

Just one approach for several temporal logics in Computing: the topological semantics

Inma P. de Guzmán and Manuel Enciso and Carlos Rossi*

Universidad de Málaga. Pza El Ejido s/n. 29013 Málaga
{gimac}@tecma1.ctima.uma.es

Abstract

In this work we present an unified frame for temporal logic that allows us to consider

- discrete time and continuous time,
- points and intervals jointly, and
- the absolute and relative approaches.

in an adequate way for computing. This unified frame is built up from a new kind of semantics that we introduce in this work: the *topological semantics*, which avoids the use of first order logic as the base of the semantics and allows to develop natural connectives that relates to common properties in real systems.

We present a point logic over discrete time (the LN logic [5]), a point logic over continuous time (the IRLN logic [7]), and an interval-point logic over discrete time (the LNint logic [13]).

Key words: Temporal logic and ontologies, continuous and discrete time, point and interval logics, semantics, absolute and relative approaches.

1 Introduction

The introduction of special temporal connectives is considered very useful for several applications in Computer Science. To give a formal account in these applications of the intended meanings of the connectives, first order logic is used as a model theory. In our opinion, this approach implies several difficulties:

- It obscures the nature of the time that we have in mind (and therefore the intuitive meaning of the connectives).
- It makes difficult to consider approaches that treat points and intervals jointly and to combine absolute and relative temporal information.

- Furthermore, it renders difficult the proof of the validity of the formulas and the search for natural and efficient proof systems for temporal logics (particularly, it is difficult to build automatic theorem provers that reflect the semantics and that allow model generation).

Because of the previously mentioned shortcomings, we introduce a new semantics, which we call *topological semantics* because it reflects the topology of the flow of time.

Using this common semantic framework, we introduce a point logic over discrete time (the LN logic [5]), a point logic over continuous time (the IRLN logic [7]), and an interval-point logic over discrete time (the LNint logic [13]).

The connectives introduced in these logics consider the fact that every application of the temporal logic in Computer Science is intrinsically concerned with the possibility of testing the first (last) occurrence of an event after (before) the instant in which we are speaking—or executing. So, they reflect the relations of precedence, posteriority and simultaneity.

The well-behavior of our semantic approach has been shown in the proof of the Separation Theorem for LN [5] and in the design of efficient and parallel automatic theorem provers for several systems in temporal logics [6].

2 The LN Logic

The LN logic is a linear temporal logic over discrete time. Its well formed formulas are built on the following alphabet

- An enumerable set Ω of atoms.
- The usual classical connectives \neg, \wedge, \vee .
- The temporal connectives \preceq, \succ .

The well formed formulas (wffs) of LN are inductively defined as follows:

- An atomic $p \in \Omega$ is a wff.
- Let A and B be wffs, then $\neg A, A \wedge B, A \vee B, A \rightarrow B, A \preceq B$ and $A \succ B$ are wffs.

$A \preceq B$ is read as *sometime in the future A, and the next occurrence of A will be before or simultaneous with the next occurrence of B*.

*This work has been partially supported by CICYT project TIC94-0847-C02-02.

$A \succ B$ is read as *sometime in the past A occurred, and the last occurrence of A was after or simultaneous with the last occurrence of B*.

Topological semantics of LN

We define a topological semantics for LN considering (\mathbb{Z}, \leq) as the flow of time, where \mathbb{Z} is the set of integer numbers with the *smaller than* relation.

The key concepts of this approach reflect the *well order* of \mathbb{Z} and are the m_{tA}^+ and m_{tA}^- defined below:

Definition 1: Given a temporal formula A and an instant of time $t \in \mathbb{Z}$, we define

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

We will convey that $\min \emptyset = +\infty$ and that $\max \emptyset = -\infty$, and we shall consider the *smaller than* relation extended as usual to $\mathbb{Z} \cup \{+\infty\} \cup \{-\infty\}$.

These concepts represent the first instant after t in which A is true (m_{tA}^+) and the last instant before t in which A was true (m_{tA}^-). This is why this logic is called LN (Last-Next). m_{tA}^+ and m_{tA}^- are powerful tools to obtain the desired advantages: to give the interpretation for the temporal connectives of LN in such a way that directly reflects their intuitive meaning and to facilitates the use of temporal logic in applications.

Definition 2: We define a *temporal interpretation* for LN to be a function that associates with each atom $p \in \Omega$, a subset $h(p)$ of \mathbb{Z} :

$$h: \Omega \longrightarrow 2^{\mathbb{Z}}$$

Informally, $h(p)$ is to be thought of as the set of time points $t \in \mathbb{Z}$ at which p is true.

The function h can be extended to any wff of LN as follows:

1. $h(\neg A) = \mathbb{Z} \setminus h(A)$
2. $h(A \vee B) = h(A) \cup h(B)$; $h(A \wedge B) = h(A) \cap h(B)$
3. $h(A \rightarrow B) = (\mathbb{Z} \setminus h(A)) \cup h(B)$
4. $h(A \preceq B) = \{t \in \mathbb{Z} \mid m_{tA}^+ < +\infty \text{ and } m_{tA}^+ \leq m_{tB}^+\}$
5. $h(A \succ B) = \{t \in \mathbb{Z} \mid m_{tA}^- > -\infty \text{ and } m_{tA}^- \geq m_{tB}^-\}$

where:

$$\begin{aligned} m_{tA}^+ &= \min((t, +\infty) \cap h(A)) \\ m_{tA}^- &= \max((-\infty, t) \cap h(A)) \end{aligned}$$

Definition 3: A formula A in LN is *valid* in a *temporal interpretation* h , denoted $\models_h A$, if $h(A) = \mathbb{Z}$. A formula A is called *valid*, denoted $\models A$, if $\models_h A$ for all interpretations h . The formulas A and B are called *equivalent*, denoted $A \equiv B$, if $A \equiv_h B$ for any interpretation h .

Theorem 1 *The logics LN is expressively complete over linear discrete time [5].*

This primitive system allows us to derive other connectives which reflect other precedence situations and simultaneity (we present in the table 1 only the future connectives).

3 The IRLN Logic

The choice of a discrete time or a continuous time temporal logic depends on the particular problem we are dealing with. Since most of these problems are tractable with temporal logics over discrete time, most of the temporal logics have a discrete flow of time [1], [8]. Nevertheless, in our opinion, the temporal logics over continuous time may not be rejected. In fact we may cite some authors that confirm this assertion: [4], [17], [9], in these works temporal logics over continuous time are presented, and their utility is discussed.

The extension we propose is a further step in the sense that we look for a natural extension of the LN logic to continuous time (the IRLN Logic); i.e., we want a temporal logic over continuous time that verifies the properties of LN: to have natural connectives for specifications and to avoid the use of first order logic in the semantics.

In particular, we think that IRLN is suitable for the specification of the so called *reactive systems*. These are systems that continuously interact with their environment. For instance almost all concurrent, distributed or embedded systems are reactive. In the reactive systems, a non-trivial mixture of *discrete* (program-like) and *continuous* (environment-like) components are allowed. There are some works saying that the temporal logics over continuous time and the reactive systems are good partners. ([14] is a pioneer in this area)

Finally, we may say that there are reactive systems that have some asynchronous components, and in this situation, the change of state does not occur with the ticks of a global clock (the non-negative integer \mathbb{N}); so, we will need a dense and continuous set to model the flow of time (the real number set \mathbb{R}).

3.1 Syntax of IRLN

The well formed formulas of IRLN are built on the following alphabet

- An enumerable set of atoms, Ω and the usual classical connectives : \neg, \wedge, \vee and \rightarrow .
- A set of symbols of temporal connectives, \mathcal{C}_T , whose elements will be discussed in the rest of this section.

Once the \mathcal{C}_T set was introduced, the *wffs* are defined like those presented for LN.

Connective	Description	Definition	Semantics
$A \prec B$	strong strict precedence	$A \prec B \wedge \neg(B \prec A)$	$m_{tA}^+ < m_{tB}^+$
$A \approx^+ B$	strong simultaneity	$A \prec B \wedge B \prec A$	$m_{tA}^+ = m_{tB}^+ < +\infty$
$A \sqsubset B$	weak strict precedence	$\neg(B \prec A)$	$m_{tA}^+ < m_{tB}^+$ or $m_{tB}^+ = +\infty$
$A \sqsubseteq B$	weak wide precedence	$\neg(B \prec A)$	$m_{tA}^+ \leq m_{tB}^+$
$A =^+ B$	weak simultaneity	$A \sqsubseteq B \wedge B \sqsubseteq A$	$m_{tA}^+ = m_{tB}^+$

Table 1: Derived connectives in LN

Topological semantics of IRLN

The topological semantics of IRLN is the natural extension of the topological semantics of LN.

We consider (\mathbb{R}, \leq) as the time flow, where \mathbb{R} represents the set of real numbers and \leq is the *smaller than or equal to* relation.

The tools m^+ and m^- of LN must be fitted to the topology of \mathbb{R} :

For any wff A and any $t \in \mathbb{R}$ we define:

$$i_{tA}^+ = \inf\{t' \in \mathbb{R} \mid t' \geq t \text{ and } A \text{ is true at } t'\}$$

$$i_{tA}^- = \sup\{t' \in \mathbb{R} \mid t' \leq t \text{ and } A \text{ is true at } t'\}$$

We extend as usual the *smaller than or equal to* relation to the set $\mathbb{R} \cup \{-\infty, +\infty\}$. If A is true at i_{tA}^+ and $i_{tA}^+ \neq t$, then we denote it as m_{tA}^+ (it is a maximum). Analogously, if A is true at i_{tA}^- and $i_{tA}^- \neq t$, then we denote it as m_{tA}^- (it is a minimum).

This distinction is not only a theoretical constraint, it is a method that represents all the situations in which a statement might occur in real applications. In our opinion, this distinction is fundamental to applications of temporal logic because we need to represent naturally all the many different patterns of the occurrences of the statements in time and it allows to specify *environment facts* and *program facts* in reactive systems.

Definition 4: We define a *temporal interpretation* of IRLN to be a function, $h_{\mathbb{R}}$, that associates each atom, $p \in \Omega$, to a subset $h_{\mathbb{R}}(p)$ of \mathbb{R} : $h_{\mathbb{R}}: \Omega \rightarrow 2^{\mathbb{R}}$

The extension of $h_{\mathbb{R}}$ to any wff will be defined when we introduce \mathcal{C} .

3.2 Choosing Temporal Connectives

3.2.1 Monary Connectives

As we have mentioned above, we need to specify naturally both the internal programs facts and the external environmental facts. Thus, we need monary temporal connectives which adequately reflect both modes of occurrences of an event, this is, the connectives included in the table 2 (we present only the future connectives).

where: $i_{tA}^+ = \inf((t, +\infty) \cap h_{\mathbb{R}}(A))$ and if $i_{tA}^+ \in h_{\mathbb{R}}(A)$, then we note m_{tA}^+ .

The past connectives are defined in the same way using the topological past-tools i_{tA}^- and m_{tA}^- . From

now on, we will only introduce the future-related components of our temporal logic.

3.3 Several Systems of Connectives for IRLN

We present here three equivalent systems which are fully expressive.

System 1

The philosophy of this system is to extend directly the primitive system of LN: $\{\prec, \succ\}$ (see table 3).

The connective \prec is a *very strong connective of non-strict precedence*. This name reflects the fact that when we affirm $A \prec B$ we force the existence of the first future occurrence of A .

We look for a fully expressive set \mathcal{C}_T for IRLN. Therefore, we need a *weak* monary connective. So, together with \prec , we include a monary connective [7]. The set is now defined as follows:

$$\mathcal{C}_T^1 = \{\prec, \succ, Ac^+, Ac^-\}$$

Nevertheless, \mathcal{C}_T^1 does not reflect the simple and transparent behavior that it had in LN specifications:

1. There are some precedence situations in the real world that are not directly covered with this *too strong* precedence. (If we are not able to determine a first instant on which a proposition will occur. Examples are showed in [7]).
2. The precedence and simultaneity connectives do not match properly. (For example, the semantics of $\neg(A \prec B)$ does not render any connectives of precedence).

This system is a good step in our development, but it is not the end of the road.

System 2

To avoid these problems, we consider a new binary connective which reflects directly the precedence relation on the real line. The connective is shown in the table 4.

This connective does not require any monary connective to have full expressive power because in their definition the precedence connectives implicitly include the two ways of occurrences. So, the set \mathcal{C}_T^2 is now $\{\sqsubseteq, \sqsupseteq\}$.

We may do the following criticism for this system: the precedence and simultaneity connectives match

Connective	Meaning	Definition
$Ac^+(A)$	<i>A in the future A will be true at instants arbitrarily close</i>	$h_{\mathbb{R}}(Ac^+(A)) = \{t \in \mathbb{R} \mid i_{tA}^+ = t\}$
$Al^+(A)$	<i>in the future there will be a first accessible instant at which A will be true</i>	$h_{\mathbb{R}}(Al^+(A)) = \{t \in \mathbb{R} \mid m_{tA}^+ \text{ exists}\}$

Table 2: Monary connectives of IRLN

Connective	Meaning	Definition
$A \preceq B$	<i>in the future there will be a first instant at which A will be true, and it will take place before or simultaneously with any future occurrence of B</i>	$h_{\mathbb{R}}(A \preceq B) = \{t \in \mathbb{R} \mid m_{tA}^+ \leq i_{tB}^+\}$

Table 3: System 1 of IRLN

perfectly because they reflect the precedence and simultaneity relation over the real numbers. Nevertheless, it does not seem too adequate for real applications.

System 3

In practice, when we reason over continuous time, it is impossible to determine towards the future, when we will reach *the* instant at which *A* will happen or the instant at which occurrences of *A* will accumulate and this limits our reasoning. Therefore, to apply temporal logic over continuous time we must identify both situations. This is our new starting point to seek the appropriate system of connectives.

In opposition to system 1, the lack of expressiveness for this system is due to the *strong character*. So, we need the monary connective Al^+ and the set \mathcal{C}_T^3 is now $\{Al^+, Al^-, \sqsubseteq, \supseteq\}$ (see table 5).

In our opinion, we have successfully concluded our search for a system of connectives that covers sufficiently our objective. We may summarize its advantages below:

1. \mathcal{C}_T^3 has temporal connectives that naturally correspond to the different interpretations that represent properties of interest in real systems.
2. The situations reflected by the connectives are translated easily to order relations between instants of \mathbb{R} . In particular $\neg(A \supseteq B) = B \sqsubset A$.
3. IRLN is appropriate for applications that require both program and environmental events to be integrated in the specification.
4. IRLN permits the successful extension of the automated theorem prove methods used in [2] and [6] for propositional classical logic and temporal logic respectively.

3.3.1 Other derivable connectives

All of these systems allow us to define simultaneity and severe precedence connectives as naturally as LN permits. In fact, we can present two *definition schemata* to introduce this kind of connectives. Each one of these new connectives has the character of the connective used to define it. The schemata are:

$$\begin{aligned}
 A = B &\stackrel{\text{def}}{=} A \star B \wedge B \star A \\
 A < B &\stackrel{\text{def}}{=} A \odot B \wedge \neg(A = B)
 \end{aligned}$$

Where $=$ represent simultaneity, $<$ represent strict precedence and $\star \in \{\preceq, \sqsubseteq, \sqsupseteq\}$, $\odot \in \{\sqsubset, \sqsupset\}$. (The strict precedence connective corresponding to \preceq is only a bit more complex because it must consider the Ac^+ connective in its definition).

4 A point-interval logic

In this section we present a logic with which we pretend to combine different approaches found in the literature and that we consider artificially opposed. Firstly, we may refer the classical discussion between points and intervals. Until now, temporal logic of points has been the most commonly used in computer applications and there are many results and different implementations. Nevertheless, recently we have seen an increasing tendency to work with interval-based temporal logics because they appear to be more adequate for reasoning about *change*, for temporal reasoning in general, for scheduling and planning or for natural language processing. A point-based temporal logic has several advantages, perhaps the most important is that it inherits the well-demonstrated good computational behavior of point algebra ([16]).

On the other hand, when we begin the development of a temporal logic, we have to decide if we will treat time in an absolute or relative way. The

Connective	Meaning	Definition
$A \sqsubseteq B$	<i>if in the future B will occur, then there will be an occurrence of A before or simultaneously with any occurrence of B</i>	$h_{\mathbb{R}}(A \sqsubseteq B) = \{t \in \mathbb{R} \mid i_{tB}^+ = +\infty\} \cup \{t \in \mathbb{R} \mid m_{tA}^+ \leq i_{tB}^+\} \cup \{t \in \mathbb{R} \mid i_{tA}^+ < i_{tB}^+\}$

Table 4: System 2 of IRLN

Connective	Meaning	Definition
$A \sqsubseteq\sqsubseteq B$	<i>if in the future B will occur, then there will be occurrences of A before any occurrence of B, or it will be indeterminate if A precedes B or B precedes A</i>	$h_{\mathbb{R}}(A \sqsubseteq\sqsubseteq B) = \{t \in \mathbb{R} \mid i_{tA}^+ \leq i_{tB}^+\}$

Table 5: System 3 of IRLN

absolute approach involves treating temporal assertions according to the date in which they occur as, for example, in the logics proposed by McDermott [11] and Allen [3]. This approach is typically followed by reified logics and logics with temporal arguments. On the other hand, when we consider relativity, time is not absolutely defined, but the passage of time is marked by occurrences. Most modal logics, like the USF logic [8], *Tempura* [12], or that proposed by Halpern and Shoham [10] adopt this relative approach.

However, we conclude that we really did not have to choose between one kind of temporal logic or another because, in practice, each or both logical forms can satisfy naturally the needs of the application we have in mind.

Our approach is to have a logic that treats points and intervals jointly, which allows both a relative and an absolute treatment of time.

4.1 The Logic LNint

LNint is a modal temporal logic for handling points and intervals. At the syntactic level, we begin considering two components, one to collect assertions about: points, points that belong to intervals and dates; and other collecting *events*. We use atomic events in a similar sense to that used by Allen, i.e., expressions about intervals which are true neither at the subintervals nor, more specifically, at the points of the interval over which the expression is affirmed. These initial components, although separated at first, will be extended later to collect mixed concepts, to reach finally a perfect semantic cohabitation. This cohabitation takes shape in the following idea: we talk about points or intervals in LNint, but we are always in an (evaluation) point, the *current instant* or present. Later, we abstract this instant. In this way, we can avoid the ambiguous concept of the *current interval* that the interval modal logics impose. The concept of current interval is not very intuitive and less com-

putationally manageable and appropriate.

LNint permits a good absolute treatment of time and to handle points and intervals (when needed) like temporal logics with temporal arguments, or like reified logics. In this way, we obtain a mixture of the absolute and relative approaches to the treatment of time.

The time flow in LNint is $(T, +, \leq)$ isomorphic to $(\mathbb{Z}, \leq, +)$.

4.1.1 Syntax of LNint

The construction of the language of LNint starts considering three sets of atoms:

- The set $\Omega_p = \{p, q, \dots, p_1, q_1, \dots, p_n, q_n, \dots\}$ of point atoms that formalize statements whose executions take place in an instant (*hereditary formulas* of Shoham [15]).
- The set $\underline{T} = \{\underline{t} \mid t \in T\}$ of *date* atoms, used to name instants.
- The set $\Omega_{int} = \{\alpha, \beta, \dots, \alpha_1, \beta_1, \dots, \alpha_n, \beta_n, \dots\}$ of atomic events.

Afterwards we define three sublanguages, each one gathering different expressions:

- I) The “events language” \mathcal{L}_{int} , in which we treat event expressions.

\mathcal{L}_{int} is the inductive closure of Ω_{int} under the boolean connectives, the monary temporal connectives $\langle ab^+ \rangle$, $\langle ab^- \rangle$, $\langle beg \rangle$, $\langle \widehat{beg} \rangle$, $\langle end \rangle$, $\langle \widehat{end} \rangle$, and the *absolute* connective $\langle \underline{m}, \underline{n} \rangle$.

These six unary connectives are sufficient to express any temporal relation between two intervals [15]. The absolute connective allows us to name dated intervals.

- II) The “points language” \mathcal{L}_p , in which we treat point expressions.

\mathcal{L}_p is based on the language of the LN logic, extended to talk about dates. So we add to the alphabet of LN the elements of \underline{T} .

In \mathcal{L}_p we can define an absolute connective that links a point formula and a date:

$$A \text{ at } \underline{m} = (A \wedge \underline{m}) \vee A \text{ atnext } \underline{m} \vee A \text{ atlast } \underline{m}$$

This connective tells us if a formula A is true at a given instant \underline{m} ; it is independent of the instant in which we are located, i.e., we deliberately ignore the current instant.

- III) The “points and events language” $\widehat{\mathcal{L}}_p$. To avoid the intrinsic use of the interval language, we need to extend \mathcal{L}_p to treat the events by using points. Of course, although it is true that events in themselves have no sense in the points of the interval over which we affirm them, it is also true that every event must have a *start instant* and an *end instant*, and must *take place* at every point between them. So we can now characterize events by means of these points. They are formalized in the extension of \mathcal{L}_p which we denote as $\widehat{\mathcal{L}}_p$ in which we define for each atomic event $\alpha \in \Omega_{int}$, $\uparrow\alpha$, $\downarrow\alpha$ and $\overrightarrow{\alpha}$ to denote the starting instant, the ending instant and a course instant, respectively.

Then we define Ω_p^I as follows:

$$\Omega_p^I = \Omega_p \cup \mathcal{T} \cup \{\top, \perp\} \cup \{\uparrow\alpha, \downarrow\alpha, \overrightarrow{\alpha} \mid \alpha \in \Omega_{int}\}$$

We define $\widehat{\mathcal{L}}_p$ as the inductive closure of Ω_p^I under the boolean connectives, \preceq and \succ .

Finally we extend the definition (in $\widehat{\mathcal{L}}_p$) of the start instant, the end instant and the course to the well-formed formulas of \mathcal{L}_{int} . This facilitates the effective manipulation of any event based on its start and end instants.

The language \mathcal{L} of LNInt.

To establish the final language, \mathcal{L} , of LNInt, we define a translation function Tr from \mathcal{L}_{int} to $\widehat{\mathcal{L}}_p$ as follows:

$$Tr : \mathcal{L}_{int} \rightarrow \widehat{\mathcal{L}}_p$$

$$\text{where } Tr(\mathcal{A}) = \uparrow\mathcal{A} \vee \overrightarrow{\mathcal{A}} \vee \downarrow\mathcal{A}.$$

With this function we have available “point versions” of the formulas of \mathcal{L}_{int} , and then we can treat events adequately in terms of points.

Now, we define \mathcal{L} adding to the language $\widehat{\mathcal{L}}_p$ the following defined formulas:

$$\mathcal{A} =_{\text{def}} Tr(\mathcal{A}) \text{ for all formula } \mathcal{A} \in \mathcal{L}_{int}$$

We can define in \mathcal{L} the desired connective to achieve an absolute temporal treatment of events: if \mathcal{A} is a well-formed formula in the sub-language \mathcal{L}_{int} , we define

$$\mathcal{A} \text{ at}_I [\underline{m}, \underline{n}] = (\uparrow\mathcal{A} \wedge \downarrow\mathcal{A} \approx^+ \underline{n}) \text{ at } \underline{m}$$

$\mathcal{A} \text{ at}_I [\underline{m}, \underline{n}]$ is read as *the event \mathcal{A} occurs exactly in the interval $[\underline{m}, \underline{n}]$.*

4.1.2 Semantics of LNInt

Since all the expressions of \mathcal{L} are really point expressions, the semantics is basically the topological semantics of the logic LN. The only difference is that the temporal interpretations must satisfy some natural requirements to give coherent truth values to the dates and to the start, end and course points of atomic events. For example, two of the requirements are:

$$h(\underline{t}) = \{t\}, \text{ for all } \underline{t} \in \mathcal{T}$$

$$h(\uparrow\alpha) \cap h(\downarrow\alpha) = h(\uparrow\alpha) \cap h(\overrightarrow{\alpha}) = h(\downarrow\alpha) \cap h(\overrightarrow{\alpha}) = \emptyset$$

References

- [1] M. Abadi and Z. Manna. Temporal logic programming. In *Symposium of Logic Programming*. IEEE, 1987.
- [2] G. Aguilera, I. P. de Guzmán, and M. Ojeda. TAS-D++: Syntactic tree transformations for automated theorem proving. In *European Conference on Logic and Artificial Intelligence JELIA '94*, volume 838. Lecture Notes in Artificial Intelligence, 1994.
- [3] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, pages 832–843, 1983.
- [4] H. Barringer, A. Pnuelli, and R. Kuiper. A really abstract concurrent and its temporal logic. In *POPL*. ACM, 1986.
- [5] A. Burrieza and I. P. de Guzmán. 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.
- [6] M. Enciso. *Lógica temporal y demostración automática de teoremas. Eficiencia y paralelismo*. PhD thesis, Universidad de Málaga. España, 1995.
- [7] M. Enciso and I. P. de Guzmán. Topological semantics for specification and verification: the RLN logic. Submitted to *Journal of Applied Non-classical Logics*, 1994.
- [8] D. M. Gabbay. *The declarative past and imperative future*, volume 398 of *Lecture Notes in Computer Science*. Springer Verlag, 1989.
- [9] D. M. Gabbay. *Temporal Logic: Mathematical Foundations*. Oxford University Press, 1992.
- [10] J. Halpern and Y. Shoham. A propositional modal logic of time intervals. In *First IEEE Symposium on Logic in Computer Science*. Computer Society Press, 1986.
- [11] D. V. McDermott. A temporal logic for reasoning about processes and plans. *Cognitive Science*, 6:101–155, 1982.

- [12] B. C. Moszkowski. *Executing Temporal Logic Programs*. Cambridge University Press, 1986.
- [13] I. P. de Guzmán and C. Rossi. LNint: a temporal logic that combines points and intervals and the absolute and relative approaches. To appear in *Bulletin of the IGPL*, 1995.
- [14] A. Pnueli. The temporal logic of programs. In *18th Symposium on the Foundations of Computer Science*. IEEE, 1977.
- [15] Y. Shoham. *Reasoning about Change. Time and Causation from the Standpoint of Artificial Intelligence*. MIT Press, 1988.
- [16] M. Vilain, H. Kautz, and P. van Beek. Constraint propagation algorithms for temporal reasoning: A revised report. In *Readings on Qualitative Reasoning about Physical Systems*, pages 373–381. Morgan Kaufmann, 1990.
- [17] P. Wolper. ESPRIT Basic Research Action REACT(6021): Building correct reactive systems. *EATCS Bulletin*, 50, 1993.