) $G(A \prec B) \equiv GFA \wedge G \oplus \neg B$; $\quad G(A \preccurlyeq B) \equiv GFA \wedge G \oplus (B \rightarrow A)$
(b) $F(A \preccurlyeq B) \equiv F \oplus A$; $\quad F(A \prec B) \equiv F \oplus (A \wedge \neg B)$.
(c) $FG(A \prec B) \equiv GFA \wedge FG\neg B$; $\quad FG(A \preccurlyeq B) \equiv GFA \wedge FG(B \rightarrow A)$
(d) $GF(A \preccurlyeq B) \equiv GFA$; $\quad GF(A \prec B) \equiv GF(A \wedge \neg B)$

Proof. We prove some of the items. The rest of proofs are similar:

(a) $t \in h(G(A \prec B))$ if and only if
for all $x \geq t + 1$, $m^+_{xA} < \infty$ and $m^+_{xA} < m^+_{xB}$, if and only if
for all $x \geq t + 1, x \in h(FA)$ and $(x, m^+_{xA}] \subseteq (x, m^+_{xB})$, if and only if
for all $x \geq t + 1$, $x \in h(FA)$ and we have the following two cases:
if $x + 1 \in h(\neg A)$ then $x + 1 \in h(\neg B)$
if $x + 1 \in h(A)$ then $x + 1 \in h(\neg B)$ if and only if
for all $x \geq t + 1$, $x \in h(FA)$ and $x \in h(\oplus\neg B)$ if and only if
$t \in h(GFA)$ and $t \in h(G \oplus \neg B)$ if and only if
$t \in h(GFA \wedge G \oplus \neg B)$
(c) Using item (a) we have that:
$FG(A \prec B) \equiv F(\, G(A \prec B)\,) \equiv F(\, GFA \wedge G \oplus \neg B\,) \equiv$
$FGFA \wedge FG \oplus \neg B \equiv GFA \wedge FG\neg B$

6 Example

We conclude by illustrating the good characteristics of LNint-e for specifications. The following example shows a defined connective that formalizes the initial transition of a statechart diagram. The connective represents the transition (fired by a creation event) from the initial state to a target state and the execution of its entry action and activity[14]. The definition of the connective is the following:

$$\begin{aligned}\mathsf{InitialTrans}(\mathsf{CreationEvent}, \mathsf{TargetState}, \mathsf{Activity}_{\mathsf{Target}*}, \mathsf{EntryAction}_{\mathsf{Target}*}) \stackrel{def}{=} \\ \mathsf{G}\Big((\mathsf{InitialState} \wedge \ \mathsf{CreationEvent}) \rightarrow \\ [\mathsf{ab}^{+}(\mathsf{EntryAction}_{\mathsf{Target}*}, \mathsf{TargetState}) \wedge \\ \mathsf{ab}^{+}(\mathsf{EntryAction}_{\mathsf{Target}*}, \mathsf{Activity}_{\mathsf{Target}*})]\Big)\end{aligned}$$

References

1. J. F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, jul 1984.
2. I. Pérez de Guzmán and C. Rossi. LNint: a temporal logic that combines points and intervals and the absolute and relative approaches. *Journal of the IGPL*, 3(5), 1995.
3. M. Enciso, J. M. Frías, and C. Rossi. Formalización de interfaces de usuario usando lógica temporal. In *INTERACCIÓN 2001, 2nd Congreso Internacional de Interacción Persona-Ordenador*, 2001.
4. M. Enciso, I. P. de Guzmán, and C. Rossi. Using temporal logic to represent dynamic behaviour of UML statecharts. In *ECOOP 2002 Workshop on Integration and Transformation of UML Models*, 2002.
5. J. Halpern and Y. Shoham. A propositional modal logic of time intervals. *Journal of the ACM*, 38(4):935–962, October 1991.
6. B. A. Haugh. Non-standard semantics for the method of temporal arguments. In *Proceedings de la International Joint Conference on Artificial Intelligence IJCAI'87*, pages 449–455, 1987.
7. H. Kamp. *Tense logic and theory of linear orders.* PhD thesis, University of california, Los Angeles, 1968.
8. H. Kroger. *Temporal Logic of Programs.* Theoretical Computer Science. Springer Verlag, 1987.
9. D. V. McDermott. A temporal logic for reasoning about processes and plans. *Cognitive Science*, 6:101–155, 1982.
10. B. C. Moszkowski. *Reasoning about Digital Circuits.* PhD thesis, Standford University, 1983.
11. I. Pérez de Guzmán and A. Burrieza. A new algebraic semantic approach and some adequate connectives for computation with temporal logic over discrete time. *Journal of Applied Non-Classical Logic*, 2, 1992.
12. C. Rossi. *Lógica Temporal de Intervalos. Formalización de Diagramas de Estados.* PhD thesis, Universidad de Málaga, Spain, 2001.

[14] An exhaustive specification can be found in [12].